Erkan Uçar

1. Türk Yapay Zeka ve Yapay Sinir Ağları Sempozyumu

25 – 26 Haziran 1992
Bilkent Üniversitesi
Ankara

Bildiriler

Derleyenler

Kemal Oflazer Varol Akman
H. Altay Güvenir Uğur Halıcı

Destekleyen Kuruluşlar
Bilkent Üniversitesi – ODTÜ – Boğaziçi Üniversitesi
TÜBİTAK – IEEE Computer Society Türkiye Kolu

ÖNSÖZ

Önümüzdeki yıllarda insan yaşamını gittikçe daha fazla etkileyecek olan bilişim teknolojileri arasında belki de en önemlisi Yapay Zeka'dır. Yapay zekayı kısaca insanlarda zeka ile özdeşleştirdiğimiz bir takım özellikleri ve kabiliyetleri (örneğin görme, anlama, öğrenme, akıl yürütme, plan yapma, konuşma, v.b.) bilgisayarlar aracılığı ile gerçekleştirmeye çalışan bir bilim dalı olarak tanımlayabiliriz. ABD'nde 1950'li yılların sonuna doğru başlayan yapay zeka araştırmaları zamanla bir dizi konuda önemli başarılar elde etmiştir. Bir yandan çok önemli problemlerin çözümüne doğru değişik mantıksal yaklaşımlarla kuramsal temeller atılırken, diğer yandan olgunluğa ulaşmış bazı metodolojiler (örneğin uzman sistemler) son derece önemli uygulama alanları bulmuş ve oldukça da başarılı olmuşlardır. Günümüzde geliştirilen yapay zekalı sistemler artık büyük usta seviyesinde satranç oynayabilmekte, sınırlı bir kelime ve cümle dağarcığı ile de olsa konuşma anlayabilmekte, dilden dile çeviri yapabilmekte, tasarımlarda insan tasarımcılara yardımcı olmakta, matematiksel ispatlar yapabilmekte, görüntüleri inceleyip robot kollarını kontrol edebilmekte, yol görüntülerini inceleyip kısıtlı bir ortamda bir minibüsü idare edebilmektedirler. Ancak şu ana kadar yapılan araştırmaların ortaya çıkardığı bir başka gerçek de buzdağının suyun altında kalan kısmının çok büyük olduğu ve bir insan gibi algılayacak, düşünecek, çalışacak sistemleri gerçekleştirmenin daha çok uzun yıllar alacağıdır.

Yapay zekada son zamanlara dek birçok yaklaşım simgesel modelleme yöntemleri kullanmıştır. Ancak bu yöntemlerin çeşitli sorunlarının ortaya çıkması, araştırmacıları temel olarak insan beyninin yapısına benzeyen ve türlü nedenlerle 1960'larda terk edilmiş bazı modelleri yeniden kullanmaya yöneltti. Yapay sinir ağları olarak bilinen ve birbirleri ile çok değişik şekillerde bağlanmış çok fazla sayıda yapay sinirden oluşan bu modeller özellikle 1980'lerin sonlarında bir dizi önemli problemin modellenmesinde başarı ile kullanıldılar. Bu başarılar sözü edilen alandaki çalışmalarda son derece hızlı bir artışa neden oldu.

Ülkemizde bu konularda çalışan araştırmacıların olduğunu düşünerek onları bir araya getirmek ve yaptıkları çalışmaları duyurmak amacı ile bu sempozyumu düzenlemeyi düşündük. Yaptığımız duyurulara yurt içinden ve dışından beklentimizin üzerinde bir ilgi geldi. Katılımcıların çeşitli alanlardaki bildirilerini ilerdeki sayfalarımızda bulacaksınız. İçten isteğimiz bu sempozyumun giderek artan bir ilgi ile her sene tekrar edilmesi, ve ülkemizdeki yapay zeka ve yapay sinir ağları araştırmacılarının birbirleri ile iletişim sağladıkları vazgeçilmez bir ortam haline gelmesidir.

Saygılarımızla,

Kemal Oflazer
Varol Akman
H. Altay Güvenir
Uğur Halıcı

TEŞEKKÜR

Böyle bir sempozyumun düzenlenmesi en başta bir grup çalışması gerektiren bir etkinliktir. Fikrin oluşmasından sempozyumun gerçekleşmesine kadar geçen sürede bir çok arkadaşımız bize yardımcı oldu. Sempozyum fikrinin ve kapsamının tartışma dönemlerinde Boğaziçi Üniversitesi Bilgisayar Mühendisliği Bölümü'nden Dr. Ethem Alpaydın ile değerli görüş alış verişlerinde bulunduk. Bildiri çağrılarının hazırlanmasında, gelen bildirilerin kayıt edilmesi ve takibinde, bildirilerin son hallerinin bir araya getirilmesinde Bilkent Üniversitesi Bilgisayar Mühendisliği ve Enformatik Bilimleri Bölümü asistanlarından Erkan Tın ve Erkan Uçar son derece özverili çalıştılar. Bilkent Üniversitesi Güzel Sanatlar Fakültesi Dekanı ve Bilgisayar Mühendisliği ve Enformatik Bilimleri Bölümü öğretim üyesi Prof. Bülent Özgüç ise bildiriler kitabının kapak tasarımında büyük katkılarda bulundu. Sempozyumun gerçekleştirilmesine en büyük maddi katkıyı TÜBİTAK Mühendislik Araştırma Grubu yaptı. IEEE Türkiye Şubesi bildiri çağrılarının ve sempozyum programının Türkiye'deki üyelerine ulaştırılmasını sağladı. Bilkent Üniversitesi de çeşitli olanaklar sağlayarak bu sempozyumun gerçekleşmesine büyük katkıda bulundu.

Çok teşekkür ederiz.

Kemal Oflazer
Varol Akman
H. Altay Güvenir
Uğur Halıcı

İÇİNDEKİLER

Davetli Bildiri / Invited Paper

Akıl Yürütme / Reasoning

Yapay Sinir Ağları Uygulamaları / Applications of Artificial Neural Networks

Doğal Diller - Bilgi Gösterimi / Natural Languages - Knowledge Representation

Konuşma Tanıma / Speech Recognition

Yapay Sinir Ağları Yapıları / Artificial Neural Network Architectures

Tasarımda Yapay Zeka / Artificial Intelligence in Design

Örüntü Tanıma / Pattern Recognition

Görme ve Görüntü İşleme / Vision and Image Processing

Yapay Sinir Ağları Uygulamaları / Applications of Artificial Neural Networks

Davetli Bildiri

Invited Paper

TOWARDS PROGRAM UNDERSTANDING SYSTEMS

Tuncer I. Ören
Department of Computer Science, University of Ottawa
Ottawa, On. Canada K1N 6N5

ABSTRACT

Some philosophic and artificial intelligence concepts on understanding are reviewed and fundamental definitions are given. Recent studies in the field are highlighted. Research issues in program understanding are presented in a systematic way.

ÖZET

Program anlama konusunda ilgili felsefe ve yapay zeka kavramları gözden geçirilip temel bazı tarifler verilmiş ve yakın geçmişte yapılmış bazı çalışmalardan bahsedilmiştir. Program anlama konusundaki araştırma konuları sistematik bir biçimde sunulmuştur.

1. INTRODUCTION

Tools to understand computer programs can be very useful in software maintenance and reengineering to satisfy practical goals such as:

1. Automated visualization and documentation, including automated documentation of programs in a natural language.
2. Answering questions about programs. This functionality requires ability to summarize and filter relevant knowledge. (Queries and/or answers can also be in natural language including speech).
3. Automated certification of certain characteristics (or lack of them).
4. Computer-aided verification and validation of programs.
5. Offering criticism about user programs and justifying them as well as providing recommendations to enhance a citicised program. (Including tutoring student programs).
6. Program explanation.

7. Extracting or building reusable components.

A program understanding tool can be a static or dynamic tool. The first type analyzes a program without executing it, while the second type monitors it during its execution.

Different types of knowledge can be discriminated by a program understanding system. These include, for example, discrimination of application domain entities, design decisions, high level abstractions, I/O behavior, rules, plans, policies, and inhereitence relations. In special purpose programs such as simulation programs, knowledge about the elements of static structure of a simulation program such as input, state, output, auxiliary, and interpolated variables, constants, parameters, auxiliary parameters, and tabular functions; the elements of the dynamic structure such as the state transition and output functions; and the experimental conditions can be discriminated.

Böcker et al. (1991, p. 274) reiterate the fundamental difference between the two points of view in computer science. Point of view 1 states that computer science is a formal mathematical discipline, while the second view states that it is an experimental discipline. The consequences of this distinction is very important. View 1 requires that main tools for the development are formal specification techniques, the basic challenge is to ask humans to think clearly and without logical errors, and that the programming methodology would insist on formal verification of specifications before they are converted into programs. The second view, acknowledging the knowledge processing abilities of humans, relies on the development of better tools (Newell and Simon, 1976; Fisher and Böcker, 1983). Program understanding aims to provide such advanced tools and environments.

The aim in this article is to review some philosophic and artificial intelligence concepts on "understanding," provide some definitions in program understanding, and present some studies that the author is involved in the development of some tools for program understanding.

2. PHILOSOPHICAL BACKGROUND

The concept of "understanding" is a fundamental issue in philosophy. See for example, Locke (1984 - original publication: 1690), Leibniz (1985 - original publication: 1765), and Russell (1948). Here, to provide a relevant background, only a few references are cited.

Whitehead distinguishes different types of understanding in chapter 3 of his treatise on Modes of Thought (Whitehead (1968 - original publication: 1938). Paraphrasing him we have the following three definitions.

Internal understanding of a system involves the notion of composition and refers to the system's elements and to their relationships. (Whitehead 1968, pp. 45-46)

External understanding treats the system as a unity and refers to its relationships with its environment. (Whitehead 1968, pp. 45-46).

Logical understanding starts with the details and passes to the construction achieved. (Whitehead 1968, pp. 61).

Some additional notes about understanding are af follows: Understanding is not primarily based on inference. Understanding is self-evidence. Thus inference is used as a means for the attainment of an understanding. Proofs are the tools for the extension of imperfect self-evidence. (Whitehead 1968, pp. 50). "Human understanding requires the adherence to some judicious abstraction, and the development of thought within that abstraction." (Whitehead 1968, pp. 55).

Understanding has two modes of advance: the gathering of detail within assigned pattern, and the discovery of novel pattern with its emphasis on novel detail." (Whitehead 1968, pp. 58).

In chapter 9 of "How we Think," Dewey (1991 - original publication : 1910) covers "Meaning: or Conceptions and Understanding." He clarifies that to say that you do not understand something and that it has no meaning are equivalent. (Dewey, 1991, p. 117).

He distinguishes two types of understanding which are direct and indirect understandings. *Direct understanding* is apprehension. *Indirect understanding*, called comprehension, is mediated understanding. (Dewey, 1991, p. 120).

3. ARTIFICIAL INTELLIGENCE BACKGROUND

Biermann (1990, pp.377-394) gives an example of how a system could "understand" that an object is a chair. Such a system would have a knowledge base about a chair where the knowledge is expressed in terms of a semantic network. Based on the explanation of the system's characteristics, he gives the following definition of understanding:

The understanding of a perception "...with respect to a body of knowledge involves finding a set of self-consistent links between the parts of the knowledge structure and the parts of the perceived data. After such a linkage is made, the intelligent being can follow arcs in its knowledge base to obtain innumerable useful facts, the name of the perceived objects, the names of its many parts, their relationships to each other, the uses of the object, and all other information available in its knowledge base." (Biermann, 1990, p. 386).

"Reasoning is the process of finding or building a linkage from one entity in memory to another. There must be an initial entity, a target entity, and a way of choosing paths from the initial entity toward the target." (Biermann, 1990, p. 402).

4. SOME PROGRAM UNDERSTANDING SYSTEMS

Early references in program understanding are Basili and Mills (1982), Brooks (1983), and Johnson and Soloway (1983). Some empirical studies in program understanding are due to Letovski (1986) and Wiedenbeck (1986).

Böcker et al. (1986, 1991) stress program visualization in program understanding systems. They have developed tools for the visualization of data structures, control structures, object-oriented formalisms, and directed graphs. One of their systems, KAESTLE is a graphic editor for list strucures in LISP. It can be used to generate and edit the list structures of a LISP program . Another tool, FOOSCAPE, displays the static calling structure as well as dynamic behavior of a programs. ZOO is a knowledge acquisition tool for Smalltalk environment. TRACK is an extension of FOOSCAPE in object-oriented programming. TRISTAN is a tool for the visualization of directed graphs.

Ören et al. concentrated on the application of program understanding to simulation programs and to object-oriented programs written in C++. E/Slam (Elucidation of Slam programs) accepts as input a Slam II program and documents it by a series of statement and program oriented templates. The latter provides summaries of the program according to certain criteria (Ören, 1989; Ören et al. 1990a,b,c, 1992). A recent development is to modularize the natural language documentation process. An analyzer generates knowledge stored in a family of tables. A program generator based on the specification of the types of tables and the desired report generates a customized documentation program which can generate natural language documentations (King et al. 1992). Hamilton (1990) working with the author, implemented a tool, NLC++ (Natural language documentation of C++ programs) which generates english documentation of C++ programs on a PC.

Van Sickle (1992) announced a forthcoming workshop on AI and automated program understanding.

4. SOME CONCEPTS AND DEFINITIONS ON PROGRAM UNDERSTANDING

Program understanding is a model-based and knowledge-intensive activity. It requires an a priori model of the entity under scrutiny. The granularity of the model will limit the granularity of the understanding.

A system **A** can understand a system **B**, if (1) **A** has a model **C**, of **B**, (2) **A** can analyze **B** to find its elements and their relations, and (3) **A** can establish links between the elements as well as relationships of **B** and **C**.

Once the mapping between **B** and **C** is achieved, i.e., once **A** understands **B**, the knowledge of **A** can be used by other knowledge processing modules to perform several knowledge processing activities such as the ones enumerated in the introduction. For example, the following can be achieved:

1. *Visualization* and *documentation* of **B** can be done by providing knowledge about the elements and relationships of **B**. For this purpose, the documentation module can use knowledge collected as the result of the analysis of **B**, knowledge stored in the model **C**, as well as the discrepancies between the knowledge stored in **C** and generated from **B**.

2. A *critique* of **B** can be provided based on the discrepancies between the knowledge stored in **C** and generated from **B**.

3. *Enhancement* of **B** can be achieved by modifying **B** to remedy the perceived deficiencies.

5. RESEARCH ISSUES IN PROGRAM UNDERSTANDING SYSTEMS

Major research issues in program understanding are systematized in Figure 1. They are as follows:

1. Extend the scope of program understanding by identifying new goals for program understanding.
2. For each type of programming (e.g., functional, declarative, object-oriented, structured, or special purpose such as simulation), identify:
   - 2.1 What must be understood (i.e., what are the elements and which of their relations must be understood?)
   - 2.2 What should be the levels of abstractions and granularity of program understanding?

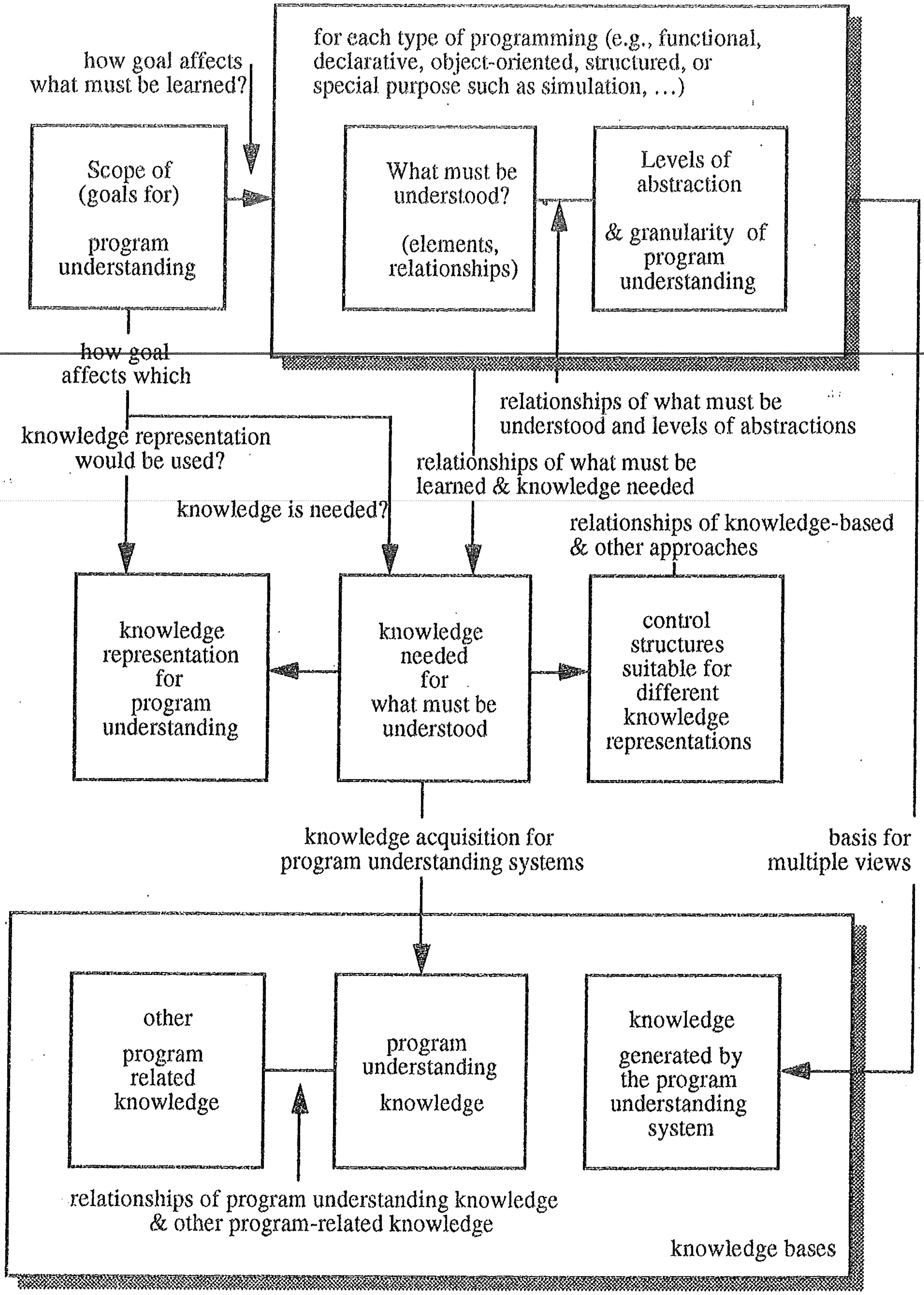

how goal affects what must be learned?
for each type of programming (e.g., functional, declarative, object-oriented, structured, or special purpose such as simulation, ...)
Scope of (goals for) program understanding
What must be understood? (elements, relationships)
Levels of abstraction & granularity of program understanding
how goal affects which knowledge representation would be used?
knowledge is needed?
relationships of what must be understood and levels of abstractions
relationships of what must be learned & knowledge needed
relationships of knowledge-based & other approaches
knowledge representation for program understanding
knowledge needed for what must be understood
control structures suitable for different knowledge representations
knowledge acquisition for program understanding systems
basis for multiple views
other program related knowledge
program understanding knowledge
knowledge generated by the program understanding system
relationships of program understanding knowledge & other program-related knowledge
knowledge bases

Figure 1. Major research issues in program understanding

3. What knowledge is needed for what must be understood? (i.e., what is the knowledge understanding knowledge?) (Ören 1990).
4. What are the knowledge representation schemes best suited (with respect to which criteria) of the program understanding knowledge?
5. Which control structures are suitable for different knowledge representations?
6. What are the elements of knowledge base(s) for program understanding knowledge?
7. What are the elements of knowledge base(s) for other program-related knowledge?
8. What are the elements of knowledge base(s) to store knowledge generated by program understanding systems? (For increasing the level of abstraction from program level, to user, user group, programming type levels, for example).
9. Determine how goal affects what must be learned?
10. Determine the relationships of what must be learned and levels of abstractions.
11. Determine how goal affects which knowledge is needed for what must be done?
12. Determine how goal affects which knowledge representation would be used?
13. Determine the relationships of what must be understood and levels of abstraction. (Are there any transformation rules between them?)
14. Determine the relationships of knowledge-based and other approaches for the control structures suitable for different knowledge representations.
15. Determine the knowledge acquisition needs of program understanding systems.
16. Determine the relationship of program understanding knowledge and other program-related knowledge.
17. Determine the knowledge base requirements of the knowledge generated by the program understanding system. (Determine the types of knowledge and how they should be processed, for example for associative knowledge processing and for learning as applied to program understanding).
18. Extend the concepts of program understanding to large programs and multi programs.

6. CONCLUSION

Software systems are becoming larger and more and more complex. Their documentation, maintenance, reengineering, and reverse engineering activities necessitate advanced tools. Program understanding tools and environments are very promising new set of software engineering tools and environments for these types of problems. In this article are discussed some basic concepts of understanding and program understanding, some existing systems and a systematic discussion of the needed research.

REFERENCES

Basili, V.R., Mills, H.D. (1982). Understanding and Documenting Programs. IEEE Transactions on Software Engineering, SE-8: 270-283.

Biermann, A.W. (1990). Great Ideas in Computer Science. MIT Press, Cambridge, MA.

Böcker, H.-D., Fischer, G., Nieper-Lemke, H. (1986). The Enhancement of Understanding Through Visual Representations. In: Human Factors in Computing Systems, CHI'86 Conference Proceedings. ACM, New York, 44-50.

Böcker, H.-D., Fischer, G., Nieper-Lemke, H. (1991). The Role of Visual Representations in Understanding Software. In: Artificial Intelligence and Software Engineering, D. Partridge (ed.). Ablex, Norwood, NJ, 273-290.

Brooks, R. (1983). Towards a Theory of the Comprehension of Computer Programs. International Journal of Man-Machine Studies, 18: 543-554.

Dewey, J. (1991). How We Think. Prometheus Books, Buffalo, NY. (Originally published: D.C. Heath, Lexington, MA, 1910).

Fisher, G. (1987). A Critic for LISP. In: Proc. of the 10th IJCAI (International Joint Conference on Artificial Interlligence), J. McDermott (ed.), Morgan Kaufman, Los Altos, CA., 177-184.

Fisher, G., Böcker, H.-D. (1983). The Nature of Design Processes and Computer Systems Can Support Them. In: Integrated Interactive Computing Systems, P. Degano and E. Sandewall (eds.), Proc. of the European Conference on Integrated Interactive Computing Systems (ECICS 82), North-Holland, Amsterdam, 73-88.

Hamilton, I.J. (1990). A Graphic Representation Scheme and a Tool for Natural Language Documentation of C++ Programs. M.Sc. Thesis, Computer Science Department, University of Ottawa, Ottawa, On., Canada.

Johnson, W.L., Soloway, E. (1983). PROUST: Knowledge-Based Program Understanding. In: Proceedings of the 7th International Conference on Software Engineering. IEEE, Orlando, FL.

King, D., Ören, T.I., Hitz, M. (1992 - In Press). Automatic Generation of Natural Language Documentation for SLAM II Programs. In: IMACS Transactions on Scientific Computing '91, E.N. Houstis and J.R. Rice (eds.), North-Holland

Leibniz, G.W. (1985 - original publication: 1765). New Essays on Human Understanding. University Press, Cambridge, England.

Letovsky, S. (1986). Cognitive Processes in Program Comprehension. In: Empirical Studies of Programmers, E. Soloway and S. Iyengar (eds.),Ablex Norwood, NJ, 58-79.

Locke, J. (1984 - original publication 1690 date). An Essay Concerning Human Understanding. Collins, Glasgow, UK.

Newell, A., Simon, H.A. (1976). Computer Science as an Empirical Inquiry: Symbols and Search. CACM 19:3, 113-136.

Ören, T.I. (1989). Program Understanding Systems in Simulation. In: J.W. Rozenblit et al. (eds.) Proc. of the 4th AI and Simulation Workshop, IJCAI-11, Detroit, MI, August 21, 1989, pp. 110-111.

Ören, T.I. (1990). A Paradigm for Artificial Intelligence in Software Engineering. In: Advances in Artificial Intelligence in Software Engineering - Vol. 1, T.I. Ören (Ed.), JAI Press, Greenwich, Connecticut, pp. 1-55.

Ören, T.I., Abou-Rabia, O., King, D.G., Birta, L.G., Wendt, R.N. (1990a) (Plenary Paper). Reverse Engineering in Simulation Program Understanding. In: Problem Solving by Simulation, A. Jávor (Ed.) Proceedings of IMACS European Simulation Meeting, Esztergom, Hungary, August 28-30, 1990, pp. 35-41.

Ören, T.I., Birta, L.G., Abou-Rabia, O., King, D.G., Wendt, R.N. (1990b). E/Slam: A Software Understanding Environment for SLAM II Programs. In: Proceedings of European Simulation Multiconference, Erlangen-Nuremberg, Germany, June 10-13, 1990, SCS International, San Diego, CA, pp. 235-240.

Ören, T.I., Wendt, N.R., Abou-Rabia, O., King, D.G., Birta, L.G. (1990c). A Reverse Engineering Application for Simulation Program Understanding. In: Proceedings of the

Bell Canada Quality Engineering Workshop II, Montréal, PQ, October 4-5, 1990. (Participation by invitation).

Ören, T.I., Wendt, R.N., King, D.G., Abou-Rabia, O., Birta, L.G. (1992 - In Press). A Template-Oriented Approach for Simulation Program Understanding. Mathematics and Computers in Simulation.

Russell, B. (1948). Human Knowledge — Its Scope and Limits. George Allen and Unwin, London, England.

Van Sickle, L. (ed.) (1992 - In Preparation). Proceedings of the AAAI-92 Workshop on AI & Automated Program Understanding, San Jose, CA.

Whitehead, A.N. (1968). Modes of Thought. The Free Press, New York. (Original publication: 1938).

Wiedenbeck, S. (1986). Processes in Computer Program Comprehension. In: Empirical Studies of Programmers, E. Soloway and S. Iyengar (eds.), Ablex Norwood, NJ, 48-57.

Akıl Yürütme

Reasoning

Proof-Checking Process-Based Reasoning About Physical Systems

Erkan Uçar and Varol Akman

Department of Computer Engineering and Information Science, Bilkent University, 06533
Bilkent, Ankara

Abstract

Having the most sophisticated set of modeling features and the most complete ontology, Forbus' Qualitative Process Theory allows better explanations of device behaviour than other systems because, for example, processes can explicitly be discussed. It can reason about state changes such as turning water to steam which can be at best implicitly represented in other systems. Forbus displays logical translations of selected process and individual view descriptions but does not attempt to axiomatize the whole theory. We believe that it is necessary to provide such a formalization. The Boyer-Moore Theorem Prover is one of the best media to do so.

1. INTRODUCTION

The behavior of a physical system can be described by the exact *quantitative* values of its variables (forces, velocities, pressures, etc.) at each time instant. Such a description, although complete, fails to provide sufficient explanation of how the system functions [12, 17, 18]. *Qualitative physics* is an alternative physics in which physical concepts are defined in a simpler yet formal basis [11]. In classical physics the characterization of physical change is defined within a nonmechanistic framework and thus is difficult to adapt to commonsense interpretation. Qualitative physics provides an alternative way of arriving, when possible, at the same conclusions and provides a simpler basis for reasoning about physical mechanisms [13].

The qualitative analysis of a system identifies all its possible behaviors (i.e., *envisioning*) [7]. This is crucial in many applications since envisioning highlights undesirable system modes that are to be avoided. Qualitative reasoning seems to provide a promising approach especially because it gives one the ability to reason with incomplete information [1, 2]. It can function with incomplete models or data whenever the information is not easily available. Qualitative analysis is more efficient than numerical simulation since it can be implemented with straightforward constraint satisfaction [16]. Forbus' *Qualitative Process Theory (QPT)* [8] is probably the best proposal in terms of its formality and approach. QPT describes the form and structure of naive theories about the *dynamics* of physical systems. QPT extends the ontology of commonsense physical models by introducing the important notion of *physical processes* [8, 9].

Since Forbus' framework is representable in first-order logic, we trust that it can be implemented using an automated theorem proving system. The *Boyer-Moore Theorem*

Prover (also known as *NQTHM*) [5] which operates on a logic that resembles an applicative subset of Lisp and provides *mathematical induction* as a rule of inference is such a system. We want to use NQTHM to make commonsense derivations from theories about the physical world coded in QPT.

2. QUALITATIVE PROCESS THEORY

QPT introduces physical processes as the mechanisms by which change occurs and provides a highly constrained account of physical causality (viz. all changes are due to a finite vocabulary of processes). A *physical process* is something that acts through time to change the parameters of objects in a situation. Objects' properties and states can change dramatically: heated water can start to boil, a rope breaks under the action of a great force, and so forth [14]. Different and meaningful states of objects need therefore to be represented. This is accomplished by *individual views (IVs)*. IVs are composed of objects and specify their state through a representation of their properties, called *quantities*. Each quantity consists of an *amount* and a *derivative*; both of them are qualitative descriptions of numbers. Relationships among quantities are modeled in terms of functional dependency by means of a *qualitative proportionality operator* ($\propto_Q$). Both processes and IVs become active only if their *preconditions* and *quantity conditions* are satisfied. These account respectively for physical and commonsense preconditions, e.g., a pot of water needs heat to boil (physical precondition) but also someone to place the pot on a stove (commonsense precondition).

Most importantly, the assumption is made that all changes in physical systems are caused directly or indirectly by processes whose direct effects are stated explicitly as *influences* and whose indirect effects are propagated via $\propto_Q$. As a consequence, the physics for a domain must include a *process vocabulary* that covers that domain. Thus a qualitative description is composed of a collection of objects, their states expressed by their quantities in the IVs, the relations holding among quantities, and the active processes. Another QPT structure worth mentioning is the so-called *encapsulated history (EH)*. Unlike a process, an EH explicitly refers to a particular sequence of times during which change takes place [13]. Encapsulated histories describe such phenomena as collisions between moving objects [14] whose description in terms of processes would be complicated [8].

Figure 1 illustrates a process specification for boiling and how it would look like when translated into predicate calculus [8, 9]. Boiling happens to a contained-liquid which is heated, and creates a gas made of the same stuff as the liquid. *T-boil* represents the boiling temperature for the piece of stuff involved. (For clarity, temporal references have been omitted from the axiomatic representation.) With each quantity, there is a special adder associated which is called its *InfluenceAdder*. Whenever a quantity is directly influenced, each direct influence is considered to be a member of the appropriate input set.

```
Process boiling

Individuals:
    w a contained-liquid
    hf a process-instance
        process(hf)=heat-flow
        dst(hf)=w

QuantityConditions:
    Status(hf,Active)
    A[temperature(w)]=A[t-boil(w)]

Relations:
    There is g ∈ piece-of-stuff
    gas(g)
    substance(g)=substance(w)
    temperature(w)=temperature(g)
    Let generation-rate be a quantity
    A[generation-rate] > ZERO
    generation-rate ∝Q+ flow-rate(hf)

Influences:
    I–(heat(w),A[flow-rate(hf)])
    I–(amount-of(w),A[generation-rate])
    I+(amount-of(g),A[generation-rate])
    I–(heat(w),A[generation-rate])
    I+(heat(g),A[generation-rate])
```

```
∀ w ∈ contained-liquid ∀ hf ∈ process-instance
(process(hf)=heat-flow ∧ dst(hf)=w ⟹
  [ ∃ pi ∈ process-instance
   process(pi)=boiling ∧ w(pi)=w ∧ hf(pi)=hf
  ∧ [(Status(hf,Active) ∧ A[temperature(w)]=A[t-boil(w)])
       ↔ Status(pi,Active)]
  ∧ [Status(pi,Active) ⟹
     [ ∃ g ∈ piece-of-stuff ∃ generation-rate ∈ quantity
        Boiling(w,hf)
        ∧ gas(g)
        ∧ substance(g)=substance(w)
        ∧ temperature(w)=temperature(g)
        ∧ A[generation-rate] > ZERO
        ∧ generation-rate ∝Q+ flow-rate(hf)
        ∧ A[flow-rate(hf)]
            ∈ MinusInputs(InfluenceAdder(heat(w)))
        ∧ A[generation-rate]
            ∈ MinusInputs(InfluenceAdder(amount-of(w)))
        ∧ A[generation-rate]
            ∈ PlusInputs(InfluenceAdder(amount-of(g)))
        ∧ A[generation-rate]
            ∈ MinusInputs(InfluenceAdder(heat(w)))
        ∧ A[generation-rate]
            ∈ PlusInputs(InfluenceAdder(heat(g)))]]]
```

Fig. 1 A boiling specification in QPT (on the left) and its first-order translation (on the right) [adapted from [9]].

The machinery which manipulates these modeling primitives operates as follows. Instantiating processes and individual views creates *individual view and process instances* with the help of a particular domain description of individuals which exist and a library of process and individual view definitions (the process and individual view *vocabulary*). Given these instances, the *process structure* can be created, i.e., the quantity and preconditions of these instances are examined to determine which instances are currently active.

Given a process structure, the influences (both direct and indirect) must be *resolved* to determine the derivatives of all the quantities. *Limit analysis* is a procedure that determines how quantities being influenced will change their amount, which may in turn lead to changes in the process structure because activation conditions may be altered. Finally, a complete *envisionment* can be built where each node has a unique process structure.

Limit analysis deserves a little more explanation. Most elements of a quantity space come from the quantity conditions of processes and individual views. They act as boundary conditions and are called *limit points.* For each changing quantity q (i.e. $s(D[q] \neq 0)$ all neighbouring points in quantity space to $A[q]$ are found. The signs of the derivatives of these quantities are examined to determine if inequality can change. The set of such changes and relative consistent combinations are the *quantity hypotheses.* Quantity hypotheses which change the process structure are called *limit hypotheses.*

Forbus was originally motivated to develop this representational apparatus in order to build a naive physical theory of the world. He created a library of general process and individual view descriptions which are to be used to model and then reason about physical situations [8, 9, 10].

3. BOYER-MOORE THEOREM PROVER AND PROOF-CHECKING QPT

NQTHM's logic is a quantifier-free first-order logic with equality [4]. NQTHM's logic forms a suitable basis for a theory coded in QPT (cf. Figure 1). The prover's language is a relative of Pure Lisp and consists of variables and function names combined in a prefix notation which is the exact pseudo form of the representation language of QPT. Basic functions of NQTHM are TRUE, FALSE, EQUAL, and IF. The logic also contains two *extension* principles under which the user can add new axioms to the system with the guarantee that any model for the logic can be extended to hold for the new axioms. *The Shell Principle* allows the user to add new axioms introducing new inductively defined data types [15]. Via this principle, any QPT domain (e.g., motion or liquids) can be modeled *The Principle of Definition* allows the user to define new functions in the logic. The rules of inference in the logic consist of *Propositional Calculus*, *Induction Principle*, and *Instantiation.*

NQTHM is automatic in the sense that once a lemma has been typed in, the user cannot interfere with the mechanical proof [5]. However, the user can *train* the prover by proving an appropriate sequence of lemmas that can then be used as *rewrite* rules. The prover also has a simple hint facility by which the user can *disable* function definitions, suggest instances of lemmas to be used and also the induction to be employed [6]. NQTHM generates an extensive commentary on the attempted proof. We believe that within the NQTHM formalism, implementing QPT will be one of the introductory steps of the proof-checking and mechanization of qualitative physics.

When a definition is proposed to NQTHM, before admitting it as a new axiom, certain conditions are mechanically checked to ensure that there exists a unique function satisfying this definition. The most important condition is that there exists a measure of the arguments of the function that is decreasing in a well-founded sense in each recursive call in the definition. The principle of definition can mechanically guess simple measures but more complicated ones can be supplied by the user. Once a candidate relation is found, NQTHM is invoked to prove theorems that are sufficient to admit the proposed definition.

Given a conjecture to prove, NQTHM performs many proof techniques under heuristic control. The main proof techniques are as follows:

Simplification – The system applies axioms, definitions, and previously proved theorems as rewrite rules to simplify expressions. To avoid looping, the simplifier contains heuristics to control the use of recursive definitions. The simplifier also contains decision procedures for propositional calculus, equality, and those formulas of Peano arithmetic that can be built up from variables, integers, $+$, $-$, $=$, and $<$. Figure 2 gives example Lisp code which illustrates a basic environment where we can apply these decision heuristics. If both of the quantities (X and Y) have positive signs then their magnitudes are the only criteria that resolve the inequalities. Even though this is implicitly determined in NQTHM, it is one of the basic qualitative restrictions that has to be implemented since our quantity space has its own characteristic polarity (i.e., $\text{SIGN}(X) \in \{-1, 0, 1\}$).

```
(DEFN SIGN (X)
      (IF (NEGATIVEP X)
          (IF (EQUAL (NEGATIVE-GUTS X) 0)
              0 -1)
          (IF (ZEROP X) 0 1)))

(DEFN MAGNITUDE (X)
      (IF (NEGATIVEP X)
          (NEGATIVE-GUTS X)
          X))

(PROVE-LEMMA INEQUALITY-5 (REWRITE)
      (IMPLIES (AND (EQUAL (SIGN X) 1)
                    (EQUAL (SIGN Y) 1))
               (AND (IFF (GREATERP (MAGNITUDE X) (MAGNITUDE Y))
                         (GREATERP X Y))
                    (IFF (EQUAL (MAGNITUDE X) (MAGNITUDE Y))
                         (EQUAL X Y)))))
```

Fig. 2 A rewrite lemma coded in NQTHM proving the inequality of two positive quantities. $[(X > \text{ZERO} \wedge Y > \text{ZERO}) \Longrightarrow (m[X] > m[Y] \leftrightarrow X > Y) \wedge (m[X] = m[Y] \leftrightarrow X = Y)]$

Elimination of undesirable function symbols – NQTHM uses axioms and previously proved lemmas to eliminate certain function symbols from the conjecture being proved. For example, it is a theorem that for each natural number i and each positive integer j there exist natural numbers $r < j$ and q such that $i = r + qj$. By replacing i with $r + qj$, the system can transform the expression $i \bmod j$ to r and i/j to q.

Strengthening the conjecture to be proved – Most of the time, it is the case that to prove some theorem by induction, it is necessary to prove a stronger theorem than that initially posed. NQTHM contains several heuristics for guessing stronger conjectures to prove. The heuristic involves "using" equality hypotheses by substituting one side for the other elsewhere in the conjecture and then strengthening the conjecture by "throwing away" the equality hypothesis.

Induction – When all else fails, it is useful to try mathematical induction. The selection of an "appropriate" induction is based on an analysis of the recursive functions mentioned in the conjecture. NQTHM's induction mechanism contains many heuristics for combining and choosing between the suggested inductions.

In the implementation of QPT we plan to use induction (when other heuristics fail) as a temporal mechanism. Therefore the commentary of the proofs will reflect the prediction and explanation of the physical system's behaviour at each time instant. Iteration on a formal time scale will be the best device to explain what the behaviour is converging to (e.g., in a situation where two containers are connected to each other from bottom and containing a liquid with different initial heights, a liquid flow will occur and the system will converge to a state where the liquid levels are the same).

4. CONCLUSION

QPT was developed to model systems in which there is no fixed topology of interaction and the set of objects in existence changes as time progresses. Procedurally, qualitative equations governing the acceptable behaviours are constructed dynamically at run time. This is achieved by summing the influences of various processes. This paper presented the need to formally define QPT within a logical framework. The idea is to use a proof-checker as a formal simulator [3] and to investigate the differences in the behaviours of physical systems coded in QPT.

References

[1] Akman, V., and ten Hagen, P. J. W. The Power of Physical Representations. *AI Magazine* **10**:49–65, 1989.

[2] Akman, V., and ten Hagen, P. J. W. Fronti Nulla Fides (No Reliance can be Placed on Appearance). Letter to the Editor. *AI Magazine* **11**:10–11, 1990.

[3] Boyer, R. S., Green, M. W., and Moore, J S. The Use of a Formal Simulator to Verify a Simple Real Time Control Program. *Beauty is Our Business: A Birthday Salute to Edsger W. Dijkstra.* ed. W. H. J. Feijen, 54–66, New York: Springer-Verlag, 1990.

[4] Boyer, R. S., and Moore, J S. Proof-Checking, Theorem-Proving, and Program Verification. *Contemporary Mathematics* **29**:119–132, 1984.

[5] Boyer, R. S., and Moore, J S. *A Computational Logic Handbook*, San Diego, CA: Academic Press, 1988.

[6] Boyer, R. S., and Moore, J S. Integrating Decision Procedures into Heuristic Theorem Provers: A Case Study with Linear Arithmetic. *Machine Intelligence 11*, Oxford University Press, 83–124, 1988.

[7] de Kleer, J. *Qualitative and Quantitative Knowledge in Classical Mechanics.* M.S. thesis, Artificial Intelligence Laboratory, Technical Report AI–TR–352, Massachusetts Institute of Technology, Cambridge, MA, 1975.

[8] Forbus, K. *Qualitative Process Theory.* Ph.D. thesis (Revised version), Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 1984.

[9] Forbus, K. Qualitative Process Theory. *Artificial Intelligence* **24**:85–168, 1984.

[10] Forbus, K. *The Qualitative Process Engine.* Technical Report UIUCDCS–R–86–1288, Department of Computer Science, University of Illinois at Urbana-Champaign, Champaign, Illinois, 1986.

[11] Forbus, K. Qualitative Physics: Past, Present, and Future. *Exploring Artificial Intelligence*, ed. H. Shrobe, 239–296, San Mateo, CA: Morgan Kaufmann, 1988.

[12] Forbus, K., and Falkenhainer, B. Self-Explanatory Simulations: An Integration of Qualitative and Quantitative Knowledge. *Proceedings of Eighth National Conference on Artificial Intelligence (AAAI-90)*, AAAI Press, 380–387, 1990.

[13] Hayes, P. The Second Naive Physics Manifesto. *Formal Theories of the Commonsense World.* eds. J. R. Hobbs and R. C. Moore, 1–36, Norwood, NJ: Ablex, 1985.

[14] Hendrix, G. G. Modeling Simultaneous Actions and Continuous Processes. *Artificial Intelligence* **4**:145–180, 1973.

[15] Kounalis, E., and Rusinowitch, M. Mechanizing Inductive Reasoning. *Proceedings of Eighth National Conference on Artificial Intelligence (AAAI-90)*, AAAI Press, 240–245, 1990.

[16] Kuipers, B. Qualitative Simulation. *Artificial Intelligence* 29:289–388, 1986.

[17] Söderman, U., and Strömberg, J. Combining Qualitative and Quantitative Knowledge to Generate Models of Physical Systems. *Proceedings of Twelfth International Conference on Artificial Intelligence (IJCAI-91)*, San Mateo, CA: Morgan Kaufmann, 1158–1163, 1991.

[18] Uçar, E., and Akman, V. Modeling An Oscillator: A Case Study. *Proceedings of Sixth International Symposium on Computer and Information Sciences (ISCIS-VI)*, eds. M. Baray, and B. Özgüç, Volume II, Amsterdam: Elsevier, 671–679, 1991.

An extension of the corresponding value technique in qualitative modeling and simulation

A. C. Cem Say and Selahattin Kuru

Department of Computer Engineering, Boğaziçi University, Bebek 80815 İstanbul, Türkiye.

Abstract

Qualitative simulation programs use tuples of corresponding values of system parameters in order to represent additional information about the relationships among the parameters of the system under consideration. Presently available qualitative simulators [2,5] require that the values appearing in these tuples be points. We show that, by allowing intervals in corresponding value tuples, even more information about the relationships of the system can be represented. This technique eliminates a class of spurious predictions of the QSIM algorithm, which other reported methods cannot detect.

1. BACKGROUND

A number of programs and methodologies for performing various forms of qualitative reasoning about physical systems have been developed [1,2,5,9,10]. In accordance with the underlying principles of naive physics [4], these programs adopt an epistemological scheme in which unknown or irrelevant details about the values and relationships in the modeled physical system are not represented. Parameter values are shown only in terms of their ordinal relationships with previously designated "important" magnitudes (*landmarks*.) The ordered set of landmarks of each parameter is called its *quantity space*, and a parameter magnitude is either a landmark or an open interval between two adjacent landmarks in the quantity space. Exact forms for the functional relationships among the parameters of the system are not given. Generally, such functions are restricted to belong to the families of strictly increasing or decreasing monotonic functions.

To be able to represent more about the "shapes" of these functions, while staying within the limits of the qualitative representation, Forbus [2,3] introduced the technique of using tuples of *corresponding values* of related parameters for such relationships. A corresponding value tuple links a landmark of one parameter with a landmark of the other parameter, therefore fixing a point in the "plot" of the function among the parameters. Consider the classic example where we want to represent the relationship between the amount of liquid in a tank and the pressure at the bottom of the tank. It is known that a strictly increasing monotonic function among these parameters exists. Using the notation and terminology of Kuipers' QSIM algorithm [5], it is known that there is an M+ constraint linking them, i. e.

$$M+(\text{amount, pressure})$$

holds.

However, this constraint alone is not sufficient to represent our (admittedly incomplete) knowledge about the considered relationship. Some of the functions that are mapped to this constraint can be seen in Figure 1. Knowing that these two particular parameters should never have negative values, consider only the first quadrant in that figure. Clearly, none of the plotted functions is satisfactory for our example, since they depict cases where one parameter can have the value zero while the other one is nonzero. Qualitative reasoners using only this information about this relationship will produce results which do not correspond to reality.

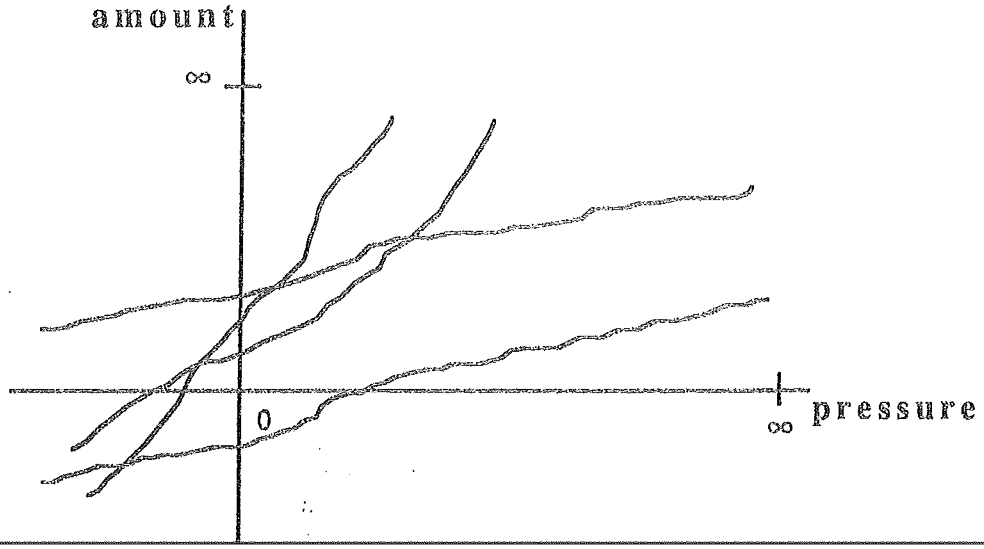


Figure 1. M+(amount, pressure) with no CVs

If one includes the corresponding value (CV) tuple (0,0) associated with the M+ constraint among the parameters in the input, none of the problematic functions of Figure 1 will be considered by the program. Only the set of increasing functions passing through the origin will be represented, which is all one can do with the present amount of knowledge.

2. INTERVAL CORRESPONDING VALUES

Even when such corresponding values are employed, one does not make full use of the qualitative setup to represent available information. We have identified the cause of this to be the insistence [2,3,5] that all magnitudes which appear in CV tuples should be landmarks. We claim that, by changing the definition of corresponding value tuples to include intervals as well as point values, and modifying the qualitative arithmetic routines that operate on these values, a better qualitative reasoner can be obtained. We will demonstrate this on examples from Kuipers' QSIM program [5].

Consider the case where we have two parameters X and Y with a functional relationship f: X → Y, where f is an increasing function, and the following is known about the function and the landmark values of the parameters::

X has two positive landmarks, x1 and x2, such that x1 < x2.

Y has two positive landmarks, y1 and y2, such that y1 < y2.

f(0) = 0

f(x2) = y2

f(x1) > y1

Note that all these are ordinal relationships and are easily representable using the qualitative scheme. But no reasoner using the "classical" corresponding value technique can represent the inequality above. Figure 2 shows three functions which are mapped to the classical representation of f. Two of these are inconsistent with our knowledge of f(x1), however, the CV tuple (x1, (y1,y2)), which embodies that information, is not allowed by the points-only representation, so the program cannot distinguish the three functions.

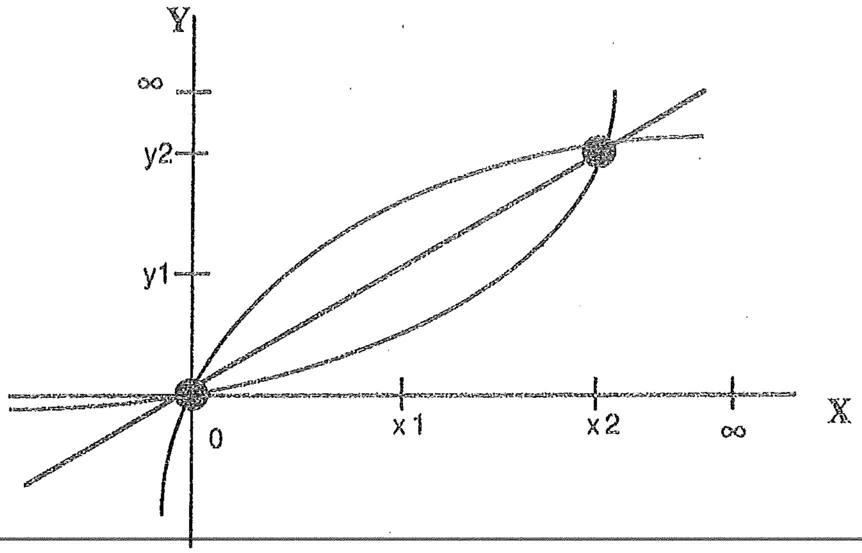


Figure 2. M+(X,Y) with CVs (0,0), (x2,y2)

If intervals were allowed in CV tuples, (x1,(y1,y2)) could also be associated with the M+ constraint, and we would have the (desirable) situation shown in Figure 3.

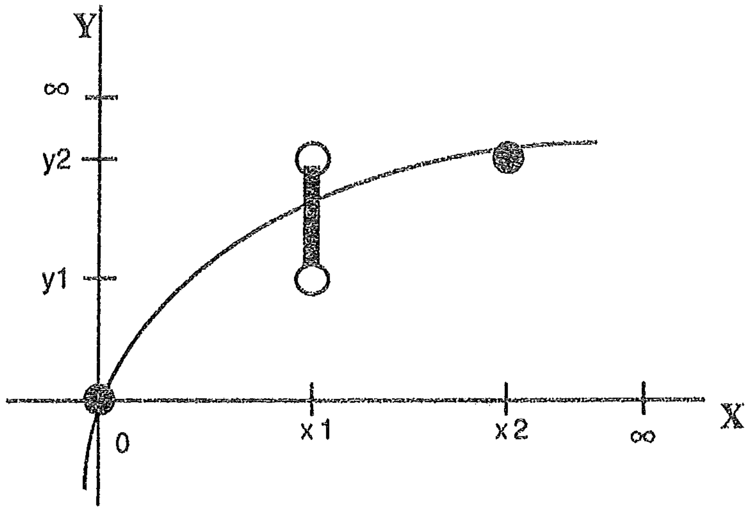


Figure 3. M+(X,Y) with CVs (0,0), (x2,y2) and (x1,(y1,y2))

3. ELIMINATING SPURIOUS BEHAVIORS

We have implemented the modifications entailed by allowing intervals in CV tuples on our version of the QSIM program [9]. An example system, for which "pure" QSIM predicts four spurious behaviors (all of which are detected and eliminated by our version) in addition to the 11 correct possibilities, will be presented now. Consider Figure 4. A little ball is thrown vertically upward from the ground. A small and powerful light source is fixed at a location of a certain height to the left of the point of takeoff of the ball. It is assumed that the ball can never reach the height of the light source. The height, velocity, and acceleration of the ball are parameters Y, V, and A, respectively. A is fixed at a negative landmark. One is also interested in the position of the ball's shadow on the ground, represented by parameter X. 0 (zero) is the point of takeoff of the ball in both X and Y's quantity spaces. The ground is level (i.e. has no "bumps") so that X is a continuous function. The highest altitude that the ball has ever reached before is the landmark alt_rec in Y's quantity space. There is a dead bug lying at a point to the right of the ball's takeoff point. X has the positive landmark bug_pt when the shadow is on the bug. Light travels infinitely fast (for the commonsense time scale at which the system is being viewed, of course.) The set of constraints is that of Table 1. (DERIV(X,Y) simply means that $\frac{dX}{dt} = Y$.) The sign of the time derivative of each parameter is also represented in the table. In this notation, **inc** means +, **std** means **0**, and **dec** means -.

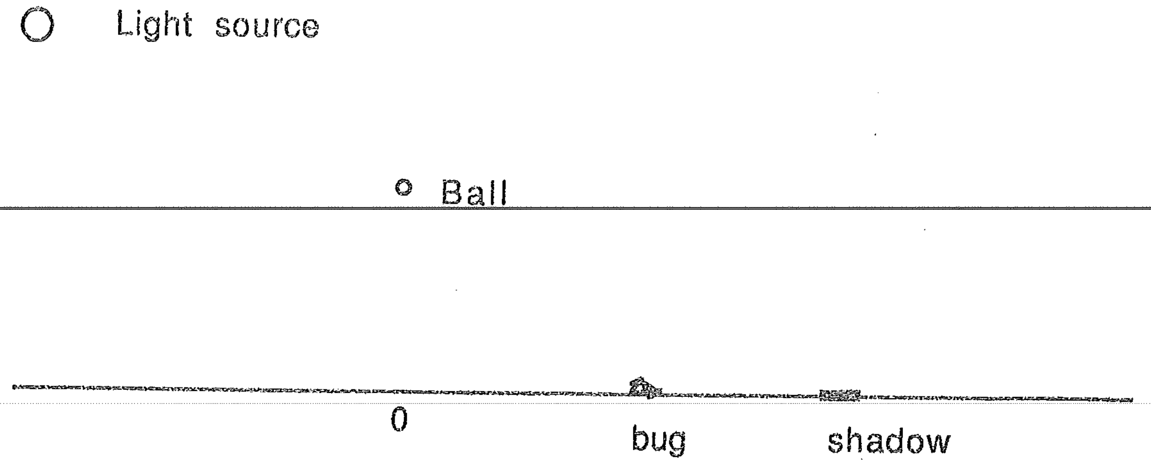


Figure 4. The ball / shadow system

Table 1
Constraints of ball / shadow system

Constraint	CVs
DERIV(V, A)	
DERIV(Y, V)	
M+(X,Y)	(0,0)

The ball is shot up with initial velocity v_0 at t_0. Table 2 contains part of one of the spurious behaviors that pure QSIM predicts. Let us examine this behavior. The ball is shot up at t_0. At t_1, it breaks the old altitude record and goes on climbing. At t_2, when the ball is at a point above alt_rec, its shadow falls on the bug. At t_3, both the ball and its shadow stop for an instant, and their magnitudes at that point are recorded as CVs. After that, the ball starts going down, crossing alt_rec at t_4. But the shadow has still not reached the bug for a second time. This is inconsistent with the available knowledge about the function from Y to X at t_1, so Table 2 contains a spurious behavior.

Our version of QSIM, which uses interval as well as point values in CV tuples, does not predict this spurious behavior. The algorithm is able to compare the X and Y values at time point t_1 with the proposed values at time point t_4. Since the CV tuple (alt_rec,(0,bug_pt)) is inconsistent with the proposed value tuple (alt_rec,(bug_pt,NewX)), (a function does not map a single point to two mutually exclusive intervals,) that proposal for t_4 will be eliminated. For M+ constraints, this check is implemented as follows: If the proposed magnitude tuple is (m_A, m_B), and a CV tuple (p, q) exists, the signs of $(m_A - p)$ and $(m_B - q)$ should be the same. In our example,

$$\text{alt\_rec} - \text{alt\_rec} = \mathbf{0} \neq (\text{bug\_pt},\text{NewX}) - (0,\text{bug\_pt}) = +,$$

so the proposed tuple fails the test.

Table 2
Spurious behavior of ball / shadow system

time	Y	V	A	X
t_0	<0, inc>	<v_0, dec>	<g, std>	<0, inc>
(t_0,t_1)	<(0, alt_rec), inc>	<(0, v_0), dec>	<g, std>	<(0,bug_pt),inc>
t_1	<alt_rec, inc>	<(0,v_0), dec>	<g, std>	<(0,bug_pt),inc>
(t_1,t_2)	<(alt_rec, ∞), inc>	<(0, v_0), dec>	<g, std>	<(0,bug_pt),inc>
t_2	<(alt_rec, ∞), inc>	<(0, v_0), dec>	<g, std>	<bug_pt, inc>
(t_2,t_3)	<(alt_rec, ∞), inc>	<(0, v_0), dec>	<g, std>	<(bug_pt,∞),inc>
t_3	<NewY, std>	<0, dec>	<g, std>	<NewX, std>
(t_3,t_4)	<(alt_rec,NewY),inc>	<(-∞, 0), dec>	<g, std>	<(bug_pt,NewX),dec>
t_4	<alt_rec, dec>	<(-∞, 0), dec>	<g, std>	<(bug_pt,NewX),dec>

Quantity space of X: {-∞, 0, bug_pt, NewX, ∞}

Quantity space of Y: {-∞, 0, alt_rec, NewY, ∞}

Kuipers and his colleagues have reported on various methods of extending QSIM so that it will predict fewer spurious behaviors. All of these approaches involve the addition of new types of constraints to the qualitative vocabulary, making use of the energy or phase properties of the system [6,8], or by making "unsafe" assumptions (which may lead to real possibilities being eliminated) about the shapes of the monotonic functions to automatically calculate values of the higher-order derivatives [7] in the system. Our modification handles a class of spurious behaviors different from the one targeted by these methods. It can be demonstrated [9] that our modified QSIM can eliminate spurious predictions which are not detected by the proposals of [6-8].

4. A STRONGER QUALITATIVE ARITHMETIC

Our implementation of the interval corresponding value feature also involved an improvement to the qualitative arithmetic routines employed during the CV checking phase. In order to check proposed parameter magnitudes for consistency with the previously recorded corresponding values, the QSIM algorithm uses qualitative subtraction and division. When intervals are allowed as corresponding values, one may be faced with a situation where one has to apply these operations to two instances of the same interval magnitude; i. e. one has to evaluate

$$(a,b) - (a,b)$$

or

$$\frac{(a,b)}{(a,b)} .$$

In the general case, these operations give ambiguous results. However, keeping in mind that both operands in the above operations are values of the same parameter, and that QSIM keeps all of the previous values of all parameters for cycle detection purposes, it is possible to determine unambiguous signs for these in some cases. Assume that the qualitative states through which parameter P has passed since the beginning of the simulation are as in Table 3. Suppose now that the arithmetic routine is trying to subtract P's value at t_1 from its proposed value at t_3 . Clearly, the result of this operation is unambiguously - (negative), since the parameter has been decreasing in all the time from t_1 to t_3 . The extension of this idea to similar situations in which the parameter is increasing, rather than decreasing, and to the division operation is straightforward.

Table 3
Proposed behavior of parameter P

time	P
t_0	<(a,b),dec>
(t_0,t_1)	<(a,b),dec>
t_1	<(a,b),dec>
(t_1,t_2)	<(a,b),dec>
t_2	<(a,b),dec>
(t_2,t_3)	<(a,b),dec>
t_3	<(a,b),dec>

5. CONCLUSION

We presented a method of better exploiting the qualitative representation to obtain tighter qualitative simulations. The ideas explained here have been incorporated into our implementation of QSIM. Case runs and technical details can be found in [9].

REFERENCES

1. de Kleer, J. and Brown, J.S., A qualitative physics based on confluences, *Artificial Intelligence* 24 (1984) 7-83.
2. Forbus, K.D., Qualitative process theory, *Artificial Intelligence* 24 (1984) 85-168.
3. Forbus, K.D., Qualitative physics: Past, present and future, in: D.S. Weld and J. de Kleer (Eds.), *Readings in Qualitative Reasoning About Physical Systems* (Morgan Kaufmann, Los Altos, CA, 1990) 11-39.
4. Hayes, P.J., The second naive physics manifesto, in: J.R. Hobbs and R.C. Moore (Eds.), *Formal Theories of the Commonsense World* (Ablex, Norwood, NJ, 1985) 1-36.
5. Kuipers, B., Qualitative simulation, *Artificial Intelligence* 29 (1986) 289-338.
6. Kuipers, B., Qualitative reasoning: Modeling and simulation with incomplete knowledge, *Automatica* 25 (1989) 571-585.
7. Kuipers, B. and Chiu, C., Taming intractable branching in qualitative simulation, in: *Proceedings IJCAI-87*, Milan, Italy (1987) 1079-1086.
8. Lee, W.W. and Kuipers, B.J., Non-intersection of trajectories in qualitative phase space: A global constraint for qualitative simulation, in: *Proceedings AAAI-88*, Saint Paul, Minn (1988) 286-291.
9. Say, A.C.C., Qualitative System Identification, Ph.D. Dissertation, Department of Computer Engineering, Boğaziçi University, İstanbul, Türkiye (1992).
10. Williams, B.C., Qualitative analysis of MOS circuits, *Artificial Intelligence* 24 (1984) 281-346.

Qualitative Reasoning Experiments with the MVL Theorem Proving System

M. Ülkü Şencan and Varol Akman

Department of Computer Engineering and Information Science, Bilkent University, 06533 Bilkent, Ankara.

Abstract

An experimental program, QREM, is implemented using the inference mechanism of the MVL Theorem Prover System of Ginsberg. QREM uses Forbus' *Qualitative Process Theory* for its description of physical situations and constructs the interpretations of measurements. In this paper, we mainly concentrate on the representation of process descriptions and basic deductions, and give some idea about what MVL can provide for writing qualitative physics programs.

1 Introduction

Commonsense reasoning is one of the most popular topics in Artificial Intelligence [7, 8]. Nowadays, a great deal of attention is being given to studies in qualitative physics which tries to formalize one's commonsense knowledge about the physical world [1, 11].

Our conventional physics cannot succinctly give the intuitive meaning behind the functioning of a physical system. However, qualitative physics provides this information by giving a commonsense description of the situation. This difference between the two physics mainly stems from the fact that in conventional physics we describe physical behavior in terms of quantitative values and numerical equations, whereas in qualitative physics we employ qualitative values and say, interval arithmetic.

Since in qualitative physics we only use qualitative information for reasoning, a need arises for representing the physical system in a more formal way. Fortunately, there are several formalisms for the representation of physical systems and especially Forbus' *Qualitative Process Theory* (QPT) serves as an important guide for many of the current qualitative physics programs [2, 3].

In this paper, we introduce QREM—Qualitative Reasoning Experiments with the MVL Theorem Proving System. This is an experimental qualitative physics program based on QPT. QREM serves as a simple, clear, and flexible representation language for descriptions of QPT. The reasoning tasks are accomplished using MVL's default logic, because we must be able to make inferences even in the case of incomplete information. In its current state, QREM can make inferences about simple dynamical systems consisting of a number of containers and fluid paths that allow the flow of liquid between specified containers. However, once we agree on the representation issues and code a domain model of the physical system into QREM, it must be a rather straightforward matter to carry out the basic reasoning tasks for other, more complicated systems as well.

2 Qualitative Physics

Qualitative physics deals with commonsense reasoning about the physical world [1]. The motivation primarily comes from the studies in engineering problem solving in which techniques for automating engineering practice are sought [12].

Conventional physics completely describes the behavior of a physical system using accurate quantitative values and numerical equations. However, most of the time, this description does not seem to be helpful for understanding the functioning of the system. In such cases, qualitative physics provides valuable insights into the system's functioning by giving a commonsense description and a causal explanation for the resultant behavior.

Qualitative physics, unlike the so called conventional physics, uses a symbolic and qualitative model of the physical world. The behavior of a physical system can be described using qualitative values for quantities of existing objects in the situation. This qualitative representation necessitates a quantity value to be chosen from a discrete quantity space rather than from a continuous one. The behavior of the physical system is effectively characterized by the derivatives of its quantities. Hence, a quantity may increase, decrease, or stay unchanged when its first derivative has a value of 1, −1, or 0, respectively.

In Figure 1, qualitative and conventional physics are compared. At first, both of the physics attempt to formalize the physical situation, one using complex numerical equations and the other using simple qualitative constraints. Then, both of them solve their related equations using their own methods. At the end, while qualitative physics comes up with a commonsense description of the solution, conventional physics comes up with a numerical value whose intuitive content may be null [9].

A computer program for qualitative reasoning requires a qualitative model of the physical system as input. This model must be adequate for specifying what constitutes the physical system of concern. Fortunately, there are different theories in AI that offer constraint-based, component-based, and process-based description models for physical systems [1]. The one that we apply in QREM is a process-based theory.

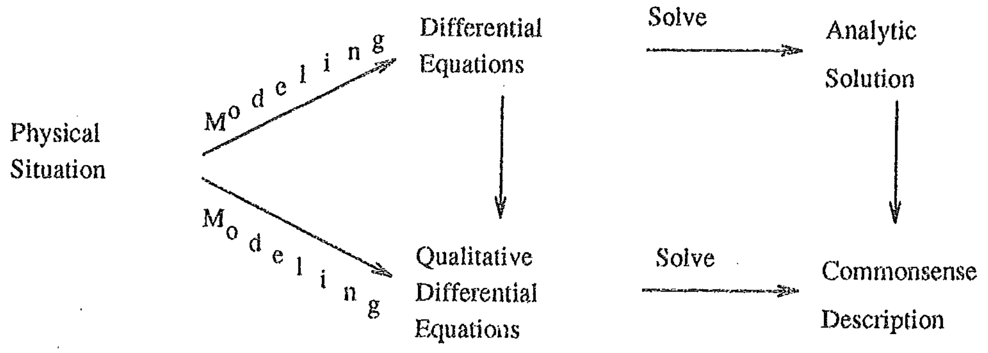


Figure 1: Comparison of qualitative and conventional physics [adapted from *Encyclopedia of AI*, Shapiro, S. C., Ed., vol. 2, p. 807].

3 Essence of QREM

3.1 Descriptions in QREM: QPT

QPT is a process-based theory for describing physical situations. Forbus characterizes his theory as one in which reasoning about dynamical systems can be made easily and effectively. He mentions this in [2]: "Qualitative Process Theory defines a simple notion of physical process that appears useful as a language in which to write dynamical theories."

According to QPT, a dynamical system changes its state as a result of active processes in the situation. In QPT, a *process* is described as something that causes changes in objects over time [2]. Motion, colliding, fluid flow, and boiling are examples of processes acting on objects. In Figure 2, a potentially existing process, namely fluid flow, is represented in the framework of a dynamical system consisting of two containers and a fluid path.

A domain model of a dynamical system consists of descriptions of existing objects in the system, relationships between those objects, and the processes that can occur in some physical situation. A specific situation occurs when all of its conditions hold. Particularly, active processes in each situation need not be given individually; they can be inferred using the process specifications in the domain model.

QPT describes a process using the following components:

- **Individuals** Objects that the process acts on.
- **Preconditions** Conditions that are imposed by the external world.
- **Quantity Conditions** Conditions that are necessary for the process to become active.
- **Relations** Relations between quantity values and process variables (i.e., what holds when the process is active)
- **Influences** Direct effects of a process. Each process has at least one direct influence.

Now, we have some idea about how to describe a physical system using processes, but how are we going to perform reasoning tasks using those processes? QPT's reasoning goes through the following basic deductions:

- **Finding Possible Processes** Processes whose preconditions hold are potential processes that can occur in some situation.
- **Determining Activity** A process instance is *Active*, if its preconditions and quantity

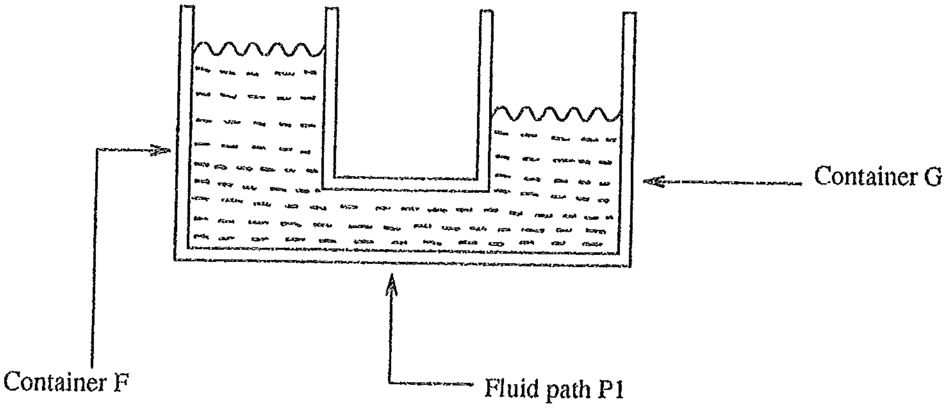


Figure 1: A physical system where a liquid flow process can be *Active*.

conditions hold. Otherwise, it is *Inactive*.

- **Determining Change** Direct and indirect influences of a process are resolved in order to find the changes in the system.
- **Limit Analysis** Changes in quantities may change some process instances. Limit analysis is used to determine those.

Using QPT's basic deductions we can perform *predictions*, *measurement interpretation*, and *causal reasoning* about dynamical systems.

3.2 Inference in QREM: MVL

The MVL theorem proving system written in Common Lisp is an implementation of theoretical work done at Stanford University [4]. The core of the system relies on the "multivalued logics" paradigm of Ginsberg [5].

MVL provides facilities for making satisfactory inferences in various logics—default logic, circumscription, probabilistic logic are some of these—and allows one to define new logics. The most important feature of MVL is its use of multiple truth values for logical statements. Unlike Prolog, MVL does not simply label a statement to be true; it also considers "true by default," "true by some assumption," "false," "false by default" as bona fide truth values and uses these when determining the answer for a query [6].

The MVL database consists of sentences that are represented as LISP *s*-expressions and labeled with truth values. The logical connectives NOT, OR, and AND are used for constructing MVL statements.

For inference tasks there are two types of connectives: "=>" and "<=". The connective "<=" is used for backward chaining and " =>" is used for forward chaining. The form of backward- and forward-chaining rules are as follows [6]:

(<= *Conclusion Premise*$_1$ *Premise*$_2$ ... *Premise*$_n$)
(=> *Premise*$_1$ *Premise*$_2$ ... *Premise*$_n$ *Conclusion*)

4 Implementation

4.1 Representational Aspects of QPT

Individuals and Quantities Individuals in a physical system are characterized by their existence. If an individual may exist in a situation it must have the quantity property. Hence, we represent any object that has a quantity as an individual, e.g.,

```
(<= (and (individual ?x) (?q ?x))
    (quantity-type ?q)
    (has-quantity ?p ?q)
    (?p ?x))
```

Here, (has-quantity ?p ?q) means ?q is a quantity for individual ?p whereas ?x in (?p ?x) is a particular instance of ?p. The expression (?q ?x) instantiates quantity ?q for ?x.

Some individuals in QPT can be defined by making the existing ones more specific. One example would be contained-stuff if we already have an individual called piece-of-stuff. Here, contained-stuff is a piece-of-stuff contained in a place and contained-liquid is a contained-stuff:

```
(<= (contained-stuff (?substance ?state ?place))
    (piece-of-stuff ?substance)
    (state ?substance ?state)
    (contains ?place ?substance ?state))
(<= (contained-liquid (?substance ?place))
    (contained-stuff (?substance LIQUID ?place)))
```

In QPT, relationships between quantities are basically indicated by qualitative proportionalities, correspondences, and inequalities. A qualitative proportionality $Q_1 \; \alpha_Q \; Q_2$ means "there exists a function that determines Q_1 and is monotonic in its dependence on Q_2." Correspondences are used for mapping value information from one quantity space to another via α_Q [2, 3]. Those are written in MVL with the same notational considerations as in QPT. Below ?x is a contained-liquid:

```
(qprop+ (pressure ?x) (amount-of ?x))
```

Here, qprop+ denotes that the pressure of ?x is qualitatively proportional ('+' denotes increasing) to its amount. Inequalities can either be given directly or inferred depending on whether numeric values of those quantities are known. For this purpose greater, less, equal-to, and some algebraic manipulations are defined in MVL.

Domain Models A domain model of QPT can be specified by defining quantities, individuals, and processes that exist in the domain. The types of quantities in the domain are defined as (quantity-type < *type* >), e.g.,(quantity-type pressure).

Until now, we gave only the basic components of a domain model that can easily be described by using predicates. With those predicates in the database, we can make some simple inferences [10], e.g., "What kind of individuals exist in the domain?" However, we need more complex inferences for QPT's basic deductions. For this purpose, we are going to represent processes and some other related concepts in the form of rules.

In QREM, a process description is given in three parts, i.e., we have three rules for each process. One rule defines a process along with its individuals. If individuals exist in the situation, the process is considered to be potentially *Active*. The other two rules are related to a process as being *Active* in the situation.

Status of a process can be inferred using the following rule which says that a process is *Active* when all the conditions imposed on it hold:

```
(<= (status (process ?process ?individuals) ACTIVE)
    (hold-conditions (process ?process ?individuals)))
```

A sample process description for liquid-flow captures all necessary conditions (preconditions and quantity conditions) and individual specifications for that process:

```
(<= (process liquid-flow (individuals ?source ?destination ?path))
    (contained-liquid ?source)
    (contained-liquid ?destination)
    (not (equal ?source ?destination))
    (fluid-path ?path)
    (fluid-connection ?path ?source ?destination))
```

Individuals for liquid-flow process are contained-liquids—one source, one destination—and a fluid-path between source and destination, e.g., there is a fluid-connection between source and destination.

```
(<= (hold-conditions
        (process liquid-flow (individuals ?source ?destination ?path)))
    (process liquid-flow (individuals ?source ?destination ?path))
    (aligned ?path)
    (greater (a (pressure ?source)) (a (pressure ?destination)))
    (status (view liquid-flow-support
                  (individuals ?source ?destination ?path)) ACTIVE))
```

When preconditions and quantity conditions hold, a process becomes *Active*. For the liquid-flow process, (aligned ?path) is a precondition indicating that the fluid path is isolated from any other external effect. If the pressure of the source is greater than the pressure of the destination and the geometric properties of the fluid path allow it, then there will be a flow of liquid from the source to the destination:

```
(<= (and (I+ (amount-of ?destination) (flow-rate ?path))
         (I- (amount-of ?source) (flow-rate ?path)))
    (status (process liquid-flow
                     (individuals ?source ?destination ?path)) ACTIVE))
```

In QPT, processes are the only source of direct influences [2, 3]. I+ and I- represent direct influences of flow-rate on the amount of source and destination ('+' positive influence, '-' negative influence) when the liquid-flow process is *Active*. If flow rate is increasing, then the " amount of destination" will also increase whereas the "amount of source" will decrease.

4.2 Basic Deductions

Finding Possible Processes Possible processes are simply characterized by their individuals. All processes that have their individuals exist in the situation are potentially *Active* and can be found them using a simple inference on the rules of process descriptions.

The following MVL query searches for all possible process instances in the domain:

```
;; Backward search for all processes
   (bcs '(process ?process ?individuals))
```

The binding list of this query is passed to a Lisp function which makes status assumptions about process instances. Inconsistent process instances (process instances that cannot be *Active* together in the same situation) are thrown away. This procedure is called *Elaboration*.

Determining Activity Process instances found by *Elaboration* can be *Active* if they satisfy their conditions. To find which process instances are *Active*, we invoke the following query that tries to prove whether a potential process is *Active*:

```
;; Backward search for all processes and
;; try to prove that a potential process is Active.
;; Take all processes that are proved to be Active.
   (bcs '(status (process ?process ?individuals) ACTIVE))
```

Determining Change In QPT, changes are imposed by *Active* processes; processes are the only source of direct influences [2, 3]. Quantities may change either because of some direct or indirect influences (expressed by qualitative proportionalities) on them. A quantity is said to be *directly influenced* if there exists at least one process directly influencing it at some particular time. On the other hand, a quantity is *indirectly influenced* if it is a function of some other quantity that is changing.

The derivative of a directly influenced quantity equals the sum of all of the direct influences on it. In QREM, an *influence adder* is used for finding this derivative just as described in QPT [2, 3].

4.3 Measurement Interpretation

The importance of measurement interpretation is emphasized in [12]: "The problem of interpreting observations of a system over time is fundamental to intelligent reasoning about the physical world. We view interpretation as the task of determining which possible behaviors predicted by the current model are consistent with the sensory data, including which are most plausible."

Measurement interpretation through time is more difficult than interpretation at a given time. Although QPT mentions the notion of time, there is no satisfactory temporal representation that can be easily embedded in an implementation. Hence, for experimentation we only use interpretation at a time instant. The algorithm used for measurement interpretation is given in Figure 3.

```
                    MEASUREMENT-INTERPRETATION

1. find process instances PROCESS-INS
2. make status assignments about PROCESS-INS
   2.1. find all combinations of process instances
   2.2. throw away inconsistent combinations
3. for each combination do
   3.1. resolve influences of quantities
   3.2. if measurement of a given quantity is
        equal to the one found in resolving influences, then
        this combination is a possible situation at that time,
        hence give it as a cause of the measurement
```

Figure 3: Algorithm for measurement interpretation.

5 Conclusion and Future Work

We introduced an experimental program called QREM for qualitative reasoning about dynamical systems. In general, representation of physical systems plays an important role in qualitative physics. A clear representation of physical system descriptions proves to be useful when writing domain models.

In QREM, we make use of a flexible representation tool, viz. the MVL theorem proving system that provides multivalued inference. When reasoning qualitatively, we may lack some information about the situation, and some assumptions need to be made. In these cases, the default logic of MVL allows us to make some default assumptions.

In QREM, measurement interpretation of an observed quantity value is implemented. In the future, other important reasoning tasks such as limit analysis and prediction may be implemented. Yet another project may be to concentrate on the ATMS part of MVL in order to make the inferences more efficient.

References

[1] Forbus, K. D., "Qualitative Physics: Past, Present, and Future," *Qualitative Reasoning About Physical Systems*, Eds. Weld, D. S., and de Kleer, J., pp. 11–39, Morgan Kaufmann, San Mateo, Calif., 1990.

[2] Forbus, K. D., "Qualitative Process Theory," *Artificial Intelligence*, vol. 24, pp. 85–168, 1984.

[3] Forbus, K. D., *Qualitative Process Theory*, Ph.D. Thesis, Electrical Eng. and Computer Science Dept., M.I.T., Cambridge, Mass., 1984.

[4] Ginsberg, M. L., "The MVL Theorem Proving System", *SIGART Bulletin*, vol. 2, no. 3, pp. 57–60, 1991.

[5] Ginsberg, M. L., *Multivalued Logics*, Technical Report, Computer Science Dept., Stanford University, Stanford, Calif., 1989.

[6] Ginsberg, M. L., *User's Guide to the MVL System*, Technical Report, Computer Science Dept., Stanford University, Stanford, Calif., 1989.

[7] Hobbs, J. R., and Moore R. C., Eds., *Formal Theories of the Commonsense World*, Ablex, Norwood, New Jersey, 1988.

[8] Lenat, D. B., and Guha, R. V., *Building Large Knowledge-Based Systems—Representation and Inference in the CYC Project*, Addison-Wesley, Reading, Mass., 1989.

[9] Shapiro, S. C., Ed., *Encyclopedia of Artificial Intelligence*, vol. 2, pp. 807–813, John Wiley, New York, 1987.

[10] Sharples, M., Hogg, D., Hutchison, C., Torrance, S., and Young, D., *Computers and Thought—A Practical Introduction to Artificial Intelligence*, The MIT Press, Cambridge, Mass., 1989.

[11] Weld, D. S., and de Kleer, J., "Qualitative Physics: A Personal View," *Qualitative Reasoning About Physical Systems*, Eds. Weld, D. S., and de Kleer, Morgan Kaufmann, San Mateo, Calif., 1990.

[12] Williams, B. C., and de Kleer, J., "Qualitative Reasoning About Physical Systems: A Return to Roots," *Artificial Intelligence*, vol. 51, no. 1–3, pp. 1–9, 1991.

TEKDÜZE OLMAYAN USAVURUM YÖNTEMLERİ VE YAPAY ZEKA

Mustafa M. DAĞLI
TÜBİTAK - EEEAG (YAGÜ)
Yazılım Araştırma ve Geliştirme Ünitesi
06531 ODTÜ/ANKARA

ABSTRACT: This paper tries to be an essay-type research attempt on the conceptual basis of nonmonotonic reasoning (similar to the one submitted [in English] at DECSYM'92 [1]). I have tried (in a manner) to determine the scope and limits of nonmonotonic reasoning (for AI purposes). My ambition this time is, to relate the topics in some way to ethics, in addition to metaphysics.

Kategori: Felsefi temeller

Anahtar Kelimeler: Yapay zeka, usavurum (tekdüze olmayan), mantık (felsefi), bilgi gösterimi

Genel Terim: Teori

ARTAKALAN

kuşlar:
uçarlar
ama penguen kuşları uçmazlar

insanlar
kuşlar gibidir

kuşlar
uçar
ya deve kuşları?...
(1.Ağu.1990, Gebze-İng.
Çev: 2 Ocak 1992, ANK.)

I. GİRİŞ

Geleneksel (standart) mantıkların, bilginin eksik bulunduğu durumlarda çözüm gösterme konusunda yetersiz oldukları, günümüzde genellikle kabul edilmektedir. Yaşamın gerçeği olduklarını anladığımız zaman, birtakım bilgileri, "yeni bilgi" olarak bilgi dağarcığımıza sürekli olarak eklediğimizden, bu bilgisel konum, günlük yaşantımızdaki çoğu duruma da karşılık gelir görünmektedir. Çıkarım yapmak için kullanıldıklarında, adıgeçen gündelik yaşam durumlarında içerilen mantığı dile getirmek ve açıklamak için tekdüze olmayan kavramlara ihtiyaç duyarız. Ben, bildirimde, tekdüze olmamayı açmaya, ve tekdüze olmayan usavurum yöntemlerinin yapay zeka (YZ) için önemini vurgulamaya çalışacağım.

I.1 AÇIKLAYICI BİR ÖRNEK

H1: Kuşlar uçar

şeklinde bir varsayım düşünelim. Sorulduğunda, sağduyumuz, böyle bir bildirime 'yanlış' demektense 'doğru' demeyi yeğler. Hernasılsa,

H2: Penguenler uçmaz

biçimindeki bir varsayım da, tarafımızdan, "yanlış" değil, doğru olarak algılanacaktır. Üstelik,

H3: Penguenler 'kuş'turlar

biçimindeki bir savı da "doğru" sayarız.

Eğer penguenler (ve devekuşları, ve örneğin Malta doğanı) kuşsalar -ki öyledirler-, ve uçmazlarsa -ki uçmazlar da?-, neden "Kuşlar uçar" deriz?

Bu özgün 'kuşlar-dünyası' örneği, günümüzde, tekdüze olmamanın çok bilinir bir örneğini oluşturmuş durumdadır [8, 12, 7, 9, 10, 5, 4, 11, 2].

I.2 BİR DİĞER ÖRNEK

1, 2, 3, ...

şeklinde bir sayı dizimizin olduğunu, ve doğru yanıtı bulmamızın istendiği test biçimindeki bir sınavda, aşağıdaki şıkların bulunduğunu düşünelim:

a) 4 b) 5 c) 6 d) hepsi e) hiçbiri

Neden çoğumuz bu soruyu "4" diye yanıtlar?

Soru, devam şeklini 'asal' sayılar kümesinde arama biçiminde ele alınabilirse, yanıt pekala '5' olabilir.

Ya da, dizinin, önce iki kat sonra 1.5 kat alma biçiminde iki ardışık hareketten oluştuğunu düşünebilirsek, doğru yanıt 4 te 5 te değil, 6 olacaktır.

Kendimizi { 0, 1, 2, 3 } tabanlı bir aritmetik ortamında varsaydığımızda, yanıt açıktır ki "hiçbiri" olacaktır.

Ve, 'a', 'b', 'c', ve 'e' şıklarının tümünün, değişik ortamlarda geçerli açıklamaları bulunduğu için, pekala yanıt "hepsi" olabilir.

Hangisini seçmeliyiz?

Kendimizi, soruyu - 10 tabanlı aritmetik ortamında ve- 'doğal' sayılar kümesinde çözmek üzere, nasıl sınırlandırabiliriz?

Tanım kümesini "doğal" sayılar olarak algılayabiliyorsak, soruda açıkça belirtilmediği halde, niçin 'asal' sayılar kümesi olarak algılayamıyoruz?

Araştırmanın hangi çerçevede yürütüleceğine dair, sorunun ifade edilişinde herhangi bir belirleme var mıdır?

Bunu nasıl anlarız?

II. TEKDÜZE OLMAMA NEDİR?

Bir sisteme yeni aksiyomlar eklediğimizde eski teoremlerini geçersizleştirebiliyorsak "tekdüze değildir" deriz; ki bu durum, gündelik yaşamdan gelişigüzel tablolar seçtiğimizde, elde edeceğimiz çoğu manzaranın gerçek biçimini oluşturur.

0, 2, 4, ..., 996, 998, 1000, ...

biçiminde bir sayı dizisini irdelemek istediğimizi varsayalım. Bu, '+2' serisidir deriz. Böyle düşündüğümüz için, nasıl devam edilmesi gerektiği soruldugunda, oldukça emin bir biçimde, '1002' deriz. Herşey onun lehinedir, mantıksal olarak başka bir seçenek görünmemektedir.

Ne var ki, oldukça benzer -fakat hafifçe farklı- bir durumda, muhtemelen fikirlerimizi değiştirmek gereği duyacağız. İlkokul çocuklarının, aritmetik dizileri yeni öğrendikten sonra bir sınavda '+2' serisine -bir daktilo hatası ile- devam etmelerinin istendiğini düşünelim:

S. Aşağıdaki sayı dizisine en uygun devam şeklini bulunuz:

994, 996, 998, 1000, 1004, ?

Sorunun, aslında, '1004' yerine '1002' biçiminde hazırlanmış bulunduğunu varsayalım. Fakat yeni başlamış ilkokul çocuklarının bunu bilmeleri doğal olarak beklenemez; ve, öğretmenin de (sekreterinden kaynaklanan) bu hatayı sınav bitmeden önce farketmemiş olduğunu varsayalım. İyi hazırlanmış çoğu öğrencinin '1006', bir cingöz öğrencinin ise '1008' işaretlediğini düşünelim. Doğru yanıt 1006 mı 1008 mi olacaktır? Sınavın iptali, hiç hazırlanmamış öğrencilerin ödüllendirilmesi sonucunu yaratacağı gibi, (zaman kaybı, güven azalması, ciddiyetsizlik, vb.) başka sorunların ortaya çıkmasına da yol açabilecektir. Öğretmenin örnek davranışı nasıl olmalıdır?

Wittgenstein'ın bu ve benzeri durumlar için tepkisi

> ... Her aşamada, bir sezgi değil, bir yargı gerekir demek nerdeyse daha doğru olacaktır [13, parag. 186].

biçimindedir.

Neyseki, gündelik yaşamlarımızı gözönüne aldığımızda, her zaman en sivri örneklerle karşılaşmayız. Sekreterler seyrek olarak '1002' yerine '1004' yazarlar; ve öğretmenler daktilo hatalarını görüp düzeltir. İyi-hazırlanmış öğrenciler '994, 996, 998, 1000, 1002, ?' dizisinin devamı olarak '1004'ü işaretler, ve günlük yaşamda, çoğu zaman herşey yolunda gider. Ve, '994, 996, 998, 1000' serisinin devamı olarak '1002' yerine '1004'ü işaretleyen öğrenciler, öğretmenlerinden geçer not alamazlar.

Eğer dizi, '994, 996, 998, 1000, 1004, ?' biçiminde verilmiş olsaydı, tarafımızdan, "1000'e kadar '+2' daha sonra '+4'" biçiminde algılanabilirdi. Fakat, '994, 996, 998, 1000, ?' benzeri bir durum, basitçe, "'+2' serisi" şeklinde anlaşılmalıdır. Devam'ın '1004' olduğu öğrenilinceye kadar, kabul edilecektir ki, '994, 996, 998, 1000' dizisinin, sade bir biçimde, "'+2' serisi"nin devamı olarak, '1002, 1004, ...' şeklinde süreceğini bekleriz.

Bir müddet için, gözlediğimiz sayıların, oyun düzenleyicinin bilgisayarında gizlenmekte olan önceden belirlenmiş bir fonksiyon tarafından türetildiklerini; ve bu dizinin elemanlarının, etkileşimli olarak (bir tahmin yapıp o tahmin'i bilgisayara bildirdikten sonra) birer birer ekranda göründüklerini varsayalım.

Eğer '994, 996, 998, ?' biçiminde bir dizi verilmiş ve bir tahminde bulunmamız istenmişse, normal olarak '1000' deriz. '994, 996, 998, 1000, ?' dizisini gördüğümüzde, bir tahminde bulunmamız istenince '1002' deriz.

Devam'ın '1004' olduğunu görünce ne olur? Basitçe, '1002' olmadığını, '1004' olduğunu anlarız. Ve düşünürüz ki, gördüğümüz dizi "'+2' serisi" değildir; "1000'e kadar '+2' daha sonra '+4'" olabilir. Akabinde, devam biçiminin gerçekten belirlediğimiz gibi olup olmadığını araştırmaya başlarız.

* * *

Devam'ın '1004' olduğunu ve fakat '1002' olmadığını gördüğümüz zaman anladık ki sayı dizisi "'+2' serisi" değildir. Sayı dizisinin kuralının, "'+2' serisi"nin kuralı olduğunu varsaydık.

Gerçeği gördüğümüz zaman, fikrimizi değiştirdik.

III. NEDEN 'TEKDÜZE OLMAMA'?

Bilgilerimizin kaynağı, temel olarak, dışımızdaki dünyadır. Gerek duyduğumuz verileri, gözlemlerimizden edinmeye çalışırız. Beş duyumuzun değişik kullanımları, "bilgi" dediğimiz şeyin ana kaynağını oluştururlar. Bu bilgi yığını başlıca görerek, duyarak, okuyarak, tadarak, dokunarak, koklayarak toplanır. Lütfen aşağıdaki cümlelere kulak veriniz:

> Çok inanmam ben kulaklarıma, çünkü çok yalan dinlediler. Ben gözlerime bile çok inanmam, çünkü aynı şeyin hem düzünü hem tersini gözlediler. Bir önermeye (sava) inanacağım zaman (özellikle onun için)

araştırmada bulunmaya başlar ve beklerim; yeterli ('tam olarak' yeterli'nin de ne demek olduğunu bilmem) gözlemlerden sonra, kendi kendime, "Muhtemelen (belki de?) bu böyledir (ya da, böyle görünüyor)" derim...

Bazı öğeleri verilen bu profil, kime, bir bilim dünyası insanına mı, yoksa tümevarıma inanmayıp herşeyden de kuşkulanmayı savunan 'şüpheci' bir kişiye mi ait? Kanımca, bu, yalnızca "sağduyu" dediğimiz şeyi, bir cins içerme girişimidir.

Bir an için düşünelim: "Aksiyom" nedir? Tartışılmaksızın kabul edilecek 'gerçekler' nelerdir? Nelerin aksinin de olabileceğini "kesin olarak" düşünmemek isteriz? Peki, niçin böyle yaparız? Kayıtsız şartsız kabul edilip, sonradan tartışma konusu yapılmış olan sorunlar, bilim tarihi'nde çok mu az sayıdadır? Bilim tarihi bile, (kendisi) bir yerden sonra, tartışma konusu değil midir?

Öğrendiğimiz yeni koşullar ışığında eski bildiklerimizden (hiç olmazsa bazılarından) vazgeçmek hakkına sahip miyiz? Bunu hangi akla, hangi usavurum yöntemlerine uyarak yaparız?...

Frege, aksiyomatik sistemler konusunda önemli çalışmalar yapmış bir insandır. Kendi orijinal aksiyomatik sistemi, Russell'ın, "kendisinin elemanı olmayan küme"nin incelenmesi önerisinden sonra değişikliklere uğramış mıdır? Küme kuramcıları bunu, Frege'nin yönelimini korumak, ve onarmak için yapmamışlar mıdır? Peki, bu onarım sırasında bazı eski 'kabul'lerden vazgeçilmiş midir? Bu davranış, bir yöntem genellemesi içinde değerlendirilebilir mi?

Değişme'nin kavranışı, bir "değişmez"lik düzleminde oluşmaktadır. "Değişen dünya"nın kavranışında da çoğu kez -hemen hemen hep-, içsel bir değişmezlik varsayımı vardır. Newton'ın F = ma formülünü dile getirişi, dünyanın hep, belli ve kesin bir çekim ivmesine sahip oluşu gözleminden hareketle varılmış bir genelleme değil midir? Ama düşününüz, kuvvet ile kütle arasındaki bu belirli bağıntı, -sonra geliştirilmiş olan- enerji kavramının da etkisi ile nasıl bir değişikliğe uğramıştır?

İki nokta arasındaki en kısa mesafe, her zaman bir doğru parçası mıdır? Belirli bir biçimde düşünmeye yönlendirildikten sonra, zihinlerimiz düz bir çizgi arıyor. Oysaki Euclides gibi düşünmeye zorunlu olmasak, belki farklı da yanıtlayabilirdik bu soruyu.

İçi dolu bir kürenin üstünde -sözgelişi- en kısa mesafe, iki nokta arasında **bir doğru parçası** değildir. Bir küpün iki yüzeyindeki noktalar arasında ise en kısa mesafe, bir kırık çizgi olacaktı.

İki nokta arasındaki en kısa yol, üstelik, "bir tane" olmak zorunda da değildir. Dünyamızı küresel olarak düşünseydik, iki kutup arasındaki en kısa uzaklık (yaya uzaklığı), bir yarım daire olacaktı. Ama bu yarım daire tek değildi, sonsuz sayıdaydı.

Dünyamız, bir mükemmel küre de değildir. İki kutup arasını en kısa yoldan aşmak isteyen bir gezgin, bir "yarım daire"yi izlemez. Onun kararlarını değiştirecek engeller, dağlar ve denizler, ve işi yokuşa sürmeden aşılabilecek daha kestirme yollar vardır. Biz, bir gezginin izleyeceği en kestirme yolu ölçü alsaydık, bir "yarım daire" de bulmayacaktık. Zeka açısından incelediğimizde, büyük çoğunluğumuz da kabul eder ki, izlenen bu yol, en "kestirme" yoldur. Çünkü dünyamız, -televizyon ekranından göründüğü gibi- "yuvarlak" bile değildir...

Gündelik konuşma dilimizden bir örnek irdeleyelim: "Paris'in başkenti Londra'dır." önermesi, sağduyumuz açısından -açıktır ki- doğru değildir. "Paris'in başkenti Londra değildir." demek te doğru değildir; zira dinleyen kişiye 'Paris'in bir başkenti olduğu' sezdirilmektedir, "Paris'in bir başkenti olduğu" doğru değildir. Dikkat edersek, burada iki değerli mantıkta olduğundan daha çok sayıda değer ölçüsü geçerli olmuştur; çünkü 'doğru değil', "yanlış"tan daha geniş anlamlı olarak kullanılmaktadır.

Gezginimize yukarıda sözettiğimiz **en kısa** yol üstünde bir tuzak kurulmuş olduğunu varsayalım. Olayların gelişiminden, -ve çeşitli izlerden- gezginimiz kendisine bir tuzak kurulmuş olduğunu anladığında, güzergahını değiştirir, ve kendine daha emin bir yol seçer. Bu yol *bir açıdan* bakıldığında daha uzun bir yoldur ama, bir diğer açıdan bakıldığında "amaca giden en kısa yol" kapsamında da -pekala- görülebilir. Görecelilik yaşamımızın bir parçasıdır da belki, zira tuzak kurmayı bildiğimiz gibi, -belki- faka basmamayı da bilebiliriz.

Hepimizin elbirliği işlerin güçleşmesine neden olur kimizaman; o güçlüklerin zahmetini de hepimiz birlikte çekeriz.

Klasik mantığın, modus ponens gibi en temel, en yerleşik sayılan bir kuralı bile, tanıtlamaya çalıştığımız usavurum yöntemlerince kimi zaman tanınmayabilmektedir. Başlangıçta "Kuşlar uçar" doğru mudur diye sormuş, ve yanlış demektense doğru demeyi yeğlemiştik. Ozzie isimli devekuşu sorgulandığında ise Ozzie'nin uçmayacağını söyleriz. Devekuşu kuş mudur dendiğinde ise "Evet, devekuşu da bir kuş cinsidir" deriz. Oysaki 'modus ponens', kuşlar uçar'sa ve devekuşu kuşsa, devekuşunun da uçacağını söyler!

Kimizaman, gündelik yaşamlarımızda, *akıl yürütürken bile,* kimi "mantık" kurallarının dışına da çıkmakta mıyız? Bunu, hangi akla uyarak yapıyoruz? Uçmak, "kuş olma" nın ana fikri ile birlikte mi ele alınmalıdır? Neden böyle bir varsayıma inandık? Zira, kuşların kanatları vardı; ve tünerken kullanılmasa da, kanadı kırık kuşlar uçamasa da, kuşların kanatları yavrularını korumak için kullanılabilse de, 'kanat-uçmak-için'di.

'Kanat-uçmak-için'se kümes hayvanları niye uçmuyordu? Üstelik uçmak, kuş olmanın ana fikri ile birlikte ele alınacaksa, uçmayan kuşlar nasıl (İstanbul'un arasokaklarına sığınan "Çarlık devri görmüş bir Rus düşesi" gibi mi) algılanmalıdır? Açıkta kalan sorular, belki de kapsanmak istenen sorunların genişliğiyle de bağıntılıdır; ama gündelik hayattan seçeceğimiz gelişigüzel örneklerde kullandığımız akıl yürütme biçimleri, onları formüllere dökmek için kullanmamız gereken usavurum yöntemlerini de -belki- belirlemektedirler.

IV. TEKDÜZE OLMAYAN USAVURUM YÖNTEMLERİNİN YZ AMAÇLARI İÇİN YÜKLENMELERİ İSTENEN ROL

Üç ana zeka türü ayırtetmek isterim: insan, hayvan, ve makina. Örneğin bir robot için, ya da genel amaçlı (ya da herhangi bir diğer amaçlı) "zeki" bir bilgisayar programı yapmak istiyorsak, gerçek dünya hakkında bazı bilgileri bu programda gösterimlememiz gerekir. Böyle bir gösterimin, usavurumun neliği ve gerçek yaşam alanlarında nasıl yürütülebileceği konusunda, bazı bilgileri içermesini gerektireceğini de söyleyebiliriz.

Yapay zekanın etkinlik ve yaygınlık kazanması, yeni ve değişik bir teknoloji ahlakı (ve hatta genel ahlak) gerektirebilecektir. Kanımca bu yeni ahlak, eski ahlak anlayışını (anlayışlarını) yalnızca yetersiz değil, kimi zaman hatta anlamsız kılmaktadır. YZ için konunun önemi, ahlakın, yapmak ve yapmamak üzerine yorumlarda bulunabilmesi nedeniyledir.

Yapay Zeka ve Bilgisayar Bilimleri, başlangıçlarından beri satranç oyunu ile çok yakından ilgilendirilmişlerdir. Ekim 1990'da *Scientific American*'da "Satranç bilgisayarları yeni başlayanları, uzmanlaşmış oyuncuları, ve şimdi büyükustaları yendi." diye bir saptama yapıldıktan sonra soruluyor: "[Bu makinaların] birisinin, dünya şampiyonunu yenmesi yalnızca zamana mı bağlı [3]?"

Ben sanıyorum ki, bu yarın değilse öbürkü günün (ya da 500 yıl sonranın) işidir. Satranç sınırlı bir oyundur, ve -veri sıkıştırma teknikleri sayılmazsa- tek, ve sonlu bir ağaç yapısına sahiptir. Bir oyuncu, her hamlesinde sınırlı sayıda değişken seçenek arasında tercih kullanır; ve yalnızca "50 Hamle Kuralı" bile, oyunun sınırlı karakterini belirlemeğe yeter. "Pozisyon Tekrarı Kuralı", ve diğer kurallar da, ek sınırlamalar getirirler.

Bir *en küçük* satranç kitabı vardır ki sayfaları sınırlı sayıdadır; ki tüm mümkün varyantları (farklı dalları olarak) içerir, ve her seçenek bu kitabın indeksinde belli bir sayfada gösterilmiştir, ki o belli sayfada seçilebilecek o dalın oynanabilecek kaydı mevcuttur.

Turnuvalarda oynanmış olan oyunlar, gerçekte bu "en küçük" satranç ağacının dalları ya da yandallarıdırlar; ve uygun bir kodlama tekniği ile, en küçük satranç ağacını oluşturmak üzere kayıt ta edilebilirler. Ya da, tamamen veya kısım kısım, teknik bir "Satranç Verileri Üretme Projesi"nde de oluşturulabilirler.

Öte yandan, satrancın sınırları sanıldığınca dar da değildir; ve satranç oyunu, günümüz koşullarında, sanki sonu açıkmış gibi bile düşünülebilir (bkz. [6, s. 309]).

Satranç, belki ilk çıktığı zaman, oynayanlarının (insanların) akıllarının, düşünsel ve duygusal yönlerden eğitilmesini sağlayacak biçimde çalıştırılmasını amaçlayan bir oyundu. Önceki biçemlerinden birisi, "filozofların oyunu" (ya da felsefe oyunu) diye de isimlendirilmişti (şimdiki masanın iki katı büyüklüğünde (2x[8x8]), 64 yerine 128 kareli).

Makinaları satranç oynamaya yönlendirirken, amaçlarımızı, yeni oyuncuları (satranç bilgisayarları vasıtasıyla) yenmekten ziyade, onları eğitmek biçiminde de tanımlayabiliriz (ve belirleyebiliriz). Ya da, -aynı teknik kapasiteyi kullanarak- gerçek yaşam hakkında ihtiyaç duydukları şeyleri, daha iyi ve daha yoğun olarak öğrenmelerine yardım etmeği amaçlayabiliriz.

Sorun'un bu bildiride fazla irdelemiş olmadığımız (ve incelemeyeceğimiz), estetikle (ve dolayısıyla sanatla) ilişkili bir yönü daha vardır. Kimi antik çağ düşünürüne göre ise, etik ile estetik, koşut bile sayılabilir...

"Kuşlar uçar" dediğimiz zaman, "Bazı kuşlar uçar" ya da "Çoğu kuşlar uçar" da diyebilirdik (ve hala da diyebiliriz); fakat, sunulmuş olan problemin önemi, ilkesel olarak, "uçma"nın, kuş olmanın anafikri ile birlikte düşünülmesinden dolayıdır. Yakup bir robot-kontrol-programı'nın adı ise, örneğin ay yüzeyinde yürüdüğü zaman (A'dan B'ye *doğruca* gitmesi istendiğinde sözgelişi) ay üstündeki küçük kraterlere ve çukurlara düşmemeyi bilmelidir.

Eğer Yakup hakkında bütün bildiğimiz onun bir "kuş" olduğu ise, "Yakup uçar" deriz. Yakup, gerçekte bir papağandır, ve çıkarımımız, kanatları kırık filan değilse, normal olarak doğrudur. Böyle bir usavurum, onun bir serçe ya da bir kanarya olması durumunda da, -normal olarak- geçerlidir. Fakat, Yakup'un bir devekuşu olduğunu öğrenseydik hemen fikir değiştirir, ve "Yakup uçmaz" derdik.

Birşeye çok sevindiğimiz zaman "kanatlanıp, uçarız"; uçmak, içimizde yüzyılların özlemi gibidir...

V. SONUÇ

Yapay zeka, bir yönü ile insanoğullarının, düşüncelerini araç ve gereç yardımı ile gerçekleştirme (araç yapma) serüveninin bir parçası; değerli ve güncel bir kısmı olarak algılanabilir. Psikoloji fonksiyonlarımızı incitmeksizin ele alınmak istendiğinde, yapay zeka, belirli bir disiplin olarak, "felsefe mühendisliği" ya da "uygulamalı felsefe" diye bile isimlendirilebilir. Konu gerçekten de önemlidir: Sözgelişi, CFC (kloro-floro-karbon) gazını, günün birinde, gecenin geç bir saatinde, bir tek kişinin -kocaman bir laboratuvarda- "keşfettiğini" düşünelim. Bu kişi, üzerine sanayiler kurulduktan sonra aynı gazın, "doğal dengeyi bozduğu ve ozon tabakasını yok edebileceği" nedeni ile kullanımının sınırlandırılıp, ve hatta yasaklanması gerekeceğini düşünebilir miydi -üstelik aynı anda-?

Maddenin atomlarını ve onun parçacıklarını merak eden 20. yüzyıl ilk yarı fizikçileri, atom bombası yapmayı -ve bundan iki tane Japon adalarına atmayı- mı amaçlıyorlardı?

Bazı basit kararlar, temelden etkiliyor yaşamımızı; ve kimi sonuçlarını onların, "yaşandıktan sonra" algılayabiliyoruz ancak. Yapay zeka, iki ağzı keskin kılıç gibidir; insanlığın diğer gezegenlere ve yıldızlara hayatı taşımasına yardım edebileceği gibi, belki bir "yapay zekasızlık"ın da yardımı ile dünyamızdaki yaşam sona da erebilecektir...

TEŞEKKÜR

Bu konudaki eğitimime katkıda bulunmuş oldukları için hocalarım Teo Grünberg, Ahmet İnam ve Akın Ergüden'e; ve güzelyazım yardımları dolayısıyla arkadaşım Ufuk Özlü'ye teşekkürlerimi bildirmek isterim. Beni bu çalışmamda desteklediği için kurumum TÜBİTAK'a şükranlarımı belirtirim.

KAYNAKLAR

[1] Dağlı, M., Nonmonotonic Reasoning and Artificial Intelligence, *Proceedings of DECSYM'92* (DECUS At-Large, Side, Antalya, Mart 1992) 25-38.

[2] Geffner, H. ve Pearl, J., Conditional Entailment: Bridging Two Approaches to Default Reasoning, *Artificial Intelligence* 53/2-3 (Şubat 1992) 209-244.

[3] Hsu, F., Anantharaman, T., Campbell, M. ve Nowatzyk, A., A Grandmaster Chess Machine, *Scientific American* 263/4 (Ekim 1990) 18-24.

[4] Lukaszewicz, W., *Non-monotonic Reasoning: Formalization of Commonsense Reasoning* (Ellis Horwood, Chichester, 1990).

[5] McCarthy, J., Applications of Circumscription to Formalizing Commonsense Knowledge, *Artificial Intelligence* **28** (1986) 89-116.

[6] McCarthy, J., Mathematical Logic in Artificial Intelligence, *Daedalus* **117**/1 (1988) 297-311.

[7] McDermott, D.V., Non-monotonic Logic II: Non-monotonic Modal Theories, *Journal of ACM* **29** (1982) 33-57.

[8] Minsky, M., *The Society of Mind* (Simon and Schuster, New York, 1986).

[9] Moore, R.C., Semantical Considerations on Non-monotonic Logic, *Artificial Intelligence* **25** (1985) 75-94.

[10] Perlis, D., On the Consistency of Commonsense Reasoning, *Comput. Intell.* **2** (1986) 180-190. Ayrıca: Ginsberg, M. (Ed.), *Readings in Nonmonotonic Reasoning* (Morgan Kaufmann, Los Altos, CA, 1987) 56-66.

[11] Poole, D., The effect of Knowledge on Belief: Conditioning, Specificity and the Lottery Paradox in Default Reasoning, *Artificial Intelligence* **49** (1991) 281-307.

[12] Reiter, R., A Logic for Default Reasoning, *Artificial Intelligence* **13** (1980) 81-132.

[13] Wittgenstein, L., *Philosophical Investigations* (Basil Blackwell, Oxford, 1953).

Issues in Commonsense Set Theory

Müjdat Pakkan and Varol Akman

Department of Computer Engineering and Information Science Bilkent University, 06533
Bilkent, Ankara

Abstract

This paper discusses commonsense set theory. First a brief review of classical set theory is given. Then the need for a commonsense set theory is questioned and the properties of a possible theory are examined. Finally, previous work in the area is presented.

1. BRIEF REVIEW OF EXISTING SET THEORIES

Set theory is a branch of modern mathematics with a unique place because other branches can be formally defined within it [1, 2]. The theory had started with the work of Cantor on infinite series. In his early conception, he thought of a set as a collection into a whole of definite, distinct objects of our perception or thought [3]. It did not take a very long time for this theory to be revised. Several paradoxes, including Russell's famous paradox of "the set of all objects which have the property of not being members of themselves," were introduced, leading to new axiomatizations of the theory [4].

Among the various axiomatizations, are Russell and Whitehead's *Theory of Types* [5] which brings in a hierarchy of types to forbid circularity and hence avoid paradoxes, and von Neumann, Bernays, and Gödel's NBG which introduces the notion of a class for the same purpose [6]. Strengthening NBG, a new theory, called MK (Morse-Kelley) was obtained [7, 8]. This theory is suitable for mathematicians who are not interested in the subtleties of axiomatic set theory. In 1937, Quine proposed his *New Foundations*, NF, to overcome some unpleasant aspects of the Theory of Types while keeping the main idea same [9]. He introduced the notion of *stratification* for this purpose. NF is a nice theory, avoiding Russell's Paradox, and allowing all mathematics to be defined within it, but has some strange properties. For example, Cantor's Theorem, $a < 2^a$, cannot be proven in NF, because a set used in the proof is not defined since it is not stratified. In general, the most popular axiomatization is ZF, originated by Zermelo [10] and later modified by Fraenkel [11]. The intention was to build up mathematics by starting with the empty set and then construct further sets cumulatively by various operators. This hierarchy works as follows [12].

Initially, the Axiom of Extensionality says that a set is completely determined by its members. The *Null Set Axiom* states the existence of the empty set $\emptyset$. The *Pair Set Axiom* then constructs the sets $\emptyset$, $\{\emptyset\}$ and $\{\emptyset, \{\emptyset\}\}$, and other one and two element sets. It is the *Sum Set Axiom* which constructs sets with any finite number of elements by

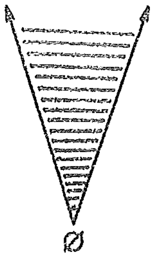


Figure 1: ZF universe extending in a cumulative hierarchy

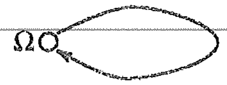


Figure 2: AFA representation of the circular set $\Omega = \{\Omega\}$

defining union of existing sets. Finally, *the Axiom of Infinity* states the existence of at least one infinite set from which others can be formed, viz. $\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \ldots\}$. This hierarchy can be depicted as in Figure 1. There is also the important *Axiom of Foundation* which basically prohibits sets which are members of themselves, thus avoiding the classical paradoxes.

In spite of its popularity in current research, ZF has its drawbacks. It is weak to decide some questions like the *Continuum Hypothesis.* It is also weak in some applications (set-theoretic, linguistic) which make use of self-reference, since circular sets like $x = \{x\}$ are prohibited by the Axiom of Foundation. ZF is also strong in some ways, viz. it violates the *parsimony principle* which states that simple facts should have simple proofs [13]. (This can be seen in the use of the Power Set Axiom in the proof of a simple notion like $a \times b$.) It can also be claimed that it is bad for mathematical practice that all the mathematical objects are to be realized as sets.

New theories have been developed throughout the century besides the huge amount of research in ZF. The *Admissible Set Theory* which originated in the Sixties is one of them. Work on admissible ordinals resulted in a first order set theory called KP (Kripke-Platek). Barwise weakened KP to a new theory KPU by readmitting *urelements* which are indivisible individuals [13]. KPU is an elegant theory which supports the cumulative hierarchy. It overcomes some of the disadvantages of ZF and seems to obey the parsimony principle. But it still cannot handle circular sets.

It is the *Hyperset Theory* which can do that. The origins of this theory are in the late Twenties, but it evolved throughout the century. The theory makes use of graphical representation of sets. In the Eighties, Aczel got interested in the subject and proposed the *Anti-Foundation Axiom (AFA)* [4]. With the existence of AFA, one can easily represent circular sets as in Figure 2. But what is more remarkable with the theory is that it does not require other axioms of ZF to be thrown out. Aczel's theory was successfully used by Barwise and Etchemendy in their thought-provoking work on the *Liar Paradox* [14].

2. COMMONSENSE SET THEORY

Representing all compound entities and the relations between their parts in terms of sets gave rise to the success of set theory in mathematics. This also seems to apply to commonsense reasoning as well. McCarthy has mentioned the possible use of set theory in AI and invited researchers to concentrate on the subject [15]. Unfortunately, not much progress has been made since then. In the sequel, we will study some noteworthy research essentially due to Perlis [16], Zadrozny [17], and Mislove et al. [18].

If we want to design an intelligent machine which will work in the realm of human beings, then we must make sure that it has commonsense knowledge and that it is able to make inferences from that knowledge. Commonsense reasoning involves three main parts: a domain theory, knowledge representation, and inference. We are primarily dealing with knowledge representation issues. The first idea is to represent commonsense notions by sets. To take a classical example, we can consider the commonsense notion of *society* as a relation between a set of people, rules, customs, traditions, etc. Here, we face the problem that commonsense ideas do not have precise definitions as mathematical ideas do. For example, in the definition of *society,* the notions of *tradition* and *custom* are as complex entities as the definition itself, and should best be left to intuition. However, sets may still be useful in conceptualizing such terms. One may, for instance, want to consider the set of societies disjoint from any set of individuals. Moreover, the idea of collecting a set of individuals for further thought is still an important process, e.g., the Comprehension Principle.

A theory proposed for use in commonsense reasoning can be examined in a variety of ways. The first point to examine is the concept of set formation, immediately leading to the question whether to admit urelements or not. It seems intuitive to answer affirmatively because this matches with the naive notion of a set as collecting individuals satisfying a property into a whole.

But then we directly face Russell's Paradox. The problem is due to using a set whose completion is not over yet in the formation of another set, or even in its own formation. Then we are led to the question of when to consider a set of individuals satisfying a property as an individual itself. This brings the notion of *cumulative hierarchy* into picture. Cumulative hierarchy is one common construction of our intuition and can be illustrated by the example in Figure 3 where we start with simple blocks, and make towers out of blocks, and make walls out of towers, and so on.

In the cumulative hierarchy, any set formed at some stage must be consisting of urelements (if any) and the sets formed at some previous stage. At this point the problem of "sets which can be members of themselves" arises, because they are used in their own formation. Circularity is obviously a common means of commonsense knowledge representation. For example, we should allow the unique set of all non-profit organizations to be a member of itself, since it may also be a non-profit organization (which is not an unexpected event). So, a possible theory should allow circular sets. Barwise and Etchemendy's work also demonstrates beyond doubt that circularity is an integral part of our daily discourse and hence should be allowed by a commonsense set theory [14].

One further aspect to be considered is "possible" membership. A commonsense set theory may be helpful in providing representations for dynamic aspects of language by

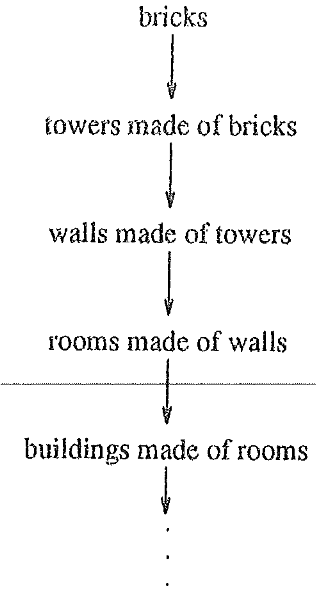


Figure 3: A simple hierarchical construction representing the cumulative hierarchy

making use of *partiality.* We can allow our sets to have possible members as well as real members. Then we can have an operation of *clarification* to determine the real members among possible ones.

Other-set theoretical issues to be considered include cardinality and well-ordering. For example, imagine a box of 34 black and 16 white balls. We know that there are 50 balls in the box, or formally the cardinality of balls in the box is 50. After shaking the box, we would say that the balls are not ordered any more. But this is not true in classical set theory since a set with finite cardinality must have a well-ordering [17]. Counting is another aspect to be considered. While the formal principles of counting are precise enough, we may observe that people also use quantifiers like "many," "more than half." For example, a system which can represent the phrase "A group of people are walking towards me" should not probably answer questions like "Who is the first one?" since there does not exist a well-ordering for the set under consideration.

There is relatively little work done in commonsense set theory. Perlis introduced a series of theories for this purpose. He first proposed CST_0 which is a version of naive set theory with only an axiom of comprehension. Then he extended it to CST_1 using Ackermann's Schema [19] to support cumulative hierarchy but still could not handle circular sets with it. Finally he proposed CST_2 by combining CST_1 with the universal reflection theory of Gilmore and Kripke [20]. While this theory can represent circular sets, its consistency is not proven yet.

Zadrozny, who does not believe in a "super" theory for commonsense reasoning, concentrated on cardinality and well-ordering issues mentioned above and proposed some representation schemes. He introduced a non-standard class of *Nums* for counting purposes and using *Nums*, he gave non-classical interpretations of cardinality and well-ordering. In the context of his interpretations, he proved that there are sets with finite cardinality which do not have a well-ordering and that there are well-orderings elements of which do not form a set (hence solving the Box Problem above).

Mislove, Moles and Oles worked on *protosets*, a generalization of HF, the set of well-founded hereditarily finite sets [18]. Protosets are sets with some packaging which might obscure some elements of the set. They proposed their *Partial Set Theory* based on protosets, with a relativization of Aczel's work. Their theory is consistent with respect to Aczel's, because it is a conservative extension of the latter.

3. CONCLUSION

We conclude that set theory can be useful in commonsense reasoning. The methodology may change: a universal commonsense set theory may be developed, or different set theoretic concepts may be examined and modified. No matter what proposal is followed, we believe that research in this field is promising, considering the current success of set theory in mathematics.

References

[1] P. Suppes. *Axiomatic Set Theory.* Dover, New York, NY, 1972.

[2] P. R. Halmos. *Naive Set Theory.* Springer-Verlag, New York, NY, 1974.

[3] G. Cantor. Fondaments d'une théorie générale des ensembles. *Acta Mathematica*, 2:381–408, 1883.

[4] P. Aczel. *Non-Well-Founded Sets.* Number 14 in CSLI Lecture Notes. Center for the Study of Language and Information, Stanford, CA, 1988.

[5] A. N. Whitehead and B. Russell. *Principia Mathematica (Three Volumes).* Cambridge University Press, Cambridge, UK, 1910–1913.

[6] J. von Neumann. Eine Axiomatisierung der Mengenlehre. *J. für Math.*, 154:219–240, 1925.

[7] J. Kelley. *General Topology.* Van Nostrand, Princeton, NJ, 1955.

[8] A. Morse. *A Theory of Sets.* Academic Press, San Diego, CA, 1965.

[9] W. V. Quine. New Foundations for Mathematical Logic. *Amer. Math. Monthly*, 44:70–80, 1937.

[10] E. Zermelo. Untersuchungen über die Grundlagen der Mengenlehre. *Mathematische Annalen*, 65:261–281, 1908.

[11] A. A. Fraenkel. Zu den Grundlagen der Cantor-Zermeloschen Mengenlehre. *Mathematische Annalen*, 86:230–237, 1922.

[12] M. Tiles. *The Philosophy of Set Theory.* Basil Blackwell, Oxford, UK, 1989.

[13] J. Barwise. *Admissible Sets and Structures.* Springer-Verlag, Berlin Heidelberg, Germany, 1975.

[14] J. Barwise and J. Etchemendy. *The Liar: An Essay on Truth and Circularity.* Oxford University Press, New York, NY, 1987.

[15] J. McCarthy. Acceptance address, 1985. IJCAI Award for Research Excellence, Los Angeles, CA.

[16] D. Perlis. Commonsense Set Theory. In P. Maes and D. Nardi, editors, *Meta-Level Architectures and Reflection*, pages 87–98. Elsevier, Amsterdam, Holland, 1988.

[17] W. Zadrozny. Cardinalities and Well Orderings in a Common-Sense Set Theory. In R. J. Brachman, H. J. Levesque, and R. Reiter, editors, *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pages 486–497. Morgan Kaufmann, San Mateo, CA, 1989.

[18] M. W. Mislove, L. S. Moss, and F. J. Oles. Partial Sets. In R. Cooper, K. Mukai, and J. Perry, editors, *Situation Theory and Its Applications*, pages 117–131. Center for the Study of Language and Information, Stanford, CA, 1990.

[19] W. Ackermann. Zur Axiomatik der Mengenlehre. *Mathematische Annalen*, 131:336–345, 1956.

[20] P. Gilmore. The Consistency of Partial Set Theory without Extensionality. In T. Jech, editor, *Axiomatic Set Theory*, pages 147–153. American Mathematical Society, Providence, RI, 1974.

Yapay Sinir Ağları Uygulamaları

Applications of Artificial Neural Networks

A Survey of Neural Network Aplications for Scheduling Problems

B. Gürgün & İ. Sabuncuoğlu

Department of Industrial Engineering, Bilkent University, Bilkent, Ankara 06533 Turkey

Abstract

Artificial Neural networks (ANNs) attempt to emulate the massively parallel and distributed processing of the human brain. They are being examined for a variety of problems that have been very difficult to solve. The objective of this paper is to review the current applications of ANN to scheduling problems and to identify possible research areas. The paper also adresses to Travelling Salesman Problem (TSP) due to some inherent similarities between scheduling problems and TSP.

1 INTRODUCTION

Scheduling refers to the time-phase allocation of all the system resources such as machines, tasks and materials handling system. In scheduling literature, job and machine pairs can represent patients and hospital equipment, classes and teachers, ships and dockyards, dinners and cooks, cities and traveling salesmen. To perform a job, each of its operations must be processed in the order given by the sequence. The processing of an operation requires the use of a particular machine for a given duration. Each machine can process only one operation at a time. We seek to find a processing sequence of jobs on each machine in order to minimize a given cost function. The cost function consists of one or more of the following basic criteria [1]:

- criteria based upon completion times,
- criteria based upon due dates, and
- criteria based upon the inventory and utilization costs.

Job-shop scheduling is cosidered as the general case of scheduling problem where n jobs, and m machines are modeled ($n|m$ job-shop). Except a few cases the job-shop scheduling problem is in the class of NP hard. As a consequence, a near optimal solution is considered to be good enough for real life problems. In the literature, a number of analytical approaches have been proposed to solve this problem ranging from priority dispatching rules to integer programming formulations. A heuristic method using ANNs might also lead to the near optimal solutions of this problem. This is the basic motivation behind this research. Our objective is to review applications of ANN to scheduling problems and identify possible future research areas.

2 OPTIMIZATION BY ANN

ANN is a new approach to find optimal and near optimal solutions for the NP hard problems. With a few exceptions in all ANN applications to the optimization problems, Hopfield Neural Network Model are used.

Hopfield Neural Networks are highly interconnected networks that minimize a given energy function by local perturbations until the network becomes stable at a local or global minimum. The interconnection weights are determined by the energy function which represents the objective function.

Since TSP is one of the famous NP hard problem in a classical optimization literature, most of the ANN applications to optimization problems are directed to TSP problem [2,3,4,5]. Thus, we give a brief summary of these applications before investigating ANN in the contex of job-shop scheduling problem.

2.1 TSP Solutions by Hopfield Neural Networks

The problem is mapped into a 2D matrix using N^2 neurons (neurons are represented in the matrix as $a_{i,j}, i, j = 1, \ldots, N$), where N is the number of cities. A permutation matrix is required to have a feasible solution. The constraints and the cost function of the problem is coded into the network via the interconnection weights $T_{Xi,Yj}$ (between neurons $a_{X,i}$ and $a_{Y,j}$) determined by the energy function E [2].

$$\begin{aligned}
\mathrm{E} = \; & A/2 \textstyle\sum_X \sum_i \sum_{j \neq i} V_{Xi} V_{Xj} \\
& + B/2 \textstyle\sum_i \sum_X \sum_{X \neq Y} V_{Xi} V_{Yi} \\
& + C/2 (\textstyle\sum_X \sum_i V_{Xi} - n)^2 \\
& + D/2 \textstyle\sum_X \sum_{Y \neq X} \sum_i d_{XY} V_{Xi} (V_{Y,i+1} + V_{Y,i-1})
\end{aligned}$$

The interconnection matrix defined by the above energy function is as follows.
$T_{Xi,Yj} = -A\delta_{XY}(1 - \delta_{ij}) \; -B\delta_{ij}(1 - \delta_{XY}) \; -C \; -Dd_{XY}(\delta_{j,i+1} + \delta_{j,i-1})$
($\delta_{ij} = 1$ if $i = j$ and 0 otherwise)

A, B, C, D are problem dependent scaling constants. The terms of A, B, C are hard constraints in order to have a feasible solution, and D is the soft constraint for satisfying the objective function.

There are two basic problems rising up with Hopfield Neural Networks when size of the problem gets larger:

1. When the energy function gets stable at a local minimum and can not even find feasible solutions because of the complexity of the energy function and size of the problem.

2. The computational requirements of the network increases as the problem size gets larger.

In most of the cases,an annealing approach is used to escape from the local minima. There are a number of different annealing methods recommended in the literature. Some of them are: Simulated Annealing (SA), Fast Simulated Annealing (FSA), Mean Field Annealing (MFA), Rejection-less Simulated Annealing (RSA), etc. The methods are applied for getting to a global minimum instead of a local minimum. In general, the annealing process increases the computational time to get a stable state of the hopfield network.

2.2 Job-shop Scheduling by ANN

2.2.1 Job-shop Sheduling by Hopfield Model

Hopfield Neural Networks have also been applied to scheduling problems. The time-dependent traveling salesman problem may be stated as a scheduling problem in which n jobs have to be processed at minimum cost on a single machine. The set-up cost associated with each job depends not only on the job that precedes it, but also on its position (time) in the sequence. Hence, the $(n|m)$ job-shop problem is more difficult to handle than TSP. Several approaches to the job-shop problem with Hopfield Model are summerized as follows:

Gulati & Iyengar [7] proposed a hopfield model using a mapping of the uniprocessor scheduling problem with hard deadlines, and task priorities. The problem is mapped into a Hopfield Neural Network with $nlog(n_p)$ neurons. Where n is the number of jobs and n_p indicates the total processing times of all jobs. In order to reduce complexity the time axis is scaled logarithmically. The neuron matrix shows the starting times corresponding to each job.

The energy function to be encoded to the network has four terms:

1. Term for tardiness (E_T),
2. Term for waiting time (E_W),
3. Term for overlapping (E_O),
4. Term for precedence (E_P).

Hence the energy function to be minimized by the network will be : $E_{total} = E_T + E_W + E_O + E_P$. In the simulations, Fast Simulated Annealing (FSA) is used to have near optimal solutions with 20 job problem.

Foo and Takefuji [3] developed a hopfield net by mapping the $n|m$ job-shop scheduling problem with the objective of minimizing mean flow time on a mn by $(mn + 1)$ 2D neuron matrix similar to those for solving the traveling salesman problem. Constant positive and negative biases are applied to specific neurons as exitations and inhibitations, respectively, to enforce the operation precedence relationships. At the convergence of neural network, the solution to the job-shop problem is represented by a set of cost function trees encoded in the matrix of stable states. Each node in the set of trees represents a job, and each link represents the interdependency between jobs. The cost attached to each link is a function of the processing time of a particular job. The starting time of each job can be determined by traversing the paths leading to the root node of the tree. A computation circuit computes the total completion times (costs) of all jobs, and the cost difference is added to the energy function of the stocastic neural network. Using a simulated annealing algorithm, the temperature of the system is slowly decreased according to an annealing schedule until the energy of the system is at a local or global minimum. By chosing an appropriate annealing schedule, near optimal and optimum solutions to job-shop problems can be found.

There are three basic (hard) costraints to be encoded in the neural network to have a feasible solution. These conditions are:

1. no more than m number of jobs are allowed to start at time 0,

2. self-dependency on each operation is not allowed, and

3. precedence relationships between operations must be obeyed.

If there are n jobs and m machines available then at most m jobs can start independently at time 0. Constraint of type (1) arises when $n > m$. In this situation, it is necessary to randomly choose m out of n unique jobs to start at time 0. Constraint of type (2) is necessary to avoid self-recurring dependency paths. This condition is enforced by placing strong inhibitions (negative bias) at appropriate neurons such these neurons will not trigger. Constraint (3) is important such that the precedence order of processing each operation is not violated.

Based on the problem formulation the global energy function which contains the constraints of a job-shop problem is defined as:

$$\begin{aligned} \mathrm{E} = \quad & A/2 \textstyle\sum_X \sum_i \sum_{j \neq i} V_{Xi} V_{Xj} \\ & + B/2 (\textstyle\sum_X \sum_i V_{Xi} - z)^2 \end{aligned}$$

The first term is zero if and only if each row in the matrix of neurons do not contain more than one firing neuron. The second term is zero if and only if there are z neurons being turned on in the whole matrix of neurons. The interconnection (conductance) matrix defined by the above energy function is as follows: $T_{Xi,Yj} = -A\delta_{XY}(1 - \delta_{ij}) - B$

The energy function, composed of the hard constraints and the cost of total completion times of all jobs is applied to an annealing schedule. Their method gives feasible and near optimal solutions for 4|3 job-shop problems.

In the future work, Foo and Takefuji [5] represented the same problem in an integer linear programming form with the cost function of the total finishing times of all jobs. They coded the integer linear program to the hopfield network with less complexity and solved the same size problem above.

Integer linear programming neural network (ILPNN) for the job-shop problem handles the following integer linear programming problem:

Minimize

$\sum_{i=1}^{n} S_{ik_i}$

Subject to

$s_{ik} - s_{ih} \geq t_{i,j-1,h}$ if operation (i,j-1,h) precedes (i,j,k)

$s_{pk} - s_{ik} + H * (1 - y_{ipk}) \geq t_{ijk}$ i≥1, p≤n, $1 \leq k \leq m$

$s_{ik} - s_{pk} + H * y_{ipk} \geq t_{pqk}$ i≥1, p≤n, $1 \leq k \leq m$

$s_{ik} \geq 0$

$y_{ipk} = 0 or 1$

where

k_i = The machine which the last operation of job i is assigned.

s_{ik} = The starting time of operation k of job i.

t_{ijk} = The duration of operation j of job i at machine k.

There is a total of mn^2 constraints in the problem. The constraints are encoded to the ILPNN using $nm(nm+1)/2$ neurons. Simulation studies based on solving the linear differential equation of neural network show that the ILPNN approach produces optimal or near optimal solutions, although it does not guarantee the finding of optimal solutions.

Zhou,Cherkassky,Baldwin, and Olson [6] further improved the performance of ILPNN for job-shop scheduling problem. In their approach, they unified the indices indicating operations and machines to have the following integer programming representation of the problem with less number of constraints using nm neurons (similar notation with the above is used). For any two operations (i,k) and (j,p) assigned to the same machine.

Minimize

$\sum_{i=1}^{n} S_{ik_i}$

Subject to

$s_{ik} - s_{ik-1} + t_{i,k} \leq 0$

$s_{i1} \geq 0$

$s_{ik} - s_{jp} + t_{ik} \leq 0$

$s_{jp} - s_{ik} + t_{jp} \leq 0$

The number of constraints and the computational time grows linearly as the size of the problem gets large.

Comparison of the 4|3 job-shop for the three different models:

method	number of neurons	number of interconnections
TSP like [3,4]	156	24336
ILPNN [5]	78	2880
Improved ILPNN [6]	12	48

Finally, Hulle [12] formulated the job-shop problem on a goal programming model and mapped this model into a goal programming network. This kind of network finds the optimal goal programming solution which will also satisfy the job-shop problem's constraints. The solution is obtained by repeatedly performing goal programming relaxations and binary adjustments until convergence. According to this method, an optimal solution for the original problem can not be guaranteed, but the solution is always feasible with respect to the constraints of the job-shop problem.

2.2.2 Other ANN Applications to Machine Scheduling Problem

Chryssolouris, Lee, and Domroese [8] used a back propopagation network to establish adequate weights on the criteria used locally at the workcenter level based on performance measure goals for the job shop.

Carlos and Alptekin [9] proposed an integrated (expert systems and ANN) scheduling system in which, a backpropogation network was used to rank some scheduling rules based on the current status of the system and job characteristics. The outputs (relative weights) of the network were further analyzed by an expert system to generate schedules.

If both of the above studies, ANNs were mainly used for finding appropriate weights of either scheduling rules or decision making criteria, rather that developing schedules directly by a backpropogation network.

Osman and Potts [10] used the simulated annealing approach for the permutation flow-shop problem. The results indicated that SA improved the performance measures better than known constructive analytical algorithms.

3 THE PROPOSED APPROACH & SUGGESTIONS FOR FURTHER RESEARCH

The performance of the hopfield model in large sized problems is not satisfactory. The reason for that is stabilization of the network at a local minima of the energy function. In most of the cases, hard constraints (terms for a feasible solution) as well as soft constraints (terms for optimal solution), are coded into the energy function. In this case, the energy function becomes complicated (i.e. the number of local minima are too many). In general, an annealing process is used to reach to the global minimum. But annealing is not easily applied to all problems. Because the process is time consuming and the solution may be still far from the global minimum due to too many local minima. A better solution may be obtained by a less complicated energy function which is composed of soft terms only. Hence, the number of local minima in the energy function will be significantly reduced. This may be done by an appropriate normalization process. Also, if we can control bias terms of some neurons that are known to have a tendency to reach a certain activation level at the optimal solution, we can reduce the feasible solution set. Hence, further reduction in the number of local minima can be achieved by reducing the range of the energy function. In addition, the biased neurons might be changed during the emulation of the network to adjust the global minimum into the reduced feasible set.

Another approach for decreasing the number of the local minima, and increasing the size of the problems solvable by the network may be dealing with the hard constraints and the soft constraint seperately. This may be done by composing the energy function (interconnections) only by the soft constraints. While the network stabilizes, the activation levels of the neurons can be manupilated so that, the orthogonal projections to the hard constraints (feasible set) are done. This method decreases the number of local minima. The orthogonal projections to the hard constraints guarantees the feasibility of the final stable state of the network. This proposed approach is currently being investigated by the authors.

Another research direction could be to reduce the size of scheduling problems. This can be done by clustering the jobs and machines into groups and solving each of these subproblems. This clustering can be performed by using a self organizing neural network.

References

[1] Rinnooy Kan, A.H.G., Machine Scheduling Problems, The Hague, 1976.

[2] Hopfield, J.J. and Tank, D.W., Neural Decisions in Optimization Problems, Biological Cybernetics, Springer-Verlag, 1985.

[3] Foo, Y.P.S. and Takefuji, Y., Stocastic Neural Networks for Solving Job-Shop Sheduling: Part 1. Problem Presentation, ICNN 1988, Vol.2, pp 275-282.

[4] Foo, Y.P.S. and Takefuji, Y., Stocastic Neural Networks for Solving Job-Shop Sheduling: Part 2. Architecture and Simulations, ICNN 1988, Vol.2, pp 283-290.

[5] Foo, Y.P.S. and Takefuji, Y., Integer Linear Programming Neural Networks for Job-Shop Scheduling, ICNN 1988, Vol.2, pp 341-348.

[6] Zhou, D.N., in: V.Cherkassky,T.R.Baldwin,D.E.Olson, A Neural Approach to Job-Shop Scheduling, IEEE Transactions on Neural Networks Vol.2, No.1, (Jan 1991), pp 175-179.

[7] Gulati, S. and Iyengar, S.S., Nonlinear Networks for Deterministic Scheduling, ICNN 1987, Vol.4, pp 745-752.

[8] Chryssolouris, G., in: M.Lee, M.Domroese, Journal of Manufacturing Systems, Vol.10, No.2, (1991), pp 166-175.

[9] Rabelo, L.C. and Alptekin, S. Synergy of Neural Networks and Expert Systems for FMS Scheduling, Proc. of the third ORSA/TIMS Conf. on FMS, (1989), pp 361-366.

[10] Osman, I.H. and Potts, C.N., Simulated Annealing for Permutation Flow-Shop Scheduling, Omega, Vol.17, No.6, (1989), pp 551-557.

[11] Picard, J.C. and Queyranne, M., Operations Research Vol.26, No.1, (1987), pp 86-110.

[12] Van Hulle, M.M., International Journal of Neural Systems, Vol.2, No.3, (1991), pp 201-209.

A Study in Connectionist Modelling of Hard-constraint Problems: Solving Tangram Puzzles

Kemal Oflazer

Dept. of Computer Eng. and Information Sci. Bilkent University Bilkent, Ankara, 06533 TÜRKİYE
E-mail: ko@trbilun.bitnet, Fax: (90)-4-266-4127

Abstract

We present a connectionist approach for solving Tangram puzzles. Tangram is an ancient Chinese puzzle where the object is to decompose a given figure into seven basic geometric figures. Our connectionist approach models Tangram pieces and their possible placements and orientations as connectionist neuron units which receive excitatory connections from input units defining the puzzle and lateral inhibitory connections from competing or conflicting units. The network of these connectionist units operating as a Boltzmann Machine, relaxes into a configuration in which units defining the solution receive no inhibitory input from other units. We present results from an implementation of our model using the Rochester Connectionist Simulator.

1 Introduction

Tangram is an ancient Chinese puzzle in which the objective is to decompose a given figure into seven basic geometric figures [2].[1] Tangram figures may belong to a broad spectrum, varying from geometrical Tangrams, such as triangles, trapezoids or parallelograms, to representational ones, such as human figures. Our previous work has used traditional artificial intelligence and computational geometry techniques to solve these puzzles [6]. In this paper, we present a connectionist approach for solving Tangram puzzles. Our connectionist approach models Tangram pieces and their possible placements and orientations as units which receive excitatory connections from input units defining the puzzle and lateral inhibitory connections from competing units. The system of units operating as a Boltzmann Machine relaxes into a configuration in which units defining the solution receive no inhibitory input from others. We have implemented this connectionist model using the Rochester Connectionist Simulator [4] and present results from our implementation.

The seven-piece set of Tangram is cut from a square as shown in Figure 1. The small triangles are called basic triangles. The other pieces are all compositions of basic triangles. The rules of Tangram are

[1] Henceforth, we will refer to these geometric figures as Tangram pieces, or pieces for short.

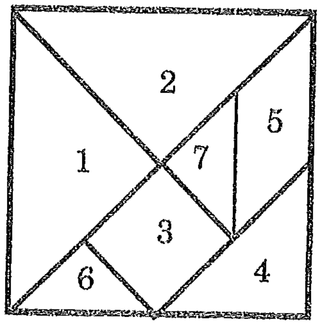


Figure 1: The Tangram pieces: 1,2 - large triangles, 3 - square, 4 - medium triangle 5 - parallelogram, 6,7 - small triangles.

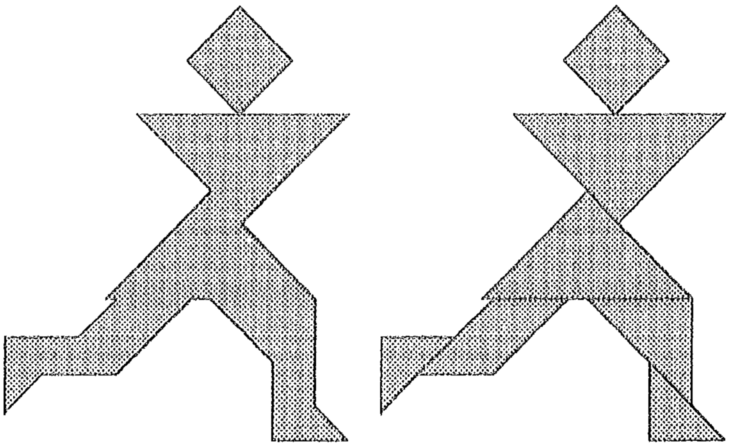


Figure 2: A representational Tangram which depicts a running man, and its solution.

self-evident: the given figure is to be decomposed into the Tangram pieces, and the decomposition has to use all seven pieces. For example, in Figure 2, part (a) shows a Tangram which depicts a running man, and part (b) gives the solution for that Tangram.

2 Grid Tangrams

Grid Tangrams constitute a subclass of Tangrams in which every vertex of each of the seven pieces that comprise the Tangram coincides with the points on a square grid. The grid points are separated from each other by unit length, the edge length of the square piece (cf. Figure 1). Figure 12 later in the paper gives some examples of grid Tangrams and their solutions.

3 Connectionist Models

Connectionism has come out as novel paradigm in artificial intelligence owing to results of the work of the past decade. The basic computational paradigm is based on a network of massive number of simple processing units whose functionality models that of real neurons [12,3]. There have been a tremendous number of applications of this computational paradigm in widely varying areas, such as cognitive science, signal processing, combinatorial optimization etc. It the latter, such network models have been used to obtain approximate solutions to a number of well-known hard problems [7,9].

Connectionist models have been applied to solving puzzles by Kawamoto [10] who has used an auto-associative network with some extensions to implement hill-climbing for solving a number of puzzles including the DOG ⇒ CAT puzzle where the objective is to generate a sequence of 3-letter words starting with DOG and ending with CAT changing 1 letter at a time. Takefuji [13] has used a neural network model to obtain a parallel algorithm for tiling a grid with polyominoes.

3.1 Boltzmann Machines

Our approach in this work uses a Boltzmann Machine model [1] for solving grid Tangram puzzles. Boltzmann Machines are neural network models that are specifically applicable to constraint satisfaction problems which can be (approximately) solved by relaxation search. They combine the Hopfield model [8] with the simulated annealing paradigm [11] to perform relaxation search through a state space. Contrary to Hopfield networks where the state space search may get stuck in local minima of the energy measure being minimized, the Boltzmann Machines introduce a probabilistic component to the state change decision of the neuron units along with the concept of a *Temperature* parameter to simulate annealing. The basic idea behind Boltzmann Machine operation is the following: Each neuron units computes a weighted sum of the inputs that are connected to it. If the sum exceeds the unit's threshold then the output of the unit is set to 1 indicating active unit state. If, however, the sum is below the threshold, the output is determined by a probability distribution (described later in Section 4.4). The unit's output may be set to 1 with a certain probability determined by the distribution. Thus, instead of only moving downhill towards a minimum, the network exhibits some random behavior so that it may occasionally go uphill in the energy space to skip over or jump out of local minimum and not get stuck there.

The probability distribution that determines this random behavior is a function of a parameter commonly called the *Temperature* in reference to the use of temperature in the process of annealing [11]. The higher the temperature, the higher the probability a unit may turn on even though its net input is below the threshold. During network's operation, the temperature is reduced slowly, reducing the chances that a unit may switch on with a given net input.

Boltzmann Machines are especially suited for constraint satisfaction problems where neuron units represent hypotheses and the connections between units represent constraints among hypotheses. Hypotheses which mutually reinforce each other have positive weights on their mutual connections, which competing hypotheses have negative weights. However, such machines are more suited to deal with *weak constraints* where the problem is to find a configuration of units in which the most important constraints (indicated by connection strength) are satisfied. This is in contrast to *hard constraint* problems where the objective is to find a configuration where all constraints are satisfied. Solving Tangram puzzles involves satisfying a number of hard constraints, but as will be seen later, Boltzmann Machines have been successful in solving this problem.

4 A Connectionist Model for Grid Tangrams

4.1 Representing the puzzle

Our model represents a grid Tangram puzzle as sets of active units representing grid points covered by the puzzle. Since such Tangrams may have holes or concave boundaries, we have chosen to represent each grid point by a set of 8 units as shown in Figure 3. The grid points are labeled with coordinates (i, j), i indicating the row and j indicating the column in the grid in standard matrix notation. The eight units associated with each grid point indicate in which orientations the *puzzle area* or *boundary* extends around that grid point. The orientations are: Left(L), Right(R), Down(D), Up(U), Right–Up(RU), Right–Down(RD), Left–Up(LU), Left–Down(LD). A grid Tangram puzzle is represented by externally setting the grid units for relevant orientations of each point covered by the puzzle. For example, a grid point which is totally covered by the puzzle has all of its orientation units on, so does a point which is on a boundary but has a wedge missing as shown in Figure 4. Figure 5 shows a complete example of the representation of a grid Tangram in our model. The reader may notice that there is a rather superficial similarity between our representation and Freeman's "chain code" [5].

4.2 Representing the Tangram Pieces

In grid Tangrams, the placement and orientations of the base puzzle pieces are limited. For example the only way that the square piece can appear is when its corners are on the grid. There are 4 possible orientations of each of the other pieces:

1. *Small triangles* can appear in orientations Left-Up (LU), Left-Down (LD), Right-Up (RU) and Right-Down (RD) depending on where the right angle corner of the triangle is placed with respect to the grid point (i, j). If the corner is on point (i, j) then it is in LU orientation, if the corner is on point $(i+1, j)$ then it is in LD orientation, if the corner is on $(i, j+1)$ then it is in RU orientation and when the corner is on $(i+1, j+1)$, it is in RD orientation. When we talk about the "LD

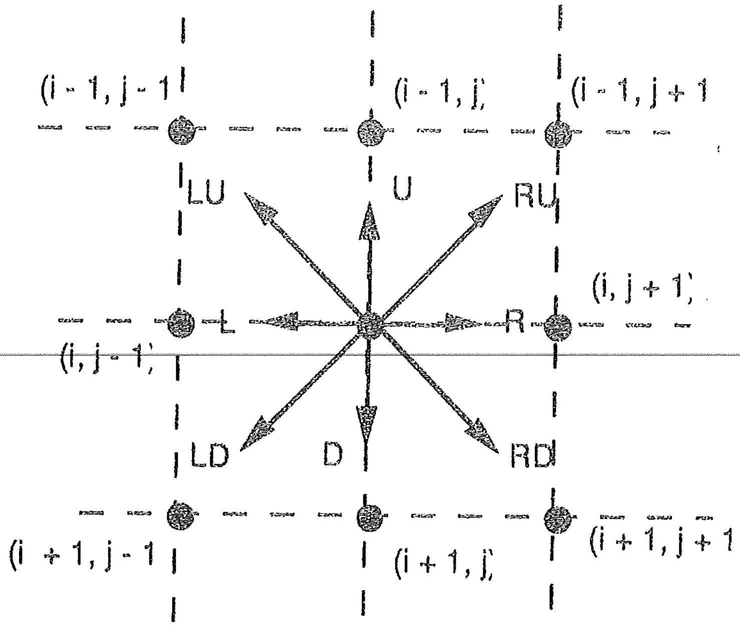
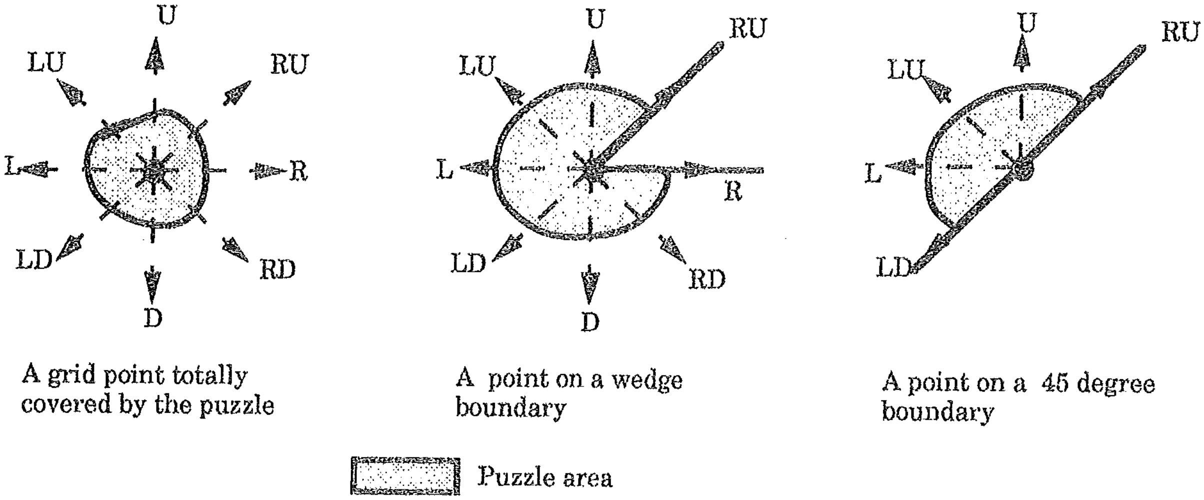


Figure 3: Representation of grid points by a group of 8 units

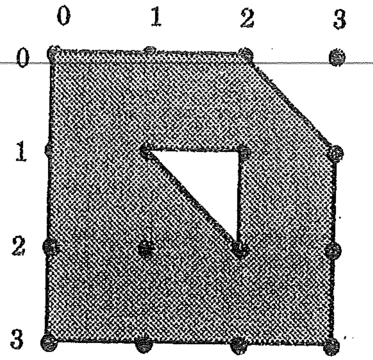


A grid point totally covered by the puzzle

A point on a wedge boundary

A point on a 45 degree boundary

Figure 4: Representation of the puzzle space around a grid point by orientation units



(0,0):	R, RD, D
(0,1):	L, LD, D, RD, R
(0,2):	L, LD, D, RD
(1,0):	U, RU, R, RD,D
(1,1):	L, LU, U, RU, R, RD, D, LD
(1,2):	L, LU, U, RU, R, RD, D
(1,3):	LU, L, LD, D
(2,0):	U, RU, R, RD, D
(2,1):	L, LU, U, RU, R, RD, D, LD
(2,2):	L, LU, U, RU, R, RD, D, LD
(2,3):	U, LU, L, LD, D
(3,0):	U, RU, R
(3,1):	L, LU, U, RU, R
(3,2):	L, LU, U, RU, R
(3,3):	U, LU, R

Figure 5: An example puzzle representation showing the orientation units activated for grid points covered by the puzzle

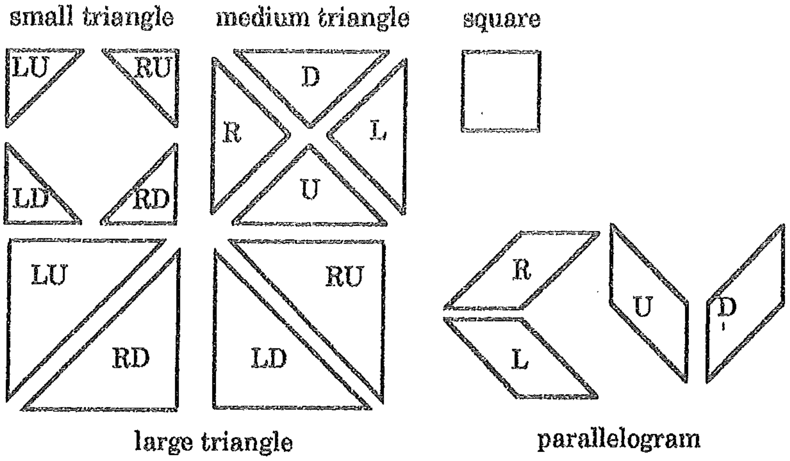


Figure 6: Orientations and labeling of Tangram pieces

small triangle at (i, j)", we mean the triangle with the right-angle corner at $(i+1, j)$ with the other corners placed at (i, j) and $(i+1, j+1)$.

2. *Medium triangle* can appear in orientations Up (U), Down (D), Left (L) and Right (R) depending on where the right-angle corner points to. For example, when we talk about the "R medium triangle at (i, j)", we mean the medium triangle with the right angle corner at $(i+1, j+1)$ and the two other corners at (i, j) and $(i+2, j)$. Similarly, a U medium triangle at (i, j) has its right angle corner on $(i, j+1)$ and its two other corners at $(i+1, j)$ and $(i+1, j+2)$.

3. *Parallelogram* can appear in orientations L, R, U and D. In the U and D orientations the longer dimension is along the vertical axis, and in L and R orientations the longer dimension is along the horizontal axis. For example the "L parallelogram at (i, j)" has its corners at (i, j), $(i, j+1)$, $(i+1, j+1)$, $(i+1, j+2)$.

4. *Large triangle* can be placed in the same orientations as the small triangle - i.e., LU, LD, RU, RD - except that these pieces are larger. For example, the "RD Large triangle at (i, j)" has it right angle corner at $(i+2, j+2)$, and the two other corners at $(i, j+2)$ and $(i+2, j)$.

Since the grid is finite, certain placements of the pieces around the boundary are not applicable. Figure 6 shows the set of possible orientations for all the pieces.

In a grid of size I rows by J columns, there are $4(I-1)(J-1)$ possible placements of a small triangle, $(I-1)(J-1)$ placements of the square, $2(I-1)(J-2)+2(I-2)(J-1)$ possible placements for the medium triangle and the parallelogram, and $4(I-2)(J-2)$ placements for a large triangle. In our connectionist model, all these placements of pieces are represented by neuron units (depicted in Figure 7) which receive excitations from input units representing the grid area covered by the puzzle and inhibitory inputs from the outputs of other conflicting neuron units. They sum their inputs and determine their output probabilistically in the manner described later in Section 4.4. For example, the unit representing a square at (i, j) receives excitatory inputs from the following grid orientation units:

1. For grid point (i, j): R, RD, and D units.
2. For grid point $(i, j+1)$: L, LD, and D units.
3. For grid point $(i+1, j)$: U, RU and R units.
4. For grid point $(i+1, j+1)$: U, LU, and L units.

Similarly for instance, the unit representing an LD large triangle receives excitatory inputs from the following grid orientation units:

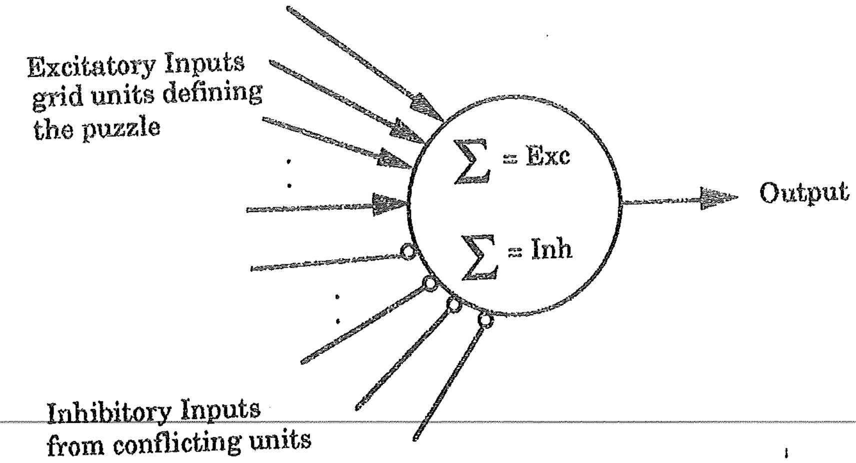


Figure 7: A neurón unit representing a single placement and orientation of a Tangram piece

1. For grid point (i, j): D, RD units.
2. For grid point $(i + 1, j)$: U, RU, R, RD, D units.
3. For grid point $(i + 2, j)$: U, RU, R units.
4. For grid point $(i + 1, j + 1)$: LU, L, LD, D, RD units.
5. For grid point $(i + 2, j + 1)$: L, LU, U, RU, R units.
6. For grid point $(i + 2, j + 2)$: LU, L units.

Figure 8 shows the inputs for one orientation of each of the 5 distinct Tangram pieces.

4.3 Representing Placement Constraints

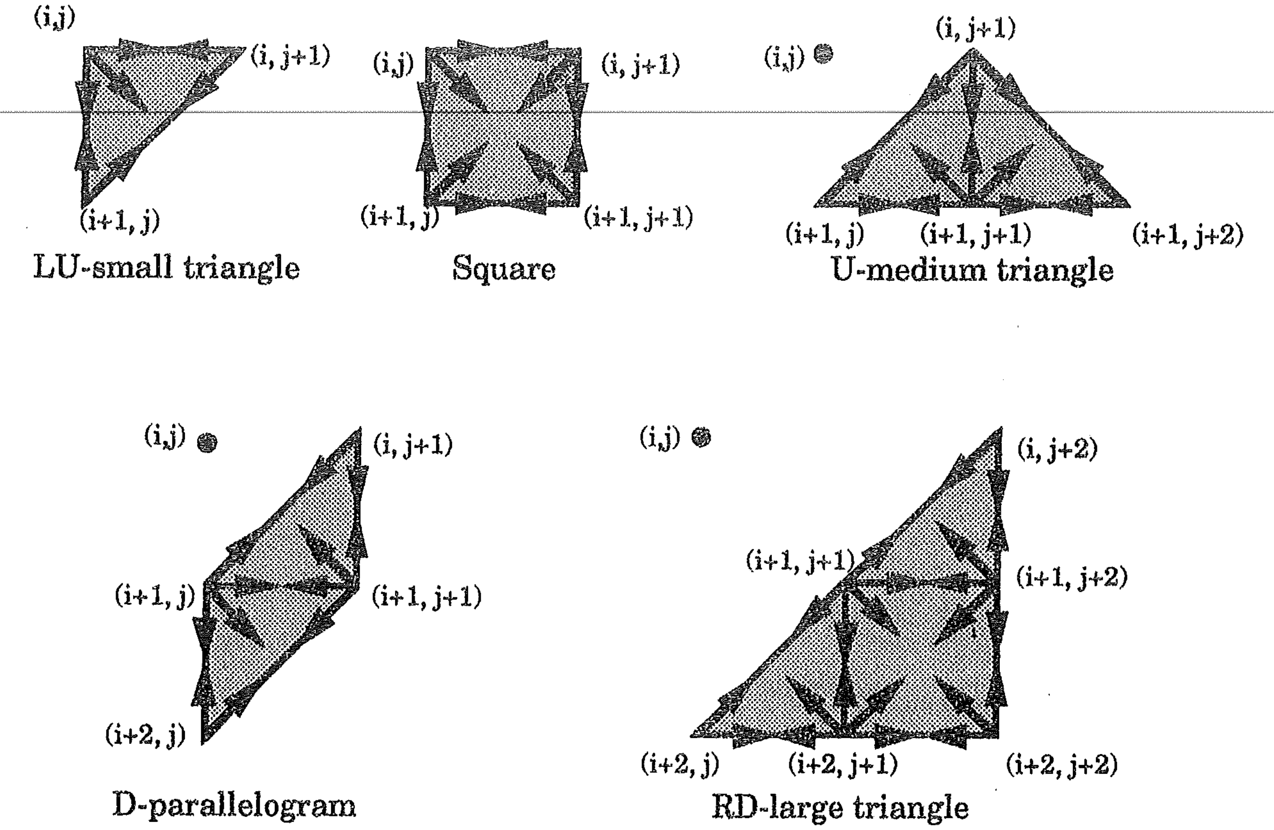
Any unit which has *all* of its excitatory inputs active can be part of a solution of the given grid Tangram puzzle provided it does not conflict with another unit. The conflicts can be in one of two ways:

1. Only one of the units representing a class of units (e.g., squares) can be active as there is only one such piece in the puzzle set.[2]
2. A given piece of area on the puzzle can be covered by only one piece, hence only one of the units covering an area can be active.

These constraints are represented by lateral inhibitory links between conflicting units. For the first set of the constraints above, we use a standard *winner-take-all* network organization [3]. All units within a single class (e.g., squares or medium triangles) are linked to each other with inhibitory links representing the fact that only one of them can be active. Thus, within the resulting connectionist network for a given Tangram puzzle, there are 7 separate such winner-take-all networks.

For the second set of constraints, we establish mutually inhibitory links between any two units representing piece orientations and placements whose areas of coverage intersect. For example Figure 9 shows all the other kinds and orientations of pieces corresponding to the units with which a single UP medium triangle unit has mutually inhibitory links. These correspond to the second set of constraints above. The

[2] For convenience, we treat the second small and large triangles as distinct classes of units, that just happen to have the same shape as their counterpart.



Arrows originating from a grid point indicate the grid point orientation units sending excitatory input to the units representing the piece with the given orientation.

Figure 8: Excitatory inputs to the some of the units

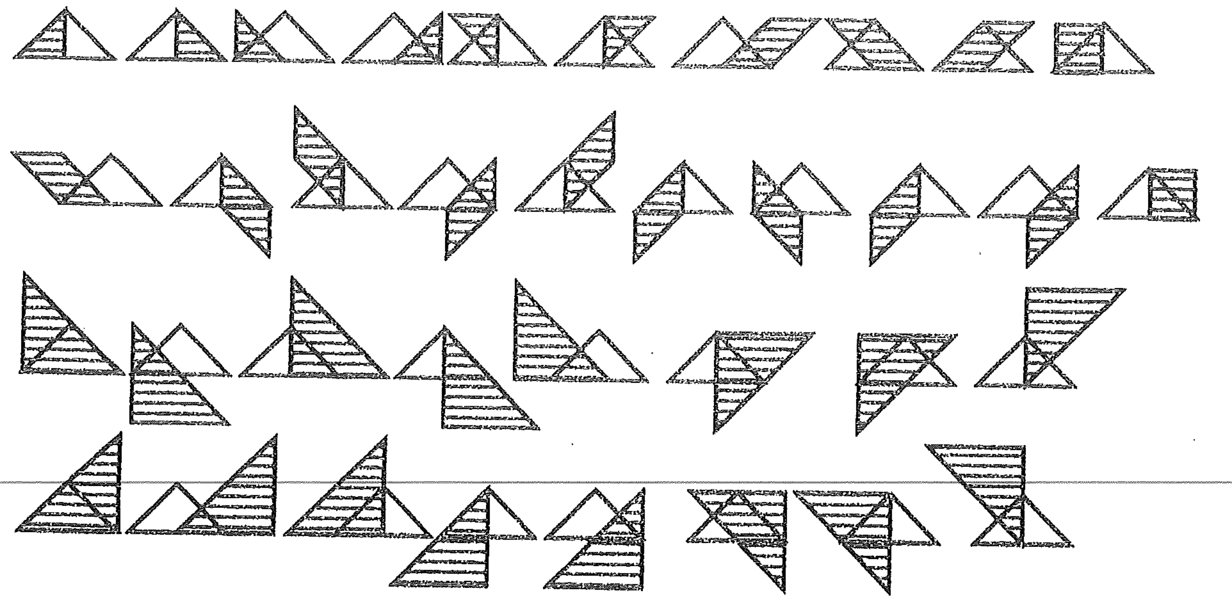


Figure 9: Piece orientations for units (shown cross-hatched) mutually inhibitory with unit for a medium triangle piece in the UP orientation (shown in white)

Grid Size $I \times J$	Number of Units	Excitatory Links	Inhibitory Links	Avg. Inh. Links per Unit
4×4	161	1,892	9,472	58.83
5×4	224	2,688	16,008	71.46
9×3	288	3,376	22,696	78.81
6×4	287	3,484	23,794	82.91
5×5	312	3,824	27,488	88.10
6×7	626	7,912	88,474	141.33

Table 1: Network Statistics from a number of grid configurations

inhibitory links enforcing the first set of constraints handles overlaps with other medium triangles. Thus, in the solution, a given piece of area of the puzzle is covered by only one piece which inhibits all other competing pieces.

Figure 10 shows the general architecture of the resulting network and Table 1 shows some network statistics from a number of different grid configurations. It should be noted that the average number of inhibitory links per unit is rather high.

We use the following approach for setting the weights: For the excitatory links from the grid input units the link weights are set to $1/G$ where G is the number of grid point orientation units that are connected to a given unit.[3] For the inhibitory links we use a link weight -1.1, so that a unit with all excitatory inputs 1 and one inhibitory input active gets a net input of -0.1 which gives it a reasonable probability of switching on. Figure 11 shows a plot of the logistic function and the probability of a unit's changing state for different temperatures and number of inhibitory inputs.

This network operates as a Boltzmann Machine where units determine their output according to a certain probability distribution which changes depending on a *Temperature* parameter, T, which is slowly reduced

[3] The only reason the the weights are selected in this way is that when *all* the excitatory inputs to a unit are active then the net excitatory input to a unit will be 1. $G = 7$ for the small triangle units, 12 for square, medium triangle and parallelogram units, and 22 for the large triangle units.

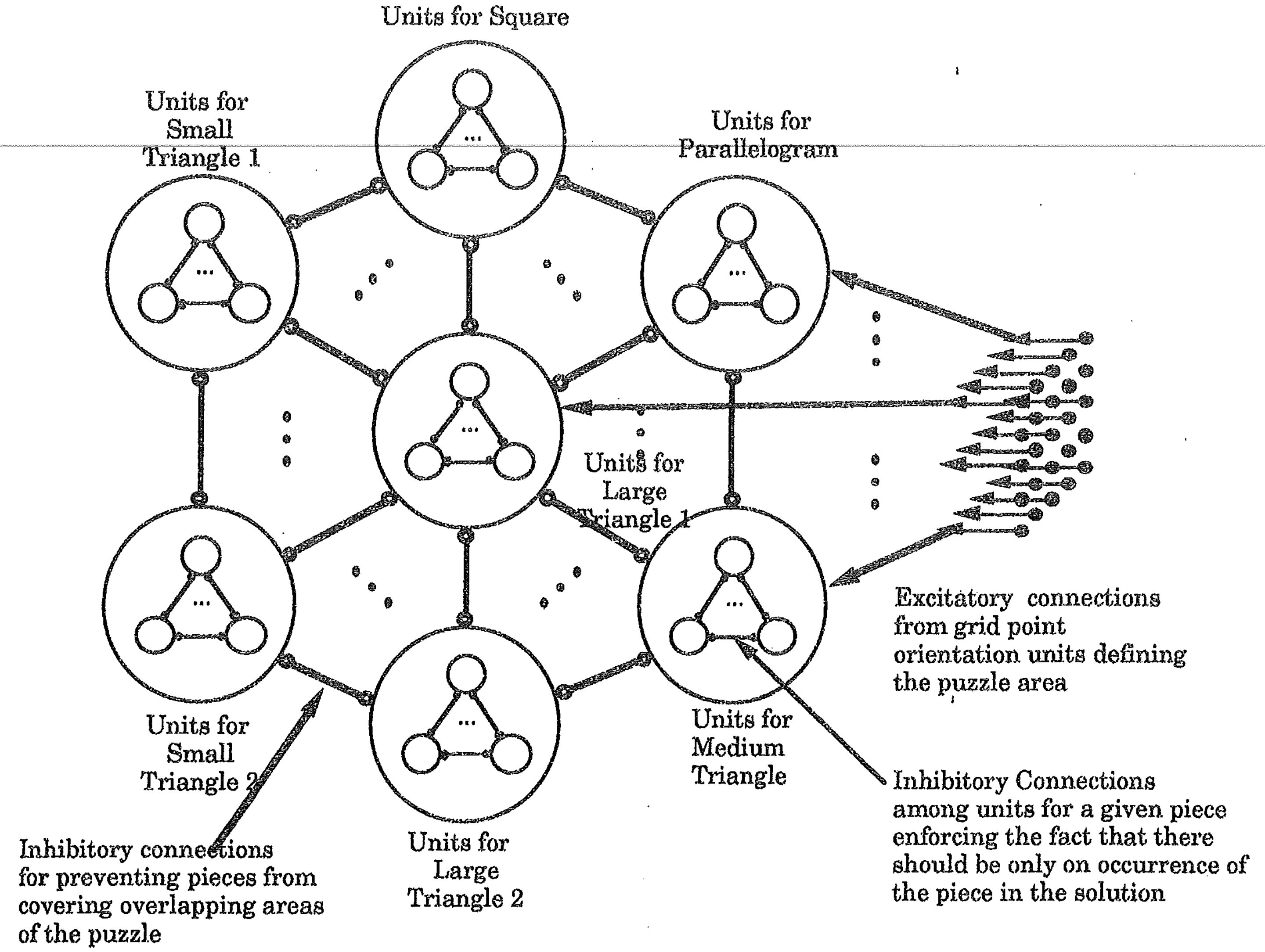
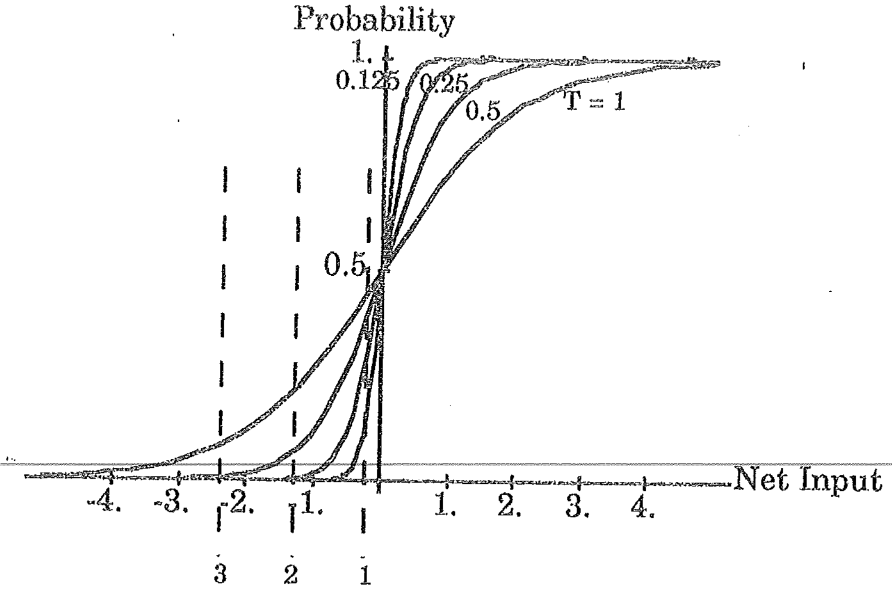

Units for Square
Units for Small Triangle 1
Units for Parallelogram
Units for Large Triangle 1
Units for Small Triangle 2
Units for Large Triangle 2
Units for Medium Triangle
Excitatory connections from grid point orientation units defining the puzzle area
Inhibitory Connections among units for a given piece enforcing the fact that there should be only on occurrence of the piece in the solution
Inhibitory connections for preventing pieces from covering overlapping areas of the puzzle

Figure 10: General architecture for the Tangram network



Probabilities for units with 1, 2 or 3 inhibitory inputs
at different Temperatures

Figure 11: The logistic probability function for Bolztmann Machines

as time progresses.

4.4 Operation of the network

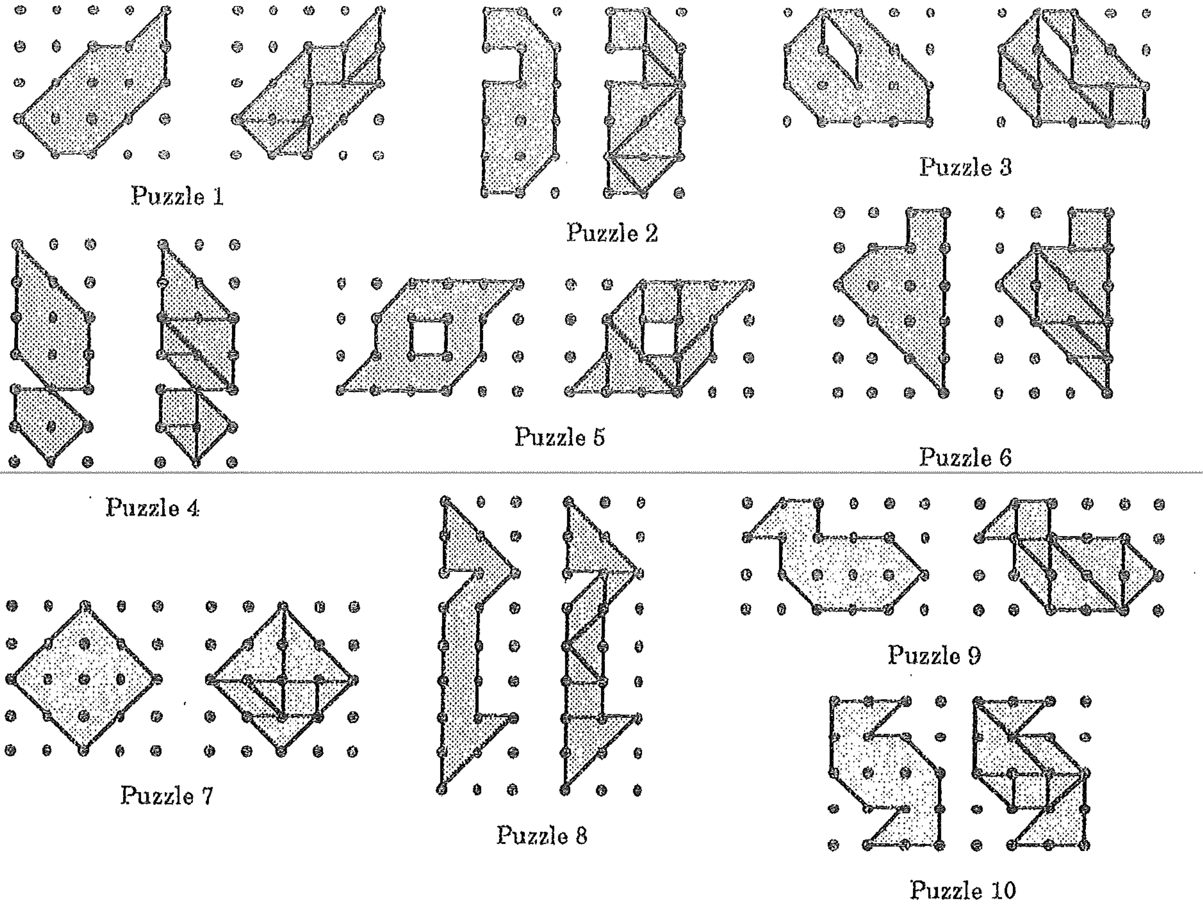
Once the network is constructed, the grid units corresponding to the grid points covered by the puzzle are activated and these send excitatory inputs to all the units. With T, set to its initial value, the units start operating in the following manner:

- If the total excitatory input to the unit is less than 1.0 (i.e., some of the underlying grid units are not active), then the output of that unit is set to 0 since this unit can never be a part of the solution.

- If the total excitatory input to the unit is 1.0, and there is no inhibitory input the unit output is set to 1.0.

- If the total excitatory input to the unit is 1.0, the unit output is set to 1.0 with a probability $p = 1/(1 + e^{-net/T})$ where $net = 1 + inh$ is the net input to the unit with inh being the total inhibition (hence negative). Thus, units receiving a slightly negative net input (i.e., only one inhibitory input active) have a good chance (around 0.5 when the temperature is high) to turn on, while units receiving more inhibitory input have a much less chance of turning on.

In a cycle, the network of units are updated in an asynchronous manner in a random order determined by the simulator. T is reduced by a factor of 0.9 after every I iterations determined by the cooling schedule. Thus as the temperature "cools down," the probability of a unit with a given negative net input switching on is reduced. If, in three consecutive cycles, no units change state, then this is interpreted to be a local extremum and the temperature is slightly raised to force the network out. For all the puzzles that we have experimented with, the corresponding networks have converged to a solution in which 7 active units do not receive any inhibitory inputs.



Puzzle 1

Puzzle 2

Puzzle 3

Puzzle 4

Puzzle 5

Puzzle 6

Puzzle 7

Puzzle 8

Puzzle 9

Puzzle 10

Figure 12: Some grid Tangram puzzles and their solutions

4.5 Results

We have modeled a number of grid Tangram puzzles using the model described above. Ten such puzzles are shown in Figure 12. We have tried a number of cooling schedules for the Boltzmann Machine by changing the initial temperature and the temperature change cycle – obviously many similar schedules are possible. Table 2 shows the number of cycles required for converging to a solution for the puzzles shown in Figure 12 for four cooling schedules. It can be observed that no schedule is in general better than the others. It is also possible to change the strength of the inhibitory connections to more negative values and get a (possibly) slightly different behavior. Figure 13 shows the plot of the range of cycles required for convergence to a solution, versus the cumulative count of the runs, for 100 runs of two puzzles (puzzles 7 and 10) using the last schedule in Table 2. The only difference among these runs is the initial random number seed that determines the order of updates. It can be seen that a large percentage ($\approx$ 70%) of runs converge to a solution in relatively small number of iterations (< 500) in both cases.

5 Conclusions

We have presented a connectionist model for solving grid Tangrams based on the Boltzmann Machine model. This model represents the placements and orientation of the Tangram pieces by units which receive excitatory inputs from grid point units included in the puzzle area. The placement constraints between the units are represented by inhibitory links between conflicting units. We have implemented this model using the Rochester Connectionist Simulator and presented results from our implementation. Our

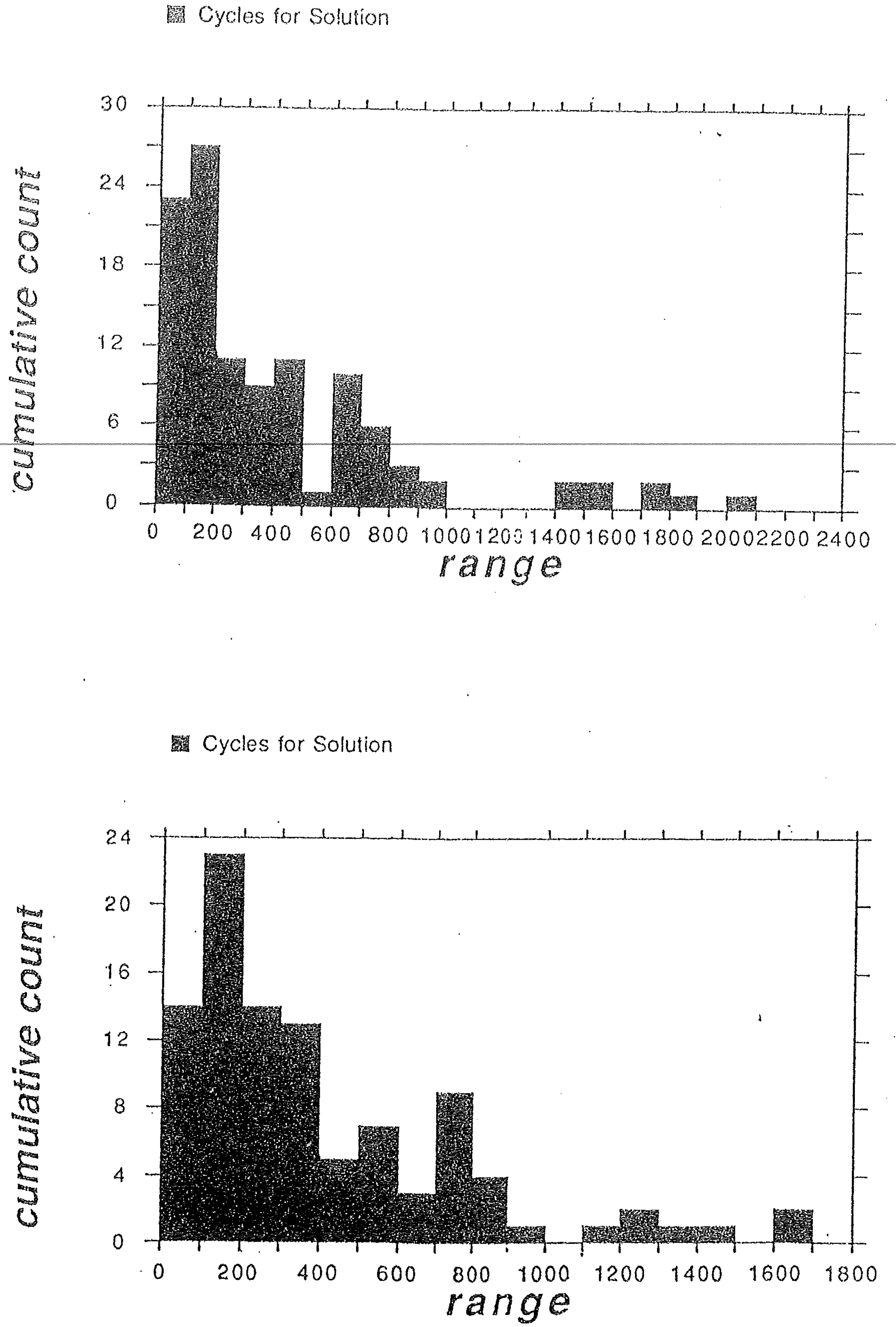


Figure 13: Distribution of cycles for solution for puzzles 7 and 10

Sch.		Puzzles									
T_0	I	1	2	3	4	5	6	7	8	9	10
1.0	10	618	392	934	129	1603	201	605	249	350	197
1.0	5	1423	188	103	92	2254	108	200	548	705	534
0.5	10	101	80	1459	377	255	353	225	276	332	201
1.0	5	1055	763	783	2079	791	756	184	1291	464	70

T_0 is the initial temperature and I is the number of cycles after which the temperature is reduced by 0.9.

Table 2: Cycles required for solution for various Boltzmann Machine cooling schedules

results show that the networks constructed for a variety of puzzles converge in a few hundred iterations.

References

[1] D. Ackley, G. Hinton, and T. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9, 1985.

[2] J. Elffers. *TANGRAM, The Ancient Chinese Shapes Game.* Penguin Books, London, 1976.

[3] J. A. Feldman and D. H. Ballard. Connectionist models and their properties. *Cognitive Science*, 6:205 - 254, 1982.

[4] J. A. Feldman, M. A. Fanty, and N. H. Goddard. Computing with structured neural networks. *Communications of ACM*, 31(2):170 - 187, February 1988.

[5] H. Freeman. Computer processing of line-drawing images. *ACM Computing Surveys*, 6(1):57–93, March 1974.

[6] Z. Güngördü and K. Oflazer. Solving Tangram puzzles. In M. Baray and B. Özgüç, editors, *Proceedings of ISCIS VI - International Symposium on Computer and Information Sciences*, volume 2, pages 691–702, Amsterdam, October 1991. Elsevier Science Publishers BV.

[7] J. Hopfield and D. Tank. "Neural" computation of decisions in optimization problems. *Biological Cybernetics*, 52:141–152, 1985.

[8] J. Hopfield and D. Tank. Computing with neural circuits: A model. *Science*, 233:625–633, 1986.

[9] A. Kahng. Travelling salesman heuristics and embedding dimension in the Hopfield model. In *International Joint Conference on Neural Networks*, volume 1, pages 513–520, Washington 1989, 1989. IEEE Press, New York.

[10] A. H. Kawamoto. Solving puzzles with a connectionist network using a hill-climbing heuristic. In *Proc. of the Conference on Cognitive Science*, pages 514–521, 1986.

[11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671 - 680, May 1983.

[12] J. McClelland, D. Rumelhart, and the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 2. MIT Press, Cambridge, MA, 1986.

[13] Y. Takefuji and K.-C. Lee. A parallel algorithm for tiling problems. *IEEE Transactions on Neural Networks*, 1(1):143–145, March 1990.

Solving maze problems by cellular neural networks

[1]Uğur Halıcı and [2]Ufuk Yaranlı

[1,2] Department of Electrical and Electronics Engineering, 06531, METU, Ankara, Turkey,
[1] email: halici@trmetua.bitnet

Abstract

In this study, the problem of orientation in maze like patterns is aimed to be solved by using neural networks in which the neurons are organized as cellular arrays. The maze patterns are coded in the array of the neurons by considering the corresponding cells in the maze. A probabilistic algorithm that handles the loops in a way similar to Tremaux's algorithm is proposed to find out a path from the initial cell to the final one by using only the local informations. The paths found in this way are not necessarily the shortest. A competitive learning algorithm is introduced by which the network learns one of these paths. The path learned in this may not be the shortest one but the learning strategy favors shorter paths as confirmed in the experimental results.

1. INTRODUCTION

A maze or a labyrinth is a network of paths through which it is hard to find one's way [5]. In greek mythology, Labyrinthos is the name a building with intricate passageways designed by Daedalus for King Minos of Crete to bewilder the uninitiated . The Minotaur, a monster with a bull's head and a man's body was kept there and fed with human flesh. Theseus, the chief hero of Athen, killed the Minotaur and escaped from the Labyrinth with the help of a silken cord given him by Ariadne, the doughter of King Minos [1]. Architectural labyrinths of this sort were not uncommon in the ancient world. The temple in Egypt, in the east to the Moiris lake, that contained 3.000 chambers, 1500 of which is under ground, is an example of this kind. The labyrinth in Lemnos having 150 columns, and the labyrinth in Closium, Italy, containing rooms on within another, are the other examples. Throughout the Middle Ages the walls and floors of many catedrals in Continental Europe, were decorated with mazes that symbolized the snakelike twists of sin and the diffuculty of keeping on the true path. In England, mazes were cut in the turf outside the churches, and they were traversed as a part of a religious rutal . Garden mazes made of high hedges and intended solely for amusement became fashionable during the late Reneissance. The one in the Hampton Court Palace, England, and the other in Versailles Palace, France, are the famous gardens with labyrinths[1,3].

From the mathematical standpoint a maze is a problem in topology. A maze can be solved quickly on paper by shading all the blind alleys until only the direct routes remain. But in the case a map of the maze is not available, such a method is not applicable. If the maze has one entrance, and the object is to find the way to the only exit, it can always be solved by placing your hand against the right (or left) wall and keeping it there as you walk. In such a solution, to reach the exit is guaranteed, although the route is not likely to be the shortest one. The same method can be applied for the mazes in which the goal is within the labyrinth, provided there is no route by which you can walk around the goal and back to where you started. If the goal is surrounded by one or more such closed circuits, the hand-on-wall method simply takes the searcher around the largest circuit and back out of the maze [3].

Mazes having no closed circuits are called 'simply connected' (Figure 1.a). If a maze is simply connected, then it means that it has no detached walls. Mazes with detached walls contain closed circuits, they are called 'multiply connected ' (Figure 1.b). The Tremaux Algorithm formulated by Edouard Lucas in 1882 is able to solve all mazes, including multiply connected ones with closed loops that surrounds the goal. As the maze is walked through , a line is drawn on one side of the path, say the right side. When a new junctures of the paths is confronted, any path can be chosen. If in walking along a new path, a previously visited junction is confronted or a dead end is reached, turn around and go back the way you came. If in walking along an old path, which are already marked on the left, a previously visited juncture is reached, then a new path is taken , if one is available, otherwise an old path is taken. A path marked on both sides is never entered [3] .

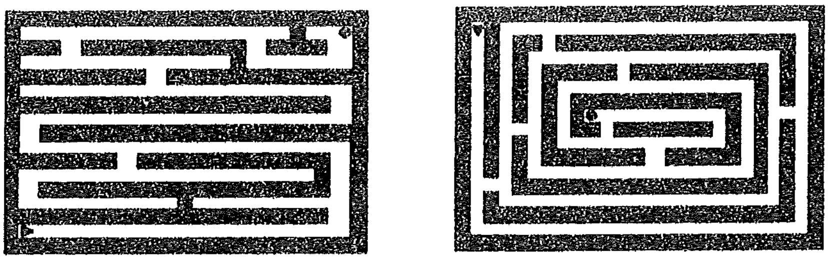


Figure 1. a) Simply connected maze b) Multiply connected maze

Psychology and computer are the fields of science in which interest in mazes is high. Psychologists uses mazes for several decades to study the learning behaviour of men and animals. For the computer point of view, they can be used for the development of the learning machines. In this paper, cellular neural networks are used to solve mazes. For this purpose the maze pattern is placed in the cellular array. In such a maze, while some of the cells permit to pass through them, some others forbid. Therefore the set of cells that permit to pass through them constitutes some passages in the maze and the elements that forbid to pass constitutes the walls.

Given a maze pattern, the aim is to find out a path in the passages connecting a predetermined initial cell to a final one, and meanwhile to learn a reasonable short path as the experiment on the maze is repeated for several times.

A probabilistic algorithm for cellular neural networks, that handles the loops in a way similar to Tremaux's algorithm is proposed to find out a path from the initial cell to the final. A path found in this way is not necessarily the shortest one. But a competitive learning algorithm is introduced by which the network learns the paths favoring the shorter paths. It has been observed through the simulations that most of the time the network learns the shortest or the next shortest paths.

2. CELLULAR NEURAL NETWORKS

The structure of the cellular neural networks is similar to the cellular automata. Each unit is a neuron and it is called cell. Any cell in the network is connected only to its neighbor cells and they can interact directly with each other. Cells which are not neighbours can affect each other indirectly because of the propogation effects of the network [2,4]. The structure of the cellular neural network used in this paper is shown in Figure 2.

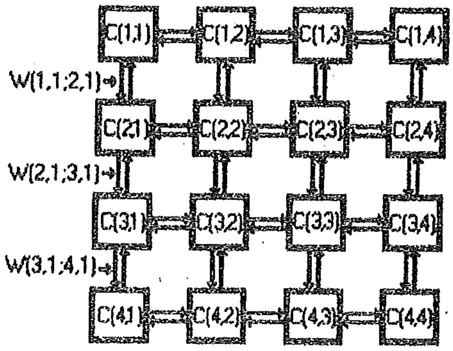


Figure 2. The structure of the two dimensional cellular net used for solving maze problem

Here C(i,j) is used to denote the cell in row i, column j. For a cell C(i,j) the neighbors are

$$N(i,j)=\{C(i-1,j),C(i+1,j),C(i,j-1),C(i,j+1)\} \qquad (1)$$

which are the cells in the left, right, up and down respectively. The notation (i,j;k,l) is used to denote the connection from C(i,j) to C(k,l), and W(i,j;k,l) to denote its weight. In this paper, the connection weights are not assumed to be symmetric, that is W(i,j;k,l) is not necessarily equal to W(k,l; i,j) . Furthermore, it is guaranteed that the summation of the weights of the connections from a cell to the neighbors is equal to one, that is for each C(i,j) in the network:

$$W(i,j;i+1,j)+W(i,j;i-1+j)+W(i,j;i,j-1)+W(i,j;i,j+1)=1 \qquad (2)$$

Here, the cellular neural network is not used in the classic meaning. The structures of the units and the connections are almost the same as before, but the connections are activated instead of the units for describing a path from the initial cell to the final. Only the strengths of the active connections are updated when the network is forced to learn this path. The outputs of the neuorons are either 0 or 1. The connection weights are effective in choosing the next neuron in the path, whose output is to be set to 1.

For a cell C(i,j) the connection weights are assigned initial values as follows: W(i,j;k,l) is set to 0 and never changed if C(k,l) is a cell in the wall, else W(i,j;k,l) is set to 1/n where n is the number of neighbors that are in the passages. The weights of the connections are changed by learning.

For the maze problem, one of the cells is predetermined as the initial, and one another as the final cell. A maze pattern is represented by clamping the outputs of the cells in the walls to 0. However the outputs of the other cells are not clamped, so they may have output value either 0 or 1. A cellular array to represent the maze pattern given in Figure 1 b, is shown in Figure 3. In the figure, the connections are not shown for the simplicity and the dark cells, corresponds to the neurons whose output are clamped to 0 since they are in the walls.. The other cells corresponds to the passages, and the shaded part represent the cells taking place in a path from the initial cell to the final one. For the given example, the cell C(2,2) is the initial cell and C(8,10) is the final. In the figure several different paths connecting the initial node to the final are shown.

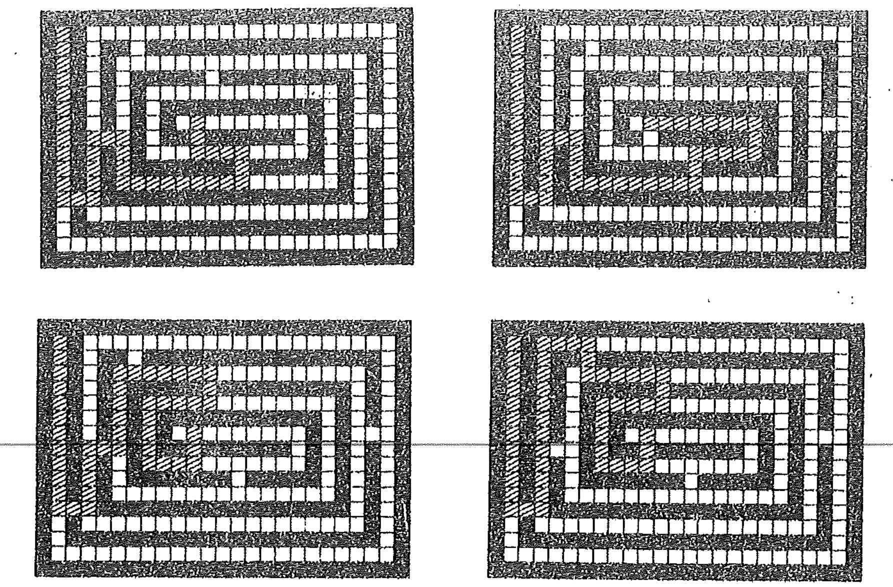


Figure 3. The cellular array representing the maze of Figure 1.b. and some possible paths having length a) 38 units b) 46 units c) 46 units d) 50 unit

3. WALKING IN THE PASSAGES

At a time, only one of the cells has output value 1 indicating where we are in the maze. For the next time, one of the neighbors of the current cell becomes on and the previous node becomes off. Such a cell in the neighborhood is chosen probabilistically by considering the connection weights. The neighbor having connection with larger weigth is more probably to be chosen. For this purpose, each neighbor is assigned a random value between 0 and the weight of the corresponding connection. Then, the one being assigned the largest value is chosen as the next node to be visited. Since the connection weights to the wall cells are initially set to 0 and never changed, such a cell is never selected as the next one.

In the beginning, the neuron for the initial cell has output value 1, and all the others are 0. Finally it is expected that the final node will be on and it will remain as 1, unless a new experiment has not been started. Since the wall cells are clamped to 0, they never have value 1, that means it is not possible to pass through a wall cell.

Within an experiment, the network is assumed to be able to remember the activated path in its short term memory. When a cell C(k,l) in N(i,j) is chosen just after cell C(i,j) then the connection (i,j;k,l) is activated if connection (k,l;i,j) is not already active, otherwise the connection C(k,l;j,i) is deactivated instead of activating C(i,j;k,l). The first part corresponds to propagating in a passage and the second corresponds to return. Special care should be taken for the loops of the passages. Such a situation is implied if one of the outgoing connection say C(i,j;k,l) is already active when we arrive a cell C(i,j) and it can be checked by using local information. If existence of such a loop is distinguished by the active connection (i,j;k,l) when we arrive in C(i,j), the algorithm directs to choose C(k,l) as the next cell and deactivate the connection (i,j;k,l). Therefore the loop is erased by passing through it for a second time in the same direction. When the final node is reached, which is the end of the experiment, we have a path made of active connections whose information is stored in short term memory.

4. LEARNING THE PATHS

Initially, the weights of the connections going out a cell is set inversely proportional to the number of neighbors that are in the passages but not in the walls. When an experiment terminates, only the weights of the connections going out from the cells that are on the path are updated. Say, C(i,j) is such a cell then W(i,j,k,l) is updated by the formula:

$$W(t+1)=\begin{cases} W(t)\ +c/L \text{ if connection (i,j;k,l) is activated in the path} \\ W(t) - c/Ln, \text{ connectiion (i,j;k,l) is not activated,but C(k,l) is not on a wall} \\ \text{no change} \quad \text{if C(k,l) is on a wall.} \end{cases} \qquad (3)$$

where W(t) is the weight of the connection (i,j;k,l) at trial t, L is the length of the path, n is the number of neighbors of C(i,j) that are not on a wall. In the formula, c is constant about Lmax/100 where Lmax is the length of the longest path in the chosen maze pattern. Notice that after the update of these weights, still the summation of the weights of the outgoing connections is 1. In general, updating weights means storing information, that is correlated with the path in the last trial, in the long term memory as a result of learning.

5. EXPERIMENTAL RESULTS

A cellular neural network of size 17x25 as shown in Figure 3 is used for the solution of the maze given in Figure 1.b. The network is tested for 30 different random seeds.

For each seed the following procedure is applied:
1. Clamp the outputs of the neurons corresponding to the walls of the maze to 0,
2. Set initial weights as explained in section 2,
3. Find a random path as explained in section 3,
4. Update the weights as explained in section 4,
5. Goto step 3 for the next trial (repeated for 300 times in the experiments)

Figure 4. shows the results for some different seed values in which the time elapsed in finding a path, the length of these paths and the time/length ratio for the consecutive trials are demonstrated. Figure 5. is the histogram showing the distribution of the paths that are learned for 30 different seeds.

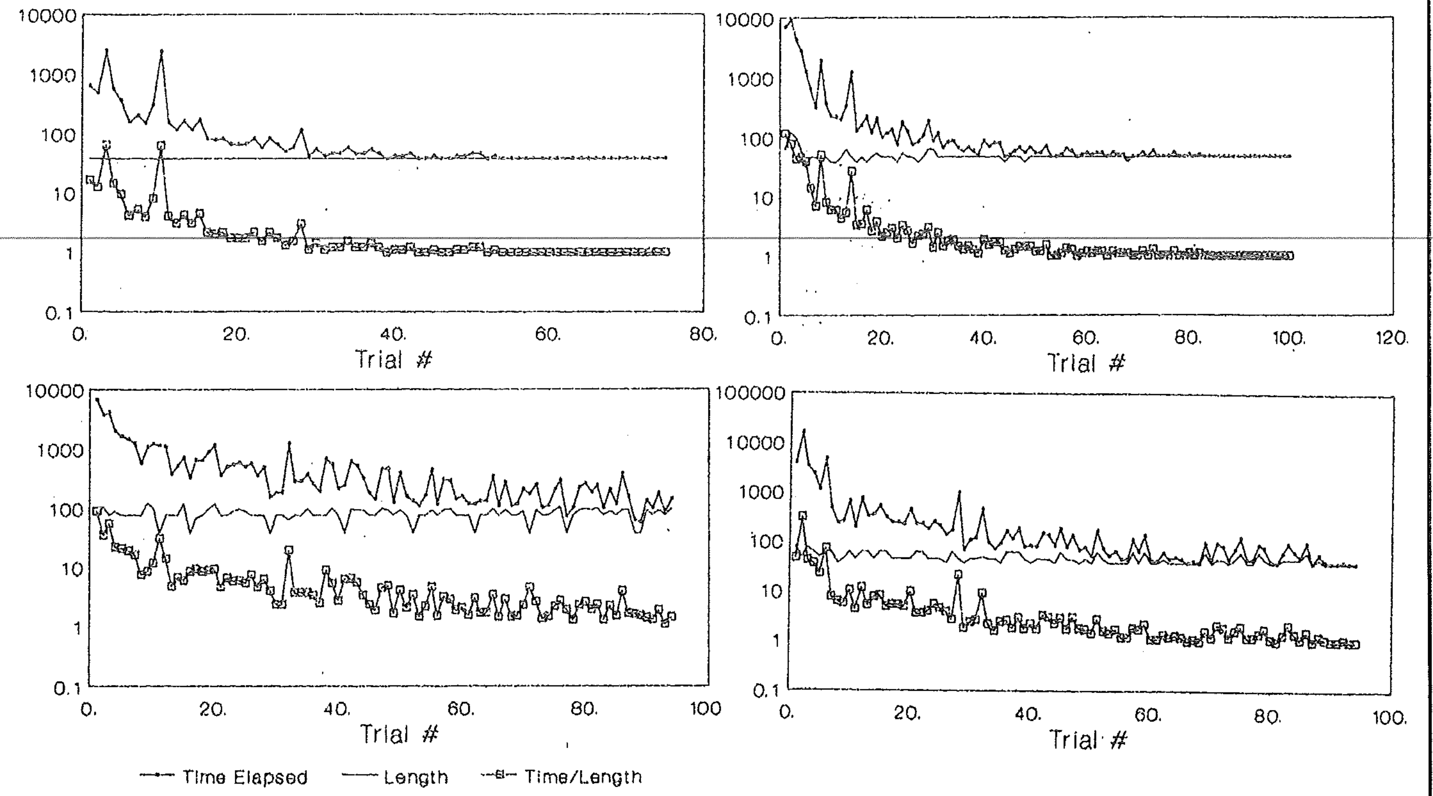


Figure 4. Experimental results for some different seeds a) seed 0 b) seed1 c) seed 7 d) seed 15

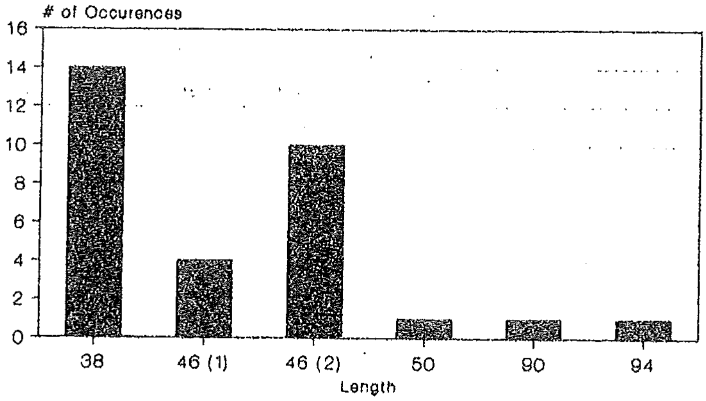


Figure 5. Distribution of paths that are learned for 30 different seeds.

Through these experiments it has been observed that:
1. Initially it takes a long time to find a path. However as the system learns in each trial, the time/length ratio converges to 1 indicating that the path is directly followed without going back and forward unneccessarily.
2. The learning strategy favors the shorter paths, but does not guarantee that it will be the shortest one.

6. CONCLUSION

The problem we put forward is to find a correct path which connects an initial cell to a desired final one in a cellular neural network representing a maze. A probabilistic algorithm based on cellular neural networks is proposed to find a path as the solution of the problem and a strategy to learn these paths is also given. In such a schema it is not guaranteed to learn the shortest path, but it favors the shorter paths. The experimental results are quite reasonable indicating the existence of such a learning.

As a feature work, we are going to search the effect of forgetting some prefix of a path when it is to be learned and also the effect of partially remembering the previous paths on the performance of learning the shortest path. Furthermore, a genetic algorithm will be developed aiming to find out the shortest path as a result of the mutations on the paths that generated previously.

REFERENCES

1. Labirent, *AnaBrittanica* ,(Anayayıncılık A.Ş. and Encyclopedia Brittanica Inc., 1989).
2. Chua L., Yang L., Cellular Neural Networks: Theory, *IEEE Trans. on Circ. Sys*,35(10), October 1988.
3. Gardner M., Mazes, *More Mathematical Diversions*, (Pelican Books, 1961, pp 88-93)
4. Savran M.E. and Morgül Ö., On the Design of Hopfield and Cellular Neural Networks, in *Proceedings of ISCIS VI*, November 1991, pp 899-905
5. Thorndike E.L., Barnhart C.L, *High School Dictionary*, (Doubleday & Company Inc., Newyork, 1941)

Ping-Pong oynamasını öğrenen yapay sinir ağı

Selami Aratma ve H. Levent Akın

Bilgisayar Mühendisliği Bölümü, Boğaziçi Üniversitesi, 80815 Bebek-İstanbul

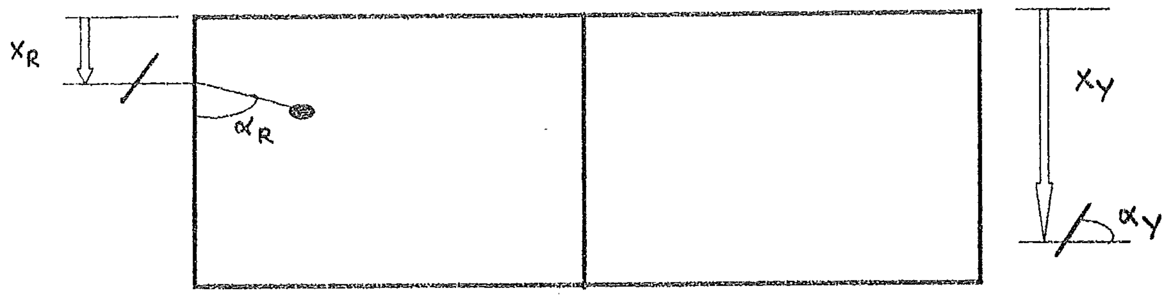
Özet

Bu çalışmada çok katmanlı bir perseptron yapısındaki yapay bir sinir ağına ping pong oynaması öğretilmiştir. Yapay sinir ağına girdi olarak rakibin topu attığı açı ve konumu beslenmekte, buna karşılık kendi raketi için bir konum ve açı belirlemesi istenmektedir. Değişik eğitim kümeleri ve ağ yapıları denenmiş ve başarıları karşılaştırılmıştır.

1.GİRİŞ

Bugüne kadar yapay sinir ağları bir çok öğrenme problemine başarı ile uygulanmıştır [1,2]. Bu çalışmada ise çok katmanlı perseptron yapısındaki bir yapay sinir ağına ping pong oyununu oynamayı öğretmek ve performansını incelemek amaçlanmıştır.

2. MODEL



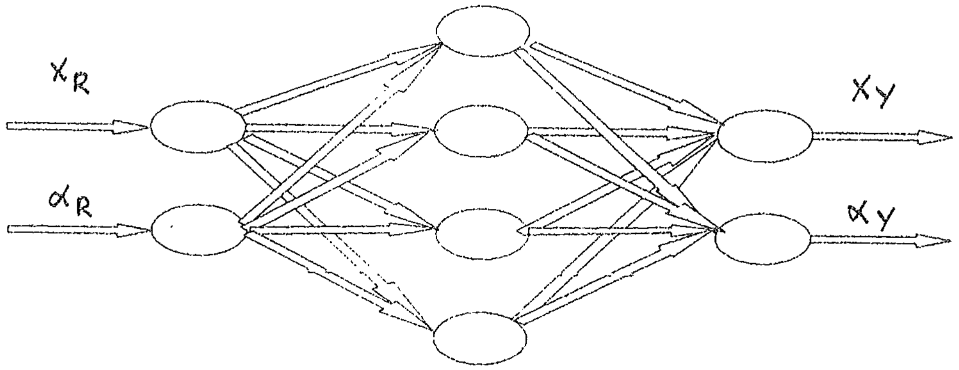
Şekil 1. Model.

Modelimizde, bir yapay sinir ağı (YSA) otomatik bir rakibe (R) karşı ping-pong oynamayı öğrenmekte ve oynamaktadır. R, saha içine düşecek topları, yani geçerli atışları, hatasız karşılamakta ve topu geri gönderirken her zaman geçerli bir atış yapmaktadır. R, hedef olarak sahanın iki yakası arasındaki noktalardan herhangi birini eşit olasılıkla seçmektedir. Oyunun iki boyutta oynandığı bu yüzden topun hiçbir zaman fileye takılmadığı, rüzgar gibi bozucu dış etkenlerin olmadığı, her iki oyuncunun da topa her zaman aynı kuvvetle vurduğu ve yansıma kanunlarının geçerli olduğu kabul edilmiştir.

YSA'ya girdi olarak R'nin topu attığı açı ve konumu beslenmekte, buna karşılık kendi raketi için bir konum ve açı belirlemesi istenmektedir. Yani, YSA'ndan iki tür davranışı öğrenmesi beklenmektedir. Birincisi, R'nin attığı topu karşılamaktır. (Karşılayamadığı durumda K tipi hata ortaya çıkmaktadır.) İkincisi ise, yalnızca savunmada kalmaması amacıyla karşılanan topu R'nin bulunduğu yerden en uzak noktaya düşürmeyi sağlamak ve aynı zamanda da R'yi şaşırtmak amacıyla tepe noktası masanın uzak ucunda olmak üzere bir normal dağılımdan açı seçmektir. (Eğer, YSA topu karşıladığı halde geri atarken saha içine düşüremezse A tipi hata ortaya çıkmaktadır.)

3. AĞ YAPISI

YSA, çok katmanlı perseptron topolojisindedir [1]. Ağ, girdi katmanı, saklı katman ve çıktı katmanından oluşmaktadır. Her katmandaki birimler yalnızca bir sonraki katmandaki birimlerle bağlantılıdır (bkz. Şek.2) Girdi katmanında iki girdi birimi bulunmaktadır. Bunlar sırasıyla R'nin topu atış açısı αR ve R'nin vuruş konumu x_R'ye karşılık gelmektedir. Aynı şekilde, çıktı katmanında da iki birim bulunmakta ve bunların çıktı değerleri YSA'nın topu yanıtlayacağı konum x_Y ve raketin yanıtlama açısı α_Y cinsinden tepkisini belirlemektedir. Saklı katmandaki birim sayısı ise en düşük A ve K tipi hata verecek şekilde deneme yanılma yöntemiyle belirlenmektedir.



Şekil 2. Yapay Sinir Ağı (Eşik birimleri gösterilmemiştir).

4. EĞİTİM YÖNTEMİ

YSA'nın eğitimi için eğitmenli bir yöntem olan *hata geri yayma yöntemi* kullanılmıştır [1]. Eğitimde, doğru girdi-çıktı çiftlerinden oluşan bir eğitim kümesinden yararlanılır. Bu yöntem, her bir veri takımı için tekrarlanan iki evreden yani *ileri geçiş* ve *geri geçiş* evrelerinden oluşmaktadır. İleri geçiş evresinde girdiler girdi birimlerinden YSA ya beslenir ve ağ içinden çıktı birimlerinin çıktılarını hesaplamak amacıyla ileriye yayılır.

Saklı katmandaki birimlere girdi

$$g_j=\sum_{i=0}^{2} I_i W_{ij} \qquad (1)$$

şeklinde hesaplanır. Burada, I_i girdi katmanındaki i. birime girdi, W_{ij} i. girdi birimiyle j. saklı katman birimi arasındaki bağlantının ağırlığıdır. 0 numaralı birim eşik birimi olup çıktısı her zaman +1 değerindedir. Saklı katmandaki birimlerin

$$h_j=f(g_j) \qquad (2)$$

çıktısı

gibidir ve bunlar çıktı birimlerine

$$g_k=\sum_{j=0}^{n_s} h_j W_{jk} \qquad (3)$$

şeklinde girdileri oluşturur. Bu denklemde, n_s saklı katmandaki birim sayısı, W_{jk} ise j. saklı katman birimi ile k. çıktı birimi arasındaki bağlantıların ağırlıklarıdır.

Çıktı katmanındaki birimlerin çıktıları ise

$$o_k=f(g_k) \qquad (4)$$

denklemiyle verilir. Bu çalışmada, (2) ve (4) numaralı denklemlerdeki etkilenim fonksiyonu olarak

$$f(x)=\frac{1}{1+e^{-x}} \qquad (5)$$

şeklindeki sigmoid fonksiyonu kullanılmıştır. Saklı katmandaki birimlerin etkilenimleri çıktı katmanına beslenirken çıktı katmanındaki birimlerin etkilenimleri YSA'nın çıktılarını oluştururlar. Bu çıktılar istenen çıktı değerleriyle karşılaştırılır ve

$$E=\frac{1}{2}\sum_{k=1}^{2} (d_k-o_k)^2 \qquad (6)$$

şeklinde bir hata terimi hesaplanır. Bu denklemdeki d_k istenen çıtı değeridir. Bundan sonraki evrede önce saklı katmanla çıktı katmanı arasındaki bağlantı ağırlıkları, sonra da girdi katmanıyla saklı katman arasındaki bağlantı ağırlıkları

$$\Delta W_{ij}=-\eta\frac{\partial E}{\partial W_{ij}} \qquad (7)$$

şeklinde *en dik iniş yöntemiyle* değiştirilerek hatanın en aza indirilmesine çalışılır. Eğer η küçükse iniş çok yavaş olabilir, η'nin büyük olduğu durumlarda da yüksek genlikli salınımlar gözlenebilir. Bu soruna getirilen çözümlerden biri, (7) numaralı denkleme

$$\alpha\,(W_{ij}^{t}-W_{ij}^{t-1}) \tag{8}$$

şeklinde bir momentum terimi eklemektir. Bu denklemde α, momentum katsayısı, W_{ij}^{t} t. adımdaki bağlantı ağırlığı, W_{ij}^{t-1} ise (t-1). adımdaki bağlantı ağırlığıdır. Buradaki düşünce, değişime bir eylemsizlik katarak aşırı salınımlar yapması yerine ortalama bir azalış karakterine sahip olmasını sağlamaktır.

İki evreli bu işlem, eğitim kümesindeki bütün elemanlar için tekrarlanır. Eğitim kümesindeki bütün elemanların YSA'ya bir defa gösterilmesine bir *devir* denir. Hata belli bir değere düştükten sonra eğitim işlemine son verilir ve eğitim kümesinde bulunmayan elemanlardan oluşan bir deneme kümesiyle YSA'nın genelleme yeteneği ölçülür. Eğer yeterli görülmezse saklı katmandaki birim sayısını arttırmak, daha başka eğitim kümesi kullanmak gibi yöntemlere başvurularak daha iyi bir sonuç alınmasına çalışılır.

5. EĞİTİM VE DENEME KÜMESİNİN OLUŞTURULMASI

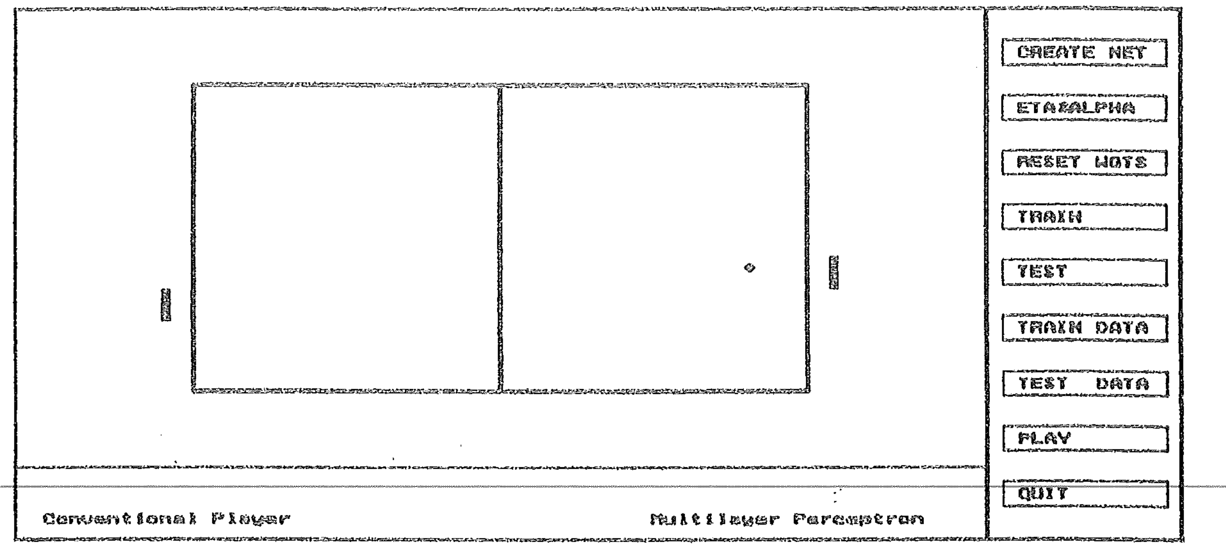
Herhangi bir YSA'nın eğitimindeki en önemli etmenlerden biri de eğitim kümesidir. Eğitim kümesindeki elemanların sayısı yanında temsil etme nitelikleri de YSA'nın genelleme yeteneğini etkiler. Bu çalışmada incelenen etmenlerden birisi de bu olmuştur. Olası konum ve açı değerleri kullanılan etkilenim fonksiyonunun özelliği nedeniyle [0.05,0.95] aralığındaki değerlere çevrilmiş ve bu aralık değişik sayıda eşit parçaya bölünmüş ve değişik kompozisyonlara sahip eğitim kümeleri elde edilmiştir.

Bu çalışmada sabit bir deneme kümesi kullanmak yerine gerçek oyun koşullarını daha iyi temsil etmek düşüncesiyle, YSA her on devirde bir 3000 kere oynatılmış ve **K** ve **A** tipi hata oranları hesaplanmıştır. Bu oyunlar sırasında **R**, atış hedefini rasgele seçmektedir.

6. SONUÇLAR VE TARTIŞMA

YSA'nın ping pong oynamak üzere eğitilmesini kolaylaştırmak ve çeşitli etmenlerin etkilerini incelemek amacıyla PC tipi bilgisayarlarda çalışabilen etkileşimli bir yazılım gerçekleştirilmiş ve bütün çalışmalar bu benzetim yazılımı kullanılarak yapılmıştır. Yazılım, Turbo Pascal 6.0 ortamında geliştirilmitir. Kullanıcı arayüzü Şekil 3'de görüldüğü gibidir. Bütün benzetimler, 25 MHz hıza sahip 80486 işlemcili bir bilgisayarda yapılmış ve bu benzetimlerde η=0.2, α=0.7 değerleri kullanılmıştır.

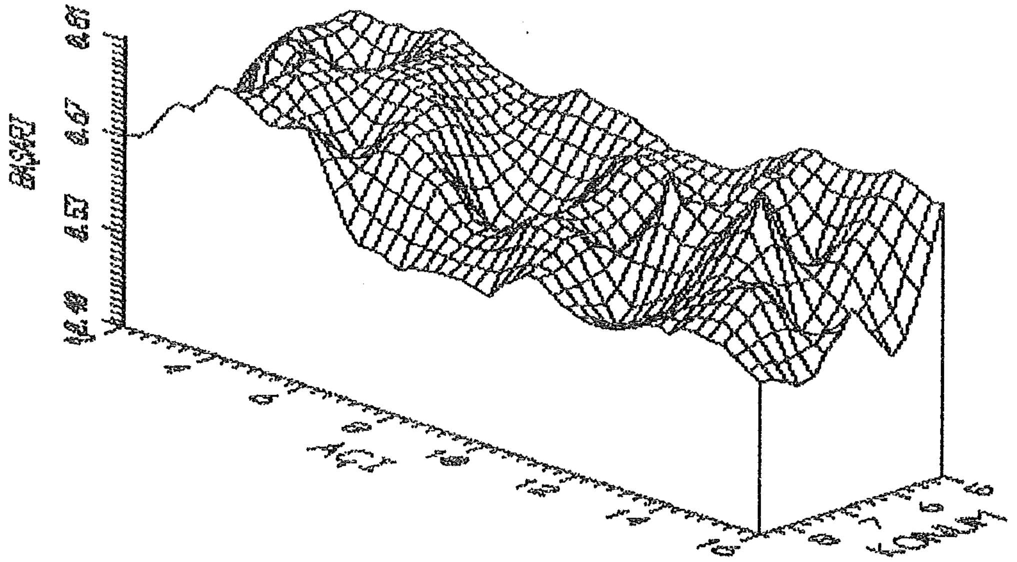
YSA'nın eğitimi sırasında eğitim kümesinin büyüklüğü ve saklı katmandaki birim sayısının **K** ve **A** tipi hatalar üstündeki etkisi incelenmiştir. Yapılan benzetimler sonunda **A** tipi hatanın bir kaç devir sonra sıfıra indiği ancak **K** tipi hatanın % 20 nin altına düşmediği gözlenmiştir. Şekil 4. den de görüldüğü gibi **K**



Şekil 3. Benzetim yazılımı kullanıcı arayüzü.

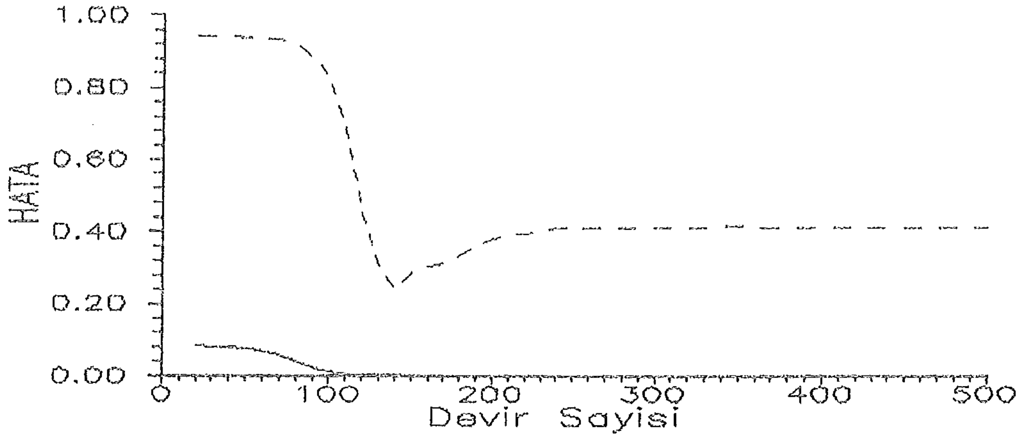
tipi hatanın toplam atış sayısına oranı olarak tanımlanan başarı oranı üstünde yalnızca eğitim kümesindeki veri takımı sayısı değil bu veri takımlarının açı ve konum uzayını tarama sıklığının da önemli olduğu anlaşılmaktadır.

Şekil 5 ise daha ilginç bir sonucu göstermektedir. Eğitim kümesinden bağımsız olarak devir sayısı arttıkça K tipi hata sürekli azalmamakta aksine belli bir değerden sonra artmakta ve bir başka değerde sabit kalmaktadır. Bu da, buradan sonra *aşırı eğitim* yapıldığını [2] göstermektedir. Bu yüzden eğitim işlemine K tipi hatanın en alt düzeyine erişildiği noktadan sonra devam etmek gereksiz hatta zararlı olduğu sonucuna varılabilir.

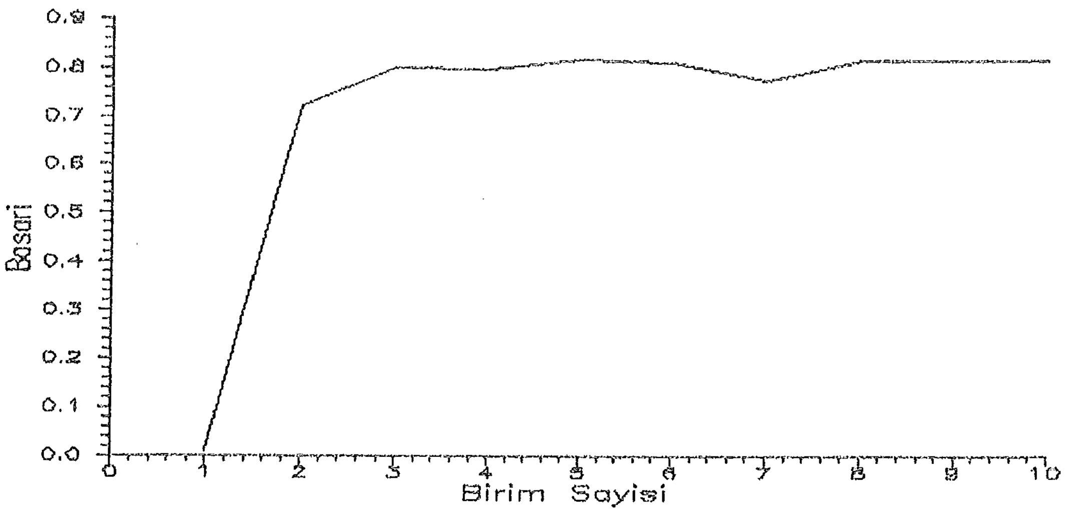


Şekil 4. Başarı oranının eğitim kümesi kompozisyonuna göre değişimi.

Saklı katmandaki birim sayısının ise başarı oranı üstünde fazla etkili olmadığı da Şekil 6 dan görülmektedir.



Şekil 5. K tipi hatanın ve E nin devir sayısıyla değişimi. (Burada kesikli çizgi K tipi hataya, sürekli çizgi ise E'ye aittir.)



Şekil 6. Başarı oranının saklı katmandaki birim sayısına göre değişimi.

Bundan sonraki çalışmalarda üçüncü boyutun eklenmesi, bozucu dış etkenlerin göz önüne alınması ve topa farklı hızlarla vurulmasının etkilerinin incelenmesi düşünülmektedir.

7. KAYNAKLAR

1. Hertz, J., Krogh, A. ve Palmer, R. G., *Introduction to the Theory of Neural Computation* (Addison-Wesley, Redwood City, CA, 1991).
2. Hecht-Nielsen, R., *Neurocomputing* (Addison-Wesley, Reading, MA, 1990).

Doğal Diller – Bilgi Gösterimi

Natural Languages – Knowledge Representation

Towards a formal semantics of Turkish

Erkan Tın and Varol Akman

Department of Computer Engineering and Information Science,
Bilkent University, Bilkent, Ankara 06533 Türkiye

Abstract

Jon Barwise and John Perry were instrumental in formalizing the commonsense notions that language is an integral part of our everyday experience and consists of situated activities such as talking, reading, writing, and listening. Primary function of these activities is to convey information. Depending on the time and place of its utterance, and its speaker, a statement can convey different information and hence can have different meanings. Various versions of the theory of meaning and reference set within Barwise and Perry's *situation semantics* have been applied to a number of linguistic issues (mainly) in English. In this paper, a semantic account based on situation semantics is proposed to establish a framework for understanding sentences in Turkish. The full version of this paper has been submitted to a journal for publication.

1. INTRODUCTION

Language is an integral part of our everyday experience. Some activities related to language include talking, reading, writing, and listening. These activities are *situated*; they occur in situations and they are about situations [2]. What is common to these situated activities is that they convey information [10, 12, 13]. When uttered at different times by different speakers, a statement can convey different information to the listener and hence can have different meanings. Consider the sentence "That really attracts me." Depending on the reference of the demonstrative *that*, its interpretation and hence meaning would change. For example, it would be uttered by a boy referring to an icecream cone or by a cab driver referring to fast driving, meaning absolutely different things [18].

To give a viable analysis of meaning in Turkish, we follow the theory of meaning and reference set within the framework of *situation semantics*. Situation semantics is a rather recent approach to language and information, first formulated by Jon Barwise and John Perry [6]. It provides a fundamental framework for a realistic model-theoretic semantics of natural language [5]. Various versions of this theory have been applied to a number of linguistic issues (mainly) in English [1, 3, 4, 9, 16]. The ideas emerging from research in situation semantics have been also joined with well-developed linguistic theories such as *lexical-functional grammar* and these works led to rigorous formalisms [17]. On the other hand, situation semantics have been compared to some other mathematical approaches to the theory of meaning such as the Montague Grammar [8, 11].

In this paper we propose a semantic account based on situation theory to establish a framework for understanding sentences in Turkish [22]. We focus, as a preliminary step, on utterances in which speaker uses simple sentences to convey information to listener.

What we mean by *understanding a sentence* is to identify the constraints placed on its utterance and to fill the roles of individuals (a point that will be explained in Section 2) by clarifying issues such as co-reference of phrases [19]. We consider simple SOV (Subject-Object-Verb) sentences formed from names, singular pronouns, and verb phrases. For such sentences, we are developing an artificial language and defining semantics to identify the meaning of expressions in this language.

2. SITUATION THEORY AND SITUATION SEMANTICS

Situation theory is an attempt to develop a mathematical field for meaning, which will clarify and resolve the problems in the study of language, information theory, intelligence, and the mind. It is a mathematical theory of meaning, but requires little background in mathematics. According to the theory, individuals, properties, relations, and spatio-temporal locations are the basic constructs. The world is viewed as a collection of objects, sets of objects, properties, and relations.

Individuals are conceived as invariants. Having properties and standing in relations, they persist in time and space. *Objects* are the parts of individuals. Words are also objects, i.e., invariants across utterances. All individuals, including spatio-temporal locations, have *properties* (like being *fragile* or *red*). They stand in *relations* to one another (like being *prior* to, being *in*, *on*, or *under*).

A sequence such as $< r, x_1, \ldots, x_n >$ where r is an n-ary relation over the individuals $x_1, \ldots, x_n$ is called a *constituent sequence*. Suppose an individual, named Bilge, was eating icecream yesterday at home and she is also eating icecream now at home. Both of these situations share the same constituent $< eats, Bilge, icecream >$. These two events occurring at the same location, but at different times, have the same *situation type* s (cf. [2] for the origin of this idea). Situation types are partial functions from relations and objects to the values 0 and 1.[1] The situation type s, in our example, assigns 1 to the constituent sequence $< eats, Bilge, icecream >$:

In s: eats, Bilge, icecream; 1.

Situation types are independent of locations. Given a location and a situation type, they form a *state of affairs* which in fact is a static situation. That is, it stands through a small duration in time. In order to obtain a structure which keeps track of the changes, *courses of events* are used. A course of events is a partial function from locations to situation types. A course of events may contain information about events at more than one location. A course of events e that Bilge is eating icecream at location l_1 (say 11:00 a.m., at home) and she is sleeping at a temporally succeeding location l_2 (say 12:15 p.m., at home) is represented as follows:

In e, at l_1: eats, Bilge, icecream; 1,
l_2: sleeps, Bilge; 1,
$l_1 < l_2$.

We allow spatio-temporal locations to stand in relation with each other in three different ways: l_1 temporally precedes l_2 ($l_1 < l_2$), l_1 temporally overlaps l_2 ($l_1 \circ l_2$), and l_1 spatially overlaps l_2 (l_1 @ l_2). Clearly, in the last example we also have l_1 @ l_2.

[1] They can only take 0 or 1 as a value, not a third truth value.

All of these abstract notions are used to classify real situations and to study the meaning of expressions. According to situation theory, meanings of expressions reside in systematic relations between different types of situations. They can be identified with relations on *discourse situations* d, *(speaker) connections* c, the expression φ itself, and the described situation e. Some public facts about the utterance (such as who its speaker is and when it is uttered) are determined by the discourse situations [20]. The ties of the mental states of both the speaker and the listener with the actual world constitute the connections [18].

A discourse situation, as mentioned before, involves the expression uttered, its speaker, the spatio-temporal location of the utterance (i.e., when and where it is uttered), and the individual to whom it is uttered. Each of these aspects defines a linguistic role: the role of the speaker, the role of the addressee, etc. Then, we have a type of *discourse event* D in which these roles are left unfilled. For example, if the indeterminates $\dot{a}$, $\dot{b}$, $\dot{\alpha}$, and $\dot{l}$ denote the speaker, the addressee, the expression, and the location of the utterance, respectively, then the discourse event D is given as:

$$\begin{aligned} D := \text{ at } \dot{l}:\ & \text{speaking}, \dot{a};\ 1, \\ & \text{addressing}, \dot{a}, \dot{b};\ 1, \\ & \text{saying}, \dot{a}, \dot{\alpha};\ 1. \end{aligned}$$

Using a name or pronoun, the speaker refers to an individual. A name directly refers to an individual, independent of whether the individual is imaginary or real. A pronoun can either refer to an individual deictically or else it may be used indirectly by co-referring with a noun phrase. For the uses of names and deictic uses of pronouns, an individual must fill the role of referent in a discourse event. A situation s in which the referring role is uniquely filled is called a *referring situation.* If in s the speaker uses a noun phrase α to refer to a unique individual, this individual is called the *referent* of α (cf. [14]).[2]

Tense markers of the tensed verb phrases can also refer to individuals; spatio-temporal locations. The past tense marker `-DI` in Turkish, for example, is used to refer to some location l which temporally precedes the location l_d of the utterance. Therefore, s can be seen as a partial function from the referring words α_i to their referents $s(\alpha_i)$. This function is the speaker's *connections* for a particular utterance [20].

Utterance of an expression φ constrains the world in a certain way, depending on how the *roles* for the discourse situations, connections, and the described situation are filled. For example, "I am crying" describes a three-place relation $[\![$I am crying$]\!]$ on discourse situations, connections, and a described situation. Given a discourse situation d, connections c, and a course of events e, this relation holds just in case there is a location l_d and speaker a_d such that a_d is speaking at l_d, and in e, a_d is crying at l_d. In this case, this relation is denoted by $d, c[\![$I am crying$]\!]e$. This approach towards identifying linguistic meaning is essentially what Barwise and Perry call the *Relational Theory of Meaning* [6, 7].

Utterance of a simple sentence φ describes a situation. Utterance of `EROL KOŞUYOR` "Erol is running," for example, describes a situation e, at a spatio-temporal location l overlapping the spatio-temporal location of the utterance l_d, in which the individual named Erol is running:

[2] Obviously, the speaker may not refer to anything at all. In this case, the role of the referent is left unfilled.

$e :=$ at l: koşuyor, Erol; 1 where $l(=c(\text{-YOR})) \circ l_d$, Erol$=c(\text{EROL})$.

The connection c assigns a location l to the progressive tense marker -YOR. This is denoted by $l(=c(\text{-YOR}))$. It also assigns an individual, named Erol, to the word EROL.[3]

The meaning of an utterance of φ and hence its interpretation are effected by many factors such as stress, modality, and intonation. However, the situation in which φ is uttered and the situation e described by this utterance seem to play the most influential roles. For this reason, in the situation-theoretic account, the meaning of φ is taken to be a relation defined over the expression φ, the discourse situations d, the speaker connections c, and the described situation e.

The constituent expressions of the sentence φ do not describe a situation when uttered in isolation. Uttering a verb phrase in isolation, for example, does not describe a situation e. The other parts of the utterance (of which this verb phrase is a part) must systematically contribute to the description of e by providing situational elements such as an individual or a location. For example, the situational elements for the utterance of the tenseless verb phrase KOŞ "run" provide a spatio-temporal location for the act of running and the individual who is to run. For the utterance of the tensed verb phrase KOŞUYOR, an individual must be provided. The situational elements form the *setting* σ for an utterance.[4] The meaning of φ is a relation not only defined on discourse situations d, connections c, and described situation e, but also on the setting σ.

The meanings of the constituent expressions of an utterance must be defined to obtain the meaning of the whole. This is what we shall do in the sequel.

3. MEANING of SINGULAR PRONOUNS

The pronoun BEN "I" has a straightforward meaning. When a speaker a_d utters it in discourse situation d, it stands for that speaker. Hence, its meaning is dependent on the discourse situation:

$d, c[\![\text{BEN}]\!]a_\sigma, e$ iff $a_\sigma = a_d$.

SEN is a singular pronoun in Turkish corresponding to the singular pronoun "you." Its meaning is similar to that of BEN except that it designates the addressee b_d in the discourse situation d, instead of the speaker:

$d, c[\![\text{SEN}]\!]a_\sigma, e$ iff $a_\sigma = b_d$.

When uttered, its speaker and the location of utterance can be easily determined through discourse situation. The addressee may not be so obvious though. This is especially the case when there are more than one candidate to be the addressee. The listener can pick up the addressee by noticing, for example, whom the speaker is looking at or whom he is pointing towards.

[3] Following Barwise and Perry [6], we consider the meaning of a name β as a relation between utterances of β and individuals it is used to refer to. For simplicity, we assume that a name occurs only once in an utterance.

$d, c[\![\beta]\!]a_\sigma, e$ iff $c(\beta) = a_\sigma$.

[4] The elements provided by a setting σ can be any individual including spatio-temporal locations. Throughout this paper, we will use a_σ and b_σ for the roles to be filled by the individuals provided by the setting σ of an utterance, and l_σ for the utterance location (if not implied by the utterance itself). a_σ and b_σ will especially be used for the subject and object filling roles, respectively.

Turkish has no grammatical gender. There is no distinction between "he," "she," and "it." The personal pronoun O can be used for all of them. O is used to refer to things with which the speaker is connected. It can pick up its referent by going directly through the speaker or by having some noun phrase as an antecedent:

(1) KEDİ-NİN TIRMALA-DIĞI KÖPEK O-NU ISIR-DI.
Cat-GEN scratch-NOM dog he/she/it-ACC bite-PAST-3SG[5]
"The dog that the cat scratched bit him/her/it."

The two uses of O are illustrated in (1). In one case, the speaker uses ONU to refer to someone mentioned elsewhere. This individual may be say, a child named Erol, playing in a nearby garden. In this case, the utterance of (1) has connection c which binds the pronoun to this individual, i.e., c(ONU)=Erol.

In another case, the speaker uses ONU to refer to the individual described by KEDİ "the cat." Suppose that Erol is telling a story about a cat he has seen in the garden. When he utters (1), Erol does not directly connect ONU to the cat but his use of this pronoun picks its referent up by having the noun phrase KEDİ as antecedent.

The two uses of the pronoun O will be presented by subscripted forms. (1.1) and (1.2) represent the respective uses of ONU in (1):

(1.1) KEDİNİN$_1$ TIRMALADIĞI KÖPEK ONU$_0$ ISIRDI.
(1.2) KEDİNİN$_1$ TIRMALADIĞI KÖPEK ONU$_1$ ISIRDI.

The subscripts indicate the different uses of the pronoun, not the existence of new pronouns ONU$_0$ and ONU$_1$, having two different meanings. Independent of whether a pronoun is inflected or not, we will talk about its meaning as it is used in the sentence. For example, the meaning of ONU$_0$ in (1.1) is defined as a relation on the discourse situations, connections, a situation, and an individual provided by the settings:

$d, c[\![\text{ONU}_0]\!]a_\sigma, e$ iff $c(\text{ONU}_0)=a_\sigma$.

4. THE MEANING OF VERB PHRASES

The predicate of a sentence in Turkish can be a verb or a construction with verb as its principal member. That is, it can be a verb with its object and various adverbial elements. Such a construction is called a *verb phrase*. A minimal verb phrase basically consists of three elements: a *verb stem* which can be a single unit or built up from several elements as in GEL "come" or DÖV, a tense marker added after the stem, and a personal ending following this tense marker [22, 23]. The personal endings are attached to a verbal predicate in the same way as they are added to a nonverbal predicate:

(2) [EROL-U DÖV-ÜYOR][6]
Erol-ACC beat-PROG
"am/is/are beating Erol"

(3) EROL-U DÖV-ÜYOR-UM.
Erol-ACC beat-PROG-1SG
"I am beating Erol."

[5] The abbreviations used for the English glosses are borrowed from [23].

[6] If a phrase is extracted from a larger one, it will be bracketed to show that the verb of the phrase can have some suffixes added in a larger utterance.

This is illustrated in (3) by adding the first singular personal ending to the verb phrase in (2). Consider the meaning of the verb phrase [EROL'U DÖVÜYOR] "am/is/are (SG) beating Erol" with the present progressive tense marker -YOR. It describes a uniformity across the meaning of sentences of which it is a part:

(4) AHMET EROL-U DÖV-ÜYOR.
Ahmet Erol-ACC beat-PROG-3SG
"Ahmet is beating Erol."

(5) EROL-U DÖV-ÜYOR-SUN.
Erol-ACC beat-PROG-2SG
"You are beating Erol."

An utterance of any of these sentences describes a situation. What is common to all such utterances is that the speaker is describing a situation in which an individual is beating Erol. Then, the meaning of the verb phrases can be seen as a relation between discourse situations, connections, courses of events, and individuals provided by the other parts of the sentence, that is, the settings:

$d, c[\![$[EROL'U DÖVÜYOR]$]\!]a_\sigma, e$ iff in e: at l: dövüyor, a_σ, Erol; 1

where Erol=c(EROL) and l(=c(-YOR)) $\circ$ l_d.

The tensed verb phrases can be used as they are to form a complete sentence. The third person suffix is not added to the verb as in (4). In such cases, the meaning of the verb phrase is defined similarly. Meanings of the negated verb phrases such as (6) can also be defined as is done for the positive ones:

(6) [EROL-U DÖV-MÜ-YOR-DU]
Erol-ACC beat-NEG-PROG-PAST
"was/were not beating Erol"

The meaning of (6) can be defined as a relation again:

$d, c[\![$[EROL'U DÖVMÜYORDU]$]\!]a_\sigma, e$ iff in e: at l: dövüyor, a_σ, Erol; 0

where Erol=c(EROL) and l(=c(-YORDU)) $< l_d$.

The use of a subject pronoun in Turkish is optional unless the subject has an emphatic or contrastive function. If this is the case, the pronominal form of the pronoun becomes obligatory:

(8) BEN DÖV-ÜYOR-UM.
I beat-PROG-1SG
"I am beating."

(9) O DÖV-ÜYOR.
He/she/it beat-PROG-3SG
"He/she/it is beating."

The meanings of the sentences above do not change when the subject pronouns are omitted since the verbs already have person suffixes:

(10) Ø DÖV-ÜYOR-UM.
beat-PROG-1SG
"I am beating."

(11) Ø DÖV-ÜYOR.
beat-PROG-3SG

"He/she/it is beating."

Whether the pronoun exists or not, for (11) the individual who beats is provided by the settings of its utterance. For (10), the individual is directly determined as the speaker a_d provided by the discourse situations:

$d, c[\![\texttt{DÖVÜYORUM}]\!]b_\sigma, e$ iff in e: at l: dövüyor, a_d, b_σ; 1 where $l(=c(\texttt{-YOR})) \circ l_d$.

4.1. Tenseless Verb Phrases

We consider a verb phrase with no tense marker as a *tenseless verb phrase*. A tenseless verb phrase then consists of a verb stem and an object (if the verb stem is transitive) such as [EROL'U DÖV] "beat Erol" or [EROL'DAN KORK] "be afraid of Erol." Note that in all verb phrases the object receives a case suffix required by the verb stem:

(12) [EROL-U DÖV]
Erol-ACC beat
"beat Erol"

The tenseless verb phrases define uniformities across all the tensed verb phrases. The setting σ in which the tenseless verb is used provides not only an individual a_σ but also a location l_σ and a truth value tv_σ. The meaning of the tenseless verb phrase in (12) is then defined as:

$d, c[\![\texttt{[EROL'U DÖV]}]\!]\sigma, e$ iff in e: at l_σ: dövüyor, a_σ, Erol; tv_σ

where Erol=c(EROL).

Even though the verb is transitive the object may be omitted. In such cases, the settings provide an individual b_σ, standing for the object position, as well as an individual a_σ:

$d, c[\![\texttt{[DÖV]}]\!]\sigma, e$ iff in e: at l_σ: dövüyor, a_σ, b_σ; tv_σ.

4.2. Sentence Connectives and Antecedent Relations

In Turkish, VE "and" and VEYA "or" are the two basic sentence connectives. A statement made by a conjunctive sentence usually describes situations described by both of its conjuncts. A statement made by a disjunctive statement usually describes those situations described by only one of its disjuncts. A statement made by a conjunctive/disjunctive sentence is not a conjunction of independent statements made by its constituents. Consider the sentence:

(13) EROL BİLGE-YE BAĞIR-DI VE O ONA KIZ-DI.
Erol Bilge-DAT shout-PAST-3SG and he/she/it him/her/it angry-PAST-3SG
"Erol shouted at Bilge and he/she/it got angry with him/her/it."

Assume that the pronouns O and ONA have antecedent relations with Bilge and Erol, respectively:

(14) EROL$_1$ BİLGE$_2$'YE BAĞIRDI VE O$_2$ ONA$_1$ KIZDI.

When we break this sentence into its constituents, we lose the connection provided by the antecedent relationships between pronouns and names. The sentence O ONA KIZDI, when uttered in isolation, reflects different semantic properties. For a meaningful utterance of it, there must be connections which assign individuals to the free pronouns O and ONA. In (14), however, due to the semantic interaction between two constituents, these pronouns directly pick up their antecedents. For this reason, when the meaning relation is defined for sentences formed via connectives, the antecedent relations must be set right. This is possible if there exists a connection which is in the domain of both of the meaning relations of the constituents of the sentence:

$d, c[\![\varphi \texttt{ VE } \psi]\!]e$ iff there is a c' extending c such that d,c' is in the domain of each relation $[\![\varphi]\!]$ and $[\![\psi]\!]$, and both $d, c'[\![\varphi]\!]e$ and $d, c'[\![\psi]\!]e$.

The same considerations apply to the disjunctive sentences:

$d, c[\![\varphi \texttt{ VEYA } \psi]\!]e$ iff there is a c' extending c such that d,c' is in the domain of each relation $[\![\varphi]\!]$ and $[\![\psi]\!]$, and either $d, c'[\![\varphi]\!]e$ or $d, c'[\![\psi]\!]e$.

6. CONCLUSIONS

We have overviewed an interesting semantic account of natural language, viz. situation semantics, and applied it to clarify meanings of simple structures in Turkish. In this direction, we have defined the meanings of the constituent expressions of sentences to obtain the meanings of the whole in the framework of situation theory. To illustrate how the semantics for SOV sentences in Turkish can be formed in situation semantics, we are currently developing an artificial language. The expressions of this language will not be the syntactic entities of Turkish. Rather, they will represent the structures for a small fragment of Turkish and the expressions of this language will reflect the semantic properties of this fragment. We believe that this language will provide a basis for computationally identifying and filling the roles of individuals, and clarifying the co-reference of phrases in Turkish.

References

[1] V. Akman and E. Tın, "What is in a context?" in E. Arıkan, editor, *Proceedings of the 1990 Bilkent International Conference on New Trends in Communication, Control, and Signal Processing, Volume II*, Amsterdam, Holland: Elsevier, 1990, pp. 1670–1676.

[2] J. L. Austin. "Truth," in J. O. Urmson and G. J. Warnock, editors, *Philosophical Papers*, Oxford, U.K.: Oxford University Press, 1961, pp. 117–133.

[3] J. Barwise. "Noun Phrases, Generalized Quantifiers and Anaphora," in E. Engdahl and P. Gärdenfors, editors, *Generalized Quantifiers: Linguistic and Logical Approaches*, Dordrecht, Holland: Reidel, 1986.

[4] J. Barwise. *The Situation in Logic*, CSLI Lecture Notes Number 17, Center for the Study of Language and Information, Stanford, Calif., 1989.

[5] J. Barwise and J. Etchemendy. "Model-Theoretic Semantics," in M. I. Posner, editor, *Foundations of Cognitive Science*, Cambridge, Mass.: MIT Press, 1989, pp. 207–243.

[6] J. Barwise and J. Perry. *Situations and Attitudes*, Cambridge, Mass.: MIT Press, 1983.

[7] J. Barwise and J. Perry. "Semantic Innocence and Uncompromising Situations," in A. P. Martinich, editor, *The Philosophy of Language*, New York: Oxford University Press, 1985, pp. 401–413.

[8] R. Cooper. "Meaning Representation in Montague Grammar and Situation Semantics," in B. G. T. Lowden, editor, *Proceedings of the Alvey Sponsored Workshop on Formal Semantics in Natural Language Processing*, 1987.

[9] R. Cooper, K. Mukai, and J. Perry, editors. *Situation Theory and Its Applications*, CSLI Lecture Notes Number 22, Center for the Study of Language and Information, Stanford, Calif., 1990.

[10] K. Devlin. *Logic and Information*, Cambridge, U.K.: Cambridge University Press, 1991.

[11] D. Dowty, R. Wall, and S. Peters. *Introduction to Montague Semantics*, Dordrecht, Holland: Reidel, 1981.

[12] F. Dretske. *Seeing and Knowing*, Chicago, Illinois: University of Chicago Press, 1969.

[13] F. Dretske. *Knowledge and the Flow of Information*, Cambridge, Mass.: MIT Press, 1981.

[14] M. Enç. "The Semantics of Specificity," *Linguistic Inquiry*, 22(1): 1–25, 1991.

[15] E. Erguvanlı-Taylan. "Pronominal Versus Zero Representation of Anaphora in Turkish," in D. I. Slobin and K. Zimmer, editors, *Studies in Turkish Linguistics*, Amsterdam, Holland: John Benjamins, 1986, pp. 209–231.

[16] D. A. Evans. "A Situation Semantics Approach to the Analysis of Speech Acts," in *Proceedings of the 19th Annual Meeting of the ACL*, Association for Computational Linguistics, 1981.

[17] J. E. Fenstad, P.-K. Halvorsen, T. Langholm, and J. van Benthem. *Situations, Language, and Logic*, Dordrecht, Holland: Reidel, 1987.

[18] H. P. Grice. "Utterer's Meaning, Sentence-Meaning, and Word-Meaning," *Foundations of Language*, 4: 1–18, 1968.

[19] D. Lewis. "General Semantics," in D. Davidson and G. Harman, editors, *Semantics of Natural Language*, Dordrecht, Holland: Reidel, 1972, pp. 169–218.

[20] J. R. Searle. *Speech Acts: An Essay in the Philosophy of Language*, Cambridge, U.K.: Cambridge University Press, 1969.

[21] S. Stucky. *The Situated Processing of Situated Language*, Report No. CSLI–87–80, Center for the Study of Language and Information, Stanford, Calif., 1987.

[22] R. Underhill. *Turkish Grammar*, Cambridge, Mass.: MIT Press, 1976.

[23] R. Underhill. "Turkish," in D. I. Slobin and K. Zimmer, editors, *Studies in Turkish Linguistics*, Amsterdam: John Benjamins, 1986, pp. 7–21.

An Approach to Machine Translation

M. Kutluk Özgüven and Jun'ichi Tsujii
Centre for Computational Linguistics, UMIST, Manchester, UK.

Abstract

In the last five decades, research on Machine Translation of human languages, has produced many translation systems operating on a wide range of languages, text types and subject domains. Despite moderate success, there have been only a few systems to answer the industrial demand. In this paper we discuss criteria that determine the feasibility of Machine Translation, and propose optimum points at which Machine Translation can be better implemented. Structure of a prototype system for Turkish language is also included in discussions, which is based on the results of Turkish-English Newsreport Translation project.

Özet

Doğal yazı dilinin Bilgisayar Çevirisi üzerinde araştırmalar, yaklaşık elli yıldır, dil, metin türü, konu gibi farklılık gösteren ortamlarda çalışan çeviri düzenlerini ortaya koymuştur. Görecel başarılara karşın bunların çok azı gerçek iş ortamlarının gereksinimlerini yanıtlayabilmektedir. Bu bildiride Bilgisayar Çevirisi'nin uygunluğunu belirleyen etmenler üzerinde duracağız, Bilgisayar Çevirisi'ni daha verimli kullanabileceğimiz özel durumlardan bahsedeceğiz. Haber raporlarının çevirisi üzerine yapılan araştırmamızda kullandığımız İngilizce - Türkçe ilkörnek düzenin yapısına da değindik.

1 Introduction

The earliest ideas for automatic translation of natural language by electronic computers were proposed by Warren Weaver (Hutchins 1986). In 1947 Weaver, having witnessed cryptographers breaking a 100-word coded text in Turkish without knowing the content or the language, argued that a foreign language text could be treated as an English text coded with different symbols and could therefore be similarly decoded. Based on Shannon's Information Theory and WW II cryptography techniques, MT research, one of the pioneers of AI, was accelerated after a Georgetown-IBM experiment in 1954 demonstrating full intelligible output, despite the fact that the system was consisted of six grammar rules, 250 Russian-English dictionary entries and a selected set of input sentences. International attention to the experiment led to the formation of independent MT groups both in the West and the East. Major US sponsors were the National Science Foundation, the CIA, and US Military Forces and the source language was Russian for obvious reasons. This stream is known as the **first generation** systems which continued until the mid-60s. Because these systems start the translation process without any intermediate steps their approach is called **direct translation**.

Failure of first generation MT can primarily be explained by the lack of attention in linguistics, which was then admittedly premature for computational requirements. The ALPAC report, released in 1963 by the titular committee set to inspect the large MT research budgets of the government and labelled all MT activities as far from being worthy of funding. Consequently, most projects of the time had to come to an end. The exception is Peter Toma's SYSTRAN which is still in operation (Hutchins et al 1992). This system relies on first generation ideas but is still one of the foremost commercial systems as well, due to the thirty years of expertise and a powerful mixture of human effort with machine output.

1.1 Second Generation

Following the witch-hunt of MT scientists in US, the subject found roots in Europe, where the necessity for routine translation of long documents was not confined to defence and intelligence. European academics, long interested in MT, have independently formed research groups, and being better in multilingualism and worse in code-breaking, adopted a linguistics-based approach. With the help of relatively advanced computer technology and novel linguistic formalisms pioneered by Noam Chomsky's Transformational Grammar, MT architectures were structured into mutually independent system software and linguistic modules so as to enable non-computer scientists to participate in the development of a system. Therefore **second generation** started in the 70s.

Second generation MT systems basically work in three steps. Having been pre-processed down to basic linguistic units, the sentence is analysed and converted to a symbolic representation by a parser which is constructed according to a certain linguistic theory or grammar. Then, this intermediate symbolic representation of input is transferred to a corresponding formalism specific to the target language. In the final phase, the target language representation is input in a more or less deterministic generator for the production of natural language output. Building of the transfer module often follows the results of analysis module construction, i.e. the transfer process is dependent on the success of the linguistic theory that is employed to analyse the source language. Therefore the independent analysis of source language is the bottleneck of design and construction. Other groups which initially make a global emphasis of theoretical linguistics generally converge with the second generation techniques within a few years, e.g. the Rosetta, Phillips - Eindhoven MT project, which originally intended to follow the Montague grammars, isomorphism, and compositionality, had to diverge from its principled path. Similar examples have gathered second generation MT systems around the analysis - transfer - generation architecture with a clear separation of linguistic rules and computational procedures (Somers 1990).

Translation problems in organizations with multinational relations, e.g. Comisson of European Communities, have created multilingual MT systems. EUROTRA, a common European MT project since 1978, is the typical example of a multilingual system with a colossal organizational split over twelve member countries with dozens of computational linguists, who participate in one of the 72 pairs of 9 official EC languages (Copeland at al 1991). Despite the heavy research and funding EUROTRA failed to deliver a fully operational MT system that translates official documents of the Comission, because of unconstrained treatment of language, the high number of translation pairs and consequent logistic reasons. In such multilingual environments an **interlingua** can be considered an alternative to the transfer-based approach. Here, input is not transformed into an intermediate representation dependent of the source language but into one that is independent of source or target, reducing the number of modules into two (analysis and generation) for one language and 18 for 9 languages in the system. The Interlingua approach is particularly rational when constituent languages are close to each other.

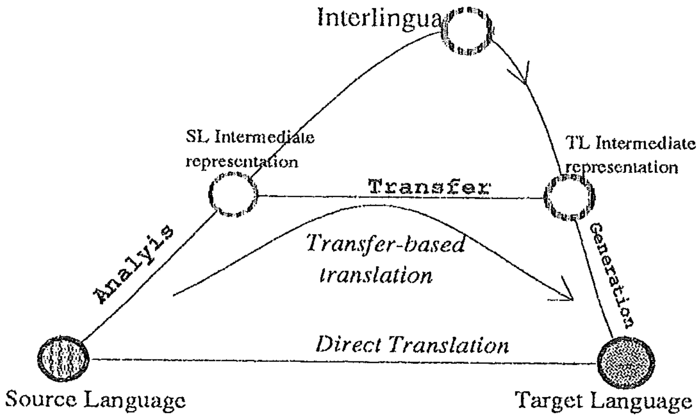

Interlingua
SL Intermediate representation
TL Intermediate representation
Transfer
Analysis
Transfer-based translation
Generation
Direct Translation
Source Language
Target Language

Figure 1: Translation methods.

2 Aspects of Machine Translation

Apart from the direct, transfer-based, and interlingual translation methods with linguistic rules, there have been attempts to build more intelligent systems by enhancing thr traditional methodologies with knowledge components, particularly when linguistics is inadequate in solving problems of pragmatics, anaphora, or ambiguity. For solving lexical ambiguity, for example, IS-A graphs, and the inclusion of semantic variables in unification formalisms are both popular. The CMU MT laboratory has been long developing their knowledge-based translation approach. There have been more radical AI based experiments but they could not get out of the laboratory either. The employment of statistical learning, on the other hand, enjoys increasing popularity in constructing decision mechanisms for parsers and taggers, and there is the radical IBM project which follows a purely statistical method, mainly by comparing the occurence of patterns in large bilingual corpora and concluding with translation probabilities of lexical units with respect to its context (Brown et al 1992).

2.1 Human Intervention

Most MT systems necessitate human correction at the end, and quality is determined by the amount of postediting. Restricted language and controlled input can lead to high quality but then this would not be translation of natural language. To decrease the postediting requirement, some MT systems allow human interaction during translation routines. By consulting the user for ambiguity or dubious patterns during any of the three phases, interactive systems can generate more reliable output. However the amount of human interaction should not reach the level where systems are used as an assistance for human translation task. Human interaction can be monolingual or bilingual. Monolingual interaction can take place during analysis or synthesis. UMIST's NTRAN project is an example of interaction by a source-language monolingual user (Somers 1990). In terms of industrial applications, a monolingual user environment is preferable to a multilingual one.

2.2 Industrial Availability

As mentioned earlier MT was initially found interesting in its prospects to translate technical data acquired from the enemy so that non-Russian speakers could judge developments in the Eastern Block. The USAF still employs a SYSTRAN version for such purposes. Since late 1960s China translated mathematics journals into an intelligible English through domestic MT systems. India, and South Asian countries, as well as Europe, have been benefiting from MT for a wide range of purposes, especially translation of technical manuals. In bilingual Canada every weather report is translated from English into French by METEO, installed in 1976 and considered the first fully automatic high quality MT system. Japan, which joined the race in the last decade, is now the leading country, with the majority of international Japanese companies having their own tailored MT frames. The Malaysian government have decided to translate all textbooks and documents into Malay which is replacing the former colonial English in university education. Independent computational linguistics groups in that country have developed their domestic systems within a national MT policy, and techinal books are already being translated with these products.

2.3 Linguistic Ambiguity

Ambiguity, lexical or structural, determines parsing time in linguistics-based MT systems. Most of the analysis phase is spent on resolving these, and constructing the tree structure which represents the actual sentence. Examples for lexical ambiguity are : categorial as in *Foot Heads Arms Body* words having both noun and verb value (and headlining Michael Foot is heading a committee on arms), analytical as in **paper** which is the name of the material and the kind of text at the same time, or transfer as in Greek, which can be translated both as Rum or Yunan into Turkish. Structural ambiguity is the case when multiple parse trees can be obtained, such as **he saw a girl with a telescope**, the prepositional phrase modifying either the verb or the object. In translation to another language, either interpretations may have different equivalents. Although resolution of ambiguity is effortless for human brain, computationally they are difficult.

The aim of Machine Translation is to transfer messages in a source text into a target language text. This aim can be achieved with incomplete analyses and ignorance of meaning. In this point MT is different from other NLP fields which intend to transform linguistic messages into a form to activate computational routines. If we can get away with using short-cuts and unprincipled pragmatism in our design, it is perfectly acceptable. Turkish does allow such short cuts. Historically, Turkish has been subject to various points of influence, lexically, morphologically, and even syntactically. The language is, as a result, extremely flexible in terms of linguistic structures. Most syntactic divergencies from canonical form in major world languages have sufficient equivalence in Turkish (Özgüven 1991). Therefore Turkish can be mapped into from syntactically ambiguous constructions of English language. The syntactic flexibility makes Turkish a good target language and a bad source language.

2.4 Sublanguages

The real difficulty of translation between two languages rises not from the difference in structural difference in linguistic symbolism of concepts but the difference between the concepts themselves. Each linguistic environment has its distinctive perspective to real world events. The difference is larger in abstract concepts. To determine the "best" rendering the translator should "understand" the meaning in the source language, thus fully automated translation of general languages is only possible when the ultimate targets of AI are achieved. Instead of the general language, researchers have been designing MT to target specified subfields with parallel conceptual organization, such as technical manuals. Sublanguages are not artificially restricted languages; the rules and the frequencies of the domain hve to be carefully studied. The MT research in **sublanguages** therefore requires the analysis of large corpora with statistical methods as well as a careful description of the terminological structure attached to the concepts.

3 Translation of English News Reports into Turkish

The sublanguage of news reports is not as firmly defined as technical fields. Many reports contain parts of human speech, some which report less important events use a less formal language. But there is a clear distinction from newspaper reports and commentaries, and coherence in terminology. We designed the backbone of an integrated system that will allow the semi-automated translation of wired newsreports in English for Turkish-only speakers.

3.1 Working Environment

The design foresees an integrated environment that provides the monolingual user tools necessary to run the translation task alone. These are dictionaries, thesauri, and other suplementary software. The user is to answer the questions directed by individual translation modules to fine-tune the output. Most lexical ambiguities or alternative translations can be resolved by the user. The environment should give the user the freedom to overview the partially translated text and leave certain answers to the end or to refer to other text.

The design of the interactive software tools is as important as the core translation machine. Even so, the output may still not be the perfect translation, e.g. the Turkish text will contain forms that appear more frequent than normal. Therefore the output quality will not be good enough for Turkish newspapers, but it will be fully intelligible for the monolingual user, and ready for forwarding to a human translator who will make the final corrections. In our model system we have not simulated the environment but only the core translation mechanism.

3.2 Translation Procedure

Input news reports are firstly preprocessed and broken down into sentences, which are built of lexical items with pointers to dictionary entries. The Phrase Analysis module marks closely bound groups of words, such as noun phrases. From this point on translation is divided into two mutally independent parts that process the phrases internally and externally. Phrase Translation returns Turkish equivalents of every group. On the parallel branch, the sequence of phrases are structurally analysed, and Structural Translation transformes the representations into Turkish equivalents. In the final stage two lines are assembled to produce the Turkish text.

We proposed a novel linguistics approach and left the structural ambiguities, that create computational problems in parsing, after we observed that most of them exist in Turkish. Also, a small amount of knowledge representation of terms is employed in the Phrase Translation phase.

3.3 Conclusions

Having analysed 1600 newsreport sentences manually, we tested a 46 sentence corpus (with corresponding dictionary) and the results were good. We wrote the prototype software in **LISP** on a **SUN** 4 architecture. Average translation time was 4 seconds per sentence. Of course these results are produced by only a small model of the actual design, and to realize the actual working environment, the interactive software has to be integrated with the core system.

Machine Translation is still a new subject, it is used in less than 0.1 % of the industrial translation market. Therefore the large gap in MT should be filled particularly if engineering is easier in points where concepts are universal, language is regular, and expectations are moderate.

4 References

1. Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Robert L. Mercer and Surya Mohanty, *Dividing and Conquering Long Sentences in a Translation System* (Proceedings of Fifth DARPA Workshop on Speech Natural Language, Defence Advanced Research Projects Agency Information Science and Technology Office, February 1992).
2. Jeremy, G. Carroll, *Graph Grammars: an Approach to Transfer-based MT Examplified by a Turkish-English System* (PhD Thesis, Centre for Computational Linguistics, UMIST, Manchester, 1989).

3. C. Copeland, J. Durand, S. Krauwer, and B. Maegaard, *Studies in Machine Translation and Natural Language Processing, Volume 1* (Comission of the European Communities, Luxembourg, 1991).

4. Jerry R. Hobbs, Douglas E. Appelt, John Rear, and Mabry Tyson, *Robust Processing of Real-World Natural Language Texts* (Proceedings of the Third Conference on Applied Natural Language Processing, Association for Computational Linguistics, Trento, Italy, April 1992) 186-192.

5. W. J. Hutchins, *Machine translation: past, present, future* (Ellis Horwood, 1986).

6. W. John Hutchins and Harold L. Somers, *An Introduction to Machine Translation* (Academic Press, London, 1992).

7. Makoto Nagao, *Machine Translation: How far can it go?* (Oxford University Press, 1989, transl. by Norman D. Cook of *Kikai Hon'yaku wa doko made kano ka*, Tokyo, 1986).

8. Sergei Nirenburg, *Machine Translation: Theoretical and methodological issues* (Cambridge University Press, Cambridge, 1987).

9. M. Kutluk Özgüven, *An English - Turkish Machine Translation System of Newsreports - Which Preserves Ambiguities* (MSc Dissertation, Centre for Computational Linguistics, UMIST, Manchester, 1991).

10. Harold [L.] Somers, *Current Research in Machine Translation* (Proceedings of the 3rd International Conference on Theoretical and Methodological Issues in the Machine Translation of Natural Language, Austin, TX, 1990) 1-12.

11. Harold [L.] Somers, John McNaught and Zaharin Yusoff *A user-driven interactive machine translation system* (Proceedings of the SICONLP/90, The Seoul International Conference on Natural Language Processing, Seoul, 1990) 140-143.

12. Jun-ichi Tsujii and Kimikaze Fujita, *Lexical Transfer based on bilingual signs: Towards interaction during transfer* (Fifth Conference of the European Chapter for Computational Linguistics, Berlin, Proceedings, 1991) 275-280.

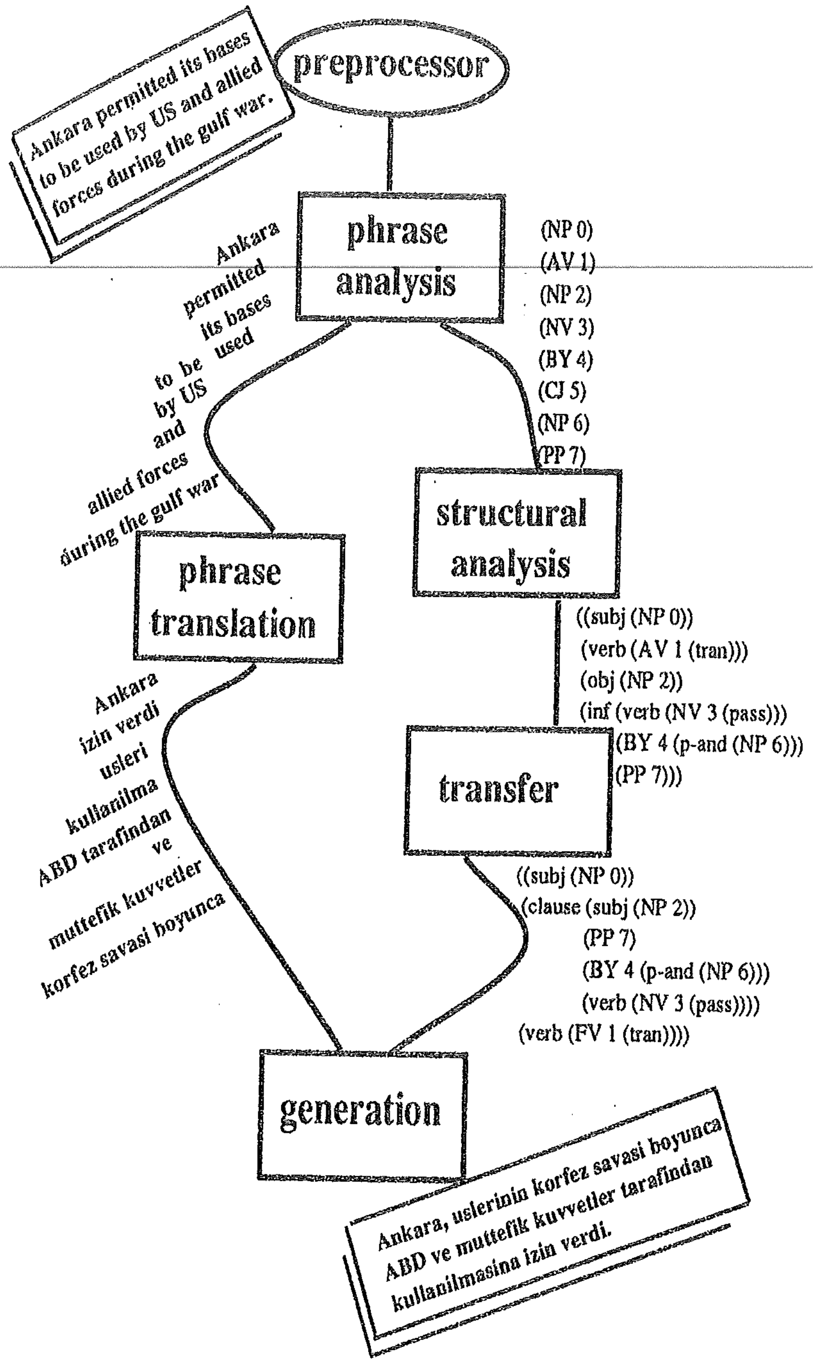


Figure 2: Translation procedure.

FUNCTIONAL CATEGORIZATION OF KNOWLEDGE

Sakir Kocabas

Marmara Research Center, PK21 Gebze, Kocaeli, Turkey

Abstract

The continuous increase of human knowledge rendered the classification of knowledge an important task, from very early ages, for philosophers like Aristotle down to the modern age, to Wittgenstein. These classifications were necessitated by the difficulties in understanding, memorization and transmission of knowledge. An analogous task now is faced in knowledge based artificial intelligence systems as the needs arise to build larger and more versatile systems. In this paper we introduce a method for organising knowledge into several linguistic categories. We describe how this categorization introduces clarity in representing different types of knowledge, how it facilitates the analysis of complex propositions into their simple constituents, and how these in turn can be assembled into complex constructs such as frames and schemata.

1 Introduction

Large knowledge systems of the future, especially those that will represent scientific theories in physics, chemistry, biology astronomy, etc., cannot be confined to narrow domains of expertise. However, as the amount of knowledge given to or acquired by a system increases, two main problems arise: knowledge acquision bottleneck (Lenat, Prakash, & Shepherd, 1986) and brittleness (McCarthy, 1983; Holland, 1986). Knowledge acquisition bottleneck is related to the acquisition of new knowledge by instruction or various other methods of learning. Minsky (1984) and Lenat et al. (1986) point out that, in acquiring new knowledge, the human mind overcomes this problem by recalling similar concepts it already knows about and by recording the exceptions to the case in consideration. Brittleness on the other hand, is related to the difficulties in having the knowledge system expand beyond the scope originally contemplated by their designers.

One solution to the brittleness problem, proposed by McCarthy (1983), is to provide a system with the ability to acquire commonsense knowledge and reasoning. Another solution, which emphasizes inductive learning and is based on general purpose learning algorithms, has been proposed by Holland (1986). His general purpose "classifier systems", are essentially reactive rule based systems relying on genetic algorithms.

Holland (1986) proposes induction as the basic - and perhaps the only - way of making large advances in overcoming brittleness. In considering the specific problems induction faces in this context, Holland identifies the creation of useful ways of categorizing input as the primary task. He suggests that categories must be incorporated into rules that "point" both to actions and to an aura of associated categories. That is, as they are induced, they must be arranged in a "tangled hierarchy", enabling the system to model its environment appropriately.

[1] In his Categories, Aristotle defines nine concept based categories (e.g., see, Edwards, 1967, p. 155). According to his classification, individual expressions signify substance, quantity, relation, time, position, condition, action, or passivity'

Holland seems to have an Aristotelian notion of concept-based categorization1 in mind here as opposed to functional categorization of knowledge. By "category", he understands common abstract features between objects or frames, so that when these are captured, one set of feature can be used in developing a similar frame. (Frames can be regarded as complex propositions linked together by inheritance features.)

Lenat (1983) proposes to carry along multiple representations simultaneously and to shift from one representation to another to enable the knowledge system to carry out the most frequent operations more quickly. He says that this has not been much studied or attempted in artificial intelligence, except in very small worlds. Lenat's CYC system (see, Lenat & Guha, 1990) makes use of this idea.

Porter and Kibler (1986, p. 283) state that in machine learning research, most systems have been built on a small number of rules (heuristics) without having to address the problem of organizing learned knowledge into a coherent, efficiently accessible whole. Also, Cheng and Fu (1984) emphasize that, compared with knowledge representation or the formalization of concepts, little work has been done in the area of knowledge organisation.

Woods (1986) offers a simple classification of knowledge. He states that the "knowledge of the world" consists of two kinds of things - facts about what is or has been true (the known world state) and rules for predicting changes over time. He mentions the need for taxonomic organisation. Woods also argues that the standard notations and semantics of the predicate calculus are insufficient by themselves - they need to be supplemented with additional mechanisms e.g., for non-monotonic reasoning, and metalogical reasoning.

A "functional" knowledge system described by Brachman and Levesque (1983), distinguishes between "definitional" and "factual" information. Their system contains two "boxes" of knowledge; one for maintaining analytical knowledge, and the other to build descriptive domain theories. They also use two languages for their representation, a frame language for analytical knowledge and an "assertional langage" for the descriptive domain theories.

Langley (1986) states that although concept learning has been a basic mainstay of the machine learning community, most research in this area has ignored a number of well-established psychological phenomena. He says that basic-level categories appear to have a special status in human memory, being retreived more quickly and being acquired earlier than other concepts, and suggests that more work is needed on concept formation, for such research would yield a better understanding of human concepts and their acquisition, and it should also lead to improved methods of nonhuman concept learning (also see, Gennari, Langley & Fisher (1989).

His notion of category referred here, is also concept based, in which clusters of concepts are regarded as categories, which in turn, are organised in hierarchies. A concept based approach addresses the issue of conceptual organisation of knowledge. Whereas, it can be argued that a considerable proportion of human cognitive activity is propositional. Therefore a functional organisation of knowledge needs to be developed at least in equal priority and depth.

The categorization intoduced in this paper is based on fundamental methodological and linguistic criteria: methods of verification, meaning (use), and the function of expressions. It is not aimed at the classification of concepts, but of simple propositions. In philosophical terms, our concept of category is based on the deep grammar of propositions and therefore, is quite different from the concept-based notions.

2 Theoretical and Philosophical Background: Piaget and Wittgenstein

The future of artificial intelligence, to a certain extent, depends on the studies on cognitive development. Because of its better tractability by means of natural language, human cognitive development is still the best source for developing knowledge based models in artificial intelligence. The foundations of cognitive science was laid by Piaget's work on human cognitive development in the early 1920s.

Piaget (1971) made extensive studies on human cognitive development and the development of language. There are several reasons why his work is of interest: It is related with 1) the linguistic methods of knowledge acquisition (questions and their classification), 2) the order of knowledge acquisition according to the types of knowledge acquired, 3) the methods of relating the acquired knowledge, and 4) the theoretical foundations for the organisation of knowledge.

Piaget (1971, p. 30) draws our attention to the importance of questions in cognitive development. From the standpoint of cognitive development, a question is a spontaneous search for information. He studies questions asked by the child between the ages of six and seven, and classifies them as questions 1) for causal explanation, 2) about reality, 3) on actions and intentions, 4) rules, 5) about classification, and 6) arithmetical questions.

It may be worthwhile to consider the child's questions from the standpoint of the organisation of acquired knowledge. Explanations seem to play a critical role in these activities. It is conceivable that the child's mind is actively involved in organising its knowledge during the knowledge acquisition processes. Analogously, intelligent knowledge systems may have to be given the ability to ask as many meaningful questions as possible during knowledge acquisition and learning, particularly in the development stage of such systems. The difficulties encountered in knowledge acquisition can be avoided by the use of such strategies of learning frequently used by the child. Lenat et al. (1986) use a similar strategy in developing the knowledge base of their CYC system, but their knowledge acquisition methods are not automated.

Piaget's observations on the "why" questions of the child can be viewed as the manifestation of the operations of an effective organizational capability for acquired knowledge. The child's lack of interest in logical justification of explanations can also be explained within this perspective: Confronted with a vast amount of knowledge to be learned, the primary cognitive problem must naturally be how to organise the acquired knowledge rather than learning how any proposition can be theoretically, logically or mathematically justified. Piaget's classification of questions and explanations, albeit for psychological purposes, constitutes an important step in understanding how expressions are used in language and how they might be classified according to their function and methods of verification.

Having seen some of Piaget's views on human cognitive development, we can now take a brief look at Wittgenstein's philosophical work on the grammatical distinctions between various types of expressions. Wittgenstein considered the functional classification of expressions as one of the most important tasks in philosophy. In one of his earlier works (Wittgenstein, 1965, pp. 44-45), he provides an illustrative analogy between the task of organising a heap of books into a library and the classification of knowledge. Wittgenstein pronounces that some of the greatest achievements in philosophy could only be compared with taking up some books which seemed to belong together and putting them on different shelves; nothing more being final about their positions than that they no longer lie side by side.

Wittgenstein's use of the methods of distinction between various types of expressions are at times implicit and scattered in his books. At Cambridge in 1939 (see, Wittgenstein, 1974), he devoted a whole series of lectures on how to distinguish mathematical sentences from non-mathematical statements. For example, in (Wittgenstein, 1974, p. 34) he says: "'Professor Hardy believes that *x1 > x0'* is not a mathematical statement. It is no more a mathematical statement than 'William said that *7x8=54'* is a mathematical statement." Here, Witggenstein is trying to draw a distinction between a factual statement and a mathematical sentence. Indeed, later (Wittgenstein, 1974, p. 111), he clearly states that in his discussions he aims to show the essential difference between the uses of mathematical propositions and non-mathematical propositions which seem to be "exactly analogous" to the former. One of the criteria that he proposes for distinguishing between mathematical and non-mathematical statements is the prefix "by definition ...", which is applicable to logico-mathematical and formal statements, but not to factual statements. In his various studies, Wittgenstein distinguishes between philosophical statements and theoretical statements; first-person psychological statements and theoretical statements; and basic belief statements and theoretical statements (e.g., see, Wittgenstein, 1953; Wittgenstein, 1970; Wittgenstein, 1978).

3 The Method of Categorization

The development of large and reliable knowledge systems requires an effective knowledge organisation. It seems that one way of achieving this is to categorize simple propositions into functionaly different classes and then assemble them into frames as necessary. Here there are two questions to be asked: 1) How many grammatically different categories can be identified in language? 2) What can be the criteria to be used in distinguishing between these categories? Before we attempt to answer these questions, the problems of consistency and completeness in knowledge systems need to be reconsidered.

3.1 Consistency, Completeness, Validity and Truth

Since Gödel's (1962) famous theorem, it is widely accepted that an overall completeness and consistency may not be achievable in large bodies of knowledge. However, a domain dependent consistency and completeness can and needs to be maintained. It was Wittgenstein (1981, p. 118) who pointed out that a contradiction is not the end of the world, but something that sets a limit to a particular "language game" or "grammar". His remarks opened the way to a context dependent concept of consistency in the philosophy of language. More recently, from the viewpoint of artificial intelligence, de Kleer (1986) argues that there is no necessity that the overall database be consistent in a knowledge system. He suggests that a context dependent concept of consistency provides a better way of achieving control in problem solving. Similar arguments can be found in recent artificial intelligence literature (see, e.g. Lenat & Feigenbaum, 1987; 1991.)

It may even be meaningless to try to achieve an overall consistency and completeness in complex bodies of knowledge. The simple reason is that "truth" and "verification" have different meanings in different "grammars". In other words, quite different methods and criteria are employed in establishing the "truth" of propositions in different categories. To illustrate, let us consider the expressions:

(1) $1^3 + ... + n^3 = (n(n+1)/2)^2$.

(2) Jupiter is a planet.

(3) In a nuclear reaction, the amount of energy released is equal to the rest-mass difference multiplied by the square of the velocity of light ($E=mc^2$).

(4) Calculus was invented by Newton and Leibniz independently.

The criteria and methods that are used in establishing the "validity" of each of these expressions are quite different: The first expression is a mathematical statement which can be proved by the axioms of arithmetic using mathematical induction, and no factual investigation is needed to prove it. The second is a formal statement the "truth" of which is implied by the order of the concepts "planet" and "Jupiter" in language. Unlike theoretical, hypothetical, empirical statements, formal statements are not compared with facts for their validity. The third one is a theoretical statement which is used as part of a model to understand and explain certain physical phenomena. Finally, the last sentence is a historical statement, the "validity" of which is established by the methods used in historical study.

These four statements can be considered as belonging to four different categories, namely the categories of logico-mathematical, formal, theoretical, and historical statements. Propositions belonging to these categories can be found in the body of almost any scientific work. Therefore, knowledge systems should take into account such categorical distinctions. A system of criteria has been developed for functional classification of propositions, based on the works of Piaget (1971) and Wittgenstein (1974) outlined above. The principles for the functional categorization of propositions will now be described.

3.2 The Categorization Criteria

Propositions can be categorized according to their functions (uses) in language. The criteria that can be used in such categorizations can initially be divided into three groups:

1) Methodological (epistemic) criteria. The differences between the methods of verification (or falsification) of propositions may help to determine their categories. Verification methods of theoretical statements are different from those of logico-mathematical statements. In addition, verification may not apply to some propositions at all (e.g., allegorical and fictional sentences). A theoretical, hypothetical or empirical statement must be testable, or else must be derivable from testable theoretical statements, while logico- mathematical and formal statements are not verified by testing (i.e. by observation and/or experimentation).

Verification of formal statements is embedded in the logical structure of their constitutent concepts in language. In this way, formal statements reflect the concept structure in language. For example the sentences, "A cat is an animal," and "Calcium is an element" are formal statements and no one would ask for their verification by the verification methods of theoretical, hypothetical or empirical statements. Definitions also belong to the category of formal statements. On the other hand, logico-mathematical statements (e.g., theorems, non-theorems, lemmas and corollaries) are subject to the criterion of provability in a formal or informal calculus.

2) Grammatical labels used in language in the form of prefixes and postfixes., which play a role in the categorization of expressions. Some examples are: "By definition, ...", "According to the theory, ...", "According to the hypothesis, ...", "According to our experiences, ...", "I believe that, ...", "I do not believe that, ...", "I feel that, ...", "Probably, ...", "Possibly, ...", etc.. It seems that at times we use these prefixes like flags or labels for classes of expressions, to signify their places in our knowledge.

Each of these prefixes are meaningful with the expressions of a certain category, while with others they are not. For example, it is meaningful to say: "By definition, *2+2* is *4*," while it is not meaningful to say: "By definition, the gravitational force between two bodies is proportional to their masses and inversely proportional to the square of the distance between them." Similarly, it is not meaningful to say: "According to the hypothesis, *2+2* is *4*," while it is meaningful to use the prefix with a theoretical or hypothetical statement.

3) Comparisons and contrasts of propositions with the ones already identified as belonging to a category. Comparisons can be very useful in cases where other criteria may fail to identify the category of a statement. Consider Fermat's "last theorem" for example. One might think that it is a hypothesis like those which physicists use to hypothesize certain subatomic particles (e.g. quarks). Fermat's last theorem is easily comparable with some other familiar arithmetical theorems, whereas physical hypotheses are more readily comparable with other similar physical hypotheses.

By using these criteria, it is possible to build a system of categorization which can be used in distinguishing the following categories: i) logical propositions, ii) mathematical statements, iii) formal statements, iv) grammatical (or meta) statements, v) theoretical-hypothetical-empirical statements, vi) historical sentences, vii) factual statements. In addition to these, there are other identifiable categories which can be listed as: a) basic belief statements, b) sensory and intentional statements, c) metaphorical and allegorical statements, d) fictional sentences. However, these will not be considered here, because the seven categories above seem to be sufficient to represent scientific knowledge to a large extent.

A list of questions that can be used in determining the category of a given statement S, is as follows:

- Does the statement S describe an event currently in effect?
- Does the statement S describe a past event or state of affairs?
- Does the statement S define a concept?
- Is the statement S testable against facts?
- Is the statement S varifiable by observations/experiments?
- Is the statement S provable/deducible?
- Is the statement S about another statement?
- Is the statement S a comment on a property/relation?
- Is the statement S a comment on a state of affairs?
- Is the statement, "Possibly, S" meaningful?
- Is the statement, "Probably, S" meaningful?
- Is the statement, "By definition, S" meaningful?
- Is the statement, "According to the hypothesis, S" meaningful?
- Is the statement, "According to the theory, S" meaningful?
- Is the statement, "According to the experiences, S" meaningful?
- Is the statement, "According to the rules, S" meaningful?
- Is the statement, "According to (so and so), S" meaningful?

The determination of the category of a given statement is a classification problem, in which the parameters are the questions listed.

3.3 Examples of Categorized Propositions

Some examples of statements belonging to the seven categories named above are as follows:

Logical Statements:

- A proposition and its negation imply all other propositions.
- A proposition P is derivable from propositions P and Q.

Mathematical Statements:

- The sum of the inner angles of a triangle is 180 degrees.
- There are natural numbers x, y, z, such that $x^3 + y^3 = z^3$.

Formal Statements:

- Jupiter is a planet.
- Cu is the chemical symbol of copper.

Grammatical Statements:

- Superconductivity is an important property.
- The statement, "Aluminium substitution reduces superconductivity," is consistent (with the knowledge base).
- Why $BaPbBiO_3$ is a superconductor is not explainable (by the existing knowledge).

Theoretical-hypothetical-empirical Statements:

- The decay products of neutron are proton, electron and antineutrino.
- Electronegativity of nickel is 1.9.
- In metallic superconductors, superconductivity and electron density are positively related.

Historical Propositions:

- Superconductivity was discovered by H.K. Onnes in 1911.
- Partial substitution of sulfur for oxygen in $LaNiO_3$ has been tried in superconductivity experiments.

Factual Statements:

- The price of Aluminium is 1.75 US dollars.
- The price of Scandium is over 50,000 US dollars.

Simple and complex propositional knowledge can be further organised within each category into levels of expressions. The levels can be based on Russell's logical theory of types (see, Whitehead & Russell, 1970). The categorization has been applied to varying extents in four different computational models of scientific reasoning and discovery in particle physics (see, Kocabas, 1989a; 1989b; and 1991), and is currently implemented in the development of a computational model of discovery in oxide superconductivity. These will not be described here, but a discussion based on our observations about the implementations will be given instead.

4 Discussion

The importance of the categorization of knowledge lies in the following possible advantages it may provide in the acquisition, representation, refinement and effective use of knowledge: 1) simple and complex propositions of different categories can be represented, accessed and used separately, 2) automatic transformation of knowledge from one form of representation into another one, especially from predicate statements into frames, can be made easier, which can be very useful in knowledge acquisition during the development of large knowledge systems like CYC (Lenat et al., 1986), 3) some useful general formal and informal rules applicable to each category can be found, 4) logic mistakes in designing knowledge systems can be minimized, 5) search procedures can be made more effective, for categorization eliminates unnecessary search activities in the system, (e.g., formal questions can be answered by conducting search only in formal knowledge, theoretical questions in theoretical knowledge and so on), 6) more detailed and effective identification and resolution of conflicts and theory revision (or truth maintenance) is possible, and 7) dynamic reorganisation of knowledge can be made easier.

Categorization of knowledge facilitates truth maintenance, as in such systems "truths" of propositions of different categories are established differently.(A more detailed description of conflict resolution based on the categorization is given in Kocabas, 1989c.) Furthermore, some priorities of validity can be given to knowledge belonging to certain categories. For example, if a factual statement is in contradiction with a hypothesis, the problem is resolved by simply giving priority to factual knowledge over the theoretical, rather than removing the hypothesis.

In a categorized knowledge system, hypothesis generation is more systematic. New hypotheses are generated from factual and theoretical knowledge by induction and other forms of generalization. Similarly, new hypotheses can be generated from theoretical knowledge by specialization, abstraction and abduction. Certain forms of reasoning do not apply to certain categories of knowledge (e.g., specialization is not applicable to factual knowledge).

Combined uses of frame and logic representation provides a more structured representation of knowledge. Frame representation by itself is less efficient in making full use of logical and extralogical inference such as deduction, abstraction and abduction (see, Lenat & Guha, 1990). On the other hand, logic representation alone provides a less efficient basis for certain kinds of reasoning (e.g. taxonomic and analogical reasoning). The integration of frame and logic representations allows the copying and transformation of knowledge from frames into predicate statements. Brewka (1987) describes a method for translating knowledge from frames into predicate statements. However, he makes no mention of any theoretical work on translating predicate statements into frames in a general and systematic way. Categorization provides this opportunity, as it allows the transfer or transformation of predicate statements into frames.

In a categorized system, knowledge acquisition does not have to be in the form of frames. Knowledge can be entered in simple predicate statements to be categorized by means of a small set of transformation functions. They can then be automatically structured into frames by means of a set of knowledge assembly functions. Naturally, the frames must reflect the categorical distinctions in their structure. For example, every frame can have slots corresponding to the category names such as "logical", "formal", "the" and "factual".

Categorization facilitates the analysis of descriptive knowledge. Other methods of expressing complex propositions in terms of their atomic constitutents had been developed. E.g. Kowalski (1979, p. 33) describes the method of expressing n-ary relationships as a conjunction of n + 1 binary relationships. Complex descriptions can also be represented in this way. This method is directly applicable in frame systems, but it decomposes propositions into decriptions instead of atomic sentences, which may not always be desirable for reasons of hindering the effective use of logical and extralogical inference.

Complex propositions can be analyzed to and represented in simple, categorized predicate statements. Consider, for example the proposition: "The pulsar, which was discovered by radio astronomers is a rapidly spinning neutron star whose radio signal regularly turns off and on." This sentence can be split into simpler propositions:

formal:	A pulsar is a neutron star.
the:	A pulsar spins rapidly.
the:	The radio signal of a pulsar rapidly turns on and off.
historical:	The pulsar was discovered by radio astronomers.

As is seen, the constituent propositions are not only representationally, but also categorically distinguishable. Feigenbaum has recently redefined problem solving in terms of "knowledge assemby" rather than search (see, Engelmore & Morgan, 1988, vii). In this new definition the emphasis is on "finding the right piece of knowledge to build into the right place in the emerging solution structure". In a categorized knowledge system simple propositions can be assembled together to build more complex constructs such as complex propositions, frames and schemata.

Inevitably, the categorization has its own drawbacks, beside introducing hopes in resolving some important knowledge level problems in the development of large knowledge systems. First of all, it imposes a structure on descriptive kowledge, which is based on a set of criteria. The validity of these criteria can be argued, but if humans utilize such criteria in organising their knowledge, it is worth considering to utilize them in computational models.

Another problem can be with certain propositions whose category may not be easy to decide. In such cases, the proposition can be maintained in more than one category. However, such cases are rare, and the duplications do not pose a serious difficulty. On the other hand, categorization errors can be identified and resolved by a set of "knowledge administration functions" which can be built to supervise the "distribution", "maintenance" and the "assembly" of knowledge in the categories of the system.

Holland's (1986) suggestion of using general purpose learning algorithms and giving emphasis on inductive learning as a solution to the brittleness problem has appeal. However, systems based on such methods use complex input and output units and their complexity grows as the system is given knowledge of different kinds and levels. In the end, the brittlenes problem transforms into two problems: Input and output management. To a certain extent this can be avoided in a multi layered general purpose system, but the theoretical basis for such systems has not been sufficiently developed. Additionally, Holland's proposal underestimates the role of the deductive methods in learning.

The categorization scheme introduced, is much more detailed than the classification suggested by Woods (1986) in which he divides knowledge into "facts" and "rules". His "facts" include what we call factual statements, simple logical, formal, mathematical, theoretical, and historical propositions. His "rules" include complex logical, formal, mathematical, and theoretical rules, which can be called rules of inference, as well as action rules (or production rules). The categorization meets some of the requirements proposed by Aiello et al., (1986), as it allows knowledge and metaknowledge to be represented in the same form. In this way, it allows the inference mechanisms to be accessible at both levels.

5 Conclusions

In this paper we described a methodology for organising descriptive knowledge into several functional categories, and discussed the ways in which it can integrate different methods of representation such as predicate logic and frame representations. The categorization helps to resolve some of the major problems of developing and maintaining large knowledge systems. These problems are, brittleness, knowledge acquisition bottleneck, and the identification and resolution of conflicts. Clarity and simplicity are essential in building complex knowledge systems, because as the system grows, it becomes more and more difficult to keep track of the relationships between domain concepts. (Lenat et al., 1986; 1990) provide dramatic examples of the complexities of adding more knowledge to a large knowledge system.) The categorization scheme introduced, provides clarity and simplicity, as it allows different kinds of knowledge to be represented in an organised way. Commonsense, as well as scientific knowledge can be represented in the categories described. The categorization has been implemented in several computational systems which model reasoning and discovery in astronomy, particle physics, and high-temperature superconductivity, incorporating various methods of learning and conflict resolution. More detailed implementations are being carried out.

References

Brachman, R.J. & Levesque, H.J. (1983). Krypton, a functional approach to knowledge representation. *IEEE Computer. October 1983. (67-73).*

Brewka, G. (1987). The logic of inheritance in frame systems. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence, (483-488).* Morgan Kaufmann.

Cheng, Y.S. and Fu, K.S. (1984). Conceptual clustering in knowledge organization. *The First Conference on Artificial Intelligence Applications*, (274-279). IEEE Computer Society Press.

Edwards, P. (1967). *The Encyclopedia of Philosophy.* Collier MacMillan Ltd.

Engelmore, R., & Morgan, T. (Eds.) (1988). *Blackboard systems.* Addison Wesley.

Godel, K. (1962). *On formally undecidable propositions of Principia Mathematica and related systems.* New York: Basic Books.

Holland, J.H. (1986). Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems. In R.S. Michalski, J.G. Carbonell and T.M. Mitchell (eds.), *Machine learning: An artificial intelligence approach.* Los Altos, CA: Morgan Kaufmann.

Kocabas, S. (1989a). Qualitative and quantitative reasoning in particle physics. *Proceedings of the 12th International Congress on Cybernetics, 250-271,* Namur, Belgium.

Kocabas, S. (1989b). Scientific explanation by exclusion. *Proceedings of the 12th International Congress on Cybernetics, 840-851,* Namur, Belgium.

Kocabas, S. (1989c). *Functional categorization of knowledge: Applications in modeling scientific research and discovery.* PhD Thesis, Department of Electronic and Electrical Engineering, King's College London, University of London.

Kocabas, S. (1991). Conflict resolution as discovery in particle physics. *Machine Learning, Vol 6, No 3, (277-309).*

Kowalski, R. (1979). *Logic for problem solving.* New York: Elsevier North Holland.

Langley, P. (1986). Editorial: Human and Machine Learning. *Machine Learning, Vol. 1, No.3, (246-247).*

Lenat, D.B. (1983). The role of heuristics in learning by discovery: Three case studies. In R.S. Michalski, J.G. Carbonell & T.M. Mitchell (Eds.), *Machine Learning: An artificial intelligence approach.* Los Altos, CA: Morgan Kaufmann.

Lenat, D.B. Prakash, M., Shepherd, M. (1986). CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *The AI Magazine 7, No. 4, (65-85).*

Lenat,D.B. & Feigenbaum, E.A. (1987). On the thresholds of knowledge. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence, (1173-1182).*

Lenat,D.B. & Feigenbaum, E.A. (1987). On the thresholds of knowledge. *Artificial Intelligence, 47, (185-250).*

Lenat, D.B. & Guha, R.V. (1990). *Building Large knowledge based systems: Representation and inference in the CYC project.* New York, NY: Addison Wesley.

McCarthy, J. (1983). Some expert systems need common sense.*Annals of New York Academy of Sciences, 426, 129-137.*

Minsky, M. (1984). *Afterword to: Vinge, V., True Names.* (130-153). New York: Bluejay Books.

Piaget, J. (1971). *The language and thought of the child.* Routledge & Kegan Paul Ltd. London.

Porter, B.W. and Kibler, D.F. (1986). Experimental goal regression: A method for learning problem-solving heuristics, *Machine Learning, Vol 1, No. 3, (249-286).*

Rose, D. & Langley, P. (1986). Chemical discovery as belief revision. Machine Learning, 1, (423-452).

Whietehead, A.N. & Russell, B. (1970). *Principia Mathematica to 56.* Cambridge: Cambridge Univertsity Press.

Wittgenstein, L. (1953). *Philosophical Investigations.* G.H. von Wright and G.E.M. Anscombe (Eds.). Oxford: Basil Blackwell.

Wittgenstein, L. (1965). *The Blue and Brown Books.* New York: Harper Torchbooks.

Wittgenstein, L. (1970). *Lectures and conversations on aesthetics, psychology and religious belief.* Ed. by C. Barrett. Oxford: Basil Blackwell.

Wittgenstein, L. (1974). *Wittgenstein's lectures on the foundations of mathematics, Cambridge, 1939.* C. Diamond (Ed.) Sussex: The Harvester Press.

Wittgenstein, L. (1978). *Remarks on colour.* G.E.M. Anscombe (Ed.) Oxford: Basil Blackwell. Wittgenstein, L. (1981). *Zettel.* G.E.M. Anscombe and G.H. von Wright (Eds.) Oxford: Basil Blackwell.

Woods, W.A. (1986). Important issues in knowledge representation. *Proceedings of the IEEE. Vol. 74, No. 10. (1322-1334).*

FOUR LEVELS OF LEARNING AND REPRESENTATION IN MODELING SCIENTIFIC DISCOVERY

Sakir Kocabas

Department of Artificial Intelligence
Marmara Research Center, PK 21, Gebze, Kocaeli, Turkey

Abstract

In this paper we examine how different levels of representation and learning can be integrated in computational models of discovery. We have designed a program, CER, which simulates some of the research activities carried out in the process of the discovery of high-temperature superconductors by Bednorz and Muller in 1986. These activities include goal and strategy choosing, proposing experiments, designing experiments, data collection, generating and testing hypotheses, modifying hypotheses, and generating explanations. The CER system integrates learning methods such as classification, learning by instruction, generalization and specialization, and by explanation based learning. In this way the program can perform learning at three different levels.

1 Introduction

Computational modeling of scientific discovery has emerged as an important research field in artificial intelligence in recent years. A number of discovery systems have been developed, utilizing different methods of learning and representation. The CER system (Kocabas, 1989a) introduces improvements on earlier systems in its representation of scienfic knowledge and integration of methods of learning and discovery at different levels. The system has been designed to model the discovery of high-temperature oxide superconductors in an interrelated set of tasks such as proposing research goals, choosing research strategies, proposing experiments, designing experiments, generating and testing hypotheses, verifying, modifying or deleting hypotheses, and generating explanations.

Newell (1982) describes five different levels of representation in knowledge systems in increasing abstraction as device level, circuit level, logic level, symbol level and knowledge level. Kocabas (1991c) looks at representation in three levels in knowledge systems as device level, symbol level and knowledge level, and investigates the relationships between representation and learning through various methods. Accordingly, he classifies various methods of learning into these three levels of abstraction, and identifies learning in neural nets as device level learning systems, classifiers (Nilsson, 1965), genetic algorithms (Holland, 1975) and semantic nets as symbol level systems, and considers higher level learning methods such as similarity based (Dietterich & Michalski, 1983), explanation based (Mitchell, Keller & Kedar-Cabelli, 1986), and analogical learning as knowledge level methods. Here we add another level of learning: the control level, which concerns learning control rules. Some might object to the separation of knowledge as control and knowledge level (see, e.g. Lenat & Feigenbaum, 1987), but there are some practical reasons as to why the two are to be treated as distinctly. The distinction will be made clear in the following pages.

Majority of the current discovery models integrate learning methods horizontally. For example, the learning methods of AM (Lenat, 1979), EURISKO (Lenat, 1983), GLAUBER (Langley, Simon, Bradshaw & Zytkow, 1987), STAHL (Langley, et al., 1987), KEKADA (Kulkarni & Simon, 1988) and AbE (O'Rorke, Morris & Schulenburg, 1990) can be described as knowledge level methods. Only a few systems such as IDS (Nordhausen & Langley, 1987) integrate different levels (i.e. device and knowledge levels) of representation and learning methods.

The CER system (Kocabas, 1989) integrates learning at three different levels as symbol, knowledge and control levels. The system's symbol level learning is performed by a classifier, knowledge level by a set of research operators, and control level by its explanation based generalization subsystem. In this paper we will describe how CER integrates various methods of learning in its simulation. We will also describe the system's knowledge organisation and representation, as we believe that these are closely related with learning.

2 An Overview of CER

CER consists of a knowledge base, an explanation based generalization program (Mitchell, Keller, & Kedar-Cabelli, 1986), a set of system operators each consisting of a series of condition-action rules, a classifier (Nilsson, 1965), and a small set of primitive Prolog function definitions to assist input, output and list processing. In this section the system's representation of descriptive knowledge, its control architecture and research operators are described and its classifier and how it is controlled by the program's research operators is explained.

2.1 Organisation and Representation of Descriptive Knowledge

CER's descriptive knowledge is represented as categorized predicate statements in Prolog. This representation is basically the same as in our earlier systems (see, Kocabas, 1989b; 1991a), but carries more detail and includes the following categories: 1) logical knowledge, 2) formal knowledge, 3) meta knowledge, 4) theoretical, hypothetical, empirical knowledge, 5) factual knowledge, 6) historical knowledge. An important addition in CER is the representation of processes and events by qualitative schemas (Forbus, 1984).

The system's logical knowledge contains the definitions of logic functions and logical relationships between types of expressions. Formal knowledge contains domain definitions, class memberships and class-superclass relationships. Factual knowledge includes factual statements about domain objects; while theoretical knowledge contains theoretical, hypothetical and empirical statements acquired or generated by the program. Some examples of the system's descriptive knowledge with category labels are as follows:

Logical: "larger" is a transitive relation.
Logical: "same_group" is a reflexive relation.
Logical: Ni is an element in LaNiO3.
Logical: "Na" is the symbol of sodium.
Factual: The price of Scandium is over US$ 50,000 per kilogram.
Theoretical: The critical temperature of Y-Ba-Cu-O compound is 91 K.
Theoretical: Specific heat and oxide superconductivity are related.

Some theoretical statements refer to process schemas such as

```
theoretical(reduces,process([substitution/of,tve,y,ybco],_),tc).
```

which means that a tetravalent element substitution for*Y* in *Y-Ba-Cu-O* superconductor reduces the critical temperature, where the process itself is represented separately by a schema.

CER's meta knowledge contains statements about its domain predicates, hypotheses, etc. It is important to know in record, if certain acquired or generated hypotheses can explain a certain phenomenon. Historical knowledge contains the records of the historical development of research in superconductivity starting from Onnes' discovery of the phenomenon in 1911. It also includes the records of the conducted experiments to avoid repetititons and the records of the refuted hypotheses. Some examples of meta and historical knowledge are:

Meta: The isotope effect holds for metal superconductivity.
Meta: Electron-phonon interaction mechanism explains superconductivity.
Historical: "Superconductivity was discovered by H.K. Onnes in 1911."
Historical: "Al was substituted for Ni in formula LaNiO3 to yield compound LaAlO3."

CER's inputs are the logical, formal, theoretical, factual, meta and historical knowledge in its knowledge base. Since the system's descriptive and definitive knowledge is maintained as separate from its prescriptive knowledge, new descriptive domain knowledge can be added to it. The system also accepts, by interaction, the results of the experiments it has proposed. Some other interactions occur during its activity (e.g. assigned time limits to strategies, methods and experiments).

2.2 CER's Control Level Learning

The system's design includes seventeen system operators and a control operator with an explanation based learning subsystem. As described below, each operator consists of a set of condition-action rules, and carries out a certain research task such as formulating and choosing goals, choosing strategies and designing experiments. Currently, only five of these operators have been implemented.

In order to accomplish its research objectives, the system has to carry out its research tasks in a certain order. (For example, a research strategy is selected only after research goals are formulated and a research goal is selected.) For this reason, CER must know which operator to activate first, and what other(s) after the completion of a certain task. The program learns its control knowledge by explanation based generalization (Mitchell, et al., 1986).

CER has a hierarchic homuncular control architecture (Kocabas, 1991b), basically consisting of two levels of independent operators or "homunculi" (see Figure 1). In this architecture, a system's methodological knowledge is represented as condition-action rules partitioned into a set of level-1 research operators, which in turn, are controlled by a level-2 or control operator by means of a message list. This architecture divides a system's overall activities into actions, activities, tasks and jobs.

Each condition-action rule represents a set of conditions and a set of actions. A condition can be a proposition (to be proved in the knowledge base) or a message (in a message list). A set of actions are carried out, when the conditions of an action rule are satisfied. An action adds/deletes a proposition to/from the knowledge base, or a message to/from the message list. The totality of the actions performed by such a rule is called an activity. The set of activities performed by a level-1 operator is called a task, and the set of tasks performed by a group of such operators is called a job.

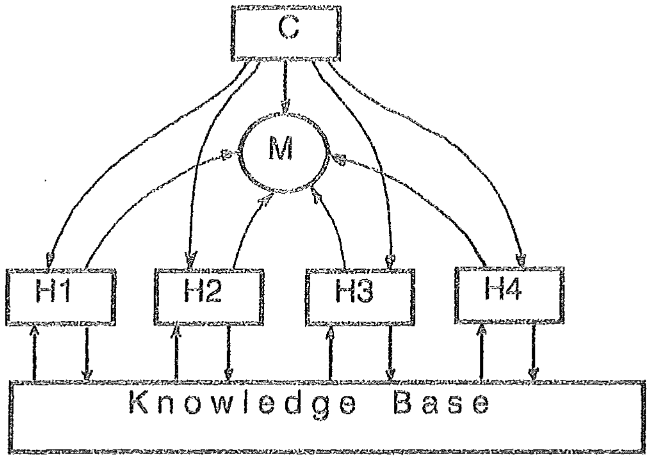


Figure 1. CER's control architecture in which the system's prescriptive knowledge is organized into rules, level-1 research operators (H1,...,Hn) and a control operator (C). Each research operator sends its message to a message list (M). The control operator reads the message list and activates the research operators according to the message states.

The changes in the message list are periodically monitored by the control operator, which activates one or more of the research operators in acordance with the state of the list. As each active operator completes its task, the state of the message list changes, and the control operator activates other operators accordingly. This goes on until all achievable tasks are performed, or the system is interrupted internally or externally. Since the research operators communicate with one another only through the message list, each acting as an autonomous agent in this architecture, like those described by Dennett (1978).

The benefits of this architecture are twofold: First, the system's activities are divided into independent and manageable tasks, activities and actions. Secondly, the control operator can be taught how to distribute control among its research operators. CER learns the necessary control knowledge by using its on-line explanation based generalization module (or EBG for short). During training, the EBG system partially generalizes the message states and generates control rules such as

If the message list consists of the messages
message(p,_,_,_) and
message(q,_,_,_),
then activate operator H2.

So, basically, the learning consists of learning how to associate the new message states with a particular operator. Once the training is completed, the system can focus on new research problems directly under the control of its control operator. Table 1 shows the control rules of CER created during a particular run, where the condition parts of the rules contain generalized message states, and the action parts indicate which operator to be activated.

2.3 Knowledge Level Learning: CER's Research Operators

CER has seventeen research operators (see, Kocabas, 1989), of which only five, namely Goal Setters (GS), Goal Choosers (GC), Framework Setters (FS), Strategy Proposers (SP) and Experiment Proposers (EP) have been implemented, while six others are partially developed. The rest have yet to be developed from design to computational implementation.

Some of CER's operators (namely, GC, SP, KG and EP) employ a simple trainable classifier (Nilsson, 1965) in choosing goals, strategies, methods and experiment materials among alternatives. CER uses only one classifier with different matrices for different classification activities. The classifier is activated by some of the action rules of these operators, and as will be described below, can itself be trained on line. The use of such classifiers as symbol level learning and search units is essential in decision making that involves conflicting constraints. We can now describe some of the system's operators in terms of inputs, activities and outputs. In their descriptions, occasionally their acronyms are used for conciseness.

2.3.1 Goal Setters

Most intelligent activities can be considered as oriented towards achieving a set of goals, though some of these can be ambiguous or general. CER's goal proproposing rules (GS) are confined to scientific interest. Well-defined goals usually have a time limit for their accomplishment.

CER proposes research goals such as explaining an unusual phenomenon, or studying the possibilities of improving a desired property of a compound. One of the GS rules of CER assigns time limits to the proposed goals, and another one checks if the goals are correctly formulated, and asks them to be reformulated otherwise. The system currently has thirteen such rules, high level description of some of which are as follows:

If a phenomenon has not been explained, then make it your goal to explain it.
If a phenomenon is not explainable, then make it your goal to study it
If a physical property is an important property, then try to enhance that property on some substance.
If a goal is entered to the agenda, then assign a time limit to it.

Table 1. Some of CER's control rules generated by its explanation based learning module. Here, the ep(_), sp(_), gc(_), and gs(_) refer to the system's research operators.

```
control :-
        and([message(goal,_,_,_), message(strategy,_,_,_),
              message(current/goal,_,_,_),message(current/strategy,_,_,_)]),
        activate(ep(_)).
control :-
        and([message(goal,_,_,_), message(current/goal,_,_,_)],
        activate(sp(_)).
control :-
        and([message(goal,_,_,_),
        activate(gc(_)).
control :-
        empty_messagelist,
        activate(gs(_)).
```

Inputs to the GS operator are logical, formal, grammatical and theoretical statements in the knowledge base as below, by means of which the system can generate its research goals:

The superconductivity of Ba(Pb,Bi)O3 is not explained.
The superconductivity of oxide coated films of aluminium is not explained.
Superconductivity is an important property.
Oxide superconductivity is a kind of superconductivity.
Ba(Pb,Bi)O3 has superconductivity.

The outputs of GS are expressions labeled as messages with the internal label "goal" and an assigned time limit. A goal can be like

Explain the superconductivity of Ba(Pb,Bi)O3.
Improve superconductivity in an oxide compound.

The time limits to goals are given by interaction, which indicate the length of time that the goal can be on the agenda.

2.3.2 Goal Choosers

Choosing between goals, strategies and experiments can be an important task in scientific research. CER's GC operator currently has two rules, which have both been implemented. One of these rules uses the system's classifier to choose between proposed goals. The classifier uses several criteria: cost, reward, time limit, achievability (by existing knowledge and by existing technology), and likeliness to be achieved by other researchers before. The values to these parameters are qualitative, and some are given to the classifier externally, while others (such as cost, availability) are drawn from the system's factual, theoretical and meta knowledge. The rules of this operator are as follows:

If there are more than one goals, then use the classifier to choose a goal, and label it as the current goal. If there is only one goal, then label it as the current goal.
If the classification fails, then select a goal randomly.

Inputs to this operator is a set of goals labeled as messages with assigned time limits and the qualitative values given to the parameters of the classifier for each goal, while the output is a single, labeled as a message with the internal label "current goal".

2.3.3 Strategy Proposers

When a research goal is chosen, there may be alternative ways (e.g. strategies) to achieve that goal. Moreover, a research strategy may use different methods. Some research goals are simply achieved by literature search, some others by theoretical analysis, and yet others by experimentation. Scientists choose the most appropriate research strategies to achieve their goal in an economic way. CER's SP operator performs its task in accordance with the system's current research goal. The system has fully implemented fifteen SP rules, one of which assigns time limits to the strategies while another one uses the classifier to choose between alternative strategies, on the basis of cost, time required, and likeliness to succeed. The following are two of the SP rules:

If the current goal is to explain a phenomenon, then make a list of all the strategies for finding and explanation.

If the current goal is to improve a property P_1 and another property P_2 is positively related with P1 and a process S_1 improves P_2, then propose experiments to apply S_1.

Inputs to this operator can be formal and theoretical statements, and the message that states the current goal. Outputs are messages that indicate the considered strategies and the current strategy with assigned time limits.

2.3.4 Experiment Proposers

The tasks of this operator include the following: Making a list of relevant processes and techniques for the research, determining the relevant test properties for an experiment, determining the experiment materials having these properties, choosing the best material from a list of materials by classification against criteria such as availability, likeliness to produce success, cost and relative hazards. CER currently has eleven EP rules, three of which use the classifier to choose experiment materials, substitution elements and substituting elements among alternatives. Some of the rules of this operator are as follows:

If the current strategy is experimentation applying a particular process, then record that process as the current process.

If the current strategy is experimentation, and a process has been chosen, then select the experiment materials with the relevant properties.

If there are alternative experiment materials for the same process, then choose the best material by classification.

Inputs to Experiment Proposers can be formal and theoretical statements,and messages that indicate the current goal and current strategy. Outputs are messages indicating the relevant processes, relevant properties, the current process, experiment materials, current experiment materials, substitution elements and substituting elements.

2.3.5 CER's Other Operators

We will not describe the system's other operators, but will provide a summary description of its Hypothesis Generators and explain the methods of learning used by this operator. Hypothesis formation is one of the most important activities in scientific research, and has been studied extensively (e.g. see, Lenat & Feigenbaum, 1987; Darden, 1987).

The tasks of CER's Hypothesis Generators include: generating hypotheses on the variations in the physical, chemical, etc., properties of the system under study, using induction, abstraction, abduction, and analogy. The program has currently forty such rules most of which were extracted from the research reports on oxide superconductivity. Some of these rules are as follows:

If a physical effect P1 cancels another effect P_2, then hypothesize that there is another efect related with P1 and P_2.

If the value of a property P_1 changes in parallel with the changes with the value of another property P_2, then hypothesize that P_1 and P_2 are related.

Table 2. An example of CER's classification. Here, the classifier is to choose an experiment material from three alternatives M1, M2 and M3, and chooses M2.

Qualities	weights	M1	M2	M3
availability/h	2	0	1	1
availability/m	1	1	0	0
availability/l	0	0	0	0
cost/h	-1	0	0	0
cost/m	0	1	0	0
cost/l	1	0	1	1
success hope/h	2	0	1	0
success hope/m	1	0	0	1
success hope/l	0	1	0	0
hazards/h	-3	0	0	0
hazards/m	-1	0	0	0
hazards/l	0	1	1	1
S		1	5	4

The results of experiments are generalized into hypotheses in stages. For example, if the experiment result is: "Substitution of *Al* for *Ni* in $LaNiO_3$ did not improve conductivity," then this is generalized in an increasing degree of abstraction stages as follows:

a) Substitution of *Al* for *Ni* in $LaNiO_3$ does not improve conductivity.
b) Substitution of *Al* for *Ni* in oxide superconductors does not improve conductivity.
c) Substitution of *Al* in oxide superconductors does not improve conductivity.

If, in a later experiment, say, the result is: "Substitution of *Al* for *Bi* in *Ba-Pb-Bi-O* improves consuctivity," then the hypothesis (d) is deleted. This example contains multiple levels of abstraction (Darden, 1987).

2.4 Symbol Level Learning: CER's Classifier

As has been mentioned, CER uses only one classifier (Nilsson, 1965; Hunt, 1975) for several different classification tasks.[1] In experimental scientific research, preference is normally given to materials that are less costly, more easily available, more likely to yield success, and less hazardous. It is easy to see that these criteria can be in conflict with one another. For example, a particular material can be cheap, but highly toxic; another material can be easily available, but less likely to yield successful results in the proposed experiments. In such cases, the problem is to find the best material against a set of conflicting parameters. Purely rule based methods cannot resolve such problems efficiently, unless some supplementary methods are used to reduce the number rules, for, as the number of classification parameters increase, the number of rules required for classification can increase exponentially. Whereas a linear classifier can pack *n!* sets of rules in a vector of *n* parameter ranges. Additionally, classifiers can provide approximate solutions with incomplete data.

CER's classifier runs as a module, like the explanation based generalization program described earlier. It uses different evaluation matrices for different classification tasks such as choosing goals, strategies, methods, processes and experiment materials from

[1] The classifier is a short program (about 4K), also implemented in Prolog.

among alternatives. These matrices can be created, developed and modified on line without impairing the system's activities, and can be saved for future use. The classifier can be trained in two different ways: 1) by directly providing qualitative values to each parameter (e.g. cost, availability, relative hazards, etc.), 2) by learning from failure, in which case, the parameter values given to the correct object are used as increments in modifying the matrix.

To illustrate how the classifier can be trained directly, consider that we want to train it for choosing an experiment material. When the classifier is fired by the rule, a list of experiment materials, say *M1*, *M2* and *M3* is given as objects. Similarly, the classification parameters are entered. These can be cost, availability, likeliness to succeed, and relative hazards (see, Table 2). Each parameter can be further divided into qualitative ranges such as "high", "medium" and "low". For each object, parameter values are also entered. The values given to each parameter is related with its desirability. For example, high cost is not desirable, and therefore can be given a negative integer value such as -1 in an interval of -3 to +3. ~~Similarly, medium and low costs can be given the values of 0 and 1 respectively. This direct~~ way of training the evaluation matrix relies heavily on domain knowledge.

Once an evaluation matrix is created for a particular classification task, it can be used in similar classifications. To explain how the classification works, let us return to Table 2. When the list of experiment materials is known to the classifier, the qualitative values to availability, cost, likeliness to succeed, and relative hazards are provided to the classifier for each material. These values are multiplied with the corresponding values in the matrix (column 2 in the table) for each experiment material. The experiment material that has the highest weighted sum is reported by the classifier as the choice (which is M2 in the table).

The classifier asks the teacher if its choice is the correct one, and if the answer is affirmative, it is recorded as the correct choice. However, if the teacher is not happy with the choice, the second type of learning, i.e. learning by failure takes effect. The teacher enters the correct choice, and the values given to the parameters are added to the values in the matrix to increase the weights of the choice. In this way, with its modified matrix, the classifier favours the trainer's bias in the later classification activities. CER's classifier can build its evaluation matrices entirely by this second type of learning. In other words, learning by failure is applicable to it even when the evaluation matrix is blank. However, in this method learning is incremental and therefore slow.

3 The System's General Behaviour

Initially CER has the formal, theoretical, meta and historical knowledge from the earlier work on superconductivity, some examples of which were given above. CER's initial knowledge also includes detailed formal, theoretical and factual knowledge about all the chemical elements in the periodic table, which include more than forty properties for each element. The system's behavior in its research activities depends on the control knowledge that its control operator has acquired through training. What follows is the general description of the system's behavior after such a training session.

In its first run, the program generates a set of research goals from the records about unexplained phenomena, important physical properties and contradicting hypotheses in its knowledge base. As illustrated in Figure 2, CER generates basically two kinds of research goals: finding an explanation to a phenomenon, and studying a phenomenon. Once it has formulated its research goals, CER assigns time limits to them. Its next task is to choose a research goal to focus on. When the research goal is selected, a research framework has to be drawn, so that the system can focus on the relevant aspects of the research problem, by recalling the relevant information about the objects, properties and relations from its static knowledge base. Then, the system proposes research strategies, and selects a strategy from the general strategies that it knows.

CER knows two general strategies for explaining a phenomenon (gathering knowledge from external sources and theoretical analysis) and two strategies for studying a phenomenon (theoretical analysis and experimentation and/or observation). For each strategy there may be a number of alternative processes and techniques. For example, in studying a physical phenomenon, new materials with certain properties may need to be synthesized, which may require the application of certain processes.

In order to study a phenomenon, CER proposes strategies, and chooses a strategy. If the strategy is experimentation, then it proposes processes for experimentation, and chooses a process. Then it finds the appropriate materials for the experiments, from which the system chooses the best experiment materials. After the experiment materials are determined, experiments are designed, expectations are stated, and then experiments are conducted. The experiment results are compared with the current goal, and if the goal has not been achieved, alternative materials, processes, and strategies are tried. The research continues until all goals are achieved, or their time limits expire.

CER's search in pursuing its goals can be viewed as a combination of heuristic search and best-first search. Heuristic search is employed by the activities of the system's methodological rules. The program's search in choosing goals, strategies, methods, etc., can be viewed as best-first search, for its behavior in these cases is dependent on the weights given to the alternative choices. CER's control message list functions like a constantly changing agenda, by means of which the system's control operator directs its activities.

4 Evaluation of CER's Methods

In this section, we will take a critical look at the system's organisation of descriptive knowledge, its research operators, learning methods, and control architecture.

4.1 Knowledge Organisation

Current discovery systems employ either a frame representation, a rule based or predicate logic representation, or some combination of these. CER's representation of descriptive knowledge in categorized predicate statements is different from the methods employed by other researchers. CER also uses qualitative schemas (Forbus, 1984) for representing processes. This enables the system to reason about processes in an efficient way. To illustrate, consider the theoretical statement taken from CER's knowledge base,

```
theoretical(reduces,process([substitution/of,tve,y,ybco],_),tc).
```

New hypotheses from such theoretical statements can be generated by perturbation over the arguments before any direct reference to the process descriptions themselves. The categorization also facilitates the integration of methodological and control knowledge with the system's descriptive and definitive knowledge (see, Kocabas, 1992).

4.2 System Operators

CER's design aims at modeling various tasks carried out in scientific research. Its research operators cover a wide range of research activities, in which they can serve as a starting point for the design of such models. CER's operators contain some of the methodological rules of problem generators, problem choosers, strategy proposers and decision makers of Kulkarni and Simon's (1988) KEKADA with minor modifications or improvements.

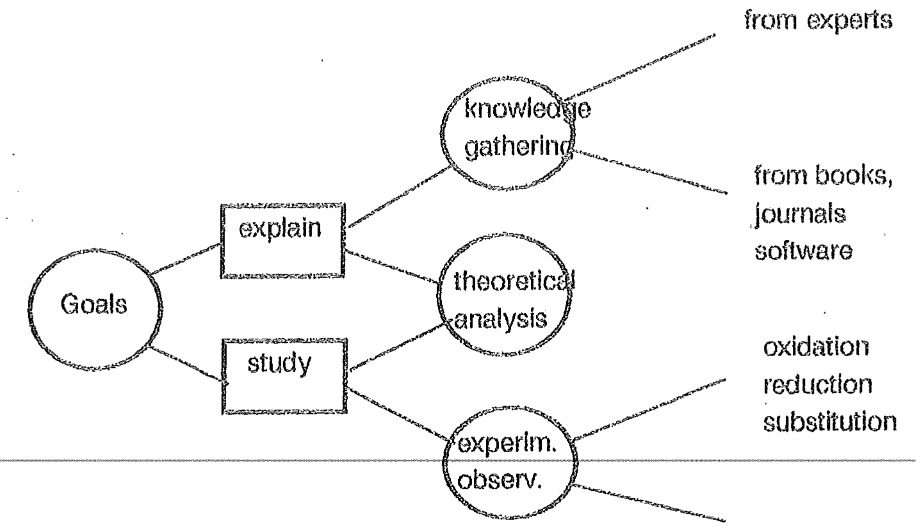


Figure 2. General structure of CER's behaviour.

However, an important difference between the two system's is the former's use of a classifier, while the latter relies on a series of rules for choosing processes and experiment materials. Classifiers provide three main advantages over the rule based methods in such classification activities: 1) They are more flexible, for it is much easier to dynamically modify or change the evaluation matrix of a classifier than modifying or changing methodological rules, 2) They can pack more information per unit of computational space, and 3) They can be trained, while rules cannot.

4.3 Methods of Learning and Discovery

CER conducts search at three different levels: 1) search in directing tasks through the control rules of its control operator, 2) knowledge level search by the action rules of its level-1 operators, and 3) symbol level search by its classifier. In this, the system integrates the methods of learning at three different levels, and also provides an example to knowledge level control of symbol level systems at the same time, in the way its action rules control the system's classifier. Currently, unlike IDS(Nordhausen & Langley, 1987), CER does not have device level learning units. The system's methods of learning includes learning by experimentation, by generalizations (of its hypotheses), explanation based learning (of its control rules), and learning by classification. The program's generalizations of its experiment results and hypotheses consist of replacing individual objects names (e.g., of an element) with class names that appear as arguments in the expressions.

4.4 Control Knowledge and its Organisation

CER's control architecture has some similarities with blackboard control architecture (Hayes-Roth, 1985; Engelmore & Morgan, 1988), particularly in that its message list can be seen as functioning like a blackboard. However, there are some basic differences. First, the former's knowledge organisation is entirely different from those implemented in the blackboard systems. Second, the latter systems use a global blackboard, while CER uses its message list only as a medium for communication between its operators, and can also use a multiplicity of such message lists in a hierarchic homuncular organisation (Kocabas, 1991b). Third, the system's operators only represent methodological knowledge as opposed to the "knowledge sources" of the blackboard systems, which may also represent descriptive and definitive knowledge. Finally, blackboard systems use sophisticated schedulers to determine which configuration of the blackboard should cause the activation of which knowledge sources, whereas CER can learn its control knowledge.

Several other learning systems such as LEX (Mithchell, Keller, & Kedar-Cabelli, 1986), PET (Porter & Kibler, 1986), and PRODIGY (Minton & Carbonell, 1987) also have the capability of learning control knowledge by explanation based methods. Among these, PET has the additional capability of learning control in sequences of activities or "episodes". However, these systems operate in relatively narrow domains. They do not employ message lists as CER does, but use the states of their dynamic memory as the problem states. Consequently, they are "flat" systems as opposed to CER's essentially hierarchic control architecture.

A significant point about the organisation and control of CER's methodological knowledge is that it provides a prototypical model of how scientific research and discovery can be taught. According to this model, methodological knowledge (in the form of action rules) can be taught by instruction, while the control and coordination knowledge, by explanation based methods. CER's control operator can also be trained by other methods such as classification besides explanation based generalization. However, for this task, the classification requires a large number of training examples, and therefore, is cumbersome.

5 Conclusions

In this paper we have described how in a computational model of discovery different methods of learning can be integrated, through a program, CER, which constitutes a step towards a comprehensive model of scientific research. The program learns and conducts search at three levels: symbol level, knowledge level and control level. The symbol level search is carried out by by the program's classifier, while knowledge level search is conducted by its rules of action, and control level search by the system's control operator. CER's knowledge organisation facilitates the integration of search at different levels, and the use of such different methods of learning as classification, abstraction, and explanation based learning.

Modern scientific research is a complex enterprise usually requiring a large number of small but necessary inventions and discoveries of tools, techniques and subsidiary hypotheses before its main goals and strategies are accomplished. Even the design of an experiment requires a great deal of background knowledge about the methods, materials, processes, experimental tools and their proper arrangement, and about the conditions of measurement. Therefore, a few hundred rules in a computational system can only provide a sketchy model for scientific research and discovery. Nevertheless, computational modeling of discovery provides us the insights on the reasoning behind the critical decisions taken by the scientists during the course of their research.

The system's hierarchic control architecture introduces a new perspective into the design of computational models of research and discovery: A system can learn its control knowledge for distributing its tasks over its subsystems by explanation based learning. We have described how this method has been applied to train CER's control operator. This control architecture can be extended downwards to the system's action rules, so that each operator can learn which rule to activate next, adding more flexibility into the representation of methodological and control knowledge. Another advantage of the hierarchic organisation of such operators is to enable the system to make maximum possible use of parallelism.

References

Darden, L. (1987). Viewing the history of science as compiled hindsight. *The AI Magazine 8, 2, 33-42.* Dennett, D.C. (1978). *Brainstorms: Philosophical essays on mind and psychology.* Sussex: The Harvester Press.

Dietterich, T. and Michalski, R.S. (1983). A comparative review of selected methods for learning from examples. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell (eds.) *Machine Learning: An artificial intelligence approach,* Morgan Kaufmann.

Engelmore, R., & Morgan, T. (Eds.) (1988). *Blackboard systems.* Addison Wesley.

Forbus, K. (1984). Qualitative process theory. *Artificial Intelligence, 24, 85-168.*

Hayes-Roth, B. (1985). A blackboard architecture for control. *Artificial Intelligence, 26, 251-321.*

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems.* University of Michigan Press.

Hunt, E.B. (1975). *Artificial Intelligence.* New York: Academic Press.

Kocabas, S. (1989a). *Functional categorization of knowledge: Applications in modeling scientific research and discovery.* PhD Thesis, Department of Electronic and Electrical Engineering, King's College London, University of London.

Kocabas, S. (1989b). Qualitative and quantitative reasoning in particle physics. *Proceedings of the 12th International Congress on Cybernetics, 250-271,* Namur, Belgium.

Kocabas, S. (1991a). Conflict resolution as discovery in particle physics. *Machine Learning, Vol 6, No 3, (277-309).*

Kocabas, S. (1991b). Homuncular learning and rule parallelism: An application to BACON. *Proceedings of the International Conference on Control,* IEE Publications, 950-954.

Kocabas, S. (1991c). A Review of Learning. *The Knowledge Engineering Review, 6. 3.*

Langley, P., Simon, H, Bradshaw, G. and Zytkow, J. (1987). *Scientific discovery: Exploration of the creative processes.* MIT Press.

Kulkarni, D. & Simon, H. (1988). The processes of scientific discovery. *Cognitive Science, 12, 139-175.*

Lenat, D.B. (1979). On automated scientific theory formation: A case study using the AM program. In: Hayes, J., Mitchie, D., Mikulic, L.I. eds., *Machine Intelligence,* 9, 251-283, Halstead, New York.

Lenat, D.B. (1983). EURISKO: A program that learns new heuristics and domain concepts. *Artificial Intelligence 21, (1-2), 61-89.*

Lenat, D.B. & Feigenbaum, E.A. (1987). On the thresholds of knowledge. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence, (1173-1182).*

Minton, S. & Carbonell, J.G. Strategies for learning search control rules. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence, 228-235.*

Mitchell, T. Keller, R.M., and Kedar-Cabelli, S.T. (1986). Explanation-based generalization: A unifying view. *Machine Learning 1, (1) 47-80.*

Newell, A. (1982). The knowledge level. *Artificial Intelligence, 18, 1, 87-127.*

Nilsson, N. (1965). *Learning Machines: Foundations of trainable pattern classifying systems.* McGraw Hill.

Nordhausen, B. & Langley, P. (1987). Towards an integrated discovery system. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence, 198-200.*

O'Rorke, P., Morris, S., & Schulenburg, D. (1990). Theory formation by abduction: A case study based on the chemical revolution. In J. Shrager and P. Langley (eds.), *Computational models of scientific discovery and theory formation.* Morgan Kaufmann.

Porter, B.W. and Kibler, D.F. (1986). Experimental goal regression: A method for learning problem-solving heuristics, *Machine Learning, Vol 1, No. 3, (249-286).*

Konuşma Tanıma

Speech Recognition

KONUŞMA İŞARETİNİN TANINMASI İÇİN KULLANILAN YAPAY NÖRON AĞLARI VE DİĞER TEKNİKLER

Fikret Gürgen ve Filiz Güneş
Boğaziçi Üniversitesi ve Yıldız Üniversitesi
İstanbul

ABSTRACT

Neural Networks (NN) have recently attracted particular attention from scientists and engineers. Various aspects of NN have been studied and the applications of them are searched for some problems such as robotics, navigation, speech recognition... Speech recognition has become one of the very successful application of NN.

This study gives an overview of speech recognition NN. It introduces some recent results achieved by authors and compares these with the results of other techniques for speech recognition (specifically Hidden Markov Model (HMM)). The paper specifically mentions Time-Delay Neural Network (TDNN) and successor architectures with training algorithms using the recognition performances of a subset of phonemes of Japanese.

ABSTRACT

Yapay Nöron Ağları (YNA) son zamanlarda bilim adamları ve mühendisler tarafından büyük ilgi görmüştür. YNA ile ilgili çeşitli konular bilim adamları tarafından çalışılmış ve robotlar, yöneltme, konuşma işaretinin tanınması gibi çeşitli uygulamalarda bu ağlar kullanılmıştır. Konuşma işaretinin tanınması YNA nın en başarılı olduğu uygulamalardan biridir.

Bu çalışma konuşma işaretinin tanınması için kullanılan YNA hakkında genel bir fikir vermeyi amaçlamıştır. Bu alanda bizim elde etmiş olduğumuz genel sonuçlardan yararlanarak diğer bilinen teknikler (örneğin Markov Modeli) ile YNA nin konuşma işareti için karşılaştırılmasını hedefledik. Bu çalışma özellikle zaman gecikmeli YNA ve arkasindan gelen nöron ağlarının ve eğitme algoritmalarının Japonca ses parçacıklarının bir altkümesi (/b,d,g,m,n,N/) kullanılarak yapılan sonuçlardan yararlanmıştır.

GİRİŞ

Günümüzde Yapay Nöron Ağları (YNA) konusunda çeşitli araştırmalar yapılmaktadır [7],[9]. Bu alanda yapılan çalışmalar herbiri değişik amaç ve değişik kriterleri olan kategorilerde yol almaktadir. Bunlar arasında en başarılı olan çalışmalardan biri konuşma işaretinin tanınmasi ile ilgili araştırmalardır.

Bu çalışmada konuşma işaretinin tanınması için kullanılan güncel teknikler arasında YNAnın yerini ve elde ettiğimiz (genel olarakta son zamanlarda elde edilen) sonuçları sunmayı amaçladık.

Konuşma işaretini bilgisayarlar ile tanınmasına dayanan araştırmalar son 20 yıldan beri gittikçe artan ilgi görmüştür [1]. Bugün az sayıda kelimenin tanınması için sistemler artık kullanılmaya başlanmıştır. Daha çok sayıda kelime (orta derecede büyüklükte kelime sayısı) için ise %100'e yakın tanınma oranı elde edilmiştir. Çok kelimeden oluşan (günlük hayatta kullanılan kelimelerin büyükçe bir kısmını kapsayan) sistemler için ise araştırmalar sürmektedir. Her geçen gün bilgi ve tecrübe birikimleri sonucu daha büyük adımlar atılmaktadır. Sonuçta, konuşma işaretinin bilgisayar ile tanınması dolayısıyla insan ve makina arasındaki iletişimin artması için yeni ufuklar açabilecektir.

Konuşma işaretinin tanınması için genel olarak üç temel teknikten bahsetmek mümkündür: Dinamik zaman ayarlaması (Dynamic Time Warping or DTW) [5], Markov Model (Hidden Markov Model or HMM) [6],[10], ve Yapay Nöron Ağları (YNA, Neural Networks) [7]. Dinamik Zaman Ayarlaması (DTW) ile konuşma işaretini oluşturan kalıpların (templates) birbirine uygunluğu araştırılır. Konuşma işaretini oluşturan vektörlerin (feature vectors) birbirine yakınlığını dinamik programlama ile araştıran bu teknik istatistiksel kapsamdan yoksundur, daha ziyade klasik anlamda şekil, form tanıma (pattern recognition) ve dinamik programlama (dynamic programming) tekniklerinin birleşimidir. Bugün ticari olarak geliştirilmiş pekçok sistemde ve araştırma laboratuarında bu tekniğin uygulamalarını görmek mümkündür. Sürekli (ağızdan çıktığı gibi) konuşma işaretinin tanınmasında (Connected speech recognition) diğer tekniklerin yanında temel bir yöntem olarak kullanılmaktadır. Bu yöntemin stokastik şekli "Viterbi algoritması" olarak bilinip, HMM in kullandığı temel yöntemlerden biridir.

HMM ve YNA tekniklerinden gelecek bölümlerde daha ayrıntılı bahsedilecektir. Son bölümde ise bu tekniklerle yapılan deneylerde elde edilmiş sonuçlar sunulacaktır.

MARKOV MODELİ (HMM)

Markov Model (HMM) olasılık teorisine dayalı günümüzde çok kullanılan kuvvetli bir tekniktir. Son 10 yılda bu tekniğin konuşma işareti tanınması problemine uygulanması için önemli çalışmalar yapılmış ve başarılı sonuçlar alınmıştır.

HMM son zamanlarda geniş olarak araştırılmış ve kullanılmış bir teniktir [10](Baker, 1975; Bahl ve Jelinek, 1975; Jelinek, 1976; Rabiner ve diğerleri, 1983; Rabiner ve Huang, 1986). Bir HMM aralarında geçiş elemanları (transitions) ile bağlanmış durumların (states) bir topluluğudur. Ses parçacıkları az sayıda durumun geçiş elemanı ile bağlanması ile oluşan modeller ile gösterilir. Her bir durumda bir çıkış işareti gözlemek mümkün olsada, durumun kendisini gözleyemeyiz. Bu nedenle "gizli, hidden" sözcüğü kullanılır.

Genel bir HMM, (A,B,□) büyuklükleri ile tanımlanır. A= (aij) ve B= (bj(k)) terimleri durum geçiş matrisi ve çıkış işareti gözlenme olasılığı olarak tanımlanır. □ ise başlangıç durum olasılıklarının kümesidir. "aij" i durumundan j durumuna geçiş olasılığı, "bj(k)" ise k inci ayrık ses elemanının (k inci vektör kod elemanının, VQ: vector quantization technique) gözlenme olasılığıdır. □i, i durumunun başlangıç durumu olması olasılığını gösterir.

HMM in eğitme fazında; örneğin, 100 ses parçacığı için

$O(n) = (O_t(n)),\quad t=1$ to T_n,

T_n : ses parçacığını oluşturan birim sayısı

gözlemleri kullanılıp, (A*,B*, □*) modeli aranır, öyle ki

$$(A^*,B^*,\square^*) = \underset{(A,B,\square)}{\text{argmax}} \ \underset{n=1}{\overset{100}{\square}} \ \text{Prob}(O(n)/\ A,B,\square).$$

Baum-Welch algoritması kullanılarak, yukardaki eşitliğin sağ tarafındaki şartlı olasilık maximize edilir. Viterbi algoritması kullanılarak en uygun durum dizisi veya ses parçacığını oluşturan birimlerin dizisi saptanır ($O(n) = O_1(n)\ O_2(n) \ldots\ O_T(n)$). HMM in bilinmeyen ses parçacıklarını tanıma aşamasında, sonrada forward algoritması ile herbir bilinmeyen ses parçacığının, bulunan modeller için, olasılıkları hesaplanır ve tanıma skoru elde edilir.

Sürekli HMM de, kodlanmış vektörler yerine orijinal vektörler (feature vectors of phonemes) kullanılmaktadır [10].

YAPAY NÖRON AĞLARI

YNA, HMM e göre daha yeni, üzerinde çalışılması gereken bir tekniktir. Alınan sonuçlar konuşma işaretinin tanınması konusunda YNAnin HMM yanında (veya birlikte) kullanılabilecek diğer bir başarılı teknik olduğunu kanıtlamıştır. Çalışmalar YNA nın konuşma işaretinin tanınmasına uygulanmasında karşılaşılan performans, dayanıklılık (robustness), modulerlik (modularity) gibi karşılaşılan belli başlı problemlerin çözümü yönünde devam etmektedir.

Bu çalışmada konuşma işaretinin tanınması için kullanılan YNA nın ses parçacıkları için elde ettikleri performans sonuçlarını sunmayı amaçladık. YNA birçok nonlineer işlem elemanının paralel olarak işlem gördüğü bir informasyon sistemidir. Temel olarak, N adet ağırlıklı giriş işareti nonlineer transfer fonksiyonlu (örn. sigmoid fonk.) işlem elemanından geçirilir. En başarılı olarak kullanılan YNA çeşidi çok katmanlı perseptron (multilayer perceptron, MLP) olup, çeşitli sınıflama ve form tanıma problemlerine başarı ile uygulanmaya başlamıştır [9]. Çok katmanlı perseptron (MLP) bir ileri-beslemeli devre olup, giriş ve çıkış katmanlarının arasında bir veye birden fazla katmanlar yer alır. Bilindiği gibi MLP un kapasitesi nonlineer işlem elemanlarından oluşmasından gelir, hatanIn geriye iletilmesi (back propagation, BP) gibi bir eğitme algoritması ile kullanılacağı uzayın çeşitli örnekleri ile eğitilmesi gerekir.

Hatanın geriye doğru iletilmesi (BP), YNA nın üzerinde işlem göreceği uzayın örnekleri ile eğitilip, ağırlık değerlerinin bulunmasını sağlayan bir iteratif eğitme algoritmasıdır. Daha önceki çalışmalarımızda BP algoritmasını önerdiğimiz "fuzzy eğitme" algoritması ile geliştirmiştik. Bu konudaki çalışmalarımız devam etmektedir.

Çalışmalarımızda MLP geliştirilerek önerilen zaman gecikmeli YNA (TDNN) [8] ve geliştirilmeye devam edilen yeni YNA nın sonuçlarını tartışmayı planladık [2,3,4]. Diğer tarafta surekli HMM (continuous HMM) ve çeşitleri bu karşılaştırma için temel alındı

SONUÇLAR

Bu çalışmada Japonca konuşma işaretini oluşturan 6 ses parçacığı olan /b,d,g,m,n,N/ (phoneme) sesleri (5240 kelimeden oluşan kelime dağarcığından bu sescikler toplanıp ve bu sescikler ile deneyler yapılarak) için elde edilen genel sonuçlar karşilaştırıldı. Ayrıca konuşma hızına bağlı olarak izole edilmiş kelimeler (isolated words), ibareler (phrases) ve sürekli konuşma (continuous speech) olarak elde edilen değerler gözönüne alındı.

HMM uzerinde uzunca bir süredir çalışılan, başarılı bir teknik olarak, gerçekten çığır açıcı performanslara ulaşmıştır. Öğrenme ve test kümelerinden sadece test kümesi üzerinde elde edilen sonuçlar (doğal olarak) karşılaştırmaya esas alınmıştır. Sürekli HMM ile bu sonuçlar: Izole kelimeler için %97-98, ibareler için %85-86 ve sürekli konuşma için %70 ler civarindadir.

YNAnin özellikle konuşma işaretinin tanınması için tasarılan şekli TDNN ve arkasından gelen YNA yapilari hatanın geriye iletilmesi (back propagation) algoritması ile aynı /b,d,g,m,n,N/ deneylerinde HMMlere yakın sonuçlar göstermeye başlamışlardır: Test kümesi sonuçları izole kelimelerde %97 lere, ibarelerde %85 lere yaklaşmıştır. Ama YNAnin

ISOLATED SPEECH RECOGNITION SYSTEM

Mustafa AKTEKİN[1], Ayşen DALOĞLU[2]
[1]TELETAŞ, Araştırma Geliştirme Bölümü Ümraniye/İstanbul, Türkiye
[2]YILDIZ Üniversitesi, Kocaeli Müh. Fakültesi İzmit /Kocaeli, Türkiye

ABSTRACT

In this work, an isolated speech recognition system designed for a digital exchange is introduced. The hardware structure of the system is given and the functions of the each module are explained. Software steps and the algorithms are given, the training phase of the system and to build up the library in the system memory are discussed.

1. INTRODUCTION

Nowadays, digital exchanges are getting to be more powerful and more functional to serve the user. Basic function of a switch is setting up a proper speech communication channel. If we want to implement an operator function as an automated service in the system, we require a speech recognition process anyway.

Telephone services covers some manually operated services by an operator such as alternate billed calls. Alternate Billed Calls (ABC) refer to those calls billed to a number other than that of the originating telephone. This includes to collect calling information from the caller, to make a conversation with called party about the call acceptance, and to proceed the call control under called party wish. In essence, Alternate Calls service has been implemented by TELETAŞ as an Automated Alternate Billing System (AABS). In essence, AABS simulates the customer interaction that would normally be performed by an operator.

AABS has the following main functional steps;

1-Calling party dials a special service calls to initiate AABS service, a special announce, instruct to the calling party to dial the called party phone number.

2-AABS service, establish direct connection between "AABS and called party.

3-After off hook of the called party phone AABS generate a synthetic speech to explain the calling party phone number and service usage procedure.

4-Speech recognition part of the AABS recognize the called party answer and activate a proper AABS function for remaining actions.

5-If, the called party accepts the call, a direct connections established between the calling and called party, but charging information is transferred to the called site.

If, the called party refuse the call, calling party receive an announce about the refusal.

This paper concerned about the speech recognition system employed in the TELETAŞ's Automated Alternate Billing System. The system, basically depends on the isolated speech recognition process.

2. Implemented Speech Recognition System

Any speech recognition system can be implemented on the following functional steps;

-Speech segmentation and parameterisation
-Speech segment identification and recognition
-Word identification

Normally, word recognition require speech segment recognition, and word recognition processes on the sequence. Although, any recognition process requires two logical processes.

a- Training Process
b- Recognition Process

Training process may help us to set up two reference library for the recognition system.

Turkish language speech segments library
Turkish language word definition library

The speech segment library may contain all of the basic Turkish Language speech segments with our parameter set on the vector quantized form. Any extracted segment parameters can be identified by using existing segment library.

Word definition library contains relevant speech segment set to define each word .

3 . THE HARDWARE STRUCTURE OF THE SYSTEM

In the design of this speech recognition system, two microprocessors are used. One of them is a digital signal processor as TMS320C15, that realizes the feature analysis process. The other microprocessor, Intel 80C186 is the master controller of this system. 80C186 takes speech samples from PCM channel via a serial port (8920) and writes them to RAM. The data stored in RAM is converted to packets, each containing 80 speech samples. 80C186 writes the packets to a dual port RAM and the digital signal processor reads them. Then, signal processor makes parameter extraction process from the samples. All speech data are sampled at 8 KHz, PCM frequency and analysed for 10 ms window length . Digital signal processor, makes windowing and computes the auto correlation coefficients, the linear predictive coding coefficients (LPC), zero crossing counts. Then, determine the formant frequencies from the LPC coefficients.

Digital signal processor writes these analysis parameters to the dual port RAM. The 80C186 reads the packet of parameters from the dual port RAM, quantizes them by using vector quantization algorithm. The evaluation of these parameters for segmentation and making decisions are realized on the 80C186 part of the system. The hardware structure of the system is illustrated in Fig. 1

4 . THE ANALYSIS PROCESS OF THE SPEECH

The software which is implemented on the digital signal processor, makes the feature analysis process. The feature analysis, distills the information necessary for speech recognition from the raw speech waveform. Just as important, it discards information such as; background noise, channel distortion, speaker characteristics and manner of speaking.

Commonly used feature set for recognition is the LPC (Linear Predictive Coding) based feature set. The basic idea behind linear predictive coding is that a given speech sample can be approximated as a linear combination of the past speech samples. By minimizing the sum of the squared differences (over a finite interval) between the actual samples and the linearly predicted ones, a unique set of predictor coefficients can be determined. Linear predictive coding can be readily shown to be closely related to the basic

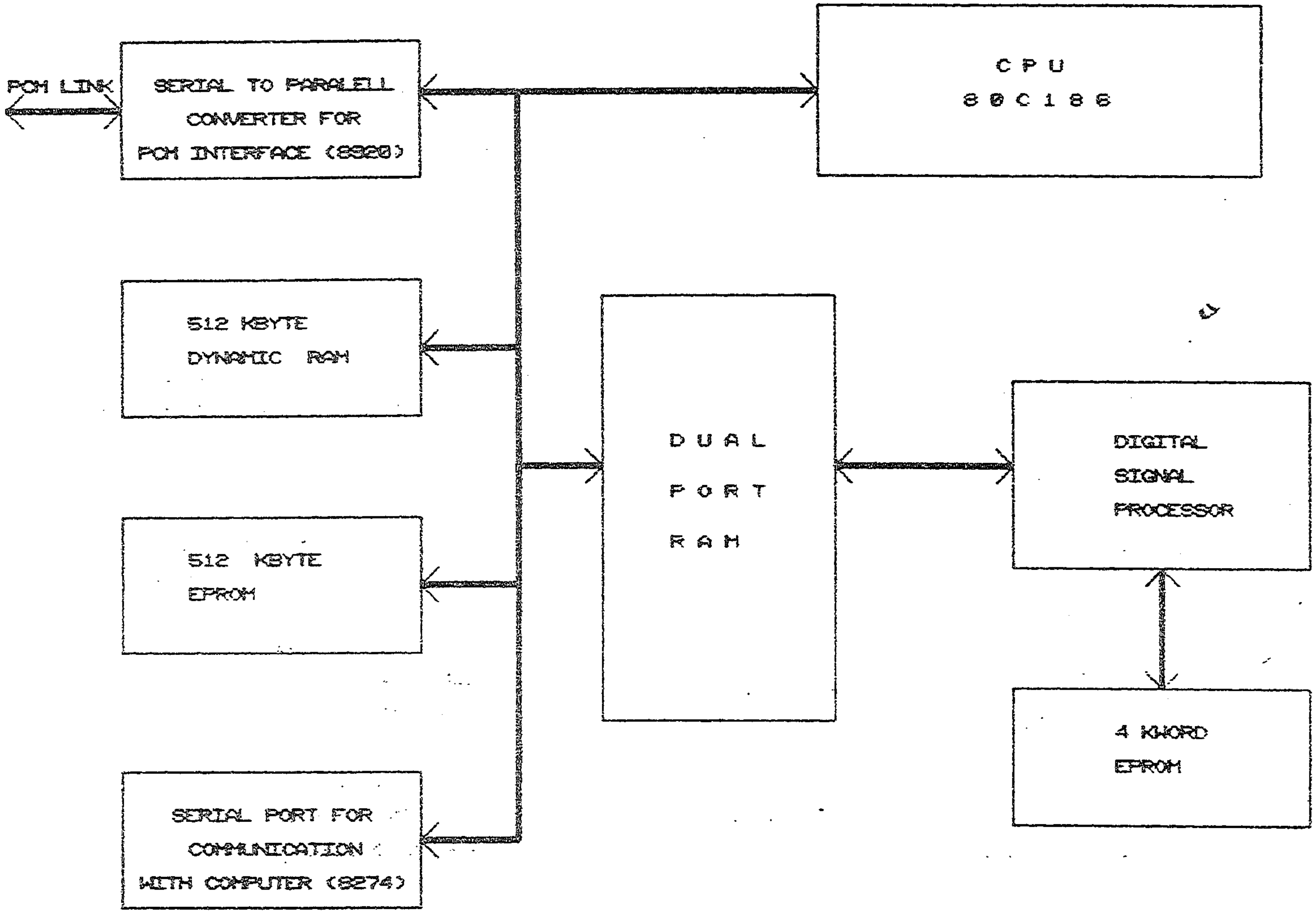


FIGURE 1. THE HARDWARE BLOCK DIAGRAM OF THE SYSTEM

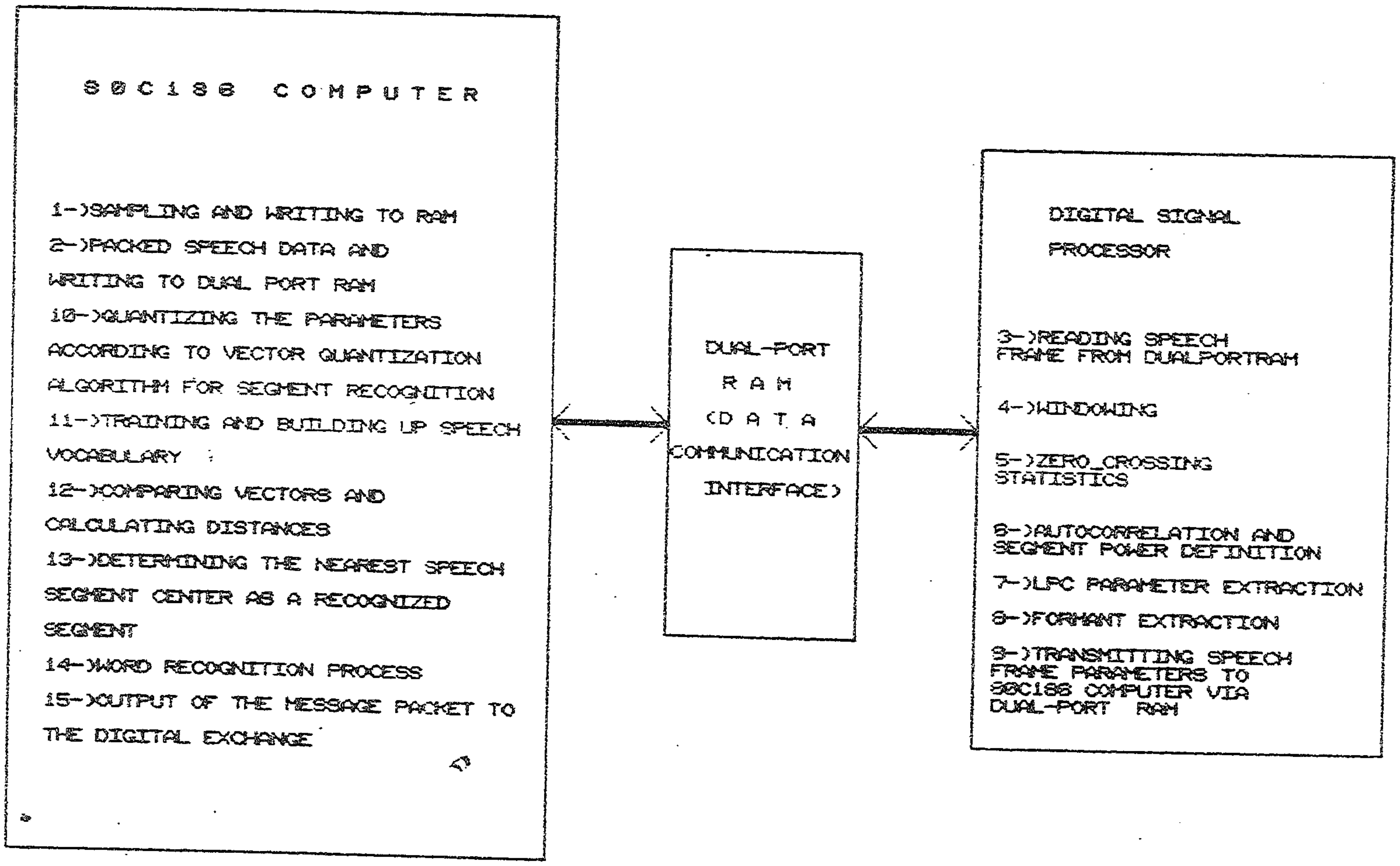

80C186 COMPUTER
1-)SAMPLING AND WRITING TO RAM
2-)PACKED SPEECH DATA AND WRITING TO DUAL PORT RAM
10-)QUANTIZING THE PARAMETERS ACCORDING TO VECTOR QUANTIZATION ALGORITHM FOR SEGMENT RECOGNITION
11-)TRAINING AND BUILDING UP SPEECH VOCABULARY
12-)COMPARING VECTORS AND CALCULATING DISTANCES
13-)DETERMINING THE NEAREST SPEECH SEGMENT CENTER AS A RECOGNIZED SEGMENT
14-)WORD RECOGNITION PROCESS
15-)OUTPUT OF THE MESSAGE PACKET TO THE DIGITAL EXCHANGE
DUAL-PORT RAM (DATA COMMUNICATION INTERFACE)
DIGITAL SIGNAL PROCESSOR
3-)READING SPEECH FRAME FROM DUALPORTRAM
4-)WINDOWING
5-)ZERO_CROSSING STATISTICS
6-)AUTOCORRELATION AND SEGMENT POWER DEFINITION
7-)LPC PARAMETER EXTRACTION
8-)FORMANT EXTRACTION
9-)TRANSMITTING SPEECH FRAME PARAMETERS TO 80C186 COMPUTER VIA DUAL-PORT RAM

FIGURE 2. THE FUNCTIONAL BLOCK DIAGRAM OF THE SYSTEM

model of human speech production in which the speech signal S_l modelled as the output of a linear, time varying system excited by either quasiperiodic pulses (for voiced sounds) or random noise (for unvoiced sounds). The linear predictive coding method provides a robust, reliable, and accurate method for estimating the parameters that characterize the linear, time - varying system.

LPC based feature analysis system is a block processing model in which a frame of N samples of speech is processed, and F feature vector is measured.

To obtain F vector, the speech signal is first pre-emphasized using a fixed first order digital system with transfer function;

$$H(z) = 1 - a.z^{-1}, \ a = 0.95 \qquad (1)$$

giving the signal,

$$\tilde{s}(n) = s(n) - a.s(n-1) \qquad (2)$$

The signal is next blocked into l sample sections (frames) for feature measurement. In order to minimize the effects of the short time duration the analyzing of the speech waveform, a smoothing window, w(n) is applied to the data packet to taper the speech samples to zero at the end of the frame, giving the windowed signal,

$$\tilde{X}_l(n) = X_l(n) \ . \ w(n) \qquad (3)$$

A typical smoothing window, used in LPC analysis systems is the Hamming window defined as,

$$w(n) = 0.54 - 0.46 \ \text{Cos}\left(\frac{2\pi n}{N-1}\right) \qquad (4)$$

The next step in the feature analysis software is to perform an autocorrelation analysis of the windowed frame of data, giving.

$$R_l(m) = \sum_{n=0}^{N-1-|m|} \tilde{X}_l(n). \tilde{X}_l(n+m) \ ; \ m = 0,1,..p \qquad (5)$$

Where, p is the order of the analysis system. The set of the equations can be expressed in a matrix form as,

$$\begin{bmatrix} R_l(0) & R_l(1) & R_l(2) & R_l(p-1) \\ R_l(1) & R_l(0) & R_l(1) & R_l(p-2) \\ R_l(2) & R_l(1) & R_l(0) & R_l(p-3) \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ R_l(p-1) & R_l(p-2) & R_l(p-3) & R_l(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ . \\ . \\ . \\ a_p \end{bmatrix} = \begin{bmatrix} R_l(1) \\ R_l(2) \\ R_l(3) \\ . \\ . \\ . \\ R_l(p) \end{bmatrix} \qquad (6)$$

By taking advantage of the Toeplitz nature of the coefficients matrix (6), several efficient recursive procedures have been devised for solving this system of equations. In the design of our feature extraction software, to obtain LPC parameters ($a_j = a^{(p)}_j$, $1 \leq j \leq p$), Durbin's recursive solution is used.

Linear predictive analysis of speech has several advantages when applied to the problem of estimating the formants for voiced sections of speech. Formants can be estimated from the linear prediction parameters. Given the LPC coefficients a_j, $j = 1,2, ..., p$ computing $A_k = \text{DFT} \{ 1, a_1 , a_2, ..., a_p , 0 ......0\}$ to obtain the discrete Fourier transform of the inverse filter, simple minimal picking on $|A_k|^2$ for each frame gives the raw data from which formant frequencies can be estimated.

Finally, in the parameter extraction process, the short time energy of the speech signal and zero crossing rate of the speech data are calculated by the digital signal processor to be used in decision making stage.

5 THE EVALUATION OF ANALYSIS PARAMETERS

Analysis parameters are coded by using vector quantization algorithm for the segment recognition process. In vector quantization, we need to determine the reconstruction levels r_i and corresponding cells C_i. A list of reconstruction levels is called a reconstruction codebook or a codebook. If there are L reconstruction levels in the list, the list is said to be an L - Level codebook. A codebook is normally generated by a training procedure which minimizes the average distortion resulting from coding a suitably long sequence of vectors. We suppose that we have M training vectors denoted by f_i for $1 \leq i \leq M$. Since, we estimate L reconstruction levels from M training vectors, in that calculation we assumed M >> L. The reconstruction levels r_i are determined by minimizing the average distortion, which is defined by,

$$D = \frac{1}{M} \sum_{i=1}^{M} d(f_i, \hat{f}_i) \qquad (7)$$

The algorithm steps can be summarized such as,

1) We begin with an initial estimate of r_i for $1 \leq i \leq L$.

2) We then classify the M training vectors into L different groups or clusters, corresponding to each reconstruction level using the equation;

$$VQ(f) = r_i \text{ , if and only if } d(f, r_i) \leq d(f, r_j),\ j \neq i,\ 1 < j < L \qquad (8)$$

This can be done by comparing a training vector with each of the reconstruction levels and choosing the level that results in the smallest distortion.

3) A new reconstruction level is determined from the vectors in each cluster. Let us suppose, f_1 for $1 \leq i \leq M_1$ are M_1 training vectors quantized to the first reconstruction level r_1. The new estimate of r_1 is obtained by minimizing ;

$$\sum_{i=1}^{M1} d(f_i, r_1) / M_1 \qquad (9)$$

4) A new estimate of all other reconstruction levels r_i for $2 \leq i \leq L$ is similarly obtained.

5) This iterative procedure can be stopped when the average distortion D does not change significantly between two consecutive iterations.

The basic idea, that by using vector quantization; it is possible to set up a separate codebook for each analyzed speech segment. During the training phase, each distinct speech segments are determined and quantized by a vector, called 'Speech Segment Center'.in the vocabulary. Segment recognition is then simply a question of which speech segment center in the segment library fits the analyzed segment vector. The vectors that are generated in a parameter extraction step, are compared with speech segment centers in the vocabulary, then their distance measures are calculated. In the making decision stage, the nearest speech segment center in the library is determined as a recognition segment. Then, extracted segment information is stored for word recognition process.

The following each other similar speech segments are given with the unique common identity. If, the segment identities exceed limits of the speech segment center, the forward alternation in voice signal is observed and directed new speech segment center is determined. The origin point of this alternation and the destination point are registered. The silence segments are deleted. But, spoken word is determined by using stored recognized speech segment centers in two consecutive silence segment as a distinct word.

In the word library, the conception which is correspond to speech segments is evaluated as the recognized word. The word causes a specific message flow, at the digital exchange to initiate a proper process.

6 TRAINING and BUILDING UP THE LIBRARY

Training system has similar hardware architecture with the recognition system. Training system has two phases ;

a-Speech segment library generation phase
b-Word library generation phase

6.1 Speech Segment Library Generation

Speech segment library requires manual training on the following sequence;

1) All of the vowel sounds of the speech are analyzed end classified by manually. The vocal tract maintains a relatively stable configuration during the production of Turkish vowel sounds. Turkish vowels are characterized by a negligible nasal coupling, and by radiating from the mouth. This feature gives us an opportunity to introduce vowel speech segment directly from the speech channel to the system. Each vowel is classified with a set of parameter called 'segment parameter set'. All of the test subject produce their own characteristic 'segment parameter set'. Gravity center of the total training results for each voiced sound is called 'segment parameter center'.

2)Consonant sounds are classified as 'fricative consonants', 'stop consonants', nasal consonants', 'glides and semivowels'. Consonant sounds speech centers are classified by using voiced sounds as being before or after the vowels. Training program use existing vowel speech segment information to determine the transition. This speech transition gives a similar transition over from one voiced speech segment center trough the consonant sound segment center, or vice versa. Each phoneme may cause to be produced a set of phonetical speech segment center by the system . Training system may capture the phonetic speech segment centers and the sound travailing route on the phoneme via two or more speech centers.

The speech segment parameter centers are analyzed and classified by manually for each phonemes. From the vowel sounds trough the longest phoneme in turkish language may create the personnel speech segment library. After using a number of test persons for training program, the system determine the boundaries for each speech segment center. This information is sufficient enough to build the 'speech segment library'.

6.2 Word Library Generation and Usage

Word library is a set of information about each word which is generated many test person's articulation. Training program stores speech segment center addresses sequentially for each word. Various persons can be used to determine the pronunciation difference for each word. This teaching process gives to the system general traveling trace via the speech centers for each word. TELETAŞ's system aims to recognize a distinct word, which means to take an action this way or the other way. Remaining process will be done on the high level system. After recognition of the word a special message packet will be prepared to activate the proper process .

7 CONCLUSION

This work aimed a practical purposes on our application. This process requires a limited number of recognition in the practice. But implementation has been done for a larger word dictionary. In the future work, the recognition system feature can be extended toward the sentence recognition.

8 REFERENCES

1. H. Abut, R.M.Gray, G. Rebolledo, "Vector Quantization of speech and Speech - Like Waveforms " IEEE Transactions on acoustics, speech, and signal processing, vol, ASSP - 30, June 82

2. Y.Linde , A.Buzo, and R.M Gray, "An Algorithm for Vector Quantizer Design ", IEEE trans. commun., vol. com-28, jan.1980, pp 84-95

3. L.R Rabiner and R.W Schafer , "Digital Processing of Speech Signals. Englewood " Cliffs, n.j Prentice _ Hall ,1978

4. J.D. Markel , A.H. Gray , H. Wakita , "Linear Prediction of Speech Theory and Practice" , Speech Communications Research Laboratory, Inc. september,1973.

5. L.R. Rabiner , S.E. Levinson , "Isolated and Connected Word Recognition."IEEE Transactions on Communication, Vol. Com.-29,5,
May 1981

VOCAL TRACT SHAPE ESTIMATION

M. Bilginer Gülmezoğlu[1] and Atalay Barkana[2]

[1,2] Department of Electrical and Electronics Engineering, Anadolu University, Eskişehir.

Abstract

The previous works done by other researchers to determine the cross_sectional areas of the vocal tract assumes that the glottal wave shape consists of only an impulse for each vibration of the vocal cords.

At the beginning of this work, it was expected that if the actual sawtooth input signal is taken as the glottal wave input, obtained cross_sectional areas of the vocal tract can be closer to actual ones. Input signals from the glottis and the output signals from the lips are measured and digitized simultaneously with the sampling frequency of 10kHz.

In this work, the transfer function of the vocal tract is obtained by using least_squares technique. The coefficients of the denominator polynomial of the transfer function are used to determine the cross_sectional areas of the vocal tract.

At first, the previous works which assume an impulse wave shape for the input is also repeated to find the cross_sectional areas of the vocal tract. It is observed that the obtained areas are similar to previous ones.

In the second step, the cross_sectional areas obtained by taking the actual input signal as the glottal wave input. The results of this work are compared with the work which takes the impulse signal as the glottal wave input.

The work with actual input signal seems to yield the position of the constriction point of the tongue better than the results of the previous work.

1.THE LEAST_SQUARES TECHNIQUE

The identification of systems with constant parameters which form the parameter vector is related to the measurements by linear matrix relation

$$\mathbf{y} = \mathbf{A}\mathbf{a} + \mathbf{v} \tag{1}$$

where $\mathbf{y}$ is an $N \times 1$ vector of measurements, $\mathbf{a}$ is a $p \times 1$ parameter vector to be estimated, $\mathbf{v}$ is an $N \times 1$ vector of noises or errors in the data taken, and $\mathbf{A}$ is an $N \times p$ matrix of data to be transformed by the model. Minimization of the sum of the squares of the errors yields[1],

$$\hat{\mathbf{a}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y} \tag{2}$$

where $\hat{\mathbf{a}}$ is the estimate of $\mathbf{a}$. The matrix $\mathbf{A}^T\mathbf{A}$ is $p \times p$ and must be inverted. If $p > N$, then the rank of $\mathbf{A}^T\mathbf{A}$ is less than p and it will be singular hence not invertible.

2. VOCAL TRACT AREA FUNCTION

The discrete_time transfer function of the vocal tract is known [2] as

$$G(z) = \frac{Y(z)}{U(z)} = \frac{1}{1+\alpha_1 z^{-1}+\ldots+\alpha_n z^{-n}} \tag{3}$$

where $Y(z)$ is the discrete_time output signal, $U(z)$ is the discrete_time input signal, n is the order of the denominator and α_i's are coefficients of denominator polynomial or linear prediction coefficients (LPC's).

The equivalent difference equation of the transfer function is

$$y(k) = -\alpha_1 y(k-1) - \alpha_2 y(k-2) - \ldots - \alpha_n y(k-n) + u(k) \tag{4}$$

$y(k)$ and $u(k)$ are assumed zero for negative indices k, so that the difference equation of (4) in matrix form is

$$\begin{pmatrix} y(1) \\ y(2) \\ \vdots \\ \\ \vdots \\ y(N) \end{pmatrix} = \begin{pmatrix} u(1) & -y(0) & 0 & \ldots & 0 \\ u(2) & -y(1) & -y(0) & \ldots & 0 \\ \vdots & \vdots & & & \vdots \\ & & & & 0 \\ & & & & -y(0) \\ \vdots & \vdots & & & \vdots \\ u(N) & -y(N-1) & -y(N-2) & \ldots & -y(N-n) \end{pmatrix} \begin{pmatrix} 1 \\ \alpha_1 \\ \vdots \\ \\ \vdots \\ \alpha_n \end{pmatrix} + \begin{pmatrix} v(1) \\ v(2) \\ \vdots \\ \\ \vdots \\ v(N) \end{pmatrix} \tag{5}$$

The expression (2) is applied to (5) to get the best α_i's

Reflection coefficients (k_m) can be obtained from the LPC's. Thus the computational expression [2] is

$$a_{m-1,i} = \frac{a_{m,i} - k_m a_{m,m-i}}{1-k_m{}^2} \tag{6}$$

with $k_m = a_{mm}$ for $m = M, M-1, \ldots, 1$ and $i = 0, 1, \ldots, m-1$ and $|k_m| < 1$. M is the number of sections from the lips.

The discrete area function of the estimated vocal tract shape are then computed from the reflection coefficients [2] as

$$A_{m-1} = \frac{1+k_m}{1-k_m} A_m \tag{7}$$

for $m = M, M-1, \ldots, 1$, keeping in mind that A_M is an artificial area. Having no absolute reference value, A_M is usually assumed to be unity.

The relation between sampling frequency (f_s), number of sections (M), length of the acoustic tube (L=Ml; l is section length) and speed of sound (c) is given [2] as

$$f_s = \frac{Mc}{2L} \tag{8}$$

The values of f_s and c are constant as 10kHz and 350m/sec respectively. Since L can be chosen as 17 cm, M is approximately taken as 10 for this work.

3. RESULTS AND COMPARISON

In this work, two periods of the glottal and speech signals are chosen as the interval of analysis. Preemphasis by a filter $1 - z^{-1}$ is applied to the speech wave output and Hamming window is applied to both glottal wave input and speech wave output. In the first step, unit impulse is taken as the glottal wave input. The cross_sectional areas of the vocal tract are determined for eight Turkish vowels. When the results of five Turkish vowels /a/, /e/, /i/, /u/ and /o/ are compared with those of Ishizaka Flanagan [3] model and those of Sadaoki Furui's [4] works, it is seen that the obtained results are globally similar to Ishizaka Flanagan and Furui's results. Differences at some points must be expected because Ishizaka Flanagan's results are obtained for phonetics from Russian language and Furui's results are obtained for phonetics from Japanese language. Since the areas vary in the case of strong voice, the results are normalized.

In the second step, measured and digitized values at the glottal wave from piezo crystal are used as an input in expression (5). It must be noted that the location of piezo crystal on the vocal cords is important in measuring the glottal wave shape. The cross_sectional areas of the vocal tract are also determined for eight Turkish vowels. From Figure 1, it can be seen that the obtained results are a lot closer to X_ray data [5]. Especially, the position of the constriction point of the tongue is more significant compared with the case of impulse input.

4. REFERENCES

1. Jacquot, R. G., Modern Digital Control Systems (Marcel Dekker Inc., New York and basel, 1981)
2. Markel, J.D. and Gray, A.H., Linear Prediction of Speech, (Springer Verlag, New York,1976).
3. Ishizaka, K. and Flanagan, J. L., Synthesis of Voiced Sounds from a Two-mass Model of the Vocal Cords, in Bell System Tech. J. 51, 1233-1286 (1972).
4. Furui, S. Digital Speech Processing, Synthesis and Recognition, (Marcel Dekker, Inc. New York and Basel, 1989).
5. Flanagan, J. L., Speech Analysis, Synthesis and Perception, (Springer Verlag, New York, 1972).

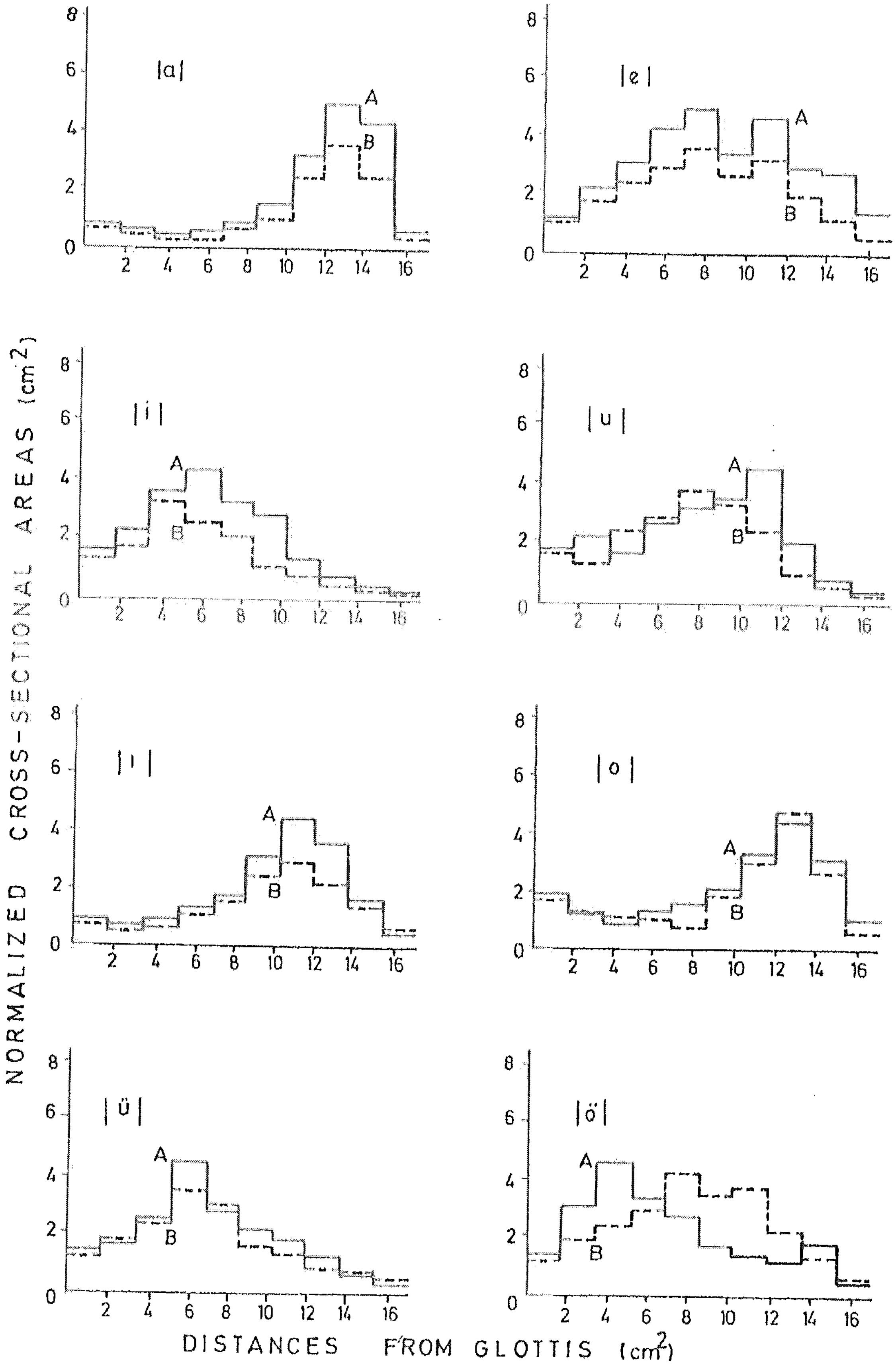


Figure 1. Estimated area functions: (A) impulse glottal wave; (B) actual glottal wave

COMPARISON OF HUMAN AND LOVEBIRD SPEECH

Osman Parlaktuna[1] and Atalay Barkana[2]

[1,2] Electrical and Electronics Engineering Department, Anadolu University, Eskişehir.

Abstract

It is well known that parrots and lovebirds imitate the human words if they are trained well. The speech production systems of parrots and lovebirds are different from humans'. For example, the length of the vocal tract of a human is almost equal to the whole length of a lovebird. Therefore, it is expected that the time signal of word produced by a lovebird would be different from the time signal of the same word produced by a human. Although, the time signals for the same word produced by a human and a lovebird are different, we still could understand the word produced by a lovebird. It is important to determine the parameters of lovebird speech which cause us to understand the words spoken by a lovebird. Knowing those parameters will help us in the field of speech recognition. In this study, the speech produced by a lovebird and by its trainer is analyzed in the time and frequency domain and those properties of the lovebirds' speech which are similar with the humans' are discovered.

INTRODUCTION

In this study the speech produced by a lovebird trained by a teenage girl is recorded. Name of the lovebird is "Dilosh" and this word is analyzed in the time and frequency domain, and compared with the same word spoken by the trainer.

The vocal tracts of the humans and lovebirds are very different from each other. The length of the vocal tract of a human is almost equal to the whole length of a lovebird. Singing birds have pepulus which vibrates as the human vocal cords. A vibrating trachea and a beak replaces the resonant frequency vibrating cavities of a human vocal tract. The schematic diagram of the vocal tract of a lovebird is shown in Figure 1 [1].

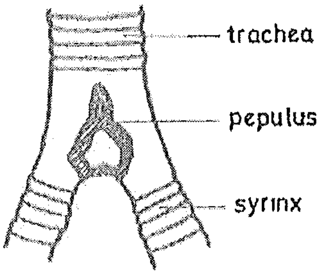


Figure 1. Schematic Diagram of the Vocal Tract of a Lovebird.

Time Analysis of the Word "Dilosh"

Since the speech production systems of a human and a lovebird are very different, it is expected that the word "Dilosh" produced by the lovebird and its trainer would be very different. This can be seen from Figures 2 and 3 which are the parts of the word "Dilosh" produced by human and lovebird, respectively. The periodic structure of the phonemes "d", "i", "l" and "o" of the human speech is not so obvious in the lovebird speech. The word produced by the lovebird has a shorter duration than the human word duration. But, average durations of each phoneme as the percentage of the whole word of lovebird and human speech are very close to each other as shown in Table 1 [2].

Table 1. The Average Durations of Phonemes

Phoneme	Human	Lovebird
D	0.050	0.055
I	0.250	0.210
L	0.076	0.060
O	0.320	0.250
Sh	0.270	0.345

Careful examination of "i" and "o" vowels of lovebird speech reveals that lovebird tries to imitate the pitch frequency of the human speech. The pitch frequency of the trainer is 350Hz on the average. Energy envelopes of the lovebird speech is almost periodic with the frequencies 280Hz and 350Hz for "i" and "o" vowels, respectively. The high frequency components are very effective in the lovebird speech which does not exist in the human speech.

Frequency Analysis of the Vowels of "Dilosh"

The 3-D spectrograms of the word "Dilosh" for the lovebird and its trainer are shown in Figures 4 and 5. In the bird speech, the high frequency components above 1500Hz seem to carry the whole information. The peak frequencies of the vowel "o" of the lovebird are between 1600Hz and 3500Hz. The formant frequencies of the vowel "o" of the trainer are 614, 1228 and 1848Hz. Although the lovebird speech seems to be the shifted version of human speech to the high frequencies, there is no one to one correspondence between high energy frequencies.

Modulation of the Vowels "i" and "o"

From the time and frequency domain graphics it seems that "i" and "o" vowels are modulated as DSBSC by the lovebird with carrier frequencies 2400Hz and 1600Hz, respectively. In order to generate those vowels, "di" and "losh" parts of the human speech are multiplied by sinusoidals of frequencies 2400Hz and 1600Hz, respectively [3]. To avoid overlaps human speech is filtered by a low pass filter with the cutoff frequency 700Hz. When the word is listened back a sound close to the bird speech is heard. 3-D spectrogram of this modulated wave is given in Figure 6.

CONCLUSION

Although the lovebird spends less time than the trainer to generate the word "Dilosh", duration ratio for each phoneme seems to be the same both for the lovebird and the trainer. The lovebird tries to imitate the pitch frequency by using energy envelopes. High energy components are highly effective in the lovebird speech. "i" and "o" vowels of the lovebird may be generated from the human vowels using DSBSC modulation technique. After examining other vowels generated by the lovebird, the lovebird speech may be synthesized and the synthesized words may be used to train the young lovebirds [3, 4].

REFERENCES

1. Öktay, Melekper, "Omurgalı Hayvanların Karşılaştırmalı Anatomisi," Istanbul, 1988.
2. Arslan, Beyhan, "Muhabbet Kuşu Konuşmasının Analizi," Bitirme Ödevi, Anadolu Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü, Eskişehir, 1991.
3. Kaya, Akile, "Muhabbet Kuşu Sesinin Sentezi," Bitirme Ödevi, Anadolu Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü, Eskişehir, 1992.
4. Hulse, Stewart H. et al., "An Integrative Approach to Auditory Perception by Songbirds," Comparative Perception, Volume II: Complex Signals, Edited by William C. Stebbins and Mark A. Berkley, John Wiley & Sons, 1990.

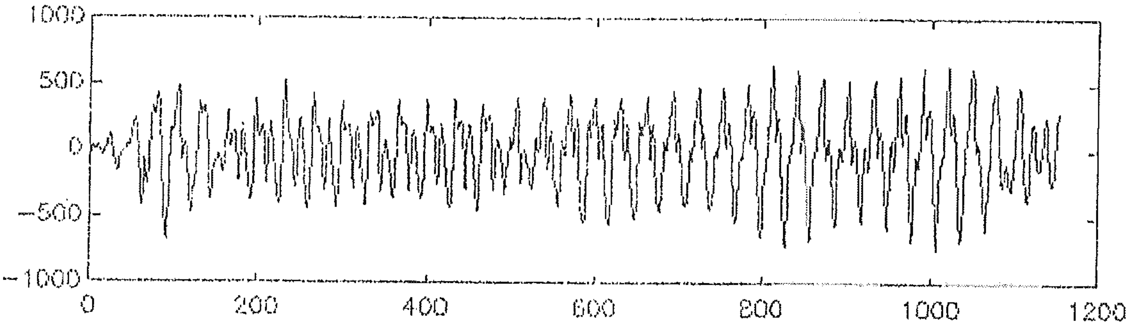


Figure 2. Sample of "Dilosh" Produced by Trainer.

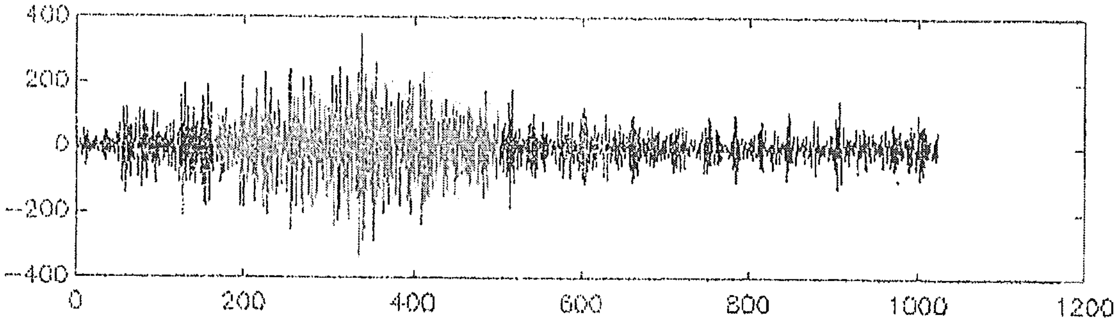


Figure 3. Sample of "Dilosh" Produced by Lovebird.

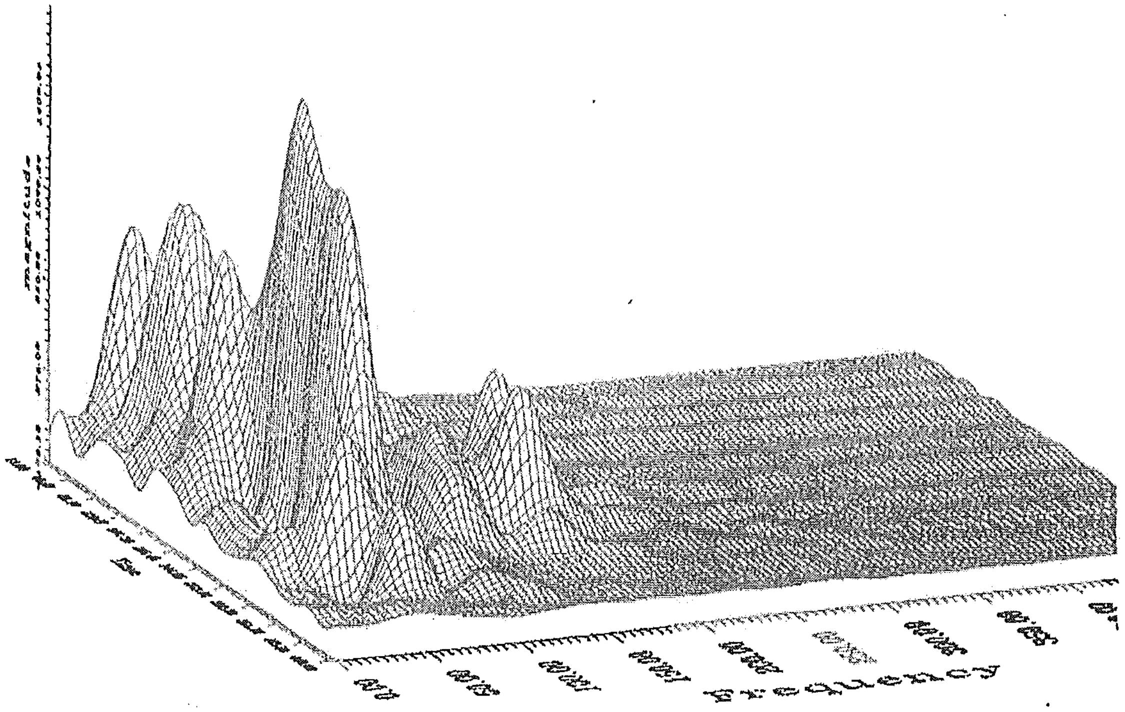


Figure 4. 3-D Spectrogram of "Dilosh" Produced by Trainer.

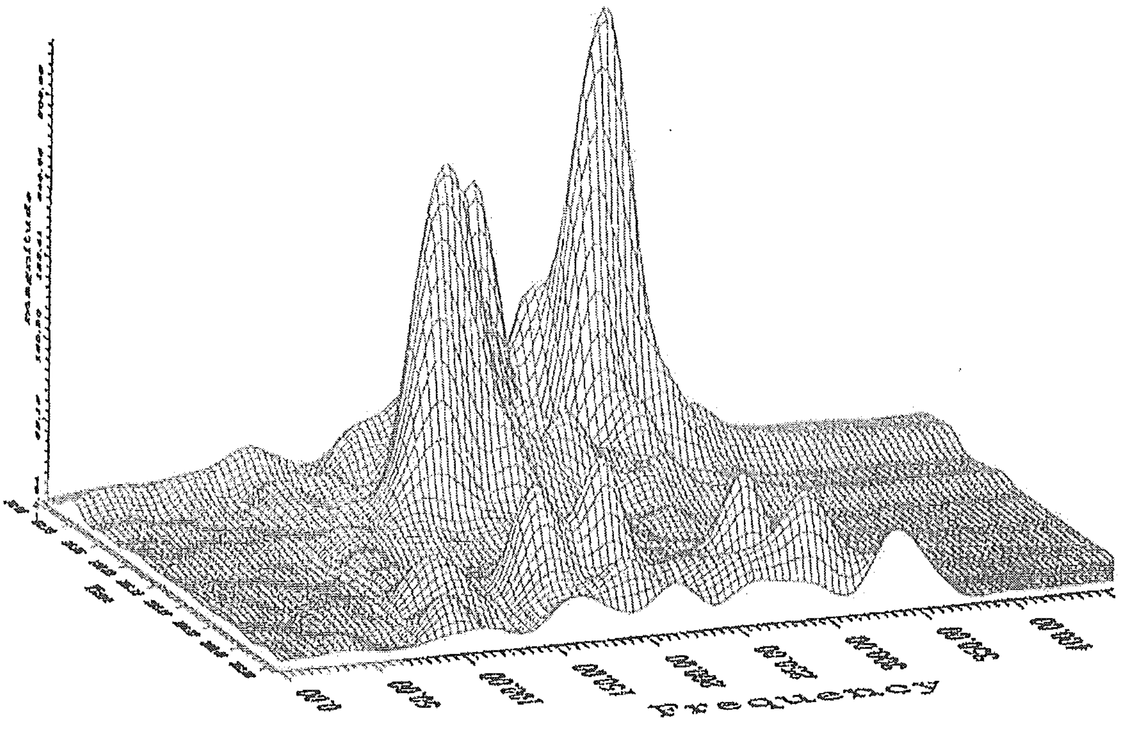


Figure 5. 3-D Spectrogram of "Dilosh" Produced by Lovebird.

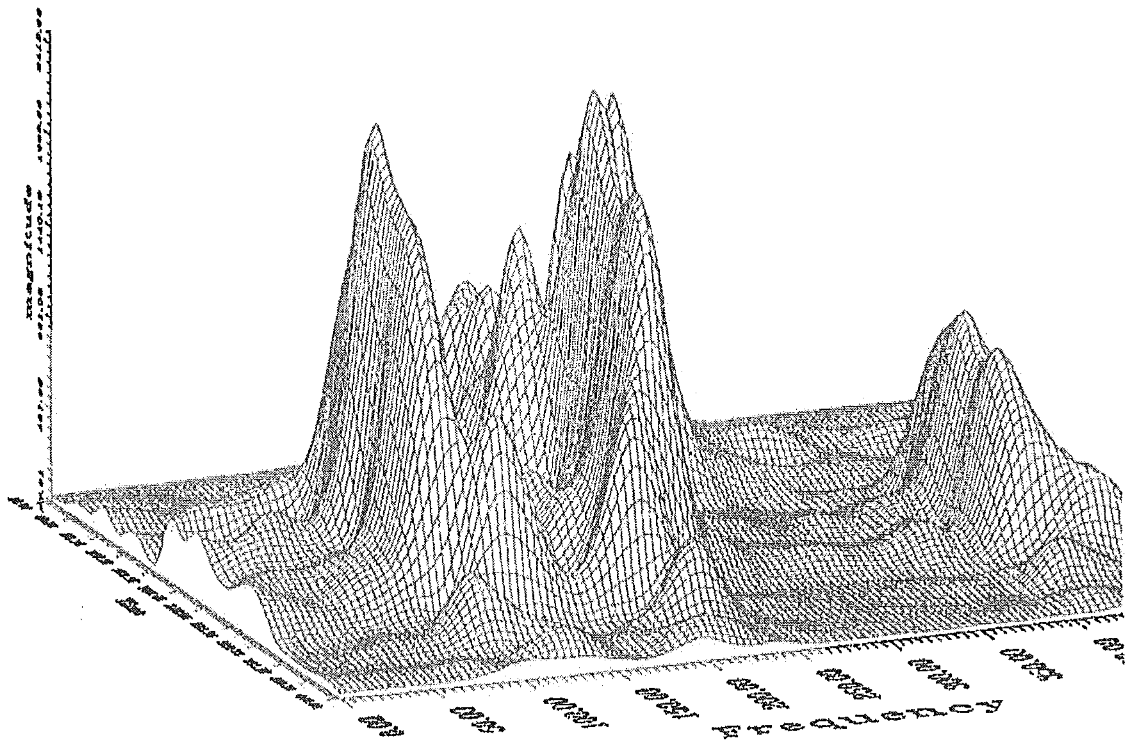


Figure 6. 3-D Spectrogram of the Modulated "Dilosh".

Yapay Sinir Ağları Yapıları

Artificial Neural Network Architectures

Artificial neural networks that grow when they learn and shrink when they forget
Öğrenince büyüyen, unutunca küçülen yapay sinir ağları

Ethem Alpaydın

Bilgisayar Mühendisliği Bölümü, Boğaziçi Üniversitesi, TR-80815 Istanbul
alpaydin@trboun.bitnet

Abstract

Learning when limited to modification of some parameters has a limited scope; the capability to modify the system structure is also needed to get a wider range of the learnable. In the case of artificial neural networks, learning by iterative adjustment of synaptic weights can only succeed if the network designer predefines an appropriate network structure, i.e., number of hidden layers, units, and the size and shape of their receptive and projective fields. This paper advocates the view that the network structure should not, as usually done, be determined by trial-and-error but should be computed by the learning algorithm. Incremental learning algorithms can modify the network structure by addition and/or removal of units and/or links. A survey of current connectionist literature is given on this line of thought. The reader is referred to (Alpaydın, 1991) for the author's own contribution to the field.

Özet

Eğer öğrenme sadece bazı parametrelerin değerlerini değiştirebilmek ise kullanım alanı kısıtlı kalır; öğrenilebileceklerin olabildiği kadar geniş olması için sistemin yapısının da değiştirilebilir olması gerekir. Yapay sinir ağlarında sadece bağlantı ağırlıklarının ayarlanması ile gerçekleşen öğrenmenin başarılı olabilmesi ancak ağ yapısının, yani saklı katman ve saklı ünite sayıları ve bunların önceki ve sonraki katmanlara bağlantı şeklinin, kullanım alanına uygun olması ile gerçekleşebilir. Bu tebliğ, ağ yapısının belirlenmesinin sıkça yapıldığı gibi deneme-yanılma ile değilde, yine öğrenme yordamının kendisi tarafından yapılması gerektiğini savunur. Bu tip yordamlarda ağ yapısı, örneğin hata geri yayma yordamında olduğu gibi durağan değildir; gerektiğinde yeni ünite ve bağlantılar eklenebilir veya olanlar çıkarılabilir. Bu tebliğ, bu konuda yapılmış çalışmalarin bir özetini verimektedir; yazarın bu alana katkısı için, okuyucu (Alpaydın, 1991)'e başvurmaya davet edilir.

1. INTRODUCTION

1.1. Assessing the quality of a neural network solution

There are three factors that affect the quality of a neural network solution:

[1] *Success achieved on test data* indicates how well the network generalizes to data unseen during training which one wants to maximize. This generally is taken as the only performance criterion.

[2] *Network complexity* by itself can be very difficult to assess but two important factors are the *network size* and the *processing complexity of each unit*. Network size gives the memory required which is the product of the number of connections and the number of bits required to store each connection weight. Processing complexity depends on how costly it is to implement processing occurring in each unit, e.g., sigmoid vs. threshold non-linearity, fan-in, fan-out properties, precision in storage and computation, etc. This has a negative effect on the quality as one prefers smaller and cheaper networks.

[3] *Learning time* is the time required to learn the given training data till one gets a reasonable amount of performance. This is to be minimized also.

In the ideal case, learning algorithms where a certain cost function is minimized should take into account not only success but the whole quality measure including success, network complexity, and learning time. However the actual relative importances of these three factors depend on the application and the implementation constraints. In tasks like optical character recognition where the environment does not change and thus learning is done only once, learning time is not a critical factor. On the other hand, when a hardware implementation is envisaged, network complexity is important and a smaller but less successful network can be preferred over a more complex but very successful one. In tasks like robotics where rapid adaptation to the environment is necessary, learning time has crucial importance. The best neural network for a given application is one having the highest quality and thus it does not make sense to say that one algorithm is better than another one per se; only based on a certain application and a set of implementation constraints can solutions be compared among themselves. This implies that with different hardware and environmental constraints, for the same training set, different networks may be required. The learning system may have a repertoire of learning algorithms and depending on the current constraints, one is chosen and employed. For example, when rapid adaptation is necessary, a one-shot learning method may be used to quickly learn encountered associations. When the system later has time to spare, an iterative fine-tuning process may be employed to improve performance.

1.2. Why smaller and simpler is better

In the case of feed-forward layered networks, the mapping capability of a network depends on its structure, i.e., the number of layers, and the number of hidden units (Lippman, 1987; Hanson & Burr, 1990; Hertz et al., 1991). Given a certain application and training data, the network structure should be pre-determined as algorithms like the back-propagation (Rumelhart et al., 1986) can modify only the synaptic weights but not the net structure.

Networks with more layers and hidden units can perform more complicated mappings however better performance on unseen data, i.e., generalization ability, implies lower order mappings. Given a certain training set, there are very many possible generalizations and one is interested in the simplest possible generalization. One reason for this is that simpler explanations of a phenomenon, i.e., those that require a shorter description, are more plausible and have a higher probability of occurrence (Rissanen, 1987). By having a smaller network, one also decreases the network size and thus less memory is required to store the connection weights, and the computational cost of each iteration decreases. However note that although one iteration takes less in a smaller network, the number of iterations to learn a certain training set can be more. Frequently an analogy is made between learning and curve fitting (Duda & Hart, 1973). There are two problems in curve fitting: finding out the *order* of the polynomial and finding out the *coefficients*of the polynomial once the order is determined. For example given a certain data set, one first decides on that the curve is second order thus has the form $f(x)=ax^2+bx+c$ and then computes somehow values of a, b, and c, e.g., to minimize sum of squared differences between required and predicted $f(x_i)$ for x_i in the training set. Once the coefficients are computed, $f(x_i)$ value can be computed for any x_i even for x_i that are not in the training set. Orders smaller than the good one risk not to lead to good approximations even for points in the data set. On the other hand, choosing a larger order implies fitting a high order polynomial to low order data and although one hopes that the high order terms will have zero coefficients to have their effect cancelled, this practically is not the case; it leads to perfect fit to points in the data set but very bad $f(x_i)$ values may be computed for x_i not in the training data, i.e., the system will not generalize well.

Similarly a network having a structure simpler than necessary cannot give good approximations even to patterns in the training set and a structure more complicated structure than necessary, i.e., with many hidden units, "overfits" in that it leads to nice fit to patterns in the training set performing poorly on patterns unseen. Bigger networks also need larger data samples for training; it was pointed out (Müller & Reinhardt, 1990) based on an information theoretic measure that the required number of patterns in the training set grows almost linearly with the number of hidden units.

As currently there is no formal way by which the network structure can be computed given a certain training set or application, the usual approach is trial-and-error, i.e., a series of attempts are made each one involving deciding on a more complicated network structure and iterating the learning algorithm a considerable number of times until one is content with the performance, which can be assessed by cross-validation. In determining the structure, the network designer is only guided by his/her intuition and rather limited knowledge of the application and the learning algorithm. Any knowledge related to the problem concerning the geometry or the topology of the input should be introduced to the network as help (Denker et al., 1987). When the input is an image for example, most of the constraints are local, i.e., nearby pixels have correlated output, thus it makes more sense to define local receptive fields than completely connected layers (Le Cun et al., 1989). A recent approach is to use a genetic algorithm to be able to "produce" better structures (Harp et al., 1990). The problem however is that "parent" networks should be trained for their fitness to be assessed and in tasks where training set is large or many generations are necessary, this turns out to be not very practical.

1.3. One-shot on-line learning

The time it takes to learn a given training set is crucial in many applications. Iterative algorithms based on gradient descent require very many iterations to converge and thus one is compelled to learn *off-line.* Another reason for off-line learning besides learning time is that, network models in which associations are distributed over a set of connections need to be introduced patterns in an unbiased fashion which cannot be guaranteed in a real world operational environment. One cannot for example add a certain association to network's memory by training with one pattern only; as weights are distributed, the whole training set should be re-learned together with the new pattern. However in an *on-line* learning system, one does not have time to do this and neither there is memory to store the whole training set. This is the case in many robotics applications where rapid adaptation to environment is a must. Iterative algorithms or networks using a distributed representation thus cannot learn at *one-shot* on-line. This fact led to the belief that neural network models cannot learn one-shot on-line and this became a frequent point on which learning limits of neural models are negatively judged (McCarthy, 1990; Leveit, 1990). To be able to learn on-line, addition of a new association should be done very quickly, i.e., one-shot, and without affecting the past existing knowledge of the network for other inputs. GAL algorithm (Alpaydın, 1991) using a local representation and based on an incremental approach has both of these properties and is a connectionist method that learns at one-shot.

2. INCREMENTAL LEARNING

The idea of incremental learning implies starting from the simplest possible network and adding units and/or connections whenever necessary to decrease error (Alpaydın, 1990a). To be able to decrease network size and increase generalization ability, one also wants to be able to get rid of units-connections whose absence will not degrade significantly system's performance. In both cases, as opposed to a static network structure, small modifications to a dynamic network structure during learning is envisaged. Determination of the network structure and computation of connection weights are not done separately but together, both by the learning algorithm.

Approaches given in the connectionist literature leading to network structure modification can be divided into two classes. There are those that start with a big network and eliminate the unnecessary and there are others that start from small and add whatever is necessary[+]

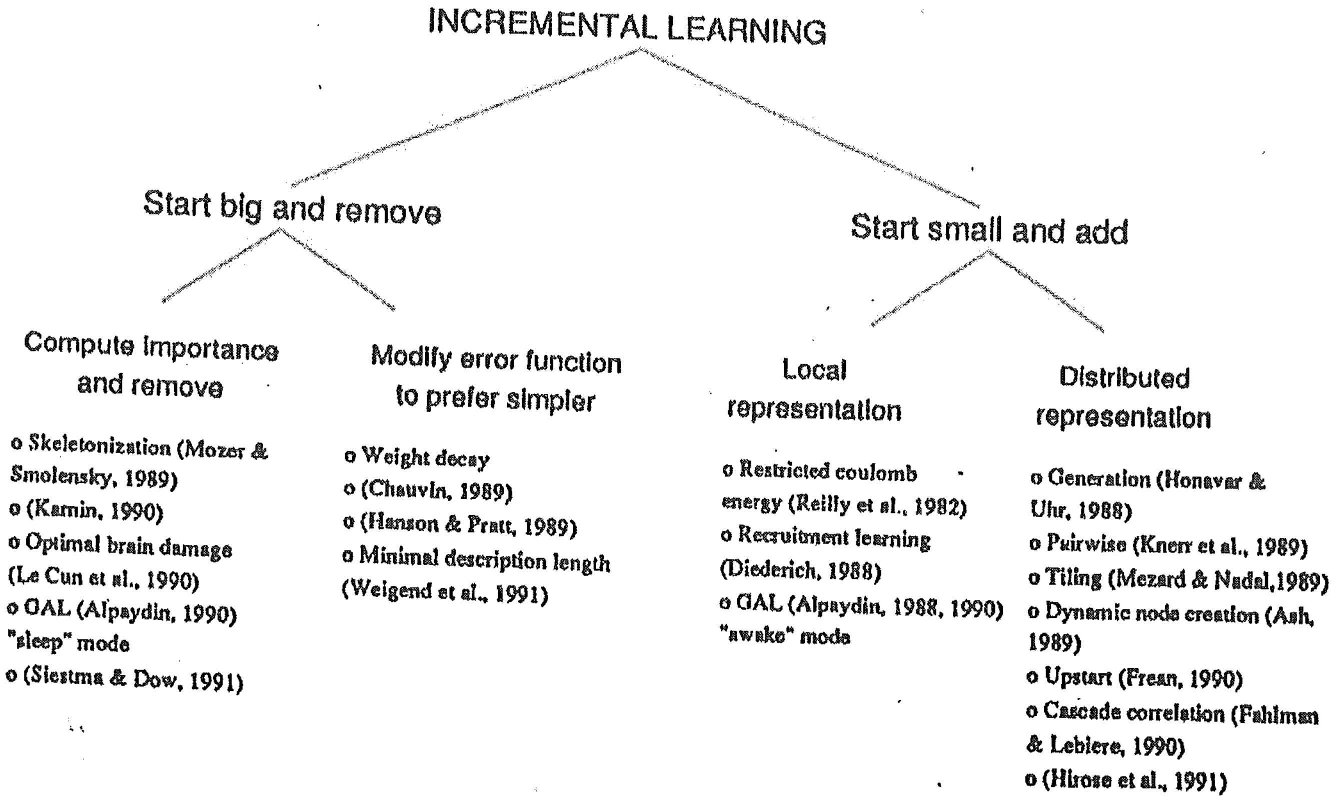

INCREMENTAL LEARNING
Start big and remove
Start small and add
Compute importance and remove
Modify error function to prefer simpler
Local representation
Distributed representation
o Skeletonization (Mozer & Smolensky, 1989)
o (Karnin, 1990)
o Optimal brain damage (Le Cun et al., 1990)
o GAL (Alpaydin, 1990) "sleep" mode
o (Siestma & Dow, 1991)
o Weight decay
o (Chauvin, 1989)
o (Hanson & Pratt, 1989)
o Minimal description length (Weigend et al., 1991)
o Restricted coulomb energy (Reilly et al., 1982)
o Recruitment learning (Diederich, 1988)
o GAL (Alpaydin, 1988, 1990) "awake" mode
o Generation (Honavar & Uhr, 1988)
o Pairwise (Knerr et al., 1989)
o Tiling (Mezard & Nadal,1989)
o Dynamic node creation (Ash, 1989)
o Upstart (Frean, 1990)
o Cascade correlation (Fahlman & Lebiere, 1990)
o (Hirose et al., 1991)

Fig. 1. Taxonomy of incremental learning.

2.1. Start big and remove

In the context of polynomial curve fitting the "start big and remove" approach implies starting from a high order polynomial and eliminating those high order terms which do not contribute significantly to success. Such methods are also called *pruning* or *destructive*. If one starts with a large network and if the problem in fact requires a simpler network, one likes to have the weights of all unnecessary connections and the output of all unnecessary units equal to zero. There are two approaches in achieving this:

[1] One may explicitly try to compute how important is the existence of a connection/unit in keeping the error low after the network has been trained and a number of the least important may then be deleted. The remaining network needs to continue to be trained. In the ideal case, understanding the importance of a connection/unit requires training two

[+] Note that there are also incremental *unsupervised* learning algorithms like ART (Carpenter & Grossberg, 1987) and GAR (Alpaydın, 1990a) which are beyond the scope of this paper. In unsupervised incremental learning, one adds a new cluster index whenever the current input is not similar to any of the existing clusters. The similarity measure is thus done in the input space regardless of the class to which the input patterns belong.

networks one with the connection/unit and one without. As this is not practical for large networks, heuristical approaches have been proposed with the back-propagation algorithm where the sensitivity of the error function to the elimination of a connection/unit is estimated.

- In the "skeletonization" procedure (Mozer & Smolensky, 1989), the network is trained till a certain performance criterion is met. The "relevance" of each connection is then computed which is given as the partial derivative of error with respect to the connection. However this value tends to zero when error decreases thus a poor relevance is computed when error is low. Using a linear error function for computation of relevances, i.e., the sum of the absolute value of the differences of required and actual values, leads to better relevance values.
- (Karnin, 1990) computes the "sensitivity" in the same way but sums the values computed throughout learning instead of computing only once at the end. More memory and computation is required but the usual quadratic error measure can be used.
- "Optimal brain damage" (Le Cun et al., 1990) uses an information theoretic measure to compute the "saliency" of a connection using the second derivative of the error function. Training proceeds till error reaches down to a certain value at which point saliencies are computed and a number of the least salient are deleted and the remaining network is re-trained.
- Grow and Learn (GAL) algorithm (Alpaydın, 1990a), has a "sleep" mode during which the network is closed to the environment, the inputs are generated by the system itself, and units that are no longer necessary due to recent additions are removed.
- Siestma and Dow (1991) examine the behavior of units under the presentation of the entire training data and decide to prune accordingly. From "broad" networks with few layers and many units on each layer, after training, they trim as many units as possible and by adding extra layers, generate "long narrow" networks with many layers but few units on each layer; they discover however that networks of the latter type generalize poorly.

[2] Instead of approximating how much the error will change if the unit/connection is eliminated, one may also modify the learning algorithm so that after training, the unnecessary connections/units will have zero weight/output.

- One may build a tendency in the learning algorithm to have those weights that are not relevant decay to zero by decrementing them by a certain factor at each weight update (see review in Hertz et al., 1991). Weights that are necessary to store associations will be moved away from zero but those that are not needed will not be increased and will finally be close to zero.
- This decay can be done also implicitly by modifying the error function. Terms can be added to the error function to penalize large weights (Chauvin, 1989) and hidden units that have small outputs (Hanson & Pratt, 1989).
- Another possibility is to use the information theoretic idea of "minimum description length" and add a term to the cost function that penalizes network complexity, i.e., number of connections (Weigend et al., 1991). Thus during gradient descent, the algorithm will settle to the network that has the best trade-off between error and complexity. Such a cost function is similar to the quality measure proposed in the first section; however the network complexity is defined very simply as the number of connections.

2.2. Start small and add

The other approach in dynamic modification of network structure during learning, which can be named "start small and add," implies starting from a simple network and adding units and/or connections to decrease error. These methods are also called *growth* or *constructive*. In the context of curve fitting, it implies starting with a low order polynomial and adding higher order terms whenever the polynomial of current order cannot give a good fit for any set of coefficients. Note that this cannot be done in a straightforward manner especially in networks where associations are distributed over a number of shared connections; the whole training should be re-done in such a case. One needs a certain mechanism whereby addition of a new unit improves success instead of corrupting the harmony as one would normally expect. There are two possibilities:

[1] If one can make sure that when the new unit gets activated, none of the ancient units get activated, there will be no problem. The units should thus somehow be able to suppress other units when they get control. This implies a competitive strategy and a local representation.

- The first incremental neural learning algorithm is the Restricted Coulomb Energy (RCE) model (Reilly et al., 1982) which is an incremental version of Parzen windows. Associated with each unit is a number of prototypes where a prototype gets activated only if the input falls into its domination region, determined by a distance computation followed by a thresholding. If an input does not activate any prototype, a new prototype unit is created at that position with an initially large domination region. Prototypes that get activated for inputs that belong to different classes are penalized by having their regions decreased which is done by modifying the threshold. The input space is thus divided into zones dominated by prototype units. A number of sweeps is necessary to finetune the thresholds where units closer to class boundaries have small zones and units interior have larger domination zones.
- Recruitment learning (Diederich, 1988) is used in the case of structured connectionist networks where a previously free unit is committed to represent a new concept and required connections built up dynamically (Feldman, 1982). This is a one-shot learning algorithm, i.e., one iteration is sufficient to learn a new concept.
- In the first version of Grow-and-Learn (Alpaydın, 1988), weights in a single layer were learned by Hebbian learning at one shot. However if an association could not be learned or if addition of this association corrupted the previously learned associations, a new hidden unit was added with input weights equal to the input vector. The output weight was computed in such a manner to compensate for the effect of the input layer and thus impose any output. The problem was that as Hebbian learning was used, orthogonality of input patterns were necessary and as this is rarely the case, many units were allocated. However Hebbian learning made the algorithm a one-shot learning one.
- The current version of Grow-and-Learn (GAL) algorithm (Alpaydın, 1990a), uses also a local representation by having a number of exemplars associated with each class. It learns at one-shot but orthogonality of patterns is no longer required.

[2] Another possibility is to divide the network into separately trained subnetworks where such subnetworks can be added in an incremental manner. One approach is to have subnets that have competition between subnets, another is to have each subnet as another hidden layer.

- The "generation" method proposed by Honavar and Uhr (1988) enables a "recognition cone" to modify its own topology by growing links and recruiting units whenever performance ceases to improve during learning by weight adjustment using back-propagation.
- The "stepwise procedure" uses subnets of different conceptual interpretations (Knerr et al., 1989). In this method, one first trains a one layer network with the Perceptron learning algorithm assuming that classes are linearly separable. For a class where this is not satisfied, one adds a subnet to separate classes in a pairwise manner. For cases where this does not work either, one performs a piecewise approximation of boundaries using logical functions by additional subnets. As linear separability is rarely the case, one generally is obliged to separate classes in a pairwise manner two by two. The major drawback of this is that the number of hidden units increase exponentially with the number of class units.

- Another approach named the "tiling" algorithm adds a new hidden layer whenever the required mapping cannot be done with the existing network (Mezard & Nadal, 1989; explained also in Hertz et al., 1991). There is a "master" unit which is trained to be the output unit by the pocket algorithm---a variant of the Perceptron learning algorithm. If this unit cannot learn all the required associations, additional "ancillary" units are added to learn the rest and another layer is created with a master unit and learning proceeds till the master unit can learn to behave like the output unit.
- The "dynamic node creation" method (Ash, 1989; explained also in Müller & Reinhardt, 1990) trains networks with one hidden layer only. Given a certain net that is being trained, if the rate of decrease of error falls down a certain value, a new hidden unit is added and training is resumed when all connections are continued to be modified.
- The "upstart" algorithm (Frean, 1990) uses binary units. Like the "tiling" algorithm, first one unit is trained to learn the required associations using the pocket algorithm. If this is not successful, "daughter" units are created to correct the output of this "parent" unit, for "wrongly on" and "wrongly off" cases. This is repeated in a recursive manner to lead to a binary tree which can then be "squashed" into one hidden layer.
- In the "cascade correlation" algorithm (Fahlman & Lebiere, 1990), if the required mapping cannot be learned by one layer, a hidden unit is added and trained while the previously trained weights are "frozen." If this does not work either, another hidden unit is added as another hidden layer and so on. A hidden layer has only one hidden unit but connections skip layers, i.e., a unit has connections to all the following layers.
- Method proposed by (Hirose et al., 1991) is quite similar to that proposed by Ash (1989), namely, using a network with only one hidden layer, if the rate of decrease for error becomes small, additional hidden units are added. Their contribution is that, once the network converges, the most recently added hidden unit is removed and the network is checked to determine whether the same function can be achieved by fewer hidden units. If the network cannot converge when a hidden unit is removed, the last network that converged is chosen as the final network.

REFERENCES

[1] Alpaydın, E. (1988) "Grow and Learn" *Internal Note, Lami--EPFL,* Switzerland.

[2] Alpaydın, E. (1990a) "Neural models of incremental supervised and unsupervised learning" *PhD dissertation,* Ecole Polytechnique Federale de Lausanne, Switzerland.

[3] Alpaydın, E. (1991) "GAL: Networks that grow when they learn and shrink when they forget" *International Computer Science Institute;* TR 91-032, Berkeley, CA.

[4] Ash, T. (1989) "Dynamic node creation in backpropagation networks" *Connection Science;* 1, 365--375

[5] Carpenter, G.A., Grossberg, S. (1987) "ART2: Self-organization of stable category recognition codes for analog input patterns" *Applied Optics,* 26, 4919--4930

[6] Chauvin, Y. (1989) "A back-propagation algorithm with optimal use of hidden units, *NIPS;* D.S. Touretzky (ed.), 1, 519--526, Morgan Kaufmann

[7] Denker, J., Schwartz, D., Wittner, B., Solla, S., Howard, R., Jackel, L., Hopfield, J. (1987) "Large automatic learning, rule extraction, and generalization, *Complex Systems,* 1, 877--922

[8] Diederich, J. (1988)} "Connectionist recruitment learning", *Proc. of the 8th European conf. on Artificial Intelligence;* London, UK

[9] Duda, R.O., Hart, P.E. (1973) *Pattern classification and scene analysis;* John Wiley

[10] Fahlman, S.E., Lebiere, C. (1990) "The cascade-correlation architecture, *NIPS;* D.S. Touretzky (ed.), 2, 524--532, Morgan Kaufmann

[11] Feldman, J. (1982) "Dynamic connections in neural networks" *Biological Cybernetics;* 46, 27--39

[12] Frean, M. (1990) "The upstart algorithm: A method for constructing and training

feedforward neural networks" *Neural Computation*, 2, 198--209

[13] Hanson, S.J., Pratt, L.Y. (1989) "Comparing biases for minimal network construction with back-propagation, *NIPS*, D.S. Touretzky (ed.), 1, 177--185, Morgan Kaufmann

[14] Hanson, S.J., Burr, D.J.(1990) "What connectionist models learn: Learning and representation in connectionist networks" *Behavioral and Brain Sciences*, 13, 471--518

[15] Harp, S.A., Samad, T., Guha, A.(1990) "Designing application-specific neural networks using the genetic algorithm" *NIPS*, D.S. Touretzky (ed.), 2, 447--454, Morgan Kaufmann

[16] Hertz, J., Krogh, A., Palmer, R.G. (1991) *Introduction to the theory of neural computation*, Addison Wesley

[17] Hirose, Y., Yamashita, K., Hijiya, S. (1991) "Back-propagation algorithm which varies the number of hidden units" *Neural Networks*, 4, 61--66

[18] Honavar, V., Uhr, L. (1988) "A network of neuron-like units that learns to perceive by generation as well as reweighting of its links, *Proc. of the 1988 Connectionist Summer Schod*, D. Touretzky, G. Hinton, T. Sejnowski (eds.), Morgan Kaufmann

[19] Karnin, E.D. (1990) "A simple procedure for pruning back-propagation trained neural networks" *IEEE trans. on neural networks*, 1, 239--242

[20] Knerr, S., Personnaz, L., Dreyfus, G. (1989) "Single layer learning revisited: A stepwise procedure for building and training a neural network" *Neurocomputing: Algorithms, architectures, and applications*, F. Fogelman-Soulie, J. Herault (eds.), NATO ASI Series, Springer

[21] Le Cun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jeckel, L.D. (1989) "Backpropagation applied to handwritten zip recognition" *Neural Computation*, 1, 541--551

[22] Le Cun, Y., Denker, J.S., Solla, S.A. (1990) "Optimal brain damage, *NIPS*, D.S. Touretzky (ed.), 2, Morgan Kaufmann, 598--605

[23] Leveit, W.J.M. (1990) "On learnability, empirical foundations, and naturalness" *Behavioral and Brain Sciences*, 13, 501

[24] Lippman, R.P. (1987) "An introduction to computing with neural nets" *IEEE ASSP magazine*, 4, 4--22

[25] Mezard, M., Nadal, J.-P. (1989) "Learning in feedforward layered networks: The tiling algorithm" *Journal of Physics A*, 22, 2191--2204

[26] McCarthy, J. (1990) "Interview: Approaches to artificial intelligence" *IEEE Expert*, 5(3), 87--89

[27] Mozer, M.C., Smolensky, P. (1989) "Skeletonization: A technique for trimming the fat from a network via relevance assessment" *Connection Science*, 1, 3--26

[28] Müller, B., Reinhardt, J. (1990) *Neural networks: An introduction*, Springer

[29] Reilly, D.L., Cooper, L.N., Elbaum, C. (1982) "A neural model for category learning" *Biological Cybernetics*, 45, 35--41

[30] Rissanen, J. (1987) "Stochastic complexity" *Journal of Royal Statistical Society B*, 49, 223--239 and 252--265

[31] Rumelhart, D.E., Hinton, G.E., Williams, R.J. (1986) "Learning internal representations by error propagation" *PDP*, D.E. Rumelhart, J.L. McClelland (eds.), 1, MIT Press, 151--193

[32] Siestma, J., Dow, R.J.F. (1991) "Creating artificial neural networks that generalize" *Neural Networks*, 4, 67--79

[33] Weigend, A.S., Rumelhart, D.E., Huberman, B.A. (1991) "Generalization by weight-elimination with application to forecasting, *NIPS*, R.P. Lippman, J. Moody, D.S. Touretzky (eds.), 3, Morgan Kaufmann

Sinir Aglarinin Optik Cihazlarla Tasarimi

B.S. Yılbaş[1] ve N. Karaboga[2]

[1]Erciyes Üniversitesi, Mühendislik Fakültesi, Kayseri/Turkey
[2]UWCC, ELSYM, Bute Building, CF1 3YH, CARDIFF/UK

Ozet

Nörokompütür, ya anolog cihazlarla yada dijital donanımla tasarlanabilen bir sinir agı yapısıdır. Fakat, büyük sinir agı yapılarının kurulmasında üsteki her iki tasarım tekniklerininde dezavantajları vardır. Böyle durumlarda, optik cihazlarla yapılan tasarımlar optigin tabiatından gelen bazı özelliklerden dolayı diger tekniklerle yapılan tasarımlara nazaran daha iyi sonuçlar vermektedir.

Bu makalenin amacı, optik nörokompütürlerin avantajlarını özetlemek, çalısma mekanizmasını ve temel elemanlarını tarif etmektir.

Anahtar Kelimeler: Yapay sinir agları; anolog tasarım; optik

Implementation of Neural Networks Using Optical Devices

Abstract

Neurocomputer is a neural network architecture which can be implemented using either analog devices or digital hardware. However, for constructing a massive neural network architecture the above implementations have their own drawbacks. In such case, optical-based analog implementations could prove more efficient due to their inherent parallelism, faster speed and less cross-talk or noise interference among different elements.

The aim of this paper is to summarize the advantages of optical-based neuro computer and to describe the functioning of its basic elements and mechanisms.

Keywords: Artificial neural networks; analog implementation; optic

1. GIRIS

Nöral hesaplamanın çok dikkat çeken özelliklerinden biri nöral algorıtmaların nisbeten kolay bir şekilde anolog donanımlarla tasarlanabilmeleridir. Sinir agları elektronik, optik veya her ikisinden yada diger donanımlardan (kimyasal, akustik, mekanik) faydalanarak analog olarak tasarlanabilir [1]. Fakat bugünlerde en çok kullanılanlar elektronik VLSI ve optiktir.

Anolog VLSI olarak tasarımın avantajları, silikon cihazların fabrikasyon teknolojisinin hazır olması ve non-lineer yarı iletken elemanların yapılarının kompleksligidir. Dezavantajı ise, ara baglantılardan kaynaklanmaktadır. Nöron birimler başına düşen baglantı sayısının fazla oldugu aglarda, VLSI tasarım alanının çogu, birimleri birbirine baglayan kanalların alanı tarafından işgal edilir. Optik tasarımların avantajları ise üç boyutlu lineer baglantıların oluşturulabilmesi ve bu ara baglantıların agırlıklarının nisbeten kolay bir şekilde degiştirilebilmesidir. Optik, optik hesaplamanın düşük hassasiyetine ve yapılan işlemlerin sınırlı sayısına ragmen, ihtiyac duyulan paralellige izin vermesi ve çok yogun ara baglantı oluşturulmasına musaittir. Bundan dolayı büyük sinir aglarının tasarımına imkan sagladıgından, günümüzde oldukca fazla ilgi çekmektedir.

Bu makalenin, ikinci bölümünde yapay sinir aglarının optik elemanlar kullanılarak tasarlanması ile ilgili kısa bir giris yapılacak ve üçüncü bölümünde optik cihazlar kullanılarak tasarlanmış basit bir optik nörokompütürün elemanları tanıtılacaktır.

2. SINIR AGLARI VE OPTIK

Herhangi tek katlı bir sinir agının hesaplama işlemi aşagıdaki gibi ifade edilir [2].

$$y_j = f\left(\sum_i w_{ij} \; x_i\right)$$

Burada nöron birimi (i) arabaglantı katın girişinde, nöron birimi (j) arabaglantı katının çıkışında bulunmaktadır. y_j, nöron birimi (j) nin x nöron birimi (i) nin çıkışı ve w_{ij} de bu birimler arasındaki arabaglantıların agırlıklarıdır. f fonksiyonu nöron birimin nonlineerligini temsileder. Parentez içerisindeki terim (ρ_j), arabaglantı agırlık matrisi ve giriş vektörü arasında bir matris-vektör ürünüdür. f fonksiyonu sonuç vektörünün her bir elemanının üzerinde farklı bir sekilde etkilidir. Nokta nonlineeritesi olarak adlandırılan bu özellik uzaysal ısık modülatörü ile tasarımı mümkün kılar.

Günümüzdeki ögrenme algoritmalarının çogunu birkaç sınıfta toparlamak mümkündür [2]. Böyle bir sınıf şu formülle tarif edilebilir:

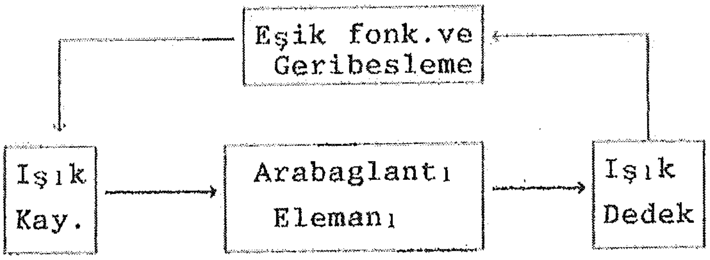
$$\Delta w_{ij} = \alpha \, \delta_j x_i - \beta \, w_{ij}$$

Burada $\Delta w_{ij} = w_{ij}(k+1) - w_{ij}(k)$ yenilenmiş agırlıgı, k ögretme (training) indeksini belirler. α ögrenme kazanç sabiti, β özellikle hardware uygunlugu için kullanılan azalma (decay) sabitidir. Ögretme terimi $\delta_j$nin uygun farklı degerleri ile farklı ögrenme algorıtmaları elde edilebilir (Hebbian, Widrow-Hoff, tek katlı aglar için least minumum squares ve çok katlı sinir agları için back-propagation).

Ögrenme algoritmalarının diger sınıflarınında optik olarak tasarlanması mümkün olmasına ragmen araştırmacılar genellikle yukarıda formülle ifade edilen sınıf uzerinde çalısmalarını yogunlaştırdılar. 1985 de Hopfield modeli benzeri bir sinir agını elektronik ve optik elemanlar kullanarak tasarlayan Farhat ve Psaltis [3] optik elemanları yapay sinir agının anolog olarak tasarımında kullanan ilk araştırmacılar dır. 1986 da Bernard Soffer ve digerleri [4] dünyanın ilk tamamen optik cihazlarla tasarlanmış sinir agını gerçeklestirdiler. Optik elemanların kullanıldıgı ve back-propagation ögrenme tekniginin uygulandıgı üç katlı Rumelhart benzeri bir sinir agı modeli Wagner ve Psaltis tarafından 1987 de [5] geliştirildi ve son zamanlarda Ishikuma ve digerleri [6] tarafından bu model tasarlandı.

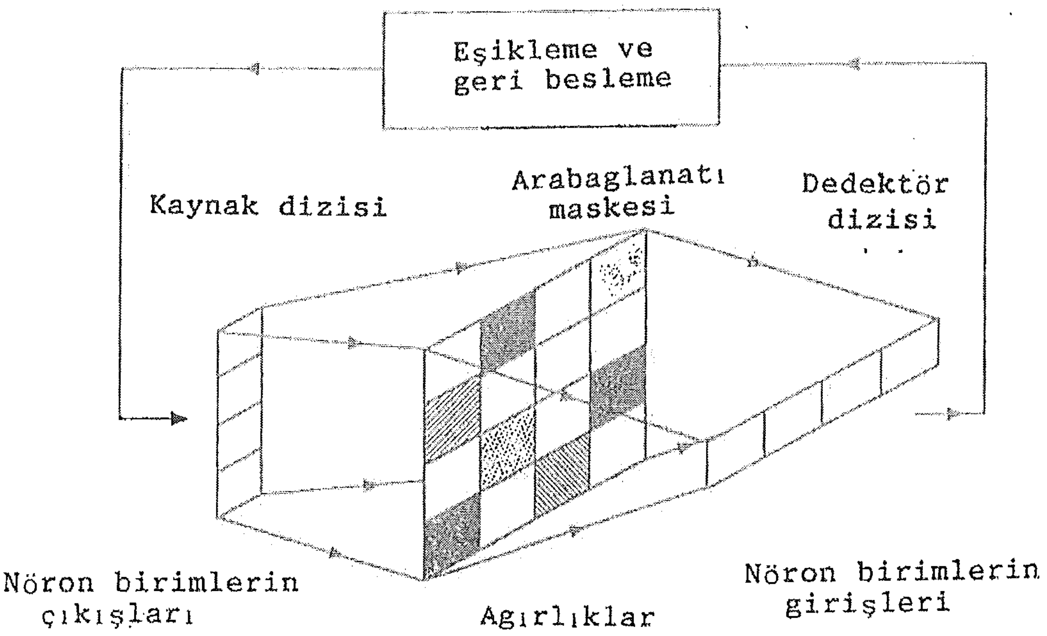
3. BASIT BIR OPTIK NOROKOMPUTURUN YAPISI

Basit bir optik nörokompütürün blok diagramı şekil 1. de verilmektedir. Tasarım, nöron birimlerin bir boyutlu (1-D) veya iki boyutlu (2-D) dizileri ile 2 boyutlu veya 3 boyutlu (3-D) ara baglantı elemanı kullanarak yapılır. Optik tasarımlarda, bir nöronun girişi ışık dedektörü, çıkışı ise bir ışık kaynagı veya dedekte edilmiş işaretle elektriksel olarak kontrol edilen bir modülatördür.



Sekil 1. Temel bir optik nörokompütür sistemin blok diagramı

Şekil 2., bir boyutlu kaynak ve dedektör dizilerinin, iki boyutlu arabaglantı elamanı ile birlikte kullanılmasıyla dizayn edilmiş bir sistemi göstermektedir. Girişteki her piksel, silindirik mercekler kullanmak suretiyle optik olarak genişletilir ve arabaglanti matrisinin karşılık gelen sırası aydınlatılır. Maske, anolog agırlıkları depolar ve maskedeki bir sütunun bir çıkış pikseline karşı gelmesini saglamak amacıyla ışının daraltılmasından önce bir nokta nokta çarpım islemi uygular. Yukarıda Bölüm 2 deki deklemde parantez icine alınmış terimin temsil ettigi gibi tam bir parelel anolog optik matris-vektör çarpımı saglar.



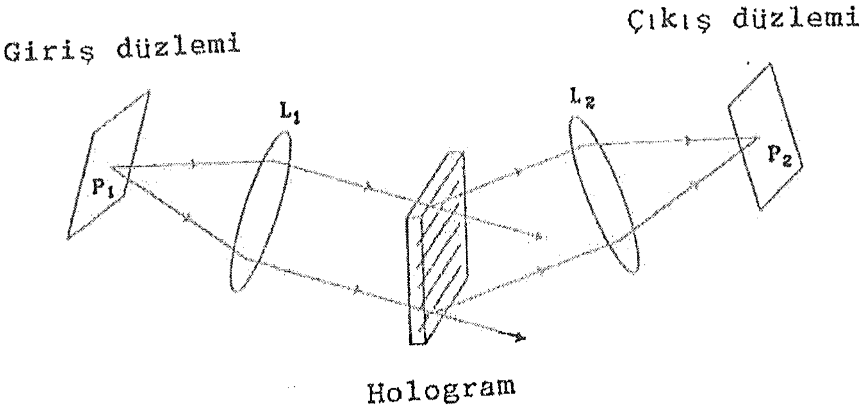
Sekil 2. Iki boyutlu hologramla tek boyutlu giriş ve çikiş kullanarak dizayn edilmiş optik nörokompütür.

3.1. Arabaglanti elemani

Şekil 3 de düzlemsel bir hologramla tasarlanmış optik bir korelatörün şematik diagramı (vander lught correlator) [7] gösterilmektedir. Giriş düzlemindeki bir nokta (P_1) çıkış düzlemindeki bir noktaya (P_2) şu sekilde baglanır: Ilk mercek (L_1), P_1 noktasından gelen ışını hologramı aydınlatan tek bir düzlemsel dalgaya çevirir. Bu düzlemsel dalganın yayılım yönü giriş düzlemindeki P_1 in pozisyonuyla bire bir eşleşmeye sahiptir.

Yukarıda da bahsedildigi gibi orta düzleme yerleştirilen hologramın gayesi gelen ışıgı çıkış düzlemindeki noktalara dogru kırmak ve böylece giriş ve çıkış noktaları arasında baglantı oluşturmaktır. Hologramı sinuzoidal izgaraların (gratings) bir süperpozisyonu olarak duşünebiliriz. Herbir ızgara gelen dalganın belli bir kısmını çıkış düzlemine dogru yayılan başka bir düzlemsel dalga halinde kırar [8] .

Gelen dalganın yayılım yönü ile kırılmış dalganın yayılım yönu arasındaki fark uzaysal frekans ve herbir ızgaranın (fringes) lerinin oriyantasyonları ile belirlenir. Ikinci lens (L_2) kırılmıs dalgaların herbirini odaklanmış bir noktaya dogru yöneltir. Bu noktanın çıkış düzlemindeki pozisyonu kırılmış ışının yayılım yönüne karşılık gelir. Bu şekilde hologramda kaydedilen her sinüzoidal ızgara P_1 i bir çıkış noktasına baglar. Baglantının agırlıgı kaydedilen ızgaranın ışıgı kırıcılıgı ile orantılıdır.



Sekil 3. Vander lught correlator (iki boyutlu bir hologramın kullanıldıgı arabaglantı sistemi)

Düzlemsel bir hologram ve bir boyutlu giriş ve çıkış kullanan bir optik sistem, 1000 civarında tamamen birbirlerine baglanmış nöron birimlerden oluşan bir sinir agının tasarımı için kullanılabilir. Çok daha fazla sayıda nöron birimlerinin kullanılması durumlarında optik sistem üç boyutlu hologramla dizayn edilir. Herbir nöron giriş birimi (i) ni tek bir çıkış nöronuna (j) baglayan ayrı bir hacim ızgarası vardır. Öncedende bahsedildigi gibi herbir ızgaranın ışıgı kırma degeri agırlık w_{ij} ile dogru orantılıdır. Hacim ızgaraları ile düzlemsel ızgaralar arasındakı farklılık, hacim ızgaralarının ışıgın geliş açısına da duyarlı olmalarıdır.

3.2. Isik Kaynaklari

Monokromatik optik güç farklı tipteki laserler kullanılarak elde edilebilir. Bu tip laserlere örnek olarak argon-iyon, Nd-YAG, He-Ne, He-Cd, sıvı ve yarı ileteken laserler gösterilebilir. Harcadıgı elektrik güce, sebeb oldugu ısıya ve maliyete bakıldıgında yarı iletken laserlerin digerlerine üstünlükleri vardır. Bu tip laserlerin fiziksel büyüklükleri, harici bir sogutucuya ihtiyaç duymadıkları müddetce bir entegre devreye yerleştirilecek kadar küçüktür.

Bu laserlerin en önemli özelliklerinden biri yüksek band genişlikli direkt modülasyona müsait olmalarıdır. Yani laserin çıkış şiddetinin, cihazın uçlarına uygulanan voltajın direkt olarak degiştirilmesiyle Giga-Hertz (GH) ler seviyesinde modülasyonunun yapılabilmesidir [2].

Kaynak olarak kullanılan diger bir eleman LED dir. Bunun avantajı herhangi bir eşik operasyonunun kullanılmamasıdır. Dolayısıyla LED, uygulanan voltajın çok genis bir sahasında kullanılabilir. Buna zıt olarak nisbeten geniş bandlı kaynaklar olmaları nedeniyle hologramların kaydedici elemanlar olarak kulanıldıgı uygulamalarda kaynak olarak kullanılmaya pek elverişli degildirler.

3.3. Isik Dedektorleri

Dedektörler, opto-elektronik elemanlardır. Kısacası foton-elektron çeviricidirler. Yani, gelen optik şiddeti genellikle akım veya voltaj şeklindeki elektron formuna sokarlar. Bundan dolayı dedektörler nöron birim çıkışlarının optik olarak temsil edilmesine imkan saglarlar.

En basit dedeksiyon elemanı foto-iletkenlerdir. Bunlar dirençlerini ışıgın şiddetine göre degiştirirler. En yaygin kullanılan tek-piksel dedektörler foto-diyodlar dır. Diger bir ışıga duyarlı cihaz foto-transistör lerdir. Normal bir transistörde oldugu gibi bunlar bir akım kazancı saglarlar ama tek fark bias akımının optik formda olmasıdır [8]. Dedektör dizileri foto-diyod gibi küçük dedektörlerin dizileridir. Böyle diziler, paralel halde bulunan bütün piksellerden real-time çıkış elde edilmesini saglarlar ama hassasiyetleri düşüktür.

3.4. Esik fonksiyonu ve geribesleme elemani

Eşik fonksiyonları ve geribesleme baglantıları ya optik olarak yada elektronik elemanlar kullanılarak gerçekleştirilir. Eşik fonksiyonu, bir çok farklı tipteki optik malzemeler kullanarak tasarlanabilir. Bu malzemelerin özellikleri, bölgesel optik şiddetin ilk ve daha yüksek dereceden güçleri ile orantılı olarak kırılma indislerinde veya absorbsiyon katsayılarında bir degişim göstermeleridir.

Böyle bir malzemeye örnek film kalınlıgı içerisinde erişilebilecek maximum optik yogunlugun, saturasyon degerini belirledigi fotografik film dir. Diger bir optik nonlineerlige örnek, Cadmium Sulfat ve silikon gibi belli yarı iletken malzemelerin foto-iletken doyum davranışlarıdır. Bu optik tekniklerin uzun tepki süresi veya saturasyonu gerceklestirmek için çok yüksek optik şiddetlere ihtiyaç duymaları gibi dezavantajları vardır. Dedektörler ve modulatörlerle birlikte elektronik devreler kullanılmasıyla opto-elektronik nonlineeriteler elde edılerek bu dezavantajlar yok edilebilir[2].

4. SONUC

Bu makalede, yapay sinir aglarının optik cihazlar kullanılarak tasarlanmasının avantajları özetlenmiş ve basit bir optik nörokompütürün elemanları tanıtılmıştır. Yapay sinir agları'nın anolog tasarımı için optigin ve elektronigin avantajlarının birlikte kullanıldıgı opto-elektronik teknik gelecek için ümit verici görünmektedir.

Tesekkur

Yazarlar, Erciyes Üniversitesi'ne bu çalışmayı yapmasında N. Karaboga'ya gösterdikleri yardımlar dolayısıyla teşekkür ederler.

5. KAYNAKLAR

1. Hecht-Nielsen, R. *Neurocomputing* , Addison-Wesley, 1991.
2. Kosko, B. *Neural Networks for Signal Processing*, Prentice-Hall, 1992
3. Farhat, N.B. and Psaltis, D., Prata, A. and Paeh, E. Optical Implementation of Hopfield Model, *Appl. Opt.* Vol. 24, pp.1469-1475, 1985.
4. Dunning, G.J., Marom, E., Owecko, Y. and Soffer, B.H. Optical Holographic Associative Memory Using a Phase Conjugate Resonator, *SPIE Proc.* 625, January, 1986.
5. Wagner, K. and Psaltis, D. Multilayer Optical Learning Networks, Proc. *SPIE- Int. Soc. Opt. Eng.* Vol. 752, pp.86-97, 1987.
6. Ishikawa, M., Mukohzaka, N., Toyoda, H. and Suzuki, Y. Experimental Studies on Learning Capabilities of Optical Associative Memeory, *Appl. Opt.* Vol. 29, No. 2, pp. 289-295 January 10, 1990.
7. Vander Lught, A.B. *IEEE Trans. on Information Theory*, IT-10, 98, 1985.
8. Feitelson, D.G. *Optical computing*, MIT Press, 1988.

Classification and Computation on Non-uniform Finite Cellular Automata Networks

Hürevren KILIÇ[1] and Marifi GÜLER[2]

[1] Bilkent Computer Center, Bilkent University, Bilkent, 06533 Ankara, Türkiye

[2] Department of Computer Engineering, Middle East Technical University,
06531 Ankara, Türkiye

Abstract

In this paper, a model called *Non-uniform Finite Cellular Automata (CA) Network* is introduced, and its classification and computation power studied. The suggested model is similar to the original Cellular Automata Network model, with its local neighborhood property, but neighborhood definitions of cells are not the same (non-uniform) for each cell and determined by an algorithm. The model is similar to the Neural Network (NN) model with its different local cell (neuron) transition function definitions and with its training (or feature extraction) mode. Depending on the nature of input-output templates, computation can be done on the system. Necessary and sufficient conditions for doing computation on this model are based on the work of Tchuente, and will be elaborated upon in this paper.

1. INTRODUCTION

An Automata Network may be defined as a locally connected large set of cells (finite automata), which can evolve at discrete time steps through mutual interactions. In this model, space, time and cells are discrete. Automata networks have many application areas such as, Artificial Intelligence, Pattern Recognition and Learning Systems.

A particular class of Automata Networks is the Cellular Automata (CA) Networks. In computer science cellular automata is used to model parallel processing and Von Neumann (*self-reproducing*) machines. In cellular automata, space is divided into discrete small units called *cells* or *sites*. Each cell can take k different state values. At time t, all the cells will have a specific state value. Rules local to a specific cell determine what the value of that cell at time $(t+1)$ will be. Rules are the same for each cell. For k number of states per cell and n number of cells in the neighborhood of a cell, k^z (where $z = k^n$) possible local transition functions for that cell exist.

Another important class of Automata Networks, called Neural Networks (NN), with similar characteristics to CA Networks, use only threshold type functions as their neural transition functions. Unlike cellular automata, in neural networks each neuron does not have to evolve according to the same neural transition function. However, in general, neural networks without the neighborhood concept that exists in cellular automata model, require great number of neural connections (or neural dependency) between their neurons (e.g. Hopfield model, multi-layer perceptron model).

2. AUTOMATA NETWORKS

An automata network can be defined as locally interconnected set of cells, which can evolve at discrete time steps. This evolution occurs through mutual interactions between these locally connected cells. Formally, an automata network can be described as mapping F from S^n into itself, where S is finite state space and n is the number of inter-connected cells. The structure of connection is determined by F; if i^{th} component of mapping F depends on j^{th} variable, cell i receives a connection from cell j. A state of the network is a vector X in S^n and dynamics on the network can be defined as:

$Y = F(X)$ where F : mapping function $X, Y \in S^n$ are state vectors

At each time step, each automaton in cell i computes its next state according to the rules of mapping F_i, and causes a global evolution of network. This is known as parallel iteration. Other iteration modes such as sequential mode where cells are updated in a prescribed order and memory mode where previous cell values are used, also exist [1].

Since the state space S is usually finite, at the end of at most k^n steps of evolution, (where k is the number different states that a cell can be in and n is the number of cells), the system will enter into a cycle or a fixed point which can be accepted as a cycle of period 1. The mapping function F defined above is deterministic, i.e., we can guess exactly the next state of the system from its present state. In this paper, only the *deterministic* (not random) automata networks are considered and studied. Automata networks are discrete and dynamical systems in time and space and they can be represented by a graph, where each node of the graph takes one of the states in a finite set. Moreover, the evolution of the network results from changing states of each site (or node) according to a transition rule that takes into account only the state of its neighbors in the graph.

An important class of Automata Networks are *Neural Networks*. In this model, the graph representing the network is non-oriented, i.e., direction between nodes (neurons) is not important and graph is finite. Nodes can take one of two state values { -1,1 }. The transition rule is a threshold function whose inputs are the output of other threshold units (neurons) weighted by real numbers. In general, the sign threshold function is used to calculate the new state of a neuron. If the weighted sum of neuron values, other than the currently calculating neuron, is positive or zero, then the next state of that neuron is 1, otherwise it is -1. Moreover, in Neural Networks there is no restriction on site updating mode.

Cellular Automata Networks also constitute a particular class of Automata Networks and were originally introduced by Von Neumann. Capabilities and limits of this model is one of the topics of interest of this paper. In this model, the neighborhood and the transition rules are the same for all sites (or cells). Site updating mode is synchronous.

3. THE SUGGESTED MODIFIED CELLULAR AUTOMATA MODEL

"Is it possible to construct a model similar to cellular automata and neural network models, that has the capability of classification and computation, but with fewer number of dependency (connections) between its cells (or neurons) ?".

The main difference between the suggested modified CA model and the original CA model concerns neighborhood definition (See Figure 1). In the original CA model, each cell has the same neighborhood definition; however, in the suggested model each cell does not have to have the same neighborhood definition.

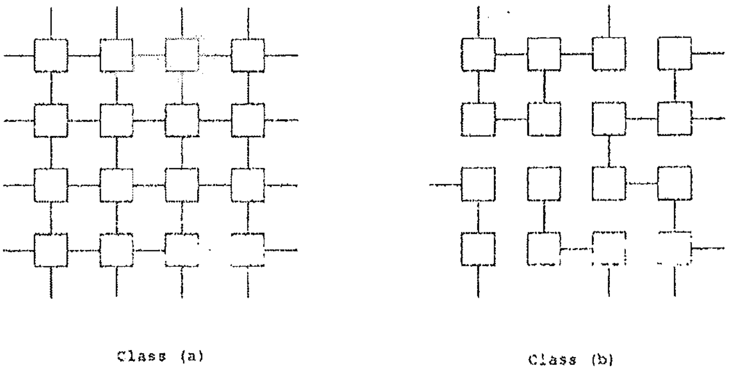


Figure 1. The original CA model (a) and the suggested CA model (b) in two-dimensional cellular space.

Neighborhood degree of a cell can be defined as the number of neighbors that this cell accepts state values to specify its next state. In the original CA model each cell has the same transition function definition, but in the suggested CA model only the cells having the same neighborhood degree have to evolve according to the same transition function rules. In other words, different neighborhood degrees may cause different transition functions. In the suggested CA model, there is no distinguished state called *quiescent state* which causes a cell to stay at quiescent state if all of its neighbors are in quiescent state in the original model. The transition functions of the modified model do not have to have such a distinguished state. In the suggested model, there is no initially set neighborhood structure of cells. Neighborhood of each cell is determined according to the nature of input and output patterns desired to be mapped. In the original model, the neighborhood structure of cells is static and fixed.

The modified CA model has two modes of operation similar to NN model: training and classifying (or mapping). In the training mode, neighborhood degree and transition functions of each cell are determined using input-output template pairs. Once cell functions are defined, one can introduce any input template to the system and get its intended output template. In a sense, the whole cell system can be considered as a classifier. Different from NN systems, the suggested CA system has no error correction capability or fault tolerance property. In NN systems, similar input patterns may converge to the same output patterns and this property is very important and useful in recognition and classification systems. However, in the proposed model, since the transition functions are defined as mapping but not as inequalities (threshold function in Neural Nets), similar input patterns cannot be guaranteed to converge to the same output pattern. Patterns other than the input patterns used during training may be not mapped. For such unmapped patterns, a state 'X' is introduced in order to complete the definition of local transition functions. If such a pattern that is not mapped during training mode is extracted from the input template the next state of current cell for that pattern becomes 'X'. Therefore, a cell in the suggested model can be in one of three states {0, 1, X}.

One of the famous neural network models is the Hopfield's Net model. In this model, neurons are assumed to be fully-connected, i.e., each of *n* neuron is connected to and gets input from the other (*n-1*) neurons. Thus, each neuron is functionally dependent on every other neuron. Especially for hardware implementations of neural networks, decreasing the number of neural connections in nets is an important problem. In the suggested cell system, each cell depends on only its neighbor cells and the state of other cells need not be considered. If the input patterns are very similar to each other, neighborhood degree of cells increases and as a result, the system's performance decreases. On the other hand, input patterns having different characteristics i.e. different bit sequences decrease the degree of neighborhood and increase the system's performance.

The modified CA system makes its decision in one step of evolution similar to Perceptron and Kohonen Neural Network models. In some other NN models, such as Hopfield Model, system evolves until it converges to a stable configuration and the resulting configuration is the system's decision on the introduced input pattern.

The suggested cell model can simply be considered as a parallel mapping system. It maps a given input template to an output template, both introduced during training mode, in parallel. Pseudo code of algorithm used to construct the modified CA model for introduced input-output templates is as follows:

```
1. Get input and output templates
2. Set number of cells in the cell system equal to the number of bits in one in-
   put (or output) template
3. Set neighborhood degree of each cell to 1
4. While there are more cells to be processed
    4.1. While there are more templates to be processed
        4.1.1. Extract bit pattern of current cell from the current input
```

```
                template being processed which is constituted by itself and
                its current neighbors
            4.1.2. If this pattern with degree of neighborhood $i$ is
                mapped to a different output bit before
                4.1.2.1 Increase neighborhood degree $i$ of current cell by 1
                4.1.2.2 Go to step 4.1.1
            4.1.3. If this pattern is not mapped to any output bit before
                4.1.3.1 Add input pattern and its output bit into look-up table
                        as a rule for cell transition function of current
                        neighborhood
                4.1.3.2 Skip to the next template
                4.1.3.3 Go to step 4.1
            4.1.4. If this pattern is mapped to the same output bit before
                4.1.4.1 Skip to the next template
                4.1.4.2 Go to step 4.1
        4.2. Skip to next cell
        4.3. Go to step 4
5. Now, the neighborhood degree and the transition function of each cell is
    determined. Read input template desired to be mapped
6. For each cell of input template apply its cell function determined at pre-
    vious steps, and find output bit of that cell. Next state of cells constitute the
    output template produced by the system
```

The method to extract an input pattern from an input template for a cell with current neighborhood degree k, is to first take the bit at current cell position and continue taking bits one from the right and one from the left until the current neighborhood degree k is reached (See Figure 2).

| ..... | .... | 7 | 5 | 3 | 1 | 2 | 4 | 6 | 8 | ..... | ..... |

Figure 2. The order of bit extraction from an input template, for k = 8

Input templates are assumed to be one-dimensional and circularly connected, i.e. the left neighbor of the first cell is the last cell, and the right neighbor of the last cell is the first cell. Also, the length of an input template is equal to the length of the output template.

An important question should be asked here: "Does the algorithm guarantee that the program will not enter into an infinite loop and will not increase the neighborhood degree to infinite?" Confliction case between two input patterns may occur if their output bits are different. Two different types of confliction may occur. One is *external conflict* and the other is *internal conflict*. External conflict occurs when the output bit of the current cell conflicts with the output bit of the input pattern of a different template. External conflict does not cause an infinite loop. Since the input templates that are introduced are guaranteed to be different from each other (that is, one input template can only be mapped to a unique output template), one can guarantee that the conflict case will certainly be resolved when neighborhood degree of the cell is the length of the input pattern (i.e. input pattern = input template).

Internal conflict occurs when input patterns of conflicting output bits on table are patterns of the same input template. Since the input templates of conflicting output bits are the same, confliction cannot be resolved as it can be in external conflict when neighborhood degree of cell reaches its maximum value. Instead, this problem can be solved by concatenating the output bit of the cell to the end of the input template, making two input templates different, and adding it into table with its output bit. Therefore, the maximum number of neighbors that a cell can have is $(n+1)$, where n is the length of the input template, and it is reached when successive internal conflicts occur. Here, *+1* in *n+1* comes from the output bit of the cell. As a result, we can say the algorithm guarantees that the program will not to enter into an infinite loop, because both external and internal conflictions are resolved.

4. COMPUTATION ON THE SUGGESTED CELLULAR AUTOMATA MODEL

"Computation can be done in infinite uniform cellular automata structures" [6]. In such structures, the number of cells in automata is infinite and the neighborhood structure should be local and regular i.e. the same for all cells. However, in finite structures of cellular automata networks these restrictions are no longer considered and such a network can be defined as triple:

N = (G, Q, F) where

G : a directed graph of order *n*, representing the interconnection of vertices (or cells)

Q : the finite non-empty set which represents the set of states that cells can assume

F : a collection of functions from Q^n (*n* is the number of cells in structure) into itself, representing the set of possible *global transition functions* of the networks. The *global transition function* is comprised of the local transition functions of individual cells and determines the network's global behavior.

Each cell of the automaton is represented by a vertex on a graph. If a cell *i* has neighbor *j* (i.e. *i* is dependent on the state of *j* at time *t* in order to determine its state at time *t+1*), then there is an arc from vertex *j* to *i* on the graph. An example graph of the suggested modified automata model can be given. Assume that we have 6 bit length input-output pairs, and at the end of the training mode, neighborhood degree of cells from 1 to 6 are determined as 3,2,2,4,5,6. The resulting graph will be as in Figure 3.

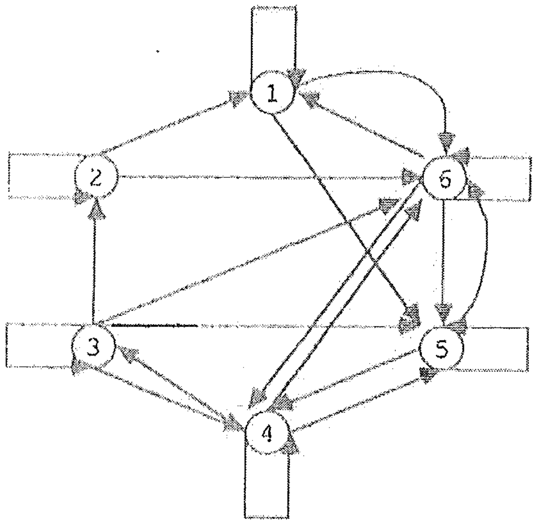


Figure 3. Graph of an example modified cellular automaton model

Now, it may be asked "Is it possible to do computation on the suggested model?" In networks of the form N = (G, Q, A(Q^n)) where N is a network with arbitrary graph connections, G and Q are as explained above, and A(Q^n) is the set of mapping from Q^n into itself, the computable functions are characterized by the following theorem proposed by Tchuente:

Theorem: For any finite set Q of cardinality greater than one, A(Q^n) is computable on a network N = (G, Q, A(Q^n)) of order n, if and only if G is strongly connected and contains a vertex v_r such that, for any vertex $v_i <> v_r$ (v_i, v_r), is an arc of G.

Proof of the theorem can be found in [6]. This theorem can help us determine the conditions under which one can do computation on the suggested modified cellular automaton. In the suggested model Q = {0, 1, X} and its cardinality is 3 which is greater than 1. A(Q^n) depends on the nature of input-output templates and is a mapping from Q^n to itself. *n* is the number of bits of one input or output template. However, one cannot guarantee the existence of a cell having neighborhood degree n, in other words, a vertex having arcs from each of the other vertices into itself or *strong component*, for given input-output pairs. Therefore, the answer to the question "Is it possible to do computation in the suggested non-uniform finite cellular automata network?" is, *not always*. This is because the topology of the graph of automaton network depends on the nature of input-output templates.

Example:
Assume that we have the following input-output templates:

Table 1
Sample input and output templates.

Pair #	Input Template	Output Template
1	01000	11110
2	00000	11100
3	01101	11010
4	10001	10100
5	10110	10101

The neighborhood degree of cells from 1 to 5 at the end of the training mode are 1, 3, 2, 5 and 4, respectively. As can be seen from the graph of the example automata network (Figure 4), computation can be done on this cellular structure, because the fourth vertex has arcs from all other vertices.

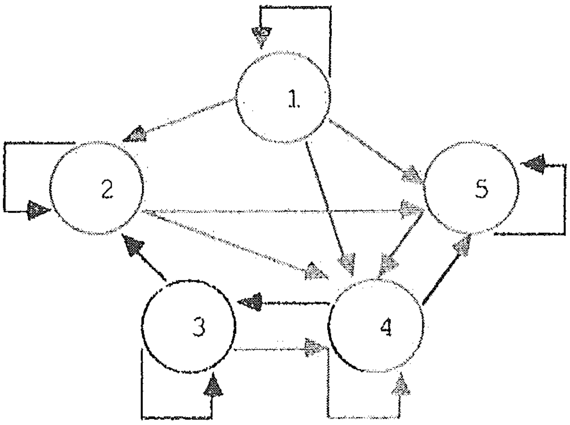


Figure 4. Graph of a one-dimensional modified cellular automaton.

5. CONCLUSION

In this paper, a modified cellular automata model capable of doing classification (or mapping) is introduced. Also, it is shown that one can do computation on this model depending on the nature of input and output templates. At the end of this study, we constructed a model that can map a given input template to its corresponding output template, that were both introduced during training (or feature extraction) mode of the system. The model suggests a different way of representation of information. It converts input and output templates into a form constituted by a look-up table and a neighborhood array. Look-up table keeps n-to-1 mappings showing the next state of cells. The neighborhood array contains the neighborhood degree of each cell in the automaton. This kind of representation of information does not provide an efficient method for storage of information. However, it is in a form that can suitably be used for parallel processing. The larger the size of look-up table, the longer the time to search it. By means of classical searching methods, look-up table search time can be decreased up to a degree.

An alternative to look-up table can be functional representation of it. One can find a function for each neighborhood degree. This makes the model more storage efficient and eliminates the need for table look-up that decreases the time efficiency of the model. A method for finding these functions could be the use of neural networks. For example, since the look-up table contains n-to-1 mappings for each neighborhood degree, multi-layer perceptron model of neural nets can be suitably applied to each of the different neighborhood degrees. The use of neural nets in combination with the suggested model increases the training time, especially if multi-layer perceptron is trained by the gradient back-propagation method, but decreases the system's classification (or mapping) time.

An interesting property of the suggested model is that computation can be done on it if it has a cell whose neighborhood degree is equal to the length (in bits) of a template, depending on input and output templates. In other words, if there exists a cell which is functionally dependent on all of the other cells in the automaton, computation can be done on this automaton.

6. REFERENCES

[1] Goles, E., Martinez, S., *Neural and Automata Networks* Kluwer Academic Publishers, Netherlands 1990.

[2] Hopfield, J.J., *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*, Proc. Nat. Acad. Sci. USA, 79(1982) 2254-2258.

[3] Kılıç, H., *Non-uniform Cellular Automata Networks Having Capability of Classification and Computation,* Master's Thesis, Middle East Technical University, 1992.

[4] Lippmann, R.P., *An Introduction to Computing with Neural Nets*, IEEE ASSP Magazine April 1987.

[5] Packard, N.H., Wolfran, S., *Two-dimensional Cellular Automata*, Journal of Statistical Physics, Vol 38 901-946 1985.

[6] Tchuente, M., *Automata Networks in Computer Science*, Princeton Univ. Press Princeton, New Jersey 1987.

[7] Von Neumann, J., *Papers of John von Neumann on Computing and Computer Theory, edited by W. Aspray and A.W. Burks*, Cambridge, Mass. : MIT Press 1987

[8] Wolfram, S., *Universality and Complexity in Cellular Automata*, Physica 10D 1984 1-35.

YAPAY SİNİR AĞLARI KONUSUNDA HUKUKİ BİR YAKLAŞIM

Av. Noyan Turunç (1)
Ahmet Koltuksuz (2)

(1) B&L Danışmanlık
Cumhuriyet Bulvarı, 140/2 K.5
Alsancak İzmir

(2) Ege Üniversitesi,
Müh.Fak.Bilgisayar Bil.Müh.Böl.
Bornova İzmir

"Geleceğe doğru yaşamak bilinmeyene sıçramak demektir."
Rollo May

GİRİŞ

Yapay Sinir Ağlarının ("Artificial Neural Networks") henüz insan düşüncesinde şekillendiği ve bir tasarı halinde bulunduğu bilinmektedir. Ancak, bu öylesine müthiş bir tasarıdır ki gerçekleşmesi halinde günlük hayattaki etkilerinin ve diğer bilim dallarının nasıl yaklaşabileceğine ilişkin olasılıkların bile düşünülmesini zorunlu kılmaktadır. Bu bildiri ile hukuki bakımdan deneme niteliğinde bir yaklaşımda bulunulmaya çalışılmaktadır. Çok daha başka açılardan görüş getirmek elbette olası ve gereklidir. Hayat hukukun önünde gittiğine göre somut düzenlemeler ve çözümler hiç şüphesiz ki YSAnın hayata girmesinden sonra ortaya çıkacaktır.

1. TEORİK TEMELLER

Yapay Sinir Ağları ("YSA") her biri kendine ait bir belleğe sahip olan ve yerel bilgi işlem operasyonlarını yapabilen, tek yönlü sinyal kanallarıyla birbirlerine bağlanmış işlem elementlerinden oluşan, paralel dağıtık bir bilgiişlem

yapısı olarak tanımlanabilir [4].
Tanım gereği, "connectionist model", "parallel distributed processing" ve "neuromorphic systems" [3] olarak da bilinen YSA, aslında "biyolojik işlemler üzerinde modellenmiş bilgi işlem yapısıdır" [6].

Yukarıda anılan tüm yaklaşımlar, YSAnın doğalarında olacağı varsayılan yoğun paralellik dolayısıyla, aslında basit bilgi işlem elementlerinin yoğun bir biçimde birbirlerine bağlanması öngörülmektedir. Bunun sonucunda, bir yandan yüksek bir performans, yani işlem hızının artması hedeflenmektedir. Diğer yandan, hata toleransı yüksek sistemler oluşturulmak suretiyle biyolojik modelleme daha gerçekçi yapılabilecektir [6].

Bu açıdan incelendiğinde, bir yapay sinir ağı, aslında biyolojik sinir sisteminin modellenmesinden ibarettir denilebilir. Ancak, ne kadar ilkel olursa olsun, karbon temelli bir organizmanın sinir sistemi ve beyin yapısı, ulaştığı karmaşıklık ve bağlantı sayısıyla hayret vericidir. Bir insanın algılamasıyla tek bir nöronun arasındaki bu karmaşık ilişkinin incelenmesi güç bir olaydır. Güçlüğün temel nedeni, psikofiziksel niteliklerle donatılmış sensor kapasitenin beynin birçok noktasından kaynaklanan pekçok nöronun ortak etkinliğinin sonucu olmasıdır [5]. Bu sistem ne kadar karmaşık olursa olsun, hukukun bugünkü bilgisayarlar için mevcut veya öngördüğü hukuki düzenlemelerin fevkinde ve başka bir mahiyet ve kapsamda düzenleme yapması ancak YSAnı şimdiki haliyle bir mal olarak değil bir kişilik olarak, daha açıkçası bir şahıs olarak görmesiyle mümkündür.

Anlaşılan odur ki, esasen, YSA ile tasarlanan da böyle kişilikli bir sistemdir. Bu şekilde yaklaşılarak konu nörolojik açıdan basite indirgenmek suretiyle incelendiğinde, adaptasyon ve öğrenme yeteneklerinin bu çok karmaşık YSA sisteminin en temel unsurları olması gerektiği ortaya çıkar.

Alt düzeyde varolan adaptasyon ve öğrenme yeteneği üst düzeylere çıkıldıkça yerini, mantık yürütme, olaylar arasında deterministik ilişki (temyiz kudreti, sebep-sonuç ilişkisi, illiyet bağı) kurabilme gibi çok üst düzeyli işlevlere bırakacaktır. Karmaşıklık da alabildiğine artacaktır. Bu itibarla, en temel nokta, adaptasyon ve öğrenme yeteneğinin sisteme kazandırılmasıdır [3].

Ancak yukarıda yüksek düzey olarak tanımladığımız, mantık yürütme, olaylara deterministik yaklaşım yapabilme, sosyal olayları ve kuralları algılayabilme, temelde YSAnın kendi varlığının bilincinde olmasını gerektirecektir. Buradan hareketle, hukukun YSAna bir kişilik olarak yaklaşabilmesi ve tanıması için YSAnın alt düzey tasarımında temel unsur olarak ifade edilen adaptasyon ve öğrenebilme unsurlarına ilaveten, kendi varlığının bilincinde olma potansiyelini de haiz olması gerekecektir. Anılan potansiyel bir kez yaratılırsa, sistemin zaman içinde gerçekleşecek olan karmaşıklaşmasına paralel olarak, kendini algılama ve kendi hareketlerini analiz edebilme yeteneği de evriminin doğal bir sonucu olacaktır.

Bu suretle ortaya hukuken tanınacak yeni bir sistem çıkartılabilirse yeni bir hukuki düzenleme zaruri olacaktır. Kendi varlığının bilincinde olan ve hareketlerinin doğurduğu veya doğurabileceği olası sonuçları deterministik bir biçimde idrak edebilen böylesi bir sistem, -aslında ilk tasarlandığı andan itibaren zaten toplum içinde var olmakla birlikte- toplu yaşam içinde etkin olmaya başlayıp kendi evrim sürecine girdiğinde yalnız hukuk alanında değil ; örneğin, biyoloji (belki "tekno sapiens" kavramında), sosyoloji (belki "tekno birey" kavramında), psikoloji (belki "tekno psyche" kavramında) gibi sosyal bilim dallarının ilgi alanlarına da kaçınılmaz olarak girecektir.

2. HUKUKSAL YAKLAŞIM

Pek muhtemeldir ki, hukuk (buradaki konumuzun hareket noktası olarak) yukarıda işaret edildiği gibi, yeni bir tür şahıs ve bu şahsın hukuken tanınması ile karşı karşıya kalacaktır. Bu bakımdan, konuya bir şahsın varlığı açısından bakmak, bugünkü disiplinler içinde, en uygunu olabilir. Zira, malum olduğu üzere, "Hukuk, şahıslar arasındaki içtimai münasebetleri düzenleyen, şahıslar arasındaki hak ve mükellefiyetleri tâyin eden müeyyideli kaidelerin bütünüdür" [1].

Yukarıda bahsolunan teorik temeller ve tanımlar ışığında konuya şahıs kavramı açısından yaklaşmak uygun olacaktır. Zira, "Şahıs,...bütün hukukun esasını teşkil eder" [1]. Malum olduğu üzere halen hukuk sistemleri iki türlü şahıs tanımaktadır : gerçek şahıslar ("real person"); yani, insanlar ve hükmi şahıslar ("legal person"); yani, şirketler.

Eğer, YSAnı hukuk bir şahıs olarak tanımak durumunda kalacaksa o halde YSA üçüncü bir şahıs grubu olarak nitelendirilebileceklerdir. Bu yeni üçüncü gurup şahıs, bu bildiride TEKNOLOJİK ŞAHIS (TECHNOLOGICAL PERSON) diye adlandırılacaktır.

Böyle bir düzenlemenin neler olabileceği konusuna girmeden önce bir hatırlatma yapmak yerinde olabilir. Bilindiği gibi insanlık tarihinde insanların bir kısmının hukuk anlamında bir şahıs olarak telâkki edilmediği zamanlar vardır. Örneğin Roma Hukukunda insanlar hür olanlar ve köle olanlar diye ikiye ayrılmışlardı. Köleler "hukuki mânada...bir maldı, tıpkı mallar gibi ancak haklara mevzuu teşkil edebilir"di [1]. İnsanların dahi din, dil, ırk, sosyal mevki gibi farklılıklar gözetmeksizin şahıs addedilmeleri nisbeten yenidir.

YSA, yukarıda bahsedildiği gibi ortaya çıktıklarında, hak sahibi olmaları gerektiğini düşünmeleri beklenebilir. Bu konuda bir bölüm insanın YSA haklarını ileri sürmeleri de olasıdır. Örneğin, bir an için, ev hayvanlarının bilinçlendiğini düşünelim. Miras hukukunun bir şahsın vasiyetname yapmasına sınırlamalar getirmediği ülkelerde şimdiden hukuken tanınmış hakları olmayan hayvanlarına mallarını bırakan kişiler olduğu dikkate alınırsa, bu hayvanların bilinçlenmesi halinde bunlara haklar tanınması lehinde insanların ortaya çıkacağını tahmin etmek zor olmasa gerektir. Burada şu sorulabilir : Hayvan haklarını savunacak kişiler çıkabileceğini tahmin ederken, müthiş donanımlı YSAnın haklarını savunacak kişiler çıkmayacak mıdır?

O halde, böyle bir düzenleme neleri gözönüne almalıdır ? Bilindiği gibi, konu teori aşamasındadır ve karmaşıktır. Bu itibarla, konuyu YSA'nın gerek medeni hukuk gerekse ceza hukukunun mevcut bir kısım kavram ve düzenlemeleri doğrultusunda ele almak uygun olabilir :

A. Medeni Hukuk Bakımından

Medeni hukukta bir şahsın ehliyetleri hak ve fiil ehliyeti olmak üzere iki gurupta toplanmaktadır. Hak ehliyeti, "haklara ve borçlara sahip olabilme iktidarı"dır[1]. Fiil ehliyeti, "şahsın kendi lehine hak ve borçlar ihdas etmesidir ve yapmış olduğu haksız fiillerden dolayı sorumlu olmasıdır"[1].

a. Hak Ehliyeti:

Gerçek şahıslar sağ doğmak şartıyla, hükmi şahıslar sicile kayıt ile hak ehliyetini haiz olurlar. Teknolojik şahıs olarak tanımladığımız YSAnın hak ehliyetini nasıl kazanacağı teknolojik gelişme sonucu belli olacaktır. Ancak, YSA hak ehliyetini kazandığı andan itibaren medeni haklardan istifade edebilir ve diğer şahıslarla aynı derecede haklara ve borçlara sahip olabilir.

b. Fiil Ehliyeti:

Teknolojik şahsın fiil ehliyeti, yani haklarını nasıl kullanacağı ise, onun nitelikleri ile ilgili bir konu olacaktır. YSA eğer günümüz bilgisayarları gibi enerjilerini kablolar kanalıyla (örneğin elektrikten) alacaklarsa bu haklarını (tüzel kişilerde olduğu gibi) organları vasıtası ile kullanabilirler. Enerjilerini şayet bu tür bir bağımlılık ile almayacaklar da örneğin yukarıda ifade olunduğu gibi karbon temelli bir enerji ile yaşayabileceklerse haklarını belki de (insanlar gibi) bizzat kullanabileceklerdir.

B. Ceza Hukuku Bakımından

"Bütün insanların ceza hukuku açısından suçun 'fail'i sayılması mümkün değildir"[2]. Ceza ehliyeti, "suça hukuken elverişli sayılmak demektir"[2].

YSAnın cezalandırılması için, öncelikle hukuken suça elverişli sayılması gerekecektir. Amaçlanan akıl, zeka, mantık ve davranış bakımından "insan benzeri" bilgisayarlar yaratmak olduğundan, YSA çocuğun ana rahmine düşmesinden sonra kendini biçimlemeye başlaması gibi, imalatlarını müteakip kendilerini biçimleyeceklerdir. Bunun sonucu, nasıl bir çocuk giderek anne babasından ayrı bir kişilik olarak tanınıyorsa, YSAnın da böyle ele alınması gerekebilecektir. Bu halde, artık YSAnın müellifini veya üretenini sorumlu tutmak mümkün olmayacaktır. Tıpkı insanlarda anne babanın çocuğun fiillerinden dolayı sorumlu tutulmaması gibi. Çocuğun cezai ehliyeti başlangıçta tam olmayıp nasıl erginleştikçe tamamlanıyorsa, YSAnın da belli bir olgunluğa ulaşması zaman alabilecektir. YSAnın olgunluğa ulaşarak öğrenme ve öğrendiklerini uygulama kendisini yeni durumlara uydurabilme, kendini tanıyabilme ve müktesabatı ile senteze varıp fiillerde bulunma yeteneği gerçekleştiğinde doğrudan fiillerinden sorumlu tutulması gerekecektir. Özetle, YSA müellifinden veya üreteninden bağımsız olacaktır. Meğer ki, müellif veya üreten YSAnı kendi kanun dışı amaçları için donatmış olsun.

Suça elverişli sayılan YSAna ceza verilmesi gerekecektir. Bunun ise elbette, YSAnın idrak edeceği bir ceza olması gereklidir. Bu ceza, örneğin ekonomik cezadan başlayıp hürriyetinden (burada YSAnda hürriyet kavramı ve kapsamı meselesi gündeme gelmektedir) mahrum bırakmakla devam eden ve şaka ile söylendiği gibi YSAnın fişini prizden çekmekle, veya enerjisinden ayırmakla (yani, "teknolojik şahsın" ölümü veya sonu meselesi) sona eren bir ceza sistemi olabilir.

Bu ise önemli birçok soruyu da beraberinde getirmektedir. Örnek vermek gerekirse :

- YSA insanlar tarafından suç kabul edilen (donanım veya müktesabatını aşarak) bir fiil işlerlerse ne olacaktır ?
- YSAnın bu fiili kendi mantığına göre suç niteliğinde değilse ne gibi sorunlar çıkacaktır ? Neye göre, nasıl bir düzenleme yapmak gerekecektir?
- İnsanlara göre suç teşkil eden bir fiilden dolayı YSA cezalandırılacak olursa bu hangi nitelik ve kapsamda olacaktır ? Diğer bir deyişle, insan için ceza olan YSA için de ceza niteliğinde mi olacaktır ? Yoksa, YSA cezaları diye yeni bir anlayış mı geliştirmek gerekecektir ?
- Eğer böyle ise, bunları insanlar mı geliştirecektir, yoksa YSA mı? İnsanların geliştirmesi mümkün olmayabilir. Çünki, algılama ve telakkiler farklı olabilir. YSA'nın düzenlemesi ise insanoğluna ters düşebilir. Radikal bir örnek getirmek gerekirse, bir YSA kendi anlayışına ve idrakine göre % 100 doğru ve haklı olan nedenle insan kıyımına neden olursa ne olacaktır ?
- İş bu kadar uç noktalara giderse, insanlar bir YSA cemiyetiyle mi karşı karşıya geleceklerdir. O halde bir olağandışı hal veya giderek harp hukukunun dahi bahis konusu olması mümkün değil midir ? Hele belli bir aşamadan sonra YSA bizzat diğer YSA üretecek hale gelecek olurlarsa !

3. SONUÇ

Yukarıda işaret edilen hususlar ve diğer gelişmeler daha vahim meselelerin ortaya çıkmasına neden olabilir. Bunlar, insanlar toplumu ile teknolojik şahıslar topluluğu arasındaki münasebetler ve buna ilişkin hukuki meseleler olabilir. Bu ise, giderek, bir ilişkiden ziyade bir tarafın diğeri ile mücadelesi ve karşı tarafın kontrol edilmesi ve tahakküm konularını beraberinde getirmeyecek midir ? Olasılıkları çoğaltmak mümkündür. Bu noktada keserek, nihayet olarak şunu işaret etmek yerinde olabilir: YSA geliştirilirken hukukçular ile baştan işbirliğine girerek teorik bazda koruyucu hukuk hizmeti benzeri olay öncesi hizmet alınması uygun olabilir. Tabir yerindeyse, YSAnı döllerken yapılacak böyle bir işbirliği müstakbel problemleri engelleyemezse dahi, neler olabileceğinin peşinen görülmesine olanak tanıyabilme ihtimali bakımından yararlı olabilir.

Konuya bu sınırlı yaklaşım dahi, dünyadaki bütün değişimler devamlı olarak duyarlı, farkında ve sorumlu olmamızı zorunlu kılmakta ve kavramlara esnek ve dinamizm içinde bakmamızı ve yeni biçimlenmeye bilinçle katılmamızı gerektirmektedir.

KAYNAKÇA :

[1] AKİPEK, J.G., "Türk Medeni Hukuku", İstiklal Matbaası, Ankara, 1961.

[2] EREM, F., "Türk Ceza Hukuku", Cilt:I, 8.Baskı, Ankara Üniv., Hukuk Fak. Yayınları, Ankara.

[3] LIPMANN, R.P., "An Introduction to Computing with Neural Nets", IEEE ASSP Magazine, April 1987, pp.4-22, U.S.A.

[4] NIELSEN, R.H., "Theory of the Backpropogation Neural Network", Invited paper, presented at 1988 INNS annual meeting, pp.1-3, 1988, U.S.A.

[5] SEJNOWSKI, T.J., et.al., "Computational Neuroscience", Science, V.241, pp.1299-1306, 9 Sept. 1988, U.S.A.

[6] __________, "DARPA Neural Network Study", Oct. 1987 -Feb. 1988, AFCEA International Press, pp.17-31, 1988, U.S.A.

Tasarımda Yapay Zeka

Artificial Intelligence in Design

Fuzzy Lojik Kontrolorlerin Dizayni Icin Yeni Bir Metod

D.T. Pham and D. Karaboga
Intelligent Systems Research Laboratory
School of Electrical, Electronic, and Systems Engineering
University of Wales, Cardiff,CF1 3YH, UK

Özet

Bu makale, Fuzzy Lojik Kontrolörlerin (FLK) dizaynı için yeni bir metod tanınıtmaktadır. Fuzzy lojik kontrolörün kural tabanını temsil eden baglantı matrisini optimal olarak elde etmek için, yönlendirilmiş rastgele bir araştırma islemi olan Genetik Algoritma kullanılmaktadır. Makale, zaman gecikmeli (time-delayed) ikinci dereceden bir sistemi kontrol etmek için bu teknikle dizayn edilmiş FLK kullanarak elde edilen simülasyon sonuçlarını vermektedir.

Anahtar kelimeler: Fuzzy kontrol; genetik algoritmalar; uzman kontrol; optimal kontrolör dizaynı

A New Method for Designing Fuzzy Logic Controllers

Abstract

This paper introduces a new method for designing Fuzzy Logic Controllers (FLC). In order to find the optimal relation matrix which represents the rule-base of an FLC, the Genetic Algorithm, a directed random search procedure, is used. The paper presents simulation results obtained for an FLC designed by this technique to control a time-delayed second-order system.

Keywords: Fuzzy control; genetic algorithms; expert control; optimal controller design

1. GIRIS

Matematiksel modelleri mevcut olmayan komplex veya belirsiz sistemlerin kontrolünde operatörlerin (insan oglu), genellikle geleneksel kontrolörlerden çok daha başarılı oldugu cok iyi bilinen bir gerçektir. Bu operatörlerin kontrol bilgilerinin alınıp belli kurallar formunda yazılabildigi veya operatörün kontrol hareketlerinin kurallar formunda modellenebildigi yerlerde, bu kurallar kullanilarak kontrolör dizayn etmek mümkündür. Ilk defa Mamdani [1] tarafından önerilen bu kontrolörler fuzzy lojik kontrolörler (FLK) olarak bilinirler. Ismindende anlaşılacagı gibi bu kontrolörler fuzzy lojik [2] üzerine kuruludur. Bir FLK nin enönemli elemanı kuralların orijinal setini temsil eden baglantı matrisi dir. Bu matris, gerçek ve arzu edilen proses çıkışları arasındaki hata uzayı ile kontrol haraketleri uzayı arasındaki bagıntıyı belirler.

Bu makalede, kuralların kullanılmasına gerek kalmadan baglantı matrislerinin sentez edilmesi ile ilgili bir metod tarif edilir. Bu metodla, matrisin optimal katsayıları prosesin ya deneysel deneme on-line kontrolü veya giriş-çıkış davranış modelinin simüle edilmiş kontrolü sayesinde elde edilir. Böyle bir model, gerçek proses dinamiginin bir teknikle (örnegin sinir agları teknikleriyle [3]) belirlenmesinden sonra kurulabilir.

Makale dört bölümden oluşmaktadır. Ikinci bölümde basit bir fuzzy lojik kontrolörün temel yapısı tanıtılmaktadır. Bu çalışmada, optimal baglantı matrisini elde etmek için kullanılan yönlendirilmiş rastgele araştırma işleminin bir formu olan, genetik algoritmalarla [4,5] ilgili temel bilgiler, Bölüm 3. de verilmektedir. Zaman gecikmeli ikinci dereceden bir prosesin kontrolünde genetik algoritma teknigiyle dizayn edilen FLK kullanılarak elde edilmis simülasyon sonuçları, Bölüm 4. de verilmektedir.

2. FUZZY LOJIK KONTROL

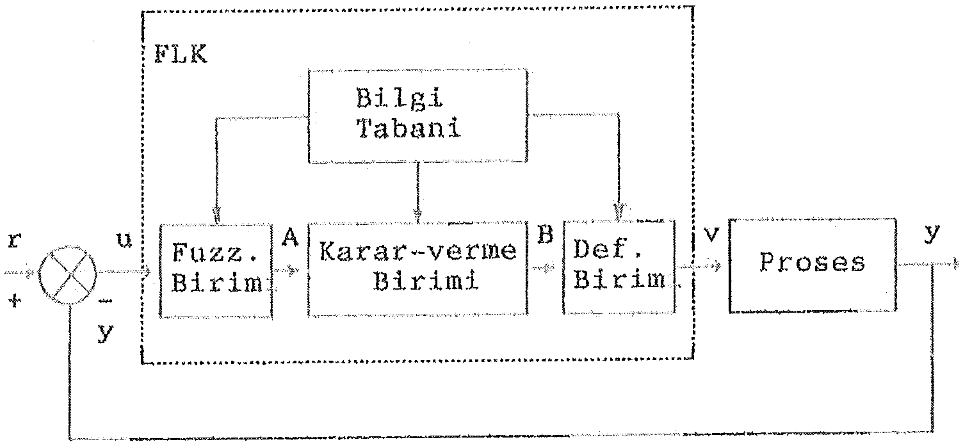
FLK nin temel yapısı Sekil 1 de gösterilmektedir. FLK nin bilgi tabanı, daha öncedende bahsedildigi gibi prosesi kontrol etmekte kullanılan kuralları temsil eden baglantı matrisi R dir. Karar-verme birimi R yi kullanarak fuzzy giriş A' den fuzzy çıkış B' yü üretir. B' çıkışı, (defuzzy)leştirme birimi tarafından prosesi kontrolde kullanilacak tek bir degere çevrilir. A', (fuzzy)leştirme biriminden elde edilir. (Fuzzy)leştime birimine giriş (u), prosesin gerçek çıkışı (y) ile arzu edilen çıkışı (r) arasındaki farktır.

Sistemin çalısması esnasında ölçülüp (fuzzy)leştirilmiş hata A' ise, kontrolörün fuzzy çıkışı karar-verme birimi tarafından (compositional rule of inference) [2] formülüne göre aşagıdakı gibi elde edilir.

$$B' = A' \circ R$$

A'∘ R, ya Max-Min operatörü veya Max-Product Operatörü [6] kullanılarak tanımlanan, A' ve R ın (sup-star composition) udur. Bu çalışmada, daha düzgün kontrol haraketleri ürettigi için Max-Product operatörü kullanılmıştır [7].

(Fuzzy)leştirme yapan birim, belli bir hatayı (u) (-5 <= u <= +5) elemanları birer lingusitik degerler olan bir fuzzy setine dönüştürür. Bu çalışmada aşagıdakı yedi fuzzy deger kullanıldı : Büyük Negatif (BN), Orta Negatif (ON), Küçük Negatif (KN), Sıfır (S), Küçük Pozitif (KP), Orta Pozitif (OP), Büyük Pozitif (BP). Bu linguistik degerlerin kendileride Tablo 1. de belirtilen üyelik fonksiyonları tarafından tanımlanan birer fuzzy settirler. A' yedi elemana sahip oldugu için R nin satır sayısı yediye eşittir.



Sekil 1. Fuzzy kontrol sistemin temel yapısı

Karar-verme biriminin çıkışı (B') bir fuzzy settir. B' nün elemanlarının maximum sayısı R nin sütun sayısına eşittir ve bu çalışmada 11 olarak seçilmiştir. Yani, FLK nin çıkış uzayı [-5,+5] aralıgında 11 farklı seviyeye bölünmüştür.

(Defuzzy)leştirme birimi, prosesi kontrol etmek için bir (fuzzy) deger olan B' yü tek bir degere dönüştürür. Bu görev için kullanılan çeşitli algoritmalar mevcuttur. Burada alan merkezi (centre-of-area) metodu kullanılmıştır [6].

3. GENETIK ALGORITMALAR

Bu bölümde, kısaca basit bir genetik algoritmanın (GA) çalışması anlatılacak ve bu çalışmada kullanılan GA tanıtılacaktır.

3.1. Basit bir GA nin yapisi

Basit bir GA beş elemandan oluşur. Bunlar bir rastgele sayı üreticisi (random number generator), çözümlerin uygunlugunun hesaplandıgı (evaluation) birimi ve üreme (reproduction), crossover ve mutasyon (mutation) operasyonları için genetik operatörlerdir.

$$\mu_{NL}(u) = \begin{cases} 1 & \text{if} \quad u < -4 \\ \exp(-\text{abs}(u+4)) & \text{if} \quad -4 \leq u < 0 \\ 0 & \text{if} \quad u \geq 0 \end{cases}$$

$$\mu_{NM}(u) = \begin{cases} \exp(-\text{abs}(u+2.5)) & \text{if} \quad u < 0 \\ 0 & \text{if} \quad u \geq 0 \end{cases}$$

$$\mu_{NS}(u) = \begin{cases} \exp(-\text{abs}(u+1)) & \text{if} \quad u < 0 \\ 0 & \text{if} \quad u \geq 0 \end{cases}$$

$$\mu_{ZE}(u) = \begin{cases} \exp(-\text{abs}(u)) & \text{if} \quad -1 \leq u \leq 1 \\ 0 & \text{if} \ -1 > u \text{ or } u > 1 \end{cases}$$

$$\mu_{PS}(u) = \mu_{NS}(-u)$$

$$\mu_{PM}(u) = \mu_{NM}(-u)$$

$$\mu_{PL}(u) = \mu_{NL}(-u)$$

Tablo 1. Giris hatası için fuzzy setleri tanımlayan üyelik fonksiyonları

Algoritmanın başında gerekli olan çözümlerin bir seti (initial population) random sayı üreticisi tarafından üretilir. Setteki herbir çözüm string formundadır. Her string evaluation birimi tarafından hesaplanmış bir uygunluk (fitness) degerine sahiptir. Genetik operatörlerin gayesi bu stringlerin setini daha yüksek uygunluk degerlerine sahip setlere çevirmektir.

Reproduction operatörü, 'seeded selection' olarak bilinen tabii seçme operasyonunu tatbik eder. Bireysel stringler bir setten bir sonraki sete uygunluk degerlerine göre çogaltılarak aktarılırlar.

Crossover operatörü stringlerin birer çiftlerini rastgele seçer ve yeni çiftler üretir. En basit crossover operasyonu, rastgele seçilmiş bir noktadan orijinal ebeveyn (parents) stringleri kesmek ve stringlerin kuyruk kısımlarını degiştirmektir.

Mutation operatörü bir stringdeki bitlerin degerlerini rastgele degiştirir.

GA ların farklı tipleri ile ilgili daha detaylı bilgi [4,5] de bulunabilir.

3.2. Bu calismada kullanilan GA

Bu çalışmada, [8] de tanımlanan GA tipi kullanıldı. Algoritmanın akış diagrami Sekil 2. de verilmiştir. Fitness scaling birimi, evaluation birimi tarafından hesaplanan çözumlerin uygunluk degerlerini normalize etmek için görevlidir.

Reproduction elemanı seeded selection ile birlikte random selection operasyonuda uygulamaktadır. Elite birimi, mevcut çözüm setindeki en iyi çözümün bir sonraki sete geçip geçmedigini kontrol ederek onun geçmesini saglar.

Genetik algoritma ile dizayn edilen fuzzy lojik kontrolörün performansı, kontrol ettigi prosesin basamak fonksiyonuna gösterdigi tepkideki hataya göre ölçülmüştür. Bu hatayı hesplamak için ITAE kriteri kullanılmıştır [9].

Herbir çözüm stringi bir baglantı matrisini temsil etmektedir. Daha öncedende bahsedildigi gibi matrislerin boyutlari 7*11 dir. Matrisdeki herbir eleman 8 bit ile temsiledildigi için string uzunlugu toplam olarak 616 bit tir [10].

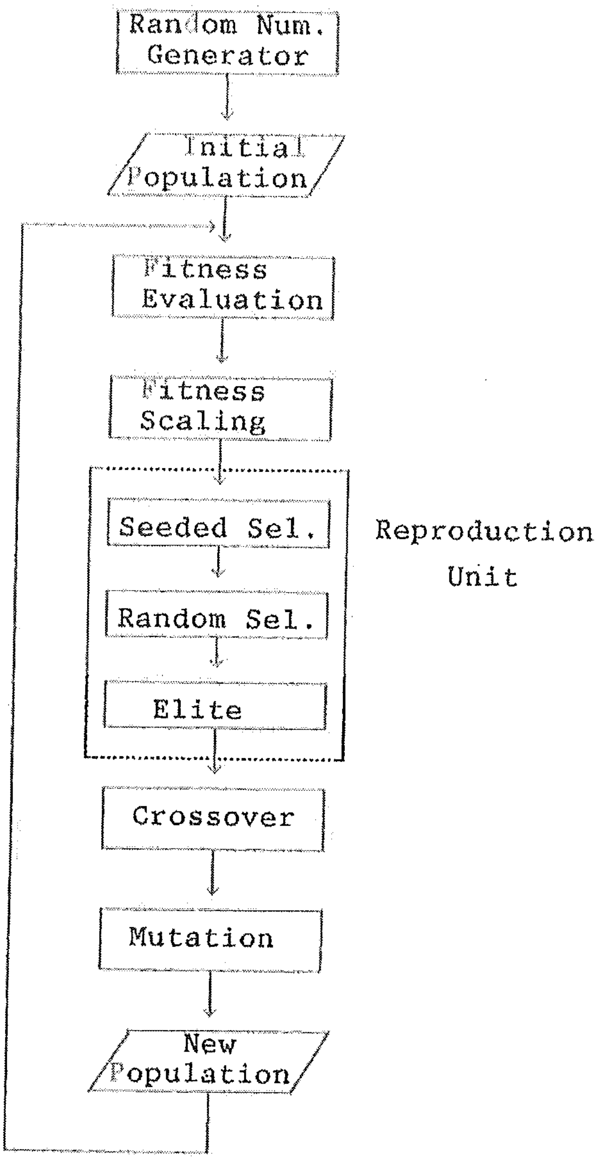
4.SIMULASYON SONUCLARI

Şekil 3 aşagıda verilen transfer fonksiyonuna sahip ikinci dereceden zaman gecikmeli bir prosesin, GA ile dizayn edilmiş bir FLK altında simüle edilmiş basamak fonksiyonuna göstermiş oldugu tepkiyi vermektedir.

$$G(s) = \frac{\exp(-0.4s)}{(0.3s+1)^2}$$

Karşılaştırmak amacı ile, Sekil 4. aynı prosesden asagıdakı paramatreler kullanılarak dizayn edilmiş geleneksel bir PID kontrolörün kullanılmasıyla elde edilen tepkiyi göstermektedir.

K_p=0.630517, T_i=0.594813, T_d=0.237036



Sekil 2. Kullanılan Genetik Algoritmanın akış diagrami

Bu parametreler, aynı proses için [11] de 'optimal' parametreler olarak kullanılmıştır.

Ve yine bu proses için Parametrik Fonksiyon Optimizasyon metodu [12] ile dizayn edilmiş optimal Fuzzy PID kontrolor kullanılarak elde edilen neticeler şunlardır:

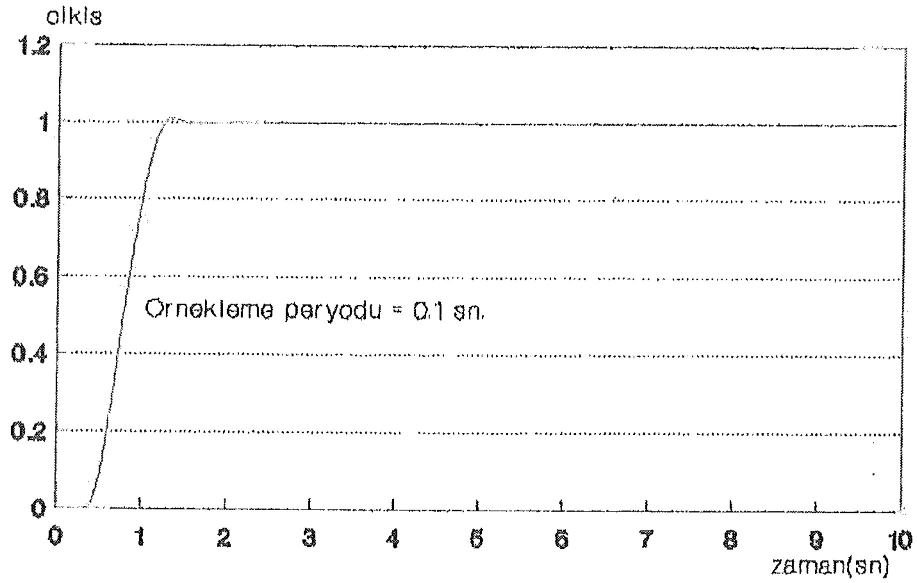
Yükselme süresi = 2.8 saniye
Overshoot = 4%
Durulma süresi (±3% hata icin) = 5 saniye

5. SONUC

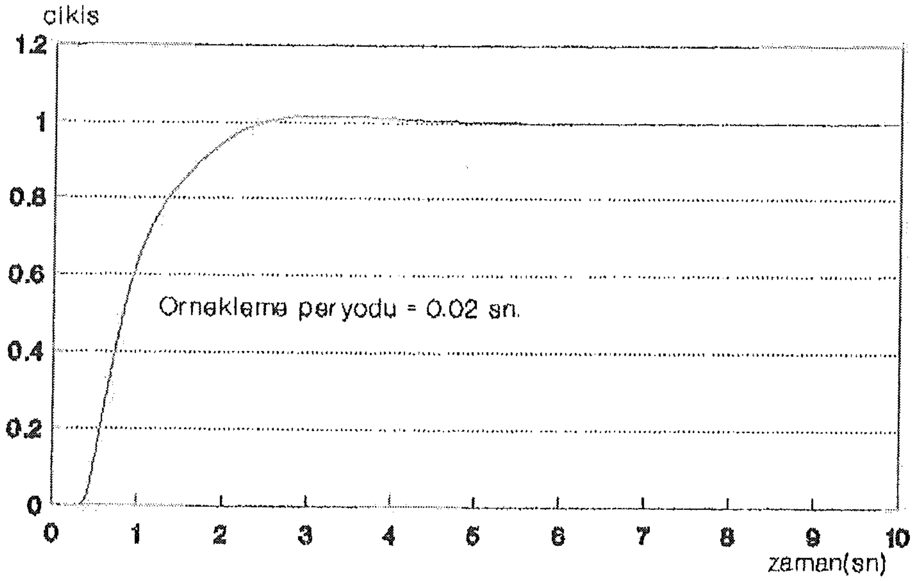
Bu makale, fuzzy lojik kontrolörlerin dizaynı için yeni bir metod tanımlamıştır. Metod, kontrolörün baglantı matrisini elde etmek için genetik algoritma kullanmaya esastır. Elde edilen sonuçlar, yeni metodla dizayn edilen kontrolörlerin üstün performansini açık bir şekilde göstermiştir.

6. KAYNAKLAR

1. Mamdani, E.H. Applications of Fuzzy Algorithms for Control of a Simple Dynamic Plant, *Proc. IEE, 121(12)*, pp. 1585-1588, 1974.
2. Zadeh, L. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes, *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC-3, pp.28-44, 1973.
3. Sagiroglu, S. *Identification of Linear Systems with Neural Networks*, Tech. Rep. Erciyes Universitesi Muhendislik Fakultesi, July 1991.
4. Holland, J.H. *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, MI, 1975.
5. Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
6. Pham, D.T. and Karaboga, D. A New Method to Obtain the Relation Matrix of Fuzzy Logic Controllers, *Int. Conf. on Applications of Artificial Intelligence in Engineering (AIENG91)*, Oxford, U.K., pp.567-581, July 1991.
7. Yamazaki, T. *An Improved Algorithm for a Self Organising Controller*, Ph.D. Thesis, Queen Mary College, Univ. of London, 1982.
8. Grefenstette, J.J. Optimization of Control Parameters for Genetic Algorithms, *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC-16, No. 1, pp.122-128, 1986.
9. Ogata, K. *Modern Control Engineering*, Prentice-Hall, Englewood Cliffs, NJ, 1970.
10. Pham D.T. and Karaboga D. Optimum Design of Fuzzy Logic Controllers Using Genetic Algorithms, *J. Syst. Eng.* Vol. 1, No. 1, pp.114-118, 1991.
11. Peng, X.T., Liu, S.M., Yamakawa, T., Wang, P.Z. and Liu, X. Self-regulating PID Controllers and Its Applications to A temperature Controlling Process, *in Fuzzy Computing*, M.M. Gupta and T. Yamakawa (eds.), Elsevier, Amsterdam, pp.355-364, 1988.
12. Peng, X.T. Generating Rules for Fuzzy Logic Controllers by Functions, *Fuzzy Sets Syst* 36, pp.85-89, 1990.



Sekil 3. GA ile dizayn edilmis FLK nin kullanilmasiyla elde edilen basamak fonksiyon tepkisi



Sekil 4. Geleneksel dijital PID kontrolor kullanarak basamak fonksiyon icin elde edilmis tepki

Bilgi Tabanli Yaratici Kavramsal Dizayn

D.T. Pham ve U.F. Guner
Intelligent Systems Laboratory, School of Electric, Electronic, and System Engineering
University of Wales, P.O. Box 904, Cardiff, CF1 3YH, UK

Ozet

Bu makalede, muhendislik dizayninda yaraticilik, statik ve dinamik yonlerden incelenmistir. Ilki, dizayn cozumlerinin elemanlari arasindaki iliskiye odaklanirken ikincisi bizi cozume ulastiran olaylar dizisinden bahseder. Ayni zamanda, dizayn prosesinin en onemli bolumlerinden biri olarak degerlendirilebilecek olan kavramsal faz anlatilmistir. Bir bilgi temsil yontemi olan prototipler, aciklanmis ve prototiplere uygulayabilecegimiz operatorler sunulmustur. Yaratici dizayna bilgi tabanli bir yaklasim anlatilmistir.

Anahtar Kelimeler : dizaynda yaraticilik, insan yaraticilik stratejileri, kavramsal dizayn, prototipler.

Knowledge-Based Conceptual Design

Abstract

In this paper, creativity in engineering design is analysed from static and dynamic points of view. The former focuses on the relationships among the components of a design solution, and the latter emphasises the sequence of events leading to the solution. Also the conceptual phase which can be considered as the most important part of the design process is introduced. Prototypes, which are a way of representing knowledge, are described and the operations that can be applied to the prototypes are presented. A knowledge based approach to creative design is explained.

Keywords : design creativity, human creative strategies, conceptual design, prototypes

1. GIRIS

En basit tarifiyle, yaraticilik, elimizde olan bilgileri bir araya getirip yeni dizayn veya urun elde etme, zun yillar boyunca bilim adamlarinin dikkatini cekmistir. Cogu zaman matematiksel olarak anlatilmasi imkansiz olan bu beyin fonksiyonuna milattan onceki yillardan baslayarak gunumuze kadar bir aciklama bulunmaya calisilmistir.
Yapay zeka ilminin gelismesi ve ozellikle bilgi tabanli sistemlerdeki ilerlemeler sonucunda gecmiste bulunan cozumlerini deneyebilecegimiz bir ortam olusmustur.
Iste bu makalede, gecmiste ortaya atilmis insan yaraticilik stratejilerin ve yapay zeka teknikleri kullanilarak elde edilmis belli baslilarindan ornekler ve kullanilan tekniklerin aciklamasi yapilmistir. Uygulama alani olarak ise dizaynin ozellikle mekanik ve insaat

muhendisligi dallari secilmistir. Yazarlarin calisma konusu kavramsal dizayn oldugundan dolayi, dizaynin ikinci fazi olan detayli dizayndan bahsedilmemistir.

Makallenin ikinci bolumunde dizaynin tarifi, ucuncu bolumunde ise yaraticilik deyiminin aciklanmistir. Dorduncu bolumde ise bilgi tabanli sistemlerde bilginin temsili ve prototipler anlatilmistir. Son bolumde ise makalede anlatilacak konular bir ornek olarak verilmistir.

2. DIZAYN

Her ne kadar muhendisligin en onemli bolumlerinden biri olsada, uzun yillar boyunca ayri bir bilim dali olarak kabul edilmeyen dizayn, ozellikle bilimde ilerlemeler ve bilgisayar teknolojisindeki gelismeler neticesinde yeni yeni bir dal olarak kabul edilmeye baslanmistir.

Su ana kadar her ne kadar cesitli acilardan bakilarak bir cok tarifler yapildiysa da en genel tarifiyle dizayn, kullanici talepleri ile urun arasindaki islemlerin tumudur (Sekil 1). Muhendislik acisindan dizay isleminin tarifi ise: bireyin, kurulus veya isletmenin veya sosyal ihtiyaclarin karsilanmasi icin belirli bir elemanin, hizmet veya sistemin yaratilmasi veya yeniden duzenlenmesidir [1]. Literaturdeki diger onemli tariflerden biri ise: muhendislik urununun kesfi, kavramin olusturulmasi, prototipleme, hesaplama, veri organizasyonu, degisik bilgi domenlerinin faydali hale getirilmesi, ve detaylari belirlenmesidir [2].

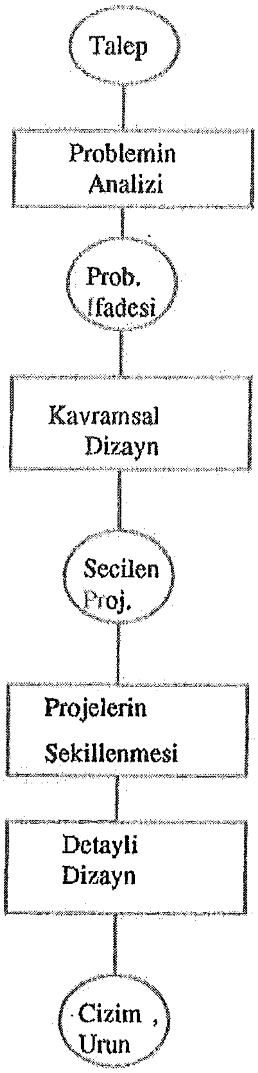
Kullanici
Talepleri
Dizayn
Son
Urun

Sekil 1 : Dizayn

Dizayn islemi mevcut olan urunlerin ihtiyaci karsilamamasindan dolayi, bir ihtiyacin ortaya cikmasiyla baslar ve bir cizim seti ve uretim icin gerekli diger bilgilerin saglanmasiyla sona erer. Dizaynin ilk bolumleri problemin analizi ve problem hakkinda ilk ifadelerin ortaya cikmasi, dizayni mumkun olan fikirlerin ortaya atilmasi ve detaya inilecek fikir uzerinde cesitli kriterlere (maliyet, hafiflik, ekonomik omur, vs.) dayanarak yapilan kontrol ve detayli olarak incelenecek fikirlere bu kriterlere bakilarak karar verilmesi, son olarakta cesitli cizim veya prototip urunlerin olusturulmasi dizayn isleminin tumunu belirler (Sekil 2).

2.1 Kavramsal Dizayn

Bu makalenin esas konusu fikirlerin olusturulmasi ve belirli bir kritere gore uretilen bu fikirlerin gecerliliginin incelenmesi olarak tarif edilebilir, ve ayni zamanda makalenin basliginin bir bolumunu olusturan kavramsal dizayndir. Her ne kadar detayli dizayn bu gune kadar yeterince incelenmis olsa da, fikirlerin uretildigi, ve cogunlukla daha onceki tecrubelere dayali karmasik bir dusunce zinciri olan, ve matematiksel temellere oturtulmasi guc kavramsal dizayn denilen faz ile ilgili su ana kadar sadece iki referans kitap bulunmaktadir [2], [3].



Sekil 2 : Dizayn akis semasi

Dizayn isleminin en onemli sayilabilecek bu fazi temel olarak dort bolumde incelenebilir.

1. Yeni bir urunun kavramsal dizayni (invention)
2. Yeni bir urun veya sistemin farkina varma (innovation)
3. Var olan bir urunu gelistirmek icin kavramsal dizayn (improvement)
4. Var olan bir urun veya sistemin farkina varip yeni dizaynda kullanmak (refinement)

Yukaridaki tariften de anlasilacagi gibi, cogunlukla, kullanicidan gelen talepleri karsilamak icin yapilan ve dizayn olarak adlandirilan islemlerin ozellikle kavramsal dizayn veya ayni zamanda fikir uretme fazi denilen bolumu dizayn isleminin en onemli parcalarindan biridir. Yapay zeka tekniklerinin bu alana uygulanmasiyla [4], [5], [6], [7], [8], [9] dizaynda buyuk olarak nitelendirilebilecek ilerlemeler kaydedilmistir.

3. YARATICILIK

Yaraticilik, bir kisinin yeni veya beklenmedik bir fikir ve dolayisiyla urun ortaya koymasi olarak anlasilir ve zekiligin bir gostergesi olarak kabul edilir. Ama bu beklenmedik zamanlarda beklenmedik ve kaliteli fikirler ortaya koyabilma yetenegi uzun sureler bilim adamlarinin ilgisini cekmesine ragmen buyuk birbolumu hala bir sir olmaktan oteye gisdememistir fakat zekilik ve yogun bir egitimin insan yaraticiligi icin gerekli iki sart oldugu bilim adamlarinin cogu tarafindan kabul edilmistir.

Pisiko-analitik acidan yaraticilik kisinin bir eksikligi giderme ihtiyacindan dolayi (bir yakinin olumu, derin ozlem) baska bir objeyi kayip olan obje yerine transfer etmesiyle gelisen bir yatenektir. Yapay zeka temel alinarak bu aciklama incelendiginde bir simulasyon ve bilgi tabanlarinda birbiri ile acik baglantisi olmayan bilgi domenlerinin kullanildigi gorulur.

Pisiko-fizyolojik yonden yaratici proses bagimsiz ve ani duyusal simulasyon olarak anlasilabilir, boylelikle beyinde yeni aktivasyon sekilleri ve donguleri olusur. Bu tarif ise yapay zeka acisindan bakildiginda sinir aglarinin bilgisayara uygulanmasindan ote bir sey degildir ve bu teknik pattern ogretmeden lineer sistemlerin aciklanmasina kadar buyuk uygulama alani bulmustur [10].

Muhendislik acisindan ele alindiginda ise yaraticiligin elde olan eleman veya bir dizayn cozumunun tumunun, kullanicidan gelen taleplere gore yapilmakta olan dizayna uygulanip ortaya daha onceden bilinmeyen dizayn cozumleri atmaktir. Kavramsal dizaynda yaraticilik ise elde olan bilgilerin kullanilip yeni dizayn fikirleri elde etmektir. Temelde dusunulurse yaraticilik denilen kavram bir dusunce sistemidir ve gerceklestirilebilirse sadece dizaynda degil; tabiatta su anda var olan fakat farkinda olmadigimiz bilgi kombinasyonlarinin ortaya cikartilmasinda kullanilabilir.

Yukaridaki degisik yonlerden yapilan tariflerdende anlasilacagi gibi yaraticilik yapay zeka biliminin gerceklestirebilecegi bir dusunce sistemi veya yetenegidir.

3.1. Insan Yaraticilik Stratejileri ve Modellenmesi

3.1.1 Deneme Yanilma (Trial an Error)

Muhtemel cozumler uretip daha sonra bunlari test etmekten ibaret olan bu metod bir cok yaratici ve rutin dizayn problemlerine uygulanmistir. Uretip test etme islemi, uretilen cozumlerden birinin veya birkacinin testi gecmesiyle son bulur. Bu teknigin bilgisayara uygulanmis hali Uret ve Testetdir (Generate and Test). Bunda ise bir dongu ile uretilen sonuclar kontrol modulunde degerlendirildikten sonra kullanicinin istegine gore yeteri kadar gecerli cozum bulundugunda islemi durdurmaktir.

3.1.2 Engellenmemis Fikir Uretimi (Brainstorming)

Bu teknik belirli bir zor problemin çözümünde degisik mesleklerden kendi konulariyla ilgili iyi bir bilgi temeli olan kisilerin bir araya gelip, yöneltilen problemle ilgili herhangi bir kesintiye veya elestiriye ugratilmaksizin bütün düsüncelerini söylemesidir. Bu esnada söyledikleri not alinip degerlendirilmesi sonra yapilmaktadir. Butun istirakciler problem hakkinda dusunduklerini soylemesiyle seans sona ermektedir. Bazi yonleriyle bu metodun yapay zekaya uygulanmasi SOAR denilen bir sistemle yapilmistir [11].

3.1.3 Mutasyon (Mutation)

Mutasyon, bir obje veya fikrin bilinen herhangi bir kurala uymaksizin, ozelliklerinin degistirilmesi islemidir. Mutasyonun bir cok uygulanma teknigi vardir, bunlar arasinda Genetic Algoritma [12], islemin toplama, cikarma, carpma ve bolme olarak gerceklestirilmesi [13] vs. Mutasyonun bilgisayara uygulanmis hali oldukca basarili olmustur, bu teknigin en onemli uygulamalarindan biri Lenat (1982) tarafindan yapilan AM dir [14].

3.1.4 Benzerlik (Analogy)

Yeni karsilasilan problem ve eski cozulmus problemler arasinda kurulan benzerlik neticesinde, eski cozum teknik veya bilgilerin yeni karsilasilan probleme benzerliginden faydalanarak, elde olan problemin cozulmesi islemine Benzerlik veya Analogy denir. Benzerlik cok iyi bir problem cozme teknigi olmasinin yani sira, ayni zamanda iyi bir de ogrenme teknigidir [15]. Bu teknigin etkin bir sekilde kullanilmasi icin, gecmis tecrubelerle yeni problemi baglayabilecek guclu bir metoda ve genis bir bilgi tabanina ihtiyac vardir. Yapay zekaya en basarili uygulamalarindan biri Prieditis tarafindan yapilmistir [16].

3.1.5 Ters Cevirme (Inversion)

Adindan da anlasilacagi gibi bu teknikte hafizadaki bilgilerle yeni karsilasilan dizayn problemine cozum bulunamazsa, hafizadaki bu bilgileri ters cevirerek yeni cozum yontemleri aramaya calismaktir. Bunun en guzel orneklerinden biri manyetik alana dik hareket ettirilen iletkende bir akim uretilir ve bunun tersi olarakta, iletkenden akim gecirildiginde iletken etrafinda manyetik alan olusur. Bilgi tabanli yaratici dizaynda bu teknigi kullanarak yapilan en kayda deger calisma PROMPT dur[17].

4. PROTOTIPLER [18]

Yaratici turde bir bilgi organizasyona ulasabilmak icin, bilgi tabaninda sahip oldugumuz bilgilerin sistemlere nasil aktarilacagi, diger bir deyisle bilginin nasil temsil edildigi cok onemlidir. Bilgi tabanli sistemlerin ortaya atilisindan bu gune kadar basta kural tabanli temsil, cerceve tabanli temsil ve semantik aglarla baslayarak bir cok bilgi temsil yollari denenmis ve bunlarin bazi noktalardaki yetersizlikleri metodlarin hibridlenmesiyle giderilmeye calisilmistir.

Su ana kadar genel amacli bir cok bilgi temsil yontemleri ortaya atilmis olsa da; dizayna yonelik olarak hazirlanmis en kayda deger bilgi temsil yontemi Prototiplerdir. Bir dizayn sistemindeki genellestirilmis eleman guruplari sinifina Prototip denilir [tech rep]. Bir sinif olarak prototipler, diger ust siniflardan cesitli deger ve metodlari miras edinirler boylelikle

elde edindikleri bilgiler neticesinde dizayna baslamak icin gerekli olan veri miktari azaltilmis olur.
Bir Prototipe P dersek

$$P = (D, I, V, K)$$

ifadesi elde edilir. Bu ifadede

D : dizayn tanimi
I : dizaynin yorumu
V : dizayn elemanlari
K : dizayn tanimini dizayn yorumuna baglayan sistem icindeki bilgi

Bir prototipte, Prototipi hazirlanan objeye ait Fonksiyon, Davranis va Yapi olmak uzere uc ana bilgi unsuru vardir. Fonksiyon bir objenin fonksiyonalitesinden yani o objenin ne yapabileceginden bahsederken; yapi o objeyi olusturan elemanlardan bahseder, davranis ile o yapiyla o fonksiyonun nasil gerceklestirilecegini anlatir (Sekil 3).

4.1 Dizaynin Prototiplere Dayanarak Siniflandirilmasi [19]

4.1.1 Prototiplerin Gelistirilmesi

Sinifindan ornek olarak alinan prototip kullanici taleplerini karsiliyorsa, bulunan bu prototipin sadace degerlerini degistirerek dizayn gerceklestirilir. Bu haliyle, yapilan dizayn turune rutin dizayn denir.

$$D = \tau(D, I)$$

4.1.2 Prototip Adaptasyonu

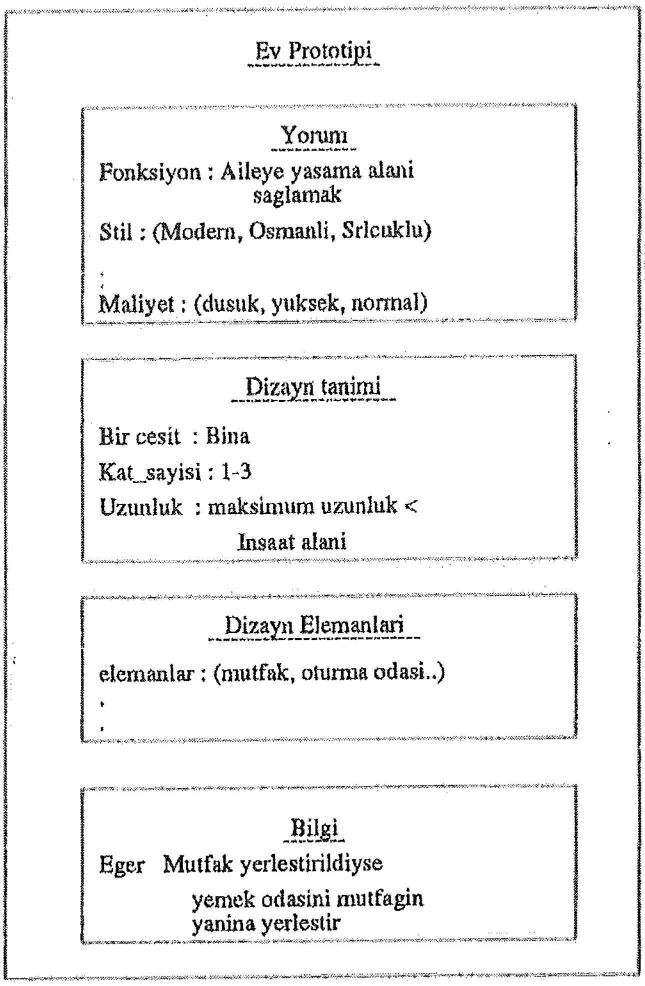
Bulunan prototip veya prototipler herhangi bir sekilde cozum aranan dizayn problemine uygun degilse, bu takdirde, kullanici prototipleri istedigi sekilde adapte ederek problemin cozumune ulasmasi mumkundur. Yorum (I), elemanlar (V), sistem bilgisinden (K) herhangi birini degistirerek dizayn tanimi icin yeni potansiyel degerler elde edilebilir.

$$D' = \tau(K', V', I')$$

Prototip adaptasyonunun sonu bir anlamda yaratici dizayndir. Uygun adaptasyon saglandiktan sonra yapilan islemler prototiplerdeki ilk islemlerin aynisidir.

4.1.3. Prototip Uretimi

Prototip uretimi, prototiplerle yapilabilecek en yuksek islem seviyesidir. Bu islemde ilk iki seviyede problemin cozumune cevap veremeyen sistem, yeni ve hafizadakilerden farkli bir prototip uretir. Bu seviyenin formal olarak karsiligi bilinmemektedir.



Ev Prototipi

Yorum

Fonksiyon : Aileye yasama alani saglamak

Stil : (Modern, Osmanli, Srlcuklu)

Maliyet : (dusuk, yuksek, normal)

Dizayn tanimi

Bir cesit : Bina

Kat_sayisi : 1-3

Uzunluk : maksimum uzunluk < Insaat alani

Dizayn Elemanlari

elemanlar : (mutfak, oturma odasi..)

Bilgi

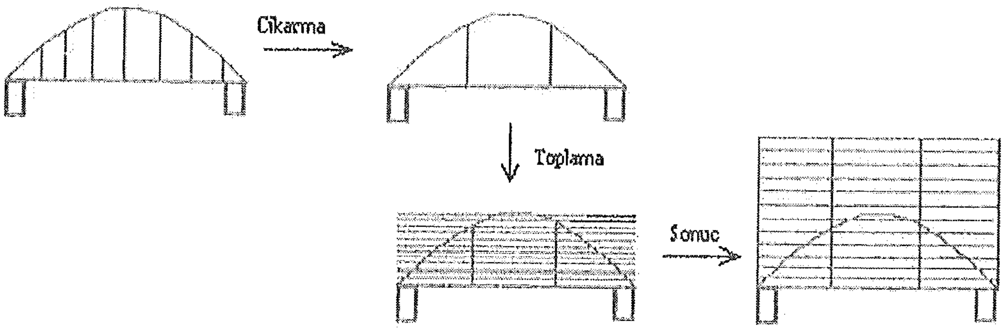
Eger Mutfak yerlestirildiyse yemek odasini mutfagin yanina yerlestir

Sekil 3 : Ev Prototipi

5 BILGI TABANLI YARATICI DIZAYNA BIR YAKLASIM

Bir otoyol uzerine bir bina yapilma ihtiyaci duyulmustur. Bu da uzun span (kopru iki ayagi arasindaki mesafe) gerektirmektedir. Kullanicidan gelen talepler ise ekonomik sebeplerden dolayi 30 ft. uzerinde olmamali ve binanin tabani yuk tasimaya elverisli olmalidir. Prototiplerle bilgi temsili yapilan bu sistemde ilk adim, verilen taleplere uygun prototiplerin bulunmasidir. Bu sebeple hafiza incelendiginde analoji yardimiyla uzun span mesafelerini karsilayabilecek sadece bir cozum bulunabilmistir. Bu cozum ise daha once dizayni yapilan bir kemer kopruye aittir. Uygun prototip secildikten sonra, korunun bina icin uygun olmayan ozellikleri mutasyon ile degistirilmistir. Mutasyon islemi, cikartma olarak kablo sayisinin azaltilmasi ile baslar bu seviyeden sonra kemer altinda kalan suspansiyon kablo kullanici tarafindan, gerilme bloklari ile degistirilir [13].

Bu problemdeki analoji bir benzerlik kriterine gore kismi karsilastirma ile yapilmaktadir. Kismi karsilastirma sonucunda bir cok potansiyel cozum bulunmasi mumkundur. Kismi analojinin en onemli ozelligi farkli ve birbiriyle direk iliskisi bulunmayan bilgi tabanlarinin kullanilmasidir. Verilen ornekte, mutasyonun daha onceki bolumlerde bahsdilen toplama, cikarma, vs. sekli kullanilmistir. (Sekil 4.) de de gorulecegi gibi ilk islem suspansiyon kablo sayisindaki mutasyondaki cikartma ile azaltma yapilmasidir.



Sekil 4 : Mutasyonun toplama ve cikarma olarak uygulanmasi

Daha sonra ise toplama islemi ile bina icin gerekli olan yatay yuzeyler yerlestirilmistir. Yukaridaki ornektende anlasilacagi gibi problemin en onemli yonleri: kurallara bagli kalmadan yapilan degisiklikler, ve daha onceden de sozedildigi gibi birbiriyle dogrudan iliskisi olmayan bilgi tabanlarinin kullanilmasidir.

6. SONUC

Makalenin cesitli bolumlerinde de belirtildigi gibi, yaraticilik, insanin sahip oldugu ustun bir bilgi organizasyon turudur. Bu organizasyon tekniginin dizaynin ozellikle kavramsal fazina uygulandigi takdirde insan hayatini kolaylastirabilecek ve bu turde bir dusunce yapisina sahip olmayan insanlara yardimci olabilecek asistan bir sistem uretebiliriz [20], Her ne kadar bu alanda daha uzun bir zaman arastirmalar yapilmasi gerekse de, bu turde sistemlerin gelistirilmesi bilimde bir donem olacaktir.

7. TESEKKUR

Yazarlar bu calismaya verdigi destekten dolayi HES KABLO A. S. ne, Overseas Research Students Awards Scheme ve British Council e tesekkur eder.

8. KAYNAKLAR

1. Rabins, M.J., Ardayfio, D., Balzar, R., Fenves, S., Nadler, G., Richardson, H., Rinard, I., Roth, B., Seireg, A., and Wozny, M.,*Goals and Priorities for Research in Engineering Design*, A Report to the Design Research Community, The American Society of Mechanical Engineers, New York, NY, July 1986.

2. French, M.J., *Conceptual Design for Engineers*, The Design Council, London, Springer-Verlag, 1986.

3. Fraser, D.J., *Conceptual Design and Preliminary Analysis of Structures*, Pitman, Marshfield, MA, 1981.

4. Pham, D.T., and Onder H.H., An Expert System for Ergonomic Workplace Design Using a Genetic Algorithm, in: G. Rzevski, and R.A. Adey (Eds.), *Applications of Artificial Intelligence in Engineering VI,* Computational Mechanics Publications, Southampton and Elsevier, Amsterdam, 1991, 287-300 .

5. Maher, M.L., and Zhao F., Dynamic Associations for Creative Engineering Design, *Int. Round Table Conf. on Modelling Creativity and Knowledge-Based Creative Design,* Heron Island, Queensland (11-14 Dec. 1989), 155-172.

6. Fischer, G., Creativity Enhancing Design Environments, *Int. Round Table Conf. on Modelling Creativity and Knowledge-Based Creative Design,* Heron Island, Queensland, (11-14 Dec. 1989), 13-34.

7. Pham, D.T., and Tacgin, E., Techniques for Intelligent Computer Aided Design, in: D.T. Pham (Ed.), *Artificial Intelligence in Design,* p. 5-25, Springer-Verlag, Berlin, 1991.

8. Akagi, S., Expert System for Engineering Design Based on Object Oriented Knowledge Representation Concept, in: D.T. Pham (Ed.), *Artificial Intelligence in Design,* p. 61-100, Springer-Verlag, Berlin, 1991.

9. Pham, D.T., and Lazaro, A. de S., Knowledge-Based Design of Jigs and Fixtures, in: D.T. Pham (Ed.), *Artificial Intelligence in Design,* p. 371-386, Springer-Verlag, Berlin, 1991.

10. Sagiroglu, S., Identification of Linear Systems with Neural Networks, *Technical Report,* Erciyes Universitesi, 1991.

11. Laird J.E., Newell, A., and Rosenbloom, P.S., SOAR:an Architecture for General Intelligence, *Artificial intelligence* 33(1) 1987, 1-64.

12. Pham, D.T., and Karaboga, D., Optimum Design of Fuzzy Logic Controllers Using Genetic Algorithms,*Journal of Systems Engineering* 1(2) 1991, 114-118.

13. Maher M.L., Zhao, F., and Gero, J.S., An Approach to Knowledge-Based Creative Design, *Preprints, NSF Engineering Design Research Conf., College Of Engineering, Univ. Of Massachusetts, Amherst* June 1989, 333-346.

14. Lenat, D.B., An Artificial Intelligence Approach to Discovery in Mathematics as a Heuristic Search, in: R. Davis and D.B. Lenat (Eds.),*Knowledge-Based Systems in Artificial Intelligence*, McGraw-Hill, New York 1982.

15. Kocabas, S., A Review of Learning, *The Knowledge Engineering Review*, 6(3), 1991, p.195-222.

16. Prieditis, A. (Ed.), *Analogica,* Pitman/Morgan Kaufmann, Los Altos, California, 1988.

17. Murthy, S.S., and S. Adanaki, PROMPT: an Innovative Design Tool, in: J.S. Gero (Ed.),*Expert Systems in Computer-Aided Design,* North-Holland, Amsterdam, 1987, p.323-341.

18. Gero, J.S., Prototypes: a New Schema for Knowledge Based Design, *Working Paper,* Architectural Computing Unit, Department of Architectural Science, University of Sydney, Sydney 1987.

19. Coyne, R.D., Rosenman, M.A., Radford, A.D, Balachandran, M., and Gero, J.S., *Knowledge-Based Design Systems*, Addison-Wesley, Reading, MA, 1989.

20. Guner, U. F., Design Creativity and AI, *Technical Report,* Hes Kablo A.S. Hacilar Yolu 8. km. Kayseri, Turkey, 1992

A Distributed Expert System Architecture

Faruk Polat* and H. Altay Guvenir*

*Department of Computer Eng. & Info. Sci. Bilkent University, Bilkent 06533 Ankara, Turkey

Abstract

As a subfield of Distributed Problem Solving, cooperating experts approach allows several specialized problem solvers to work together for solving complex tasks such as design, medical diagnosis, business management, and so on. Due to differences in goals, knowledge and viewpoints of agents, conflicts might arise at any phase of problem solving. Managing diverse expertise requires good models of conflict resolution. In this paper, a model for cooperating experts which openly supports multi-agent conflict detection and resolution is proposed. The model is based on the insights that each agent has its own conflict knowledge which is separated from its domain level knowledge. The proposed model is compared with the existing approaches.

1. INTRODUCTION

Distributed Problem Solving (DPS) is a subfield of AI which is concerned with solving problems by using both AI techniques and distributed processing capabilities. DPS is different from distributed processing in that it does not only distribute data, as in the case of distributed processing, but also control. In addition, DPS involves extensive cooperation among problem solvers [2,8]. One of the application domains of DPS is cooperating expert systems. Cooperating expert system approach is concerned with solving complex tasks that require diverse expertise to generate comprehensive solutions.

Applications of cooperating expert systems can be seen in human problem solving tasks such as design, medical diagnosis, research, business management, and human relations. There have been several systems reflecting cooperating expert systems approach such as Hearsay-II, Contract Net, Distributed Vehicle Monitoring Testbed, MDX, Coop, etc [7]. Managing diverse expertise is difficult because one has to take into account the problems which will arise in working out solutions in the face of conflicting goals, constraints, viewpoints, and knowledge of heterogeneous experts.

In this paper, we propose a model in which a set of knowledge-based agents cooperate for solving design problems. The model is based on resolution of conflicting solutions generated by experts having different goals, priorities, and evaluation criteria. Existing approaches [1,4,5] to conflict management rely on coordinated resolution strategies which require resolution of a conflict based on a globally agreed strategy. In these systems, conflict resolution knowledge is maintained either centrally, or replicated at all agents. In any case, one of the disputants is given the power to take control of the conflict and use a resolution scheme known to everybody. In our proposed model, however, agents are free to choose the most appropriate action, given their understanding of the global and local situations and their own capabilities. They maintain their own set of conflict resolution knowledge which is not globally known. Using their own conflict knowledge, the participants may come to an agreement on a revised solution. This is similar to the resolution of conflicts that occur among human beings when solving a problem.

In the next section, an overview of conflict management in cooperating expert systems is presented and the existing approaches are summarized. Section 3 explains the proposed model for cooperating experts and how the problem solving proceeds within the model. Section 4 describes how conflict resolution takes place in the proposed model. Section 5 includes a design example to illustrate how problem solving proceeds in the proposed model. Section 6, the last section, summarizes the proposed model emphasizing its characteristics.

2. CONFLICT MANAGEMENT AMONG COOPERATING EXPERTS

In cooperating experts approach, several specialized agents combine to solve a common problem. During any phase of problem solving, conflicts might appear as a result of incorrect and incomplete local knowledge, different goals, priorities, and solution evaluation criteria. When there are several conflicting proposed solutions for a (sub)problem, the agents involved in a conflict must either agree to choose one proposal, cooperatively revise one, or search for a new solution that will be acceptable to everyone.

A common practice in building knowledge-based systems is to avoid potential conflict situations through analysis and consistency checking of knowledge base at development time [6,7]. This approach, although effective, is very costly as the amount and diversity of knowledge increases. Resolving all conflicts, no matter how unlikely, at development time can be prohibitively time-consuming. Moreover, dividing the domain knowledge into smaller internally consistent collections is difficult.

The problems encountered when resolving conflicts in development time can be avoided by allowing conflicts to occur and be resolved at run-time. In other words, participating agents are allowed to generate conflicting solutions to the subproblems at run-time. In case of conflict, a set of strategies could be used to resolve the conflict. Some examples of strategies include *backtracking*, *compromise negotiation* (a solution is iteratively revised by sliding a value, or set of values along some dimension until a middle point is found that is mutually acceptable), *integrative negotiation* (identify the most important goals of each agent, and find a solution which fulfills all of these goals), *constraint relaxation*, *case-based* and *utility reasoning* methods, etc. [1,4,5]. Work in this class come closest to providing conflict resolution expertise with first-class status..

There are several important studies that emphasize the use of conflict resolution within cooperating expert systems [1,4,5]. In these existing systems, several agents solve subproblems relevant to their specific expertise and integrate their efforts using conflict resolution strategies that are appropriate to the problem solving context. All of the agents have a global knowledge of conflict resolution strategies. When a conflict is detected, agents involved in the conflict propose their alternative resolution strategies. Eventually they agree on a resolution scheme. The conflict is resolved by *third-party arbitrator* agent based on the globally known conflict resolution knowledge. The arbitrator operates in both passive, and active mode. In *passive* mode, arbitrator monitors the agent proposal process and intercedes when a problem is evident. In *active* mode, arbitrator mediates during the agent's proposal process when called upon by the agents.

3. A NEW MODEL FOR COOPERATING EXPERTS

The proposed cooperating experts environment is organized as a community of cooperating problem solving agents, where each agent is represented as a fully functional and autonomous knowledge-based system. The model is specifically designed for solving problems in the domain of design. This model is based on the insights that each design agent has its own conflict resolution expertise separate from its domain-level design expertise, and that this expertise can be instantiated in the context of particular conflicts into specific advice for resolving these conflicts. The model allows a new problem solver to be added, or an existing one to be removed without requiring any modification on the rest of the system. The model, therefore, can be considered to achieve Open Systems Semantics[1] [3] in the sense that it does not only allow *scalability* (ability to increase scale of commitments) but also *robustness* (ability to keep commitments in face of conflicts) which are two primary indicators in Open Systems Semantics.

3.1. Architecture of the Proposed Model

The proposed cooperative design environment (Fig.1) is composed of a set of design agents which are fully functional knowledge-based systems.

[1] Open Systems deal with large quantities of diverse information and exploit massive parallelism

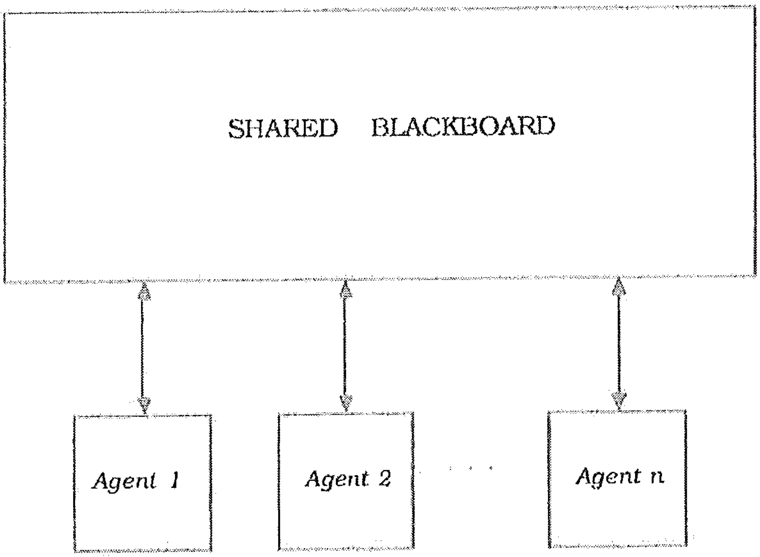


Figure 1: The Architecture of the Proposed Cooperative Design Environment

The agents communicate by posting assertions in a shared language. This requires translation capabilities to be included within the agents. The shared blackboard is a public repository available to all agents. This gives one the ability to store "global" information although the information can only be used locally by the agents. Alternatively, it would be possible to convey information directly through point-to-point communication channels or reserved-spot communication. The shared blackboard is partitioned into four chunks, allowing fast access, delete and update operations of units . They are called *problem, solution, proposal* and *conflict areas*.

The problem area of the shared blackboard contains the initial problem definition and overall requirements that must be taken into account by the design agents. The solution area of the shared blackboard includes the evolving design template to which non-conflicting design commitments produced by agents are added. Proposal area includes partial and incomplete solutions at several layers of abstraction issued by design agents. Design agents assert their solutions as proposals into this area. Owner of the proposal indicates its confidence in generating such a proposal. This information would be useful for other agents if the owner utilized inaccurate, or incomplete knowledge in producing its solution. Conflict area is the place where agents put into their critiques related to a new design commitment. A portion of this area provides communication medium with agents that are involved in a conflict situation. This area holds evaluation results and conflict resolution recommendations issued by design agents.

Description of an agent in the proposed model is given in Fig.2. An agent supports a *knowledge base*, a *database*, and a *general controller*. Knowledge base includes domain and control knowledge just like in a classical knowledge-based system. In addition, it also contains conflict resolution knowledge to be used in cooperatively managing conflicts with other agents. This knowledge is not known globally and varies with respect to the agents' beliefs and understanding of the environment. The database includes facts, goals, and constraints specific to the domain of that agent. Agents also maintain two types of history information related to the solution generation phase and conflict resolution phase. This does not only make backtracking possible, but also allows case-based information to be used later for solving similar problems encountered. General controller includes procedures for generating and evaluating design commitments, managing conflicts, and translating messages into the common language.

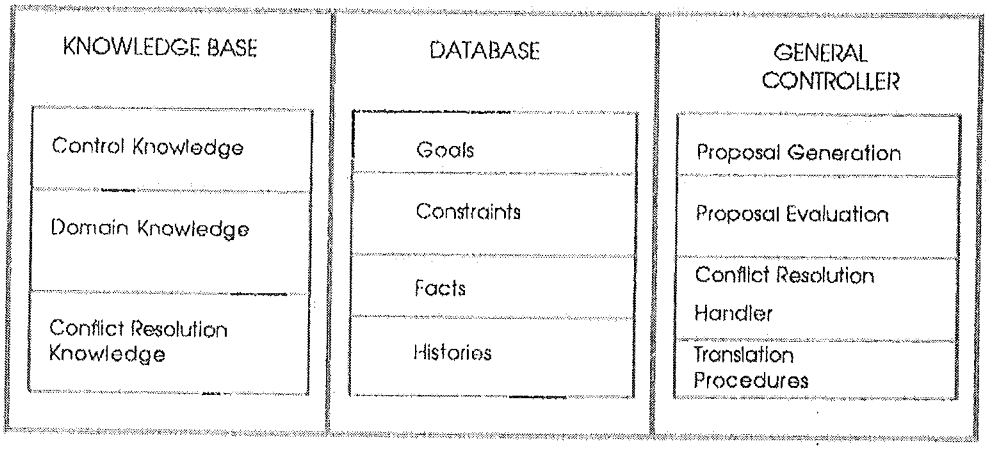


Figure 2: Internal Structure of a Design Agent in the Model

Local knowledge is represented in whatever language desired and cannot be accessed by any agent except its owner. Several knowledge representation techniques have been developed for the domain of design. If the internal language is not the same as the shared language, translation procedures are incorporated within agents. In case of a conflict, agents might make some or all of their goals, constraints, and even knowledge available to others.

4. CONFLICT RESOLUTION IN THE MODEL

The problem solving in the proposed model is initiated by one of the agents asserting a problem definition into the problem area of the shared blackboard. All interested agents, after examining the problem definition, are instantiated and they start producing design proposals related to their expertise, knowledge, and viewpoints. When a design agent generates a design proposal, it is put into the proposal area. The agent producing this proposal also includes it explanation information which indicates which of agents' goals and constraints caused such a proposal. This explanation allows other agents to understand why such a proposal has been asserted.

After the generation of a proposal, all of the agents are signalled to criticize the proposal. When an agent detects a conflict, it participates in the resolution process based on its own conflict resolution knowledge. Each agent may utilize different conflict resolution strategies. For example, suppose that we are given the problem of designing an office. The functionality agent suggests to put the PC desk close to the window so that a PC user could have a look outside when (s)he is bored. On the other hand, the computer specialist, detecting a conflict, argues that sunshine could damage the PC. The computer specialist uses a conflict resolution strategy which says that "put electrical devices far away from windows." The functionality agent, however, uses a domain-independent resolution scheme "try other subgoal alternatives." Eventually two experts revise the proposal such that the PC desk is put into a place in the office which is not exposed to sunshine, by using different resolution schemes. In deciding which strategy to apply, an agent uses information gathered until the time the conflict has occurred as well as its conflict knowledge. This information includes

- Explanation embodied within the proposal.
- Critiques made by the interested parties to the proposal.
- The relevance of the agent to particular problem being solved.
- Flexibility of agents involved in conflicts.
- Behavior and actions of other agents in resolving the conflict.

- Conflict resolution history information.
- Number of agents involved in the conflict.
- Available problem-solving resources.

When none of the interested agents detect any conflict related to a proposal, the partial design template residing in the solution area is updated by using the design contribution existing in the proposal. The process continues until design template meets requirements specified by the agent that put the initial problem definition in the problem area of the shared blackboard. The design process may also be terminated, although the agent that put the problem definition is not satisfied. This may happen in cases where none of the agents can generate a nonconflicting design proposal anymore.

5. AN EXAMPLE: OFFICE DESIGN

The following example is taken from the domain of office design and exemplifies the problem solving process of the cooperating experts that is used in our implementation. Here, we present a simplified layout problem for an office design and describe design agents and their interactions. A well-designed office encompasses different areas of expertise concerning aesthetics, functionality, energy efficiency, etc. In this example, we have incorporated four agents in the proposed framework. They are

- the client agent,
- the functionality agent,
- the electricity agent, and
- the cost agent.

The client agent is the one that puts the problem definition specifying general constraints and the global design goal to be satisfied. The functionality agent uses specific heuristic search techniques in the area of space planning. Electricity agent is concerned with all the electrical and electronically devices and wiring including computers, telephones, facsimile systems, etc. The cost agent is required to control the overall cost of the design and avoid wasteful use of resources. The design process is initiated by the client agent that puts the following problem definition into the problem area of the shared blackboard:

```
Problem Definition =
<Client-Agent,
     {goal: (design office)
      subgoals: ((minimize amount-of-walking)
                 (customize components-to-the-size-of-office)
                 (maximize efficiency)
                 (must-have PC) ...)) },
     {constraints: ((to-be-used-by faculty-member)
                    (number-of-occupants 1)
                    (cost-of-design < $6000) ... ) },
     {layout:
          shape: rectangular
               dimensions : ((length 10) (width 8) (height 2.5))
               coordinates: (upperleft (x 0 y 0))
          window:
               type : ((frame wood) (glass glass1))
               dimensions: ((height 1))
               coordinates: ((x 0 y 3) (x 0 y 5))
          door:
               type: ((made-up-of wood))
               dimensions: ((height 2))
               coordinate: ((x 8 y 7) (x 8 y 8))
          electrical-plug:
               coordinates: ((x 6 y 0))
          phone-plug:
               coordinates: ((x 6.5 y 0))..} >
```

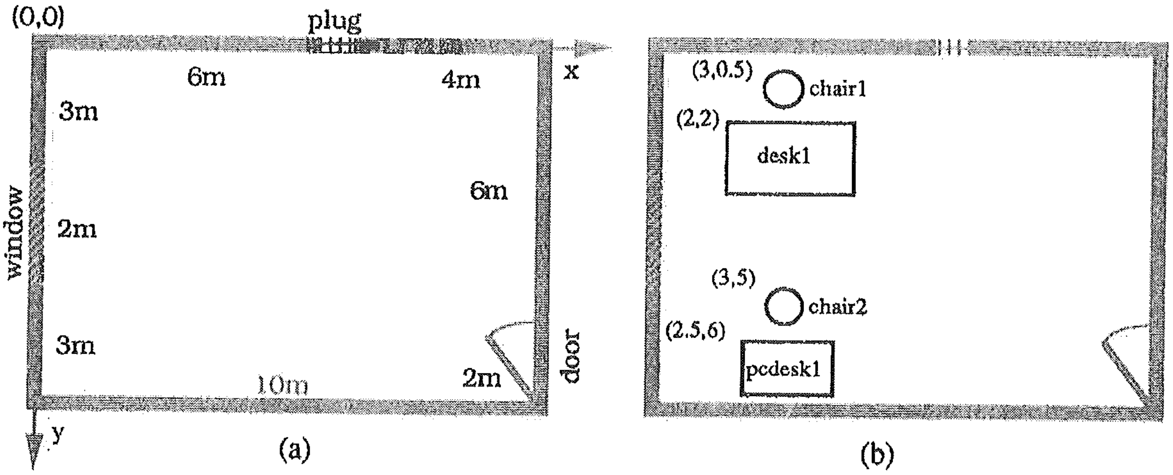


Figure 3: a) Global Layout of the Office. b) Layout of the Office After `Proposal-0`.

Fig.3a shows the global layout of an office. In this example, we ignore the third (*z*) dimension; instead the height attribute of objects is used when necessary. Also we are not concerned with the precise locations of objects. After examining the problem definition, all of the interested parties start producing design commitments. First, the functionality agent, according to its expertise and understanding of the problem asserts the following proposal into the proposal area of the shared blackboard which updates the template as shown in the Fig.3b.

```
Proposal-0 =
< Functionality-Agent,
    {(put :object desk1   :type desk    :location (2 2)),
     (put :object chair1  :type chair   :location (3 0.5)),
     (put :object pcdesk1 :type pcdesk  :location (2.5 6)),
     (put :object chair2  :type chair   :location (3 5)) },
    {(utilize sunshine)
     (have better-view) },
    nil >
```

The functionality agent has decided to put a desk and a PC desk along with two chairs nearer to the window so that the occupant could not only have a good view but also utilize sunshine. This proposal triggers the evaluation procedures within other interested agents. Client agent detects a conflict after evaluating the proposal. With this configuration, the client agent notices that the occupant must walk too much because (s)he might need to use PC (that will be put on the PC desk) very often. The functionality and client agents combine to resolve the conflict encountered. Client agent uses a specific resolution scheme which states that "keep frequently used objects close to each other," and functionality agent uses a general conflict resolution strategy which is "try other location alternatives." Client agent has two alternatives to resolve the conflict from its perspective. It may put the PC desk either to the left, or to the right of the other desk. Taking into account the explanation within the functionality agents' proposal, the client agent proposes to put the PC desk to the left of the other desk so that the PC desk will be close to the window and hence the occupant can utilize sunshine and have a better view. The revised and agreed solution is shown in Fig.4a.

At that time the cost agent realizes that there is no need for having two chairs. The cost, client, and functionality agents decide to remove one of the chair as shown in Fig.4b. In the resolution phase, agents agree on rotating the PC desk such that a chair could be used for both desks. Next is the conflict resolution tuple generated by the client agent:

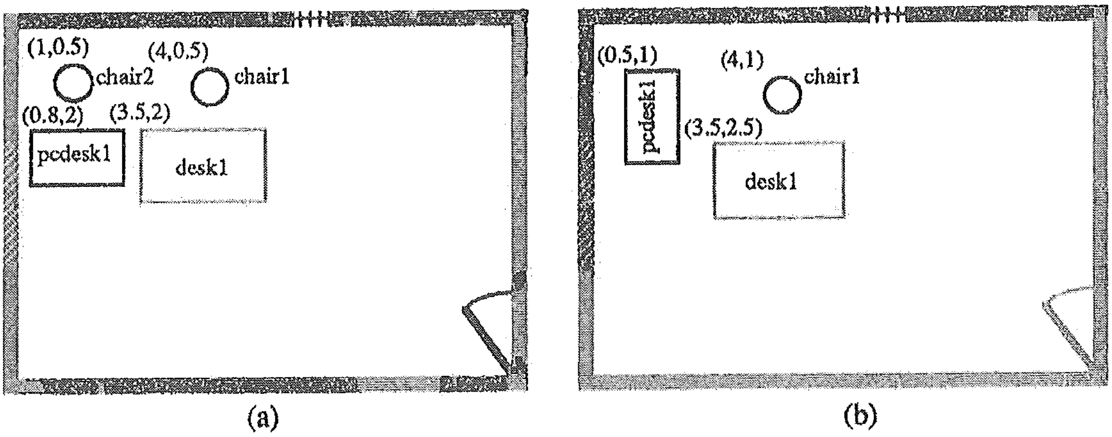


Figure 4: a) Layout of the Office After Resolving the Conflict in `Proposal-0`. b) Layout of the Office After Resolving the Conflict in `Proposal-0'`.

```
Conflict Detection Tuple =
    < Client-Agent, Proposal-0,
      { (all-actions) }, { (increased amount-of-walking) },
      "Conflicting-Proposal" >
```

The design proceeds in this manner until reaching the requirements specified by the client agent. In this example, we only gave a segment of problem solving process emphasizing the resolution of conflicts without considering precise locations of objects.

6. CONCLUSIONS

Cooperating experts approach has an important role in the field of Distributed Artificial Intelligence because many of the problems that are being encountered in real life require the application of complex and diverse expertise. One of the important problems faced in a cooperating community of experts is how to detect and resolve conflicts occurring at any phase of problem solving. Existing approaches to conflict resolution rely on coordinated conflict resolution strategies [1,4,5]. In these approaches, each agent is assumed to have a global knowledge of conflict resolution information. In case of conflicts, they agree on a conflict resolution scheme and one agent resolves the conflict using a globally agreed resolution strategy.

In this paper, we propose a cooperating experts environment for solving problems that openly supports multi-agent conflict detection and resolution. In this environment, each agent is free to choose the most appropriate action given its understanding of the global and local situation and its own capabilities. Each agent has its own conflict resolution knowledge which is not accessible and known by others. Furthermore, there are no globally known conflict resolution strategies. Each agent involved in a conflict chooses a resolution scheme according to its self-interest. Agents might use different strategies of their own and might still agree on a solution.

Currently, we are implementing the model on interconnected SUN 3 - 4 workstations. All of the problem solvers, agents, are modeled as processes running on different workstations that communicate over Ethernet. We will test the conflict resolution based model on various other examples in the domain of design.

7. REFERENCES

1. Adler, M. R., et al., Conflict resolution strategies for non-hierarchical distributed agents, in: M.N. Huhns (Ed.), *Distributed Artificial Intelligence* (Vol. 2, 1989), 139-161.
2. Gasser, L., Social conceptions of knowledge and action: DAI foundations and Open Systems Semantics, *Artificial Intelligence*, (Vol. 47, 1991), 107-138.
3. Hewitt, C., Open Information Systems Semantics for Distributed Artificial Intelligence, *Artificial Intelligence*, (Vol. 47, 1991), 79-106.
4. Klein, M., and Lu, S. C-Y., Conflict Resolution in Cooperative Design, *Artificial Intelligence in Engineering*, (Vol. 4, No. 4, 1989), 168-180.
5. Lander, S., and Lesser, V. R., Conflict Resolution Strategies for Cooperating Expert Agents, in: *Proceedings International Conference on Cooperating Knowledge-Based Systems*, (UK, October 1990).
6. Nguyen, T. A., et al., Verifying Consistency of Production Systems, in: *Proceedings of the Third Conference on Artificial Intelligence Applications*, (1987), 4-8.
7. Polat, F., and Guvenir, H. A., Coordination Issues in Distributed Problem Solving, in: M. Baray, B. Özgüç, (Eds.)*Proceedings of the Sixth International Symposium on Computer and Information Sciences*, (Elsevier, Vol. 1, Antalya, 1991), 585-594.
8. Polat, F., and Guvenir, H. A., UVT: A Unification Based Tool for Knowledge Base Verification, in: *Proceedings of the European Workshop on the Verification and Validation of Knowledge Based Systems*, (Cambridge, England, 1991), 147-163.

KNOWALL! An Experimental Inscriptor-based Expert System Shell

David Davenport, Mehmet Sen, Ahmet Nezir Erturk

Computer Engineering & Information Sciences Dept.
Bilkent University
Ankara 06533 - TURKEY

Email: DAVID@TRBILUN

Abstract: In this paper we present an experimental expert system shell, KNOWALL!, and demonstrate how it can provide a more "intelligent" solution to classification type problems than do conventional rule-based systems. The shell is based on a new form of representation, known as "inscriptors", specifically designed to store and reason with knowledge in the real-world. Inscriptor-based systems offer a methodology for knowledge acquisition and validation, ease of use and maintenance, and potentially more robust solutions!

Introduction

Expert systems are computer programs that achieve expert (problem solving) performance in some limited domain and can explain their reasoning. The expert system paradigm developed in the 1970's in response to the failure of attempts to construct general problem solvers. Efficient problem solving could not be achieved through generally applicable heuristics alone, but was seen to demand domain specific knowledge. The new approach was not able to solve 'any' problem, but rather efficiently solve a few specific problems from a narrow domain. Since to be worthwhile such problems also had to be difficult, the term "expert systems" became used to describe such programs.

Early expert systems, such as MYCIN, XCON etc.[1], were large, expensive undertakings, but they paved the way to cheaper systems by establishing the idea of separating the domain knowledge from the mechanisms which manipulate it. If a single form of representation could be used to store knowledge from any domain, then the same "inference" mechanism could be used to construct any system. This observation gave rise to the "expert system shell", a program providing the basic inference mechanisms, to which the user simply adds knowledge (in an appropriate form for the program to read and manipulate) to produce a functional expert system.

The advent of such shells greatly reduced the time and expense of constructing expert systems, making them viable propositions for a wide range of applications. The lure of "intelligent" problem solving combined with the apparent ease with which "something" could be got up-and-running, ensured that expert systems technology spread rapidly. Unfortunately, even though more and more systems were built, relatively few became commercial successes; in most cases expert systems simply did not live up to expectations. Criticisms leveled at them included,

- the difficulty of acquiring knowledge;
- the necessity for much commonsense knowledge;
- the restricted form of dialogue;
- the difficulty of design and maintenance.

There have been many attempts to solve these problems, most concentrating on techniques and methodologies to ease the development and maintenance cycle, a few suggesting alternative forms of knowledge representation (including frames, scripts and neural nets) and even some proposing extended inference mechanisms through meta-interpreters. Despite all of these, however, todays expert systems function in essentially the same manner as did the pioneering systems of the 70's.

The majority of expert systems are still pure rule-based ones, although there is a trend towards hybrid systems, combining for example, rules and frames. Generally, knowledge is stored in the form of IF/THEN statements, either as production rules or Horn clauses (single consequent term), viz.

IF a and b and c and d THEN z

The inference engine is responsible for drawing the appropriate inferences. Thus, given such a rule, together with the fact that ALL its antecedent terms (a, b, c and d) are true, the inference engine can (deductively) draw the conclusion (z). In fact, there are two common forms of inference engine, forward chaining and backward chaining (also referred to as data-driven and goal-driven respectively). They both employ simple deduction to draw their conclusions, the difference between them lying in how facts are 'discovered'.

Backward chaining systems work by choosing a goal z, and attempting to prove it by establishing the truth of all its antecedents a, b, c and d. If any of these terms is itself the consequent of another rule, the inference engine will set this up as a sub goal; if, however, there are no other references to it, then the system will query-the-user for its value. If the goal proves false the system simply selects another one, and so on, until either one is proved or no more remain. In contrast to this, forward chaining systems generally require the user to enter any available data at the beginning of the consultation and then mechanically draw all the conclusions which are possible from the given data. For a detailed description of such systems see [2].

More recently, Davenport[3] has again suggested that many of the difficulties encountered with expert systems can indeed be traced to the form of knowledge representation employed. Although the critique was based on Horn-clause systems, it applies, in essence, to other forms of symbolic representation too, and results from their lack of a philosophically plausible view of the world. He went on to propose "inscriptors" as a step towards resolving these difficulties. We will now give a brief overview of inscriptors and then present an experimental expert system shell based on them.

Inscriptors

Inscriptors represent knowledge in the form of an IF/THEN statement having only a single antecedent term, but with any number of consequent terms all logically conjoined together, viz.

IF z THEN a and b and c and d and ...

All terms are positive propositions, no disjunctions are allowed and there are no certainty factprs or probabilities. Consequent terms in one inscriptor can be antecedent terms in another, but cycles are prohibited. An equivalent graphical representation comprising unlabeled directed arcs from a node representing the antecedent terms to each of the nodes representing the consequent terms, is helpful for visualisation.

An inscriptor is simply a "record of the co-occurrence of certain terms"; a statement of what was "the case". Associated with each inscriptor is a value which represents the systems current "belief" in the proposition that its antecedent term represents, i.e. its belief that the same conjunction of terms is again prevalent.

Inscriptors are intended to be interpreted as logical implications admitting the following inferences,

```
X -> Y        X -> Y        X -> Y        X -> Y
X             ~Y            ~X            Y
------        ------        ------        ------
Y             ~X            Y or ~Y       X or ~X

 (1)           (2)           (3)           (4)
```

The first, deduction (modus ponens), is a statement of fact, generating a description of X. For example, applied to

IF strawberry THEN red and fruit

it says that strawberries are red fruits. The second, (modus tollens) says that if any one of the consequent terms is not true then the antecedent cannot be, i.e. if the item we are considering is not a fruit then it cannot be a true strawberry. The third and fourth inference methods seem useless, however the fourth, abduction, is actually the key to the whole classification process. What it says, in this example, is that something that is a fruit, 'may be' a strawberry. Of course, it may well be something else, only when it stands as the only possibility can we be reasonably sure that this is the case. Within a set of such inscriptors, this can be achieved either by collecting negated terms and using modus tollens to eliminate possibilities, or, more usually, by collecting increasing numbers of positive terms such that our "belief" in inscriptors that match all of them goes up, while that of inscriptors which do not match all of them does not. Given enough information there will thus always be a clear winner.

KNOWALL! : Users Perspective

KNOWALL! is the prototype of an experimental inscriptor-based expert system shell. To the user, KNOWALL! can appear much like any existing expert system. It begins by loading knowledge from a given text file and displaying whatever messages the knowledge-base designer desired. Once this is complete the user prompt appears indicating that KNOWALL! is ready to accept commands. To begin a consultation the user should tell KNOWALL! something about their problem, for example, in an animal expert system the user may enter 'animal', indicating that what they want to identify is known to be an animal and that the system should determine exactly what sort of animal it is. Given this initial information KNOWALL! will prompt the user for more information by asking specific questions, eg. how many legs does it have? Given the answer to this, it will then ask another question, and another, and so on until the animal is uniquely identified (or until no further progress is possible because of the system's limited knowledge).

All this, of course, is exactly how existing expert systems function. What makes KNOWALL! different is that,

(a) the user can reply "don't know" to any question. In such cases the system will simply find another question to ask, and carry right on solving the problem;

(b) the user can volunteer information, positive or negative, at any time. For example, he/she may say 'not mammal' to narrow the possible choices; alternatively, rather than saying 'animal' right at the beginning, the user may immediately restrict the field by saying ''cat' or '2Legs' or whatever;

(c) the user can always ask for auxiliary information, eg. the ATTRIBUTES-OF an elephant, known INSTANCES-OF snakes, the DIFFERENCE between a dog and a wolf, whether penguins ARE red, or whether the animal being identified COULD possibly be a dolphin and if not why not!

As a result of these new facilities (themselves a consequence of the inscriptor philosophy), consultations with KNOWALL! can be very different. Whereas conventional expert systems often resemble simple decision trees, such that the same sequence of questions is followed at each consultation, in KNOWALL! which questions are asked depends entirely on what information the user volunteers. Also, while being unable to answer a question normally results in complete failure, KNOWALL! simply asks a different question and carries on regardless.

KNOWALL!'s method of selecting queries means that it naturally asks (almost) optimal questions. Moreover, since it does not have to establish the truth of ALL relevant terms to conclude something, consultations can be shorter and more directed. Consider, as an extreme, volunteering the fact that the animal is known to have a 'trunk'. KNOWALL! will immediately conclude elephant (the only possibility) without the user having to answer a single question; yet, if asked for the attributes of 'elephant' it can still list a dozen other features. Consider that an equivalent rule-based expert system would have had to ask twelve questions before coming to the same simple conclusion! Of course, in such systems the knowledge engineer may not have bothered to include all these attributes so as to make consultations more manageable, but then if the user wanted to know how many toes the elephant had, the system wouldn't know. Equally bad, if he/she had included it, the user probably wouldn't know the answer and hence would never discover that the animal with a trunk was an elephant!

Lastly, (but by no means least), a point which is invisible to the end user, but which will be very apparent to the knowledge engineer, is the independence of functioning with respect to rule/term order. New inscriptors can be added, and existing ones extended and even reordered with confidence that previously correct operations will not be compromised. This makes knowledge-base construction, testing and maintenance much easier and could lead to significant savings in time and money.

KNOWALL! : Implementation

KNOWALL!'s implementation (in Turbo Pascal on IBM PC compatible machines) is based on the graphical interpretation of inscriptors. Nodes have a name string, a state, score and difference score. Arcs, which need to be traversed both ways, are formed from each node by keeping a pointer to a list of consequent nodes (nodes below it) and a pointer to a list of nodes where the node is itself the consequent term (nodes above it).

The only other data structures used are the sets "GivenTrue" (which holds pointers to nodes which the user has told KNOWALL! are true), "GivenFalse" (which holds pointers to nodes given false), "Hypotheses" (which holds pointers to the set of nodes which KNOWALL! is currently trying to distinguish between) and "Concluded" (which holds pointers to any nodes which KNOWALL! has inferred must be true given the available information).

KNOWALL! begins by reading inscriptors from an external (text) file and constructing its internal representation of the knowledge- base. It then enters a loop, processing user input until commanded to "quit" the program. User input comprises either terms (node names volunteered true/false), the answer to the current question (true/false terms or unknown) or a request for auxiliary information.

All inferences are based on simple recursive graph/tree marking algorithms. Initially all scores are set to zero. When the user volunteers a true term, the corresponding node is added to the "Given" set, and it and all nodes below it have their state set TRUE. Then the "Hypotheses" set is emptied and all nodes above the given node have their scores incremented, in the process of which a new Hypotheses set is constructed from those nodes whose scores first equal the number of given nodes plus the number of concluded nodes. When the user volunteers a false term the node is added to the "GivenFalse" set and all nodes above the given node have their state set to FALSE and their scores decremented. Nodes encountered on route are removed from the Hypotheses set; see figure 1.

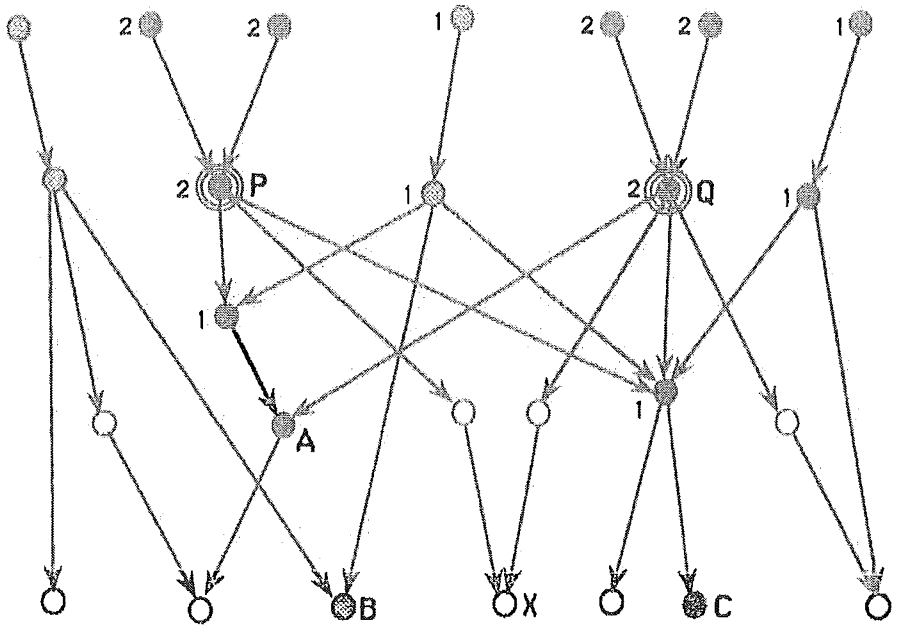


Figure 1. Generating hypotheses P & Q given A and C true and B false.

Following such inferences, if the "Hypotheses" set is empty then no solution is known, while if it has only one element in it then the corresponding node can be added to the Concluded set. If there is more than one element left in the "Hypotheses" set then KNOWALL! must attempt to find a question which will help resolve among the competing hypotheses. To find such a question it takes each node in the "Hypotheses" set and increments the difference score of every node below them. The node with maximum difference score (from those with zero score and state not marked as UNKNOWN) is used to construct a question; see figure 2.

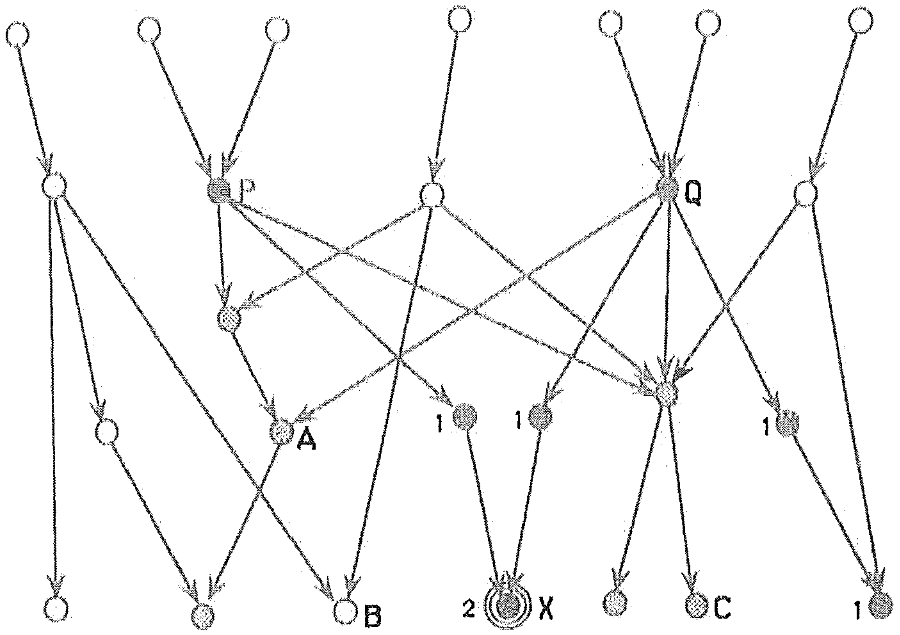


Figure 2. Determining suitable question to resolve between hypotheses P and Q, (choose X).

In replying to a question, the user may either respond "unknown" or state (in effect) that one or more nodes are true/false. This latter case is handled as above, while if the user responds "unknown", the state of the corresponding node is set to UNKNOWN (and hence excluded from further questions) and a new question is sought as before. Requests for auxiliary information are all handled in a similar fashion, and result in a list of node names and/or set contents being displayed, rather than a question being asked.

Conclusions

This paper began with a brief overview of inscriptors, a new form of knowledge representation designed specifically to help overcome the difficulties encountered in constructing real-world knowledge-based applications. It went on to describe KNOWALL!, an experimental expert system shell based on the inscriptor philosophy. This was seen to offer many advantages.

That inscriptors simply record 'observed facts' and that inference can be accomplished without regard to knowledge-base sequence, makes knowledge acquisition, validation and maintenance easier. Compared to existing expert systems, the user can ask a much wider range of questions, can volunteer information whenever he/she likes and can reply 'don't know' to any question. Since it does not have to establish the truth of everything it knows about something before concluding it, KNOWALL! can also accommodate incomplete information and even sometimes jump-to-conclusions, solving problems quicker than would a conventional rule-based system.

To conclude, KNOWALL! and the inscriptor theory it embodies, appear to provide solutions to most of the difficulties encountered in constructing classification type expert systems. It is, of course, experimental, and work is currently underway to see how well it copes in different applications and to see whether the range of commands offered is appropriate. Initial results are promising.

References

[1] Expert Systems for Experts; K. Parsaye & M. Chignell; John Wiley 1988

[2] Rule Based Expert Systems; Buchanan, B.G. & Shortliffe, E.H., Addison-Wesley, 1984

[3] "Towards a new generation of Expert Systems"; Davenport D., Computers & Information Sciences VI Vol 2, (ed) Baray & Ozguc, p651-659, Elsevier Science Pub. 1991

Örüntü Tanıma

Pattern Recognition

A Counterpropagation network model to recognize and classify chart patterns in automated manufacturing

D. T. Pham[1] and Z. Yilbas[2]

[1] School of Electrical, Electronic and Systems Engineering, University of Wales, College of Cardiff, P.O.Box 904, Cardiff CF1 3YH, U.K.

[2] Erciyes Universitesi, Muhendislik Fakultesi, Elektronik Muhendisligi Bolumu Kayseri, TURKIYE.

Abstract

The present study investigates applicability of the counterpropagation networks to recognise and classify the six chart patterns which may be produced from the industrial or/and commercial analysis. Counterpropagation networks produced initially fails to accomplish this goal due to overlapping of some of the chart patterns in data and test files. In order to avoid overlapping and complexity of the chart patterns, the data file is divided into sub-data files in which six chart patterns are included. Ten counterpropagation networks were trained using this sub-data files. Five of the networks having high performances are selected. A combined network is constructed using this five networks set and one other sixth network. The combined system then tested and it is found that the combined network system introduced in the present work gives improved performances for the chart patterns.

1. INTRODUCTION

Counterpropogation network was invented by Robert Hecht-Nielsen of Hecht-Nielsen Neuro-Computer Corporation[1]. The original basis for the network is not clearly known, but it appears to have developed as a means for synthesizing complex functions, much as backpropagation. It works as a "look-up" table, in parallel finding the closest example and reading out its equivalent mapping.

Counterpropogation network selects from a set of exemplars by allowing them to compete amongst each other. Normalized inputs and competition between exemplars the nearest neighbour. This provides a method of constructing an adaptive pattern classifier, function approximation and data compression [2].

The counterpropagation network developed formerly consist of bi-directional mapping network. X and a Y inputs are applied, which in turn interact together to select a particular processing element which is nearest to the composite vector (X,Y). The winner activates X' and Y'. In this way it acts as an be-lateral auto-associative network. The network gets its name from the counter-posing flow of information through it.

Counter propagation networks are usually trained to perform pattern mapping, the mapping of one particular pattern to another for entire set of patterns. The trained network classifies that pattern into a particular group by using a stored reference vector. This provides the target pattern associated with the reference vector as a output. In this case hidden layer performs a competitive classification grouping the patterns. It was shown that, counterpropagation works best when the patterns are tightly clustered distinct groups [3]. It was also shown that counterpropagation may provide an excellent example of a network that combines different layers from other samples to construct a new type of network [4].

Two types of layers different than each other are used in counterpropagation. This includes: i) The hidden layer is a Kohenen type, in this case competitive units accomplish unsupervised learning. ii) The top layer is the Grossberg layer. In this case this layer is fully interconnected to the hidden layer and is not competitive. The Grossberg layer is trained by a Widrow-Hoff or Grossberg rule.

Counterpropagation is considered a faster alternative to backpropagation. The improvement in the training time with counterpropagation is in general substantial. Counterpropagation can learn many pattern mapping problems well but, recent studies have shown that it often generalizes less well on new patterns [5]. It has been shown that counterpropagation requires that input pattern classes be organised into clusters that are seperated (non-overlapping) [6]. Therefore in the light of the previous work [5,6] the present study examines the recognition and classification of the six chart patterns ,which are introduced in data files, using counterpropagation algorithm. To achieve this goal, a variety of counterpropagation networks, which has different number of processing elements in their hidden layer, i.e. (60-35-6), (60-30-6), (60-9-6) and (60-7-6) are trained using the data file developed before [7]. The counterpropagation networks trained are tested by using test file and their performances are produced. However, the performance of each networks fails to recognise all the chart patterns. This may be due to the fact that the pattern classes are not clearly seperable in the data file. In order to overcome this tackle, the data file is divided into sub-data files in which chart patterns are presented avoiding the overlapping problem. This provides six chart patterns taking place in one sub-data file. Ten different counterpropagation networks are trained using ten different arbitrary selected sub-data files. Performances of these networks are examined for individual chart patterns.Two sets of five networks having better performance than the others are selected. Each set of five networks are combined to the sixth network which in turn produces combined network system, i.e. two combine network systems each having six counterpropagation networks are produced. These two combined network systems are then tested by using the all sub-data files. Finally, the performances of these combined networks are examined and hence on of the combined network system is selected due to its better performance.

2. NETWORK MODELLING

The data file developed previously [7] are divided into sub-data files such that each sub data file contains the data corresponding to six chart patterns. i.e. each sub-data file has 6x60 data. Figure 1 shows control chart patterns. Each network is trained and tested using with one of these sub-data files. This provides very high level learning (100%) for each network. After completing the training, some of the weak connections between Kohonen and the output layer of the network are deleted. After this process the simplification of the network is obtained. This procedure is repeated for the other networks.

In order to construct and analyse the combined network system, each network is studied individually. In the combined system, five networks are employed and each of them has 75 processing element (60-9-6). Their first layer has 60 processing elements and it acts as a buffer. The input vectors are "normalized" and have same length. There are 9 processing elements in the competitive (Kohonen) layer different number of processing elements are experimented and 9 processing element in the Kohonen layer are found to be suitable for this application. In this layer processing elements are computed and one of the highest output wins. For a given inputs, one and only one Kohonen neuron output is a logical one and all others are zero. 6 processing elements take place in the output layer. Output layer provides a way of decoding that single input to a meaningfull output class. Widrow-Hoff learning rule is used in this layer. The output layer fully connected to Kohonen layer, the Kohonen layer fully connected to the prior layer. In this case, it has same effects as the Grossberg Outstar learning rule, but easier to implement. Initially, ten networks were trained seperately, by using 360 (6x60) different data. Each network learned 100%. It should be noted that each network was trained with using one of the different arbitrary selected sub-data files. The performance of the each network can be defined as:

$$\text{Performance} = \frac{\text{Number of Patterns Correctly Classified}}{\text{Total Number of Tests for the Pattern}}$$

The combined network system consist of 6 networks; five working simultaneously to process the input data and producing output data to feed the sixth network. Consequently, the output of the sixth network is the output of the combined network system. The block diagram of the arrangements of the networks in the combined network system is shown in figure 2. To obtained a high accuracy for the combined network system two sets of networks are selected

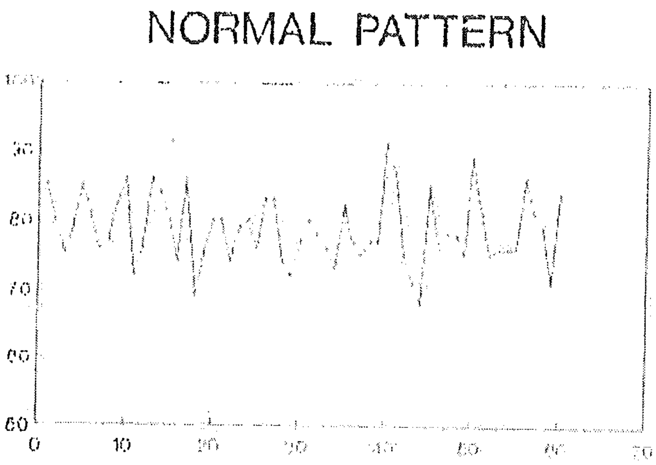


Figure 1.a. Pattern Number 1.

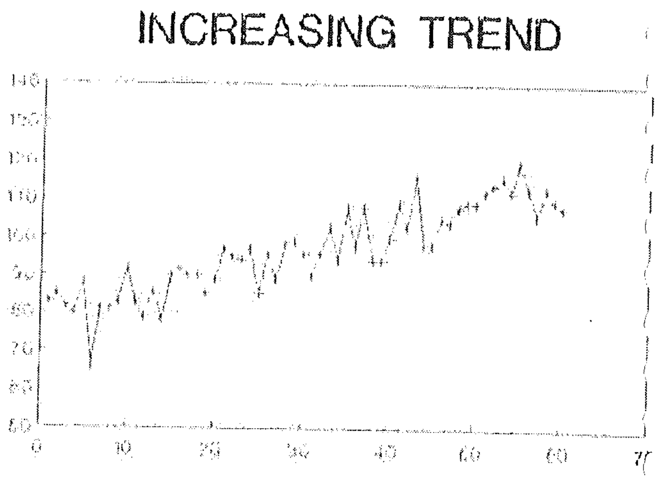


Figure 1.b. Pattern Number 2.

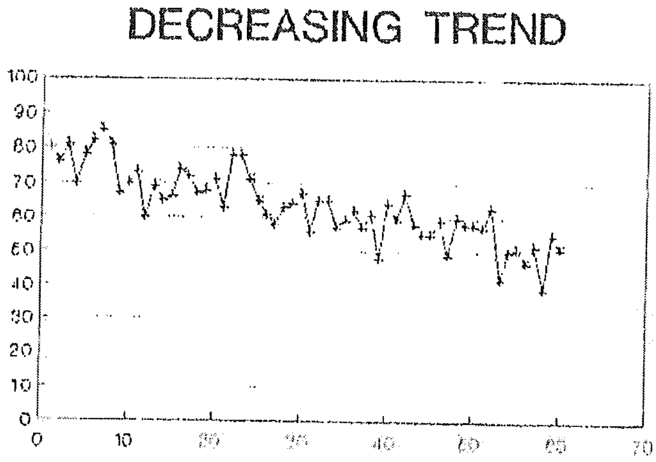


Figure 1.c. Pattern Number 3.

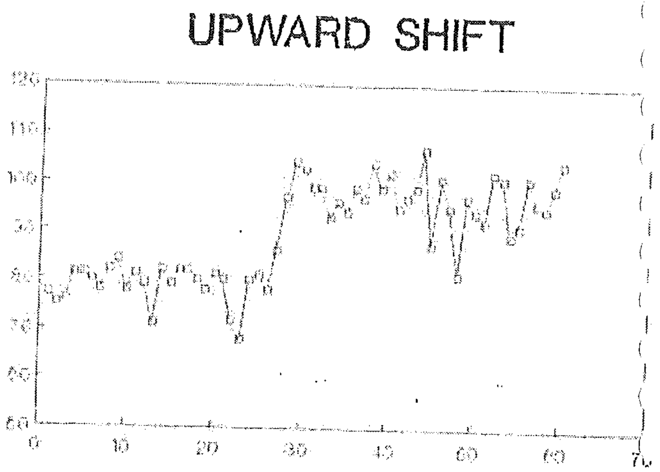


Figure 1.d. Pattern Number 4.

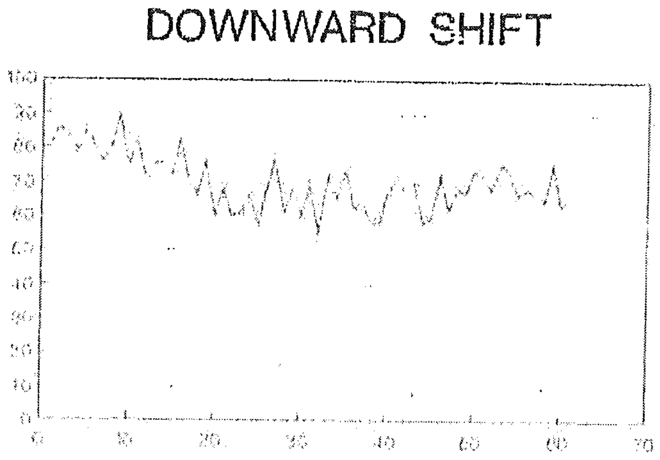


Figure 1.e. Pattern Number 5.

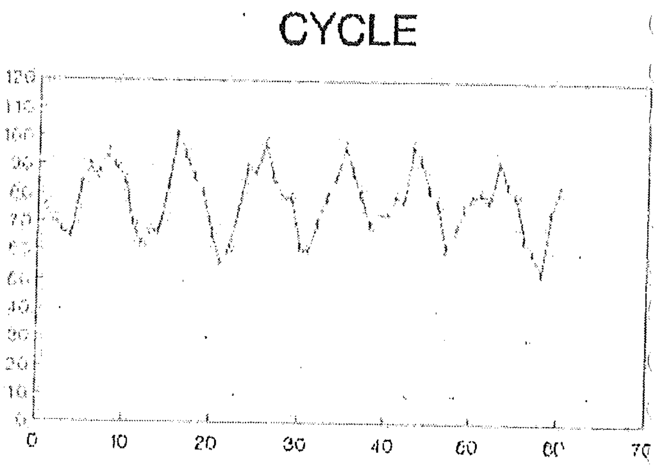


Figure 1.f. Pattern Number 6.

initially and their performances are compared. After the comparison, the set of networks having higher performance than the others are selected and this is employed in the combined network system. Finally, a data file is developed in a binary form to train the sixth network in the combined network system (Figure 2).

The sixth network in the combined network system has 42 processing elements (30-6-6). 30 of them are used in the input layer, 6 processing elements are used in the competitive (Kohonen) layer and last layer is the output layer with 6 processing elements. The data file for the sixth network contains 180 data -corresponding to 6 control chart patterns- obtained from the output of the set of five networks.

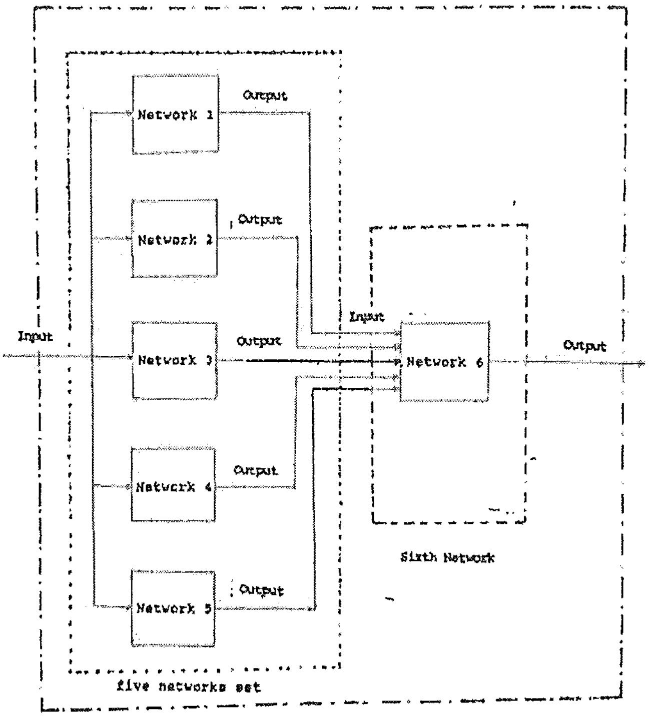


Figure 2. Block Diagram of The Combined Network System.

3. RESULT AND DISCUSSION

Each of the network in the combined network system is trained 3300-3400 times. After this training each of them learns 100%. So if the training times between counter-propagation and backpropagation is compared, counter-propagation nets learn faster than backpropagation nets. They are typically 10-100 times faster to train than conventional backpropagation; with the results that often comparable. The use of counterpropagation networks also provided rapid prototyping of the system. On the other hand, the hybrid schemes are not optimal in the sense that backpropagation is, since the hidden layer responses are not optimized with respect to the output performance. Figure 3 shows 3-D plot of performance with number of control chart patterns and number of network while figure 4 shows variation of performance with number of control chart pattern for ten networks. Performance of the networks varies with varying control chart patterns. This is due to the fact that each control chart pattern represented in different test files varies and networks are tested with the file different than their training files, which in turn gives variation in the performance. Two sets of five networks having high performance corresponding to all control chart patterns were selected from Figure 4.

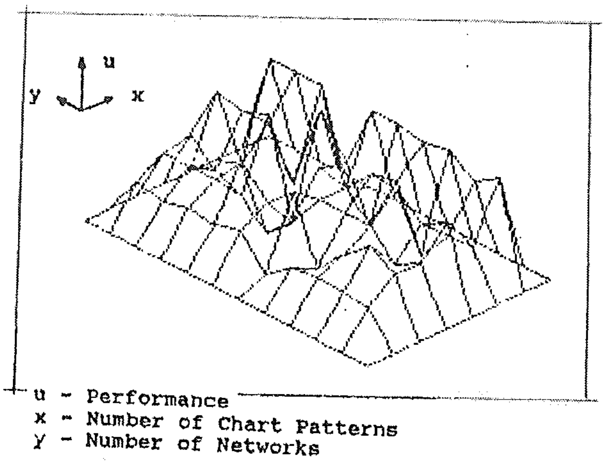


Figure 3. 3-D Plot of Ten Networks Performances With Chart Patterns.

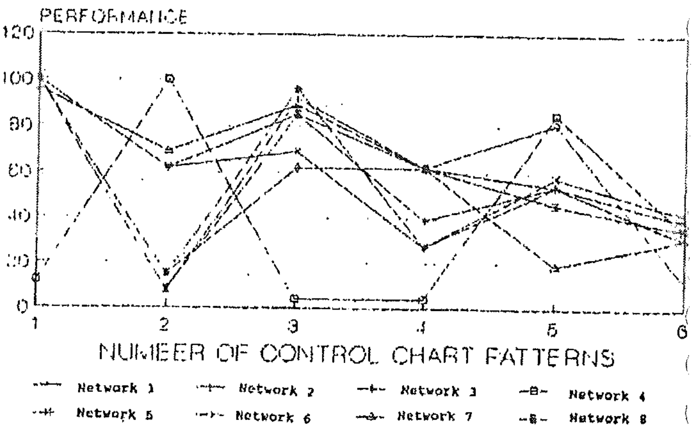
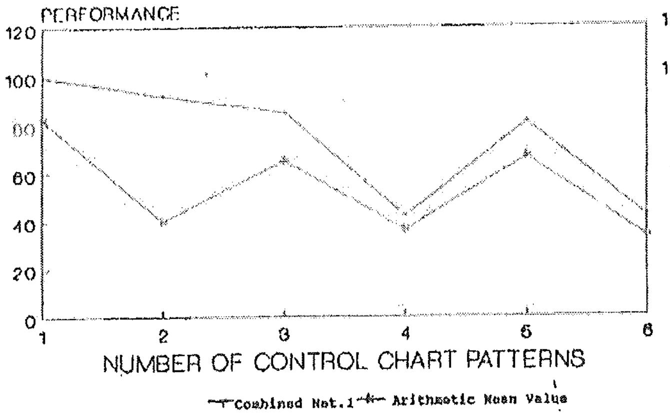


Figure 4. Variation of Performances of Networks With Chart Patterns.

Figure 5 shows the performance of the two combined network systems, providing that each combined network system employes a set of high performance five networks system selected before. The performances obtained from two combined network systems are almost the same. However, one of the combined network system corresponding to set-1 has slightly better performance than the other. Consequently, this combined network system is selected. It may be seen from Figure 5 that performance of the combined network system selected is high for the pattern numbers 1, 2, 3, 5 and it is especially low for the pattern number 6. In order to investigated the performance of each combined network system, the arithmetic mean value of the performances corresponding to each of five networks used in the combined system are plotted together with performance of the combined network system in figures 5.a and 5.b for the control chart patterns. It is obvious that the performance of the combined network system is higher than the aritmetic mean value of the performances corresponding to set of five networks. This clearly indicates that combined network system gives higher performance than the other trained networks.



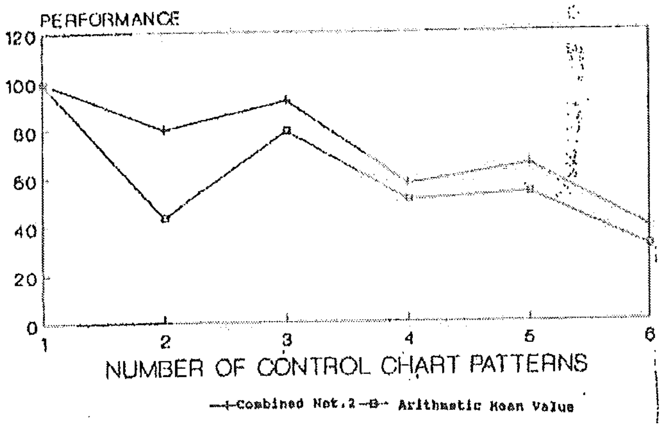


Figure 5.a.

Figure 5.b.

Figure 5.a. Comparison of Performances of Combined Network System Number 1 and Arithmetic Mean Value of The Performances of Five Set of Networks.

Figure 5.b. Comparison of Performances of Combined Network System Number 2 and Arithmetic Mean Value of The Performances of Five Set of Networks.

To study the combined network system, a definition of the performance of learning becomes necessary. This may be defined as:

$$\text{Performance of learning} = \frac{\text{Number of Chart Patterns Recognized}}{\text{Number of Tests}}$$

Figure 6 shows variation of performance of learning with control chart patterns number. It can be seen from figure 6.a that combined network system gives 100% performance for testing the control chart pattern number 1. On the other hand, combined network system gives 80% perforamce of learning for pattern number 2 and 20% for pattern number 4. This result is obtained when testing pattern number 2. However, when testing pattern number 3 with using combined network system, it gives 92% performance of learning for pattern number 3 and 8% performance of learning for pattern number 5. When testing the pattern number 4, the combined network system gives performances of learning 58% for pattern number 4 and 42% for pattern number 2. Similarly, combined network system gives 65% performance of learning for pattern number 5 and 35% performance of learning for pattern number 3, 8% for pattern number 4 and 39% performance of learning for pattern number 6. It can be seen from these figures that combined network system fails to learn 100% for pattern numbers 2, 3, 4, 5 and 6. It confuses with pattern number 2 and 4,

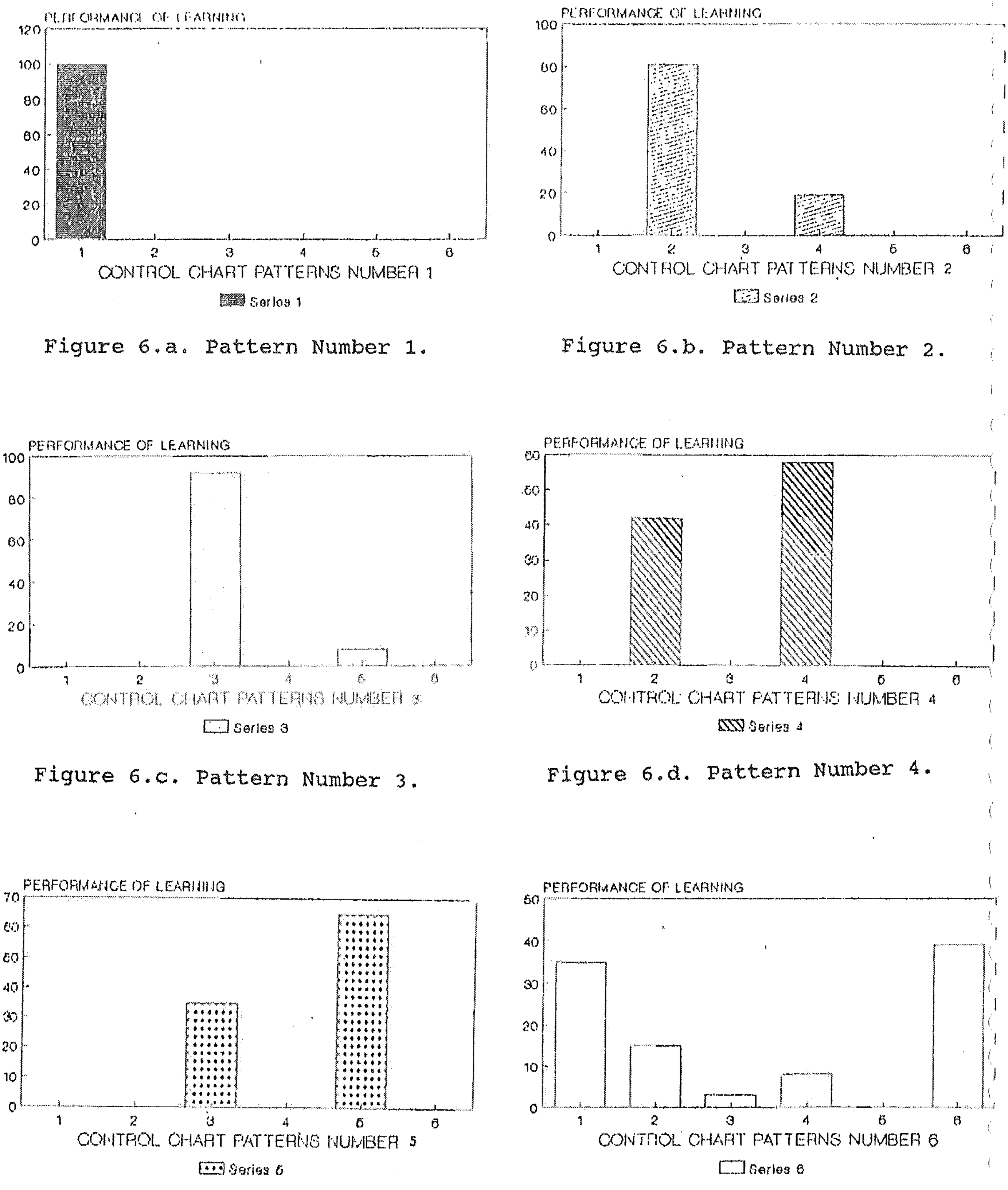


Figure 6.a. Pattern Number 1.

Figure 6.b. Pattern Number 2.

Figure 6.c. Pattern Number 3.

Figure 6.d. Pattern Number 4.

Figure 6.e. Pattern Number 5.

Figure 6.f. Pattern Number 6.

Figure 6. Variation of learning Performances of Combined System With Chart Patterns For Different Pattern Numbers.

and 3 and 5. This may be due to that the amplitudes and frequencies of these patterns are very similar and there may be overlapping of patterns occured. In this case it is very difficult to judge the relevant pattern number (Figure 6.f).

4. CONCLUSIONS

A counterpropagation network fails to recognise and classify the data developed previously for six chart patterns. However, an alternative method for the backpropagation may be developed using a counterpropagation combined network system. In this case the data file should be divided into sub-data files such that each sub-data file contains the data corresponding to six chart patterns. This is necessary, since overlapping of the chart patterns can be avoided. Consequently, counterpropagation networks are trained using different sub-data files and two sets of them are selected due to their better performances. The sixth network is introduced and trained using binary data file and it is added to these network sets which in turn produces a combined network system. The performance of the combined network system is found to be better than the any other individual network's performance. However, the combined network system fails to recognise and classify 100% for some chart patterns. This may be due to the fact that some of the chart patterns have very similar amplitudes and frequencies resulting in overlapping. On the other hand the performance of the combined network system developed in the present study gives improved performances and may be considered as sufficient for the application to the present problem. It is worth to mention here that training time for the combined system is considerably less than backpropagation networks developed for this purpose, i.e about 21000 times trained for the combined network system and over 90000 times trained for the backpropagation for the same sample.

5. REFERENCES

1. Hecht-Nielsen R., Counterpropagation networks, Proc. of IEEE First International Conference on Neural Networks, Part II, (1987) 19-32.

2. Hecht-Nielsen R., Applications of counterpropagation networks, Neural Networks, 2, (1988) 131-140.

3. Roitblat H.L., Moore P.W.B., Nachtigall P.E., Penner R.H. and Au W.W.L., Doplhin echolocation: Identification of returning echoes using a counterpropagation network, Proc. of International Joint Conference on Neural Networks, Part I, (1989) 295-299.

4. Glover D.,A hybrid optical/electronic neurocomputer machine vision inspection system, Proc. of Vision 88 Conference, Society Manufacturing Engineers, Dearborn, MI, (1988).

5. Hertz J., Krogh A and Palmer R.G.,Introduction to the theory of neural computation, Lecture Notes Volume 1, Addison-Wesley Publishing Company, New York, (1991).

6. Lippmann R.P.,Rewiev of neural networks for speech recognition, Neural Computation I, (1989) 1-38.

7. Pham D.T. and Oztemel E.,Using neural networks to recognise patterns of control charts in automated manufacturing, Send to Publisher, (1991).

8. Simpson P.K.,Artificial neural systems: Foundations, paradigms, applications and implementations, Pergamon Press, New York, (1990).

Experiments with RST, A Rotation, Scaling and Translation Invariant Pattern Classification System

Cem Yüceer and Kemal Oflazer

Department of Computer Engineering and Information Science
Bilkent University, Bilkent, 06533 Ankara, Türkiye

Abstract

This paper includes an overview and experimental results on RST, the hybrid pattern classification system. It can recognize patterns even when they are deformed by a transformation like rotation, scaling, and translation or a combination of these [11]. The system is formed of a Karhunen-Loéve transform based pattern preprocessor, an artificial neural network classifier and an interpreter. After a description of the system architecture, experimental results are provided from three different classification domains: classification of letters in the English alphabet, classification of the letters in the Japanese Katakana alphabet, and classification of five main geometric figures. The system is general purpose and has a reasonable noise tolerance.

1 Introduction

The recent interest in artificial neural networks, machine learning, and parallel computation has led to renewed research in the area of pattern recognition. Pattern recognition aims to extract information about the image and/or classify its contents. Systems having pattern recognition ability have many possible applications in a wide variety of areas, from simple object existence checks, through identity verification, to robot guidance in space exploration. Pattern classification, a subfield of pattern recognition, is concerned with determining whether the pattern in an input image belongs to one of the predefined classes. Many researchers studied parametric Bayesian classifiers where the form of input distributions is assumed to be known and parameters of distributions are estimated using techniques that require simultaneous access to all training data. These classifiers, especially those that assume Gaussian distributions, are still the most widely used since they are simple and are clearly described in a number of textbooks [4, 5]. However, the thrust of recent research has changed. More attention is being paid to practical issues as pattern classification techniques are being applied to speech, vision, robotics, and artificial intelligence applications where real-time response with complex real world data is necessary. In all cases, pattern classification systems should be able to learn while or before performing, and make decisions depending on the recognition result.

Developing pattern recognition systems is usually a two-stage process. First, the designer should carefully examine the characteristics of the pattern environment. The result is a set of features chosen to represent the original input image. Second, the designer should choose from a variety of techniques to classify the pattern which is now in feature representation. The stage of feature determination and extraction strictly determines the success of the system, since from thereon the image is represented by this feature form. Therefore, it is highly desired that the classification system itself should extract the necessary features to differentiate the example patterns that represent each class. In other words, the system should be automated to work by itself and should not depend on the human designer's success in defining the features. Further, these features should be chosen such that they should tolerate the differentiation between the patterns in the same class. The system should also have the ability to perform the classification in a rotation, scaling, and translation invariant manner. This effect is typical when the scanning device, suppose a camera, changes its orientation or distance from the specimen. Hence the image fed to the system may contain a pattern that is rotated, scaled, or translated compared to its original form when it was first presented to the system. For such a case, either the system should employ features that are invariant to such transformations or there should be a preprocessor to maintain the rotational, scaling, and translational invariancy. Even for a limited system designed for classifying only a determined type of patterns – an optical character classifier, or an identity verifier – it is hard to find features that extract useful information while maintaining the mentioned invariancies. The problem will be impractical if such a system is intended for general purpose classification, or to say it is aimed to classify any type of patterns. Artificial neural networks have recently been used for automatic feature extraction and/or pattern classification mainly owing to their learning algorithms, generalization ability and noise tolerance [1, 2, 3, 6, 8, 10, 11].

2 The Pattern Classification System

Selecting good features for relatively complex patterns, like the human face or finger prints, turns out to be impractical or even impossible [9]. The problem is more acute when there is no prior knowledge on the patterns to be classified. Therefore, a system to automatically extract the useful features is essential. Artificial neural networks extract information during the training process. RST, the pattern classification system first presented in [11] has a modular structure consisting of three main blocks, a preprocessor, a classifier, and an interpreter. The blocks are cascaded in order such that the original image is first preprocessed, then classified, and finally the results are interpreted.

2.1 PREP1: The Preprocessor with Radial Scaling Correction

The preprocessor has three cascaded blocks *R-Block*, *S-Block*, and *T-Block*. The R-Block maintains rotational invariancy, S-Block maintains scaling invariancy, and T-Block maintains translational invariancy. The order in which the blocks are cascaded is determined mainly by the functional dependencies between these blocks. In the first implementation of the preprocessor, PREP1, the T-Block comes first, S-Block second, and R-Block last.

The T-block maintains translational invariancy by computing the center of gravity of the pattern and translating the image so that the center of gravity coincides with the origin. The resulting image is passed to the S-Block. The center of gravity, (x_{av}, y_{av}), is computed by averaging the x and y coordinates of the on-pixels. The mapping function for the translation invariant image is:

$$f_T(x_i, y_j) = f(x_i - x_{av}, y_j - y_{av}) \tag{1}$$

where function $f(x, y)$ gives the value of the pixel at the coordinates (x, y). For digitized binary-valued 2-D images this function will be either 0 or 1.[1]

The S-Block maintains scaling invariancy by scaling the image so that the average radius for the on-pixels is equal to one-fourth of the grid size. The term *radius* for a pixel is defined to be the length of the straight line connecting the pixel and the origin. The scale factor is:

$$s = \frac{R}{\frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{N} f_T(x_i,y_j)} \sum_{i=1}^{N}\sum_{j=1}^{N} f_T(x_i, y_j) \cdot \sqrt{x_i^2 + y_j^2}} \tag{2}$$

where R is equal to one-fourth of the grid size. The mapping function for the scaling invariant image is:

$$f_{TS}(x_i, y_j) = f_T(s \cdot x_i, s \cdot y_j) \tag{3}$$

The R-block maintains rotational invariancy by rotating the image so that the direction of maximum variance coincides with the *x-axis*. The derivation of the function is based on the Karhunen-Loéve transformation which has been used in some applications [7, 9, 10, 11]. The transformation exploits the following: given a set of vectors, the eigenvector that corresponds to the largest eigenvalue of the covariance matrix calculated from the set of vectors, points in the direction of maximum variance [7, 9]. This property is used to maintain rotational invariancy since detection of the maximum variance direction will also reveal the rotation angle. Define:

$$T_{xx} = \sum_{i=1}^{N}\sum_{j=1}^{N} f_{TS}(x_i, y_j) \cdot x_i^2 \qquad T_{yy} = \sum_{i=1}^{N}\sum_{j=1}^{N} f_{TS}(x_i, y_j) \cdot y_j^2 \qquad T_{xy} = \sum_{i=1}^{N}\sum_{j=1}^{N} f_{TS}(x_i, y_j) \cdot x_i \cdot y_j \tag{4}$$

The *sine* and *cosine* of the rotation angle will be:

$$\sin\theta = \frac{(T_{yy} - T_{xx}) + \sqrt{(T_{yy} - T_{xx})^2 + 4 \cdot T_{xy}^2}}{\sqrt{2 \cdot \left[\sqrt{(T_{yy} - T_{xx})^2 + 4 \cdot T_{xy}^2} + (T_{yy} - T_{xx})\right] \sqrt{(T_{yy} - T_{xx})^2 + 4 \cdot T_{xy}^2}}} \tag{5}$$

[1] If x or y, the arguments of the function, are not integers then they are rounded to the nearest integer to obtain the pixel coordinates.

$$\cos\theta = \frac{2 \cdot T_{xy}}{\sqrt{2 \cdot \left[\sqrt{(T_{yy} - T_{xx})^2 + 4 \cdot T_{xy}^2} + (T_{yy} - T_{xx})\right]} \sqrt{(T_{yy} - T_{xx})^2 + 4 \cdot T_{xy}^2}} \tag{6}$$

The mapping function for the rotation invariant image is:

$$f_{TSR}(x_i, y_j) = f_{TS}(\cos\theta \cdot x_i + \sin\theta \cdot y_j\,,\ -\sin\theta \cdot x_i + \cos\theta \cdot y_j) \tag{7}$$

2.2 PREP2: The Preprocessor with Axial Scaling Correction

PREP1, performs radial scaling correction. That is, it uses the same scaling factor along all directions. This approach will perform better in certain applications like patterns being scanned in different resolutions in different dimensions. In preprocessor with axial scaling correction, called PREP2, the main blocks are reordered, such that T-Block comes first, R-Block is second, and S-Block is last. Further, the scaling factors are computed using a different function. In order to maintain rotational, scaling, and translational invariancy PREP2 computes various relevant parameters as:

$$T_{xx} = \left(\sum_{i=1}^{N}\sum_{j=1}^{N} f(x_i, y_j) \cdot x_i^2\right) - P \cdot x_{av}^2 \qquad T_{yy} = \left(\sum_{i=1}^{N}\sum_{j=1}^{N} f(x_i, y_j) \cdot y_j^2\right) - P \cdot y_{av}^2 \tag{8}$$

$$T_{xy} = \left(\sum_{i=1}^{N}\sum_{j=1}^{N} f(x_i, y_j) \cdot x_i \cdot y_j\right) - P \cdot x_{av} \cdot y_{av} \tag{9}$$

$$s_x = \sqrt{\frac{R_x \cdot P}{T_{xx}(\cos\theta)^2 + 2 \cdot T_{xy} \cdot \cos\theta \cdot \sin\theta + T_{yy}(\sin\theta)^2}} \tag{10}$$

$$s_y = \sqrt{\frac{R_y \cdot P}{T_{xx}(\sin\theta)^2 - 2 \cdot T_{xy} \cdot \cos\theta \cdot \sin\theta + T_{yy}(\cos\theta)^2}} \tag{11}$$

where s_x and s_y are the scaling factors, and R_x and R_y are the desired deviation values along the corresponding axes. R_x and R_y are equal to the grid size.

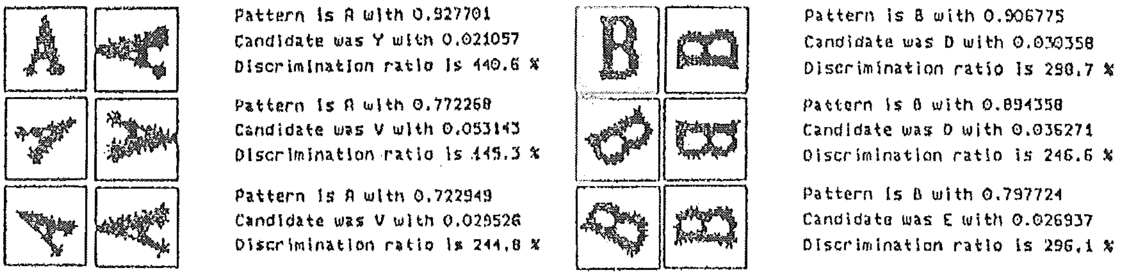
The mapping function for the preprocessor with axial scaling correction is:

$$\begin{aligned} f_{TRS}(x_i, y_j) &= f(s_x \cdot (\cos\theta \cdot (x_i - x_{av}) + \sin\theta \cdot (y_j - y_{av}))\,, \\ &\quad s_y \cdot (-\sin\theta \cdot (x_i - x_{av}) + \cos\theta \cdot (y_j - y_{av}))) \end{aligned} \tag{12}$$

2.3 The Classifier and The Interpreter

The current implementation of the system employs a multilayer feed-forward network for the classifier block. Such a network has a layered structure and only connections between neurons at subsequent layers are permitted. The training algorithm is the widely used *backpropagation algorithm.* Since the input is an image in pixel-map form, the number of nodes in the input layer is fixed and equal to the number of pixels. Further, since the output neurons are organized such that each node represents a class, the number of output neurons is also fixed. An output neuron having a value close to 1 is interpreted as a strong membership, while a value close to 0 will point a loose membership. Hence, given the input and output layer size one should decide on the number of hidden layers and the number of neurons in each hidden layer as well as the learning rate. The general structure of the neural network classifier was presented in [11]. The output of the preprocessor, which is an image in pixel-map form, is converted to a linear array by cascading the rows of the image from the top row to the bottom row. The content of each array entry is the initial input value of the corresponding input node.

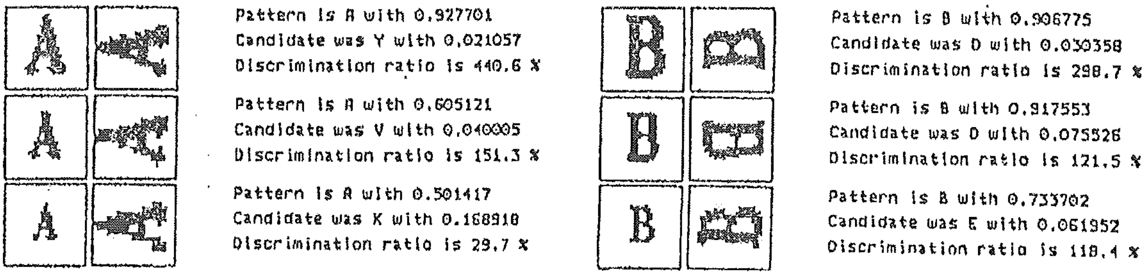
The method we have used in the interpreter block is to report *no discrimination* as long as the ratio of the maximum output to the next highest output remains under a predetermined threshold value. When the ratio exceeds the threshold, which means that the maximum output is dominant on the other outputs, the interpreter decides on the class with the maximum output. If not, then the interpreter reports that no unique discrimination could be made. This simple method has been observed to perform well in the evaluation of the classifier outputs.



```
Pattern is A with 0.927701
Candidate was Y with 0.021057
Discrimination ratio is 440.6 %

Pattern is A with 0.772268
Candidate was V with 0.053143
Discrimination ratio is 445.3 %

Pattern is A with 0.722949
Candidate was V with 0.029526
Discrimination ratio is 244.8 %
```

```
Pattern is B with 0.906775
Candidate was D with 0.030358
Discrimination ratio is 298.7 %

Pattern is B with 0.894358
Candidate was D with 0.036271
Discrimination ratio is 246.6 %

Pattern is B with 0.797724
Candidate was E with 0.026937
Discrimination ratio is 296.1 %
```

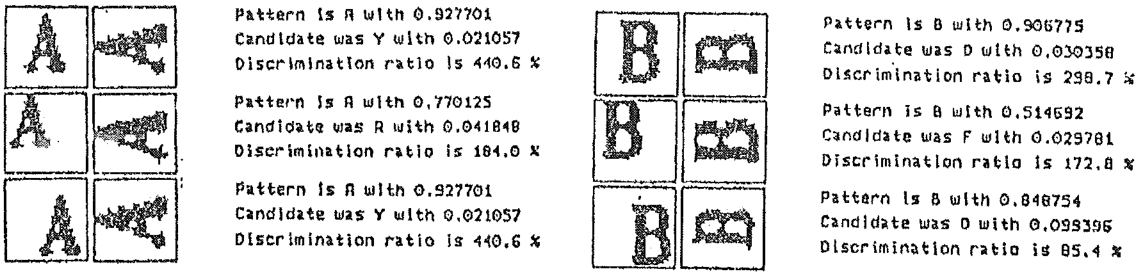
Figure 1: Classification results with PREP1 for letters A and B rotated by 0, 60, and -60 degrees.



```
Pattern is A with 0.927701
Candidate was Y with 0.021057
Discrimination ratio is 440.6 %

Pattern is A with 0.605121
Candidate was V with 0.040005
Discrimination ratio is 151.3 %

Pattern is A with 0.501417
Candidate was K with 0.168918
Discrimination ratio is 29.7 %
```

```
Pattern is B with 0.906775
Candidate was D with 0.030358
Discrimination ratio is 298.7 %

Pattern is B with 0.917553
Candidate was D with 0.075526
Discrimination ratio is 121.5 %

Pattern is B with 0.733702
Candidate was E with 0.061952
Discrimination ratio is 118.4 %
```

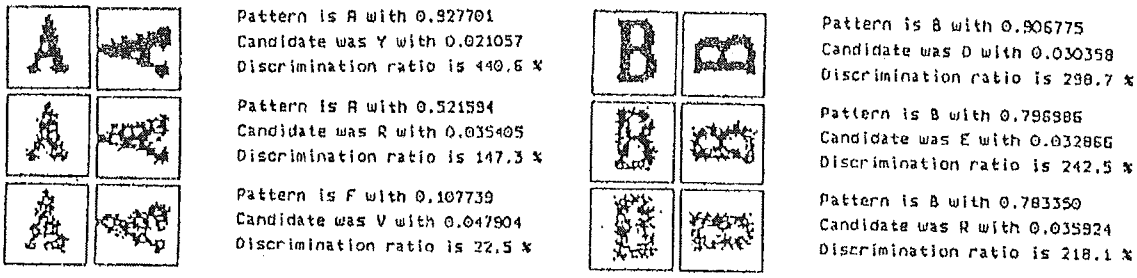
Figure 2: Classification results with PREP1 for letters A and B scaled by a factor of 1, 0.8, and 0.6.



```
Pattern is A with 0.927701
Candidate was Y with 0.021057
Discrimination ratio is 440.6 %

Pattern is A with 0.770125
Candidate was R with 0.041848
Discrimination ratio is 184.0 %

Pattern is A with 0.927701
Candidate was Y with 0.021057
Discrimination ratio is 440.6 %
```

```
Pattern is B with 0.906775
Candidate was D with 0.030358
Discrimination ratio is 298.7 %

Pattern is B with 0.514692
Candidate was F with 0.029781
Discrimination ratio is 172.8 %

Pattern is B with 0.848754
Candidate was D with 0.099396
Discrimination ratio is 85.4 %
```

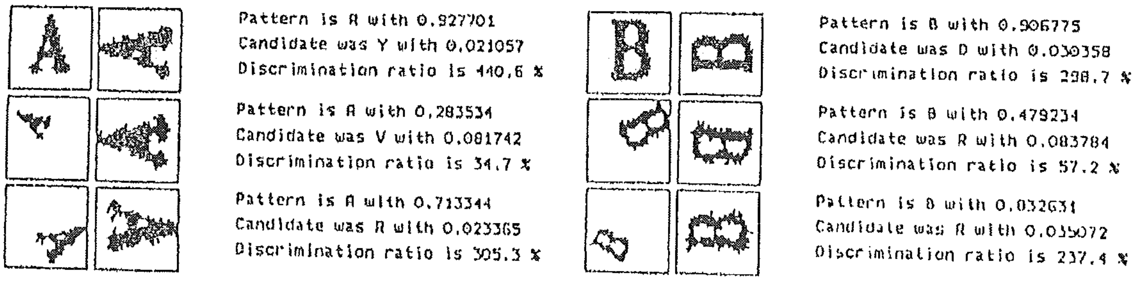
Figure 3: Classification results with PREP1 for letters A and B translated diagonally by 0, 6, and -6 pixels.



```
Pattern is A with 0.927701
Candidate was Y with 0.021057
Discrimination ratio is 440.6 %

Pattern is A with 0.521594
Candidate was R with 0.035405
Discrimination ratio is 147.3 %

Pattern is F with 0.107739
Candidate was V with 0.047904
Discrimination ratio is 22.5 %
```

```
Pattern is B with 0.906775
Candidate was D with 0.030358
Discrimination ratio is 298.7 %

Pattern is B with 0.796986
Candidate was E with 0.032866
Discrimination ratio is 242.5 %

Pattern is B with 0.783350
Candidate was R with 0.035924
Discrimination ratio is 218.1 %
```

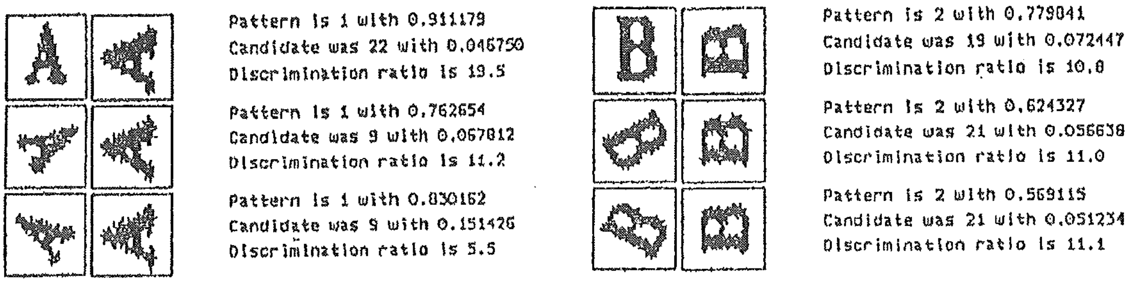
Figure 4: Classification results with PREP1 for letters A and B with 0%, 20%, and 40% noise.



```
Pattern is A with 0.927701
Candidate was Y with 0.021057
Discrimination ratio is 440.6 %

Pattern is A with 0.283534
Candidate was V with 0.081742
Discrimination ratio is 34.7 %

Pattern is A with 0.713344
Candidate was A with 0.023365
Discrimination ratio is 305.3 %
```

```
Pattern is B with 0.906775
Candidate was D with 0.030358
Discrimination ratio is 298.7 %

Pattern is B with 0.479234
Candidate was R with 0.083784
Discrimination ratio is 57.2 %

Pattern is B with 0.032631
Candidate was A with 0.035072
Discrimination ratio is 237.4 %
```

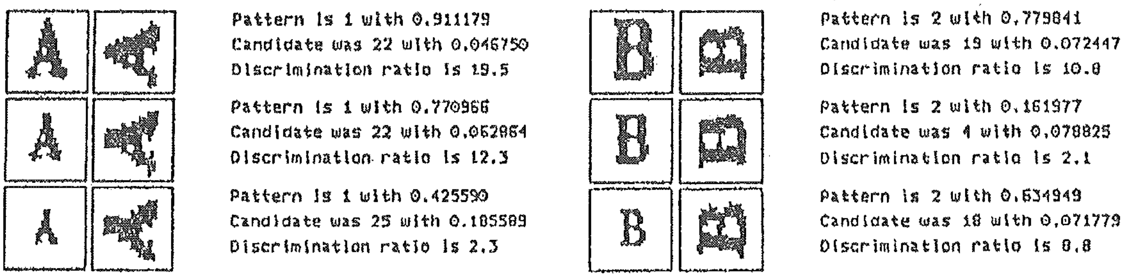
Figure 5: Classification results with PREP1 for letters A and B with random translation, scaling, and rotation applied.



```
Pattern is 1 with 0.911179
Candidate was 22 with 0.046750
Discrimination ratio is 19.5

Pattern is 1 with 0.762654
Candidate was 9 with 0.067812
Discrimination ratio is 11.2

Pattern is 1 with 0.830162
Candidate was 9 with 0.151426
Discrimination ratio is 5.5
```

```
Pattern is 2 with 0.779041
Candidate was 19 with 0.072447
Discrimination ratio is 10.8

Pattern is 2 with 0.624327
Candidate was 21 with 0.056638
Discrimination ratio is 11.0

Pattern is 2 with 0.569115
Candidate was 21 with 0.051234
Discrimination ratio is 11.1
```

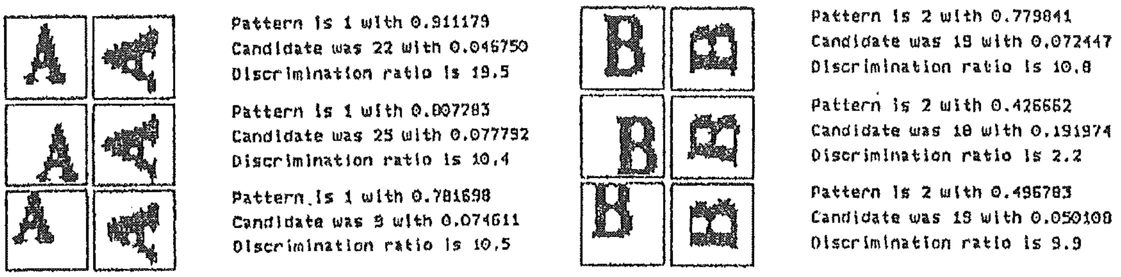
Figure 6: Classification results with PREP2 on rotated letters.



```
Pattern is 1 with 0.911179
Candidate was 22 with 0.046750
Discrimination ratio is 19.5

Pattern is 1 with 0.770966
Candidate was 22 with 0.062864
Discrimination ratio is 12.3

Pattern is 1 with 0.425590
Candidate was 25 with 0.185589
Discrimination ratio is 2.3
```

```
Pattern is 2 with 0.779841
Candidate was 19 with 0.072447
Discrimination ratio is 10.8

Pattern is 2 with 0.161977
Candidate was 4 with 0.078825
Discrimination ratio is 2.1

Pattern is 2 with 0.634949
Candidate was 18 with 0.071779
Discrimination ratio is 8.8
```

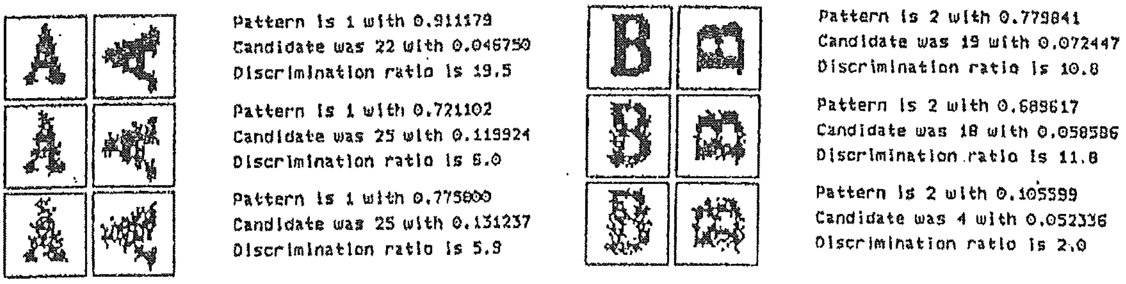
Figure 7: Classification results with PREP2 on scaled letters.



```
Pattern is 1 with 0.911179
Candidate was 22 with 0.046750
Discrimination ratio is 19.5

Pattern is 1 with 0.807283
Candidate was 25 with 0.077792
Discrimination ratio is 10.4

Pattern is 1 with 0.781698
Candidate was 9 with 0.074611
Discrimination ratio is 10.5
```

```
Pattern is 2 with 0.779841
Candidate was 19 with 0.072447
Discrimination ratio is 10.8

Pattern is 2 with 0.426662
Candidate was 18 with 0.191974
Discrimination ratio is 2.2

Pattern is 2 with 0.496783
Candidate was 19 with 0.050108
Discrimination ratio is 9.9
```

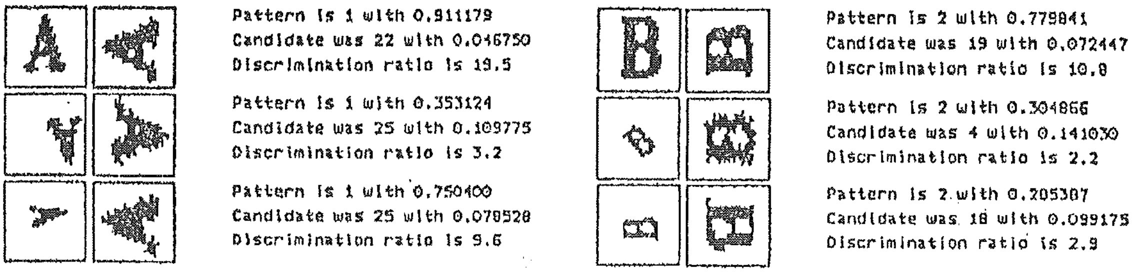
Figure 8: Classification results with PREP2 on translated letters.



```
Pattern is 1 with 0.911179
Candidate was 22 with 0.046750
Discrimination ratio is 19.5

Pattern is 1 with 0.721102
Candidate was 25 with 0.119924
Discrimination ratio is 6.0

Pattern is 1 with 0.775800
Candidate was 25 with 0.131237
Discrimination ratio is 5.9
```

```
Pattern is 2 with 0.779841
Candidate was 19 with 0.072447
Discrimination ratio is 10.8

Pattern is 2 with 0.689617
Candidate was 18 with 0.058586
Discrimination ratio is 11.8

Pattern is 2 with 0.105599
Candidate was 4 with 0.052336
Discrimination ratio is 2.0
```

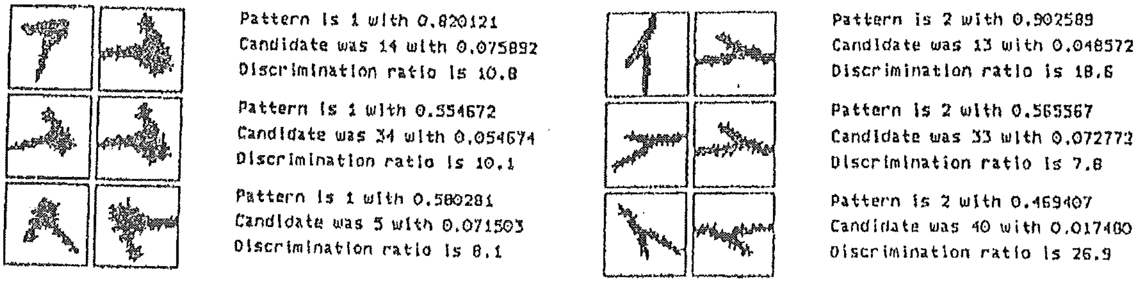
Figure 9: Classification results with PREP2 on noisy letters.



```
Pattern is 1 with 0.911179
Candidate was 22 with 0.046750
Discrimination ratio is 19.5

Pattern is 1 with 0.353124
Candidate was 25 with 0.109775
Discrimination ratio is 3.2

Pattern is 1 with 0.750400
Candidate was 25 with 0.078528
Discrimination ratio is 9.6
```

```
Pattern is 2 with 0.779841
Candidate was 19 with 0.072447
Discrimination ratio is 10.8

Pattern is 2 with 0.304866
Candidate was 4 with 0.141030
Discrimination ratio is 2.2

Pattern is 2 with 0.205307
Candidate was 18 with 0.099175
Discrimination ratio is 2.9
```

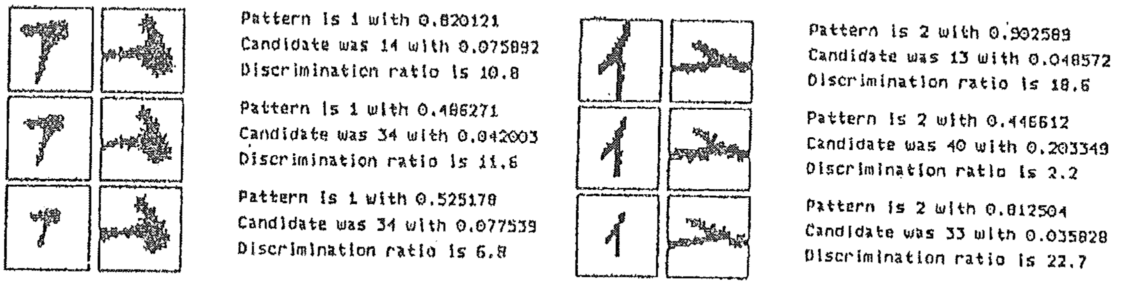
Figure 10: Classification results with PREP2 on letters with random translation, scaling, and rotation applied.



```
Pattern is 1 with 0.820121
Candidate was 14 with 0.075892
Discrimination ratio is 10.8

Pattern is 1 with 0.554672
Candidate was 34 with 0.054674
Discrimination ratio is 10.1

Pattern is 1 with 0.580281
Candidate was 5 with 0.071503
Discrimination ratio is 8.1
```

```
Pattern is 2 with 0.902589
Candidate was 13 with 0.048572
Discrimination ratio is 18.6

Pattern is 2 with 0.565567
Candidate was 33 with 0.072772
Discrimination ratio is 7.8

Pattern is 2 with 0.469407
Candidate was 40 with 0.017480
Discrimination ratio is 26.9
```

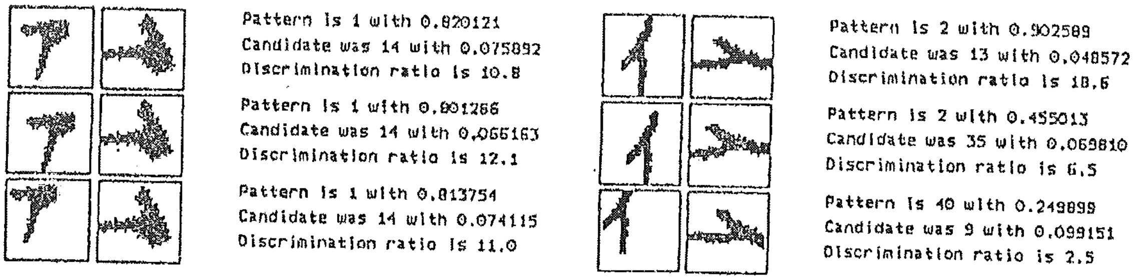
Figure 11: Classification results with PREP1 for symbols from Class 1 and 2 rotated by 0, 60, and -60 degrees.



```
Pattern is 1 with 0.820121
Candidate was 14 with 0.075892
Discrimination ratio is 10.8

Pattern is 1 with 0.486271
Candidate was 34 with 0.042003
Discrimination ratio is 11.6

Pattern is 1 with 0.525178
Candidate was 34 with 0.077539
Discrimination ratio is 6.8
```

```
Pattern is 2 with 0.902589
Candidate was 13 with 0.048572
Discrimination ratio is 18.6

Pattern is 2 with 0.446612
Candidate was 40 with 0.203349
Discrimination ratio is 2.2

Pattern is 2 with 0.812504
Candidate was 33 with 0.035828
Discrimination ratio is 22.7
```

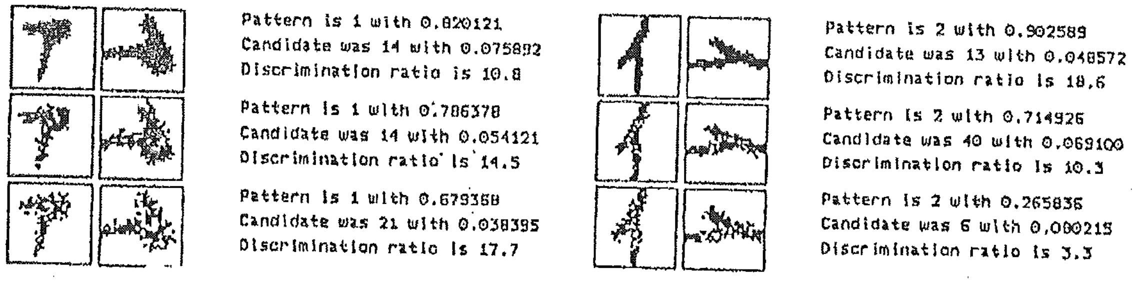
Figure 12: Classification results with PREP1 for symbols from Class 1 and 2 scaled by a factor of 1, 0.8, and 0.6.



```
Pattern is 1 with 0.820121
Candidate was 14 with 0.075892
Discrimination ratio is 10.8

Pattern is 1 with 0.801266
Candidate was 14 with 0.066163
Discrimination ratio is 12.1

Pattern is 1 with 0.813754
Candidate was 14 with 0.074115
Discrimination ratio is 11.0
```

```
Pattern is 2 with 0.902589
Candidate was 13 with 0.048572
Discrimination ratio is 18.6

Pattern is 2 with 0.455013
Candidate was 35 with 0.069810
Discrimination ratio is 6.5

Pattern is 40 with 0.249899
Candidate was 9 with 0.099151
Discrimination ratio is 2.5
```

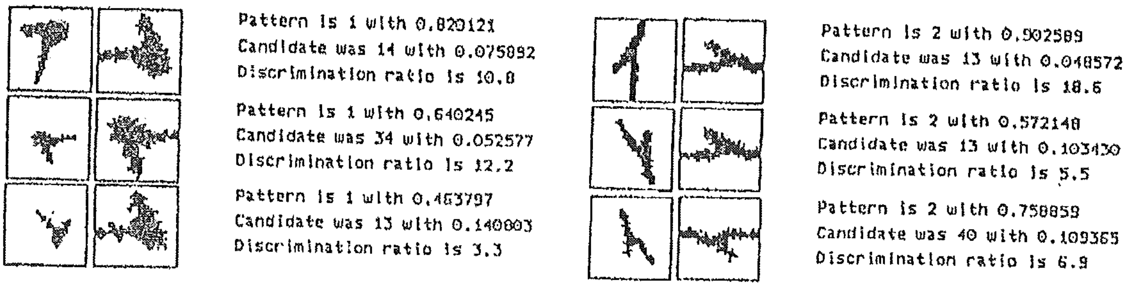
Figure 13: Classification results with PREP1 for symbols from Class 1 and 2 translated diagonally by 0, 6, and -6 pixels.



```
Pattern is 1 with 0.820121
Candidate was 14 with 0.075892
Discrimination ratio is 10.8

Pattern is 1 with 0.786378
Candidate was 14 with 0.054121
Discrimination ratio is 14.5

Pattern is 1 with 0.679358
Candidate was 21 with 0.038395
Discrimination ratio is 17.7
```

```
Pattern is 2 with 0.902589
Candidate was 13 with 0.048572
Discrimination ratio is 18.6

Pattern is 2 with 0.714926
Candidate was 40 with 0.069100
Discrimination ratio is 10.3

Pattern is 2 with 0.265836
Candidate was 6 with 0.080215
Discrimination ratio is 3.3
```

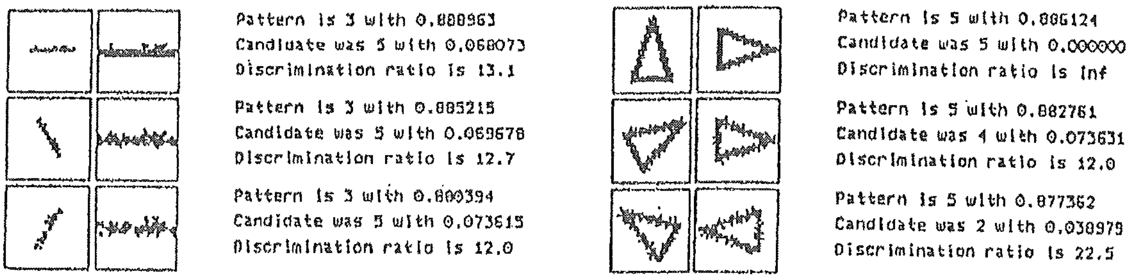
Figure 14: Classification results with PREP1 for symbols from Class 1 and 2 with 0%, 20%, and 40% noise.



```
Pattern is 1 with 0.820121
Candidate was 14 with 0.075892
Discrimination ratio is 10.8

Pattern is 1 with 0.640245
Candidate was 34 with 0.052577
Discrimination ratio is 12.2

Pattern is 1 with 0.463797
Candidate was 13 with 0.140803
Discrimination ratio is 3.3
```

```
Pattern is 2 with 0.902589
Candidate was 13 with 0.048572
Discrimination ratio is 18.6

Pattern is 2 with 0.572148
Candidate was 13 with 0.103430
Discrimination ratio is 5.5

Pattern is 2 with 0.758859
Candidate was 40 with 0.109365
Discrimination ratio is 6.9
```

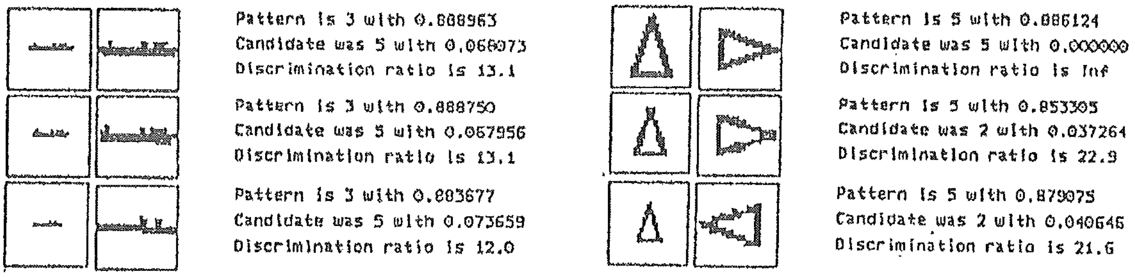
Figure 15: Classification results with PREP1 for symbols from Class 1 and 2 with random translation, scaling, and rotation applied.



```
Pattern is 3 with 0.888963
Candidate was 5 with 0.068073
Discrimination ratio is 13.1

Pattern is 3 with 0.885215
Candidate was 5 with 0.069678
Discrimination ratio is 12.7

Pattern is 3 with 0.888394
Candidate was 5 with 0.073615
Discrimination ratio is 12.0
```

```
Pattern is 5 with 0.886124
Candidate was 5 with 0.000000
Discrimination ratio is Inf

Pattern is 5 with 0.882761
Candidate was 4 with 0.073631
Discrimination ratio is 12.0

Pattern is 5 with 0.877362
Candidate was 2 with 0.038979
Discrimination ratio is 22.5
```

Figure 16: Classification results with PREP1 for geometric symbols rotated by 0, 60, and -60 degrees.



```
Pattern is 3 with 0.888963
Candidate was 5 with 0.068073
Discrimination ratio is 13.1

Pattern is 3 with 0.888750
Candidate was 5 with 0.067956
Discrimination ratio is 13.1

Pattern is 3 with 0.883677
Candidate was 5 with 0.073659
Discrimination ratio is 12.0
```

```
Pattern is 5 with 0.886124
Candidate was 5 with 0.000000
Discrimination ratio is Inf

Pattern is 5 with 0.853305
Candidate was 2 with 0.037264
Discrimination ratio is 22.9

Pattern is 5 with 0.879075
Candidate was 2 with 0.040646
Discrimination ratio is 21.6
```

Figure 17: Classification results with PREP1 for geometric symbols scaled by a factor of 1, 0.8, and 0.6.

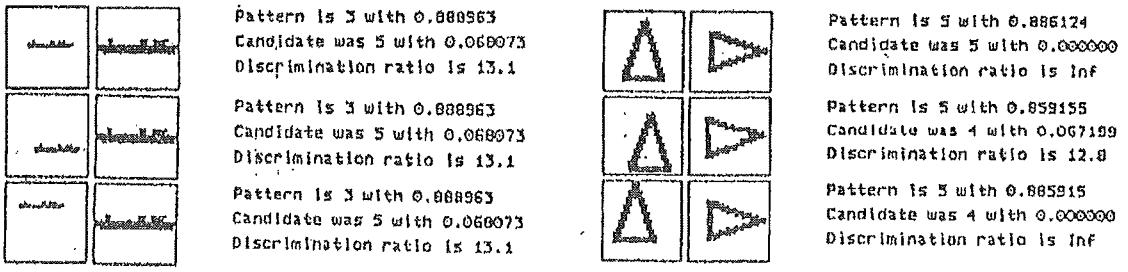
3 Experimental Results

Three problem domains have been experimented: character recognition on the English Alphabet, character recognition on the Japanese Katakana Alphabet and recognition of geometric objects. It should be noted that the R-Block rotates a pattern until the computed orientation coincides with the *x-axis*. Since a pattern and its 180° rotated version will have the same orientation of maximal variance, R-Block will not be able to differentiate between them and will apply the same angle of rotation on both patterns. The resulting mappings will conserve this 180° angle difference. Hence depending on its original orientation, a given pattern will be mapped to one of the two canonical patterns. These two canonical patterns will both represent the class. Hence for each original pattern we have two preprocessor outputs which are used during training.

3.1 Character Recognition on the English Alphabet

This classical problem is the classification of letters in the English alphabet. The number of hidden layers and the number of neurons in each hidden layer has been found by trial-and-error, which is typical for most multilayer feed-forward network applications [1, 2, 3, 6, 8, 10]. From experimentation, a network with 1024 input nodes, 20 neurons in a single hidden layer, and 26 neurons in the output layer performs best for a number of test cases. In the training phase, the network is trained on the canonical example patterns (which are the outputs of the preprocessor) until it manages to successfully classify the letters.

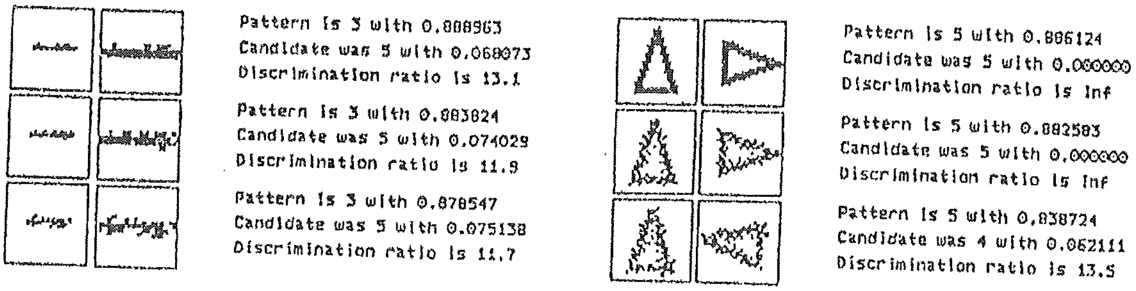
Figure 1 through Figure 10 present examples of the system performance using both preprocessors. Input images images are 32 × 32 pixels. First column is the original image given to the system. Second column is the preprocessed version of the original image, and finally third column is the resulting decision. The class name and value of the corresponding output neuron are given.



```
Pattern is 3 with 0.888963
Candidate was 5 with 0.068073
Discrimination ratio is 13.1

Pattern is 3 with 0.888963
Candidate was 5 with 0.068073
Discrimination ratio is 13.1

Pattern is 3 with 0.888963
Candidate was 5 with 0.068073
Discrimination ratio is 13.1
```

```
Pattern is 5 with 0.886124
Candidate was 5 with 0.000000
Discrimination ratio is Inf

Pattern is 5 with 0.859155
Candidate was 4 with 0.067199
Discrimination ratio is 12.8

Pattern is 5 with 0.885915
Candidate was 4 with 0.000000
Discrimination ratio is Inf
```

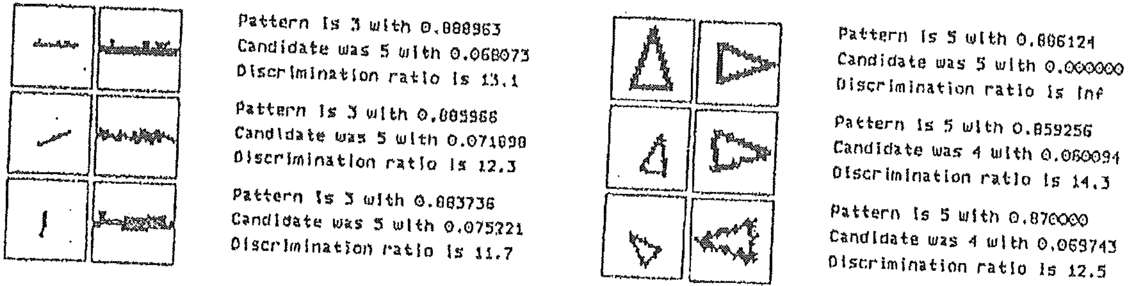
Figure 18: Classification results with PREP1 for geometric symbols translated diagonally by 0, 6, and -6 pixels.



```
Pattern is 3 with 0.888963
Candidate was 5 with 0.068073
Discrimination ratio is 13.1

Pattern is 3 with 0.883824
Candidate was 5 with 0.074029
Discrimination ratio is 11.9

Pattern is 3 with 0.878547
Candidate was 5 with 0.075138
Discrimination ratio is 11.7
```

```
Pattern is 5 with 0.886124
Candidate was 5 with 0.000000
Discrimination ratio is Inf

Pattern is 5 with 0.882583
Candidate was 5 with 0.000000
Discrimination ratio is Inf

Pattern is 5 with 0.838724
Candidate was 4 with 0.062111
Discrimination ratio is 13.5
```

Figure 19: Classification results with PREP1 for geometric symbols with 0%, 20%, and 40% noise.



```
Pattern is 3 with 0.888963
Candidate was 5 with 0.068073
Discrimination ratio is 13.1

Pattern is 3 with 0.885968
Candidate was 5 with 0.071898
Discrimination ratio is 12.3

Pattern is 3 with 0.883736
Candidate was 5 with 0.075221
Discrimination ratio is 11.7
```

```
Pattern is 5 with 0.886124
Candidate was 5 with 0.000000
Discrimination ratio is Inf

Pattern is 5 with 0.859256
Candidate was 4 with 0.060094
Discrimination ratio is 14.3

Pattern is 5 with 0.870000
Candidate was 4 with 0.069743
Discrimination ratio is 12.5
```

Figure 20: Classification results with PREP1 for geometric symbols with random translation, scaling, and rotation applied.

3.2 Character Recognition on the Japanese Katakana Alphabet

This problem is the classification of symbols in the Japanese Katakana alphabet. Since the 111 Katakana symbols are in fact combined forms of 66 unique patterns, the system is trained only on these 66 patterns. With some experimentation, a network with only one hidden layer having twenty nodes has been chosen. Figures 11 through 15 give the performance of the system with preprocessor PREP1.

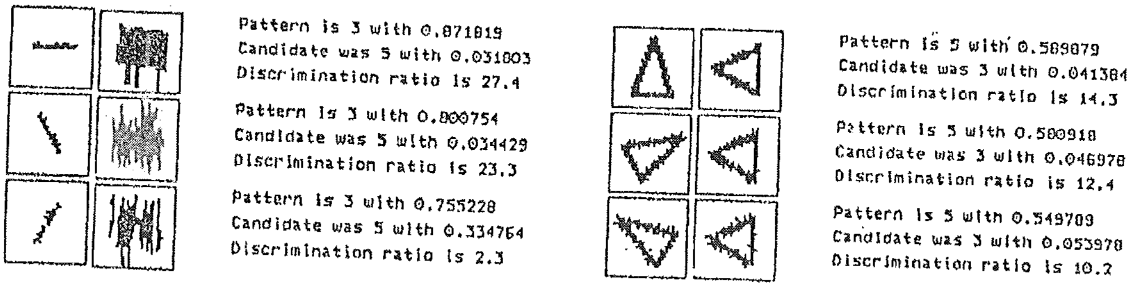
3.3 Classification of Geometric Symbols

This is the problem of classification of five main geometric symbols: circle, cross, line, rectangle, and triangle. Figure 16 through Figure 25 present examples of the system performance using both preprocessors.

Figure 26 gives the classification results for the two versions of the preprocessor on the distorted versions of the geometric symbols. PREP1 could detect only 50% of the distorted patterns, while PREP2 managed to successfully classify 90%. The performance difference emerges from the axial scaling correction ability of this preprocessor.

Table 1 shows the (average) percentage of English letters, Katakana symbols and geometric symbols correctly classified after undergoing 100 random transformations of the type stated in the first column.[2]

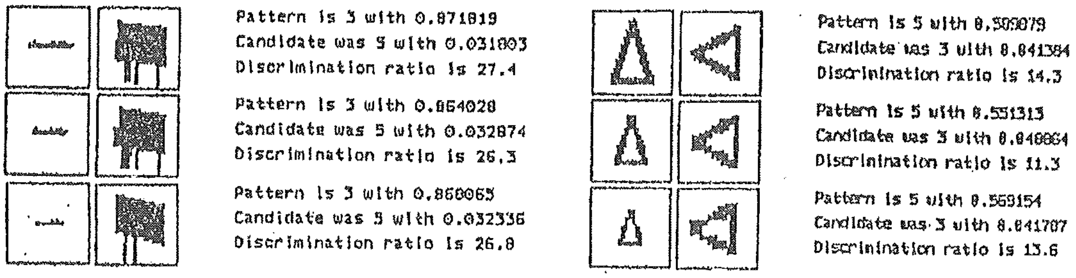
[2] Combined denotes an input that is distorted from the original by a random rotation, scaling and translation. The noise is applied to an undeformed pattern by flipping the on pixels with a certain probability. In the last row, noise is applied to the distorted pattern



```
Pattern is 3 with 0.871819
Candidate was 5 with 0.031803
Discrimination ratio is 27.4

Pattern is 3 with 0.800754
Candidate was 5 with 0.034429
Discrimination ratio is 23.3

Pattern is 3 with 0.755228
Candidate was 5 with 0.334764
Discrimination ratio is 2.3
```

```
Pattern is 5 with 0.589879
Candidate was 3 with 0.041384
Discrimination ratio is 14.3

Pattern is 5 with 0.580918
Candidate was 3 with 0.046978
Discrimination ratio is 12.4

Pattern is 5 with 0.549709
Candidate was 3 with 0.053978
Discrimination ratio is 10.2
```

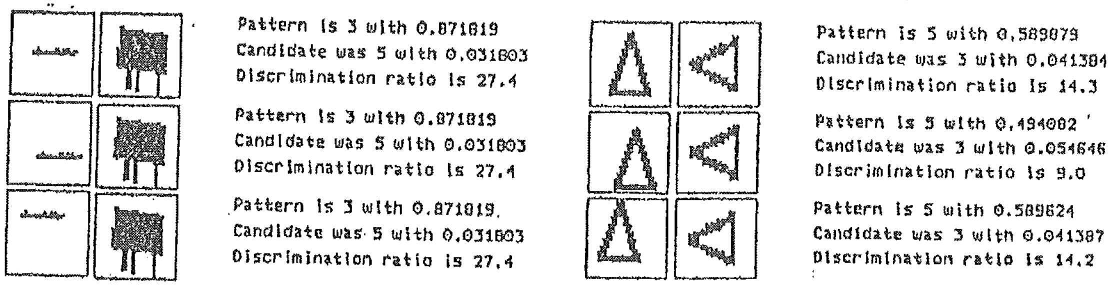
Figure 21: Classification results with PREP2 for rotated geometric symbols.



```
Pattern is 3 with 0.871819
Candidate was 5 with 0.031803
Discrimination ratio is 27.4

Pattern is 3 with 0.864028
Candidate was 5 with 0.032874
Discrimination ratio is 26.3

Pattern is 3 with 0.860065
Candidate was 5 with 0.032336
Discrimination ratio is 26.8
```

```
Pattern is 5 with 0.589079
Candidate was 3 with 0.041384
Discrimination ratio is 14.3

Pattern is 5 with 0.551313
Candidate was 3 with 0.040864
Discrimination ratio is 11.3

Pattern is 5 with 0.559154
Candidate was 3 with 0.041707
Discrimination ratio is 13.6
```

Figure 22: Classification results with PREP2 for scaled geometric symbols.



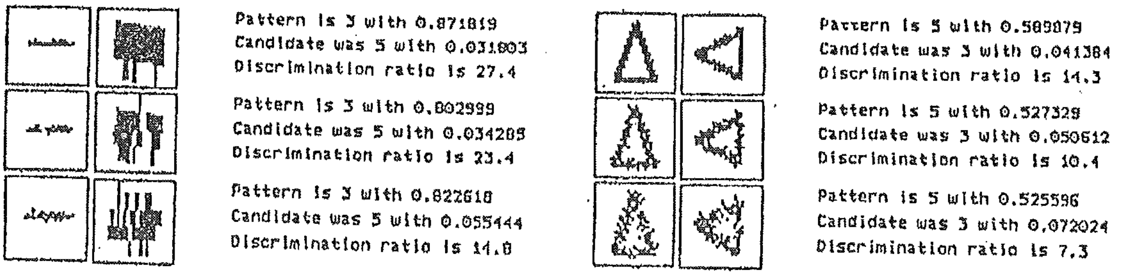
```
Pattern is 3 with 0.871819
Candidate was 5 with 0.031803
Discrimination ratio is 27.4

Pattern is 3 with 0.871819
Candidate was 5 with 0.031803
Discrimination ratio is 27.4

Pattern is 3 with 0.871819
Candidate was 5 with 0.031803
Discrimination ratio is 27.4
```

```
Pattern is 5 with 0.589079
Candidate was 3 with 0.041384
Discrimination ratio is 14.3

Pattern is 5 with 0.494082
Candidate was 3 with 0.054646
Discrimination ratio is 9.0

Pattern is 5 with 0.589624
Candidate was 3 with 0.041387
Discrimination ratio is 14.2
```

Figure 23: Classification results with PREP2 for translated geometric symbols.

4 Conclusions

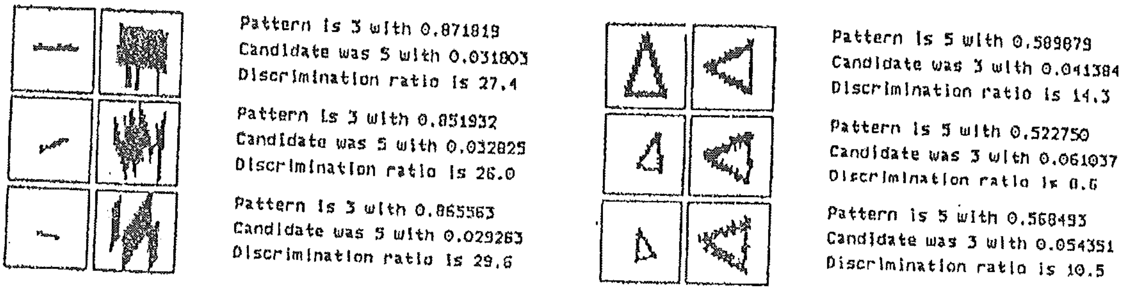
In this work we have presented a hybrid a pattern classification system which can classify patterns independent of any deformations of translation, scaling and rotation. The system uses a preprocessor which maps input patterns to a set of canonical patterns which is then classified by a multilayer neural network. The artificial neural network is trained using the popular backpropagation network. The use of the preprocessor reduces the number of training patterns to two per example pattern to be classified instead of a much larger number if the networks were to be trained on all possible distortions of the patterns. Results from three different applications were presented. In classification of letters of the English Alphabet the system was able to correctly classify 89% of the inputs which were deformed by random rotations, translations and scaling. The performance is much better when distortions were only of one kind, with 100% of the inputs distorted by only translations correctly classified. In the recognition of geometric figures, the system was able to correctly classify 88% of the inputs which were deformed by all three kinds of distortions while the performance was almost perfect when the random distortions were only of one kind. The overall performance for recognition of Japanese Katakana alphabet were worse compared to the other two applications. 68% of the inputs with combined deformations were correctly classified. Even though the performance for inputs which are distorted only by translations or scaling is very good, the performance for rotationally distorted inputs was about 75%. The main reason for this loss of performance is that some of the letters in this case are very similar to each other differing by very small visual features. When patterns that are already deformed are processed with the preprocessors, a certain amount of superfluous visual features may be introduced during mapping between images. The system presented here is independent of the application domain and leaves feature extraction to the neural network and thus can be applied in other domains with relative simplicity.



```
Pattern is 3 with 0.871819
Candidate was 5 with 0.031803
Discrimination ratio is 27.4

Pattern is 3 with 0.802999
Candidate was 5 with 0.034285
Discrimination ratio is 23.4

Pattern is 3 with 0.822618
Candidate was 5 with 0.055444
Discrimination ratio is 14.8
```

```
Pattern is 5 with 0.589079
Candidate was 3 with 0.041384
Discrimination ratio is 14.3

Pattern is 5 with 0.527329
Candidate was 3 with 0.050612
Discrimination ratio is 10.4

Pattern is 5 with 0.525596
Candidate was 3 with 0.072024
Discrimination ratio is 7.3
```

Figure 24: Classification results with PREP2 for noisy geometric symbols.



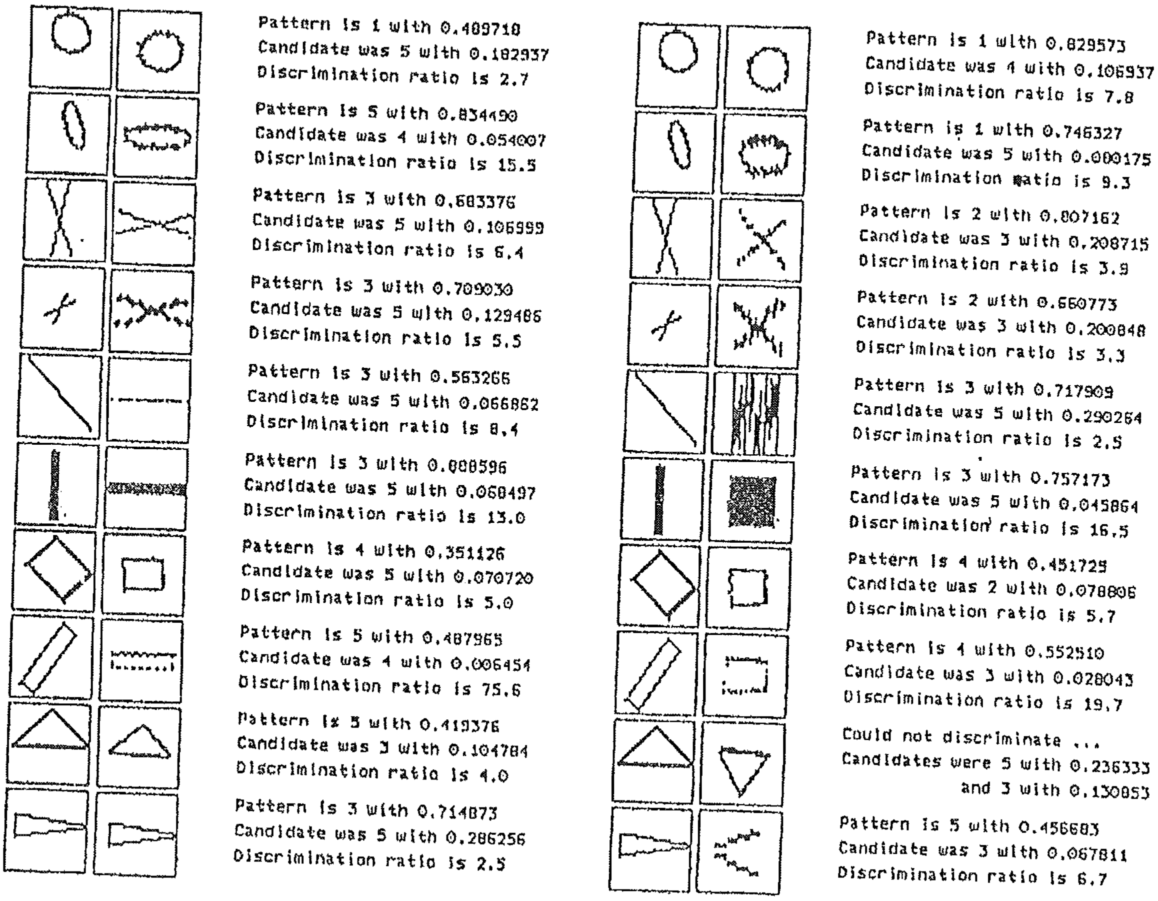
```
Pattern is 3 with 0.871819
Candidate was 5 with 0.031803
Discrimination ratio is 27.4

Pattern is 3 with 0.851932
Candidate was 5 with 0.032825
Discrimination ratio is 26.0

Pattern is 3 with 0.865563
Candidate was 5 with 0.029263
Discrimination ratio is 29.6
```

```
Pattern is 5 with 0.589879
Candidate was 3 with 0.041384
Discrimination ratio is 14.3

Pattern is 5 with 0.522750
Candidate was 3 with 0.061037
Discrimination ratio is 8.6

Pattern is 5 with 0.568493
Candidate was 3 with 0.054351
Discrimination ratio is 10.5
```

Figure 25: Classification results with PREP2 for geometric symbols with random translation, scaling, and rotation applied.

	English		Katakana	Geometric	
Transformation	PREP1	PREP2	PREP1	PREP1	PREP2
Rotation	91%	89%	75%	98%	94%
Scaling	98%	94%	92%	100%	100%
Translation	100%	100%	100%	100%	100%
Combined	89%	79%	68%	88%	84%
20% noise	98%	96%	93%	100%	100%
40% noise	92%	84%	76%	98%	97%
Combined & 20% noise	77%	60%	57%	89%	78%

Table 1: Percentage of correct classification for English letters, Katakana symbols and geometric symbols under various distortions



```
Pattern is 1 with 0.489718
Candidate was 5 with 0.182937
Discrimination ratio is 2.7

Pattern is 5 with 0.834490
Candidate was 4 with 0.054007
Discrimination ratio is 15.5

Pattern is 3 with 0.683376
Candidate was 5 with 0.106999
Discrimination ratio is 6.4

Pattern is 3 with 0.709030
Candidate was 5 with 0.129486
Discrimination ratio is 5.5

Pattern is 3 with 0.563266
Candidate was 5 with 0.066862
Discrimination ratio is 8.4

Pattern is 3 with 0.888596
Candidate was 5 with 0.068497
Discrimination ratio is 13.0

Pattern is 4 with 0.351126
Candidate was 5 with 0.070720
Discrimination ratio is 5.0

Pattern is 5 with 0.487965
Candidate was 4 with 0.006454
Discrimination ratio is 75.6

Pattern is 5 with 0.419376
Candidate was 3 with 0.104784
Discrimination ratio is 4.0

Pattern is 3 with 0.714873
Candidate was 5 with 0.286256
Discrimination ratio is 2.5
```

```
Pattern is 1 with 0.829573
Candidate was 4 with 0.106937
Discrimination ratio is 7.8

Pattern is 1 with 0.746327
Candidate was 5 with 0.080175
Discrimination ratio is 9.3

Pattern is 2 with 0.807162
Candidate was 3 with 0.208715
Discrimination ratio is 3.9

Pattern is 2 with 0.660773
Candidate was 3 with 0.200848
Discrimination ratio is 3.3

Pattern is 3 with 0.717909
Candidate was 5 with 0.290264
Discrimination ratio is 2.5

Pattern is 3 with 0.757173
Candidate was 5 with 0.045864
Discrimination ratio is 16.5

Pattern is 4 with 0.451729
Candidate was 2 with 0.078806
Discrimination ratio is 5.7

Pattern is 4 with 0.552510
Candidate was 3 with 0.028043
Discrimination ratio is 19.7

Could not discriminate ...
Candidates were 5 with 0.236333
            and 3 with 0.130853

Pattern is 5 with 0.456683
Candidate was 3 with 0.067811
Discrimination ratio is 6.7
```

Figure 26: Classification results, with PREP1 (left) and PREP2 (right), on distorted patterns of the five main geometric symbols.

References

[1] E. Barnard and D. Casasent, "Shift invariance and the neocognitron," *Neural Networks*, vol. 3, pp. 403–410, 1990.

[2] Y. Le Cun et al., "Handwritten digit recognition: Applications of neural network chips and automatic learning," *IEEE Communications Magazine*, pp. 41–46, November 1989.

[3] Y. Le Cun et al., "Handwritten zip code recognition with multilayer networks," in *Proceedings of 10th International Conference on Pattern Recognition, Atlantic City, NJ USA*, pp. 35–40. IEEE Computer Society Press, Los Alamitos, 1990.

[4] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* John Wiley and Sons, NY, 1973.

[5] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* Academic Press, NY, 1972.

[6] K. Fukushima, S. Miyake, and T. Ito, "Neocognitron: A neural network model for a mechanism of visual pattern recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, pp. 826–834, 1983.

[7] R. C. Gonzales and P. Wintz, *Digital Image Processing*, pp. 122–130. Addison-Wesley, MA, 1987.

[8] A. Khotanzad and J. Lu, "Classification of invariant image representations using a neural network," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, number 6, pp. 1028–1038, June 1990.

[9] M. Kirby and L. Sirovich, "Application of the karhunen-loéve procedures for the characterization of human faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, number 1, pp. 103–108, January 1990.

[10] H. A. Malki and A. Moghaddamjoo, "Using the karhunen-loéve transformation in the back-propagation training algorithm," *IEEE Transactions on Neural Networks*, vol. 2, number 1, pp. 162–165, January 1991.

[11] C. Yüceer and K. Oflazer, "A rotation, scaling, and translation invariant pattern classification system," in Mehmet Baray and Bülent Özgüç, eds., *Proceedings of ISCIS VI, The Sixth International Symposium on Computer and Information Science, Side, Türkiye*, volume 2, pp. 859–869, October 1991.

Application of Artificial Neural Networks to Pattern Recognition

Işıl Celasun, M. Burçin Gökgöz, M. Kemal Cılız and Yorgo Istefanopulos
Department of Electrical Engineering, Boğaziçi University, Bebek 80815, Türkiye.

Abstract

In this paper, the recognition problem of fighter / bomber planes is investigated by using artificial neural networks as pattern recognition tools. The aim was to remove noise effects and to represent images in such forms that they are quite smaller in size and so that compatible with the neural network characteristics. There are lots of redundant information in images which should be discarded to increase the performance of neural networks. Five different models of aircrafts are used to train the artificial neural networks.

1. INTRODUCTION

Reference images are obtained by using a Vidicon camera that is connected to a PC through a digitizer circuit.The images are taken under bad (nonuniform) lighting conditions to simulate real cases. Grabbed images consist of 256*256 pixels where each has a value of intensity in the range [0,63]. Then 108*108 subimages that contain the picture of airplanes are extracted. Binary images are obtained by thresholding these gray-level images.

An aircraft which was initially represented by its digitized image is embedded with Gaussian noise having different signal to noise ratio values (6, 12, 18 and 24) and with non-Gaussian noise(Uniform noise in the interval [-9,+9]) Fig.2. Two noise removal techniques which are smoothing and cellular neural networks (CNNs) are applied to remove the noise in the above two cases and then they are compared in terms of performance. Images of airplanes with two different orientations and distances are taken in order to test whether the recognition method is rotational and scale invariant respectively.

Hence, artificial neural network models are used as a noise removal tool and as a classifier. These issues are discussed in sections 2 and 3.

2. NOISE REMOVING METHODS

Cellular neural networks are used as a noise removal technique in [4]. The main difference of CNNs is their nearest neighborhood connection structure as compared to a fully connected Hopfield net. Neurons are only connected to their nearest neighbors in the network matrix. Adjacent cells can interact directly with each other and cells not directly connected may also affect each other indirectly because of the propagation effects of the continuous time dynamics of the network. CNNs can be programmed to perform several tasks through their neighborhood connection weights which are also called cloning templates. A cloning template is a space invariant matrix that shows the connection weights of the cells in a neighborhood. Connection weights are twofold. A matrix A that holds the weights for output voltages of neighboor cells, which is also called the feedback operator and a matrix B that holds the weights for input voltages of neighbor cells, the control operator. The averaging operator which is used as the feedback operator, is chosen as the dynamic rule for noise removing CNN. The steady state of a cell C(i,j) depends on the average of those of its neighbor cells. This feedback operator has been used both for Gaussian and non-Gaussian noise embedded cases of images. Cellular neural network gives better performance for all the aircrafts in all positions when embedded in Gaussian noise with different variance values but approximately similar results are obtained with using smoothing and CNN when images are embedded in Uniform noise. To improve the performance obtained, some changes can be done on the feedback operator for non-Gaussian noise embedded images.

Table 1.
Table presenting the average number of wrong pixels for a 108*108 image after noise removing process using the two methods discussed above ("Smth." denotes smoothing).

	G6		G12		G18		G24	
	Smth.	CNN	Smth.	CNN	Smth.	CNN	Smth.	CNN
F15	153	118	154	121	157	124	162	130
F117	119	91	134	128	134	134	150	150
MIG27	302	224	303	226	306	226	306	226
MIRAGE	79	36	85	39	86	42	88	43
F16	178	129	181	139	189	145	194	153

3. CLASSIFICATION

Five types of airplanes are employed for classification which are F15, F16, MIG27, F117, MIRAGE. To test the recognition performance for the methods that will be discussed in the sequel, images of airplanes with two different orientations and distances are taken into account. The planes are rotated about 45 degree to change the orientation and taken two times far away to scale it. (Fig. 1(b) and (c)) Recognition process is performed for three forms of five airplanes (normal, rotated and scaled) with two type of noise(Uniform and Gaussian with four different variance values).

Recognition process is based on using pecstrum and multilayer perceptron (MLP). The first aim of using pecstrum is to provide translation, rotation and scale invariance properties and the second is to provide small input vectors to MLP. Pecstrum is calculated by applying morphological operations to the two-valued images. This approach is generally based on the analysis of two-valued images in terms of some predetermined geometric shape known as a structing element.

There exists a well developed morphological algebra which is expressible as digital algorithms in terms of a few primitive morphological operations. Two of the fundamental operations are called Minkowski addition and Minkowski subtraction. Minkowski addition adds two regions on the analytical plane by adding all points of one region to those of the second. Minskowski subtraction finds points by first taking the symmetric region of the second operand about the origin, and then selecting those points if their addition with the symmetric region is on the first region. Erosion operation is defined as the subtraction of the symmetry of a region from another. If the second operand is already symmetrical around the origin, the erosion is equivalent to the Minskowski subtraction. Dilation operation is equivalent to the Minskowski addition. While erosion has an effect of shrinking the shapes, dilation expands them. Besides the fundamental operations, two other operations play a central role in image analysis. One of them is opening which is an erosion followed by a dilation. The other is closing which is a dilation followed by an erosion.

Pecstrum consists of a set of vectors where each gives the fractional change in the area of a shape when opening or closing is applied. The image is taken as the first operand of the opening operation. The second operand is a symmetrical region around the origin which is called the structing element. Pecstrum vector set is obtained by applying several opening operations on the original image while increasing the size of the structing element by a certain amount at each step. Two examples of opening sequences are illustrated in Fig.4 and Fig.5. Pecstrum is translational invariant since it is not related to the position but to the area and the form of the shapes. It is also rotational invariant provided the structing element is a circular one[4]. Pecstrum with above properties is not scale invariant. However, if every shape is prescaled to a standard area, pecstrum sets for the images of the same figure with different distances will be close to each other.

The pecstrum consists of at most 6 real values in our implementation (For 5 reference images; F15 : 5 , F117, :6, MIG27 : 3, MIRAGE : 5, F16 : 5). Binary form of pecstrum set is also presented to MLP network as well as pecstrum itself. In this case binary codes are concatenated to form an input vector to MLP. The length of such input vector is 36 instead of 6, since each element of pecstrum is converted into binary code up to 6 bit precision. In our implementation, these two options give the same result.

The MLP net consists of few layers where each contains a number of perceptrons.

Each perceptron in a layer has a connection with all perceptrons of the previous and the next layers. The network has an input layer of which perceptrons accept the input values to the network. The last layer of the network outputs the result of the operations. Each perceptron has a nonlinearity that maps the sum of incoming values to a single output value. Connections between perceptrons have weights which are multiplied by the output of the outcoming perceptron to calculate the input of the incoming perceptron. Weights are adjusted by the backpropagation algorithm while the network is trained.

A neural net consisting of 3 layers is used. The input layer has 36 nodes. Two hidden layers contain 45 and 30 nodes. The output layer has 5 nodes where each one is expected to output "1" while the others "0" for each airplane.

This method which is based on artificial neural network is compared with a conventional one which makes use of Fourier descriptors(FD) in shape representation. FDs in shape representations are also translational, rotational ans scale invariant. In this representaion a closed contour surrounding the shape is sampled by a certain number N of points. The axis and ordinate of each point constitute the real and imaginary part of a complex number respectively. Then a set of N complex Fourier coefficients are obtained and calculated by :

$$f_k = \frac{1}{N}\sum_{n=1}^{N} z_n \exp(-j2\pi kn/N)$$

where $k = -n_1, \ldots, n_2$ with $n_1+n_2+1 = N$.

These values are taken as $N=128$, $n_1=63$ and $n_2=64$ in the implementation. Comparison between two FDs F and G can be achieved by crosscorrelating them by the formula:

$$c = \max_{n} \left| \frac{1}{N} \sum_{k=-n_1}^{n_2} \frac{f_k * g_k}{\|f_k\| \, \|g_k\|} \exp(-j2\pi kn/N) \right|$$

or

$$c' = \max_{n} \left| \frac{1}{N} \sum_{k=-n_1}^{n_2} \frac{f_k * g'_k}{\|f_k\| \, \|g_k\|} \exp(-j2\pi kn/N) \right|$$

where g_k' is equal to $g_{n_2-n_1-k}$ then a distance term between them is d(F,G)=min(d , d') where $d^2 = 2[1-c]$ $d'^2 = 2[1-c']$ with d' being the distance term for the flipped form of the original shape. First five FDs for five different types of planes used in the application are found and stored then at the classification stage , a Fourier descriptor computed from the new image is compared by each of those five FDs. The minimum distance is taken for the classification scheme.

4.SIMULATION RESULTS

Simulations are done on a 286/16 PC with coprocessor and Turbo Pascal 6.0 is used as the programming language. Once the FDs of reference images are computed the recognition process of new coming noise removed image is completed in exactly 60 seconds. On the other side once the network is trained, the computation of pecstrum and recognition by the network takes 25 seconds for an image in average. The result of two classification methods are presented in Table 2. Fourier descriptor method best recognizes F15 but for example F16 is generally misrecognized as being confused with F15 in recognition. Pecstrum plus MLP recognizes F15 and F16, MIG27 almost in all cases. This method has the best performance at recognizing the similarly oriented planes with the reference planes and close performances in rotated orientation and the scaled cases

Table 2.

Comparison of the two methods in recognition of the airplanes ("Pec." denotes pecstrum and ■ , + denote the correct recognition for FD and Pec.+MLP respectively)

	Noise	F15		F117		MIG27		MIRAGE		F16	
		FD	**Pec.+ MLP**	**FD**	**Pec.+ MLP**	**FD**	**Pec.+ MLP**	**FD**	**Pec.+ MLP**	**FD**	**Pec.+ MLP**
Normal	G6	■	+	■	+	■	+	■	+		+
	G12	■	+	■	+	■	+	■			+
	G18	■	+	■		■	+	■	+		+
	G24	■	+	■		■	+	■	+		+
	U18		+				+		+		+
Rotated	G6	■	+		+		+	■			+
	G12	■	+				+	■			+
	G18	■	+				+	■			+
	G24	■	+					■			+
	U18		+				+		+		+
Scaled Smaller	G6		+		+		+				
	G12	■				■	+				+
	G18	■					+				+
	G24		+			■	+			■	+
	U18	■				■	+				+

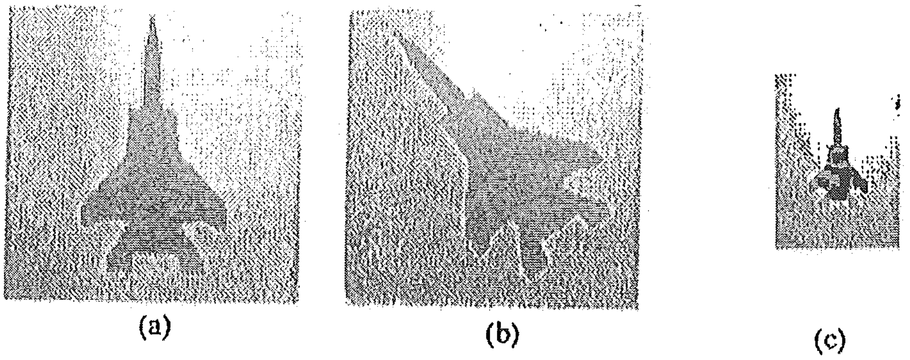


Figure 1. Gray level images of F15 (a) 108*108 reference image , (b) 45 degree rotated, (c) two times distant pose .

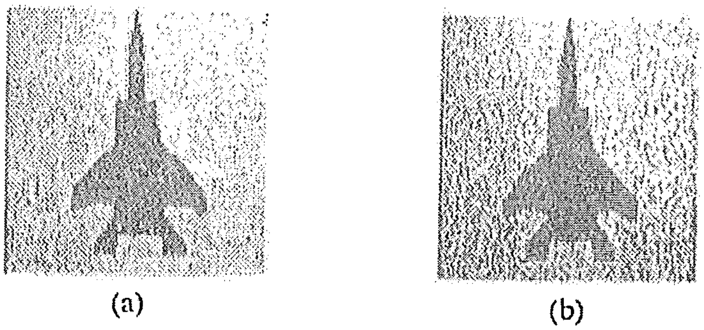


Figure 2. Noisy images of F15 (a) Gaussian noise with variance 18 is added (b) Uniform noise is added.

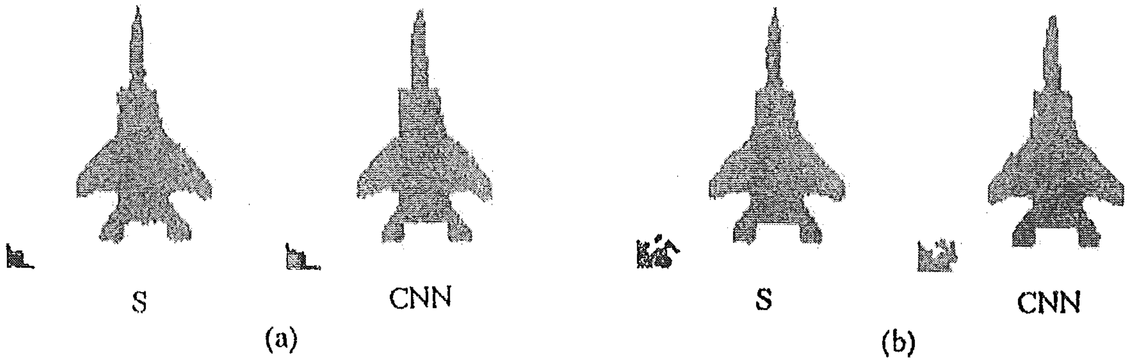


Figure 3. Binary forms of noise removed version of F15 by smoothing (denoted by S) and using CNN of (a) Gaussian with variance 18 (denoted by G18) (b) Uniform noise (denoted by U) added image of F15

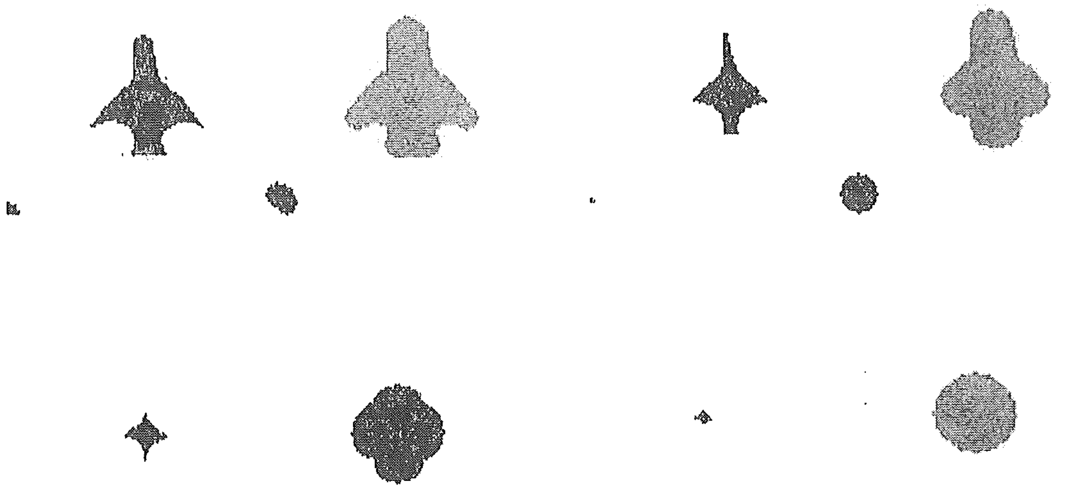


Figure 4. Opening sequence for F15.

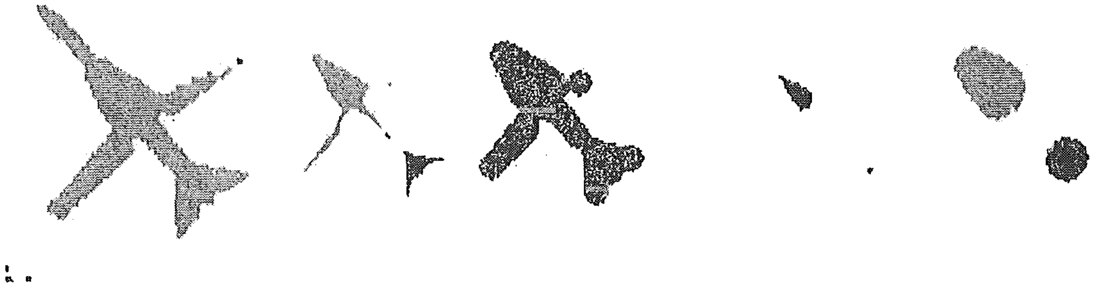


Figure 5. Opening sequence for MIG27.

5. CONCLUSIONS

With using the morphological properties such as thickness ratio, relative area of some components of the shape, different pecstrums are available for all airplanes. So, in the recognition of objects with silhouettes like airplanes pecstrum provides good results. On the other hand, Fourier descriptor method uses the contour characteristics (frequency) of the shape. Since the contours obtained from the shapes of some planes are not much different from each other, 'FD' is not as good as 'pecstrum + MLP' in recognition performance for all cases discussed previously .

6. REFERENCES:

1. R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, John Wiley & Sons, 1973.

2. Y. Istefanopulos, B. Sankur, I. Celasun, O. Toker, "Variational approach to pattern recognition using third order moments", International Signal Processing Workshop on Higher Order Statistics, France, July 1991.

3. Anil K. Jain, Fundamentals of Digital Image Processing, Prentice-Hall, New Jersey, 1989.

4. L. Chua, Lin Yang, "Cellular neural networks: Applications" IEEE Trans. CAS, Vol. 35, No.10, pp.1273-1290

5. V. Anastassopoulos, A.N. Venetsanopoulos, " The classification properties of the pecstrum and its use for pattern identification", Circuits Systems Signal Processing, Vol. 10, No.3, pp. 293-326, 1991.

6. I. Pitas, A. Venetsanopoulos, " Morphological Shape Description", IEEE Trans. PAMI, Vol. 12, No. 1, pp.38-45, Jan. 1990.

7. T. Roska, et al., "Detecting simple motion using CNN's" Proc.IEEE First Int. Workshop of CNNA, pp. 127-138, Budapest, 1990.

Segmentation of Ottoman Characters

Serdar Bıyıksız

Department of Artificial Intelligence, TÜBİTAK - Marmara
Research Center, 41470 Gebze - Kocaeli Türkiye

Abstract

This paper provides one solution to the complex problem of segmenting Ottoman words and subwords into their individual character components. Such a procedure is a critical component of being able to recognize and translate Ottoman written text automatical via computer into modern Latin character based Turkish.

1. INTRODUCTION

Much Ottoman printed text exists which has painstakenly been translated into modern Turkish by humans able to understand and decipher this type of writing. The objective of the current research is to provide for a computer based system based on scanned text which may automatically and systematically translate Ottoman into modern Turkish. What makes part of this work difficult is the nature of the Ottoman writing itself in that it is cursive, sometimes connected, overlapping, and includes marks above or below words and characters. Furthermore, the shape of characters varies based on their position within words or alone.

The important problem of character segmentation is encapsulated in the whole recognition system shown in figure 1.

```
Text---->SCANNER---Binary image file--->PREPROCESSING---

--->CHARACTER    ---Segmentation--->CHARACTER      -------
    SEGMENTATION    vectors          POSTPROCESSING

--->Character       --->CHARACTER       ---->Latin character--
    Identification      CLASSIFICATION       counterparts
    Vectors

--->TRANSLATION SYSTEM---->Translated text in Latin based
                            character set
```

Figure 1. Identification system

As can be realized in figure 1, without proper character segmentation, the remainder of what follows will be a multiplicative error. Thus the better segmentation is achieved, the better the identification.

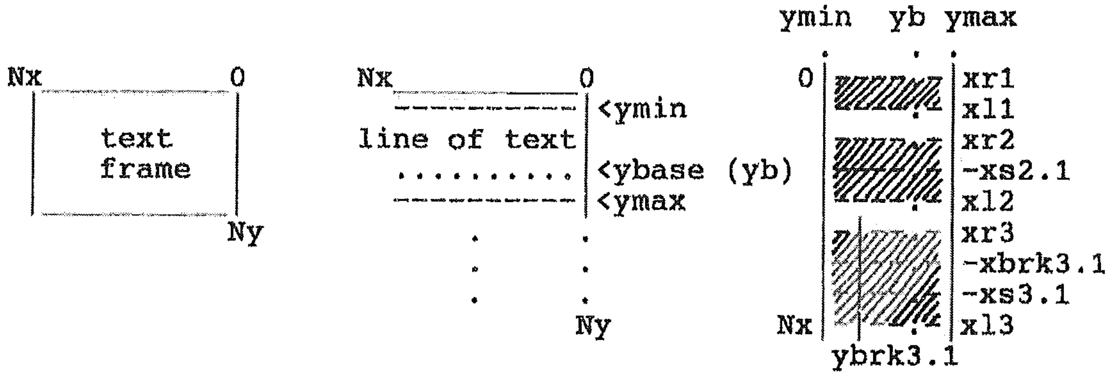
2. METHODOLOGY

Much investigation has been made by the researcher into the most current literature concerning Arabic text recognition. Little if none of the literature directly deals with Ottoman text; however Arabic text has very similar characteristics and problems. From this work the researcher has come to the conclusion that in order to form any sort of systematic segmentation, one must strive for some type of commonality in the text as a starting point and then process each line of text individually with systematic vertical and horizontal scanning. This scanning must incorporate some particulars of the Ottoman writing by becoming somewhat familiar with the writing.

The intent of this paper is to present the basic details of a different approach for Ottoman character segmentation. The exact details of the alogrithm and program are not included due to preference of nondisclosure to the general public at this time.

3. NOTATIONAL FIGURES

Figure 2 below establishes the notational references to be utlized in the discriptive alogrithm to be given in the next section. Reference should be made to these figures as needed.



(a) (b) (c)

Figure 2. Notational references

(r=right, l=left, s=segment, brk=break)
xr,xl - 1st pass, xbrk,ybrk - 2nd pass, xs - 3rd pass

4. SEGMENTATION ALGORITHM DESCRIPTION

Given below, in several parts, is a word description of the segmentation algorithm. Refer to figure 2 as required.

< PREPROCESSING >

Establish ymin, ymax, ybase, for line of text

(1) Scan with a horizontal (one pixel wide) scanline from (x,y)=(0,0) towards (0,Ny) and mark they-coordinate where an initial line of black pixel(s) is (are) detected as ymin; mark the y-coördinate where white pixel lines (white pixels along entire horizontal scanline) white count (WC) threshold Twc as ymax (note: WC reinitializes to 0 if at anytime before Twc is reached, a line of black pixel(s) is (are) detected).

(2) ybase is established for the horizontal line detecting the greatest count of black pixels (BC)along the scanline as the scan proceeds from (0,ymin) in the increasing y direction; in situations where ties occur, the most current line is the baseline.

Filter and thin line of text

(1) Median filter line of text

(2) Thin text image line between [0,ymin] and [0,ymax] from (0,0) to (0,Nx)

< SCANNING >

Scan right to left to determine major character/word segmentation zones (1st pass)

(1) Scan from right to left (x=0 to x=Nx)' using a vertical (one pixel wide) scanline bounded by [ymin,ymax].

(2) Mark the x-coordinate of black pixel(s) detected anywhere along the vertical scanline as xr, mark the x-coordinate of the first white pixel line (white pixels along entire vertical scanline) detected after the detection of black pixel(s) as xl. That is, xr and xl correspond to detecting an alternating pattern of black pixel(s) and white

```
pixel(s), (|....|//////|.....|////////////|....|....| ).
                 xl2     xr2   xl1          xr1
```

(3) Continue (2) until no other alternating patterns are detected.

Scan to determine hidden caharacter separation (2nd pass)

(4) For the x-coordinate positions bounded by [xr,xl], begin scan from y=ymin towards ymax and mark the y-coordinate of the first black pixel(s) detected along the horizontal scanline occurring after a pattern consisting of line(s) of black pixel(s) then line(s) of white pixels detected as ybrk (indicates a break in character along the vertical).

(5) If (4) occurred, extract (by making all white (0) pixels) the section bounded by [xr,xl;ymin,ybrk] and then scan with a vertical scanline from xr toward xl and mark the x-coordinate of black pixel(s) detected anywhere along a vertical scanline after a pattern consisting of black line(s) followed by line(s) of white pixels as xbreak (indicates a break in character(s)).

(6) Repeat (4) and (5) until no others are found; (4) and (5) implement a pruning operation to detect hidden character separations where hidden characters are caused by overhanging (above or below characters/words) vertically disconnected line(s) or dot(s).

Scan to determine character separations not caused by hidden character situations (3rd pass)

(1) Begin scan from xr towards xl;

(1.a) in memory stack SB (baseline stack) push into stack black or white pixels detected along ybase keeping a count of BC and WC pixels;

(1.b) in memory stack SI (pseudoimage stack) push into stack black pixel or white pixel not along ybase; this implies that even if there is more than one black or white pixel detected, not occurring along ybase-coordinate, only one black or white pixel is recorded as being pushed into SI (thus a pseudoimage results);

(1.c)

{i} if comparison of SB and SI (at the same stackpointer (SP) location) indicates SB (white), SI (white) segment at the white x-coordinate as xs (WC(base)=0 remains);

{ii} if comparison of SB and SI indicates SB (white), SI (black) start WC and if WC(base) > Twc segment at WC=0 as xs;

{iii} if comparison of SB and SI indicates SB (black), SI (black) start BC and if BC(base) > Tbc segment at BC=last count as xs;

{iv} if comparison of SB and SI indicates SB (black), SI (white) start BC and if BC(base) > Tbc segment at BC=last count at xs;

Note: 1. WC(base) and BC(base) reinitialized to zero (0) each time a new count begins,

Note: 2. if for SB (white), SI (white), (xs=xr or xl) WC remains at 0 count until a situation other than SB (white), SI (white) is encountered,

Note: 3. ii and iii are hidden zone separations and thus may yield a loss of information of one character and an addition of information to another character; adding information may yield a more difficult classification problem since it results in a more unpredictable situation, than loss of information, due to wider variations to contend with; when part of a character may be lost, some tolerence in character variations may be allowed to yield identification.

Note: 4. iv may result in segmenting a whole character (between xr and xl); if identification is not possible for xr to xs and xs to xl then reclassification for identification may be required for xr to xl as a whole.

5. OVERVIEW OF IDENTIFICATION PROCESS

So as to highlight the major steps involved in the Ottoman character identification scheme and where segmentation fits into all of this, presented below is an overview. Those steps between the dashed lines are steps beyond segmentation.

(1) <Preprocessing> .find ymin,ybase,ymax for a line of text .median filter a line of text .thin a line of text image.

(2) <Scanning for Segmentation>
.1st pass: .locate major character/word segmentation zones .2nd pass: .locate hidden character separations due to vertical breaks caused by line(s) or dot(s) .3rd pass: .locate any other character separations not covered by 1st and 2nd pass conditions.

(3) <Segmentation Coordinate Vectors> Vr1, Vr2, Vr3, ...,Vrn where Vri=[1Vri, 2Vri, ...,kVri]; 1Vri=[xri,qi], qi=xbrk;xsi; 2Vri=[qi,q(i+j)], qi=xbrki,xsi; kVri=[qi,xli].

- -

{1} Each vector indicating an isolated character is unthinned by adding one pixel to each of the four sides of a character skeleton pixel.

{2} Perform edge detection to obtain a character contour.

{3} Find Fourier descriptor (FD) vector of a character contour.

{4} FD vector is utilized as input to a neural network classifier to identify a character in the Latin alphabet (modern Turkish) according to an established rule base.

6. SEGMENTATION/IDENTIFICATION PROGRAM STRUCTURE

Shown below is how a program may be structured so as to achieve final identification encompassing the segmentation phases. It should be emphasized that Contour implies use of an edge detector to leave only a character image edge, Fourier_Desc implies utilizing Fourier descriptors of a character image contour so as to uniquely label characters, and Classifier implies use of a neural network classifier for character identification.

```
<<MAIN>>
[1] CALL SEGMENT [2] CALL CHARACTER [3] CALL CLASSIFIER

[1] <SEGMENT> (1.1) CALL PREPROCESS (1.2) CALL SCAN RET MAIN

[2] <CHARACTER> (2.1) CALL LOCATE_CHAR (2.2) CALL UNTHIN
(2.3) CALL CONTOUR (2.4) CALL FOURIER_DESC RET MAIN

[3] <CLASSIFIER> RET MAIN

[1.1] <PREPROCESS> (1.1.1) CALL LINE (1.1.2) CALL FILTER
(1.1.3) CALL THIN RET SEGMENT

[1.2] <SCAN> CALL FIRST_PASS CALL SECOND_PASS CALL
THIRD_PASS RET SEGMENT

[2.1] <LOCATE_CHAR> RET CHARACTER
[2.2] <UNTHIN> RET CHARACTER
[2.3] <CONTOUR> RET CHARACTER
[2.4] <FOURIER_DESC> RET CHARACTER
```

7. TYPICAL APPLICATION

Here a typical piece of written text is subject to the rules of the aformentioned algorithm to show the workings of the algorithm. Refer to figure 3.

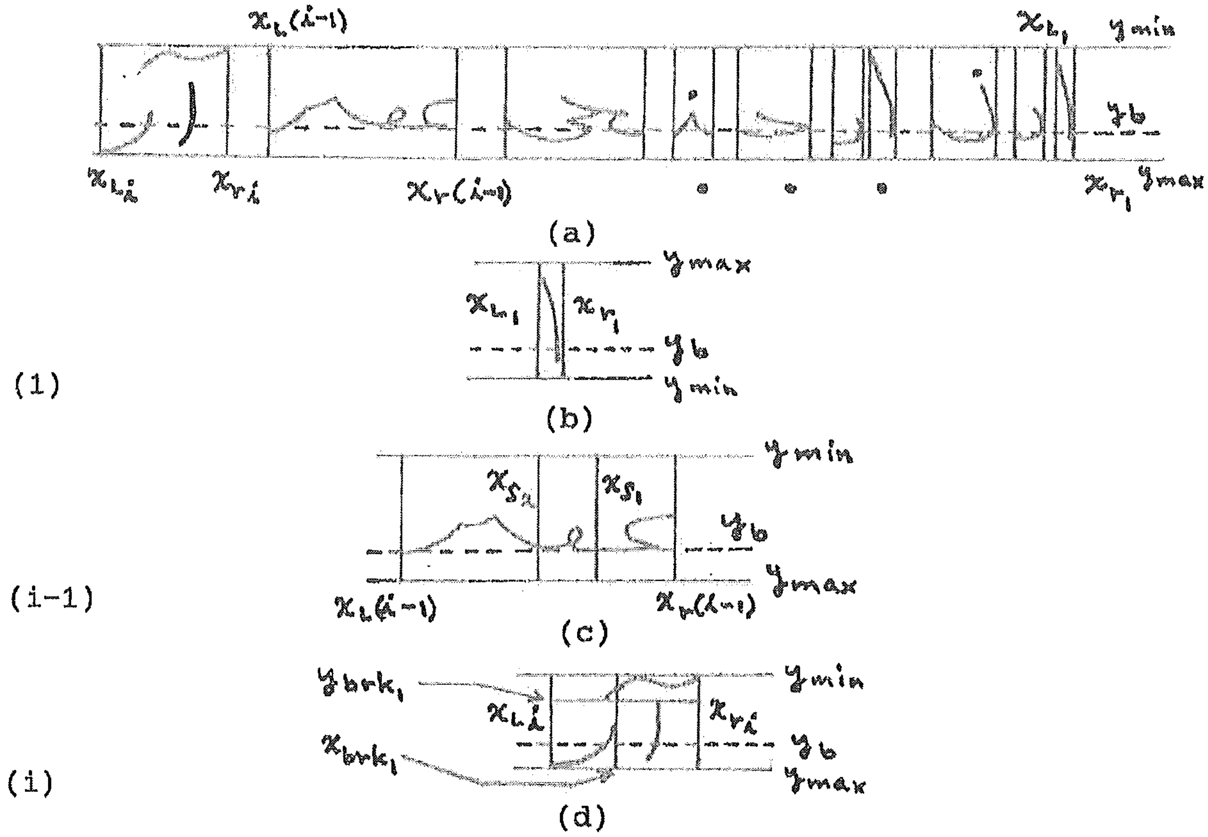
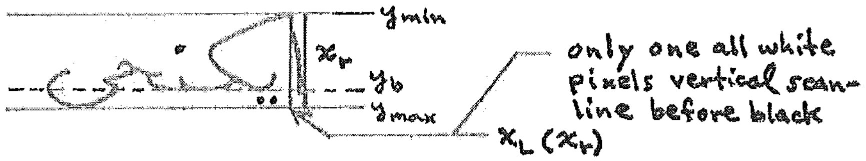


Figure 3. Typical segmentations

The three typical situations are typified above and include some segmentation situations for Ottoman characters. However shown below is a particular case which should be mentioned.



If a case shows only an xl occuring at the same point where an xr is expected, then that segment point receives a double label as xl (xr).

8. CONCLUSION

A segmentation algorithm has been presented for Ottoman characters. The algorithm examplifies a systematic approach to the problem and differs from previous work for Arabic-like text in that the common text baseline is utilized for making some decision for segmentation coordinates. Like so many others, the algorithm is not expected to be foolproof due to the nature of Arabic-like writing and its varations and anomalies. However the approach presented is new and holds promise to yield reasonable results with future refinements expected as special cases warrant.

9. ACKNOWLEDGEMENTS

I wish to thank Dr. Şakir Kocabaş within the Department of Artificial Intelligence. Also many thanks go to all my colleagues in the Department of Space Sciences (Dr. Hakan Çağlar, Dr. Mehmet Emin Özel, Ms. Hülya Yıldırım and all others) for providing me with the optimum environment to write this document.

10. REFERENCES

1. Mahmoud, S.A., Abuttaiba, I. and Green, R.J., Skeletonization of Arabic characters using clustering based skeletonization algorithm (CBSA), Pattern Recognition, 24 (1991) 453-464.

2. Kundu, A., Robust edge detection, Pattern Recognition, 23 (1990) 423-440.

3. Khotanzad, A. and Lu, J-H., Classification of invariant image representations using a neural network, IEEE Transactions on Acoustics, Speech, Signal Processing, 38 (1990) 1028-1038.

4. Al-Emami, S. and Usher, M., On-line recognition of handwritten Arabic characters, IEEE Transactions on Pattern Analysis and Machine Intelligence, 12 (1990) 704-710.

5. Amin, A. and Mari, J.F., Machine recognition and correction of printed Arabic text, IEEE Transactions on Systems, Man, and Cybernetics, 19 (1989) 1300-1306.

6. El-Sheikh, T.S. and Guindi, R.M., Computer recognition of Arabic cursive scripts, Pattern Recognition, 21 (1988) 293-302.

7. Almuallim H. and Yamaguchi S., A method of recognition of Arabic cursive handwriting, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-9 (1987) 715-722.

8. Kundu, A., Mitra, S.K., and Vaidyanathan, P.P., Application of two-dimensional generalized mean filtering for removal of impulse noises from images, IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-32 (1984) 600-609.

9. Naccache, N.J. and Shinghal, R., SPTA: a proposed algorithm thinning binary patterns, IEEE Transactions on Systems, Man, and Cybernetics, SMC-14 (1984) 409-418.

10. Shridhar, M. and Badreldin, A., High accuracy character recognition algorithm using Fourier and topological descriptors, Pattern Recognition, 17 (1984) 515-524.

Görme ve Görüntü İşleme

Vision and Image Processing

Active Vision / Etkin Görme

Ethem Alpaydın

Bilgisayar Mühendisliği Bölümü, Boğaziçi Üniversitesi, TR-80815 Bebek Istanbul
alpaydin@trboun.bitnet

Abstract

Recognition in parallel is costly to implement. Achieving a recognizer with feasible complexity and cost requires a compromise between parallel and sequential processing. Dividing operation in time in vision, seeing a part of the scene at a time, one needs to find ways first, to map spatial relations into temporal relations and second, to compare temporal signals. In the literature the name *selective attention* is employed by which a system selectively concentrates on parts of a given signal, one at a time.

In such a model one wants to be able to learn. Definition of a visual object consists of the content of the parts of the image seen one at a time, i.e., features extracted in each fixation, and the trajectory followed while seeing those so as to be able to take into account their relative positioning. Learning of such definitions should proceed in an incremental manner; one starts from short sequences where rough features are used to define objects. As similar objects are encountered, longer sequences and finer features need be extracted.Two learning algorithms, named Grow-and-Learn (GAL) and Grow-and-Represent (GAR) for respectively supervised and unsupervised learning, have previously been proposed towards this aim (Alpaydın, 1990).

This paper, in the first two sections, explains the idea and gives supporting theories from neuroscience and psychology. How such a recognizer achieves translation invariance is shown. Various compartments of such a system, e.g., pre-attentive, attentive, and associative levels, are analyzed. Difference is stressed between internal and external forms of attention, commonly confused.Saccadic system in vision, as one form of external attention, is explained.

Özet

Tamamen koşut bir tanıma, gereksinim duyulacak sistemin karmaşıklığı yüzünden pahalıdır ve her zaman olası olmayabilir. Gerçekleştirilebilecek bir karmaşıklığa ve fiyata sahip bir tanıyıcı, ancak koşut ve sıralı işlemin beraber ve uygun bir kullanımı ile olasıdır. Görmede, bütün girdinin bir anda işlendiği koşut bir sistem yerine, etkin görmede önerdiğimiz, belirli bir anda sistemin sadece bir bölüm girdiyi alıp, koşut işlemesi, bütün görüntüyü tanıyabilmek için ise, zaman içinde "dikkat"ini görüntünün çeşitli bölümlerine, sıra ile vermesidir. Bunu gerçekleştirebilmek için

[1] uzay içindeki ilişkilerin zaman içindeki ilişkilere çevrilebilmesi,
[2] bir anda değilde zaman içinde parça parça alınacak sinyallerin uygun bir şekilde saklanabilmesi ve karşılaştırılabilmesi

gerekir. Bir sistemin zaman içinde seçerek sinyalin çeşitli bölümlerini parça parça ve sıra ile birer birer işlemesine *seçici dikkat* adını verebiliriz.

Böylesi bir sistemde öğrenmek de istenebilir. Bu durumda bir cismin tanımı

[1] her görünen bölümdeki önemli özellikler ve
[2] birbirlerine göre zaman ve uzay içindeki konumlarından

oluşur. Böylesi tanımlar öğrenme sırasında detaylanabilir, dolayısı ile değişebilir. Örneğin, ilk başta cismin bir kaç kaba özelliği bu cismi diğerlerinden ayırt etmeye yetebilirken, zaman içinde benzer başka cisimlerin öğrenilmesi ile, cismin başka bölümlerine de bakmak ve daha detaylı, ince özelliklerin bulunması gerekebilir. Yapay sinir ağları ile böyle bir öğrenmenin yapılabilmesi için, öğrenme yordamlarının ağ yapısını da değiştirebilmesi gerekir. Böylesi yordamlar yine bu sempozyumda anlatılacaktır.

Psikolojik ve nörofizyolojik teorilerin etkin görme fikrini nasıl desteklediği anlatılmış ve böyle bir sistemin çeşitli parçaları, dikkat öncesi işlem, dikkat, bellek ve tanıma katmanları, tanıtılmıştır. Sık olarak karıştırılan iç ve dış dikkat arasındaki fark anlatılmış ve göz hareketleri, bir örnek olarak verilmiştir. Gerçekleştirilmiş bir benzetimde yer değiştirmeye değişkensizliğin nasıl olduğu ve sistemin nasıl daha genelleştirilebileceği anlatılmıştır.

1. INTRODUCTION

The feature idea implies checking for a particular value combination at a certain part of the image, i.e., the receptive field. The problem with this approach is that when the possible number of features or the image size gets large, one encounters the problem of *combinatorial explosion*, one reaches a hierarchical cone size that is no longer possible to implement. Assigning one unit to check for a certain feature at a certain position, i.e., conjunctive encoding (Hinton et al., 1986), one arrives to a number that is not plausible neither from the point of view of engineering nor neuroscience.

One way to get around this problem is to divide the recognition process into several subprocesses in time. Instead of performing the recognition process all in parallel, one divides it into a *sequence* of partial recognitions, each one being realized during a certain period of time according to a certain order. Thus a compromise between parallel and sequential processing is envisaged. One implements a small cone that is fed by only a small part of the image, which performs basic feature extraction in that restricted region all in parallel.The scope of this cone can be changed and thus other parts of the image can be sampled in later steps of processing. The result of these samplings are combined in time till the system gets enough knowledge to be able to identify the image content.

Whereas spatial data are presented in parallel, temporal data are presented serially. A finite temporal window must be used and events over a longer period of time must be sufficiently abstracted, i.e., by extraction of features, so that they can be efficiently represented and processed internally. Successive fixations should be generated to sample the image and these samples should be processed and abstracted to generate a temporal context in which results can be integrated over time. Temporal expectations, as I will talk about later on, can be generated to control further operation. Thus, it is one of a time-varying data interpretation task (Tsotsos, 1987).The system then has the structure seen in figure 1.

There is an "eye" that looks at an image but which can really "see" only a small part of it. This part of the image that is examined in detail is called the "fovea." The fovea's content is examined by the *pre-attentive level* where basic feature extraction takes place. This level corresponds to the small cone I have mentioned above, where all the processing is done in parallel. Sampling a part of the image is also called a *fixation*, moving eye from one part to another is a *saccade* and is governed by an *oculomotor system*. The output of the pre-attentive level is fed to the *cognitive level* which has two functional components: *attentive* and *associative*. Although processing at the pre-attentive level is bottom-up, i.e., driven by input, processing at the cognitive level is at a higher level relating to long-term plans of the system which requires a *long-term memory*. In this memory, definitions of classes that are to be recognized are stored in an abstract form based on the features detected by the pre-attentive level. The search through memory, given a series of fovea samplings, is performed by the associative part.

The attentive part induces the oculomotor system to move the eye from one part of the image to another. To minimize recognition time, the attentive part should be able to find where are the parts that convey most discriminative information. For this, one should find a criterion of being "interesting," which then should be applied to all points in the visual field and the eye should then be turned to the maximally interesting point. The idea of exchanging spatial relations with temporal ones is mentioned by Pitts and McCulloch (1947) calling it *the exchangeability of space and time*. Feldman and Ballard (1982) also touch the point briefly.

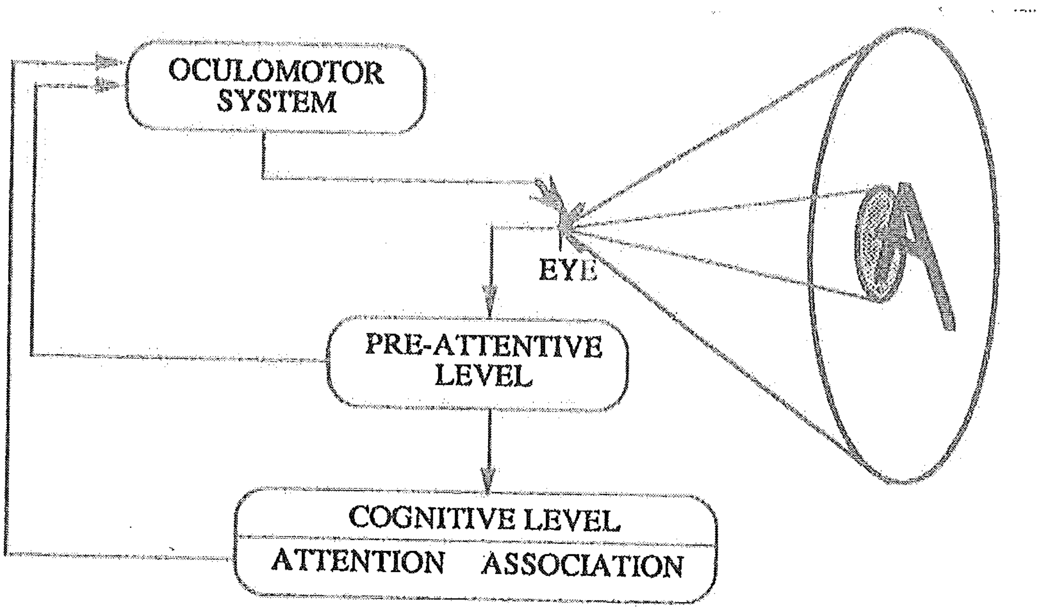


Figure 1. Structure envisaged when vision is a temporal process.

2. SUPPORTING THEORIES FROM NEUROSCIENCE AND PSYCHOLOGY

This section is a short summary of what is known of the processing of form in the visual system and saccadic eye movements. For more information, one may refer to (Hubel & Wiesel, 1977) (Hubel & Wiesel, 1979) (Kandel & Schwartz, 1985) (Sparks & Jay, 1987) (Robinson, 1987) (Grant, 1988) (Hubel, 1988).

We all have eyes whose acuity is more than 10 times greater in fovea than in the periphery. Recognition of an object whose reflection on the retina is bigger than the fovea necessitates saccadic eye movements to be able to have all the "interesting" parts of the object fall onto the fovea. When the object is still bigger, we need to move the head and eventually the body as well. The machinery dedicated to analyse a part of the visual field decreases exponentially as eccentricity, i.e., distance to the center of the fovea, increases. This is valid both in the retina and in the visual cortex. Higher spatial frequencies, i.e., details, can be detected in the fovea which are lost in the periphery.

Of the two types of ganglion cells in the retina that are related to processing of form, the X cells have small receptive fields and thus respond to details; they are concentrated in the foveal region of the retina. The Y cells have larger receptive fields and thus respond only to coarse features. Y cells transmit faster and are believed to carry an initial crude analysis of form. X cells are thought to be involved in the detailed high-resolution analysis of the visual image.

The X cells project to the lateral geniculate nucleus (LGN). The Y cells project to the LGN and to the superior colliculus (SC). The third type of cells in the retina, the W cells, project exclusively to the SC. SC is responsible for saccadic eye movements. The difference of processing for different spatial frequencies, detail and crude form, is also kept in the LGN and in later steps of processing in the visual cortex and other areas of the brain.

The *parvocellular* pathway starts from the layers of the LGN fed by the X cells and reach the visual cortex (V1, V2, V3, and V4). Simple, complex, and hypercomplex cells in the visual cortex extract lines of around 18 different orientations. The results are passed to the inferotemporal cortex which is a region where high-order associations are stored. In monkeys, lesions of this area, result in deficits in the rate of learning of visual tasks. The X pathway thus is thought to be concerned with visual discrimination learning.

The second, so-called *magnocellular* pathway starts from the LGN layers fed by the Y ganglion cells of the retina which then project to the visual cortex. The cells in the visual cortex who are part of the magnocellular pathway (V1, V2, and V3) project to the medial temporal area (V5 or MT) and from there to the posterior parietal cortex and then to the frontal eye fields which in turn project to the SC. This pathway is believed to be concerned with movement and attentional aspects. Certain cells in the parietal cortex respond to visual stimuli or during visually guided movements. These results are consistent with the notion that the parietal cortex is involved in processes associated with attention to the spatial analysis of sensory input and perhaps with the manipulation of objects in space. In particular, the activity of the cell is enhanced when the animal pays attention to the stimulus. When connections from frontal-eye fields are lesioned, patients cannot initiate voluntary saccades. They may however look compulsively at objects that appear abrubtly in the periphery, even when told not to do so. It is thought that this behaviour is caused by a SC response that has been released from higher control by the lesion.

SC is thought to be exclusively involved in saccadic performance and is believed to translate visual input to oculomotor commands. The superficial layer of the SC has a retinotopic organization. It receives visual input from the retina and the visual cortex as mentioned and these inputs are maintained in register, that is, a region of the SC that receives inputs from a specific retinal region will also receive input from an area of the visual cortex that processes information about the same retinal locus. It also receives projections from the SC. The deeper layer is a motor map. Neurons coding for the spatial location of the stimulus as well as for the direction and amplitude of saccades are arranged topographically. There is no evidence of direct connections between superficial and deeper layers however.

Added to this complexity is the existence of a thalamus which receives projections from almost all parts of the brain and which seems to project back everywhere as well. Information on any given modality is transmitted first to a primary cortical area and from there, either directly or via the thalamus, to successions of higher areas. The pulvinar of the thalamus has inputs from the SC and the primary visual cortex (V1) and has reciprocal connections with the parietal, temporary and occipital (V2 and V3) cortices.

Treisman proposed (Treisman & Gelade, 1980) that there is a parallel, pre-attentive system that performs extraction of disjunctive features only. The attentive system operating on this is a serial, i.e., sequentially operating, system incorporating a focus of attention and that deals with conjunction of features. Ullman also proposed (1984) that a "base representation" is computed in parallel everywhere which is then processed by a set of "visual routines" to find spatial relations like connectivity, inside-outside, etc.

The basic idea underlying parallel versus temporal processing can be explained as follows. Let us say we want to detect if there is a green T somewhere in the image. In a parallel scheme, we implement a "T-detector" and a "greenness-detector" and place them everywhere over the image. We then can say that "there is a green T" if both two detectors at a certain locus fire together. Thus we check for the conjunction by assigning an additional unit checking for that condition which we need to have at all loci throughout the visual field. To be able to recognize such combinations, one need to have detectors checking for such conditions. The problem, additional to cost, is that, if you do not have a "pink-P-detector," you will not be able to detect it.

In the temporal version, the T-detectors and greenness detectors are the same, but no conjunctive units are there. Instead, one has an attentive system, which first detects that there is a T somewhere by attending to the output of form detectors. This fact and its position in the visual field is recorded in a short-term memory somehow. The attentive system than attends to the output of color detector units and checks if the greenness detector is active also at the same locus. It is as if the attentive system has a "searchlight" with which first the form and later color detectors are highlighted.

The ability of selective attention in a general sense implies being able to extract, from a background of data, that part of the signal that is most relevant or interesting, e.g., one object in a scene cluttered with many objects, one speaker in a "cocktail party," etc.

3. LEARNING

There are two problems that one encounters when one wants to implement an attentive visual recognizer:

[1] Spatial relations should be mapped in a way into temporal relations. This is done by having an eye that sees a small part at a time but moves around over the image in time. Thus scene content is defined as a sequence of the content of the eye fixations. Such contents need be represented in an abstract form by a process of feature extraction. Additional to the content of fixations, the trajectory followed by the eye over the image need also be represented in a manner as to be able to take into account the relative spatial positioning of eye fixations, e.g., using polar coordinates relative to the previous fixation.

[2] Once such a sequence is generated, it should be represented in a certain way and used to look for a similar one during recognition or stored in the case of learning.

To be able to carry out the learning and recognition process as rapidly as possible, one looks for shortest possible sequences that allow one to differentiate between individual objects. This implies a somewhat on-line learning process in an incremental manner by quick changes on such sequences. One starts from short sequences where rough features are extracted during fixations. As similar objects are encountered, longer sequences and finer features need be used to be able to extract finer differences.

As previously mentioned, learning such sequences should be made very quickly, preferrably at one-shot. One cannot carry out an iterative, gradient-descent based procedure because first, there is not enough time for many iterations, and second, because iterative procedures' error definition is the sum of the errors on all of the vectors, other patterns, i.e., sequences, should also be explicitly stored somewhere as a training set in which one pattern gets modified or added and then the whole set needs to be learned once more. To get around this problem which appears due to the fact that weights are shared to store patterns, one can employ a local representation based on a competitive scheme where modifications in one unit do not affect the other units' weights. Grow-and-Learn (GAL) (Alpaydın, 1990) is one such algorithm for learning of categories where learning each pattern takes one iteration which sometimes do not require any modification of the network at all.

In the case of extracting the content of individual fixations, one needs to know the salient features. If learning is required, those should also be learned. The content of the fixation is then coded using a set of features, as the output of a set of feature detectors (Barlow, 1989). During an incremental learning process, the system should be able to add new features when significantly different patterns are encountered. What is more, as similar objects are encountered, to be able to differentiate between them, one requires a learning process that can automatically pass from coarse to finer features. These conditions are satisfied by the Grow-and-Represent (GAR) learning algorithm (Alpaydın, 1990) which is an incremental algorithm for unsupervised learning.

4. PRE-ATTENTIVE AND ATTENTIVE LEVELS

As fovea covers only a small part of the image, to be able to recognize an image, one needs to make a series of samplings moving the eye from one part to another. The attentive system, as mentioned previously, decides where to move the eye next. Relative positions of successive fixations should also be taken into account so that the trajectory followed can be known.

One strategy is to start from upper left of the visual field and scan sequentially till bottom right. Although this is feasible, it is not very intelligent. An intelligent strategy would be one that minimizes the number of fixations as this will decrease recognition time. This can be achieved if one can find a measure of being "interesting" and then look at only the interesting parts. A region, once looked at, is no longer interesting. Thus, top-down, memory-based information also has its part in the criterion that makes a region "interesting."

The bottom-up, i.e., input driven, component of the criterion of being interesting, first of all, should not be too difficult to compute as it needs to be computed in parallel for the whole visual field. On the other hand, it should give useful information. Events having smaller probabilities convey more information. Discontinuities are those parts that are not very probable, so, discontinuities are interesting. Discontinuities in an image are sudden changes in intensity, e.g., edges, lines, line-ends, corners, etc (Hebb, 1949) (Ullman, 1984).

This implies using a low-resolution filter to detect such discontinuities throughout the image in parallel everywhere. The functional system that performs an all-parallel extraction of basic features is called the *pre-attentive level* (PAL). It is completely input-driven in the sense that it is not in any way affected by any higher-level part like a long-term memory or any previous partial results. Although PAL carries out a feature extraction everywhere in parallel, the features detected are still represented locally on a retinotopic map; they are not tried to be mapped to higher order semantic features or classes. Thus, no association takes place there. PAL network does not need to be plastic; when it is, the learning is unsupervised.

So PAL first carries out a crude analysis of the periphery of the visual field, i.e., parafovea.This is to decide where next to fixate, and once a fixation is made, PAL also carries out a detailed analysis of the fovea to extract features that would allow the system to recognize classes. PAL thus has two outputs:

[1] To the attentive level, crude features detected in the parafovea,

[2] To the associative level, detailed features detected in the fovea.

It does not take too much imagination to propose that this pre-attentive processing is carried out by the visual cortex. The X pathway, passing through V1, V2, V3, and V4 carries out a detailed analysis of the fovea and later projects to the associative part, the inferotemporal cortex. The Y pathway passing through V1, V2, V3, and V5, later projects to the attentive part, the posterior pareital cortex and the frontal eye fields.

The superior colliculus has inputs coming from three sources: retina, visual cortex, and frontal eye fields. One can also conjecture as follows. As Y ganglion cells take time derivatives, the first is probably related to reflex. The second input source, the one from the visual cortex, corresponds to the bottom-up criterion of being interesting. The third one, from the frontal eye fields, is concerned with top-down, voluntary, learned control of the eye movements and thus corresponds to the top-down part of the criterion of being interesting. The thalamus then acts as a policeman mediating between the top-down and bottom-up demands on the superior colliculus.

The eye movements in the beginning are directed by the low-level information coming from the pre-attentive level giving a symbolic version of the content of parafovea. As more fixations are made, the associative system generates hypotheses about the identity of the scene and then the attentive system directs the oculomotor system to check for certain conditions through a mechanism of *selective attention* For example, "this letter can be 'y', check if it has a tail."

Types of attention

One should note the following point which was also made before. The type of coordinates one uses depends on the medium in which attention is made, thus the type of attention:

[1] One perceives the image, optionally processes it to get a more abstract form and places it in a memory which is preferrably retinotopically organized. In such a case, attention implies extracting one part of the memory, i.e., a memory to memory copy. This type of attention may be done for any modality and is not limited to eye movements or vision, it may be called *internal attention* In this case, one has absolute coordinates which are in fact memory addresses. The memory to memory copy is done by enabling one part of the memory and disabling the rest. For example, Crick proposed (1984) that the thalamic reticular complex does a similar job in enabling one part of the thalamus.

[2] The second type of attention is directly related to eye movements. Here, attending to another part means receiving a different part of the signal, i.e., looking at a different part of the image, what we can call *external attention* In this case, one cannot use absolute coordinates because the only coordinates available are retinotopic,viewer-centered, and as eyes move, those change. In such a case, the most logical is to use relative coordinates, the displacements made while making a saccade. So, the position of each fixation is coded relative to the previous one.

The best compromise between cost and speed is when one stores an abstract and rather rough version of the complete image in memory using which a first evaluation is made. When a more detailed version of any part is required because it is "interesting" enough, the eyes are moved to that region and a detailed analysis is made. The memory is organized in a retinotopic manner. What is important is that when the attentive system decides to concentrate on one part of the image based on the memory content, the mapping from the memory address to a retinotopic address should be done so that proper command signals can be sent to the motor units controlling saccades. This is for example critical when the memory is not allocated in equal amounts to parts of the visual field. In the retina and the visual cortex for example, more memory is devoted to regions closer to the center of the visual field. Grossberg and Kuperstein (1989) show how such mappings can be learned.

5. CONCLUSIONS

In rich environments where the dimensionality of the signal is high or when class definitions are complex, parallel recognizers need very many units. Assigning one unit to check for one value combination at one particular position, one soon reaches a very big number which best can be explained using the term *combinatorial explosion* However, to be able to get systems with a reasonable amount of complexity, vision may turn into an operation through time. In this way, spatial relations are mapped into temporal ones. Evidence shows that this is also the strategy chosen by nature.

Although the idea is promising, still more work needs to be done. Especially, a mathematical framework is required by which temporal operations can be defined and processed. The advantages of this approach cannot be underestimated:

[1] The complexity of the system is reduced at the expense of slower speed.

[2] A reasonable pattern-recognition task requires networks containing on the order of thousands of units. We are far away from being able to build such machines,and it seems as if we are bound for still some more time to sequential hardware. It is therefore a better idea to get the best of it by applying a sequential high-level strategy on it. At the low, basic feature extraction level connectivities are local, generally in the form of convolutions. Processing in these levels can be easily parallelized and hardware for to handle such local combinations are feasible like cellular automata machines or convolution filters. At the higher levels where connectivities are large, one either needs very complex hardware or look for simple sequential solutions because sequentiality when properly introduced, simplifies the system operation a lot.

[3] The third advantage is that sequential processing is a domain we know already. For decades, computer scientists worked on grammars, finite-state machines, parsing etc. One very popular approach in pattern recognition is syntactic pattern recognition where considerable work has been and is being done. One may, by a compromise as proposed by temporal vision, profit from all this experience.

REFERENCES

[1] Alpaydın; E. (1990) Neural models of incremental supervised and unsupervised learning, *PhD Thesis, EPFL, Lausanne,* Switzerland.

[2] Barlow, H.B. (1989) "Unsupervised learning," *Neural Computation,* 1, 295--311.

[3] Crick, F.H.C. (1984) "The function of the thalamic reticular complex: The searchlight hypothesis," *Proceedings of the National Academy of Sciences,* **81**, 4586--4590.

[4] Feldman, J.A., Ballard, D.H. (1982) "Connectionist models and their properties," *Cognitive Science,* **6**, 205--254.

[5] Grant, P.E. (1988) "Language processing: A neuroanatomical primer," in *The alphabet and the brain,* D. de Kerckhove, C.J. Lumsden (eds.) Springer, 246--272.

[6] Grossberg, S., Kuperstein, M. (1989) *Neural dynamics of adaptive sensory-motor control,* Pergamon Press.

[7] Hebb, D.O. (1949) *The organization of behaviour,* Wiley. Chapter 5: "Perception of a complex: The phase sequence," in *Brain theory:* Reprint volume G.L. Shaw, G. Palm, (eds.) (1988) World Scientific, 224--270..

[8] Hinton, G.E., McClelland, J.L., Rumelhart, D.E. (1986) "Distributed representations," in *PDP,* D.E. Rumelhart, J.L. McClelland (eds.) **1**, MIT Press, 77-109.

[9] Hubel, D.H. (1988) *Eye, brain, and vision,* Scientific American Library.

[10] Hubel, D.H., Wiesel, T.N. (1977) "Functional architecture of the macaque monkey visual cortex," *Ferrier lecture, Proc. Royal Soc. London B,* **198**, 1--59.

[11] Hubel, D.H., Wiesel, T.N. (1979) "Brain mechanisms of vision," *Scientific American,* **241(3)**, 130--144.

[12] Kandel, E.R., Schwartz, J.H. (1985) *Principles of neural science,* 2nd edition, Elsevier.

[13] Pitts, W.H., McCulloch, W.S. (1947) "How we know universals: The perception of auditory and visual forms," *Bulletin of Mathematical Biophysics,* 9, 127--147 also in Embodiments of mind, W.S. McCulloch (1965) MIT Press, 46--66.

[14] Robinson, D.A. (1987) "Why visuomotor systems don't like feedback and how they avoid it," in *Vision, brain, and cooperative computation,* M.A. Arbib,A.R.Hanson (eds.) 89-108.

[15] Sparks, D.L., Jay, M. (1988) "The role of the primate superior colliculus in sensorimotor coordination," in *Vision, brain, and cooperative computation,* M.A. Arbib, A.R. Hanson (eds.) 109--128.

[16] Treisman, A., Gelade, G. (1980) "A feature-integration theory of attention," *Cognitive Psychology,* 14, 107--141.

[17] Tsotsos, J.K. (1987) "Representational axes and temporal cooperative processes," in Vision, brain, and cooperative computation, M.A. Arbib, A.R. Hanson (eds.) MIT Press, 165--182.

[18] Ullman, S. (1984) "Visual routines," *Cognition: International journal of cognitive psychology,* **18** also in *Visual cognition,* S. Pinker (ed.) MIT Press, 97--159.

[19] Williams, R.J., Zipser, D. (1989a) "A learning algorithm for continually running fully recurrent neural networks" *Neural Computation,* **1**, 270--280.

[20] Williams, R.J., Zipser, D. (1989b) "Experimental analysis of the real-time recurrent learning algorithm," *Connection Science,* 1, 87--111.

Renkli Resim Islemenin Teorik ve Pratik Ozellikleri

D.T.Pham ve Y.Samast

Intelligent Systems Laboratory, School of Electrical,
Electronic and Systems Engineering, University of Wales
P.O. Box 904, CARDIFF, CF1 3YH / U.K.

Özet:

Renk, nesnelerin tanınmasında çok önemli bir bilgidir. Rengin dogru kullanılması, bir "yapay görme sistemi"nin gücünü ve verimini büyük miktarda artırmasına rağmen, yanlış kullanımı ise böyle bir sistemin verimini oldukça düşürebilir. Yapay görme uygulamalarında rengin etkili bir şekilde kullanılması için tamamlanmış bir kurallar seti geliştirmek mümkün olmamasına rağmen, insanın renk görme sistemi temel alınarak, elde bulunan renk teorileri yardımıyla geniş manada bazı temeller kurulabilir.

Bu çalışmada, bugüne kadar geliştirilmiş renk teorileri ve renk modelleri (uzayları) kısaca târif edilecektir. Renkli resim işleme ile ilgili bu teorik bilgilerin pratik özellikleri tartışılacak ve rengin, yapay görme ve resim işleme uygulamarında etkili kullanımı için geleceğe ait bazı araştırma yönleri belirtilecektir.

Anahtar Kelimeler: Yapay görme, renk teorileri, renk modelleri, renkli resim işleme.

Theoretical and Practical Aspects of Colour Image Processing

Abstract:

Colour is one of the most important properties of an object. Properly used, colour can be a powerful visual cue to improve the usefulness of an artificial vision system in object recognition tasks. However, the inappropriate use of colour can seriously reduce the performance of such a system. Although it is not possible to develop a complete set of guidelines for the effective use of colour in all artificial vision applications, some broad principles, based on the mechanisms of human colour perception and colour theories, can be established.

In this paper, the major colour theories and colour models (spaces) developed so far will be briefly described. Practical aspects of colour image processing will be discussed and some future research directions in the effective use of colour in artificial vision will be indicated.

Keywords: Artificial vision, colour theories, colour models, colour image processing.

1. GIRIS

Renk teorileri, rengin bütün özelliklerini açıklamakta yetersiz kalmaktadır. Bunun önemli bir sebebi "insan renk idraki"nin henüz tam manasıyla anlaşılamamasından ve idrak işleminin çok aşamalı ve karmaşık olmasından kaynaklanmaktadır. Bilgisayarlarla renkli resim işlemenin temelleri, anlaşılması ve başarısı renkli resim yapısının ve renk modellerinin iyi anlaşılmasıyla çok yakından alakalıdır. İyi bilindiği gibi, sadece gri-seviye resim ile, bir manzaradan göze gelen bütün bilgiyi temsil etmek imkansızdır. Uzay (pozisyon, mesafe, vb.) ve renk bilgisi de göz önünde bulundurulması gerekmektedir.

İdrak edilen renk insanın bir ürünüdür, nesnenin sahip oldugu fiziki bir özellik değil denilebilirse de, makineler ile yapılan renk ile ilgili işlemler, nesnelerden elde edilen bazı fiziki gözlemler ve değerlendirmeler üzerine bina edilmek mecburiyetindedir.

Rengin tarifi, rengin subjektif-enfüsi bir tecrübe olmasından dolayı, çok zordur. Rengin beyinde olan idraki için üç ana unsur gereklidir. Bunlar; "görülecek olan-nesne", "görecek olan-gözlemci" ve "gösterecek olan-ışık"tır. Renk, gözle alınıp beyinle yorumlanılan ışığa karşı psiko-fiziki bir tepkidir. İnsanın renkleri idrak etmesi henüz tam anlaşılamayan bir fizyo-psikolojik fonemen olmasına rağmen, "renk"in fiziki tabiatı teori ve deneye dayalı sonuçlarla desteklenerek formülleştirilebilir.

Renkler, temelde üç faktör göz önüne alınarak ayrılır. Bunlar, "Renk-hue", "Doyum-saturation" ve "Parlaklık-brightness"dır. Renk-hue, ışık dalgalarının karışımındaki hakim dalgaboyu ile ilgilidir. Bir nesnenin rengine kırmızı, sarı veya mavi demek ile onun "Renk-hue"i belirtilmiş olur, yani Renk-hue, bir nesnenin "dogru-renk"ini temsil eder ve "renk" kelimesi ile eşanlamlıdır. Doyum, renklerin saflığına dayanır, diğer bir deyişle bir Renk-hue ile karışmış beyaz ışık miktarının bir ölçüsüdür ve bir rengin soluk veya canlı olmasını belirleyen bileşendir. Parlaklık ise keskinliğe (intensity) dayanır.

Bu makalede, bu bölümdeki girişten sonra ikinci bölümde; renk teorilerinden Üç-renk Teorisi, Zıt-renk Teorisi ve Retinex Teorisi genel olarak gözden geçirilecek ve bu teorilerin insan renk idrakine göre eksikliklerinden bahsedilecektir. Üçüncü bölümde "renk"i temsil etmek için geliştirilen renk modelleri fiziki, (perceptual) ve (conceptual) renk modelleri başlıkları altında incelenecektir. Dördüncü sırada olan renkli resim işleme bölümünde, renk modellerinin resim işlemeye tesirleri ve birbirlerine göre üstünlükleri örneklerle verilecektir. Sonuç bölümünde ise doğru-renk bilgisinin yapay-gorme ve resim işleme uygulamalarında bazı etkili kullanımlarından ve geleceğe ait bazı araştırma konularından söz edilecektir. Yapay görme'de renk bilgisinin teorik temelleri hakkında daha ayrıntılı bilgi [1] [2] [3] [4] [5]'den elde edilebilir.

2. RENK TEORILERI [1] [2] [5]

Bugüne kadar birçok renk teorisi ortaya atılmasına rağmen rengin bütün özelliklerini açıklayabilen dünya genelinde kabul görmüş bir teori henüz geliştirilememiştir. Bu yüzden, insanların rengi nasıl idrak ettikleri ve rengin tam manasıyla nasıl temsil edilebileceği sorularına henüz cevap bulunamamıştır. Burada, Üç-renk Teorisi, Zıt-renk Teorisi ve Retinex Teorisi sırasıyla incelenip tartışılacaktır.

2.1. Uc-Renk Teorisi [1]

Üç-renk teorisine göre: "Bir nesnenin rengini, tayfdaki yerleri Kırmızı, Yeşil ve Mavi diye bilinen üç görme kanalının çıkışı belirler."

Bu teorinin basit olması, retina'nın yapısına uygun olması ve renk karıştırma prensiplerini açıklaması söylenebilecek temel iyilikleridir. Bunun yaninda bu teorinin, dezavantaj denebilecek, açıklayamadığı önemli fonomenler vardır. Bunlardan önemli ikisi:

i) Renk Sâbitliği: İdrak edilen renkler, ceşitli aydınlatma şartları altında belirli bir dereceye kadar sâbit kalırlar. Bu fonemen rengin, nesnenin üzerine gelen ışık ile değil nesnenin yüzey özellikleriyle ilgili olduğunu gösterir.

ii) Renk Kontrastı: Gözlemler neticesinde edinilen tecrübe odur ki, bir bölgenin idrak edilen rengi, etrafını saran renklere göre değişmektedir. Mesela; kırmızı geniş bir alanda gri bir nokta yeşil gibi idrak edilir.

Bu fonemenlerin açıklanamamasının sebebi, her iki durumunda da retina üzerine gelen ışığın tayfî kalitesinin değişmemesidir.

2.2. Zit-Renk Teorisi [6]

Bu teoride üç-renk teorisi gibi üç kanalın varlığını kabul eder, fakat bu kanalların özelliklerinin farklı olduğunu söyler. Bu kanallar seti ikili değişkenlerden oluşmuştur. Bunlar, 'Sarı-Mavi (S-M)', 'Kırmızı-Yeşil (K-Y)' ve 'Beyaz-Siyah (B-Si)'dir ve üç zıt işlemin tepkisini gösterirler. Yani mavi'nin idrak edilmesi sarı'nin idrak edilmesine direnir, kırmızı ve yeşil iki zıt kuvvettir ve aynı şekilde siyah ve beyaz da öyle. Yalnız bu işlemler, alıcılar seviyesinde değil daha sonraki seviyelerde olur.

Her işlemin belli bir parlaklık seviyesine bağlı olarak aktif hale geldiği farklı bir "eşik" değeri vardır. İnsanın renk idrakinden elde edilen bilgilere göre, bu teori, çarpım faktörünün ve eşik değerinin S-M işleminde K-Y işleminden daha büyük olduğunu ve B-Si işlemi için eşik değerinin diğer iki işlemin eşik değerinden daha küçük olduğunu söylemektedir. Parlaklığın artmasıyla, ilk önce B-Si işlemi aktif hale gelir ve bu da çok düşük ışıktaki renksiz görme idrakine karşılık gelir. Çok iyi bilindiği gibi, düşük ışıklarda nesnelerin renklerini ayırtetmek oldukça güçtür. B-Si'den sonra K-Y işlem aktif olur ve böylece kırmızı yeşilden ayırtedilmeye başlanır. En sonunda S-M işlemi aktif hale gelir ve bu işlemin tepkisi K-Y işleminkinden daha hızlı yükselir (daha büyük çarpim faktörü yüzünden). Parlak ışık altında sarı ve mavi, kırmızı ve yeşilden daha iyi ayırtedilir. Parlaklığın daha fazla artmasıyle bütün işlemler doyuma ulaşır ve beyaz hâkim hale gelir.

Renk Konrastı durumunda bu teori, kırmızı tepkinin etrafını saran noktalarda daha küçük yeşil tepki ile beraber olduğu, bir endüksiyon işleminin varlığını öne sürer. Üç-renk teorisi ve bu teori, idrak edilen bir noktanın renginin, o noktadan retina'ya gelen ışığın tayfi özelliğine bağlı olduğunu söyler. Renk Sâbitliği ve Renk Kontrastı özelliklerini anormallik olarak değerlendirirler.

2.3. Retinex Teorisi [7]

Bu teoriye göre, göze gelen ışık, nesneyi aydınlatan ışığın keskinliğinin ve nesnenin yüzeyinin reflektansının bir ürünüdür. Gözdeki bu bilginin iki bileşeninin, uzaydaki dagılımları farklıdır. Aydınlatma bileşeni resimde normal ve yavaş bir değişim gösterir. Reflektans bileşeni ise nesnelerin üzerinde sabit iken nesnelerin arasında çok âni değişim gösterir. Bu teoride, idrak edilen renk doğrudan nesnelerin yüzeylerindeki reflektansın bir neticesi olarak görülmektedir.

Bu teori, yerel olarak birbirine bağlı olmayan üç adet alıcı kanaldan oluşan bir renk görme modeli teklif eder. Bir resimdeki herbir nokta, üç farklı resimde birer nokta olarak değerlendirilir. Herbir alıcı kanalın çıkışı, "aydınlatma bileşeni"ni ayırmak için diğerlerinden bağımsız olarak işlenir ve bir noktanın rengi bu üç farklı işlemin çıkışının karşılaştırılması neticesinde bulunur. Bu üç işlemin birleştirilmiş çıkışı (lightness diye adlandırılır), yüzey reflektansı ile bir hayli etkileşmiştir ve idrak ile ilgili bir niceliktir. Bu niceliği (lightness) hesaplamak kolay değildir. Hesaplayabilmek için birçok kabûller yapılmaktadır. Genellikle yapılan temel kabûller:

i) Aydınlatma resimde çok hafif ve yumuşak bir şekilde değişir, yani gölgeler ve parlak yansımalar gibi etkiler yoktur.

ii) Reflektans, nesnenin yüzeylerinde sabittir; yani yüzeylerin yönü veya herhangi bir gölge tesiri reflektans değerini değiştirmez.

Yapılan bu kabûller hesaplamaların gerçek ortamlarda yapılmasını engellemektedirler. Bu yüzden, bugüne kadar gerçek resimler üzerinde çalışmalar yapılamamıştır. Kabûllerdeki şartları sağladığından, çalışmalar daha cok Mondrian diye bilinen özel resimler üzerinde yapılmıştır. Bu resimler, keskin kenarlar ile birbirinden ayrılmış düzgün bir dağılıma sahip renk bölgelerinden (parcalarından) meydana gelmiştir. Bu çalışmaların, gerçek dünyaya ait resimler üzerinde gerçekleştirilmesine çok fazla ihtiyaç vardır.

3. RENK MODELLERI [1] [2] [5]

Renkli resim işlemenin temeli, birçok sayıdaki renk modellerini ve onların uygulamalarını iyi anlamaktır. Bu çalışmada renk modelleri genel olarak, fiziki, (conceptual) ve (perceptual) renk modelleri diye üç ayrı grup altında incelenecektir. Fiziki renk modelleri, renkleri gösterme, gönderme, alma (televizyon ve video) ve basım gibi işlemlerde kullanılırlar. (perceptual) renk modelleri idrak edilebilir veya ölçülebilir olanlardır. Bu iki renk modeli gercek dünyanın özelliklerini yansıttığından önemlidir. (Conceptual) renk modelleri, (perceptual) modeller ile fiziki modeller arasında bir geciş gibidirler. (Conceptual) modeller "renk bilgisi"nin işlenebildiği modellerdir, yani bu modellerde her türlü matematik işlem renkli resimler üzerinde gerceklestirilebilir.

3.1. Fiziki Renk Modelleri

Fiziki renk modelleri, renk uzayında çalışan aletlerin (video, televizyon, basım makineleri, vb.) özelliklerini yansıtır. Renkli resim işleme uygulamalarında en çok bilinen fiziki modeller, KYM (Kırmızı, Yeşil ve Mavi) Modeli, ve YIQ (Y = Luminance, I = In-phase ve Q = Quadrature) Modeli'dir.

3.1.1. KYM Renk Modeli

Bu fiziki model, video giriş ve çıkışı icin kullanılmaktadır. Bu modelde, renk uzayı, üç köşesinde Kırmızı (K), Yeşil (Y) ve Mavi (M) diğer üç köşesinde Siyan (S), Macenta (M) ve Sarı (Sa) bulunan bir küp olarak düşünülebilir. Siyah merkezdedir ve çapraz (diagonal) karşısında Beyaz vardır. Gri'nin tonları, siyah ile beyaz'ı birleştiren (K, Y ve M değerlerinin birbirine eşit olduğu) çapraz eksen üzerinde bulunur.

KYM küpü içinde, herbir "resbir" (resim birimi), bir noktadır ve K, Y ve M koordinat değerleriyle belirlenir. Bu temsil modelinde bütün renkleri tarif etmek oldukça zordur; kahverengi'ni Kırmızı, Yeşil ve Mavi'nin miktarlarına göre tarif etmek gibi. KYM Modeli, insanın renk idraki ile de tam ahenkli değildir.

3.1.2. YIQ Renk Modeli

Bu model, başlangıçta, elde bulunan "siyah-beyaz" televizyon yayın sistemine uygun olarak geliştirilmiş ve yayın verimi mümkün olduğu kadar yüksek tutulmaya çalışılmıştır. Y bileşeni lüminanstır ve televizyon seyrederken bir insanın idrak ettiği parlaklığın bir ölçümüdür. Siyah-beyaz televizyonlar sadece bu işareti almaktadırlar. YIQ Renk Modeli, KYM Modeli'nin lineer bir çevrimidir. YIQ Modeli'nde renk bileşenlerinin (yani I, Q) taradığı saha (dynamic range), yayın bandında renk alt-taşıyıcısı icin mevcût olan bandgenişliği ile sınırlıdır. Eğer bu bileşenler bandgenişligini aşarlarsa ve lüminans bandına taşarlarsa televizyon ortasında bazi yayin bozukluklarina sebeb olurlar. I ve Q kanalındaki mümkün bandgenişliği azaltılması saklama kapasitesini düşürmek için optimum bilgi ile yapılabilir (resbir başına 8-bit'ten 3-bit'e kadar düşürülebilir). KYM gibi, YIQ koordinatlarıyla da renkleri tarif etmek kolay degildir ve bir çeşit renk isimlendirme işlemine ihtiyaç vardır.

3.2. Perceptual Renk Modelleri

Perceptual renk modelleri, idrak edilebilir ve ölçülebilir olanlardır ama bu modellerde renk işleme işlemleri için bir temel bulunmaz.

3.2.1. CIE Renk Modeli [1]

"CIE Renk Şeması", cevresinde saf renkleri ve ortasında bir nokta ile standart beyaz ışığı temsil eden bir renk modelidir.

CIE (Commission Internationale L'Eclairage- Uluslararası

Aydınlatma Komisyonu) tarafından temel renkler için verilen standart dalga boyları sırasıyla: Mavi=435.8 nm, Yeşil=546.1 nm ve Kırmızı= 700 nm'dir. Renk şeması'nda saf renkler mor renkten (380 nm) kırmızı'ya (780 nm) kadar, şeklin kenarlarına dizilmişlerdir ve kenardan iç kısımda kalan her nokta tayfi renklerin bir karışımını temsil eder, yani iç kısımda olan renkler saf değillerdir. Eşit enerji noktası diye bilinen nokta, temel renklerin eşit miktarlarda karışımı neticesi elde edilen, CIE'nin standart beyaz ışığıdır. Kenardan içeriye eşit enerji noktasına doğru ilerledikçe, renge beyaz ışık eklenmeye başlar ve renk daha az doymuştur denir. Bundan da anlaşıldığı gibi, eşit enerji noktasında (beyaz ışık) doyum sıfırdır. Renk Şeması üzerinde iki noktayı birleştiren bir doğru, iki üç noktadakı iki rengin karışımı ile üretilebilecek renkleri belirler. Mesela; Kırmızı ile Yeşil arasindaki cizgi bize kırmızı ile yeşilin degişik miktarlarda birbiriyle karıştırılınca elde edilebilecek renkleri verir. Aynı şekilde, herhangi bir tayfi rengi eşit enerji noktasına bağlayan doğru da o rengin bütün tonlarını verir. Bu ikili kombinezon, üçlü hale rahatlıkla getirilebilir. Üçlü renk durumunda, üç nokta doğrularla birleştirilir ve bir üçgen elde edilir. Bu üçgen icindeki herhangi bir renk, ücgenin köşelerini oluşturan üç rengin belli bir miktarda karışımı ile elde edilir. Temel renkler diye adlandırdığımız, kırmızı, yeşil ve mavi'nin karışımı ile üretilebilecek bütün renkler şema üzerinde bu üc noktayı birleştiren ücgen içinde kalacağına göre, şema iyi incelendiğinde KYM ile fiziki olarak temsil edilemeyen, fakat görülebilen renklerin olduğu dikkat çekecektir. Böylece sıkça söylenen 'KYM Modeli ile bütün renkleri üretmek mümkündür' ifadesinin yanlışlığı da tabii olarak ortaya çıkmaktadır.

Bu modelde renkler fiziki olarak ölçülebilirken, hesaplamaların yapılması oldukça karmaşıktır.

3.2.2. Munsell Renk Modeli

Munsell Renk Modeli, renklerin (Hue), (Chroma) ve (Value) değerlerine göre düzenlenerek oluşturulmuş ve atlas şeklinde basılmış bir modeldir. Bu model, her ne kadar kişiden kişiye değişebilirse de renklerin karşılaştırılmasında ve eşleştirilmesinde kullanılabilir. Bu modele benzer diğer bir ticari model de Pantone Renk Modeli'dir.

3.3. (Conceptual) Renk Modelleri

Bu modeller, renkli resim işleme için kullanılan modellerden en müsait olanlarıdır. HSI (Hue-renk, Saturation-Doyum ve Intensity) ve HSV (Hue, Saturation ve Value) verilebilecek örneklerdir. Örnek olarak verilen her iki modelde de bir resmin parlaklığı (HSI'da I ve HSV'de V) renk bileşenlerinden (H ve S) ayrıdır. HSI ve HSV arasındaki fark parlaklık bileşeninin hesaplanmasıdır ve bu hesap Parlaklık ve Doyum bileşenlerinin taradığı sahayı ve dağılımlarını belirler [5]. Bu modeller, KYM'den lineer olmayan çevrimlerdir ve cevrim işlemi dikkat ister [8]. Herbir bileşenin insan gözü için bir manası vardır.

4. RENKLI RESIM ISLEME [2] [4] [5]

Renkli resim işleme, "sahte-renk" ile degil "doğru-renk" ile ilgilenir. Bir doğru-renkli resim işleme usûlü üç aşamalıdır. İlk önce, resim KYM fiziki temsil modelinden bir (conceptual) modele çevrilir. Ikinci aşamada, resim işleme operasyonları çevrilen (conceptual) modele göre yapılır. Son aşamada ise, işlemler sonucu resim gösterilmek için tekrar KYM modeline cevrilir.

Bugün, resim işlemede, gri-seviye resim işlemeden doğru-renkli resim işlemeye geçmeye büyük ihtiyaç ve aynı zamanda talep de vardır. Tıb, basın-yayın, cografik bilgi işleme sistemleri, endüstriyel inceleme ve dizayn doğru-renk bilgisinin çok önemli olduğu başta gelen sahalardır.

Rengi kullanmanın ehemmiyeti şu noktalarda yatar:

- Renk, aynı şekilde ve büyüklükte olan nesneleri ayıran tek özelliktir.
- Renk, insanın gerçek dünyayı tanımasında ve idrak etmesinde önemli bir role sahiptir.
- Belli bir dereceye kadar, renkler, farklı aydınlatma şartlarında aynı idrak edilirler (Renk Sâbitliği), yani renk aydınlatma şartlarından az etkilenir.

Conceptual renk modelleri, resim işleme acısından diğer modellerden daha üstündürler ve bunların önemli faydalarından ikisi; mevcud tekrenk veya gri-seviye tekniklere olan uyğunluğu ve modeldeki herbir bileşenin insan gözü icin manalı bir bilgi olmasıdır. Bileşenlerinin, diger modellere nazaran bağımsız olması, bu modellerin tek-renk teknikler kadar hızlı olmasını ve bu tek-renk tekniklerin doğrudan uygulanmasını sağlar. Bu çeşit zaman kazanma, renk doğruluğunu bozmadan, bütün resim işleme operasyonları için geçerlidir. Mesela; kırmızı bir gülün rengi sadece Hue bileşeni degiştirilerek sarı yapilabilir. Bir elma, bir portakaldan hue histogram'ı ile ayrılabilir. KYM Modeli'nde, K-Y-M tek tek işlenmeleri ve işlem sonucunda birleştirilmeleri gerekir. Çünkü ancak hep beraber anlamlı bir bilgi verirler. Bir KYM resminin renklerini değiştirmek için, kırmızı, yeşil ve mavi'nin doğru karışımları bulunması gerekir. Bu işlemin sonuçları çoğu zaman tahmin edilemez. Sonra, elde edilen bu yeni K, Y ve M değerleri resimde doğru yerlerine konulmalıdır. Görüldüğü gibi, conceptual modellerde tek aşamalı bir işlem KYM'de çok aşamalı ve karmaşık olmaktadır.

Bu modelde bileşenler birbirini çok etkilemektedir. Bu yüzden bu modeldeki renkli resimlerin işlenmesi ve incelenmesi çoğu zaman zordur. Mesela; gri-ton resimlerin kontrastını artırmada çok iyi sonuçlar veren histogram eşitleme algoritması, KYM resimlerine uygulandığında resmin tabii renk görünüşünü bozar. Bu sebeple, renkli resimlerde kontrast artırma işlemi conceptual modeller ile yapılmalıdır. Renk-hue segmentasyon için çok etkili bir bileşendir. Renk-hue bileşeninin, Parlaklık bileşenine göre aydınlatmadaki degişikliklerden daha az etkilenmesi sebebinden, Renk-hue histogram teknikleri gayet iyi segmentasyon ve analiz sonuçları verir.

Düzgün olmayan ışıklandırma dağılımı, resimdeki bazı bolgelerde renklerin koyu tonlardan neredeyse beyaz'a kadar degişmesine sebeb olur. Sadece, Doyum bileşeni ayarlanılarak bu yanlış ışıklandırma tesirlerinin düzeltilmesi mümkündür.

Parlaklık bileşeni, renkli resimdeki gri-ton unsurudur ve tek-renk resim işleme ile renkli resim işleme arasında bir bağlantı gibidir. Çok yaygın resim işleme operasyonları, (eşikleme, laplas filtresi, mantık operatorleri, vb.) Parlaklık bileşenine rahatlıkla doğrudan uygulanabilirler ve tek-renk resimler ile aynı sürede sonuç alınır. Parlaklık artırma neticesinde istenmeyen renk parlaklıkları ortaya çıkabilir. Bu durum ozellikle yüksek doyum ve düşük parlaklık olan bolgelerde daha açık olarak görülür. Yüksek doyum, düşük parlaklık ile örtüldüğü için bu durum işlemden önce göze çarpmaz. Uygun bir paralel Doyum işleme tekniği bu etkiyi azaltabilir.

Conceptual Modeller ile çalışan kameranın veya monitörün olmaması, işlemlerinin bu modellere doğrudan uygulanabilirliğini imkansız kılmaktadır. Conceptual renk modelleri, gıda ve ziraat endüstrisine, ürünlerin incelenmesinde daha objektif sonuçların alınmasında yardımcı olur. Tıbbi uygulamar, mesela; plastik cerrahi, doku analizi ve dermatoloji conceptual modellerden çok faydalanabilir. Deri (pigment)lerinin eşleştirilmesi, Renk ve Doyum değerleri kullanılarak yapılabilir. Resim işlemede önemli alanlardan olan hareket analizi'nde ve robotik görme'de conceptual modellerle birçok kolaylık elde edilebilir.

Renk model cevrimi sırasında meydana gelebilecek bilgi kayıpları ve değişimleri çok dikkat isteyen noktalardır [8]. Resim kontrastının düşük olduğu durumlarda renk bilgisi çok önem kazanır. Böyle durumlar için renk bilgisine dayalı resim işleme tekniklerine çok fazla ihtiyaç vardır. İnsan gözünün idrak ettiği tabii renk miktarı bütün makina ve tekniklerin çok üzerindedir [9]. Bu durum, renkli resim işleme algoritmaları geliştirilirken ve bu algoritmaların başarıları karşılaştırılırken dikkate alınmalıdır.

Renkli resimlerin manalı parçalara ayrılması işleminde renk temsili için modellerden birisi veya modellerin herhangi bir ceşit kombinezonu kullanılmaktadır. Çünkü sayısal bilgi resbir (resim birimi) seviyesinde bu temsil şekliyle daha iyi idare edilir. Buna rağmen, temsil modellerinin, renkleri insanlar gibi doğrudan tarif edebilme ve temsil için ihtiyaç duydukları bilginin fazla olması açısından eksikliklere sahip olmasından dolayı alternatif temsil modellerine ve bu modellere göre de alternatif resim işleme tekniklerine ihtiyaç vardır. Bir alternatif; her resbir'deki gri-ton parlaklık değerine ve renk ismine göre rengin "sembolik" temsili önerilebilir [10]. Bu temsil modeli, mevcut segmentasyon algoritmaları ile kullanılamaz ama iyi bir bilgi indirim vasıtası olur ve bir düşük-seviye yorumlayıcının yardımıyla "düşük-seviye sembolik işlem" kâbiliyeti sağlar.

Renk aktif görme'de bir resmin "netlik" değerini ölçmede kıstas fonksiyonu olarak; kameranın nereye bakması gerektigini tespit için ve pencerenin yerine ve ebatlarına karar vermede kullanılabilir [11] [12].

Renk bakış açısından ve yoğunluğu (resolution)dan çok az etkilendiği için bilinen bir nesnenin manzaradaki yerinin bulunmasında ve/veya manzarada yeri bilinen bir nesnenin ne olduğunun tespit edilmesinde kullanılabilir [13].

Makinalara yol gösterme ve kontrol alanında önemli uygulamaları olan 3-boyutlu (3-B) resim tekniklerinin güclük cektiği eşleştirme noktalarının bulunmasındaki karmaşıklık, renk bilgişi yardımıyla düşürülebilir [14].

Renk bilgisini, diğer görme bilgileriyle (hareket, mesafe, vb.) birleştirmek [15], bir "genel amaçli yapay görme sistemi" için çok önemli olduğundan bu çalişmalar üzerinde ciddiyetle durulmalıdır.

6 - SONUC

Rengin idrakinde bilinmesi gereken önemli iki nokta: göze gelen ışığın, nesneyi aydınlatan ışığın keskinliği ve o nesnenin yüzeyinin reflektansının ürünü ve "renk"inde sinir sisteminin ürünü olduğudur.

İdrak edilen renk, insanın bir ürünüdür, nesnenin sahip olduğu fiziki bir özellik değil denilebilirse de, makineler ile yapılan renk ile ilgili işlemler, nesnelerden elde edilen bazı fiziki gözlemler ve değerlendirmeler üzerine bina edilmek mecburiyetindedir.

Bilgisayarlarla renkli resim işlemenin temelleri, anlaşılması ve başarısı renkli resim yapısının ve renk modellerinin iyi anlaşılmasıyla çok yakından alakalıdır. Bugüne kadar bir çok renk teorisi ortaya atılmasına rağmen rengin bütün özelliklerini (renk sabitliği ve renk kontrast gibi) açıklayabilen dünya genelinde kabûl görmüş bir teori henüz geliştirilememiştir. Dolayısıyla, insanlarin rengi nasıl idrak ettikleri ve rengin tam manasıyla nasıl temsil edilebileceği sorularına henüz cevap bulunamamıştır. Ayrıca, renkli resim işleme uygulamalarında kullanılan renk modelleri rengi tam olarak temsil edemediklerinden uygulamaya bağlı kalmaktadırlar. Renk temsil modellerinin uygulamaya bağımlı olması büyük bir dezavantajdır çünkü o uygulama için en uygun renk temsil modeli ancak bir sürü deneyler neticesinde bulunmaktadır. Renk temsilinin bütün zorluklarına rağmen genel amaçlı bir renk modeline kesinlikle ihtiyaç vardır. Bütün renk modellerinin iyi ve kötü tarafları var. Bu modeller, eksikliklerinin giderilmesiyle birbirlerine yardımcı olabilirler. Bu birleştirici yaklaşım renk bilimi'nde gelecekte yapılacak olan araştırmalara yön verebilir ve renk bilimi'ne yeni bilgiler kazandırabilir. Renk biliminde yapılan araştırmaların renk teorilerini geliştirmeye yönelmesi ve önemli fonemenlerin açıklanabilmesi, renkli resim işlemede ve yapay-görme çalışmalarında yeni gelişmeler kaydetmek için önem teşkil etmektedir.

Değişik aydınlatma ve renk şartları altında çalışan robotlar için geliştirilen görme sistemlerinde Renk Sabitliği özelliği çok önemlidir [16] [17] [18]. İnsan görme sisteminde bir dereceye kadar olan bu özellik değişik ortamlarda sağlıklı bir "renk görme" için makinalara kesinlikle uygulanmalıdır.

Conceptual renk temsil modellerinin diğer modellere, renkli resim işleme açısından, büyük bir üstünlüğünün olduğu açıkca görülmektedir. Artık bu modellerin tabiatına tamamiyle uygun renkli resim işleme algoritmaları geliştirilmelidir.

Tesekkur

Yazarlar, Erciyes Üniversitesi'ne bu çalışma için Y.Samast'a sağladığı maddi ve manevi yardımlarından dolayı teşekkür eder.

7 - KAYNAKLAR

1. Wyszecki,G. ve Stiles,W.S., *Color Science: Concepts and Methods, Qualitative Data and Formulae*, John Wiley, New York, (1982).
2. Ballard,D.H. ve Brown,C.M., *Computer Vision*, Prentice Hall, Englewood Cliffs, NJ, (1982).
3. Gershon,R., *Aspects of Perception and Computation in Color Vision*, Comput. Vision Graphics and Image Processing, 32:244-277, (1985).
4. Healey,G. ve Binford,T.O, *The Role and Use of Color in A General Vision System*, Proc. of DARPA Image Understanding Workshop, USC, Los Angeles, CA, pp.599-613, Feb. 23-25, (1987).
5. Samast,Y., *Renkli Resim Islemenin Bazi Teorik ve Pratik Ozellikleri*, Teknik Arastırma Raporu, Erciyes Univ., Mühendislik Fak., Kayseri, Türkiye, Ocak(1992).
6. Hurvich,L.M ve Jameson,D., *An Opponent Process Theory of Color Vision*, Psychological Review, 64(6):384-404, (1957).
7. Land,E.H., *Recent Advances in Retinex Theory*, Vision Research, 26:7-22, (1986).
8. Kender,J.R., *Saturation, Hue and Normalized Color: Calculation, Digitization Effects and Use*, Technical Report, Dept. of Computer Science, Carnegie-Mellon Univ., Pittsbourgh, PA, (1976).
9. Baldwin,L., *Color Considerations*, Byte, Sept:227-246, (1984).
10. Nazif,A., *A Survey of Color, Boundary Information and Texture as Features for Low Level Image Processing*, Research Report No: 78-7R, McGill Univ., Electrical Eng. Dept., Montreal, Canada, April (1978).
11. Pham,D.T. ve Aslantaş,V., *Otomatik Odaklama*, Birinci Türk Yapay Zeka ve Sinir Agları Sempozyumu, Bilkent Univ., Ankara, Türkiye, 25-26 Haziran, (1992).
12. Aslantaş,V., *Aktif Gorme'de Otomatik Odaklama*, Teknik Araştırma Raporu, Erciyes Univ., Mühendislik Fak., Kayseri, Türkiye, (1992).
13. Swain,M.J. ve Ballard,D.H., *Color Indexing*, Int. Journal of Computer Vision, 7:1, pp.11-32, (1991).
14. Brockelbank,D. ve Yang,Y.H, *An Experimental Investigation in the Use of Color in Computational Stereopsis*, IEEE Trans. on Systems, Man and Cybernetics, V.16, No.6:1365-83, (1989).
15. Poggio,T ve diğerleri, *The MIT Vision Machine*, Proc. of DARPA Image Understanding Workshop, Cambridge, MA, 1:177-198, (1988).
16. Harrel,R.C., Slaughter,D.C. ve Adsit,P.D., *A Fruit Tracking System for Robotic Harvesting*, Machine Vision and Applications, 2:69-80, (1989).
17. Crisman,J.D. ve Thorpe,C.E., *UNSCARF: A Color Vision System for the Detection of Unstructured Roads*, Proc. of the IEEE Int. Conf. on Robotics and Automation, 3:2496-2501, Sacramento, CA, (1991)
18. Zheng,J.Y., Barth,M. ve Tsuji,S., *Autonomous Landmark Selection for Route Recognition by A Mobile Robot*, Proc. of the IEEE Int. Conf. on Robotics and Automation, 3:2004-09, Sacramento, CA, (1991).

Otomatik Odaklama

D.T. Pham ve V. Aslantaş
Intelligent System Laboratory
School of Electrical, Electronic, and Systems Engineering
University of Wales P.O. Box 904 Cardiff CF1 3YH, UK

Özet

Aktif Görme'nin bir tekniği olan Otomatik Odaklama, üç boyutlu (3B) bir çevredeki nesnelerin mesafesini hesaplamakta kullanılabilir.

Bir genel-maksatlı servo-kontrollü video kameranın elle seçilmiş hedefler üzerinde otomatik odaklamasında ortaya çıkan iki problem vardır:

1) Mesafe bilgisi olmaksızın, bir nesnenin noktasının net görüntüsünü sağlayacak odak-motor pozisyonu en iyi nasıl tespit edilir?

2) Net bir görüntü elde edildiğinde, nesne üzerindeki bir noktanın mesafesi nasıl ölçülebilir?

Birinci problemi çözmek için bir *"netlik"* kıstası formülüze edilmelidir ve kıstas fonksiyonunun modunun yerini tayin etmek için bir arama-tekniğine ihtiyaç vardır.

Net görüntü ve mercek parametrelerinin değerleri elde edildiğinde optik geometri kanunları kullanılarak ikinci problem çözülür.

Bu makalede, yukarıda bahsedilen problemlere çözümler açıklanacaktır.

Automatic Focussing

Abstract

Automatic Focussing is a technique of Active Vision which can be employed to compute the depth of objects in a three-dimensional (3D) scene.

There are two specific problems in automatically focussing a general-purpose servo-controlled video camera on manually selected targets:

1) Without having any information about depth, how to best determine the focus motor position providing the sharpest image of an object point.

2) When a sharp image has been captured, how to obtain the depth of the given object point be measured.

To solve the first problem, a *"sharpness"* criterion must be formulated and a search-technique to locate the mode of the criterion function is needed.

The second problem is solved by applying the laws of geometrical optics when the sharp image and the values of the lens parameters have been obtained.

In this paper, solutions to these problems are explained.

1. GİRİŞ

Net görüntüler bulanık görüntülerden çok daha fazla bilgi ihtiva ederler. İnsanlar ve bir çok omurgalı hayvanlar, görüntünün kalitesini artırmak için hızlı ve içgüdüsel olarak otomatik odaklama yaparlar. Bu sebeple elde edilen görüntüdeki bilgide artırılmış olur. Otomatik odaklamanın gayeside bir robota bu kâbiliyeti vermektir. Fizyolojideki intibak (accommodation), uygun odak, netlik ve yüksek frekansın varlığı otomatik odaklama ile eş anlamdadır ve gözdeki merceğin hareketiyle gözün odaklanması manasına tekabül eder.

Odağı servo-kontrollü bir standart CCD kameranın otomatik odaklanmasında ortaya iki önemli problem çıkmaktadır.

1) Mesafe bilgisi olmaksızın, bir nesnenin noktasının net görüntüsünü sağlayacak odak motorunun pozisyonu en iyi nasıl tespit edilir ?

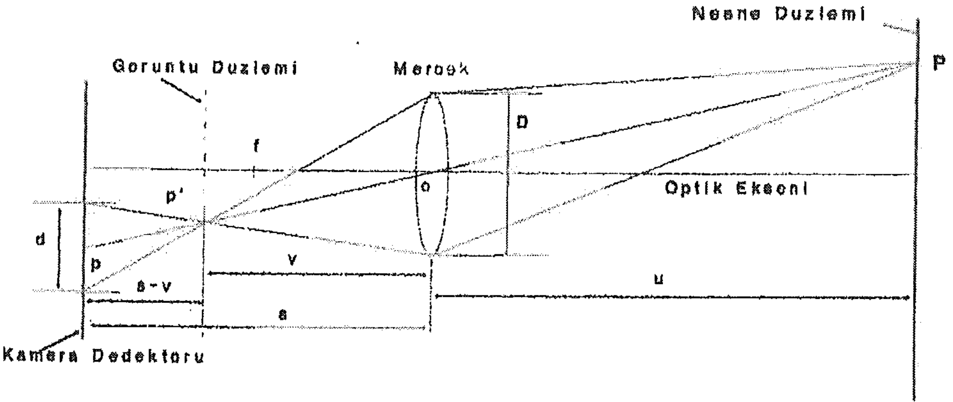
2) Net bir görüntü elde edildiğinde, nesne üzerindeki bir noktanın mesafesi nasıl ölçülebilir ?

Şekil.1'den de görülebileceği gibi nesne üzerindeki bir noktanın net görüntüsü:

1) Nesne düzlemine göre nesneyi hareket ettirerek [1];

2) Merceği hareket ettirerek [2][3][4][5];

3) Görüntü düzlemine göre kameranın yerini değiştirerek elde edilebilir [6].



Şekil 1. Kamera geometrisi.

Birinci problemi çözmek ve bu işlemi otomatik hale getirmek için bir metrik *"netlik"* kıstasına ihtiyaç vardır. Bu kıstas, mesafesi ölçülmesi istenen görüntünün netlik değerini elde etmek için bilgisayar tarafından kullanılır. Bilgisayar, bir arama-tekniği kullanarak bu kıstas fonksiyonun maksimum veya minimum değeri elde edilinceye kadar görüntü mesafesini değiştirir.

Bir nesnenin net görüntüsü elde edildikten sonra optik geometri kullanılarak ikinci problem kolayca çözülür.

Sâbit iki kamera kullanarak (stereo) ve bir (veya birden fazla) kamerayı hareket ettirerek mesafenin elde edilmesi metodlarında ortaya çıkan resim karşılaştırma problemini, otomatik odaklama ile mesafenin hesaplanmasında çözmeye ihtiyaç yoktur.

2. NETLİK KISTASLARI

Bu bölümde, odağın kalitesini ölçmek için kullanılan çeşitli kıstas fonksiyonları açıklanacaktır. Yukarıda da bahsedildiği gibi görüntüdeki bir bölgenin odağının netliğini otomatik bir şekilde hesaplamak için bir *"netlik"* kıstası formülüze edilmelidir. Kıstaslar, görüntüdeki yüksek frekans değişimlerine kesinlikle cevap vermelidir. İdealde, görüntü üzerindeki işlem yapılan bölge mükemmel odaklanınca, kıstas kesinlikle maksimum veya minimum değere sahip olmalıdır. Genelde gaye , değişik görüntüler üzerinde dengeli ve sağlam bir davranış gösteren bir kıstas fonksiyonu bulmaktır. Böyle bir yaklaşım, gelişen otomatik odaklama sistemlerinin bütün ortamlarda çalışabilmesi için önemlidir. Bu bölümde, I(x,y) görüntünün intensitisini gösterir. Otomatik odaklamada kullanılan bazı kıstas fonksiyonları aşağıda verilmiştir.

2.1. Yüksek-Geçiren Filtre

$$\max \sum_{x} \sum_{y} | L_M(X,Y) |, \qquad L_M(X,Y) \geq T \quad \text{için} \qquad (1)$$

$$L_M = (1/6) \begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix} \qquad (2)$$

L_M Laplas denklemini ($\nabla^2 I = \dfrac{\partial^2 I}{\partial x^2} + \dfrac{\partial^2 I}{\partial y^2}$) hesaplamak için kullanılır.

2.2. Histogram Entropy [7][5]

$$E = - \sum_{I=0} P(I)\ln(P(I)), \qquad P(I) \neq 0 \quad \text{için} \qquad (3)$$

2.3. Gri-Ton Değişimi

$$\max \sigma^2 = (1/N^2) \sum_{x=1}^{N} \sum_{y=1}^{N} (I(x,y)-\bar{I})^2 \qquad (4)$$

$$\bar{I} = (1/N^2) \sum_{x=1}^{N} \sum_{y=1}^{N} I(x,y) \qquad (5)$$

2.4. Toplam-Modül-Farkı [7]

$$\max TMF_x = \sum_{x=1}^{N} \sum_{y=1}^{N} | I(x,y) - I(x,y-1) | \qquad (6)$$

$$\max TMF_y = \sum_{x=1}^{N} \sum_{y=1}^{N} | I(x,y) - I(x+1,y) | \qquad (7)$$

$$\max TMF = TMF_x + TMF_y \qquad (8)$$

2.5. Toplam-Geliştirilmiş-Laplas (TGL) [1]

$$\max = \sum_{x=i-N}^{i+N} \sum_{y=j-N}^{j+N} ML(x,y), \qquad ML(x,y) \neq T_1 \quad \text{için} \qquad (9)$$

$$\begin{aligned} ML(x,y) = {} & |2I(x,y) - I(x-1,y) - I(x+1,y)| \\ & + |2I(x,y) - I(x,y-1) - I(x,y+1)| \end{aligned} \qquad (10)$$

2.6. Tenengrad [6]

$$\text{Max} \sum_{x} \sum_{y} S(p)^2 , \qquad S(p) \geq T \text{ için} \qquad (11)$$

$$S(p) = (i_x + i_y)^{1/2} \qquad (12)$$

2.7. Fourier Transform [3]

$$\text{Max } F(f_x, f_y), \qquad (f_x^2 + f_y^2)^{1/2} \geq T \text{ için} \qquad (13)$$

2.8. (Discrete)-Cosinüs Transform [8]

$$\text{Max} \sum_{u=u_1}^{u_2} | C(u,z) | \qquad (0 < u < N - 1) \text{ için} \qquad (14)$$

Yukarıda bahsedilen *"netlik"* kıstas fonksiyonlarının her biri az veya çok, doğrudan görüntüdeki yüksek frekans bileşenlerini hesaplar.

Yapılan çalışmalarda [4][5] Tenengrad kıstasının aşağıdaki sebeplerden dolayı üstünlüğü kabul edilmiştir.

1. Gürültüden fazla etkilenmez.
2. Resim içerisinde genel uygulanabilirliğe sahiptir.
3. Maksimum noktanın her iki yanında monotonik bir azalmaya sahiptir.
4. Hesaplanması oldukça kolaydır.
5. Çeşitli resimler üzerinde ve ebatları farklı pencerelerde kullanılabilir.

Belirli bir intensiti değişimine sahip nesnelerin net görüntülerinin elde edilmesinde TGL kıstası daha uygun bulunmuştur [1].

Fourier ve DC Transform kıstas fonksiyonu kullanılarak daha hassas sonuçlar elde edilir. Fakat Bu transformu yapacak elektronik donanımlar olmadan her iki kıstasında hesaplanması zor ve zaman alıcıdır.

Resimdeki renk bilgisini kullanabilecek bir kıstas fonksiyonu bulunabilirse daha iyi sonuçlar elde edilebilir [9]. Kıstas fonksiyonları hakkında daha fazla bilgi, [10],[4],[11] ve [5] numaralı kaynaklardan da elde edilebilir.

3. KISTAS FONKSİYONUNUN UYGULANMASI VE GELIŞTİRİLMESİ

3.1. Gürültünün Azaltılması için Ön İşlemler

Bir çok odaklama algoritması, çevre ile onun görüntüsü arasındaki ilişkinin bilinebileceğinin mümkün olduğunu kabul eder, fakat gürültünün tesiri ile görüntü daima bozulmaktadır. Bu sebeple kıstas fonksiyonlarının değerleri zamanla değişir. Bazı gürültüler, çalıştırılan özel elemanlara gerekli olan dijitasyon işlemlerinden dolayı ortaya çıkar. Bazıları sistemin çevresindeki diğer bilgisayar ve kabloların tesirinden meydana gelir. Diğer bir gürültü sebebi de çevreden yansıyan ışıkların her an değişmesidir Yukarıdaki sebeplerden dolayı bir çok kıstas fonksiyonu ilk fark veya parçayı hesaplar ve bundan dolayı gürültünün tesiri artırılmış olur. Bir kıstas fonksiyonunun maksimumunu veya minimumunu bulmak için bu gürültülerin tesirinin azaltılması gerekir.

Zamana bağlı ortalama alınarak yapılan gürültü azaltma metodlarında aynı çevrenin birden fazla resmi çekilir. Resimlerdeki aynı yere sahip her noktanın ortalaması alınır ve aynı yere yerleştirilir (Denklem 15).

$$I(x,y) = \frac{1}{N} \sum_{i=1}^{N} I_i(x,y) \qquad (15)$$

Resim sayısı (N) artırılarak gürültü daha da azaltılır. Bu işlem görüntüde zamanla meydana gelen gürültüleri oldukça azaltır. Fakat istatistiki karar verme usülleri kullanılarak daha iyi şonuçlar elde edilebilir.

Diğer bir usûl, her mercek noktasına tekabül eden resmi çekmek ve gürültülü sonuca bir eğri uydurmaktır. Bu usülde, odak mesafesinin her noktasında kıstas fonksiyonu hesaplanacağından net görüntüyü elde etmek oldukça yavaştır.

Tenenbaum [6] ve Schlag ve diğerleri [5] gürültüden meydana gelen zayıf kenar cevaplarının ortaya çıkmasına engel olmak için Tenengrad fonksiyonunun değerlerine eşik değer keşfettiler. Nesneye ait bir özellik, gürültüden değilde iyi odaklanamamasından dolayı zayıf cevap vereceğinden eşik değeri çok dikkatli seçilmelidir. Tenenbaum [6], eşik değerini adaptif parametre olarak, son odak mesafesinden hesaplanan maksimum değerin %75'i olacak şekilde kabul etti. Schlag ve diğerleri [5], eşik değerini bir görüntüdeki gürültünün Rayleigh dağılımının standart sapmasının bir çarpanı olarak seçti. Bu usülün zorluğu eşik değerinin keyfi olarak seçilmesidir.

Gürültüyü azaltmakta en çok kullanılan teknik uzaya bağlı ortalamadır. Dört ve sekiz noktalık olmak üzere kullanılan iki teknik vardır. Şekil 2, uzaya bağlı ortalama denklemleri (Denklem 16, Denklem 17) için gerekli nokta haritasını gösterir.

KB	K	KD
B	X	D
GB	G	GD

Sekil 2. Uzaya bağlı ortalama algoritmaları için nokta haritası.

$$X = \frac{K + D + G + B}{4}, \qquad (16)$$

$$X = \frac{K + KD + D + GD + G + GB + B + KB}{8} \qquad (17)$$

Sonuç olarak, sistem ve çevrenin hareketsiz olduğu durumlarda gürültüyü azaltmak için zamana bağlı ortalama, uzaya bağlı ortalamaya tercih edilir. Çünkü zamana bağlı ortalamada uzaya bağlı ortalamada ortaya çıkan görüntüyü bulanıklaştırma yoktur. Otomatik odaklamada görüntüyü bulanıklaştıran teknikler kullanılmamalıdır.

3.2 Pencere Seçimi

Kenarların gradientinin hesaplanmasında gürültünün büyük tesirinin olmasından dolayı içerisinde görüntü özellikleri bulunan bir pencere seçmek mecburidir. Her hangi bir özelliği bulunmayan (intensiti değişimi olmayan) nesnelerin veya boş uzayın görüntülerini odaklamaya çalışmak kıstas fonksiyonunda bir değişim yapmayacagı için herhangi bir sonuç elde edilemez. Ve eğer bir pencere farklı mesafelerdeki nesneleri ihtiva ediyorsa, kıstas fonksiyonu birden fazla tepeye sahip olacaktır. Bu penceredeki noktaların mesafeleri hesaplanamayacağı için pencere aynı mesafelerde bulunan noktaları ihtiva etmelidir.

Pencere seçiminde bir diğer önemli problem de pencerenin sahip olması gereken ebatlara karar verilmesidir. Büyük ebatlı pencerelerde kıstas foksiyonları gürültüden fazla etkilenmeyeceği için daha iyi sonuçlar elde edilir. Fakat büyük pencereler fazla nokta ihtiva edeceğinden dolayı nesnelerin mesafelerinin hesaplanmasında fazla hassas sonuçlar elde edilemez. Çünkü pencere içerisinde farklı mesafelere sahip noktalar bulunabilir. Kıstasların hesaplanmasında küçük ebatlı pencerelerin kullanılması hesaplama zamanını azaltır ve mesafe ölçümünde hassas sonuçlar elde edilir. Fakat küçük pencerelerde kıstas fonksiyonu gürültüden fazla etkilenir. Yapılan çalışmalarda, ebatları 7X7 den 20X20 noktaya kadar değişen aralıktaki pencerelerde daha iyi sonuçlar elde edilmiştir.

İntensiti degişiminin fazla olduğu bölgelerde küçük, az olduğu bölgelerde büyük ebatlı pencereler kullanılmalıdır. Diğer mesafe ölçme teknikleri veya renk kullanılarak pencerenin yeri ve ebatları otomatik olarak hesaplanabilir [1][5][9].

3.3 Görüntüdeki Değişimin Telafisi

Pratikte odaklama mesafesi değişince, merceğin görüntüyü büyültme oranıda değişir. Bundan dolayı görüntüyü meydana getiren çevredeki noktaların kamera dedektörü üzerindeki koordinatları da değişir. Kıstas fonksiyonunun hassasiyetle hesaplanabilmesi için görüntüye ait özelliğin pencere içerisinde kalması gerekmektedir. Bu problemin çözülmesinde aşağıda bahsedilen dört usûlden biri kullanılabilir.

1. Değişen odak pozisyonuna karşılık merceğin odak uzunluğu değiştirilir.
2. Yazılımda, görüntüde meydana gelen değişim hesaplanarak her odak pozisyonuna tekabül eden görüntü pozisyonu elde edilir.
3. Değişen mercek pozisyonuna karşılık pencere ebatları değiştirilir.
4. Büyük ebatlı pencereler kullanılabilir.

4. KISTAS FONKSİYONUNUN MAKSİMUMUNUN VEYA MİNİMUMUNUN ELDE EDİLMESİ

Kıstas fonksiyonunun hesaplanmasının zaman alması sebebiyle, iterasyon sayısının azaltılması önemlidir. Bu sebeple her mercek pozisyonuna tekabül eden kıstas fonksiyonunu hesaplamak uygun bir çözüm değildir. Bu bölümde, bir nesne noktasının net görüntüsünü elde etmek için bazı optimum araştırma teknikleri tarif edilecektir. Kabul edelimki, kullanılan kıstas fonksiyonu, net görüntünün elde edilmesinde maksimum veya minimum ve bu noktanın her iki tarafında da monoton bir şekilde azalan değerlere sahiptir. Maksimum/minimum nokta, araştırma uzayının herhangi bir yerinde bulunabilir.

4.1 Tepe-Tırmanma

Bu teknikte, herhangi bir mercek pozisyonunda kıstas fonksiyonu hesaplanır. Merceğin yeri değiştirilir ve kıstas fonksiyonu tekrar hesaplanır. Eger kıstas değeri azalmışsa mercek yanlış yönde hareket ettirilmiştir. Tersi durumda ise mercek doğru yönde hareket ettirilmiştir. Bu usûl, kıstas fonksiyonunun şeklinin elde edilmesinde kullanılabilirsede sadece merceğin hangi yönde hareket ettirilmesine karar vereceğinden ve adım büyüklüğünün keyfi olarak seçilmesinden dolayı pek tercih edilmez.

4.2 Fibonacci

Bu teknikte, eldeki araştırma sahasının en küçük değeri elde edilene kadar saha daraltılır. Başlangıç araştırma aralığı için kıstas fonksiyonu Fibonacci dizisi olarak seçilen iki noktada (X_1,X_2) hesaplanır. Eğer başlangıç araştırma aralığı [a,b], a<X_1<X_2<b ve kıstas(X_1) kıstas(X_2) den büyükse bir sonraki araştırma aralığı; başlangıç aralığından, [X_2,b] aralığı çıkarılarak [a,X_1] olarak elde edilir. Diğer her aralık için kıstas fonksiyonu Fibonacci dizisi tarafından karar verilen iki noktada hesaplanır ve elde edilen değerler bir sonraki aralığa karar vermekte kullanılır. Bu teknik, 15 iterasyon sonucunda kıstas fonksiyonunun maksimumunu veya minimumunu hassas bir şekilde tayin eder. Fibonacci teknikte odak motorunun yönünün sık sık değiştirilmesine ihtiyaç duyulur. Yüksek kaliteli resim çekmek için merceğin sabit olması gerekir. Bu teknikte hızlanma veya yavaşlama problemleri veya histerisiz tesirleri meydana gelmeyeceği için kaliteli görüntü çekmek mümkündür.

5. MESAFENİN HESAPLANMASI

Şekil-1'den mercek kanunu (Denklem (18) elde edilir.

$$U^{-1} + V^{-1} + f^{-1} = \varepsilon \qquad (18)$$

Net görüntü sağlanınca (ε=0), mercek denkleminin çözümünden, Denklem (19) elde edilir.

$$U = \frac{Vf}{V - f} \qquad (19)$$

Denklem 19 dan, nesne mesafesi yalnızca görüntü mesafesi ve odak uzunluğu kullanılarak hesaplanır. Yalnız V ile odağın motor pozisyonu arasındaki bağıntının deneylerle elde edilmesi gerekmektedir. Görüntü üzerindeki bir noktanın mesafesinin hesaplanmasında aşağıdaki sıra izlenir.

1. Merceğin pozisyonu değiştirilerek kıstas fonksiyonunun maksimum değeri ölçülür.
2. Odak motorunun pozisyonu okunarak V elde edilir.
3. Denklem (19) hesaplanır.

6. SONUÇ

Bu makalede, aktiv görmenin bir tekniği olan otomatik odaklama kullanılarak mesafenin nasıl hesaplanabileceği anlatıldı. Bu teknikte ortaya çıkan iki probleme çözümler sunuldu. Birinci problemin çözümünde, Tenengrad kıstas fonksiyonu ve bu fonksiyonun maksimum noktasını bulmak için de Fibonacci araştırma tekniği kullanılırsa daha iyi sonuçların elde edilmesi mümkündür. İkinci problem de optik kanunların kullanılmasıyla çözülür.

Sâbit iki kamera kullanarak (stereo) ve bir (veya birden fazla) hareketli kamera kullanarak mesafe ölçme metodlarında ortaya çıkan resimleri karşılaştırma problemi otomatik odaklama tekniklerinde yoktur. Bu tekniğin dezavantajı ise mesafenin nokta nokta veya görüntüye göre küçük ebatlı pencerelerde hesaplanmasından dolayı yavaş olmasıdır. Fakat hesabı kolay ve hızlı kıstas fonksiyonları ve bu kıstas fonksiyonlarının maksimum veya minimum noktasını daha az döngüyle hesaplayabilecek bir araştırma tekniği bulunabilirse daha hızlı sonuçlar elde etmek mümkündür.

Teşekkür

Yazarlar, bu çalışmada Veysel Aslantaş'a maddi ve manevi destekte bulunan Erciyes Üniversitesi'ne teşekkür eder.

7. KAYNAKLAR

1. Nayar, S.K., ve Y. Nakagawa, *"Shape from Focus: An Effective Approach for Rough Surfaces"*, Proc. IEEE Int. Conf. Robotics and Automation, 13-18 Mayis 1990, Cincinnati, Ohio, 218-225.
2. Abbott, A.L., *"Dynamic Integration of Depth Cues for Surface Reconstruction from Stereo Images"*, Doktora Tezi, University of Illinois, Urbana-Champaing,IL., 1990.
3. Horn, B.K.P., *"Focusing"*, MIT Artificial Intelligence Laboratory MEMO No.160 May 1968.
4. Krotkov, E.P., *"Active Computer Vision by Cooperative Focus and Stereo"*, Springer-Verlag, New York, 1989.
5. Schlag, J.F., A.C. Sanderson, C.P. Neumann, ve F.C. Wimberly, *"Implementation of Automatic Focusing Algorithms for Computer Vision System with Camera Control"*, Dept. of Electrical Eng., Carnegie-Mellon Üniv., CMU-RI-TR-83-14, August 1983.
6. Tenenbaum, J.M., *"Accommodation in Computer Vision"*, Doktora Tezi, Stanford Üniv. 1970.
7. Jarvis, R.A., *"Focus Optimisation Criteria for Computer Image Processing"*, MICROSCOPE, 24(2), 163-180, 1976.
8. Charfi, M., A. Nyeck and A. Tosser, *"Focusing Criterion"*,

Electronics Letters, 27(14), 1233-1235, 1991.

9. Samast, Y., *"Renkli resim İşlemenin Bazi Teorik ve Pratik Özellikleri"*, Erciyes Üniversitesi Teknik Araştırma Raporu, Kayseri, Türkiye, 1992.

10. Aslantaş, V., *"Aktif Görme'de Otomatik Odaklama"*, Teknik Araştırma Raporu, Kayseri, Türkiye, 1992.

11. Ligthart, G. ve F.C.A. Groen, *"A Comparison of Different Autofocus Algorithms"*, Proceedings of the Sixth Int. Conf. Pattern Recognition, Munich 597-600, Oct 1982.

12. Pham, D.T. ve Y. Samast, *"Renkli Resim İşlemenin Teorik ve Pratik Özellikleri"*, Birinci Turk Yapay Zeka ve Sinir Ağları Semp., Ankara, Türkiye, 1992.

Executing Prolog Programs with Dataflow Approach

Osman Başbuğoğlu[1] and Işık Aybay[2]

[1,2] Electrical and Electronics Eng.Dept.
Middle East Technical University, Ankara 06531 Turkiye.

e-mail : a09938@trmetu.bitnet aybay@trmetua.bitnet

Abstract

A Dataflow Graph Execution Model, PDGEM is developed for parallel execution of Prolog programs. Prolog programs are compiled into dataflow graphs and these graphs are executed using the dataflow concept. This model implements OR-parallelism, restricted AND-parallelism and stream parallelism. The target architecture is a special coarse grain dataflow architecture which will implement the eleven primitive operations of the model in hardware. The simulation results reveal that the model can exploit parallelism efficiently when the number of processing units are increased depending on the application.

1. INTRODUCTION

In Prolog, parallelism arises mainly in two forms: OR-parallelism and AND-parallelism. As a property of Prolog, there may exist different clauses with the same clause name. OR-parallel execution is the activation of all such clauses in parallel when a call occurs. AND-parallelism is the parallel execution of and-terms of a Prolog rule body.
A Prolog program consists of facts and rules. A simple Prolog program is given below to clarify the terminology used in this text.

```
r(a).
p(a).
p(b).
p(X):-q(X),r(X).
```

In this program, r and p are clause names. The last p is a rule and its body consists of two and-terms q(X) and r(X). r(a), p(a), p(b) are facts (i.e. rules with empty clause bodies).

The dataflow concept is used in some Prolog execution models [1,2,3,4)], mostly to implement stream parallelism. The conventional solution tree approach dominates the execution policies of these models.

In this study, a graph execution based model is developed. Prolog programs are compiled into dataflow graphs. Graphs which consist of nodes (primitive operations of the model) and arcs (data links between nodes) are the executable modules of a Prolog program. An execution graph contains some topographical information, the connection of nodes, and the instruction types to be executed in nodes (operations to be performed on tokens).

There are eleven primitive operations. These operations are COD (Copy and Distribute), MRS (Merge and Select), SWU (Switch-up), MRO (Ordinary Merge), CHD (Check and Distribute), UFA (Unify Fact), MRR (Merge Rule or Intermediate Graph), ACT (Activate Intermediate Graph), SWD (Switch-down), ACP (Activate Predicate) and URH (Unify Rule Head).

Compilation of Prolog programs to obtain dataflow execution graphs is a straight forward process. Moreover, no modification of the Prolog syntax is required and the exploitation of the parallelism is completely transparant to the user.

In PDGEM, all solutions are searched unconditionally, using an eager evaluation mechanism. As a consequence, an additional effort is not required for backtracking.

The model exploits OR-parallelism restricted AND-parallelism 5) and stream parallelism. This execution scheme allows parallel execution with a very small run time overhead totally eliminating the risk of binding conflicts on shared varibles. Also the cross-product operation after AND-parallel execution is avoided.

2. DESCRIPTION OF THE MODEL

2.1 The dataflow concept

The dataflow execution model is a model which eliminates the order of execution of a sequential program imposed by the lexicial ordering of its instructions. The only execution order in this model is the one which depends on the availability of the data of instructions. A dataflow program is described as a directed graph. The nodes of the graph are the instructions of the program and the arcs are the connections carrying the date from an instruction to another (the result of the execution of an instruction is used by another as input). The data carried on arcs are called tokens. The firing (execution) of an instruction requires that there is a token on each input arc (all input data is ready). Once a node is fired, tokens on the input arcs are removed, the operation is performed and the result is put on the output arc in the form of a new token.

The way a dataflow graph is executed from the point of view of reentrency defines two different dataflow models: static and dynamic 9,11). In the static execution scheme a graph is activated for a set of token and the next set is not accepted until the current execution is completed. The dynamic scheme (which is also called tagged token model), allows consecutive activations of a graph without having to wait the completion of the previous activation.

If these two approaches are examined in terms of parallelism they exploit, the static model exploits the parallelism in the code by simultaneaous activation of the nodes. The dynamic model in addition to this, exploits another form ofparallelism due to the consecutive activations of the graph (pipelined execution of the instructions). Static dataflow architectures are in general less complex to implement than the tagged token architectures. There are architectures built using both schemes 6,12).

2.2 Overview of the Model

The principal difficulty in exploiting OR-parallelism, the simulataneus activation of all rules and facts of a predicate, is the saturation of the architecture 3) (search space explosion). The problem in exploiting AND-parallelism is the risk of creating binding conflicts on the shared variables of AND-terms and taking the cross-product of the distinct solutions found by AND-terms.

PDGEM is a compromise between an efficient run time policy and parallelism extraction. OR-parallelism requires a run-time support (a feedback system) to avoid the saturation of the machine. To avoid the binding conflicts and to the cross-product operation in AND-parallel execution, the parallel activation of AND-terms of a rule is performed when all variables of the AND-terms are ground (constants, or variables bound to constants). However, when the parallel activation is not possible, AND-terms are activated in a pipelined manner exploiting stream parallelism3) .

PDGEM is a tagged token model. This allows the expoition of two kinds of parallelism introduced in section 2.1. The subgraphs of a program are extracted at compile time and they are static. The same subgraph can be used for different invocations.

In PDGEM Prolog programs are compiled into dataflow graphs. A compiled Prolog program is a group of subgraphs. A subgraph is produced for each predicate and each pair of AND-terms of rule bodies. This means that for a three-term rule body there will be two subgraphs: one for first two AND-terms and the second to call the first one and the third AND-term. Subgraphs contain model primitives. These primitives control token flow, token duplication/destruction and unification. PDGEM is not a conventional fine grain dataflow model. The primitives are simple sequential microprograms. The functions of the eleven primitive operations of the model are explained in the following paragraph.

COD (Copy and Distribute) instruction produces one copy of the input token on each output arc. This primitive initiates the OR-parallel execution. MRS (Merge and Select) instruction terminates OR-parallel execution by collecting all the solutions found on parallel branches. UFA (Unify Fact) performs the fact unification of a clause head. Unification modifies the token variables (binding environment) and marks the token as *invalid* if the unification fails. URH (Unify Rule Head) performs the head unification of a clause and activates the subgraph correspondig to the predicate if the unification is successful. CHD (Check and Distribute) makes the run-time check on AND-terms to decide wether to do AND-parallel execution. If parallel execution is possible, the incoming token is duplicated and both branches (AND-terms) are activated. MRR (Merge Rule or Intermediate Graph) is the final primitive of a rule body subgraph. If the input tokens belong to a parallel activation and both are marked as*valid* (all unification succeded), outputs one *valid* marked token. If one or both tokens are marked as *invalid*, outputs one *invalid* marked token to the otuput arc. If there is a pipelined activated token on its input then it transfers it if it is a *valid* marked one and cancels (destroys) it otherwise (except the last token of an activation which has to be transfered even if it is an invalid token). ACP (Activate Predicate) activates the subgraph of the corresponding predicate. ACT (Activate Intermediate Graph) activates the subgraph produced for body of a rule when there are more then two AND-terms. SWU, SWD, MRO are token flow control primitives of rule body execution. They check the tag of input token and direct it one of output arcs or takes one token from either input arc and transfers it to its output arc.

The target architecture is a special coarse grain dataflow architecture. The architecture has three main components: processing elements, tag memory/assignement units and token memory.

2.3 Compilation of Prolog Programs into Subgraphs

In this section three rules used in obtaining the dataflow graphs of a Prolog program will be presented. After the compilation of a Prolog program, for each predicate name one subgraph is produced. Then for the body of a rule depending on the number of AND-terms, one or more subgraphs are produced. For facts, no graph is produced, they are treated on the node base.

Rule 1 For each predicate of a program, a subgraph starting with a COD node and terminating with a MRS node, containing one UFA node for each fact of the predicate, and an URH node for each rule of the predicate is prepared.

Rule 2 The subgraph of a rule with two AND-terms starts with a CHD node and finishes with a MRR node. Two ACP nodes for the activation of AND-terms are interconnected by two SWD, one SWU and one MRO nodes.

Rule 3 The third rule explains how to treat a rule body when there are three or more AND-terms. The graph for the first two AND-terms is prepared as explained above. Then for the third AND-term, a new graph which is similar to a graph prepared using Rule-2, but an ACT node is used instead of the first ACP, and this ACT node refers (calls) the subgraph produced for the first two AND-terms. For the fourth and following AND-terms this procedure is repeated recursively.

2.4 Examples

In this section, two examples will be discussed to illustrate OR-parallel and AND-parallel executions of a Prolog program in PDGEM.

Example 1

```
/* OR-Parallel */
p(t,u). /* fact 1 */
p(b,l). /* fact 2 */
p(a,b). /* fact 3 */
p(a,c). /* fact 4 */
p(a,d). /* fact 5 */
```

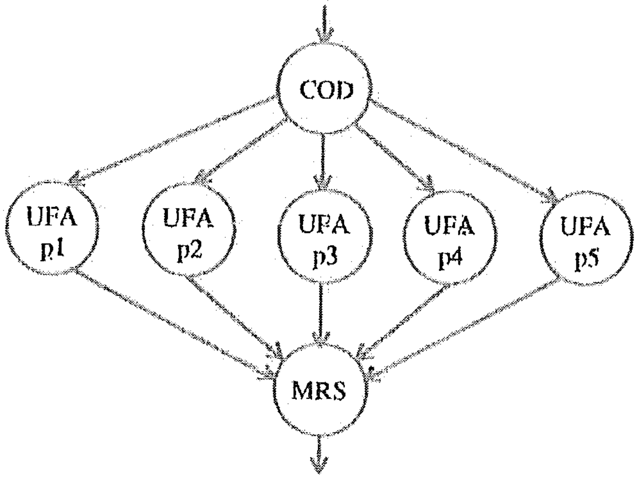


Figure 2.4.1. Subgraph for example 1.

?-p(X,Y).

The activation token prepared for the query, carries the binding environment which contains two unbound variables X and Y. The first node in the graph is a COD node which produces five distinct copies of the activation token (the number of copies may be fewer according to the current load of the machine to avoid its blocking). The copying is made by the processing element involved, in distinct zones of the token memory, and the tags are copied by the assignment unit in the same way. Each UFA node performs fact unification and sends the resulting token to the MRS node. The MRS node outputs five solutions obtained. (If one unification were unsuccessful then the corresponding token would be marked as *invalid* and destroyed by the MRS node. The token would be destroyed by the processing element and its tag would be destroyed by assignment unit).

Example 2

t(a,a).
u(a,b).
p(X,Y,Z):-t(Y,Z),u(Z,X).

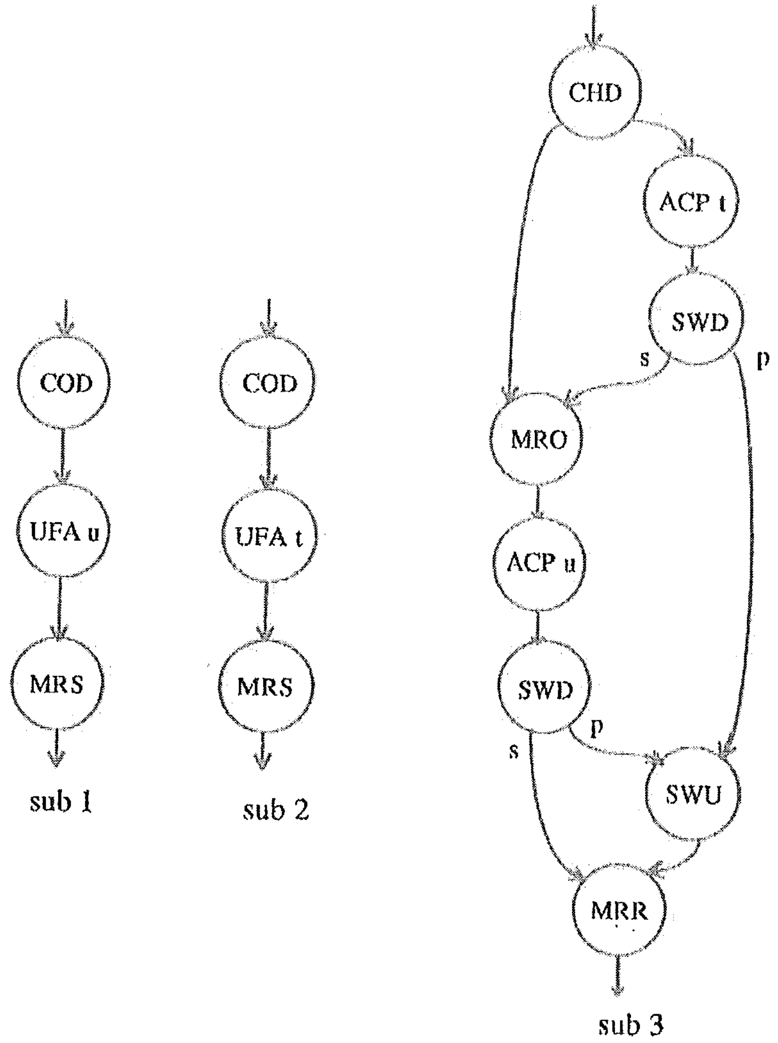


Figure 2.4.2. Subgraphs for example 2.

This program segment will be studied with two different queries.

Query 1

?-p(X,Y,Z). /* query 1 */

For query1 an activation token which contains three unbound variables is prepared and sub_3 is activated. The CHD node checks whether all variables be ground (constant or bound to a constant the parallel execution condition). Since this is not the case here, AND-terms are activated in a pipelined manner.

CHD marks the token as *serial* and sends it to ACP_t node. The ACP_t node activates the sub_1 graph. The COD node of sub_1 transfers the token to UFA_t node and UFA_t node makes the unification. The unification results in the binding of Y and Z to *a*. The MRS node of sub_1 transfers the token to the SWD node of sub_3. Since the token is *serial*,it is sent by SWD node, to MRO via the *s* marked arc. MRO node simply transfers the coming token to its output arc without any condition. ACP_u activates sub_2.

The activation progresses similarly as sub_1. In the unification, X is bound to *b* and the binding of Y is verified. The SWD node transfers the token to MRR node because it is *serial*. The MRR node outputs the bindings of the variables terminating the execution.

Query 2

?-p(b,a,a). /* query 2 */

This time another query which allows the parallel execution of AND-terms will be considered. The activation token for this query contains three ground terms (constants). The parallel execution check at CHD node succeeds. Two distinct copies of the token is prepared. One copy is sent to the MRO node, and the other to ACP_u node. Tokens are copied by the processing element and the tags by the assignment unit. The token going to MRO node is transfered to the ACPt node. Both tokens activate sub_1 and sub_2 graphs independently.

When the activations are over upper and lower SWD nodes switch the coming tokens to p labeled output arcs since tokens are marked *parallel*. When both tokens come to SWU node the SWU checks whether they are *valid*. Since the unification is successful in both sub_1 and sub_2, both tokens are *valid*. The SWU node then transfers just one token to MRR and finally MRR transfers this token to the output unit. If one or both of the tokens coming to SWU node were *invalid*, then SWU would transfer one *invalid* token to MRR.

4. CONCLUSIONS

PDGEM is tested and simulated. The simulation results show that AND-parallel execution is more efficient than the AND pipelined execution (stream parallelism). For stream parallelism to be efficient the number of tokens activating the AND-terms of a rule must be compatible with the number of AND-terms. Theoreticaly, a speed-up factor equal to the number of AND-terms is possible as the number of consecutive activations goes to infinity.

Prolog Dataflow Graph Execution Model is a model which implements full OR-parallelism, restricted AND-parallelism and stream parallelism (AND-pipeline execution) over a dataflow execution mechanism. Although PDGEM does not include all possible cases to exploit AND-parallelism 14), most of the work is done at compile time and the overhead due to run-time decisions are small 6). The use of dataflow execution concepts eliminates the need for process creation and management.

Most of the previous parallel Prolog models require some modification in the syntax of the language like annotations. PDGEM does not require any such modification. The execution and the compilation of subgraphs are straight forward. The run-time overhead of the model is kept low by shifting the graph preparation work to compile time. Run-time checks for restricted AND-parallel execution do not bring much overhead.

If the only and-term execution policy is chosen to be AND-pipelining, the model will become more practical, because the common data kept for the number of tokens at a graph execution level is not needed anymore. In this way, the critical section requirement will be eliminated. Moreover, the dataflow execution graphs will have a simpler form and execution mechanism. The SWU, SWD, CHD and MRO nodes will not be needed. A rule body will be represented as a sequence of ACP nodes (one for each AND-term), followed by a MRS node.

For a practical implementation of this model, some modifications for list and structure manipulation should be done. For this purpose, the I-structures 10) may be considered. I-structures allow a read request to a memory location before a write operation is performed on the same location.

The work explained in 3) introduces parallel and concurrent Prolog execution models which depends on dataflow concept. This parallel Prolog execution model implements sequential and parallel execution of AND-terms and OR-parallel execution. The model requires the specification of the execution mode of AND-terms (as parallel or sequential) in programs with supplementary operators. In PDGEM, sequential and parallel execution is decided at run-time so the language syntax is not needed to be modified. In 3), possible inconsistent solutions due to parallel execution of AND-terms with shared variables are eliminated by consistency check of solutions. In PDGEM, the run-time check to check parallel execution conditions is a cheap one which controls whether the arguments are ground terms or variables bound to ground terms. As a result, even in parallel execution, consistency check of solutions is avoided. 3) is a fine grain dataflow model. Primitives operate on argument basis rather than predicate basis. The overhead due to fine granularity is planned to be solved by implementing the primitives in hardware or firmware. PDGEM however, is a coarse grain dataflow model. Primitives operate on predicates. So the model does not aim to extract parallelism in low level such as unification parallelism. With this approach we avoid the overhead due to fine granularity.

PDGEM has similar characteristics with the model explained in 14). Like 14), PDGEM implements restricted AND-parallel, stream parallel and OR-parallel execution. Tokens in PDGEM are also dynamic tokens to differentiate different activations of the same graph. Tokens carry binding environments of activations. These are called binding environment frames. As cited in 14) ([Li and Martin 86] and [Tseng et al. 88]) PDGEM resembles to [Li and Martin 86] since it compiles Prolog programs into dataflow graphs and does not modify them during execution. However PDGEM 14) diverges from ([Li and Martin 86] and [Tseng et al. 88]) in parallel execution of AND-terms. In these models, parallel execution of independent AND-terms triggers an incremental cartesian product operation. This is not supported in PDGEM. The parallel execution of AND-terms in PDGEM is not optimal 4) but restriction reduces the run-time overhead and uniformizes the model (production of standard execution graphs for rule bodies and predicates (OR-terms)).

5. REFERENCES

1) Wise,M.J., 1986, Prolog Multiprocessors, Prentice-Hall.
2) Ito,N., Shimizu,H. ,Kishi,M., Kuno,E., Rokusawa,K., 1985, Data-flow Based Execution Mechanisms of Parallel and Concurrent Prolog, New Generation Computing, Ohmsha Ltd and Springer-Verlag, 1985, p:15-41.
3) Hasegawa,R. ,Amamiya,M., 1984, Parallel Execution of Logic Programs based on Dataflow Concept, Proceedings of the International Conference on Fifth Generation Computer Systems, 1984, p:507-516.
4) Biswas,P., Tseng,C.C., 1989, Multiring LogDf Arch. and Experiments in Scheduling/Load Balancing, ISCA'89 Workshop on Architectural Support for Declarative Programming Languages, Eliat, Israel, May 1989.
5) DeGroot,D., 1984, Restricted And-Parallelism, Proc. of the International Conference on Fifth Generation Computer Systems, 1984, p:361-368.
6) Arvind, Nikhil,R.S., Execution a Program on the MIT Tagged-Token Dataflow Architecture, Technical Report Laboratory for Computer Science MIT, (An early version of

this report appeared in Proc. of the PARLE Conference, Eindhoven, The Netherlands,(Springer-Verlag LNCS Volume 259), June 1987).

7) O.Basbugoglu, 1989, A Dataflow Graph Execution Model for Prolog, M.S.Thesis, METU, November,1989.

8) Arvind, D.E.Culler, Dataflow Architectures, Annual Reviews in Computer Science, 1986, Vol.1, p:225-253.

9) Veen,A.H., 1986, Dataflow Machine Architecture, ACM Computing Surveys, Vol.18, No.4, December, 1986, p:365-396.

10) Arvind, Nikhil,S.R., Pingali,K.K., I-structures: Data Structures for Parallel Computing, M.I.T. Computation Structures Group Memo 269, February 1987.

11) Dennis, J.B., Models of Data Flow Computation, adaptation from the paper presented at IEEE Computer Society CompCon 1984 for Control Flow and Dataflow Concepts of Distributed Programming, Springer-Verlag 1986.

12) Dennis, J.B., Static Data Flow Computation, adaptation from the paper presented at IEEE Computer Society CompCon 1984 for Control Flow and Dataflow Concepts of Distributed Programming, Springer-Verlag 1986.

13) ConeryJ.S., The AND/OR Process Model for Parallel Execution of Logic Programs, Ph.D. Dissertation, Univ. of California, Irvine, Tech. Report 204, Infornation and Computer Science, 1983.

14) Delgado-Rannauro, S.A. Computational Models of Parallel Logic Languages, ECRC Technical Report CA-46, May 1989, Munchen Germany.

Yapay Sinir Ağları Uygulamaları

Applications of Artificial Neural Networks

A NEURAL NETWORK ARCHITECTURE FOR EMULATING FORWARD DYNAMICS OF A ROBOT MANIPULATOR

Turhan Tunalı [1]   Mehmet Kuntalp [2]   Bahadır Şamlı [1]

1 Dept. of Computer Engineering, Ege University, Bornova, Izmir 35100, Türkiye

2 Dept. of Electronic and Communication Engineering, Dokuz Eylül University, Bornova, Izmir 35100, Türkiye

Abstract

A neural network architecture for emulating forward dynamics of a simulated three degree of freedom revolute joint robot manipulator among a predefined trajectory is developed. The performance is investigated and results are reported. The potential use of the emulator for robust control purposes is discussed.

1. INTRODUCTION

It is well known that the technique of error backpropagation attracted many researchers and incereased the popularity of artificial neural networks. As in most other neural network studies, application of neural networks to control systems involves the use of error backpropagation technique [1,2,3]. In these methods, a feedforward neural network is trained to behave like the plant which is to be controlled. Afterwords, the neurocontroller is trained. In other words, the feedforward neural network acts as an "emulator" and it learns to identify the dynamical characteristics of the plant. The training of the emulator is analogous to "plant identification" in control theory however there are two basic advantages. Firstly, there is no need to provide any information about the model. That is, emulation is done automatically. Secondly, the neural emulator is able to model the nonlinear plants including robot manipulators.

The importance of providing a good emulator for neurocontroller architectures lies under the fact that the neurocontroller cannot be trained by using error backpropagation algorithm without an emulator. The output values that can be measured are the ones of the plant, not of the neurocontroller and the error signal at the output can not propagate backward through the plant which is not a neural network; so the backpropagation algorithm in which the error must propagate backward from the output to the input can not be used. To overcome this problem, the emulator is used in place of plant during the training of neurocontroller.

In this study, a neural network architecture for emulating forward dynamics of a simulated three degree of freedom revolute joint robot manipulator among a predefined trajectory is developed. In the next section, the simulated forward dynamics of the robot arm used in training of the emulator will be given and the trajectory used will be specified. In the third section, details about the training process and the architecture will be discussed. Finally in the last section simulation results about the performance of the emulator will be given and the ability of the developed emulator in operating on different trajectories will be discussed.

2. ROBOT ARM FORWARD DYNAMICS SIMULATION

Using Lagrange-Euler formulation the dynamics of a robot arm can be written as

$$A(\Theta)\ddot{\Theta} + H(\Theta, \dot{\Theta}) + G(\Theta) = \tau \qquad (1)$$

where $\Theta \in R^n$ is the joint angle vector, $\dot{\Theta} \in R^n$ is the joint angular velocity vector, $\ddot{\Theta} \in R^n$ is the joint angular acceleration vector, n is the number of joints, $\tau \in R^n$ is the motor torque vector. $A(\Theta) \in R^n \times R^n$ is the generalized inertia matrix, $H(\Theta, \dot{\Theta}) \in R^n$ is the nonlinear term containing coriolis and centipital factors, $G(\Theta)$ is the term containing gravity factors [4]. The link structures used in this study are circular cylindrical shell, conical shell and a uniform slender rod as given in [4] for a Microbot arm. The values of parameters used are m_1= 3kg,m_2= 2kg , m_3= 1 kg where m_i is the ith joint mass, h= 20 cm e= 30 cm f= 30 cm where, h, e, f are joint lengths respectively, r= 5 cm is the link radius. Θ and $\dot{\Theta}$ are the outputs and τ is the input and n=3.

We have used

$$x(t) = a + bt + ct^2 + dt^3 \qquad (2)$$

to generate desired trajectory in base coordinates which involves going from one point to another in 2.5 sec. The initial and final points are assumed to be 80 cm apart from each other. The inverse kinematics package written for the robot generates $\Theta_d(t)$, $\dot{\Theta}_d(t)$ and $\ddot{\Theta}_d(t)$ which realizes (2) every 10 msec which is the sampling period T. The robot is assumed to be at rest at the initial and final configurations.

We have used computed torque technique [5] to generate the torque values which make robot track the desired trajectory. The robot is assumed to be on the desired trajectory initially. The output data $\Theta(t)$ and $\dot{\Theta}(t)$ is generated by using Runge-Kutta integration procedure of fourth order. As a result we have generated data for 250 sampling intervals which contain $\Theta(kT)$, $\dot{\Theta}(kT)$, $\tau(kT)$,

3. EMULATOR ARCHITECTURE AND TRAINING SCHEME

The neural network is chosen to be a feedforward one which consists of an input layer, two hidden layers and an output layer. The number of processing elements in the output layer is chosen to be six which is equal to the number of outputs of the robot arm. The input layer has nine elements among which three are the input torques and six are the initial states. Figure 1 depicts the inputs and outputs of the neural emulator. In the first hidden layer

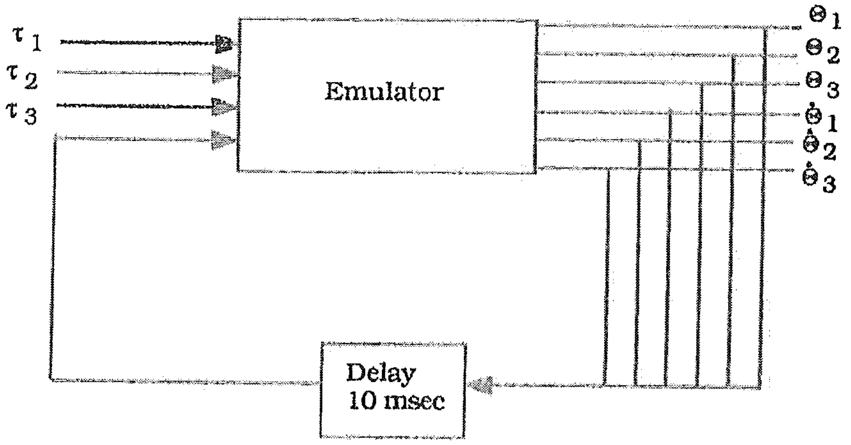


Figure 1: The Inputs and Outputs of the Neural Emulator

27 processing elements are used. The number of processing elements in the second hidden layer is chosen as that of the output layer, namely six. Thus, as a total, emulator network contains 48 processing elements and 441 connections among them.

Error backpropagation algorithm is used to train the emulator. Figure 2 depicts the training scheme of the emulator.

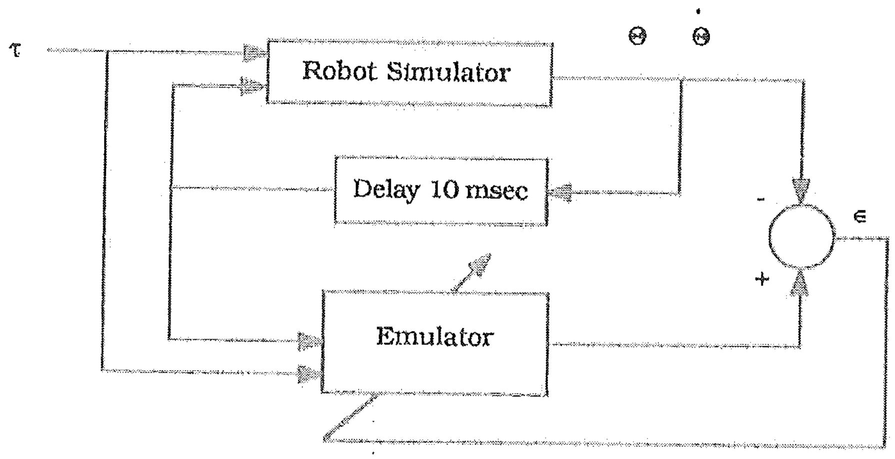


Figure 2: Training Scheme

The environment used is Neural Works Professional II on a PC. The scale factors are chosen to be 0.25, 1.00, 1.00 and 2.50 respectively. All of the transfer functions are chosen to be hyperbolic tangent except the ones used in the input layer for which linear transfer functions are used.

The order of the data generated in section 2 which contains 250 sampling interval input and outputs is randomized to prevent network from diverging. The initial connection weights are also randomized between -0.1 and +0.1 due to same reason. To improve the convergence in training process, the values of the learning coefficients are decreased several times during training. Table 3 shows this variation.

Training up to Learning coefficient	10000	15000	20000	30000	40000	50000	60000	70000
1	0.900	0.750	0.600	0.300	0.225	0.150	0.135	0.105
2	0.600	0.500	0.400	0.200	0.150	0.100	0.090	0.070
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 3: Change in Learning Coefficients

4. PERFORMANCE OF THE EMULATOR AND DISCUSSION

To measure the performance of the emulator, a program is written in C language to evaluate the errors at the outputs. After training the network 70000 times, it has been observed that the rms positional error is reduced to 0.23 radians approximately. It has also been observed that if the robot is initially on desired trajectory and computed torque method is used on simulated dynamics alone, comparable rms error occurs even if T is decreased. This indicates that the rms error measured at the output of the emulator is mainly because of numerical errors of PC environment. Table 4 indicates the mean, rms and absolute extremum errors of joint positions and velocities of the emulator.

	mean error	rms error	absolute extremum error
θ_1	-0.0023	0.0264	0.0575
θ_2	0.0059	0.0267	0.0640
θ_3	0.0027	0.0162	0.0267
$\dot{\theta}_1$	0.0011	0.0316	0.0866
$\dot{\theta}_2$	-0.0072	0.0241	0.0532
$\dot{\theta}_3$	0.0089	0.0174	0.0406

Tablo 4: Mean, rms and absolute extremum errors (in radians and radian/sec)

The next step was to determine how succesfully the network emulates the manipulator on trajectories which are not thaught. In other words, how succesfully the emulator generalizes? The emulator is tested as follows: The trajectory for which the emulator was trained is taken as a reference. Then to determine the generalization range, various trajectories which are x cm apart from the reference trajectory were created and network is tested with these new trajectories. Namely, torque values which correspond to these trajectories are given as input and the outputs of the simulated robot and emulator are compared and rms errors are examined. Then x is increased and the measurements are repeated.

We have concluded that emulator performance decreases when x>1.5 cm. In other words, emulator has learned the trajectories which are within 1.5 cm of the original one. This is good enough to use the emulator for robust control purposes in which the magnitude of error never reaches to these values. Tables 5 indicates the mean, rms and absolute extremum errors measured for trajectories which are 1.5 apart from the original one.

	mean error	rms error	absolute extremum error
θ_1	-0.010	0.057	0.101
θ_2	0.072	0.036	0.110
θ_3	0.098	0.046	0.111
$\dot{\theta}_1$	0.002	0.073	0.178
$\dot{\theta}_2$	-0.006	0.082	0.184
$\dot{\theta}_3$	0.029	0.087	0.215

Table 5: Mean, rms and absolute extremum errors (in radians and radian/sec) for trajectories 1.5 apart from the original one.

REFERENCES :

[1] Psaltis, D., Sideris, A. and Yamamura A. (1988). "A Multilayered Neural Network Controller" IEEE Control Systems Magazine, April.

[2] Kawato, M., Ono, Y., Isobe, M. and Suzuki, R. (1987). "Hierarchical Neural Network Mode for Voluntary Movement with Application to Robotics" Proceedings of the IEEE International Conference on Neural Networks.

[3] Gues, A., Eilbert, J.L. and Kam, M. (1987). "Neural Network Architecture for Control" Proceedings of the IEEE International Conference on Neural Network.

[4] Wolowich, W.A. (1987). "Robotics: Basic Analysis and Design". Holt, Rinehart and Winston 1987.

[5] Lee, C.S.G., Mudge, T.N. and Turney, J.L. (1982). "A Hierarchical Control Structure Using Special Purpose Processors For the Control of Robot Arms" Proc. of the Pattern Recognition and Image Proccessing Conference.

Stability Properties of Artificial Neural Network Based Robotic Controllers *

M. Kemal Cılız
Electrical Engineering Department, Boğaziçi University
Bebek, Istanbul 80815 Turkey

April 24, 1992

Abstract

Robotic manipulator control with *unknown or uncertain* dynamics has been an important research topic in the last decade. Without a parametric model of robot dynamics, learning control techniques are still the most effective methods for repeated trajectory following tasks. In this class of controllers, neurologically inspired algorithms have been gaining much attention in recent years. Although these techniques were shown to work effectively in simulation experiments, coupled and nonlinear nature of parameter update dynamics makes an effective mathematical analysis difficult. This paper investigates the *convergence* properties of an artificial neural network based learning controller. The results obtained reflect the local stability properties of the closed loop nonlinear system dynamics.

1. Introduction

In recent years, Artificial Neural Networks (ANN) have emerged as a viable alternative for various engineering problems of nonlinear nature. Within this new trend, many researchers have attempted to apply neurologically inspired algorithms for manipulator control. The main underlying assumption in these applications is the efficient capability of multi-layer ANNs to approximate multivariable functions. With the increasing interest in this area, IEEE Control Systems Magazine had come up with three issues which covered special sections on neural networks for control systems [1]. The interested researcher can find an exhaustive list of the previous work on the subject in [2].

The present paper discusses the stability properties of an ANN based trajectory following controller. The controller algorithm in question was originally proposed by the author in [3, 2] and successfully tested for trajectory following tasks through simulation experiments. The controller is basically similar to the adaptive inverse dynamics control algorithm of Craig et al. [4].

However instead of using an explicit parametric model, the controller utilizes *generic multi-layer ANNs* to adaptively approximate the manipulator dynamics over a specified region of the state space for a given desired trajectory. This generic neural network structure can be viewed as a nonlinear extension of a deterministic auto-regressive model which is commonly used in model matching problems for linear systems.

2. Controller Architecture

In this section we introduce the controller structure and then write the closed loop system dynamics for the proposed ANN based controller.

*This work was in part sponsored by Westinghouse Education Foundation under contract No:3597262.

Consider the vector representation of an n link rigid manipulator dynamics, given as

$$\tau = M(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{v}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) \tag{1}$$

where τ is the $n\times1$ vector of joint torques, and $\mathbf{q}$ is the $n\times1$ vector of joint positions. The matrix $M(\mathbf{q})$ is the $n\times n$ positive definite "inertia matrix". $\mathbf{v}(\mathbf{q},\dot{\mathbf{q}})$ is $n\times1$ vector function representing centrifugal and Coriolis effects, and finally $n\times1$ vector function $\mathbf{g}(\mathbf{q})$ represents torques due to gravity. The derivation of (1) can be found in common reference texts [5]. Equation (1) can be put in a more compact form as ,

$$\tau = M(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q},\dot{\mathbf{q}}) \tag{2}$$

The control designer's *task* is to devise a controller such that the manipulator tracks a given desired trajectory $(\mathbf{q}_d,\dot{\mathbf{q}}_d,\ddot{\mathbf{q}}_d)$ as closely as possible.

If exact manipulator dynamics is available, then the *control*

$$\tau = M(\mathbf{q})(K_v\dot{\mathbf{e}} + K_p\mathbf{e} + \ddot{\mathbf{q}}_d) + \mathbf{h}(\mathbf{q},\dot{\mathbf{q}}) \tag{3}$$

will result in an error equation of the form,

$$\ddot{\mathbf{e}} + K_v\dot{\mathbf{e}} + K_p\mathbf{e} = 0 \tag{4}$$

due to the cancellation of nonlinear terms, where $\mathbf{e} = \mathbf{q}_d - \mathbf{q}$ and $\dot{\mathbf{e}} = \dot{\mathbf{q}}_d - \dot{\mathbf{q}}$. K_v and K_p are the diagonal matrices of *velocity* and *position* servo feedback gains, respectively. Note that the above error equation is a decoupled one due to the diagonal nature of the constant matrices K_p and K_v. Therefore adjusting these gain matrices properly, tracking errors can be effectively forced to zero. If the dynamic model of the manipulator is not available to the designer, generally learning control techniques are used to generate the necessary feedforward torques.

Here we propose the use of a generic ANN to model the inverse dynamic structure of each joint. For an n-link manipulator, inverse system dynamics given in (2) can be defined by a nonlinear transformation

$$R^{-1} : \mathcal{R}^{3n} \rightarrow \mathcal{R}^{n} \tag{5}$$

which maps the $3n$ dimensional output space of the system (position, velocity and acceleration vectors) to the n-dimensional input space (joint torque vector τ). Such a nonlinear transformation can be effectively modeled by ANNs as discussed by Funahashi [6], Cybenko [7] and Hornik, Stinchcombe and White [8].

Mathematical Setup:

Inverse dynamics which is represented by the nonlinear transformation R^{-1} in (5), can be decomposed into n transformations for each joint, namely

$$\begin{aligned} \tau &= R^{-1}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) && (6)\\ &= \begin{bmatrix} r_1^{-1}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) \\ \vdots \\ r_n^{-1}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) \end{bmatrix} && (7)\end{aligned}$$

where each $r_i^{-1}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$, $i = 1,\ldots,n$ defines the inverse dynamics of the corresponding joint, that is,

$$r_i^{-1}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) : \mathcal{R}^{3n} \rightarrow \mathcal{R}, \text{ with } i = 1,\ldots,n$$

Each entry $r_i^{-1}(\cdot)$ of the vector function R^{-1} can be modeled by a multilayer ANN such that the overall system's inverse dynamics model is represented by,

$$\tau = \hat{R}^{-1}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) = \begin{bmatrix} \hat{r}_1^{-1}(\mathbf{z}(t)) \\ \vdots \\ \hat{r}_n^{-1}(\mathbf{z}(t)) \end{bmatrix} = \begin{bmatrix} N_1(\mathbf{z},\mathbf{p}_1) \\ \vdots \\ N_n(\mathbf{z},\mathbf{p}_n) \end{bmatrix} \tag{8}$$

where $(\hat{\cdot})$ denotes the *estimated* models, and $N_i(\cdot)$, $i = 1, \ldots, n$ represents the output of each ANN model that is used to realize the nonlinear mapping $r_i^{-1}(\cdot)$. $\mathbf{z}(t)$ is an augmented state vector of the robot dynamics,

$$[\mathbf{z}(t) = (\mathbf{q}^T(t), \dot{\mathbf{q}}^T(t), \ddot{\mathbf{q}}^T(t))^T \in \mathcal{R}^{3n}$$

which denotes the time dependent input vector of the inverse dynamics and $\mathbf{p}_i$ is the *vector* of all adjustable weights of the ith ANN model.

Here we assume that a *three-layer ANN* with "k" inputs, "m" hidden layer units and one output unit is used to model a joint's inverse dynamics. Then this model can be explicitly written as,

$$\hat{r}_i^{-1}(\mathbf{z}(t)) = N_i(\mathbf{z}(t), \mathbf{w}_i(t), H_i(t)) = \mathbf{w}_i^T \Upsilon(H_i \mathbf{z}) \tag{9}$$

where $\mathbf{w}_i(t) \in \mathcal{R}^m$ is the adaptive output layer weight (parameter) vector, $H_i(t) \in \mathcal{R}^{m \times k}$ is the hidden layer adaptive weight matrix of the "i"th ANN model. $\mathbf{z}(t) \in \mathcal{R}^k$ with $k = 3n$ is the input vector as defined before. In the rest of the text, time argument of these vectors will sometimes be dropped for notational simplicity. The vector function $\Upsilon(\cdot) \in \mathcal{R}^m$ is defined as,

$$\Upsilon(\cdot) = (g_1(\cdot), \ldots, g_m(\cdot))^T$$

where $g_i(\cdot) \in \mathcal{R}$ is by definition a monotone increasing function which is taken as a *sigmoid function* in this case, based on the justification given by the theorems in [6, 8]. Hence $g_i(x) = \frac{1}{(1+e^{-x})}$ and it is bounded as $0 \leq g_i(x) \leq 1$.

We next define a vector $\mathbf{v}_i = vec(H_i) \in \mathcal{R}^{mk}$ which represents the hidden layer adaptive weight matrix H_i of the ANN model in vectoral form. $vec(\cdot)$ operator gives a vector which is obtained by stacking the columns of its matrix argument. Based on this new parameter vector, the argument of vector function $\Upsilon(\cdot)$ can be written as

$$H_i \mathbf{z} = \Phi \mathbf{v}_i \tag{10}$$

where

$$\mathbf{v}_i = \{H_{i_{11}}, H_{i_{21}}, \ldots, H_{i_{m1}}, \ldots, H_{i_{1k}}, \ldots, H_{i_{mk}}\}^T$$

and $\Phi \in \mathcal{R}^{m \times mk}$ is a matrix which can be considered as the modified input of the ANN model and is defined as,

$$\Phi = \begin{pmatrix} z_1 & 0 & \ldots & 0 & \ldots\ldots & z_k & 0 & \ldots & 0 \\ 0 & z_1 & \ldots & 0 & \ldots\ldots & 0 & z_k & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ldots\ldots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & z_1 & \ldots\ldots & 0 & 0 & \ldots & z_k \end{pmatrix} \tag{11}$$

where $z_i \in \mathcal{R}$, $i = 1, \ldots, k$ are the elements of the input vector $\mathbf{z} \in \mathcal{R}^k$. Hence (9) can now be written as,

$$\hat{r}_i^{-1}(\mathbf{z}, \mathbf{w}_i, \mathbf{v}_i) = N_i(\mathbf{z}, \mathbf{w}_i, \mathbf{v}_i) = \mathbf{w}_i^T \Upsilon(\Phi \mathbf{v}_i) \tag{12}$$

Next we define a control law which consists of an *adaptive feedforward compensator* and a *feedback* signal as,

$$\tau = \hat{R}^{-1}(\mathbf{z}) + \ddot{\mathbf{e}} + K_v \dot{\mathbf{e}} + K_p \mathbf{e} = \mathbf{N}(\mathbf{z}) + \ddot{\mathbf{e}} + K_v \dot{\mathbf{e}} + K_p \mathbf{e} \tag{13}$$

where $K_v \in \mathcal{R}^{n \times n}$ and $K_p \in \mathcal{R}^{n \times n}$ are the diagonal gain matrices with entries k_v and k_p, respectively.

$$\hat{R}^{-1}(\mathbf{z}) = \mathbf{N}(\mathbf{z}) = \{N_1, \ldots, N_n\}^T \in \mathcal{R}^n$$

is the robot's dynamic model estimate which consists of "n" individual ANN models, each representing one joint's inverse dynamics.

3. Closed Loop System Dynamics

In this section we generate the closed loop system's dynamic equations and demonstrate the difficulty of a global stability analysis. With the control vector given in (13), system's error dynamics can be written by substituting (13) in (2),

$$\hat{R}^{-1}(\mathbf{z}) + \ddot{\mathbf{e}} + K_v\dot{\mathbf{e}} + K_p\mathbf{e} = \underbrace{M(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q},\dot{\mathbf{q}})}_{R^{-1}(\mathbf{z})} \tag{14}$$

$$\ddot{\mathbf{e}} + K_v\dot{\mathbf{e}} + K_p\mathbf{e} = \tilde{R}(\mathbf{z}) \tag{15}$$

where $\tilde{R}^{-1}(\mathbf{z}) = \tilde{R}^{-1}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) \in \mathcal{R}^n$ denotes the error between the actual inverse dynamics R^{-1} and the estimated model $\hat{R}^{-1}$, and can be explicitly written as,

$$\tilde{R}^{-1}(\mathbf{z}) = \begin{bmatrix} \tilde{r}_1^{-1}(\mathbf{z},\mathbf{p}_1) \\ \vdots \\ \tilde{r}_n^{-1}(\mathbf{z},\mathbf{p}_n) \end{bmatrix} \tag{16}$$

where

$$\tilde{r}_i^{-1}(\mathbf{z},\mathbf{p}_i) = r_i^{-1}(\mathbf{z}) - N_i(\mathbf{z},\mathbf{p}_1) = r_1^{-1}(\mathbf{z}) - \hat{r}_1^{-1}(\mathbf{z})$$

denotes the error in inverse dynamic modeling for each joint, and $\mathbf{p}_i = \{\mathbf{w}_i^T, \mathbf{v}_i^T\}^T \in \mathcal{R}^{m+mk}$ is the adaptive weight vector of the corresponding ("i"th) ANN model.

Using (15) and (16) the error dynamics (with diagonal K_p and K_v matrices) for each joint can be written as follows,

$$\ddot{e}_i + k_v\dot{e}_i + k_p e_i = \tilde{r}_i^{-1}(\mathbf{z},\mathbf{p}_i), \text{ for } i = 1,\ldots,n. \tag{17}$$

where e_i, $\dot{e}_i$ and $\ddot{e}_i$ denote the position, velocity and acceleration errors at joint i, respectively, k_p and k_v are the individual servo gains, respectively. Based on the existence theorems given in [6, 8], we can assume that there exists a three layer ANN structure (which is defined in (12)) that closely approximates the joint's inverse dynamics. That is,

$$r_i^{-1}(\mathbf{z}) = \mathbf{w}_{i0}^T\Upsilon(\Phi\mathbf{v}_{i0}) \tag{18}$$

where $\mathbf{w}_{i0}$ and $\mathbf{v}_{i0}$ represent the *desired* output layer and hidden layer weights that generate the *desired mapping*, r_i^{-1}. In the rest of the analysis the subscript "i"is dropped for the brevity of the presentation. Using the desired mapping given in (18), the error in the approximation of the inverse dynamics of a joint can be written as,

$$\tilde{r}^{-1}(\mathbf{z},\mathbf{p}) = \underbrace{\mathbf{w}_0^T\Upsilon(\Phi\mathbf{v}_0)}_{r^{-1}} - \underbrace{\mathbf{w}^T\Upsilon(\Phi\mathbf{v})}_{\hat{r}^{-1}} \tag{19}$$

Using this residual dynamic representation, we can analyze the error equation given in (17) which in fact represents a stable linear system with a nonlinear forcing function. Then the error dynamics in (17) can be written as,

$$e(t) = H(s)\tilde{r}^{-1}(\mathbf{z},\mathbf{p}) \tag{20}$$

where $H(s)$ is a strictly positive real (SPR) transfer function, due to the positive gain terms k_p and k_v in (17). Based on this error dynamic model, to update the parameter estimates $\mathbf{p}$, a simple *gradient update* algorithm can be written as,

$$\dot{\mathbf{p}} = -\Gamma\frac{\partial\tilde{r}^{-1}(\mathbf{z},\mathbf{p})}{\partial\mathbf{p}}e(t) \tag{21}$$

where $\Gamma \in \mathcal{R}^{(m+mk)\times(m+mk)}$ is a diagonal gain matrix which includes the learning rates as its diagonal entries. The state space representation of (17) is given by,

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + B\tilde{r}^{-1}(\mathbf{z},\mathbf{p}) && (22) \\ e &= \mathbf{c}\mathbf{x} && (23) \end{aligned}$$

where $\mathbf{x} = (e, \dot{e})^T \in \mathcal{R}^{2\times 2}$ is the state vector, and A, B and and $\mathbf{c}$ denote the minimal state space representation due to strictly positive real $H(s)$. Since $H(s)$ is an SPR transfer function, based on the Kalman-Yakubovich lemma [9], for a given symmetric positive definite (p.d.) matrix Q, there exists another symmetric positive definite matrix P which satisfies the Lyapunov equation,

$$A^T P + PA = -Q \qquad (24)$$

and the output relation,

$$\mathbf{c} = B^T P \qquad (25)$$

Replacing the output vector $\mathbf{c}$ in (21) by (25), the gradient update law take sthe form,

$$\begin{aligned} \dot{\mathbf{p}} &= -\Gamma \frac{\partial \tilde{r}^{-1}(\mathbf{z},\mathbf{p})}{\partial \mathbf{p}} B^T P \mathbf{x} && (26) \\ \dot{\mathbf{p}} &= \Gamma \frac{\partial N(\mathbf{z},\mathbf{p})}{\partial \mathbf{p}} B^T P \mathbf{x} && (27) \end{aligned}$$

where $N(\mathbf{z},\mathbf{p})$ is the output the ANN model. Due to the nonlinear parametric dependence of the term $N(\mathbf{z},\mathbf{p})$, the computation of the partial derivative term in (27) requires the so-called *backpropagation* algorithm. Note that the closed loop adaptive system represented by (22) and (27) defines a *coupled nonlinear* system of differential equations and this makes a global convergence and stability analysis of the closed loop system difficult. However local properties of the system dynamics can be studied through the use of linearization techniques. Linearization dictates that, subject to smoothness of the nonlinear oprators in (22) and (27), one can constitute a linearized system whose stability properties are identical to the local stability properties of (22) and (27).

4. Stability and Convergence Analysis

In order investigate the local stability and convergence properties of the closed loop system, the system dynamics is linearized around the desired system state. Let $\mathbf{p}_*$ and $\mathbf{x}_*$ denote the desired values of the parameter vector $\mathbf{p}$ and the state vector $\mathbf{x}$, respectively. That is $\mathbf{x}_* = 0$ and $\mathbf{p}_* = (\mathbf{w}_0^T, \mathbf{v}_0^T)^T$. This basically corresponds to a condition where the system operates in the vicinity of the desired trajectories $(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)$, and the ANN model parameters are close to their desired values. With this choice of the nominal signals, the perturbation vectors become,

$$\tilde{\mathbf{x}} = \mathbf{x}_* - \mathbf{x} = -\mathbf{x}$$

which is actually the tracking error vector and

$$\tilde{\mathbf{p}} = \mathbf{p}_* - \mathbf{p}$$

which is the parameter error vector. With this set up, we first linearize (22) around $\mathbf{x}_* = 0$ and $\mathbf{p}_*$ as follows,

$$\dot{\tilde{\mathbf{x}}} = \left.\frac{\partial(A\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}_*} \tilde{\mathbf{x}} - B \left.\frac{\partial(\mathbf{w}^T \Upsilon(\Phi \mathbf{v}))}{\partial \mathbf{w}}\right|_{\mathbf{z}_*,\mathbf{w}_*,\mathbf{v}_*} \tilde{\mathbf{w}} - B \left.\frac{\partial(\mathbf{w}^T \Upsilon(\Phi \mathbf{v}))}{\partial \mathbf{v}}\right|_{\mathbf{z}_*,\mathbf{w}_*,\mathbf{v}_*} \tilde{\mathbf{v}} \qquad (28)$$

where $\mathbf{z}_* = (\mathbf{q}_d{}^T, \dot{\mathbf{q}}_d{}^T, \ddot{\mathbf{q}}_d{}^T)$ basically corresponds to the desired (nominal) state $\mathbf{x}_* = 0$. Evaluating the partials, we get

$$\dot{\mathbf{x}} = A\mathbf{x} + B\Upsilon^T(\Phi_* \mathbf{v}_*)\tilde{\mathbf{w}} + B\mathbf{w}_*^T J_* \Phi_* \tilde{\mathbf{v}} \tag{29}$$

where $J_* \in \mathcal{R}^{m \times m}$ is a diagonal Jacobian matrix evaluated at the nominal values,

$$J_* = \begin{pmatrix} g_1'|_{\mathbf{z}_*,\mathbf{v}_*} & 0 & \dots & 0 \\ 0 & g_2'|_{\mathbf{z}_*,\mathbf{v}_*} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & g_m'|_{\mathbf{z}_*,\mathbf{v}_*} \end{pmatrix} \tag{30}$$

and Φ_* is the vector function (defined in (11)) which is evaluated at $\mathbf{z}_*$. Combining $\tilde{\mathbf{v}}$ and $\tilde{\mathbf{w}}$ as the *parameter error vector*,

$$\tilde{\mathbf{p}} = (\tilde{\mathbf{w}}^T, \tilde{\mathbf{v}}^T)^T \in \mathcal{R}^{(m+mk)}$$

equation (29) can be written as

$$\dot{\mathbf{x}} = A\mathbf{x} + B\Psi_*(\mathbf{z}_*, \mathbf{w}_*, \mathbf{v}_*)\tilde{\mathbf{p}} \tag{31}$$

where

$$\Psi_* = (\Upsilon_*^T(\Phi_* \mathbf{v}_*), \mathbf{w}_*^T J_* \Phi_*) \in \mathcal{R}^{1 \times (m+mk)} \tag{32}$$

can be considered as the *linearized regressor vector* of the error dynamics, and Using the same arguments and linearizing (27) explicitly for $\mathbf{w}$ and $\mathbf{v}$, we get,

$$\dot{\tilde{\mathbf{w}}} = \Gamma_1 \Upsilon(\Phi_* \mathbf{v}_*) B^T P \tilde{\mathbf{x}} \tag{33}$$

$$\dot{\tilde{\mathbf{v}}} = \Gamma_2 \Phi_*^T J_* \mathbf{w}_* B^T P \tilde{\mathbf{x}} \tag{34}$$

where Φ_* and J_* are as defined previously and $\Gamma_1 \in \mathcal{R}^{mk \times mk}$ and $\Gamma_2 \in \mathcal{R}^{k \times k}$ are diagonal matrices which are in fact the partitions of the gain matrix Γ. Combining equations (33) and (34), linearized update equation for the parameter vector $\mathbf{p}$ can be written as follows

$$\dot{\tilde{\mathbf{p}}} = -\Gamma \Psi_*^T B^T P \mathbf{x}, \quad \text{since } \tilde{\mathbf{x}} = -\mathbf{x} \tag{35}$$

where Ψ_* is as defined in (32). Equations the (31) and (35) constitute the linearized closed loop system dynamics. Based on this linearized model we can now state the following theorem [10].

Theorem 1:
Given the linearized system dynamics in (31) and the update law in (35), the tracking error vector $\mathbf{x}$ *satisfies,*

$$\mathbf{x} \rightarrow 0 \ as \ t \rightarrow \infty$$

with all signals remaining bounded.

The proof of the theorem is given in [10]. The above result ensures the convergence of the tracking error vector $\mathbf{x}$ in the vicinity of a nominal solution. However note that the above theorem does not guarantee the convergence of the parameter error vector $\tilde{\mathbf{p}}$, although it shows *its boundedness.*

5. Convergence of Weight Estimates

In order to investigate the convergence properties of the parameter error vector $\tilde{\mathbf{p}}$, let's write the linearized closed system dynamics using (31) and (35),

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\tilde{\mathbf{p}}} \end{bmatrix} = \begin{bmatrix} A & B\Psi_* \\ -\Gamma \Psi_*^T B^T P & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \tilde{\mathbf{p}} \end{bmatrix} \tag{36}$$

Equations similar to (36) arise in most adaptive control applications, and its asymptotic stability has been studied by several researchers [11, 12, 4]. Note that equation (36) defines a linear time-varying system and its asymptotic stability determines the convergence of the parameter error vector $\tilde{\mathbf{p}}$. The system given in (36) is asymptotically stable, if the transfer function

$$B^T P(sI - A)^{-1} B \tag{37}$$

is strictly positive real and if there exists positive constants β, γ and T such that

$$\beta I \succeq \int_{t_0}^{t_0+T} \Psi_*^T \Psi_* dt \succeq \gamma I \tag{38}$$

holds for all t_0 with $I \in \mathcal{R}^{(m+mk)\times(m+mk)}$ being the identity matrix [11]. The sign $\succeq$ defines the positive semi-definite ordering of the matrices. The transfer function defined in (37) is strictly positive real for the above case as discussed in Section 4. Therefore if condition (38) is satisfied, then $\tilde{\mathbf{p}}$ converges asymptotically. Equation (38) which is usually referred to as the persistent excitation (p.e.) condition, states that Ψ_* must vary sufficiently over the interval T such that the entire $(m + mk)$ dimensional space (i.e. the parameter space of the ANN model) is spanned to ensure the convergence of $\tilde{\mathbf{p}}$.

In equation (38), Ψ_* is bounded as seen from (32). Then, left hand inequality holds directly and the p.e. condition can be written as,

$$\int_{t_0}^{t_0+T} \Psi_*^T \Psi_* dt \succeq \gamma I \tag{39}$$

The outer product matrix in the integral equation given in (39) can be explicitly written as a partitioned matrix as,

$$\Psi_*^T \Psi_* = \left(\begin{array}{c|c} \Upsilon_*(\Phi_* \mathbf{v}_*)\Upsilon_*^T(\Phi_* \mathbf{v}_*) & \Upsilon(\Phi_* \mathbf{v}_*)\mathbf{w}_*^T J_* \Phi_* \\ \hline \Phi_*^T J_*^T \mathbf{w}_* \Upsilon_*^T(\Phi_* \mathbf{v}_*) & \Phi_*^T J_*^T \mathbf{w}_* \mathbf{w}_*^T J_* \Phi_* \end{array} \right) \tag{40}$$

Note that the above matrix is related to the desired trajectory vector $\mathbf{z}_* = (\mathbf{q}_d{}^T, \dot{\mathbf{q}}_d{}^T, \ddot{\mathbf{q}}_d{}^T)^T$ through Φ_* and J_*, with $\mathbf{z}_*$ acting as their arguments. Due to the nonlinear nature of the above matrix integral equation, it is rather difficult and impractical to generate persistently exciting trajectories from that equation. One method to utilize the condition given in (39) is to generate desired trajectories based on our engineering knowledge and then to test their p.e. condition. We are currently studying the structure of (39) in order to derive more explicit p.e. conditions on $\mathbf{z}_*$. In general, generation of persistently exciting trajectories is an important research topic [11]. Next we present some simulation results and demonstrate the error and parameter convergence properties of the proposed controller architecture.

6. Simulation Results

The architecture proposed in Section 2 was tested on a two link manipulator model using simulation methods [2]. As a test for the adaptation properties of the controller, the end effector mass is changed while the manipulator is closely following a prescribed trajectory. For this test, the second link mass is changed from its nominal value of 8 kgs. to 16 kgs. at the "2" second mark. The position error profiles due to this sudden change in manipulator dynamics are shown in Figure (1). As shown in the figure, the controller effectively reduces the sudden jumps in the position errors and brings the errors down approximately to their previous levels in about 1-1.5 seconds. In order to demonstrate the changes in the manipulator's inverse dynamics due to the end effector mass change and the ANN model's ability to track these changes, the torque profiles are monitored during the adaptation test (Figure (2)).

Desired torque profiles corresponding to this change and the torque profiles *generated by the ANN models* of each joint are plotted in Figure (2). The observed torque profiles converge to their desired levels by the end of the trajectory. More simulation results are given in [2].

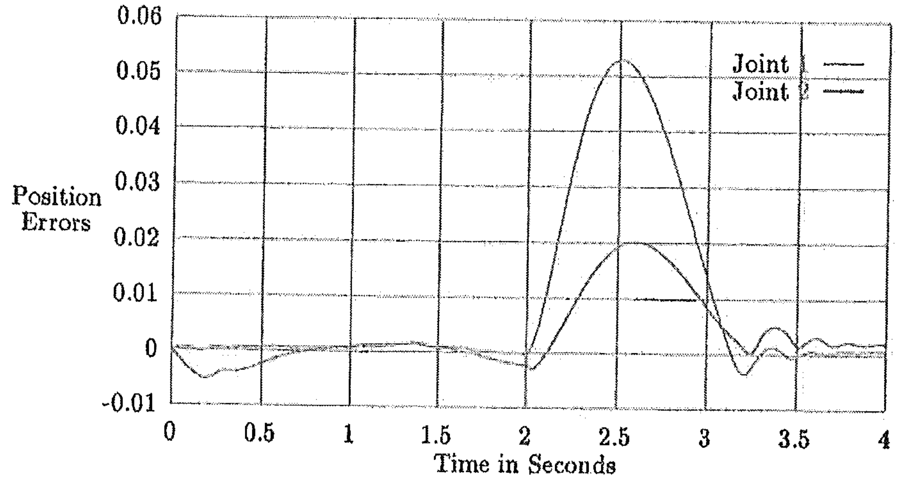


Figure 1: Position errors for the adaptation experiment. End effector mass is changed from 8 kgs. to 16 kgs.

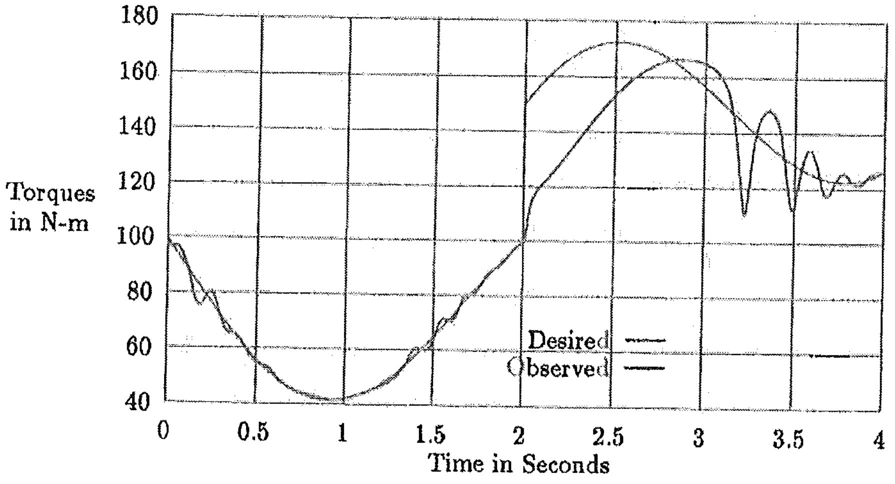


Figure 2: Desired and observed torque profiles when the end effector mass is changed from its nominal value of 8 kgs. to 16 kgs.

7. Conclusions

Neurologically inspired robotic controllers have been receiving much attention in recent years due to their effective learning capabilities. There are many reports on successful simulation results, however there is not yet a well defined mathematical analysis for their stability and convergence. This is due to the nonlinear nature of parameter update dynamics which makes use of the well known backpropagation algorithm. This paper investigates the local stability and convergence properties of an ANN based robotic controller which was previously proposed by the author. Simulation experiments demonstrate the convergence properties of the controller architecture.

References

[1] "IEEE Control Systems Magazine," April 1988-1989-1990. Special Section on Neural Networks.

[2] K. Cılız, *Artificial Neural Network Based Control of Nonlinear Systems With Application to Robotic Manipulators.* PhD thesis, Syracuse University, Electrical Engineering Department, 1990.

[3] K. Cılız and C. Işık, "Trajectory learning control of robotic manipulators using Neural Networks," in *IEEE International Symposium on Intelligent Control*, (Fairfax, VA), pp. 536–540, 1990.

[4] J. J. Craig, P. Hsu, and S. S. Sastry, "Adaptive control of mechanical manipulators," *International Journal of Robotics Research*, vol. 6, no. 2, pp. 16–28, 1987.

[5] A. J. Koivo, *Fundamentals for Control of Robotic Manipulators.* New York: John Wiley & Sons, 1989.

[6] K. C. Funahashi, "On the approximate realization of continuous mappings by Neural Networks," *Neural Networks*, vol. 2, pp. 183–192, 1989.

[7] G. Cybenko, "Approximations by superpositions of a sigmoidal function," tech. rep., University of Illinois, Center for Supercomputer Research and Development, Urbana, IL, 1989.

[8] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[9] C. A. Desoer and M. Vidyasagar, *Feedback Systems : Input-Output Properties.* New York, NY: Academic Press, 1975.

[10] K. Cılız and C. Işık, "Stability and convergence of neurologic model based robotic controllers," in *IEEE International Conference on Robotics and Automation*, (Nice, France), May 1992.

[11] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems.* Englewood Cliffs, N.J.: Prentice Hall, 1989.

[12] K. S. Narendra and A. P. Morgan, "On the uniform asymptotic stability of certain linear nonautonomous differential equations," *SIAM Jour. Cont. and Opt.*, no. 15, pp. 5–24, 1977.

Pattern Classifier

Atilla E. Günhan,

Department of Information Science, University of Bergen.
Hightechnology center, N-5020 Bergen, Norway

E-mail: gunhan@cc.uib.no Telefax: (47-5) 544107

Abstract: In the present work, an alternative multi-layer unsupervised neural network model that may approximate certain neurophysiological features of Natural Neural Systems has been studied. The Network is formed by two parts. The first part of the network plays a role as a Short Term Memory that is a temporary storage for each pattern. The task for this part of the network is to preprocess incoming patterns without memorizing, in other words, to reduce the dimensions and the linear dependency among patterns by determining their relevant representations. This preprocessing ability is obtained by a dynamic lateral inhibition mechanism on the hidden layer. These representations are the input patterns for the next part of the network. The second part of the network may be accepted as a Long Term Memory which classifies and memorizes incoming pattern informations that come from hidden layer.

1 Architecture of the Network and Training process

The network is composed of two feed forward layers (Fig. 1).

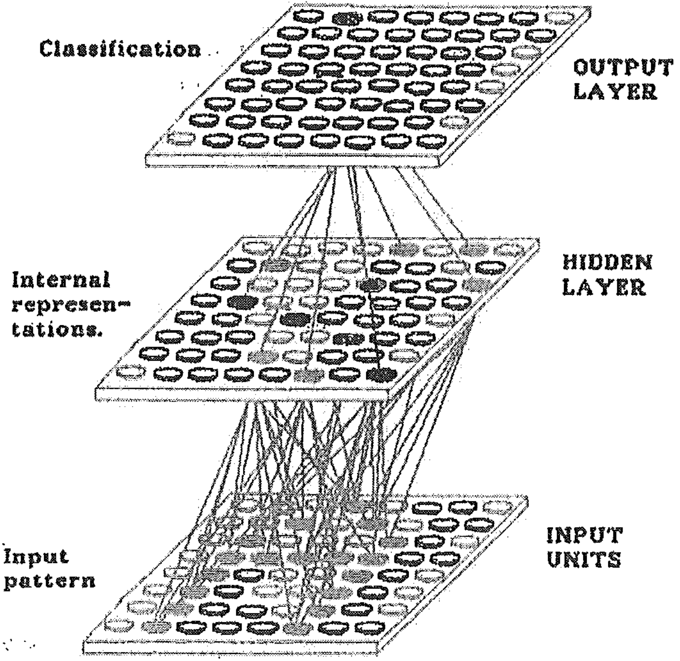


Fig. 1: Architecture of multi layer network.

64 input units, a hidden layer with 64 hidden units and an output layer with 64 units. The layers are fully interconnected to the next layer. The output and hidden layers have internal lateral inhibitions. The first part that is between input units and hidden layer accepts input patterns one at a time (Fig 2.) In other words. when we present the first pattern to this part, only this part of the network will be trained until we obtain internal representations in the hidden unit for this pattern.

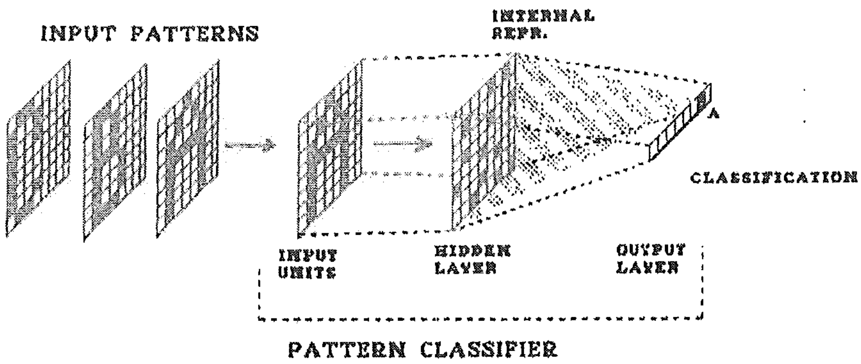


Fig. 2: Training process.

Before we can present the second pattern this part of the network will be initialized. After we have obtained an internal representation for the first pattern in the hidden layer, this representation presented to the second part of the network as an input for final classification of the pattern. There are two possible preprocessing in the first layer. We can train the network until we obtain internal representations of the input patterns. In this type of training, it is possible that patterns can have common elements. In the second type of trining, we can train the network by eliminating common activated units until we obtain unique representations for each input patterns. In this second method patterns will be reduced into their orthogonal representations.

The second part of the network can be considered as another single layer network between the hidden layer and output layer. The function of this second network differs from the first part of the network in terms of the way of training. This part accepts the internal representation as input and the layer is trained untill the pattern is classified. The next internal representation, which is the result of the second pattern that is presented in the first part, will also be accepted as second input pattern to the second part of the network without any initialization. In this way, the second part of the network is capable of training and memorizing a sequence of patterns as in traditional training.

A "neighbour inhibition" method with an Eight-directional Inhibition Strategy (Fig. 3) is used in the layer of the first part of the network that is in fact the hidden layer for the whole network. In this strategy most active unit inhibits the activity of its closest neighbour units. Units in the layer organized in a two dimensional array. In the output layer of the network, the "Winner-take-all" method is used.

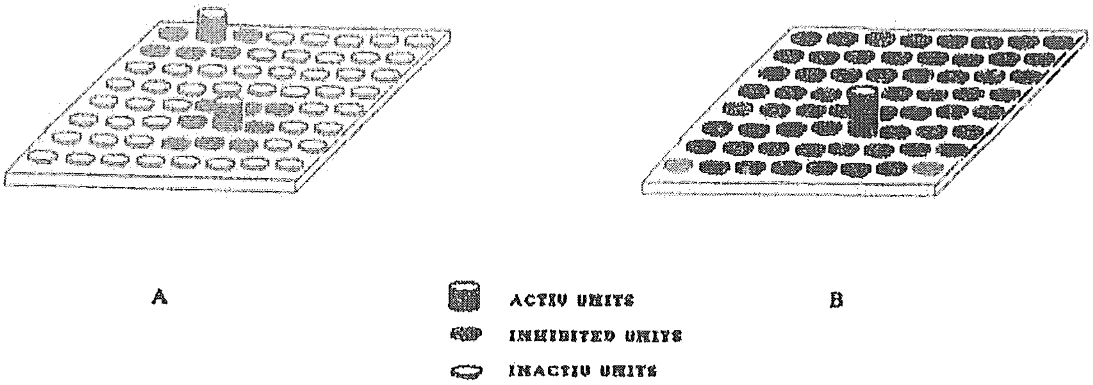


Fig. 3: a) Neighbour neuron inhibition with Eight-directional Inhibition Strategy. Most active unit inhibits the activity of its closest neighbour units. b) Winner-take-all Inhibition Strategy. Winner unit pushes other neurons into a constant minimum value.

2 Learning algorithm of the Network

There is a large number of fibers that provide synaptic connection to a given Purkinje cell (Fig. 4) in a Natural Neural Network[9]. We denote the number of all parallel fibers by N. The input carried by the k^{th} parallel fiber at the time t is denoted by $s_k(t)$: it can have the value of 1 or 0 according to whether the fiber carries an impulse or not. The effect of the input from the k^{th} fiber on the i^{th} Purkinje cell is determined by the synaptic coupling strength $W_k^{(i)}$.

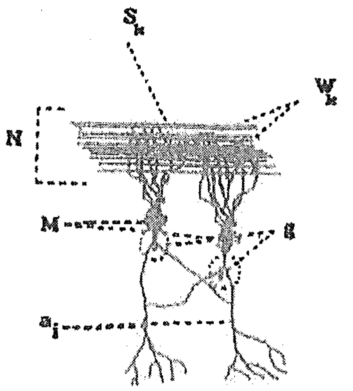


Fig. 4: Symbolic notations for two Purkinje cells.

The output of the network is carried by the axons of the M Purkinje cells and acts to inhibit muscle activity. The strength of the effective inhibition of a cell denoted by g. This activity of the i^{th} Purkinje cell is characterized by the quantity $a^{(i)}(t)$, which is 1 or 0 depending on whether the cell it originates at is active (*i.e.* has fired) or not. The activity state of the Purkinje cell in turn depends on its axon hill potential, $v^{(i)}(t)$.

2.1 Neuronic equation (Netinput)

To describe the behaviour of a neuron, I will employ a formulation based on ref. 3. The dynamic behaviour of a neuron is governed by the *neuronic equation:*

$$v^{(i)}(t+\tau) = \Big[(1-\lambda)v^{(i)}(t) + \sum_{k=1}^{N} W_k^{(i)}(t)s_k(t) - \sum_{j=1}^{M} g_j^{(i)}a^{(j)}(t)\Big](1-a^{(i)}(t)) \, , \qquad (1)$$

where

- $v^{(i)}(t+\tau)$:Hill potential at time $(t+\tau)$ of cell i. (In natural systems this time is $\tau \approx 1$ msec.)
- $(1-\lambda)v^{(i)}(t)$:The remaining potential of neuron i from time t at time $(t+\tau)$. If no other input arrives from fibers then the potential of the neuron decreases exponentially with λ decay constant ($\lambda = 0.1$ in my appl.)
- $W_k^{(i)}(t)s_k(t)$:The excitation arriving from the k^{th} parallel fiber to neuron i at time t.
- $g_j^{(i)}a^{(j)}(t)$:The inhibition arriving from other neurons to neuron i at time t.
- $(1-a^{(i)}(t))$:A multiplicative factor. This term resets the potential to its resting value after the firing and keeps it there the refractory period (the minimum time between two firing), during which the the neuron is not excitable.

A neuron will become active, if its potential $v^{(i)}$ at the axon hill exceeds a threshold value, $\theta^{(i)}$, ($\theta = 1$ in my appl.). In that case the neuron will emit a non-decreasing pulse of fixed duration τ, along its axon making inhibitory action at all of its synaptic connections.

This activity is described as,

$$a^{(i)}(t) = \begin{cases} 1, if \; v^{(i)}(t) \geq \theta^{(i)} \\ 0, \quad otherwise \end{cases}$$

Given the time-dependent input $s_k(t)$, the response of the network can be calculated with eq. (1). For a shorter period, the synaptic strengths $W_k^{(i)}$ and $g_j^{(i)}$ can be considered as constants. If the incoming pattern remains steady for several time steps; those Purkinje cells whose synaptic strength vectors have the best overlap with the input pattern vector will be most likely to fire; in turn, they will then inhibit their neighbouring neurons as a consequence of the lateral inhibition $g_i^{(j)}$. If the membrane potential, v, of a neuron exceeds a given threshold value, θ, then neuron becomes active (fired) and inhibits its neighbours with a certain inhibition value. Active neuron also resets itself into resting membrane potential value.

2.2 Memory equation (Learning)

On a longer time scale the coupling strengths may change, thus providing a learning ability for the net. When we assume that the inhibitory synapses are fixed, the excitatory synapses leading from the parallel fibers to the Purkinje cells may change according to Hebb's rule[6], which we formulate in the following *memory equation*:

$$W_k^{(i)}(t+\tau) = q^{(i)}\Big[(1-\varepsilon)W_k^{(i)}(t) + \delta a^{(i)}(t)s_k(t-\tau)\Big]\ . \tag{2}$$

where

- $W_k^{(i)}(t+\tau)$:Synaptic strength of coupling between fiber k and neuron i at the time $(t+\tau)$.
- $q^{(i)}$:Normalization constant for neuron i.
- $(1-\varepsilon)W_k^{(i)}(t)$:Exponential decay of synaptic strength ($\varepsilon = 0.001$ in my appl.)
- $\delta a^{(i)}(t)s_k(t-\tau)$:If neuron i active at time t (i.e., $a=1$), the connections between neuron i and fiber k will be strengthened by δ learning rate.

The normalization condition

$$q^{(i)} = \eta^{(i)}/(\sum_{k=1}^{N}\Big[(1-\varepsilon)W_k^{(i)}(t) + \delta a^{(i)}(t)s_k(t-\tau)\Big]) \tag{3}$$

where $\eta^{(i)}$ is a constant (i.e., $\eta^{(i)} = 1$). This condition ensures that the total synaptic strength remains constant,

$$\sum_{k=1}^{N} W_k^{(i)}(t) = \eta^{(i)}.$$

This learning mechanism simply describes the effect that if an excitatory impulse is arriving to a synaptic coupling which will lead in the following time step to a firing of the post-synaptic cell, then that connection will be strengthened by δ. Due to the normalization $\eta^{(i)}$, and the slow exponential decay of the couplings, all "non-successful" synapses will be somewhat weakened at the same time. Exponential decay $(1-\lambda)$ in *neuronic equation* and $(1-\varepsilon)$ in *memory equation* are ignored in the second layer of the network to obtain plausible classification of patterns.

Learning algorithm for the second layer of network is as follows *neuronic equation*,

$$v^{(i)}(t+\tau) = \left[v^{(i)}(t) + \sum_{k=1}^{N} W_k^{(i)}(t)s_k(t) - \sum_{j=1}^{M} g_j^{(i)} a^{(j)}(t)\right](1 - a^{(i)}(t)) \, , \tag{4}$$

and *memory equation*,

$$W_k^{(i)}(t+\tau) = q^{(i)}\left[W_k^{(i)}(t) + \delta a^{(i)}(t)s_k(t-\tau)\right] . \tag{5}$$

3 Features of the network

The learning algorithm and architecture of the network explained above are different from other network models. The advantages of the two layer model presented above are:

Ability of preprocessing the incoming pattern.

The network has ability to preprocess the incoming patterns and reduce the dimensions and the linear dependency among these patterns then memorize them. Experiments indicate that the network has ability to classify all input vectors as long as those vectors have less than 40% common elements with each other. When the patterns **A, B, C, 1** and **2** presented to the network (Fig. 5), patterns **C** and **2** have been classified into same class.

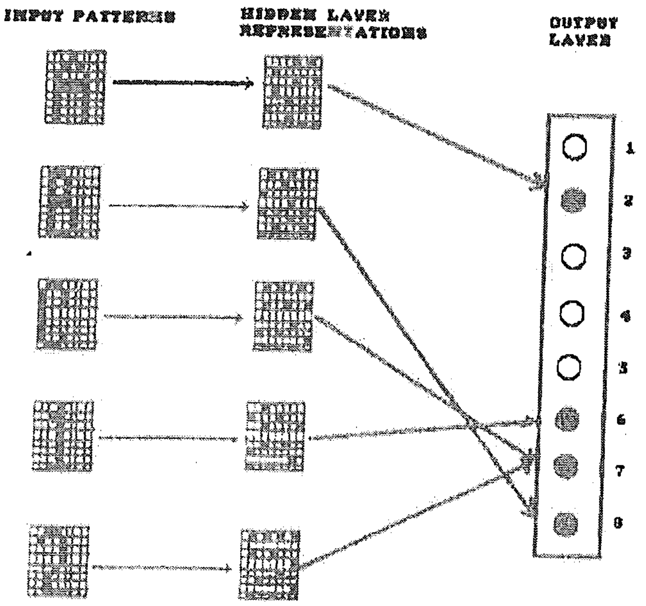


Fig. 5: Input patterns **A, B, C, 1, 2** , their Internal representations and response of the network in output layer.

The reason is that common elements of these two patterns are more than 40% of the number of representation elements. Patterns **A** and **C** had 4 common elements but they have been classified into their own classes since their uncommon elements were in majority. The second layer of the network is limited with properties of ''winner-takes-all'' method[10].

Network architecture and learning algorithm are similar to biological mechanisms.

The learning algorithm is based on neurophysiological activity of real neurons. If the membrane potential, v, of a neuron at a given time exceeds a certain threshold value, θ, then the neuron becomes active (fired) and inhibits its neighbours with a certain inhibition value. Only the weights of the active neuron are adjusted. Active neuron also resets itself into resting membrane potential value.

Network accepts repeated patterns.

Repeated patterns during training do not cause any side effect. They activate always the same output units.

Network has ability to learn a new pattern without training the whole set again.

Network trains with one at a time process. It makes possible to add a new pattern to the network in any given time.

Values of parameters are fixed.

Choice of parameter values for learning and the number of training iterations are fixed. The network may be trained with a value of learning rate 0.1, and 150 cycle for each pattern for each layer. These values are valid for any type of pattern. Inhibition values between the units are set to 0.9 when the method is neighbour inhibition.

Conclusion

In the present work, an alternative unsupervised neural network model that may approximate certain neurophysiological features of Natural Neural Systems has been studied. This work is a result of further investigations on ref. 3 and 4. However the present model shows us the ability of character recognition, It can also be investigated within the subject of signal processing. This model can work as a feature detector or signal classifier in many application areas like speech recognition, active sonar classification, *etc.*

References

[1] Caianiello, E. R. , *Outline of Theory of Thought-Processes and Thinking Machines*, J. Theoret. Biol. (1961) **2**, 204-235

[2] Caianello, E.R. , *Cybernetics of Neural Process* (Ricerca Scientifica, Roma, 1965).

[3] Günhan E. Atilla, Csernai L. P. & Randrup J., *"Unsupervised Competitive Learning in Neural Networks"*, International Journal of Neural Systems, World Scientific, London, Vol. **1**, No. **2** 177-186

[4] Günhan E. Atilla, *"Pattern recognition and Self-Organization of Neural Networks"*, Dept. of Information Science, University of Bergen, Norway. Master thesis, (March 1991).

[5] Günhan E. Atilla, *"Pattern Classifier, An Alternative Method of Unsupervised Learning"*, Neural Network World, International Journal on Neural and Mass-Parallel Computing, Czechoslovakia, Vol. **1**, No. **6** 349-354, (December 1991)

[6] Hebb, D. O. ,*The Organization of Behaviour* , (Wiley, New York 1949)

[7] Hinton, G. E. , *Connectionist Learning Procedures*, Technical Report, CMU-CS-87-115, (University of Toronto, Canada 1987)

[8] Kohonen, T., *Self-Organization and Associative Memory*, (Springer-Verlag, New York 1984)

[9] Pellionisz, A. and Llinás, R. , *A computer model of Cerebellar Purkinje Cells*, Neuroscience, (1977) **Vol. 2**, 37-48

[10] Rumelhart, D. E. , McCelland, J. L. and The PDP research group, *Parallel Distributed Processing* (The MIT press, Cambridge 1986),**Vol. 1-2**

Yapay Sinir Aglarinin Robotikteki Uygulamalari

D. T. Pham ve S. Sagiroglu

Intelligent Systems Laboratory,
School of Electrical, Electronic and Systems Engineering,
University of Wales College of Cardiff,
P.O. Box 904, Cardiff, CF1 3YH, U.K.

Ozet

Robotikte, bazi problemler vardir ki bu problemlerin analitik cozumlerinin konvensiyonel tekniklerin kullanilmasiyla bulunmasi mumkun olamamaktadir. Bu calismada Yapay Sinir Aglarinin (YSA) boyle problemleri cozmede kullanimi gozden gecirilmistir.

Bu makalede (*), YSAlar kisaca gozden gecirilmis ve YSA'nin robot kinematik, dinamik ve kontrol problemlerinin cozumunde nasil uygulandigi konusunda uygulamalar sunulmustur.

Anahtar Kelimeler : Yapay Sinir Aglari, Robotlar, Robot Kinematik, Robot Dinamik, Robot Kontrol.

Applications of Artificial Neural Networks in Robotics

Abstract

In robotics, there are a number of problems for which it is difficult to obtain analytical solutions using conventional techniques. This paper examines the use of Artificial Neural Networks to solve such problems.

The paper presents an overview of ANNs and discusses how they have been applied to the solution of problems in robot kinematics, dynamics and control.

Keywords : Artificial Neural Networks, Robotics, Robot Kinematics, Robot Dynamics, Robot Control.

1. GIRIS

YSAlar degisik bir cok problemlerin cozumunde son yillarda genis bir sahada basariyla uygulanmistir. Bunda, YSAnin insan zekasini taklit eden cazibeli, dogal bir yapiya sahip olmasinin etkisi buyuktur. YSAlarin, intibak kabiliyetleri ve ogrenme yeteneginden dolayi bir cok uygulama sahasinda bir cok pratik probleme cozum saglamasi, onun populeritesini artirir. Son gunlerde bu uygulama alani **Neuromuhendislik** (Neuroengineering) olarakta adlandirilir ve YSAlarin pratik uygulamalari uzerine kurulmustur ve insan beyininin bilinen ozelliklerinin komputer donanimi ve yazilimina aktarilmasidir [1].

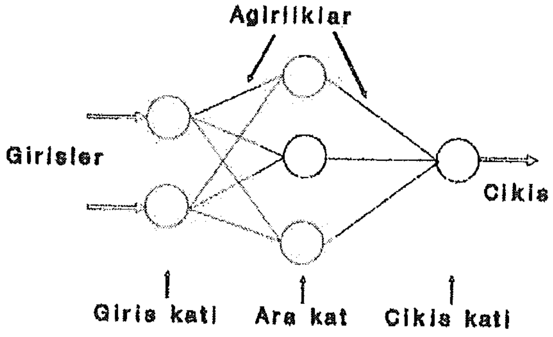
(*) Yazarlar Turkce editor kullanamadiklari icin okuyuculardan ozur diler.

Robotikte gaye, insanin caninin yapmak istemedigini veya yapmasinda tehlike olan seyleri, kendi kontrolu altinda yapan bir cihaz icad etmektir [2]. Robotlar oldukca nonlinear sistemler olup dinamik performanslari, hesaplama verimliligine baglidir. Ornek olarak; Kartezyen ve degisebilir baglanti uzaylari arasindaki koordinat transformu, baglanti motorlarini suren genellestirilmis kuvvetler veya torklar, kontrol icin manipulator atalet matrisi, degisken Kartezyen uzayindaki Jacobian matris.
Bu makalede gunumuze kadar robotikte yapilmis YSA uygulamalarinin [3,4,5,6,12,13] bircogu tanitilmakta ve yapilmis uygulamalar hakkinda kisa bilgiler verilmektedir. Ikinci kisimda, YSAlarin kisaca tanitimi yapilmis ve YSAlarin robotikdeki avantajlari ve dezavantajlarindan bahsedilmistir. Ucuncu kisimda ise, robotik ve robotik problemleri ile ilgili kisa bilgiler verilmistir. Bir sonraki kisimda ise, YSA'larin robotikdeki uygulamalari ve robotikteki problemleri cozmedeki basarisi goz onune serilmistir. Besinci bolumde verilen kaynaklarin bir degerlendirilmesi yapilmistir.

2. YAPAY SINIR AGLARI

Yapay sinir aglari temel olarak beynimizin matematiksel modellemesi uzerine kurulmustur. YSAlar basitce izah edilirse, bir cok isleme elemanlarinin (processing elements) birbirlerin arasindaki farkli baglanti sekillerinden olusur, her YSA'nin kendine ozgu bir yapisi ve ogrenme sekli vardir. Baglanti sekillerine ve yapilarina gore YSAlari siniflandirma yapmakta mumkundur [7,8,9,10,11,13]. Aslinda, YSAlarin sirri, onlarin giris ile cikis arasinda bir harita olusturmalarindan gelir yani verilmis olan bir giris setine bir cikis seti uretirler.

YSA topolojisi, ileri-beslemeli (feedforward) YSA ve geri-beslemeli (feedback) YSA diye ikiye ayrilir [7] (daha degisik siniflandirmalarda mevcuttur). Ileribeslemeli sinir agi giris uzayi ile cikis uzayi arasinda statik haritalama yapar ve verilmis bir zamandaki cikis sadece o zamanda verilmis bir girisin fonksiyonudur. Geri-beslemeli (son gunlerde recurrent olarakta kullanilmaktadir) YSAlarda ise; bazi islem elemanlarinin cikisi, ayni islem elemanlarina veya daha onceki katlardaki islem elemanlarina geri beslenir bundan oturu dinamik bir memoriye sahip olduklari soylenebilir. Verilmis bir anda boyle sinir aglarinin cikisi kendi girislerine ve daha onceki giris ve cikislarina aksettirilir. Sekil 1' de iki girisli, bir cikisli ve bir ara kata sahip basit bir yapay sinir agi sekli gosterilmistir. Agirliklar, isleme elemanlari (sekilde yuvarlak sembollerle



Sekil 1. Bir yapay sinir agi

verilmistir.) arasindaki iletisimi saglarlar. Isleme elemenlarinda kullanilan trasfer fonksiyonlari turevi alinabilir fonksiyonlar olmalidir. Sigmoid fonksiyon buna bir ornek olarak verilebilir.

YSAlari cok cekici hala getiren ozellikler su sekilde siralanabilir.

- Hizli real-time performansi,
- Genelleme yapma ozelligi ve bu ozelliginden dolayi gurultuye karsi olan toleransi,
- Daha az bilgiye ihtiyac duymasi,
- Bilinmeyen procesleri tanimlama ve kontrol edebilme ozelligi,
- Bir cok problemlere uygulanabilir olmalari,
- Konventional tekniklerin yetersizligi.

Bunun yaninda, YSAlarin robotikteki uygulamalarinda karsilasilan guclukler bazi uygulamalarda cok yavas ogrenme, problemlerin cok karmasikligindan dolayi cok fazla giris bilgisine ihtiyac duymalari ve bazi uygulamalarda da hassas sonuc alinamamasi olarak siralanabilir.

3. ROBOTIKTEKI PROBLEMLER

Giris bolumunde de aciklandigi gibi, robotikteki problemleri cozmek icin karmasik ve yogun bir matematik hesaplamaya gerek vardir. Bazen en basit bir robot hareketini hesaplamak bile cok yogun bir matematik gerektirebilir. Robotikteki problemler asagidaki sekilde siniflandirilabilir :

- Kinematik,
- Dinamik,
- Kontrol,
- Yorunge ve gorev planlama,
- Algilama.

Robotikte, Yorunge ve gorev planlama vede Algilama [14,15] onemli olmasina ragmen, bu makalede yer verilmemistir. Burada incelenmis ilk uc problem hakkinda ayrintili bilgi asagidaki bolumlerde verilecektir.

3.1. Robot Kinematik

Kinematik, harekete sebebiyet veren nedenleri goz onunde bulundurmadan robot baglanti hareketlerini cozumlemeye imkan saglar. Robot kinematigi, ileri (forward) kinematik ve ters (inverse) kinematik olmak uzere iki kisma ayrilir. Ileri-kinematikle, manipulatorun sonundaki etkili noktanin (end-effector), pozisyon ve acisinin statik problem olarak hesaplanmasi yapilir. Ters-kinematik ile etkili noktanin, verilmis pozisyon ve acisinin butun baglanti noktalarinin cozum seti hesaplanir.

3.2. Robot Dinamik

Robot dinamigi, robot baglanti koordinatlari, hizlari ve ivmeleri ile baglanti torklari arasindaki iliskiyi formuluze eder. Matematiksel formulasyonlar (Lagrange-Euler, Newton Euler) sebebiyle, karmasik trigonometri gerektirirler.

3.3 Robot Kontrol

Robot kontrol, robotigin diger bir onemli alanidir ve iki kisma ayrilir. Birincisi, pozisyon kontroludur. Pozisyon kontrolu, bir sistemin parametrelerindeki hatayi otomatik olarak telafi eder ve sistemi rahatsiz eden karisikliklari yok eder. Ikincisi ise, kuvvet kontroludur. Bu pozisyon kontrolunun tamamlayicisidir. Robot bir parcaya, calisma yuzeyine veya bir nesneye dokundugu zaman goz onune alinir. Daha once bahsedildigi gibi, robotikte kontrol onemli olmasi sebebiyle kullanilan kontrolorunde onemi yuksektir. Kontrolor, verilmis bir hedefe ulasilmasi icin bir kontrol parametresi seti saglamalidir. Bunun yaninda kontrolorun optimal kontrol yapmasi icin sistem parametrelerini bilmesi gereklidir. Kontrolu guclestiren bir kac durum vardir; sistemin modeli mevcut olmayabilir, sistem zamanla degisebilir veya kontrolor zamanla degisebilir.

4. ROBOTIKTE YAPAY SINIR AGLARININ UYGULAMALARI

Ikinci bolumde de bahsedildigi gibi ogrenme, genelleme yapma ve hizli hesaplama kapasitesinden dolayi, YSAlarin real-time uygulamalari robotikte yaygin olarak kullanilmaktadir. YSA'nin robot manipulatorlerindeki kinematik, dinamik ve kontrol problemleri cozmede gosterdigi performans yuksektir. Bundan sonraki bolumlerde, yukarida bahsedilmis problemlerinin cozumune yapay sinir aglarinin nasil katkida bulundugu, herbir problem icin ayri ayri basliklar altinda incelenecektir.

Bugun icin bilinen yirmi altidan fazla sinir agi modeli bulunmasina ragmen, bugun robotik uygulamalarda en cok kullanilan YSAlar Backpropagation ve Hopfield agi dir. Bunun yaninda daha bir cok sinir aglari robotikteki uygulamalarda kullanilmaktadir. Mesela; Competitive ve Cooperative aglar, ART1 (Adaptive Reasoning Theory), Kohonen self-organize ag, Degistirilmis Counter-Propagation ve Functional ag bunlara ornek olarak verilebilir.

4.1. Robot kinematiginde yapay sinir aglari uygulamalari

Daha once robot kinematigi bolumunde bahsedildigi gibi, kinematik problemi yogun hesaplamaya ihtiyac duymasi ve bir cok cozum seti bulunmasindan dolayi cozumu zor bir problemdir. YSAlar bu yogun hesaplamayi dusurmede basari saglamislardir.

Guez ve Ahmad [16], iki-ve-uc-serbestlik derecesine sahip bir manipulatorun ters kinematik problemini cozmede cok katli perceptron kullanmislar ve YSAlarin bir manipulator icin iyi bir baslangic degerleri tahmin edici olarak kullanilabilecegini ortaya koymuslardir. Bunun yaninda ayni probleme Hopfield ag uygulandiginda [17] daha iyi sonuc alinmistir. Iberall [18,19] ve [20], bir robot elinin parmaklarinin ters-kinematik problemini cozmek icin degisik YSAlar kullandilar. Bu problemi cozmede kullanilmis degisik sinir aglari, Functional ag, Kohonen ag ile birlestirilmis Counter-propagation ag ve mehsur Backpopagation ag ve bu aglarin performanslarinin karsilastirilmasi [21] de yapilmistir. Kartezyen pozisyonunu ve orientasyonunu bulmak icin bir iki yonlu sinir agi [22] kullanilmistir. Sonuclar gostermistir ki kullanilan aglarin performansi konvensiyonel cozum tekniklerinden ve Ileri-beslemeli ag ile ters-kinematik probleminin cozumleri [16,23]'den cok daha iyidir.

Robot kinematigi, pratik olarak gerceklestirilmesi guc ve komplex bir robotik problemidir. YSAlar bu problemlerin cozumunde yeni bir metodtur ki pozisyonel tekrarlamalari dusururler ve sonuclarin dogruluk yuzdesini artirirlar. Yapay sinir agi, sistemin guvenilirligini arttirmak icin baglanti acilari arasindaki teorik olarak elde edilen sonuc ile arzu edilen sonuc arasindaki hatayi sifira dusurmeye calisir [26]. Bir degisik uygulamada [27], Kohonen'in self-organize algoritmasi bir Widrow-Hoff tipi hata duzeltme kurali olarak kullanilmistir. Bu metod ilk kez [28] tarafindan ters-kinematik problemini cozmek icin kullanilmistir.

Kullanilmis coklu bir sinir agi [24], bir Stewart platformunun ters-kinematik problemini cozmede iyi bir performans gostermistir. Bir kompleks fonksiyon yeni bir polinomik sinir agi kullanilarak [25] de cozulmustur. Diger bir uygulamada [80] dir.

Sonuc olarak, Robot kinematik problemlerini cozmede YSAlari kullanmanin avantajlari; Real-time operasyonlarda kullanilabilme kapasiteleri, bir cozumun icin ele alinan hesaplama zamaninin linklerin sayisina bagimli olmamasi ve programlanmaya ihtiyac duymamalari olarak siralanabilir.

4.2. Robot dinamiginde yapay sinir aglari uygulamalari

Sinir aglarinin haritalama ozelligi, robot dinamigi icin çok ilgi cekicidir. Burada sinir aglari, sistemin dinamigi ve ters dinamigini ogrenir sonra da bu bir ters dinamik kontrolor olarak kullanilir.

Bir CMU Direkt-Drive robot kolu II'nin ilk iki baglanti noktasindaki dinamik tesirlerin modellenmesi ve performansinin olculmesi guctur. Iste bu robot dinamik tesirleri telafi etmek icin bir Backpropagation sinir agi kullanilmistir [29]. Bu ilgi cekici uygulama sinir aglarini robot kolu kontrolunde daha populer hale getirmis ve sinir aglari gercek bir robot kolun ters dynamigini ogrenmede [30] kullanilmistir. Burada YSA, bir tek yorungeyi ogrendikten sonra, yorungeyi genellestirmis ve oldukca degisik yorungeler icin kabuledilebilir sonuclar vermistir. Bununla birlikte, tek katli ve cok katli YSAlar verimli bir sekilde sanki robotun dinamik modeli gibi [31] robot manipulatorun kontrol edilmesinde kullanilmistir.

Diger taraftan, YSAlarla dinamik sistem tanimlama [63] son zamanlarda elde edilen yuksek basaridan dolayi sistem kontrolu icin oldukca populerdirler [32,33]. Pham ve Liu [34]'nun onermis olduklari degisik YSAlar belki robotikteki uygulamalar icin degisik bir alternatif olabilir.

YSAlarin avantajlari; Modellenmesi veya tanimlanlansi guc olan sistemlerin tanimi, daha az bilgiye ihtiyac duymalari, daha hizli olmalari ve uygulanan giris degerlerine gosterdikleri tolerans olarak verilebilir.

4.3. Robot Kontrolundeki yapay sinir aglari uygulamalari

Yapay sinir aglarinin, robot kontrolunde yaygin olmasinin sebebi gosterdigi performanstan kaynaklanir. Degisik kontrol ogrenme teknikleri ve onlar hakkinda ayrintili bilgiler [35,36,37,38] verilen kaynaklardan elde edilebilir.

Kontrol edilecek objenin (robot manipulator) tam dinamik davranisi genellikle bilinmez. Kullanilan sinir aglari [39] veya diger ogrenme algoritmalari ile bir hareketin olusmasi esnasinda ters robot dinamigi ogrenilebilir. Bunun yaninda, YSAlar, deneyimlerden ogrenme, degisen cevre sartlarina uyum saglama ve onlara tepki gostermeyi de ogrenirler [40,41,42,43].

Genel robotik kontrol icin, Albus [44] kendi modeli olan CMAC'i onerdi. Bu model bolunerek duzenlenmis bir tablo (look-up-table) metodu olarak tanimlanabilir. Miller [45], hesaplanmis bir tork kontrolore CMAC'iuyguladi ve simulasyon neticelerinden gercekten cok etkileyici sonuclar aldi. PUMA 560 robot icin kullanilan YSA [46] dadir. Bunun yaninda adapte edilmis bir model uzerine kurulmus YSA kontrolor [47], robot hareketi sirasindaki tasidigi yukun miktarini kararlastirmada kullanilir. Bir baska uygulamada da Bavarin, [37] de verdige bir kac ornekte, gurultulu giris(noisy) veya giris degerlerinin eksik olarak verilmesine ragmen kabuledilebilir sonuclar elde etmistir. Belirtilen bu ozellikler YSAlari, zeki kontrol icin cok cekici yapar.

YSAlarin basariyla uygulanmalarina daha fazla ornek gosterilebilir [48,49,50,51]. Backpropagation aginin basarili uygulamalari starford-like robot icin [52] ve uzay robotlari icin [53] verilmistir. Multi-link manipulatorlerin real-time kontrol icin farkli bir neuro-kontroler [54]'de verilir. Intelledex 605T robot manipulatoru icin baska bir real-time uygulamasida [55]'de bulunabilir. YSAlari kullanma avantajlari da ve Mars robot uygulamalari [56] da verilmistir.

Miyamoto ve meslektaslari [57], ozellikle robotlar icin kontrol sistem dizayninda neuro-psikologikal kavramlarin onemini vurguladilar. Burada YSAlar bir endustriyel robotun yorunge kontrolunu ogrenmede ters-dinamik model gibi kullanilmistir. YSAnin, ogretilmis hareketleri genelleme kapasitesinden dolayi verilmis sinir agi ne bir modele nede parametre ayarlanmasina ihtiyac duyar.

Kendi kendini kontrol eden bir mobil robotun kontrolunde backpropagation ag kullanildi [58]. Hopfield ag kullanilmis bir neural estimator uygulamasi [59] de bulunabilir. Barto ve arkadaslari [60], cart-pole problemini cozmek icin iki ag onerdiler. Buna benzer bir problem [18] de ele alinmis ve sonucta sistemin herhangi bir bilgisine ihtiyac duyulmaksizin daha kararli, daha hizli ogrenme ve daha kucuk izleme hatasi elde edilmistir.

Elsley'in [61] Backpropagation metodu kullanarak elde ettigi sonuc cok ilginctir. Kullanilmis YSA kontrolorun performansi konvensiyonel Jacobian kontrolorden daha iyidir. Buna benzer olarak diger bir calismada ise [62], uc degisik YSA kontrolorun karsilastirilmasi yapilmis birbirlerine olan avantajlari ve dezavantajlari verilmistir.

Hiyerarsik yapay sinir aglarinin yapilari [24] de sunulmus ve YSAnin hiz kapasitesi, adaptasyonu ve hesaplama avantaji bir ornekle gosterilmistir. Buna benzer, gercekten cok ilginc olan bir baska uygulamada [41] de yapilmistir. Cops-and Robbers oyunundaki bazi aliskanliklar yapay sinir aglari tarafindan ogrenilmis (Mesela; ele gecirme ve kacma gibi) ve robotlar kendi kendilerini kontrol etmislerdir.

YSAlarin, kola benzer manipulatorlerin kontrolunde uygulamalari da bir hayli coktur [42,64,65,66,67,68]. Bununla birlikte, YSAlar multi gorev gerektiren yerlerde de son zamanlarda kullanilmaya baslandi [41, 69]. Buna bir ornek MARVIN mobil robotu verilebilir [69]. Bu robotun gorus algilama, patern ogrenme, duygusal durumlar, davranis hareketi ve motor kontrolunda yapay sinir aglari uygulamalari verilmistir.

Son gunlerin popular bir uygulama alanida sualti robot araclaridir. Bu konuda goze carpan uygulamalar [70,71,72] ve uygulanan YSAlar hakkinda ayrintili bilgiler verilen kaynaklardan elde edilinebilinir. YSAlarin diger yapay zeka teknikleriyle beraber kullanilmasiyla ilgili yapilmis robot kontrol uygulamalarida bulunmaktadir [73,74,75].

YSAlarin avantajlari; Hiz kapasiteleri, adaptasyon ozellikleri, hesaplama avantaji, daha az sistem bilgisine ihtiyac duymalari, kontrol etme kabiliyetlerinin yuksek olusu ve gurultuye gosterdikleri tolerans olarak siralanabilir.

5. TARTISMA ve ANALIZ

Yukarida verilen misallerdende anlasilacagi gibi, YSAlarin robotikteki problemlerin cozumune katkilari fazladir. Verilmis uygulamalardan gorulebilecegi gibi, YSAlarin robot kontrol, dinamik ve kinematik problemlerinin cozumunde basarili olduklari ve gelecekte bu basarilarini daha da artiracaklari soylenebilir.

YSAlarin robotikte kullanilma sebebleri, sagladigi avantajlar ve dezavantajlardan daha once bahsedildigi icin burada tekrar bahsedilmemistir.

Son olarakta su soylenebilir ki hibrit yapay sinir aglarinin ve/veya sinir aglarinin diger yapay zeka metodlariyla kullanilmasi (Uzman sistemler, Fuzzy, Genetik Algoritma), YSAlari robotikte daha da cok populer hale getirecektir [50,51,54,72,73,74,76,77,78,79].

6.TESEKKUR

Yazarlar, Seref Sagiroglu'na verdigi destekten dolayi Erciyes Universitesi'ne tesekkur eder.

7. KAYNAKLAR

1. Werbos, P.J., Neural Networks for Robotics and Control, *WESCON 89, IEEE Conf.*, San Francisco, CA, 33 (Nov. 1989) 699-708.
2. Coiffet, P. and Chirouze, M., *An Introduction to Robot Technology*, (Kogan Page, London, 1983).
3. Kung, S.-Y., and Hwang, J.-N., Neural Network Architecture for Robotic Applications, *IEEE Trans. on Robotics and Automation*, 5 (1989) 641-657.
4. Kosko, B., Robotics and Neural Networks, *Neural Networks for Signal Processing*, (Prentice-Hall, New Jersey, 1992) 161-187.
5. Horne, B., Jamshidi, M. and Vadiee, N., Neural Networks in Robotics, *Journal of Intelligent and Robotic Systems*, 3(1) 1990 51-66.
6. IEEE, Proc. Conf. on Decision and Control, Honolulu, HI, Dec.1990, Vol.6.
7. Simpson, P.K., *Artificial Neural Systems*, (Pergamon Press, Oxford, 1990).
8. Kocabas, S. A Review of Learning, *The Knowledge Engineering Review*, 6(3) 1991 195-222.
9. Widrow, B. and Lehr, M.A., 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation, *Proceedings of the IEEE*, 78(9) Sep.1990 1415-1442.
10. Lippmann, R. P., An Introduction to Computing with Neural Nets, *IEEE ASSP Magazine*, (April 1987) 4-22.
11. Najim, K. and Oppenheim, G., Learning Systems: Theory and Application, *IEE Proceedings-E*, 138(4) July 1991 183-192.
12. Maren, A., Harston, C. and Pap, R., *Handbook of Neural Computing Applications*, (Academic Press, London, 1990) ISBN 0-12-471260-6.
13. Fumelhart, D.E and McClelland, J.L., *Parallel Distributed Processing*, Vol.1 (MIT press, Cambridge, MA, 1986)
14. Masory, O. and Aguirre, A.L., Sensor Calibration Methods - Performance study, *Applications of Artificial Neural Networks, SPIE*, 1294 (1990) 490-501.

15. Samast, Y., Colour in Artificial Vision, *Technical Report*, Erciyes University, Electronics Eng. Dept., April 1992.

16. Guez, A. and Ahmad, Z., Solution to The Inverse Kinematics Problem in Robotics by Neural Networks, *IEEE Conf. on Neural Networks*, San Diego 2 (1988) 617-624.

17. Guo, J. and Cherkassky, V., A Solution to the Inverse Kinematic Problem in Robotics Using Neural Processing, *Int. Joint Conf. on Neural Networks, IEEE; INNS*, Washington D.C., 2 (June 18-22 1989) 299-304.

18. Iberall, T. , A Ballpark Approach to Modelling Human Prehension, *IEEE Conf. on Neural Networks*, Los Angeles, CA, 4 1987 535-544.

19. Iberall, T. , A Neural Network for Planning Hand Shaped in Human Prehension, *IEEE Conf. on Decision and Control*, (1987) 2288-2293.

20. Amari, S. and Arbib, M., Competition and Cooperation in Neural Nets, in: *Systems Neuroscience*, Metzler, J. (Ed.), (Academic Press, New York, 1977) 119-165.

21. Nguyen, L., Patel, R.V. and Khorasani, K., Neural Networks Architectures For The Forward Kinematics Problem in Robotics, *International Joint Conference on Neural Networks, IEEE, INNS*, San Diego, California, 3 (17-21 June 1990) 393-399.

22. Lee, Sukhan and Kil, R.M., Robot Kinematic Control Based on Bidirectional Mapping Network, *International Joint Conference on Neural Networks, IEEE, INNS*, San Diego, California, 3 (17-21 June 1990) 327-335.

23. Ahmad, Z. and Guez, A., On the Solution to the Inverse Kinematic Problem, *Proceedings IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, 3 (1990) 1692-1697.

24. Geng, Z. and Haynes, L. Neural Network Solution For the Forward Kinematics Problem of a Stewart Platform, *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, California, (April 1991) 2650-2655.

25. Chen, P.C. and McAulay, A. D., Robot Kinematics Learning Computations Using Polynomial Neural Networks,*Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, California, (April 1991) 2638-2643.

26. Josin, G.,Charney, D. and White, D., Robot Control Using Neural Networks, *IEEE Int. Conf. on Neural Networks*, San Diego, California, 2 (July 24-27 1988) 625-631.

27. Kieffer, S., Morellas, V. and Donath, M., Neural Network Learning of the Inverse Kinematic Relationships for a Robot Arm, *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, California, (April 1991) 2418-2425.

28. Ritter, H., Martinetz, T. and Schulten, K. , Three-dimensional Neural Net for Learning Visuomotor Coordination of a Robot Arm, *IEEE Transactions on Neural Networks*, 1(1) March 1990 131-136.

29. Goldberg, K. and Pearlmutter, B., Using a Neural Network to Learn the Dynamics of the CMU Direct-Drive Arm II, *Technical Report*, Carnegie Mellon University, August 1988, CMU-CS-88-160.

30. Kawato, M.Y., Uno, Y., Isoba, M. and Suzuki, R., Hierarchical Neural Network Model for Voluntary Movement with Application to Robotics, *Proceedings of the IEEE International Conference on Neural Networks, IEEE*, New York, NY, (June 1987).

31. Katic, D., Using Neural Network Model for Learning Control of Manipulation Robots, *International Conference on Intelligent Autonomous Systems 2* , Amsterdam, Dec.1989 424-433.

32. Narendra, K.S. and Parthasarathy, K., Identification and Control of Dynamic Systems Using Neural Networks, *IEEE Trans. Neural Networks* 1(1) 1990 4-27.

33. Yamada, T. and Yabuta, T., Plant Identification Using Neural Network, *Symposium on Flexible Automation JAPAN-USA*, Kyoto, July 1990.

34. Pham, D. T. and Liu, X., Neural Networks for Discrete Dynamic System

Identification, *J. of Systems Engineering*, 1(1) 1991 51-61.

35. Psaltis, D.,Sideris, A. and Yamamura, A., Neural Controllers, *Proceedings of the IEEE First Int. Conf. on Neural Networks*, San Diego, CA, 1987 551-558.

36. Psaltis, D., Sideris, A. and Yamamura, A., A Multilayer Neural Network Controller, *IEEE Control Systems Magazine*, 8(2) 1988 17-21.

37. Bavarin, B., Introduction to Neural Networks for Intelligent Control, *IEEE Control Systems Magazine*, 8(2) 1988 3-7.

38. Chen, F.-C, Back-Propagation Neural Networks for Nonlinear Self-Tuning Adaptive Control, *IEEE Transactions on Robotics and Automation*, 5(5) 1989.

39. James, C. and Yan L., Neural Networks Learning Applied to the Control of Unknown Systems, *Electro International Conference Record*, NY, 16 (1991) 574-579.

40. Mel, B. W., MURPHY: A Robot that Learns by Doing, *American Institute of Physics*, 1988 544-553.

41. Nagata, S., Sekiguchi, M. and Asakawa, K., Mobile Robot Control by a Structured Hierarchical Neural Network, *IEEE Control Systems Magazine*, April 1990 69-76.

42. Rabelo, L.C. and Avula, X.J.R., Hierarchical Neurocontroller Architecture for Intelligent Robotic Manipulation, *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, California, April 1991 2656-2661.

43. Hanes, M. D., Neural Network Control of Force Distribution for Power Grasp, *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, California, (April 1991) 746-751.

44. Albus, J., A New Approach to Manipulator Control: the Cerebellar Model Articulation Controller (CMAC), *J. of Dynamic Systems, Measurement, and Control*, Sept.1975 228-233.

45. Miller III, T. W., Glanz, F.H. and Kraft III,L.C., Application of General Learning Algorithm to Control of Robot Manipulators, *The International Journal of Robotics Research* 6(2) 1987.

46. Guez, A. and Ahmad, Z., Accelerated Convergence in the Inverse Kinematics via Multilayer Feedforward Networks, *Int. Joint Conf. on Neural Networks, IEEE, INNS*, Washington D.C. 2 June 18-22 1989 341-344 .

47. Johnson, M.A. and Leahy Jr, M.B., Adaptive Model-Based Neural Network Control, *Proceedings IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, 3 1990 1704-1709.

48. Helferty, J.J. and Biswas, S. Neuromorphic Control of Robotic Manipulators, *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, California, April 1991 2436-2441.

49. Guez, A. and Selinsky, J., A Trainable Neuromorphic Controller, *Journal of Robotic Systems*, 5(4) 1988 363-388.

50. Pourboghrat, F., Neuromorphic Controllers, *Proceedings of the 28th IEEE Conference on Decision and Control*, Tampa, Florida, Dec. 13-15 1989 1748-1749.

51. Miller III, T.W., Sutton, R.S. and Werbos, P.J.,(Eds.), *Neural Networks for Control*, MIT, London, 1990.

52. Gu, Y.L. and Chan, J.W.M., On Design of Nonlinear Robotic Control System with Neural Networks, *Proceedings IEEE International Conference on Systems, Man, and Cybernetics*, Cambridge, Massachusetts, Nov.14-17 1989 200-205.

53. Hong, R. and Liu, J.S., Imaging and Controls for Mars Robots with Neural Networks, *Electro International Conference Record*, NY, 16 1991 596-600.

54. Swiniarski, R., A Neuromorphic Cascaded Context Sensitive Controller for Robot Manipulator, *Int. J. Conf. on Neural Networks*, Washington DC, 2 15-19 Jan.1990 387-392.

55. Sobajic, D. J., Lu, J.J. and Pao, Y.H., Intelligent Control of The Intelledex 605T Robot Manipulator, *IEEE Int. J. Conf. on Neural Networks*,

San Diego, California, 2 June 1988 633-640.

56.Hong, R., State-Of-The-Art of Artificial Neural Networks and Applications to Mars Robots, *Electro International Conference Record*,NY, **16** 1991 568-573.

57.Miyamoto, H., Kawato, M., Setomaya, T. and Suzuki, R., Feedback-Error-Learning Neural Network for Trajectory Control of a Robotic Manipulator, *Neural Networks*, 1 (1988) 251-265.

58.Berns, K., Dillman, R. and Hofstetter, R., An Application of a Backpropagation Network for the Control of a Tracking Behavior, *Proceedings of the 1991 IEEE Int. Conf. on Robotics and Automation*,Sacramento, California, April 1991 2426-2431.

59.Guez, A, Eilbert, J.L. and Kam, M., Neural Network Architecture for Control,*IEEE Control Systems Magazine*, April 1988 22-25.

60.Barto, A.G., Sutton, R.S. and Anderson, .W.., Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems, *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-13**(5) Sept./Oct . 1983 834-846.

61.Elsley, R.K., A Learning Architecture for Control Based on Back-propagation Neural Networks, *Neural Networks IEEE Int. Conf.*, San Diego, July 1988 587-594.

62.Wilhelmsen, K. and Cotter, N., Neural Network Based Controllers for a Single-Degree-of Freedom Robotic Arm, *Int. J. Conf. on Neural Networks*, Washington D.C., 2 15-19 Jan.1990 407-413.

63.Pham, D.T. and Sagiroglu, S., Identification of Linear Dynamical Systems Using Backpropagation Neural Networks, *Modelling and Scientific Computing*, Special Issue on Neural Networks for Identificaion and Control of Dynamical Systems, 1992 (in press).

64.Fukuda, T., Shibata, T., Tokita, M., Mitsuoka, T., Neural Network Application for Robotic Motion Control : Adaptation and Learning, *Int. Joint Conf. on Neural Networks, IEEE, INNS*, San Diego, California, **3** 17-21 June 1990 447-451.

65.Liu, H., Iberall, T. and Bekey, G.A., Neural Network Architecture for Robot Hand Control, *IEEE, Control Systems Magazine*, April 1989 38-43.

66.Xu, G., Scherrer, H.K. and Schweitzer, G., Application of Neural Networks on Robot Grippers, *International Joint Conference on Neural Networks, IEEE, INNS*, San Diego, **3** 17-21 June 1990 337-342.

67.Martinetz, T.M. and Schulten, K.J., Hierarchical Neural Net for Learning Control of a Robot's Arm and Gripper, *Int. Joint Conf. on Neural Networks, IEEE, INNS*, Washington D.C., 2 18-22 June 1989 747-752.

68.Ciliz, K., Time Optimal Control of Mobile Robot Motion Using Neural Nets, *Proc. IEEE Int. Symp. on Intelligent Control*, Albany, N.Y., 25-26 Sep.1989 368-373.

69.Saxon, J.B. and Mukerjee, A., Learning the Motion Map of a Robot Arm with Neural Networks, *Int. J. Conf. on Neural Networks*, Washington D.C., 2 15-19 Jan.1990 777-782.

70.Yuh, J.K., A Neural Net Controller For Underwater Robotic Vehicles, *IEEE Journal of Oceanic Engineering*, 15(3) (July 1990) 161-166.

71.Sanner, R.M. and Akin, D.L., Neuromorphic Pitch Attitude Regulation of an Underwater Telerobot, *IEEE Control Systems Magazine*, (April 1990) 62-67.

72.Fujii, T. and Ura, T., Neural-Network-Based Adaptive Control Systems for AUVs, *Engng. Applic. Artif. Intell.*, 4(4) (1991) 309-318.

73.Pham, D. T. and Karaboga, D., Design of Neuromorphic Fuzzy Controllers, *Journal of Systems Engineering*, 2(4) 1992 (in press).

74.Handelman, D. A., Lane, S.H.and Gelfard, J. J., Integrating Neural Networks and Knowledge-Based Systems for Intelligent Robotic Control, *IEEE Control Systems Magazine*, April 1990 77-86.

75.Baloch, A.A. and Waxman, A.M., A Neural System for Behavioral Conditioning of Mobile Robots, *Int. J. Conf. on Neural Networks*, Washington D.C., 2 (Jan.1990) 723-728.

76. Fukuda, T., Shibata, T., Tokita, M. and Mitsuoka, T., Adaptation and Learning for Robotic Manipulator by Neural Network, *Proceedings of the 29th IEEE Conference on Decision and Control*, Honolulu, HI, USA, 6 (5-7 Dec.1990) 3283-3288.

77. Pomerleau, D. A, Gowdy, J. and Thorpe, C.E., Combining Artifical Neural Networks and Symbolic Processing for Autonomous Robot Guidance, *Engng. Applic. Artif. Intell.*, 4(4) (1991) 279-285.

78. Pham, D.T., Artificial Intelligence in Manufacturing, *Proc. Sunderland Adv. Manuf. Tech. Int. Conf.*, Sunderland, UK, Apr.1992.

79. Pham, D. T. and Guner, U.F., Bilgi Tabanli Yaratici Kavramsal Dizayn (in Turkish), *The First Turkish Symposium on Artificial Intelligence and Neural Networks*,Ankara, Turkey, 25-26 July 1992.

80. Lin, C.T. and Lee, C.S.G., Fault-Tolerant Reconfigurable Architecture for Robot Kinematics and Dynamics Computations, *IEEE Transactions on Systems, Man, and Cybernetics*, 21(5) Sep/Oct.1991 983-999.

YAZAR DİZİNİ / AUTHOR INDEX