The Logical Structure of the Cognitive Mechanisms Guiding Psychological Development.

by

George Osborne

Wolfson College
Cambridge

A Thesis submitted for the Degree of Master of Philosophy at the University of Cambridge

August 1995

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration. Reference is made throughout the text to the sources and extent to which I have availed myself of the work of others.

No part of this thesis has been submitted for a degree, diploma or similar qualification at any other university.

George Osborne

Preface

This thesis is based on research carried out at the Cavendish Laboratory of the University of Cambridge, from October 1994 to August 1995. I am grateful to my supervisor, Professor Brian Josephson, for giving me the opportunity to work under his guidance and for helping me master the transition from course-work to research work.

I have learned a great deal in countless discussions with Brian, but also from several members of the Theory of Condensed Matter Group.

Above all I wish to thank my family for their constant encouragement and support both during the application process and during my stay in Cambridge.

Contents

List of Figures

Chapter 1

Introduction

The purpose of this MPhil. project is to simulate the processes by which psychological development occurs. More specifically it concerns modelling the ability for *novice* systems with only very limited behaviour or understanding, to develop into much more sophisticated *expert* systems, through a process of adaption and self-organisation.

Examples of such development abound, from infants learning to walk, talk and grasp, to adults learning to operate new technology. In each of these, the individual is presented with a challenging situation to which he adapts and eventually masters.

Psychology has provided a number of theoretical frameworks to help explain the mechanism of this development. A review of this work and its potential for use in a computer simulation is carried out in chapter 2.

The ability to innovate and produce structured but also novel solutions to such problems is of key importance in the process of adaption. However, the production of such solutions has been difficult to achieve with conventional computer simulations. A review of the main conventional approaches to the computer simulation of psychological development, in general termed artificial intelligence (AI), is given in chapter 3.

More recently, the failings of conventional computer simulations has led to alternative approaches being proposed. Two important examples, Emergent Memory Models and Modular Neural Networks, are examined in detail in chapter 4. These attempt to harness , rather than simply exploit, the phenomenon of *emergence*, in order to reproduce the structured and novel approach to development present in humans but found lacking in conventional AI.

Emergence, the production of fundamentally new behaviour in a system, is felt to be an essential feature of a model of psychological development. This assertion stems from the similarity of psychological theories of development - chiefly Piaget's genetic epistemology - with theories on the formation of emergent phenomena. For instance hyperstructure theory predicts the formation of hierarchies of emergent structures given the right conditions. The phenomena of emergence and the proposals of hyperstructure theory are addressed in chapter 5.

A computer simulation is proposed which takes into account all of the above mentioned factors. Novel structured solutions to developmental problems are formed as emergent phenomena. From a basis set of schemes - analogous to the modules in Modular Neural Networks - emergent hyperstructures of schemes are formed as a solution to a given developmental problem. This is achieved using the Bucket Brigade algorithm (used in the Emergent Memory Model reviewed) to create sequences that solve particular aspects of the problem, forming new, emergent schemes. The resulting system is found able to solve the inverse pendulum problem using a second order emergent structure, and has great potential for further development.

Chapter 2

Psychological Theories of Development

2.1 Introduction to Modern Developmental Psychology

Developmental psychology aims to provide a scientific understanding of age-related change in human mental experiences and behaviour. Although most developmental theories have been specifically concerned with children, the ultimate aim is to provide an account of development throughout the lifespan.

Before the start of the Twentieth-Century, theories of development were nothing more than anecdotal descriptions. For example John Locke (1632-1704) considered each child to be born a *tabula rasa* (blank slate) whose mind would develop solely according to experience. Thus a new-born baby is considered to be psychologically structureless, denying any importance of innate factors in its psychological development.

This is in sharp contrast to the views of Prussian philosopher G. W. Leibnitz (1646-1716) who considered all knowledge to be derived from innate structures with which every child is born. As such, all knowledge was considered to be derived from the exercise of reason and purported to give an absolute, objective, description of the world.[1]

The application of rigorous systematic testing has put theories of psychological development on a more scientific base. In addition, a number of psychologists have attempted to provide a theory of development to interrelate the roles of 'nurture' and 'nature' which previously formed two mutually exclusive bodies of theory.

Lev Vogotsky(1896-1934), for example, was concerned to show how culture and in particular language, influences the course of intellectual development. Whereas

[1] Leibnitz belonged to the school of thought now generally labelled 'rationalist' and Locke to the school of 'empiricism'. An attempt to consolidate these opposing views was later made by Kant[1], whose work was to be an influence on the Swiss psychologist Piaget.

John Bowlby (1907-1990) was primarily concerned with the role family bonding on a child's emotional development.

However, it is the work of Jean Piaget[15, 49, 50] (1896-1980) that is of greatest interest. This is due to the ability of the ideas proposed to lend themselves well to a number of ideas in complexity theory (see section 5) as well as to the processes of computer simulation.

2.2 The Work of Jean Piaget

Jean Piaget studied psychology with the intention of deriving a biologically orientated theory of the origins of knowledge. This was inspired by his basic concern with the adaption of living things to their environment. As a special case of this biological problem, he was led to consider the phenomenon of human knowledge, especially formal, logical knowledge, which acts to transcend the limitations of human physical embodiment.

The resulting 'Genetic Epistemology' attempts to describe the acquisition of knowledge as a constituent part of the process of biological evolution.

2.2.1 Piaget's Methodology

An understanding of Piaget's methodology is important in understanding his theories on epistemology.

Piaget addresses the problems of knowledge acquisition through a process of dialectical argument, to which he refers as 'dialectic constructionism'. This involves passing an argument from a *thesis* to a contradictory *antithesis*, and thence to a synthesis - which in turn acts as a new thesis.

This approach is applied by Piaget to the theoretical traditions of epistemology within biology, psychology and philosophy. In each case, he says, the thesis posits *structuralism without genesis*, the antithesis *genesis without structure* and Piaget's own synthesis offers *structure with genesis*.

As a consequence of his dialectic approach, Piaget proposes the association of a continual equilibration with the progress of the developmental process.

2.2.2 Piaget's Psychology

The process of equilibration manifests itself, in Piaget's description of psychology, in the form of *assimilation* and *accommodation*. These two are polar opposites, analogous to the 'thesis' and 'antithesis' outlined above.

Assimilation is the modification of an incoming stimulus by the activity of a pre-existent mental structure. Such structures are denoted 'schemes', and represent the regular structure of an action. For example, the general method involved in throwing a ball: the unique circumstances in which person and ball may start are *assimilated* by the ball throwing scheme, producing a throwing action.

Accommodation is the active modification of the structure - scheme - itself, so as to *adapt to* the situation. For example, the lifting of an unfamiliar object: an initial attempt is made to assimilate the object into a general lifting scheme. If this fails - the object cannot be lifted by this method - then an *accommodation* takes place; the lifting scheme is adapted, for instance by adjusting hand position to get a better grip, and then re-applied. This adjusting procedure - accommodation - is applied until the operation is successful.

Equilibrium is a relatively stable state of a structure, in that it can accept and adapt to varied input without any essential change. This is the case when given a routine, well practiced task to do, for example driving a familiar car or signing a well practiced signature.

Further discussion on the nature of scheme development is covered in section 2.3.

These ideas on development are extended by Piaget to their biological origins, fueled by his conviction that the biological problem of adaption relates to the epistemological problem of the acquisition of knowledge. He builds on the work of C. W. Waddington[17] to form a theory of 'evolution by epigenetic assimilation'. This considers a person to develop according to his particular genotype, encoded as DNA. However, it is the *interaction* of the developing person with the environment - via the dialectic processes described above - that determines phenotype. This bears a resemblance to the various process that lead to emergent behaviour in complex systems, an important point discussed in section 5.

Piaget sees the development of intelligence in successive psychological stages. Each new stage is derived from the interactions of the previous one with the environment together with the processes of assimilation and accommodation. These stages are ordered roughly in terms of the human subject age at which they are prevalent:

- *The Sensorimotoric Stage:* Active between the ages of 0-2 years. During this period, a child learns, in general, to coordinate muscle actions in an attempt to gain control over its body.

- *The Pre-Operational Stage:* Active between the ages of 2-7 years. This stage requires a number of new skills, learnt during the sensorimotoric stage. Some characteristics of development include muscle actions becoming more elaborate and the ability to make simple mental representations of the world. As such, objects out of immediate sight can still exist in the child's mind.

- *The Operational Stage:* Active between the ages of 7-12 years. By this stage, a child is capable of non-egocentric thought i.e is able to consider points of view other than its own.

- *The Logical Stage:* Active from age 12 years and beyond. The child learns to grasp fully abstract concepts such as mathematics.

2.2.3 The Successes and Failings of Piaget's Genetic Epistemology

Recent experimental research has shown Piaget to be wrong in a number of ways, due to a mixture of poor experimental technique, poor experimenter/subject communication and an inability to appreciate the full implications of other psychologists work. These problems have cast doubt on a number of his ideas. It is important to examine these criticisms to ensure that the more relevant of his ideas are sound. This task was carried out by Boden [49] and is summarised in this section.

The main objection levelled against Piaget is that he has underestimated the cognitive skills of children. For example, the work of Slater, Mattlock and Brown[16] has demonstrated that babies only four months old show surprise when one (hidden) solid object apparently passes through, or jumps over, another instead of moving along a connected, unobstructed pathway. This implies that the four-month-old infant has already developed a perceptual understanding of physical objects that is characteristic of a Piagetian pre-operational two-year-old.

Piaget also underestimates the degree to which a child's intelligence is domain specific. He believed that intelligence is grounded in a developmental sequence of domain general structures, implying a number of distinct stages to development. However a number of experiments[2] have shown this not to be the case. One example is children suffering from Williams syndrome, who have good linguistic skills even though their general intelligence is severely deficient. If language were constructed from the same intellectual structures that underlie reasoning, then such a situation would be impossible.

Following from such evidence, proponents of the 'modularity of mind' psychological stance argue that there are no general principles of intelligence[18]. This theory considers the mind to consists of domain-specific mental modules that are inborn, automatic and unaffected by other modules or thoughts at higher levels. An example of this is the visual cortex, which contains neighbouring columns of cells genetically pre-assigned to particular functions.

Under the modular regime, learning is seen as the reorganisation of unchanging semantic atoms, and development - the construction of radically new forms of representation - is impossible[2].

This view can be criticised on the grounds that the developing mind appears capable of self-organisation over a range of domains. This view is addressed by

---

[2]Recently, this approach has been adopted in the programming of neural networks so as to allow one large problem to be addressed as a collection of smaller, more manageable subproblems. These 'Modular Neural Networks' are discussed in section 4.3.

epigenesis, the aspect of Piaget's genetic epistemology which presents the developing process of the mind as a self-organising interaction between pre-existing mental mechanisms and ecologically relevant aspects of the environment.

It is with a wide range of new psychological and biological evidence[3, 47, 49]that the process of epigenesis, and with it the basic idea of structured psychological development, can be shown to be essentially correct. For instance, Boden[49] presents the case of an infant learning to speak: New-born babies are sensitive to speech in general, whereas infants and older children show increasing sensitivity to their native tongue and decreasing sensitivity to foreign sounds and words. This evidence, taken with others mentioned in the references above, helps justify the dialectical processes of assimilation and accommodation by which epigenetic development occurs.

A further vindication of the role of equilibration and epigenetic self-organisation in development, can be seen in the actions of artificial life simulations[3]. In such simulations the interactions of individual units - individuals in a population each operating under simple, local rules - can lead to the emergence of new structures not explicitly pre-figured in the population. In terms of genetic epistemology, the interactions of individuals may be considered analogous to the process of equilibration between an individual and its environment. Further, from its initial possession of only a set of simple rules - the genetic structure of the individual - the interaction process leads to the emergence, through self-organisation, of fundamentally new structures. Consider, for example, the computer simulated behaviour of birds, which, although programmed as a set of individuals, self-organise into flocks when allowed to interact. This is treated in greater detail in in section 3.3.4.

A strict modular interpretation of the mind does not allow for such a radical change in macroscopic behaviour.

2.3 Scheme Theory

The process of equilibration central to genetic epistemology allows for extensive use in computer simulation. The dialectic interaction with the environment is essentially the same as the mechanism of information feedback by which cybernetic and AI systems function.

These ideas are taken by Eckblad[19], along with notions such as the 'schema' of Bartlett[10] and cell assemblies of Hebb[8], to form 'Scheme Theory'.

2.3.1 The Concepts of Scheme Theory

Scheme theory uses the notion of a *scheme*, conceived as an organised sequence of operations. These interact with the environment to form a central feature of

[3]See section 3.3.3.

a person's behaviour. As with a computer routine, the functioning of a scheme involves input, internal processing and output operations. Eckblad gives the example of a person's habitual way of pouring from a particular jug: The hand is brought into position, fastens on the handle, lifts, tilts and pours - all the way guided by visual and tactile feedback from sensory systems. The term 'scheme' denotes the regular structure of behaviour. In the jug example, the hand travels along many trajectories from many starting points; there is nevertheless regularity in the operations performed and the successive subgoals reached.

2.3.2 The Development of a Scheme

The Piagetian idea of equilibrium is central to the way in which schemes develop. According to scheme theory, a person initially attempts to assimilate a situation by attempting to fit it in to a known scheme. If the situation cannot be successfully assimilated, then the person is thrown into a state of disequilibrium, a state which Eckblad terms 'motivational'. It is in this state that schemes adapt to (accommodate) the situation. The accommodation process may progress in a number of ways, the most relevant to this project being differentiation and re-combination.

Differentiation involves the creation of a new scheme from the re-programming of an old scheme. An example of this is the work of Lasky[47]. He compared the reactions of young and older children when reaching out for an object they could not see. The former were found to be considerably less put off if they could not see their hands than were older children. With differentiation theory, 'seen' and 'felt' objects do not exist separately, nor therefore can the 'seen' hand be differentiated from the 'felt' hand. With development, differentiation will occur and the child will rely on a 'seen' hand to reach a 'seen' object. Older infants, who have made the differentiation, will thus be more put off than younger ones.

The re-combination of schemes is effectively the addition and interaction of previously differentiated schemes which together produce new behaviour.

It is a central assumption of scheme theory that the process of scheme accommodation forms new schemes as solutions to developmental problems and organises them into a hierarchical structure, similar to the lens structure described by Brunswick[48].

2.3.3 Possible Applications of Scheme Theory in AI

Within scheme theory a fully differentiated system can be described as a basis set of schemes. The discrete and well-defined nature of the members of this set makes them ideal for use as building blocks for computational models of psychological development. As such schemes may act as a bridge linking Piagean genetic epistemology to computer simulations of development.

The nature of scheme development depends on the interaction between the scheme and environment. It is the mechanism of this interaction which must be investigated in order to produce any significant results from computer simulations based upon these ideas.

The next section reviews a number of important approaches at producing AI and developmental simulations.

Chapter 3

Conventional Artificial Intelligence

3.1 Introduction

Artificial intelligence can be broadly described as the science of making machines think. Examples of this include holding a conversation, playing chess, writing or translating stories and recognising images. The machines in question are typically computers; but AI is not the study of computers. Rather, it is the study of intelligence in thought and action.

Two fundamental approaches to the modeling of AI have emerged, into which models may be classified:

- Symbol Manipulators - the computer program manipulates symbolic representations of knowledge and logical symbols required for formal reasoning.
- Decentralised Models - the computer program manipulates a distributed array of knowledge data in parallel.

3.2 Symbolic AI

The first computer programs to demonstrate the possibility of AI represented knowledge units in a discrete, symbolic form. These programs then progress through a series of formal, logical operations upon these symbols in an attempt to find a suitable response.

Artificial intelligence, based upon propositional and predicate logic, has been applied to the problem of producing structured and novel behaviour. A number of programs, for example 'Racter'[4] and 'Tale-Spin'[5], have been written to generate novel (literary) stories using two very different approaches within a symbolic framework. More sophisticated approaches were produced as the inadequacies of the symbolic approach became apparent. Programs such as 'Day-Dreamer'[7] were developed which use a system of meta-rules, to allow the generation of further rules.

3.2.1 A Brief Review of 'Racter'

Both 'Racter' and 'Tale-Spin' address essentially the same problem, although the former uses a more flexible approach. It works by constructing sentences according to 'syntax directives' using random words and phrases. This is modified by an ability to adapt phrases used previously to give a sense of continuity. Initially it produces very novel and unpredictable output, but, with time, is found to be highly superficial and produces structures with no meaning. Since there are there are no semantics behind the output, there is no way for Racter to assess what it has produced.

3.2.2 A Brief Review of 'Tale-Spin'

Tale-Spin builds a story from a problem and how this problem gets solved. Characters are set up with different goals which they each try to meet by forming plans and taking appropriate actions. The story is a natural language trace of the events produced by this process.

The more rigid approach of Tale-Spin uses a reservoir of knowledge about each character and how they behave. Problems involved in development are solved using a plan as a template for the action of each character, thereby producing mainly meaningful stories at the expense of originality. This contrasts with the behaviour of Racter, which produced original, but meaningless behaviour.

3.2.3 A Brief Review of 'Day-Dreamer'

The rigidity of rule-based systems has led some researchers to advocate 'meta-rules', i.e. rules which can reason about and create other rules. This was brought about by programs (such as that of Yazdani[6]) which were found to produce more inventive stories in unwanted outputs created when the program went wrong, essentially breaking the rules of the program.

This approach is demonstrated by the program 'Day-Dreamer' which functions according to events in two interacting domains: a 'personal world' and the 'objective world'. Both of these domains are controlled by their own goals, where events in the objective world can affect the personal goals. The program progresses by 'day-dreaming', that is 'imagining' different objective situations which lead to the personal goals to be met. Successful plans are abstracted and stored in memory. When the program next encounters a suitable objective world situation, these plans can be re-called and applied.

'Day-dreamed' plans can also be adapted to fit into different objective goal situations by making a number of variable changes. This provides an added flexibility to the system's behaviour but raises the question of which variables to alter.

This problem illustrates the main flaw with the meta-rule system in that although providing an extra degree of flexibility for rules, meta rules themselves are too rigid. Since rules are predetermined by the meta-rules, such a system essentially requires knowledge to be programmed a priori, and any rule breaking is be algorithmically determined. In effect this is simply a move from rigid rules to rigid meta-rules.

3.2.4 Concluding Remarks

The symbolic approach to AI has made the great achievement of showing that AI is plausible, with computer programs showing characteristics such as learning and reasoning which were previously the sole domain of psychology. However, with time, the symbolic approach has been shown lacking. The main difficulty being the inability to reconcile flexibility of response with control. As a result AI programs have tended to produce either novel behaviour in a meaningless form or behaviour that is structured but essentially predictable and unimaginative.

3.3 Decentralised AI

Minsky[24] has described the mind as an integrated society of agents each with its own specialized task, where societies are built by learning from experience. These ideas have acted to motivate ideas of distributed control as opposed to the centralised control of previous thinking.

This type of approach provides a source of more flexible behaviour, due to the lack of a central controller present in symbolic systems. Each unit in a decentralist system acts at a local level and it is only as chains are constructed that high-level characteristics become apparent. This type of behaviour is called *emergent*[1] since global, high-level properties *emerge* without explicit control from the interaction of many local, low-level actions. This type of approach avoids the structural limitations imposed by central controllers, and allows flexibility as a high-level property.

3.3.1 Neural Network Models

Neural networks[40, 22, 23], in general termed 'connectionist models', are currently the main manifestation of decentralised AI models. They are programmed to mimic actual brain structure, in which knowledge is represented as a distributed array of data in the form of inter-neuron strengths. Neuron strengths are altered as the network learns from its mistakes, resulting in it self-organising to

[1] A more thorough discussion of emergence is given in section 5

yield solutions to problems posed as input data. This approach has been shown to be very effective in a number of areas; for instance in pattern recognition[11] and optimisation problems[12]. However, being essentially parallel in nature, programming of larger nets on traditional Von-Neumann style, sequential computers requires large amounts of computer time to run effectively. Furthermore, connectionism has not developed sufficiently to allow high-level *structure* to be formed to the extent present in more conventional models. An example of this limitation is illustrated by the behaviour of the connectionist computer model 'Net-talk'[21],

3.3.2 A Brief Review of 'Net-talk'

Net-talk is a three-layer neural network using a back propagation learning algorithm and is capable of learning the conjugation, in the past tense, of English verbs - both regular and irregular.

With extensive training, the program acts to self-organise, through a series of three distinct stages, into a form in which it can competently pronounce the past tense of English words of which it has had no previous experience. However, it has been observed by, Partridge and Rowe[38], that this pronunciation is limited to a Southern accent and to develop a different accent, it must be completely re-trained. As a consequence, the original accent is lost. A shift in a high-level property such as accent would much less dramatic if the model could develop a more hierarchical structure. It could then adapt at level appropriate to accent shift, reducing the need and time required for alteration at other levels.

This point plays an important role in the development of this project, and is addressed more thoroughly in later sections.

3.3.3 Artificial Life Models

A more abstract realisation of the decentralist approach is artificial life[20]. Artificial life - A-life - is the study of man-made systems that exhibit characteristics of natural living systems. Generally these are in the form of computer simulations and in particular cellular automata. The biological phenomena modeled by A-life include, for instance, the flocking of birds[25], the mutual evolution of predator and prey[13], and the mechanisms of reproduction[14].

The underlying theme behind A-life is that simple local interactions between the members of a set give rise to complex macroscopic phenomena, again via the process of emergence. This is essentially the same idea as that of Minsky and it is felt that many phenomena present in A-life simulations have relevance to the mechanisms underlying psychological development.

The flocking behaviour of birds provides a good example of the emergence of global phenomena from a collection of individuals in an A-Life simulation.

3.3.4 The Flocking Behaviour of Birds

In his simulation of flocking behaviour, Craig Reynolds[25] starts by considering a large collection of autonomous but interacting objects - individuals in the population which he term 'Boids'. Each Boid is programmed with three rules:

1. To maintain a minimum distance from other objects in the environment including other Boids
2. To match velocities with other Boids in its neighbourhood.
3. To move towards the perceived centre of mass of the Boids in the neighbourhood.

These rules constitute the genotype of the Boids system. When the simulation is started, a collection of Boids released at a random points collect into a dynamic flock which flies around environmental obstacles in a very fluid and natural manner. Occasionally the flock break up in to sub-flocks which then re-organise either to form separate flocks or to reunite into a single flock again. The flocking behaviour itself constitutes a phenotype of the Boid system.

There are no rules applied at a global level to the system; the only rules stated are in the form of each Boid's genotype and apply locally, forming individual behaviour only. As such the system's phenotype is determined by not explicitly by genotype, but by the reciprocal relation between each Boid and its environment. Thus, the system's global form - phenotype - *emerges* from the *collective* actions at a microscopic level.

Recalling the mechanism of epigenesis proposed by Piaget and Waddington, it can be seen that A-life simulations develop in a way essentially the same as that proposed in psychological development: Human phenotype - manifest as physical form and psychological nature - is determined not purely by genetic content, but by the interaction of genetically determined properties and the environment. This relationship provides evidence that epigenetic assimilation is a plausible mechanism of development and is also a possible way of introducing such development in a computer simulation.

However, this simulation is limited in that its potential to produce emergent structures is bounded. That is, flocking is the final state of the system, there is no further emergent behaviour. This limitation has to be overcome for any simulation of psychology to be successful.

3.3.5 Concluding Remarks

In conclusion, there are two main approaches to the simulation of cognitive processes: the conventional, rule-based model and the local interaction-based connectionist model. Although capable of producing structured output, the symbolic approach has yet to produce a program innovative to any meaningful degree. The connectionist approach, on the other hand, is able to produce meaningful innovation but is, at present, is limited to producing relatively un-structured high-level properties.

Chapter 4

Unconventional AI Models

4.1 Introduction

Considering the constraints of contemporary symbolic and decentralised models of AI, a number of alternative approaches have been proposed. In these, the problem of unifying structured output with a degree of meaningful originality has been addressed in an attempt to produce a more convincing AI. Two contrasting models are presented in this chapter, the Emergent Memory Models and Modular Neural Networks. Each presents an approach to producing a structured but flexible solution to a developmental problem.

4.2 The Work of Partridge and Rowe: Genesis

Genesis is a collection of programs with the common goal of producing *creative* solutions to a problem, such as the card game Elusis[26]. The programs of which genesis consists are described as 'Emergent Memory Models'. Each program uses a collection of agents, small self-contained pieces of code, that cooperate to construct representations. This method is similar to that of A-life discussed in the previous section 3.3.3. However, the emergent behaviour produced is harnessed to a greater extent than in A-life. This is achieved using a selection algorithm to to remove any emergent structures which are of little use, leaving only useful structures whose potential is enhanced.

4.2.1 Representational Fluidity in the Model

Partridge and Rowe[38] have suggested that an essential characteristic of any computational model of an intelligent process is 'representational fluidity'. This requires the knowledge representations contained in a model to be related to as many others as possible. Only small variations in the relative strength of associations and minimal variation in the nature of the relational associations are allowed. As a result, no representational combination can always dominate the others, but a wide variety of combinations (higher level representations) will be possible. Furthermore, the more primitive the basic elements of a knowledge representation scheme, the more scope there will be in building up representations at the problem level. As such, representational fluidity implies (but does not require) that the control of the build-up of high-level representations be governed by the primitive elements themselves and not by some macroscopic master controller.

An example illustrating the role of representational fluidity is the 'edge of chaos' idea for cellular automata, proposed by Langton[47]. Here, the most complex and creative structures are found to evolve from rules that essentially maximise the representational fluidity of the system.

It is suggested that a computational model of a developmental process should consist of a set of fundamental units of knowledge which interact in a way that is dependent, to an extent, on a set of local rules. The interaction and co-operation of these units in performing a task can form new knowledge units, the variety and nature of which depend on the nature of the interactions, not on macroscopic rules.

4.2.2 Credit Assignment in the Model

In order to form these new units in to a macroscopic *structure*, Partridge and Rowe employ a credit assignment algorithm, in the form of the 'bucket-brigade' algorithm, treated further in section 9.3.

The formation of a structured framework of emergent knowledge is of key importance to a model of this type, if it is to address high-order, more abstract problems. As such it is required that the model self-organise any emergent knowledge into an accessible, structured form. A similar problem in the context of artificial life has been addressed by the application of genetic algorithms[36] to select structures most suited to a particular evolutionary environment. Any emergent structures which are of no use are removed from the system, allowing the remaining structures to thrive. By applying this selection principle to emergent knowledge structures, only the most useful may be selected, which may in turn interact to form further higher-order structures. A promising algorithm for this task is the bucket-brigade algorithm, which attributes credit to any structures used in a successful chain of events, and discredits those leading to an unsuccessful chain. The solution to a problem is formed as a sequence of knowledge structures, which may then be used in a further sequence to form higher-order structures.

The self-organisation into hierarchies of higher-order structures is arguably the most important feature of this model. It overcomes the limitations of conventional decentralised models discussed in that it can address problems at a variety of levels.

4.3 Modular Neural Networks

Modular Neural Networks are a recent development in the field of neural networks. They work on the principle of 'divide and conquer', a problem is split up by the network into a collection of smaller sub-problems on which separate parts of the network - 'modules' - operate. This allows each module to specialise for each particular sub-problem.

4.3.1 Justification of this Approach

The main justification for the rationale of Modular Neural Networks is illustrated by Haykin[39], who considers the approximation problem as addressed by conventional neural networks. The approximation of a prescribed input-output may be realised using two different methods: a *local* method that captures the underlying local structure, such as radial-bias function networks, or a *global* method, for example a back propagation perceptron, that captures the underlying global structure of a mapping. The advantage of the former is its speed and ease of learning, whilst the latter requires much less memory to operate.

The use of a *modular* architecture is an attempt to combine the advantages of each of these two methods and capture the underlying structure of an input-output mapping at an intermediate level of granularity.

Justification of Modular Neural Networks can also be found in neurology, outlined in section 2.2.3, that derive from observations of the architecture of the vertebrate nervous system. However, because of the newness of this approach, it is unclear how the limitations of the 'modularity of mind' stance will affect the development of such systems in reality.

4.3.2 The Advantages of Modular Networks

The nature of a modular architecture offers a number of advantages over more traditional neural networks:

1. *Speed of Learning.* If a complex function can be broken down into a set of smaller problems, then a modular network has the ability to discover the decomposition. Accordingly, it is able to learn a set of simpler functions faster than a multi-layer perceptron can learn the starting function.

2. *Data representation.* The ability to spilt a complex task into smaller, more manageable units allows for a greater understanding of the approach and progress of the network in learning. By comparison, multi-layer networks are very much black-box mechanisms.

3. *Physical constraints.* In the human brain, there is a limit to the number of neurons that can be accommodated in the available space. Ballard[27] has

> hypothesised that to represent a space of dimension k requires N^k/D^{k-1} neurons, where N is the number of just noticeable differences in each dimension of the space, and D is the diameter of the receptive field of each neuron. To accommodate such large numbers in a finite brain size, it is suggested by Ballard that the brain adopts a modular structure, and that the brain uses a coarse code to represent space. Likewise, it may be argued for the artificial neurons in a neural network.

This last point raises some interesting thoughts. Clearly a multi-dimensional space cannot be represented in full in a real or artificial system. However, important regions in that space may demand specific responses form a system. The mapping of these regions to the desired module occurs during training. However, as the complexity of the mapping increases, for instance stringing a sequence of modules together to form a new skill, the size of the network must also increase. A problem with this is that, in general, network size is determined a priori, a fact which bounds the effectiveness of the network at tacking such problems.

This is similar to the problems faced by the analogous psychological theory outlined in section 2.2.3, in that the predetermined architecture of modular systems bounds the potential for development.

4.4 Conclusion

A contrast is evident between the two approaches illustrated in this chapter is their approach to the formation of emergent knowledge hierarchies.

The approach of Genesis is to form a *sequence* of structures into an emergent higher-order structure, using the bucket-brigade algorithm. This forms a *specific* solution, as a trajectory through phase space, to a problem from the many possible. A sequential hierarchy is produced, in which a problem may be addressed at a number of levels in the sequence.

Modular Neural Networks attempt to produce an expert system in a limited, pre-defined region of phase space - a module. These modules can be connected in a hierarchical manner, a priori, to address problems at a number of levels in parallel, rather than sequentially as in Genesis.

The combining of these two approaches may provide a mechanism capable of producing the type of behaviour associated with psychological development. Genesis is the more flexible, spanning the whole of problem space to give a *specific* solution. The modular approach forms an expert, a *general* solution, in a predetermined region of space. Using the dialectic argument characteristic of Piaget, these two may be combined such that neural network modules may be associated with Genesis-like emergent structures. The requirement to define the function of each module a priori is replaced by a system which allocates each module a specific solution about which it must generalise.

It is felt that this approach will also allow problems to be addressed at various levels, utilising the hierarchical features of both models.

Chapter 5

Emergence and Hyperstructures

5.1 Introduction to the Concept of Emergence

Emergence describes the ability of certain many-bodied systems to produce fundamentally new structures by means of non-linear interactions these systems are known as 'complex systems'. Several examples of emergent phenomena have been covered in section 3.3.3.

The process of epigenetic assimilation described earlier, shares many important features with emergent systems. Primarily, both produce fundamentally new structures and behaviour through interactions with their environment. This indicates that psychological development - which, according to scheme theory, involves the creation of new structures of thought - may have its origin in the fundamental laws of nature. An understanding of emergent behaviour is of great value when attempting to understand its role in psychology. Such an understanding would also enable the successful inclusion of such behaviour in a computer simulation.

Considering the conclusions of the previous section, the *identification* of any emergent structures formed by a computer simulation would go some way to advancing the aims of the current project. This is because emergent structures must be identified by the system before any neural networks can be associated with them.

5.2 Towards an Understanding of Emergence

Emergence has been interpreted in a number of ways[20, 28, 29], from the consideration of various different emergent phenomena. Three main interpretations have precipitated:

- Computational Emergence
- Thermodynamic Emergence

- Emergence relative to a Model

5.2.1 Computational Emergence

Computational emergence is the view that complex global forms can arise from local computational interactions. The result is a 'bottom up' computational approach which is highly compatible with decentralist ideals outlined in section 3.3. The assumption is made that a systems microscopic properties determine its macroscopic nature, but not vice-versa. An example of such behaviour is turbulence in simulated fluid flow[31]: a fluid is modeled in a computer as a cellular automaton. The fluid is considered to be made from a collection of autonomous cells, each obeying simple local rules. From these microscopic rules, the macroscopic phenomena of turbulence is found to emerge in a variety of different situations.

One particular problem that arises from this approach is the lack of an adequate definition of what an emergent computation is and how it is distinct from a non-emergent computation. There is no criteria by which an arbitrary phenomena may be considered emergent. This places a fundamental limit on the understanding of emergent phenomena when considered from the 'computational emergence' point of view.

5.2.2 Thermodynamic Emergence

Thermodynamic emergence arises from the consideration of self-organisation in physical, dynamic systems and in particular how physical systems produce stable and complex structures far from thermodynamic equilibrium. Such emergent phenomena are considered equivalent to attractors in dynamical systems theory. Many examples can be found in biology; the evolution of DNA from a 'primordial soup' and the emergence of auto-catalytic chemical networks are but a few.

As yet, however, the question of how to connect thermodynamic theories of structural stability with the appearance of new behaviour in the system is unknown.

5.2.3 Emergence Relative to a Model

This describes emergence in terms of the deviation of the behaviour of a physical system from an *observers* conception of it. Emergence, then, involves a change in the relationship between the observer and the physical system under observation.

This approach is essentially how emergence in the previous two descriptions has been determined *in practice* in the absence of a formal criteria. For example, emergent phenomena in cellular automata are determined in practice by *observing physically* the state at a given point in time and comparing it to later *observations* of the system. Similarly, in thermodynamics, DNA is considered to be emergent simply by observing biological complexity and comparing it to observations of substance akin to 'primordial soup'. Thus the 'emergence relative to a model' model provides a working definition of emergent phenomena and is, for this reason, considered to be the most valuable interpretation.

5.3 A Mathematical Formalisation of the 'Emergence Relative to a Model' Model

A mathematical formalism has been proposed by Nils Baas[32, 33] in an attempt to define emergence from the 'emergence relative to a model' approach. This may be summarised as follows:

Consider a set of primitive objects - 'first-order structures' - denoted $\{S_i^1\}$, and an observational mechanism, Obs^1, to 'evaluate, observe and describe the structures $\{S_i^1\}$'.

A general procedure is then required to construct a new set of structures - second-order structures - $\{S_j^2\}$ from $\{S_i^1\}$. To this end, the observation mechanism is applied to the members of $\{S_i^1\}$.

Using the properties derived from the observations, $Obs^1(\{S_i^1\})$, a set of interactions Int^1 may be defined. By subjecting members of $\{S_i^1\}$ to Int^1, a new structure is obtained:

$$S^2 = R(S_i^1, Obs^1(\{S_i^1\}), Int^1) \tag{5.1}$$

where R is the construction process resulting from the interaction Int^1 and S^2 is a second order structure. Second order structures may be observed by a new observational mechanism Obs^2 (it may be equal to, overlap, or disjoint from Obs^1). According to Baas, emergence may now be defined thus:

P is an emergent property of S^2 iff

$$P \in Obs^2(\{S_i^2\}) and P \notin Obs^2(\{S_j^1\}) \tag{5.2}$$

5.4 Hyperstructures

The idea of emergence is used by Baas to form the theory of 'hyperstructures', in which emergent phenomena self-organise into levels. Each successive level emerges from the interactions of previous, introducing new interactions which in turn help form further structures. Hyperstructures will form as long as new units, with new properties, are being produced by the interactions of lower levels.

An example of hyperstructure formation is the emergence of biological molecules from lower order molecules, cells from biological molecules and organisms from cells. Thus a living organism may be considered to be a hyperstructure, formed as a result of a wide range of different interactions over successive levels.

By introducing a selection algorithm - a macroscopic rule - to the emergent process, it is possible to select only useful structures and thus control, to an extent, the type of hyperstructure formed. In the above example a genetic algorithm is applied by nature, selecting the organisms that are best able to survive in their environment.

This is fundamentally similar to a number of theories in developmental psychology, chiefly Piaget's learning through stages, concerned with the acquisition of intelligent behaviour[30]. Intelligent behaviour is acquired through a series of stages, each stage forming the essential building blocks of subsequent stages. For example, a child learns to walk through the stages of standing, assisted tottering, tottering and finally walking, each stage must be mastered before the child can successfully attempt the next. This allows the coordination of domain-specific structures into more general domains.

5.5 Conclusion

The phenomenon of emergence presents interesting parallels with psychological theories discussed earlier. An understanding of the former may therefore shed light on the behaviour of the latter. Such an understanding is, in part, presented as hyperstructure theory, which gives a perspective on the nature of emergent phenomena, in particular their ability to self-organise into hierarchies to produce further emergence.

The formalisation of the mechanism of emergence presented provides a method to enable the identification and classification of emergent structures. If this method can be incorporated in a computational model, then it will allow the association of neural network modules with emergent structures within a computer program. This would provide a way of realising the the ideas presented in section 4.4.

Chapter 6

The Problem of Bipedal Motion

6.1 Why Consider Bipedal Motion?

The problem of acquiring the skills to allow a simple bipedal machine to walk has been chosen to test the ideas discussed in this project.

This problem in particular has been chosen for a number of reasons:

- Learning to walk is an important skill to which a great deal of time is devoted during a child's development.
- The means by which a child progresses in learning are easily observed and as a result, are well documented.
- The process is found to progress through a series of stages, where each stage must be mastered before progressing to further. For example before learning to walk, a child must learning to first stand and to balance.
- Development occurs through a mixture of inherited skills and feedback from the environment.

It can be seen that the process is congruent with the Piaget's notion of learning by stages, and epigenesis.

6.2 AI models of the Development of Bipedal Motion

Making a bipedal machine walk is a difficult problem and making it *learn* to walk is even more so. There have been a wide range of approaches to this problem with varying degrees of success.

Considering the difficulty of this problem, coupled with newness of the ideas proposed, it is felt that it is wise to first address an important sub-problem - that of rising and balancing. This may be compared with the inverse problem, which presents the situation in which a mass at the end of a rigid, pivoted, rod must be balanced in an inverted, unstable, position.

6.3 The Inverse Pendulum Problem

The upright standing position is is an unstable position for the human body to be in and requires constant monitoring and adjustment of posture.

The inverse pendulum problem has been addressed by several AI models in a variety of ways. The methods of more recent models include the use of fuzzy logic[42], genetic algorithms[43] and neural networks[44]. They all invariably succeed in their primary objective - attaining a balanced position - but generally form domain specific solutions. As a result, placing the pendulum in a situation outside the training domain, for example giving the pivot a velocity, inevitably requires retraining of the system, usually negating previously learned material[1] This is the same limitation shown by Net-talk in section[21], when attempting to change accents.

A more detailed look at one particular approach illustrates the limitations of contemporary, decentralised approaches to this problem.

6.3.1 Control of an Inverse Pendulum Using Neural Networks

Saravanan[44] presents a multi-layer neural network to control an inverse pendulum without a priori knowledge of its dynamics. The network is trained using an evolutionary programming algorithm, a stochastic optimisation technique which mimics the process of biological evolution. It acts as follows:

1. A population of N trial solutions is created, each taken as a pair of real valued vectors, $(\vec{x}_i, \vec{\sigma}_i), \forall i \in \{1, ..., N\}$, with their dimensions corresponding to the number of weights and biases in the neural network.

2. Each weight vector is applied to the network and it's fitness - defined as the number of time steps before the pendulum crashes - is recorded.

3. Each member generates one offspring, $(\vec{x}'_i, \vec{\sigma}'_i)$, as follows:

$$\vec{x}'_i(j) = \vec{x}_i(j) + \vec{\sigma}_i(j).N(0,1)$$

$$\vec{\sigma}'_i(j) = \vec{\sigma}'_i(j).e^{(\tau'.N(0,1)+\tau.N(0,1))}$$

[1]This phenomena is called temporal crosstalk and is covered in section 6.3.1.

where j is the j^{th} component of the vectors, $N(0,1)$ is random noise in the range (0,1) and τ, τ' are constants.

4. The fitness for each offspring is calculated

5. The fitness of ten randomly selected trial solutions from the set of parent and offspring solutions are selected. Each of these are compared with the fitness of the rest. If the randomly chosen solution performs worse in this comparison then the solution to which it was compared received a 'win'.

6. The N solutions with the most wins are selected to be parents of the next generation.

7. Proceed to step 1 until an acceptable solution is discovered.

This method is found to be capable of controlling an inverse pendulum. The main points of interest are the model's ability to firstly generalise and secondly to adapt to new situations.

The conditions presented to the fully trained model to test its ability to generalise were presented as positional perturbations within the training domain. Results showed that it is able to generalise to a reasonable degree, in that starting from a random point within the training range, the network could balance the pendulum on average half of the trials.

No direct data is given on the model's ability to adapt to new situations, that is to perturbations *outside* the training domain. However inferences may be made from the nature of the learning algorithm used.

An important drawback of a multi-layer neural network of this type is the effect of *temporal crosstalk*. This term may be explained by consideration of the neural network mentioned above: it is first trained on a a particular task and then made to change to a different one. Ideally, the network would learn the second task without its performance being unnecessarily impaired with respect to the first. However, Sutton[45], using the back propagation learning algorithm, has shown that this is not the case and that the learning of the second task destroys the original knowledge. The network may learn both tasks by training initially using a mixture of the two although at the cost of taking much longer.

It has been suggested by Jacobs and Jordan [46] that this problem is a feature of all multi-layer perceptrons and they advocate the use of Modular Neural Networks as a solution. The ability of such networks to partition parameter space in to distance regions allows different expert networks to learn a control law for each region.

6.4 A Model of a Simple Bipedal Machine

To address the developmental problems faced by a child learning to walk, a simple model of a bipedal machine is proposed. This takes the form of a computer

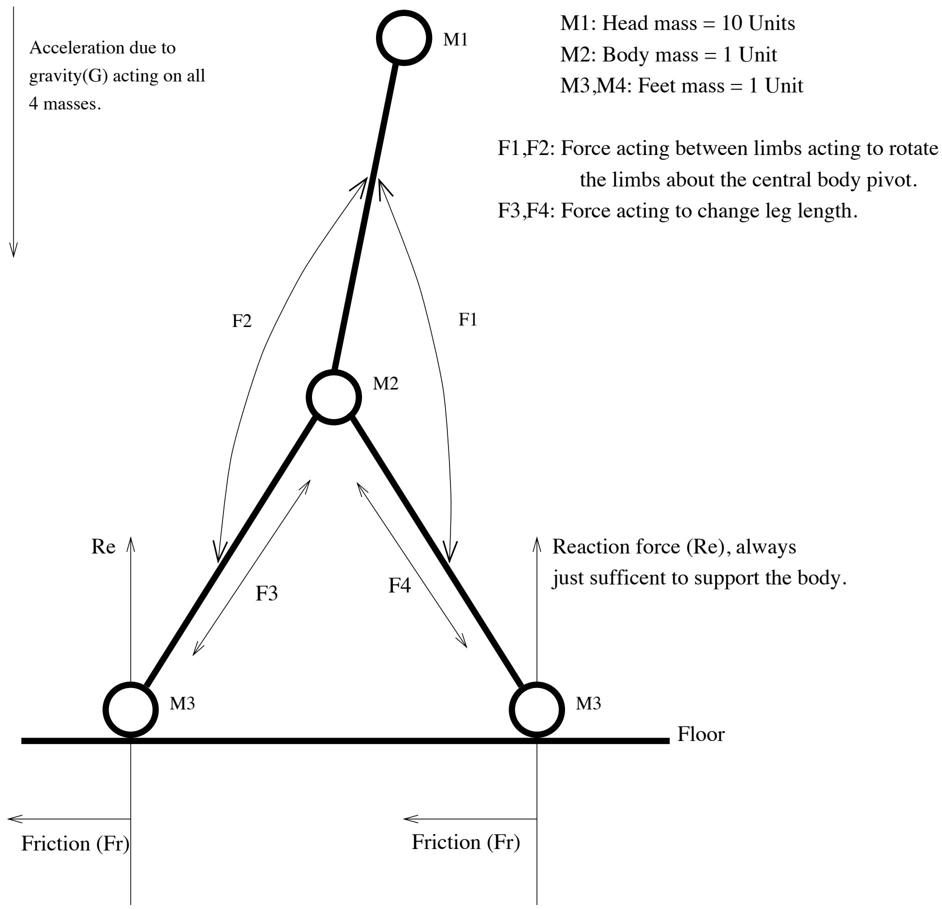

Figure 6.1: The Physical Configuration of the Bipedal Machine

simulation, written using Modula-2. The simulation places a stick-man figure in a two-dimensional Euclidean space. Important factors such as floor friction, gravity, floor reaction force and conservation laws, are all present.

The construction of the machine is illustrated in 6.1.

The machine consists of three 'limbs' connected to a central pivot which represents two legs and a central body trunk. Masses are attached to the ends of each limb, upon which gravity, floor friction and reaction act. The equations governing the machine's 'physical' evolution are a series of non-linear second-order partial differential equations, the solution to which are approximated using Eulers method.

The limbs are controlled using four muscle groups: two connect the body to each leg (F1,F2), and two adjust each leg length (F3,F4). Control of the machine is through the contraction and relaxation of these muscles. The size of the model is of the order of magnitude of a real person, with simulated limb lengths of 1 metre each. Muscle contractions are updated from a 'brain' module at a rate of the order of milliseconds in subjective time (this is also of the same order of magnitude of human reaction rates). This provides a reasonably realistic simulation in which to test the ideas discussed previously.

The change the machine's physical configuration over time is displayed as trajectories through phase space. The dimension of this space represents the number of degrees of freedom of the machine. The total space in which the machine operates is reduced to selected degrees of freedom which best illustrates it's behaviour.

Chapter 7

Addressing the Problem of Bipedal Motion

An artificial intelligence system has been programmed to act as the mental system to control a bipedal machine. The aim is to provide a system that produces a solution to the problem of learning to walk as an emergent, hierarchical structure.

The approach taken is based upon the conclusions of section 4.4. Here it was proposed that a mixture of contemporary methods would be the way forward. The problem is addressed initially by an Emergent Memory Model, capable of spanning phase space to produce a flexible, specific solution to a given sub-problem in the from of sequence of structures - schemes. This solution is then identified by the system and associated with a neural network module to act as an expert, capable of generalising about the solution. This generalisation occurs in a restricted subspace of relevance. From this approach, it is felt that a flexible but hierarchically structured solution will result.

To realise these proposals, it is necessary to bring together important features of both scheme and hyperstructure theory. Schemes are important in that they are legitimate, flexible psychological objects and may be considered to be formed from neural network modules. The formalisation of emergence given in hyperstructure theory provides a potential means by which the system can identify any emergent structures in order to develop them further.

7.1 Representing Information in the System

The systems representation of its physical configuration and its environment are modeled as points in a multi-dimensional phase-space which spans the machines' physical degrees of freedom. This gives the system the ability to represent, and thus address, any physical situation presented to it.

Although this approach is chosen on practical grounds, there is evidence[35] to suggest that spatial representations have a vectorial form in the nervous system.

7.2 The Role of Schemes in the AI System

As outlined in section 2.3, the concept of a scheme may be used as a representation of a psychological structure, controlling the regular structure of a physical action. Schemes may be produced in practice by mapping a set of sequential, controlled muscle contractions to a set of labels in the AI system's memory. Each scheme performs a unique physical action by repeating a particular set of muscle actions when activated.

By applying the principles of hyperstructure and scheme theory, it is felt that a solution to the problem of learning to walk can be produced as an emergent hyperstructure, constructed from scheme units. This may be achieved by a process similar to that of epigenesis: The starting set of schemes may be considered analogous to the machine's genotype - a set of pre-programmed instructions. Any development of the machines mental abilities - the formation of it's phenotype - is derived from the interaction of the pre-programmed schemes with the environment. From these interactions, new structures may be formed, as emergent, second and higher order schemes. This may be achieved in a way similar to the accommodation process in scheme theory, treated with in more detail in the next section.

If these new schemes can be identified and added to the set of first order schemes, then further interactions may lead to further emergence and yet higher-order structures produced. The result is the formation of a hyperstructure of new schemes, each level based upon the previous and each expanding the machines skill repertoire. These new schemes have new properties and, unlike modules in modular mind theory, are not programmed a priori to operate in a given domain.

Once identified as distinct, these emergent schemes may be associated with a neural network, as mentioned above. This will allow the broadening of the scheme from a specific solution to a more general one.

7.3 Motivation, the Driving Force of Development

The system acts according to it's position with respect to a 'motivating configuration', which represents the ideal state for the machine to be in. The use of motivation is consistent with scheme theory, which uses the idea in the activation of schemes as part of a learning process.

Motivation, represented as a point within the models multi-dimensional phase space, acts to attract the machine to a certain configuration. This form is equivalent to extrinsic motivation as described by Eckblad[19], in that it represents an external, environmental driving force. This is a crude representation analogous to, for example, some desired object being presented to a child: the child assess its position with the object and adjusts as best as it is able to reach it.

In practice, the motivating point corresponds to the upright balanced position. The machine is initially configured to a bent over, standing position. This constitutes a variation on the inverse pendulum problem, as the system must act to get the machine from bent over to upright. Attaining and maintaining an upright position is an essential part of the process of learning to walk.

7.4 The Nature of Scheme Development

In practice, scheme development occurs by initially endowing the mental system with an innate knowledge of how its body works. This is described as a basis set of simple schemes which map a set of muscle contractions to their effects on the body in a given context. This set constitutes the fundamental building blocks of further mental structures and their analogous physical skills. From hyperstructure theory, we may consider this basis set to be a collection of first order structures.

The basis set is produced by starting the machine in a standing configuration and applying a small force to each muscle in turn and all their combinations. The effects of each muscle contraction upon the body are recorded in a variable array along with the context - the body's configuration under which the force was applied - and the nature of the applied force. Each of these arrays represents a member of the basis scheme set.

When fully initialised and running, the system consists essentially of three units:

- The selection of the scheme to activate.
- The interaction of the chosen scheme and the environment.
- The new scheme construction, when a string of activated schemes have a successful outcome.

Each of these stages is related to the processes of both psychological development and hyperstructure formation, highlighted in the following sections. These relations are important since they are of use in later sections.

7.4.1 The Mechanism of Scheme Selection

The selection of a scheme to activate is an essential feature. The use of a basis scheme set restricts the search space to only twelve schemes, simplifying the search process.

In practice, the system considers it's current state (it's position in phase space), it's desired state (it's motivation point in phase space) and the means by which it may progress from it's current state (it's basis scheme set).

From this information, a scheme must be selected which has been found to best move the system through phase space towards the motivating point. This process is similar to both the assimilation process in scheme theory and the observation - Obs - mechanism in hyperstructure theory. A person is likewise attempts to find the best action for the situation by assimilating its surroundings.

7.4.2 The Mechanism of Scheme Interaction

Once selected, a scheme becomes activated and interacts with the environment in the form of a series of muscle actions, in an attempt to move the system's state towards the motivating point. Due to non-linearity of the system's physics, this interaction may act, if the scheme is applied out of context, to shape the scheme into a new form and take the system away from where it wants to go.

In scheme theory, if the result of a scheme activation do not match those expected then the person will go into a state of disequilibrium. In disequilibrium, a person attempts to construct a new scheme to deal with the new situation in hand.

The interaction between an activated scheme and the environment is equivalent to the interaction - Int - mechanism in hyperstructure theory.

7.4.3 The Mechanism of New Scheme Formation

The system goes into disequilibrium if a new situation is encountered, one in which the system has not been before or in which previously chosen schemes were found to fail. Scheme failure is defined, at present, as when a scheme that takes the system away from the motivating point. In this situation a new, successful scheme must be constructed. This is achieved using a building block process in which a series of smaller scheme units are strung together to make a new scheme.

Correctly assigning credit to the building blocks which produce a working new scheme is of key importance to the successful outcome of the construction process and has been addressed in a number of ways. It is essentially the same as the accommodation process in scheme theory. In this, an unsuccessful scheme is altered and adapted until a new, successful scheme is produced.

From the point of view of hyperstructure theory, this process can be considered analogous to the construction function, R, given in equation 5.3.

Chapter 8

Preliminary Work

8.1 Introduction to the Ideas Behind the First Model

To test the ideas presented so far, a simple model has been produced to manipulate the schemes mentioned. In this model, schemes are selected by choosing the scheme n, say, which most effectively spans the distance in phase space between the body's current state and it's desired state, using the formula:

$$\text{Scheme chosen} = \text{Min}(n) \mid (\vec{M} - \vec{C}) - \Delta\vec{S}(n) \mid$$

where, $\vec{M}$ is the motivating state, $\vec{C}$ is the current system state and $\Delta\vec{S}(n)$ is the vector distance spanned by scheme n. This is a simple selection algorithm with no consideration being given to past experience in a given state.

Once chosen, a scheme is activated and acts upon the machine via its corresponding muscle action. The resulting state is judged to see if it is successful, using the formula:

$$(\mid \vec{M} - \vec{I} \mid - \mid \vec{M} - \vec{F} \mid) > 0 \Leftrightarrow \text{Scheme is successful}$$

where $\vec{M}$ is the motivating state, $\vec{I}$ is the machine's state before the scheme activation and $\vec{F}$ is the machine's state after the scheme activation. If successful then the scheme is coordinated into a larger scheme structure otherwise the process of scheme selection repeats. This acts as a crude approximation to the accommodation process. If the system falls over as a result of its action, then it is returned to its original position, with new scheme structures intact. A flowchart of this program is given in figure 8.1:

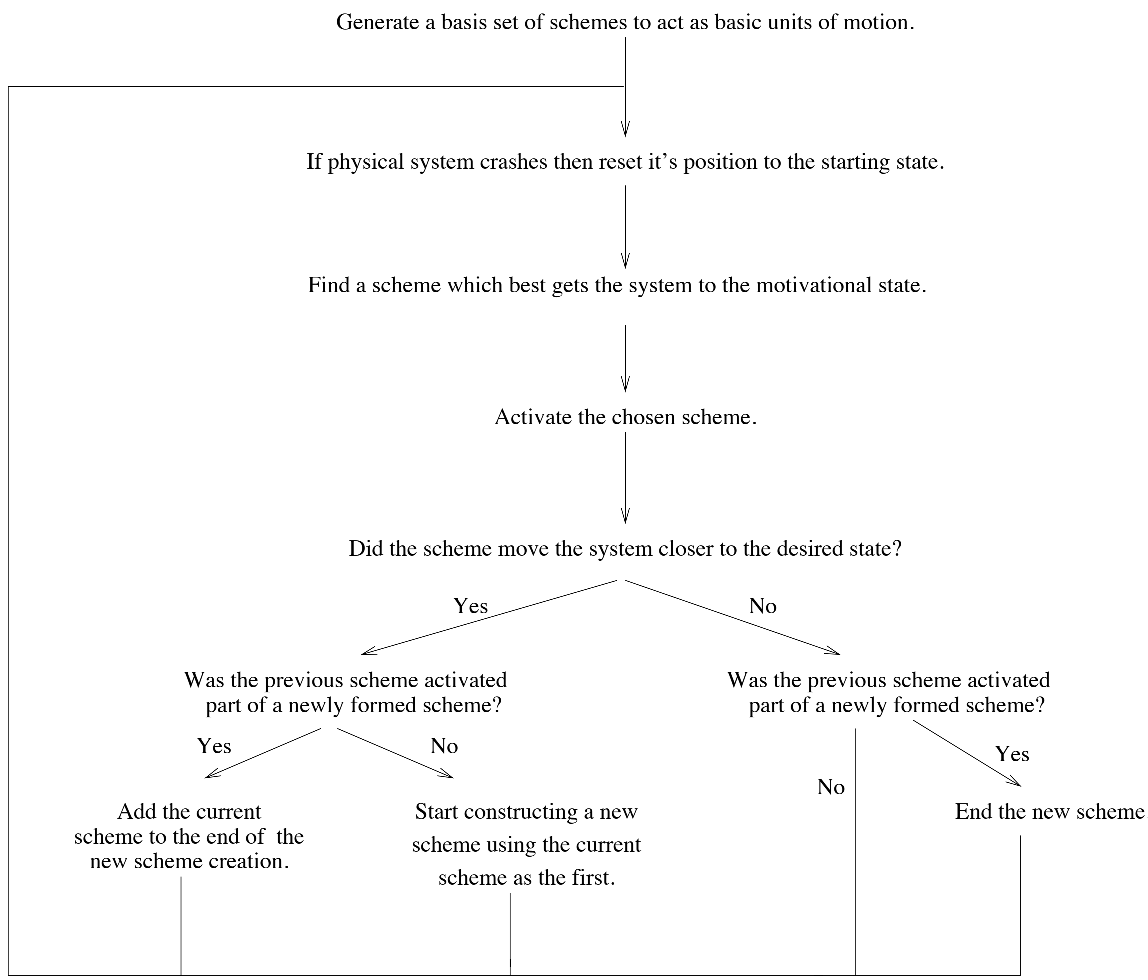

Figure 8.1: Flowchart Representation of the Preliminary Program

8.2 Results and Conclusion

Using this method, the model was unable to reach a balanced position. This has been attributed to two factors inherent in the model:

Firstly, the means by which a scheme is chosen takes no account of the context - i.e the position in phase space - in which the scheme was *created*. As such, schemes may appear to be the best choice when considered out of context, but when applied may act in unexpected ways. As a result, the selection process becomes unreliable with useful schemes becoming replaced with schemes which are outwardly attractive, but have essentially useless properties in that context.

Secondly, once a bad scheme is chosen, it cannot be removed easily. For example, if a successful scheme is replaced by an unsuccessful one, the unsuccessful scheme will still be judged to be better than the original by the selection algorithm.

Chapter 9

Program 1: Linear Context

Considering the failings of the model presented in the previous chapter, it is felt that two important features are missing:

- Consideration of the context in which a scheme is formed.
- A form of credit assignment to schemes depending on their performance.

9.1 Association of a Context to Each Scheme

One way to associate a context to each scheme is to index each activated scheme with a unique label - a context - which relates it to a unique point in phase space. This allows the consideration of the suitability of each scheme for a specific task, given its past performance in that context.

9.2 Assigning Credit to Schemes

Credit may be assigned to schemes by associating each scheme in each context a variable corresponding to it's suitability of activation - it's *favour*. The value of the favour depends on the individual schemes performance and the performance of the schemes to which it leads. The reasoning behind this is that an activated scheme may produce desirable behaviour in the short term, but may inevitably lead to an undesirable situation later on. Thus credit must be assigned not only on initial scheme success, but on the success of subsequent schemes. An algorithm to achieve this can be found in the form of the bucket-brigade algorithm, as demonstrated by Partridge and Rowe in section 4.

9.3 The Bucket Brigade Algorithm

The Bucket Brigade algorithm, as described by Holland[41] and Goldberg[36], is designed to assign credit to a sequence of operations. The difficulty, as mentioned above, is to credit operations far down a sequence from an outcome to which they contribute. To this end, each operation[1] is assigned a variable - it's favour - which is adjusted by the actions of the Bucket Brigade algorithm.

The favour of an arbitrary operation at a time t+1, $S_i(t+1)$ is given by the equation:

$$S_i(t+1) = S_i(t) - P_i(t) - T_i(t) + R_i(t) \tag{9.1}$$

where $R_i(t)$ is an increase in favour if a later scheme is deemed successful, $T_i(t)$ is a general reduction in favour to 'prevent free-loading' and $P_i(t)$ is a reduction in favour when an operation is initially selected. The resulting action produces a chain of operations linked by high favour ratings.

This algorithm needs some adjustment to complement the hierarchal nature of operation (scheme) formation of the proposed program. This adjustment replaces the two favour reduction terms with a term that is invariant with regards to the order of scheme hierarchy.

9.4 Justification of the Use of the Bucket Brigade Algorithm

It has been shown, through extensive simulation[37] that the bucket brigade algorithm does work in practice. The use of the bucket brigade algorithm in the proposed simulation can be justified on a number of grounds:

Hebb[9] proposed a method of human learning at the cellular level which results from the strengthening of the connections between frequently fired neuron cells. This mechanism forms a concise rule based on neurobiological evidence. It may be expanded and rephrased as a two stage rule:

- If two neurons on either side of a synapse (connection) are activated simultaneously then the strength of that synapse is selectively increased.
- If two neurons on either side of a synapse are activated asynchronously, then that synapse is selectively weakened or eliminated.

The bucket brigade algorithm is consistent with this method of learning, which learns through the strengthening or atrophy of connections between schemes. Further evidence of the associative nature of learning at the level of the neuron is given by Gallistel[35].

[1] In Holland's case each operation is represented as a classifier - a string of simple rules which guide a system's behaviour.

9.5 Program Structure

Each distinct situation the system has been is stored in memory as a series of steps from the starting configuration. To each of these steps is associated a set of variables corresponding to the 'favour' of each member of the basis scheme set in that context.

The scheme with the highest favour is selected, the value being updated regularly via the Bucket Brigade algorithm given in 9.5.3. When assimilating, the system checks to see if it's situation it is in matches that of any known context, where each context in memory represents a unique situation. Context is remembered simply as a series of steps form the starting point. If this is true then the situation is assimilated, otherwise it must be accommodated.

Accommodation occurs in two ways: if assimilation fails because the situation the system is in is unknown, then accommodation occurs simply by adding a new step to the end of the context series with a standard set of scheme favour values. If failure is due to the malfunction of an activated scheme in a known context, then accommodation occurs by changing favour values of the scheme and its predecessors, via the Bucket Brigade algorithm. A flowchart diagram of the program is given in figure 9.1.

9.5.1 The Consequence of a Successful Scheme Activation

If a new scheme is activated and succeeds in moving the system towards the motivating point, then it is added to the end of a chain of previously successful schemes via an increase in its favour. Each time a new scheme is added to this chain, the favour of rest of the chain is increased by a constant amount. If the chain eventually succeeds in attaining the goal i.e. reaching the motivating point, then it is formed into a new scheme. This new, higher-order scheme may then be used to form further chains and so on.

Due to the hierarchical nature of the structures produced by the system, favour changes to a high-order scheme structure will affect the lower-order schemes from which it is formed. On the assumption that a high-order scheme is formed from a set of one-degree-lower schemes activated more than once, n times say, then each time a scheme of order m is activated, the favour of a typical constituent first-order scheme will be adjusted, raised by a constant amount, of the order of n^m times. As a result, the favour of schemes in an m order structure will increase by an order of m whenever that structure is involved in the construction of further, successful structures.

9.5.2 The Consequence of an Unsuccessful Scheme Activation

If a particular scheme in a chain is found to be unsuccessful, i.e. if it takes the system away from its motivating point, then its favour is reduced, discouraging its further activation at that point. If new schemes activated at that same point also fail, then it is desirable to change the scheme structure further back down the scheme chain. To allow this problem to be addressed at each level in the scheme hierarchy, the favour values of the schemes in the chain must be reduced in the same manner as they were built up, that is, as an exponential function of hyperstructure order. This is achieved by decreasing favour values by an amount ΔF. The value of ΔF is given by the equation:

$$\Delta F = e^{k.f(n)} \tag{9.2}$$

where f(n) is a function - denoted the *frustration* - of each context, n, in a scheme series and k is a constant. The value of frustration at a particular context step in the series increases with each successive failure of assimilation at that point. When assimilation finally does occur, the frustration returns to 0, the default value. Thus a particular problem, posed as a difficult point in phase space to assimilate, will be addressed at points of increasing hyperstructure order approximated as an exponential function of scheme number. In practice, when faced with a difficult problem a scheme series will change at exponentially increasing points along its length, i.e at points increasingly distant from the problem.

9.5.3 Formal Definition of the Bucket Brigade Algorithm

The Bucket Brigade algorithm may be written as follows: If the last scheme $x(m), \forall m \in \{integers\}$ in a new series of schemes[2] $x(n), \forall n \in \{1, ..., m\}$ is successful, then the scheme favours associated with the series $F(x(n))$ are updated according to the equation:

$$F(x(n)) = F(x(n)) + 10$$

and $f(x(m))$, the frustration function, is updated according to:

$$f(x(m)) = 0$$

Otherwise, scheme favours are updated according to the equation:

$$F(x(n)) = F(x(n)) - e^{k.f(x(m))}$$

also, $f(m)$, is updated according to:

$$f(m) = f(m) + 1$$

[2] A Bucket Brigade series as opposed a context series.

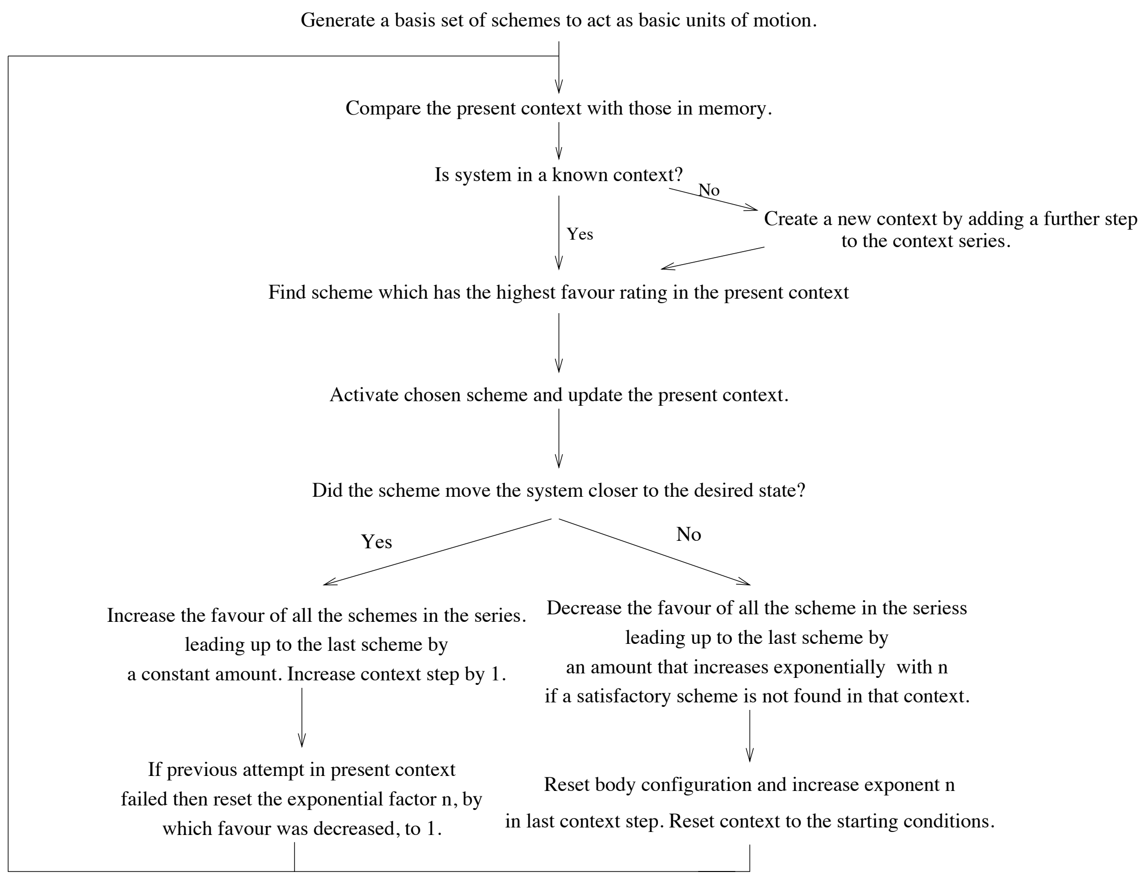

Figure 9.1: Flowchart Representation of the First Program.

9.6 Understanding the Layout of Results Presented

The results taken from program 1 are presented as a series of frames taken at various stages of development. Each scheme path is presented as a trajectory in a phase-space restricted to the Cartesian machine 'head' velocities and vertical position degrees of freedom. Each frame shows the *qualitative* nature of the scheme developed in which the trajectory of the scheme at that time is plotted along with the previous trajectories from which it developed. Thus, following the frame sequence of the results (the correct method shown in figure 9.2) illustrates how a scheme develops over time.

The last frame in the sequence - frame 9 - shows the *final* scheme structure in it's final form.

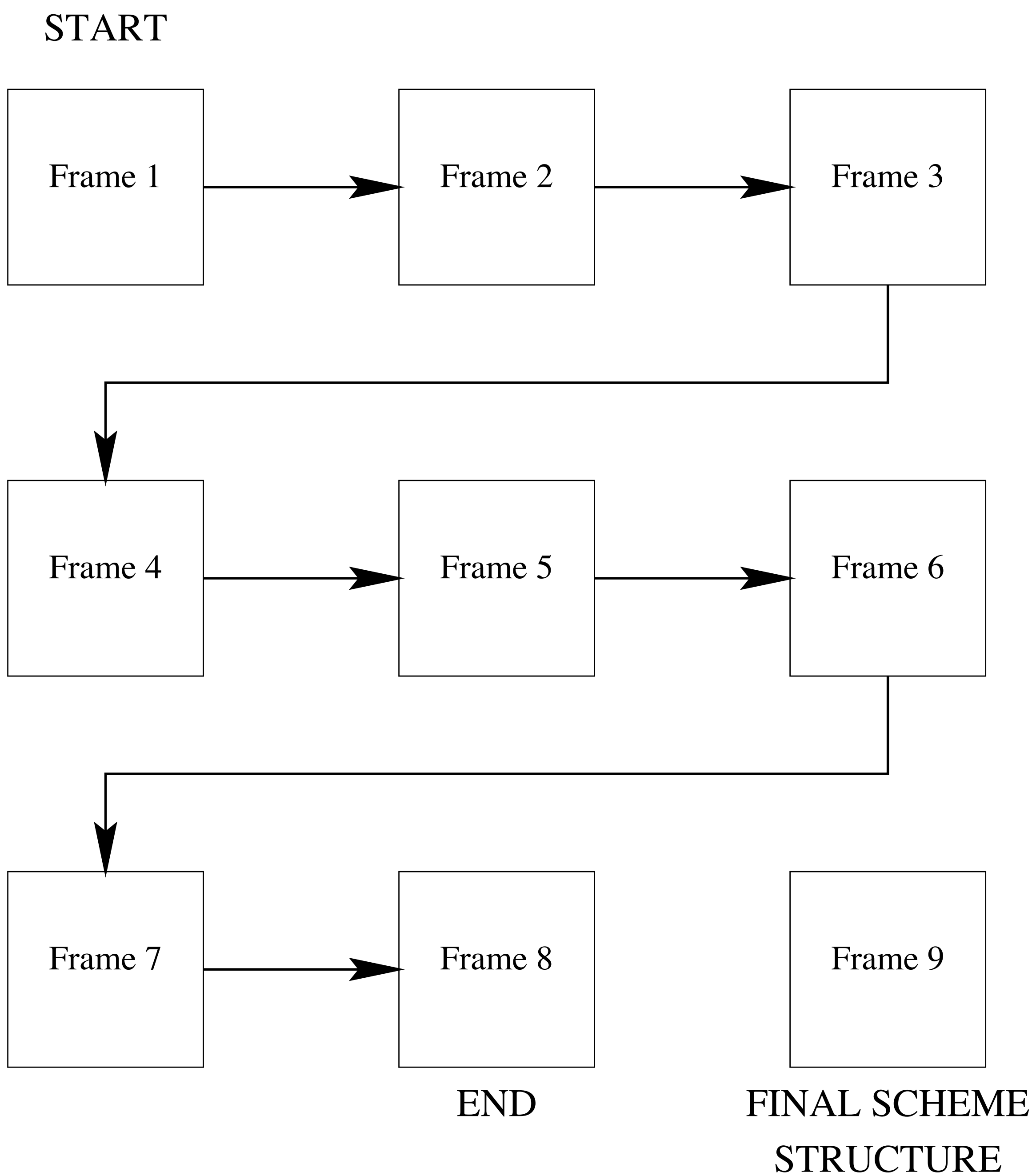

Figure 9.2: Diagram Showing How to Successfully Read the Results Given.

9.7 Results From Program 1

Results are presented in figure 9.3 as a collection of frames showing various stages of scheme development.

The value of k was varied and was found, as expected, to control the rate at which the Bucket Brigade algorithm works its way back along the scheme series. It may be possible to optimise the time in which a solution is found by way of altering the value of k. However the optimal value is almost certainly fragile.

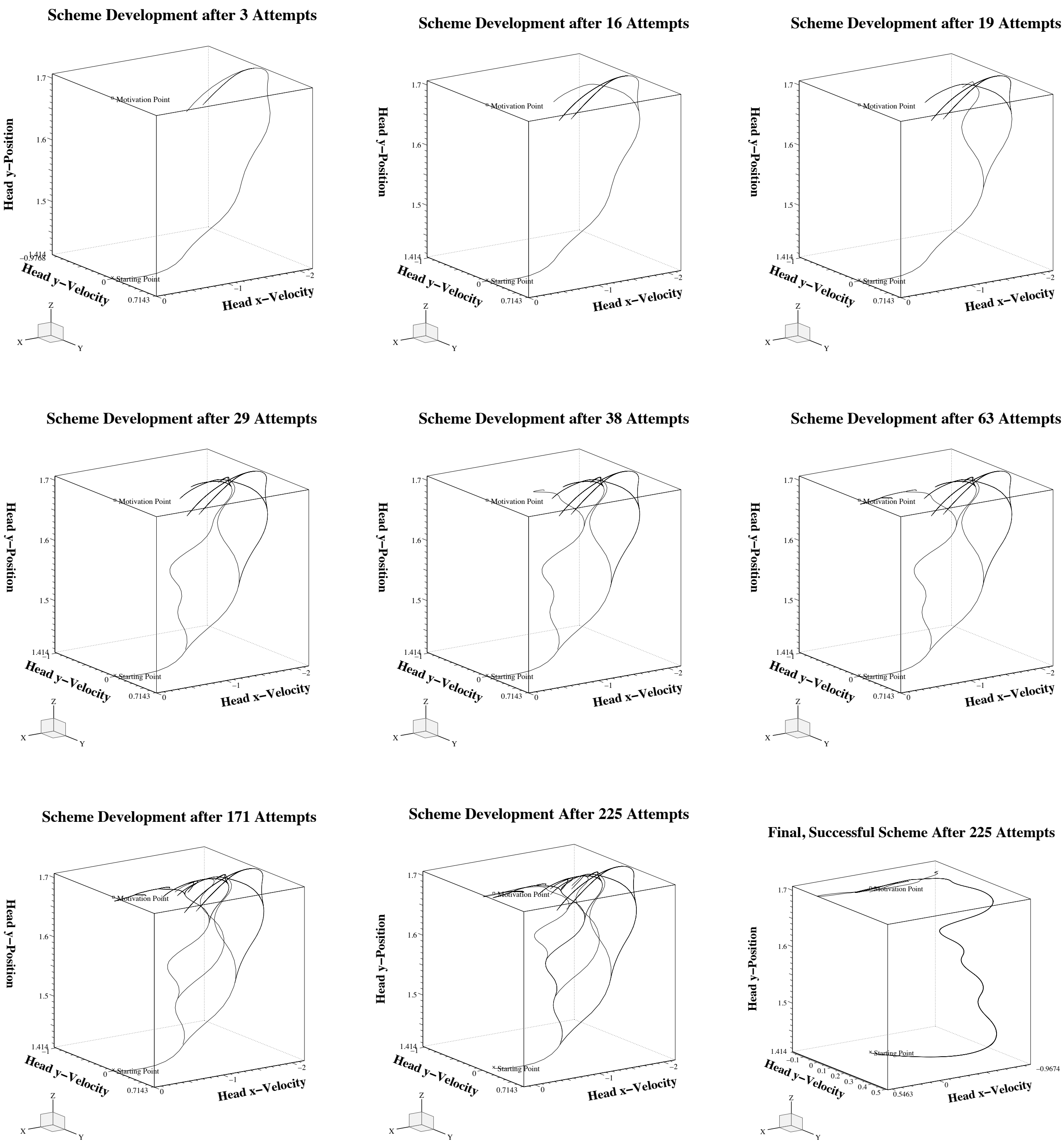

Figure 9.3: A Series of Plots of Scheme Structure for Program 1 at Various Stages in Development.

9.8 Discussion of the Results From Program 1

The program was successful in that it found a means by which to balance the inverse pendulum represented by it's head. The progress of the system can be seen in the results. Firstly, the system rises much to fast and begins to topple over immediately (frame 1). With time, this rising is slowed to a more controlled rate and by frame 7 the rate of rising is so slow that the system passes close to the motivating point. Further frames show the system attempting to improve further, resulting in an oscillation about the point. With further training, the duration of this oscillation increases, although the system always loses balance eventually. Does this behaviour constitute the formation of an emergent structure?

9.8.1 A Mathematical Analysis of the Results

By applying hyperstructure theory to the results obtained, it is possible to identify and classify any emergent structures produced. This is an important process as future development of the program requires it to identify emergent structures itself. Let the set of first order structures - the initial basis scheme set - be denoted by:

$$\{S^1_{i_1}\}$$

An observation mechanism - Obs^1 - is chosen to determine the properties of the set of first order structures $\{S^1_{i_1}\}$.

The observation mechanism acts upon the set of scheme favours in a series to determine the best schemes to take the system from the origin towards the motivating point:

$$Obs^1(\{S^1_{i_1}\})$$

Once selected, each scheme is activated and interacts with the physical environment, represented by the interaction mechanism:

$$Int^1$$

This interaction results in the alteration of the set of scheme favour values. This occurs through the action of the Bucket Brigade algorithm and is denoted by the function R producing a new set of first order structures:

$$S^1_{i_2} = R(S^1_{i_1}, Obs^1, Int^1)$$

As this process repeats, a new observation mechanism is applied - Obs^2 - which is used to observe any second order structures present.

Obs^2 may be defined as a function to 'determine the schemes that will reliably take the system directly from the starting point to a region within one scheme activation of the motivating point'.

If P is a property of subsequent scheme structures such that:

$$P \in Obs^2(\{S^1_{i_n}\}), \mathrm{but} P \notin Obs^2(\{S^1_{i_m}\})$$

then

$$\{S^1_{in}\} \equiv S^2 \tag{9.3}$$

where S^2 is a second order structure of which P is an emergent property. In this case, the property P may be interpreted as the ability to take the system to an upright position - i.e. to solve the inverse pendulum problem for a finite time. The initial basis set of schemes and their initial favours cannot move the system to the upright position without failing at some stage, as demonstrated by earlier stages of the results. However, after sufficient training, the schemes are shown to have acquired this property. Thus equation 9.3 holds for a value of n of 255 attempts, and the second-order scheme structure, S^2_1, may interpreted as a scheme to control rising motion. If the first scheme in the second-order scheme structure is activated, necessarily starting from the initial configuration, then the system will inevitably be taken to the upright position.

This shows that the solution to the above variation on the inverse pendulum problem may be considered to be an emergent structure.

9.9 Conclusion

It may be concluded that the inclusion of a fixed frame of reference - a context - for the scheme sequence combined with the use of the Bucket Brigade credit allocation algorithm produces successful results. As shown above, the application of hyperstructure theory makes clear the solution to the inverse pendulum problem is formed as an emergent, second order structure.

However, the solution does not provide a stable solution over time, as the system inevitably topples over at some point. This is due to the context being sequential in nature, the sequence inevitably comes to an end.

The solution is specific to the initial starting position as this is used as the only external reference point. If the system is started from a new, different point, it's behaviour cannot be relied on to find the target point without further training. This behaviour is comparable with that of neural network solutions to such problems, which require retraining when the system is put into a novel situation. At present, each scheme favour is indexed from a one-dimensional sequence, itself indexed in space from the initial starting configuration. As such, if a scheme favour in the sequence is changed, via the action of the Bucket Brigade algorithm then all subsequent schemes will act out of context. This has the effect of destroying knowledge built up in the old context which may have been of use later. To produce a more robust solution, each scheme in the scheme series should be indexed to the region in space in which it is activated. This will form a bijective mapping between the favour of a scheme and the region in space in which the scheme operates, preserving all useful knowledge built up.

This will potentially allow the production of a wider range of emergent structures. It is an essential part of hyperstructure theory that a number of emergent, second-order structures are produced in order to form still higher-order structures. In the context of learning to walk, the machine must learn to balance and shift weight to one leg before apply both to balance on one leg. The skill of balancing on one leg can thus be conceived as being an emergent phenomenon constructed from the interaction of balancing and weight shifting with the environment.

Chapter 10

Program 2: Multidimensional Context

As outlined in the previous chapter, a sequential method of referencing schemes is inefficient: When an alteration (accommodation) takes place at some point in a scheme context chain, all subsequent information in the series is negated.

It is felt that a more sophisticated, and realistic, approach to the process of scheme referencing is necessary in order to allow a system to make a more robust association between schemes and the situation in which they are successful. As a result, a much larger proportion of useful knowledge would be retained.

To achieve this, each scheme selected is no longer associated with a one-dimensional context, based upon it's position in a series, but with the region in the phase space representing the system's state at that time. This matches each scheme with the (practically) unique physical situation in which it was selected.

In practice, the system checks to see if it's position in phase-space lies within a small radius around each of the contexts it knows i.e. contexts with which it has associated a scheme suitable for activation. If this is so, then the situation is assimilated, otherwise it must be accommodated. Assimilation causes aforementioned scheme to be activated, moving the system through phase space. In keeping with the dialectic thinking of Piaget, a more sophisticated accommodation process has been produced in tandem with assimilation. Accommodation occurs in two ways: if assimilation fails because the situation it is in is unknown, then the context is added to memory with a standard set of scheme favour values, which it then assimilates. If assimilation fails because the scheme activated fails to move the system towards the motivating point, then accommodation is made by the alteration of the schemes favour value via the Bucket Brigade algorithm. The second process is the same as that described in the previous section, where the *sequence* of activated schemes is remembered and credit altered throughout the series according to the success of further schemes.

A flowchart diagram of the program is given in figure 10.1.

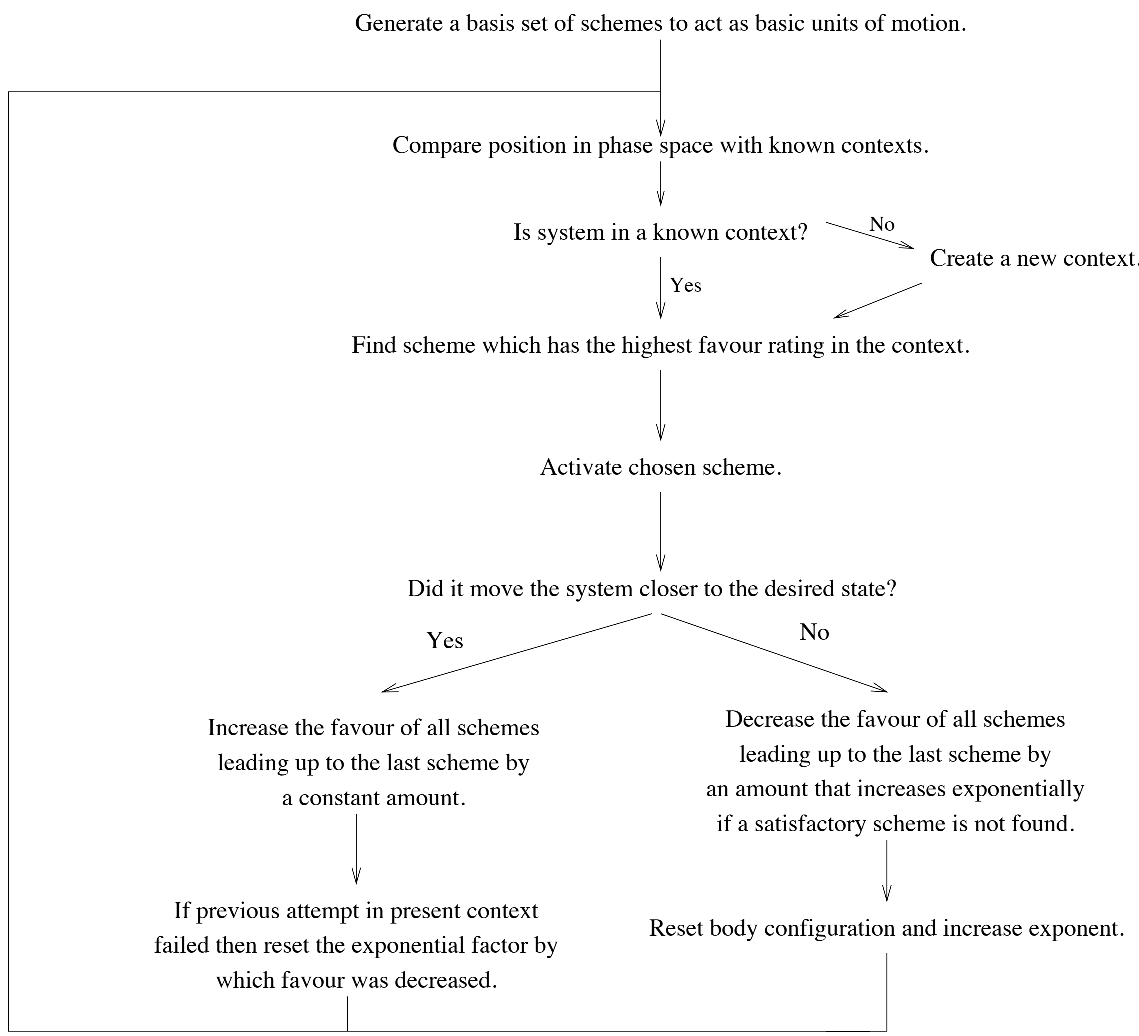

Figure 10.1: Flowchart Representation of the Second Program

10.1 Justification of this Approach

To justify this approach it is necessary to consider it from a variety of viewpoints.

Considering Neurology

Electrical methods, newly developed for investigating neurobiology, have shown certain nerve cells to have very specific tasks in perception systems. One example, in a study of the visual system of cats, has shown that certain neurons react specifically to the presence of inclined features in one particular part of the field of vision[51]. Other studies in biology have shown that there is a lateral interaction mechanism which depends on the distance between neurons receiving signals from receptor cells[52]. Together these acts to cluster neurons which perform similar functions together spatially[1].

This indicates that the human brain copes with different situations using different clusters of neurons, where *physically close* neuron clusters self-organise to deal with tasks that are physically close in the real world.

This evidence complements the approach of associating schemes with the region in phase space in which they operate. This may be considered analogous with the association of neuron clusters with the circumstances - position in phase space - in which they are activated.

Considering Psychology

The approach bears resemblance, although highly simplified, to the early part of the sensorimotoric stage of development as described by Piaget: starting with a set of basic reflex actions, a baby progresses through six sub-stages sequentially, each one an essential epistemological preparation for those that follow. The first sub-stage involves the spontaneous activation of various reflex actions, such as crying, grasping and moving the eyes and head. It consists of interacting sensory and motor processes that manifest the same general structure on each occasion of use. This is broadly analogous to the training stage of the system in which a set of muscle actions are combined to form a fully differentiated scheme set.

The sub-stage two baby continues to exercise many of its reflexes and enlarge their scope. In general these exercises act to gradually form a coordination between the baby's perceptual and motor systems.

Although driven by an extrinsic motivating force, the artificial system is also learning to coordinate what it perceives in phase space and what it does via its selection of scheme. Thus the linking of the point in phase space to the scheme chosen is in general consistent with psychological theory.

[1] For further experimental evidence supporting this view see[35].

10.2 Results From Program 2

Results are presented in figure 10.2 as a collection of frames showing various stages of scheme development. As in section 9.7, results are presented as trajectories in a phase space restricted to the Cartesian head velocities and vertical position degrees of freedom. A series of eight of these plots is given at various stages of development, showing the changes in the form of the scheme over time. The last frame in the sequence shows the final scheme structure.

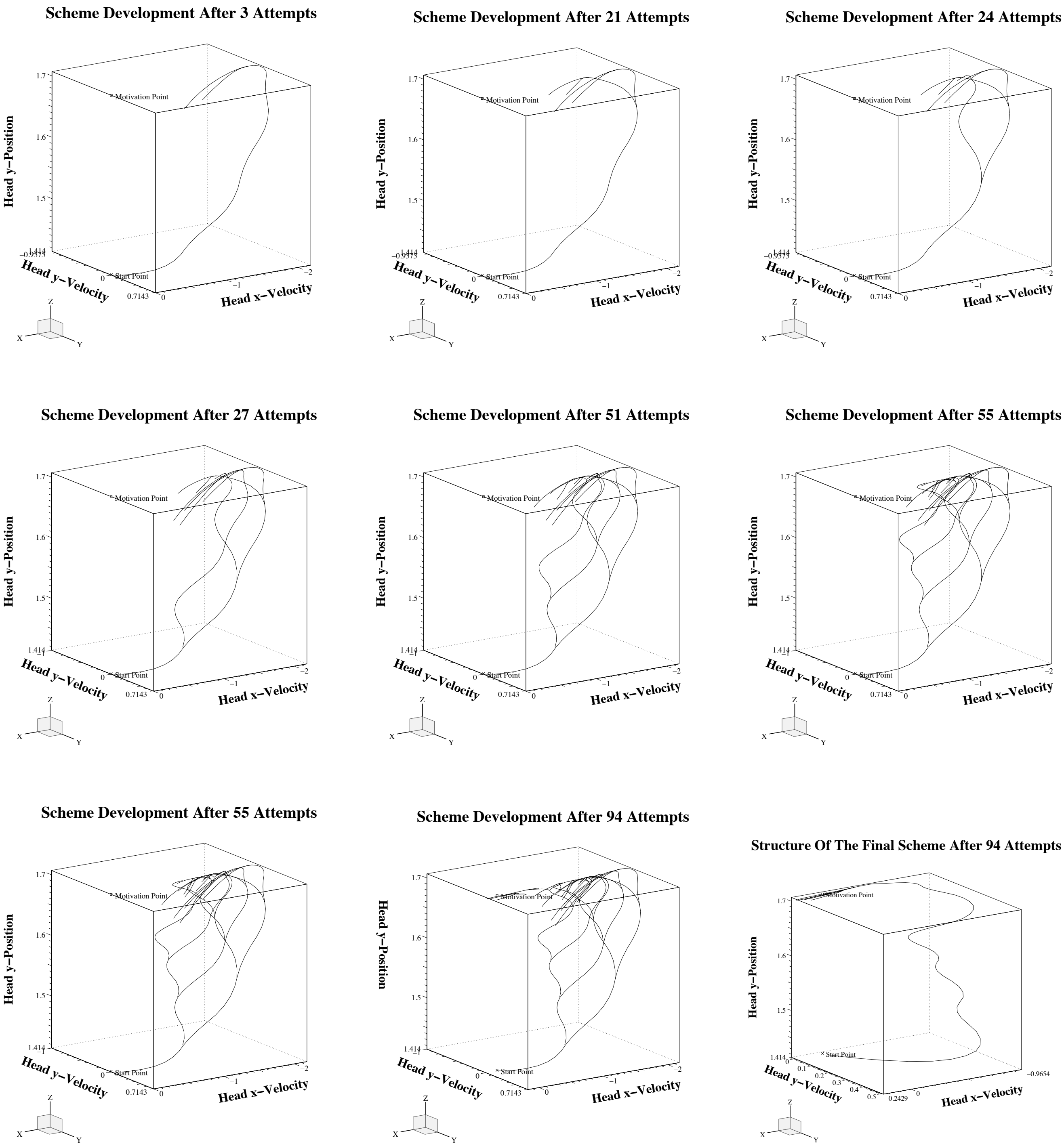

Figure 10.2: Series of Plots of Scheme Structure for Program 2 at Various Stages in Development

10.3 Discussion of the Results From Program 2

Although both successful, the most striking difference between these results and those from the previous programs is that the present system is able to balance after much fewer failed attempts. This is attributed to the different way in which context is handled and in particular the more complete description of context inherent in the second model.

In the first model, the memory of past events (the success of scheme activations) is referenced by the number of steps (scheme activations) taken from the starting position. Due to the multi-dimensional nature of events, the model's memory retains information which may not be of future use and acts to bias future decisions. For example, if the Bucket Brigade algorithm changes a scheme favour in the middle of a chain, schemes beyond that point are made irrelevant but are still present.

It is relatively easy to remove this problem in this particular situation; once a different scheme is activated due to the action of the credit assignment algorithm, then the memory of events beyond that step can be reset to their default values. This makes further choice of schemes unbiased with respect to past choices at the equivalent stage. However, in doing this, information which could under other circumstances be of use later, may be destroyed, particularly if perturbations to an established and homogeneous scheme are applied early in it's activation.

A further difference is that the skill of balancing is now stable in time. This contrasts to the previous model which eventually toppled.

Clearly, referencing the favour of each scheme to its position of activation in phase space is a useful technique, increasing the model's functional efficiency by retaining a greater quantity of useful information.

10.3.1 A Mathematical Analysis of the Results

By applying hyperstructure theory to the results obtained, it is possible to identify and classify any emergent structures produced. Let the set of first order structures - the initial basis scheme set - be denoted by:

$$\{S^1_{i_1}\}$$

where each of $S^1_{i_1}$ is a member of the set of all schemes and their favours.

An observation mechanism - Obs^1 - is chosen to determine the properties of the set of first order structures $\{S^1_{i_1}\}$

The observation mechanism acts upon the set of scheme favours in phase space to determine the best schemes to take the system from the origin towards the motivating point:

$$Obs^1(\{S^1_{i_1}\})$$

Once selected, each scheme is activated and interacts with the physical environment, represented by the interaction mechanism:

$$Int^1$$

This interaction results in the alteration of the set of scheme favour values. This occurs through the action of the Bucket Brigade algorithm and is denoted by the function R producing a new set of first order structures:

$$S^1_{i_2} = R(\{S^1_{i_1}\}, Obs^1, Int^1) \tag{10.1}$$

As this process repeats, a new observation mechanism is also applied - Obs^2_1 - which is used to observe any second order structures present.
Obs^2_1 may be defined as a function to 'determine the schemes that will reliably take the system directly from a known point in space to the motivating point'.
If P is a property of subsequent scheme structures such that:

$$P \in Obs^2_1(\{S^1_{i_n}\}), \text{but} P \notin Obs^2_1(\{S^1_{i_1}\}) \tag{10.2}$$

then

$$\{S^1_{i_n}\} \equiv S^2_1 \tag{10.3}$$

where S^2_1 is a second order structure of which P is an emergent property. In this case, the property P may be interpreted as the ability for the machine to rise to an upright position. The initial basis set of schemes and their initial favours cannot move the system to the upright position without failing at some stage, shown by the earlier stages of the results. However, after sufficient training, results show that the system has acquired this property. Thus equation 10.2 holds, and the second-order scheme structure, S^2_1, may interpreted as a scheme to control the rising motion. If the first scheme in the second-order scheme structure is activated, necessarily from the starting from the initial configuration, then the system will inevitably be taken to the upright position.

A third Observation mechanism may be applied: Obs^2_2.
Obs^2_2 may be defined as a function to 'determine the schemes that act directly to keep the system perpetually within one scheme activation of the motivation point'.
The property P may be interpreted as the ability to balance. Considering the initial basis set, results indicate that they cannot obey this criteria. This can be seen by the initial scheme failure, i.e. toppling over of the system, near the motivating point in the early frames of the results. However, again after sufficient training, the system is able to balance and the resulting emergent schemes do in fact exhibit the property P. This implies that equation 10.2 holds for the observation mechanism Obs^2_2 and that the second second-order scheme structure, S^2_2, may be interpreted as the scheme which controls balance.

10.4 Conclusion

Referencing context as a position in phase space produces better results than the sequential context used in the previous section.

The results produced were more robust to change, giving more rapid learning, and formed a stable solution in time to the problem of balancing. This stable solution results from the formation a second emergent scheme structure in addition to the 'rising' scheme formed in both programs. This scheme controls balancing and is stable in time, using a context that is referenced by the system's position in phase space.

This shows that a solution to the above problem in the form of two, second-order, emergent structures. This is an important development considering the single emergent structure produced in the previous section.

These new structures have the potential to be the building blocks for yet higher order emergent structures. For example, the system can now generate a scheme which can control balance, (the use of which will form an essential building block of an emergent model of bipedal motion.

The ability to make generalisations is of key importance to the production of higher order structures. To walk, the balancing scheme must generalise with respect to the context in which it is activated. For instance the machine must still balance equally well with one leg swinging as it does when both legs are firmly on the floor. This is not yet possible with the system produced so far, which is restricted to specific solutions to initial conditions. The task of introducing an ability to generalise is discussed in the following chapter.

The new, emergent schemes are referenced to distinct regions of space and operate in a way that is conceptually similar to the operation of 'modules' in Modular Neural Networks, and the ideas of 'modularity of mind' theories, discussed in sections 4.3 and 2.2.3. However there are a number of important differences.

Firstly, the schemes - 'modules' - in the present model are *emergent* whereas each module of Modular Neural Networks is programmed a priori. As a result, there is an added dimension of flexibility present in the current system.

Secondly, because of the large number of neurons needed to address large problem spaces, neural network modules must be restricted to small regions of problem space. This contrasts with the approach used here, in that schemes address the whole of problem space, building a specific solution through a process of trial and error.

Finally, schemes produce specific solutions to problems from given initial conditions. This contrasts with expert systems produced in a neural network module which, to an extent, generalise solutions to new situations presented in it's subspace.

Chapter 11

Future Research and Conclusion

11.1 Concluding remarks

A simple system to model a developmental process has been successfully produced. It is consistent with accepted psychological and neurological theory, and has been observed to develop a two-stage solution to the problem of balancing an inverse pendulum. Thus an essential factor in the process of learning to walk has been learnt, in a way that provides new skills that are essential to further development - the ability to rise and balance.

Development occurs in the system through an epigenetic process of system-environment interaction. This leads to the formation of emergent structures. These structures have been classified according to their observed function using hyperstructure theory notation. This classification groups the structures according to their function and the system can be observed to develop structures that deal with particular task, for example balancing and rising. These new structures may be coordinated into yet further structures and so on giving subsequent solutions a hierarchal structure.

The structures produced are similar to the modules discussed in sections 4.3 and 2.2.3. Important differences exist, which are highlighted in table 11.1 below. It is felt that these two approaches can be brought together to enable the schemes

Scheme	Module
Unable to generalise	Can generalise to an extent
Develops in all space	Develops in a small subspace
Flexible structures produced	Rigid structures produced

Table 11.1: Table Outlining the Main Differences Between the Schemes Produced and Modules

produced in the final chapters to solve problems and develop further.

11.2 Future Work

To develop further towards the goal of learning to walk, it is essential that the system learns to make generalisations about any emergent skills. This will enable it to use those skills outside the circumstances in which they were formed, so as to be of use in the construction of new schemes.

The ability to make generalisations in complex situations is exhibited by neural networks, as shown in section 6.3.1. If these systems can be associated with emergent schemes, then it may be possible to introduce an ability to form generalisations about that particular skill in the bounded subsystem in which it operates. For example it may be possible for a neural network to generalise the balancing scheme to moving frames of reference.

A further development necessary is the expansion of the motivation mechanism to take the system beyond the stage of standing and balancing.

These ideas, coupled with the achievements of this project, should allow the support of a complex developmental process such as learning to walk.

Bibliography

[1] Scruton, R. (1983) *Kant*, Oxford University Press.

[2] Karmiloff-Smith, A. (1992) *Beyond Modularity: A Developmental Perspective on Cognitive Science*, MIT Press.

[3] Johnson, M. H. 1993) *Brain Development and Cognition: A Reader*, Oxford.

[4] Chamberlain, W. (1984) *Racter: The Policeman's Beard is Half Constructed*, Warner Books, New York.

[5] Meehan, J. R. (1977) *Tale-Spin, an interactive program that writes stories*, In 5th International Joint Conference on Artificial Intelligence, pp 91-98.

[6] Yazdani, M. (1989) *A Computational Model of Creativity, Machine Learning*, Chapman and Hall.

[7] Mueller, E. T. (1987) *Day-dreaming and Computation: A Computer Model of Everyday Creativity, Learning and Emotions in the Human Stream of Thought.* PhD Thesis, University of California, Los Angeles.

[8] Hebb, D. O. (1955) *Drives and the C.N.S.*, Psychological Review, Vol. 62, pp 243-254.

[9] Hebb, D. O. (1949) *The Organisation of Behaviour*, Wiley.

[10] Bartlett, D. J. (1932) *Remembering: A Study of Experimental and Social Psychology*, Cambridge University Press, Cambridge.

[11] Fukushima, K. *Neurorecognition*, Biological Cybernetics, Vol. 36, pp 193-202.

[12] Hopfield, J. J. Tank, D. W. *Neural Computation of Decisions in Optimisation Problems*, Biological Cybernetics, Vol. 52, 141-152.

[13] Taylor, C. E. Jefferson, D. R. Turner, S. R. Goldman, S. R. (1989) *RAM: artificial Life for the Exploration of Complex Biological Systems*, Artificial Life I, Addison Wesley.

[14] LANGTON, C. G. (1989) *Artificial Life*, Artificial Life I. Addison Wesley.

[15] PIAGET, J. (1952) *The Origins of Intelligence in Children*, International Universities Press, New York.

[16] SLATER, A. MATTOCK, A. BROWN, E. *Size Consistency at birth: Newborn Infants' Responses to Retinal and Real Size*, Journal of Experimental Child Psychology, 1990, No. 49, pp 314-322.

[17] WADDINGTON, C. H. (1957) *The Strategy of Genes*, Cambridge.

[18] FODOR, J. A. 1983 *The Modularity of Mind*, Cambridge, Mass., MIT Press.

[19] ECKBLAD,G. (1981) *Scheme Theory*, Academic Press, New York.

[20] CARIANI, P. (1991) *Emergence and Artificial Life*, Artificial Life II, Addison-Wesley.

[21] SEJNOWSKI, T. J. ROSENBERG, C. R. (1986) *Net-talk: A Parallel Network That Learns to Read Aloud*, Technical Report JHU/EECS-86/01, Johns Hopkins University, Baltimore, MD.

[22] DAVALO, E. NAIM, P. (1991) *Neural Networks*, MacMillan, London.

[23] LIPPMANN, R. P. *Introduction to Computing With Neural Nets*, IEEE ASSP Magazine, April 1987.

[24] MINSKY, M. (1985) *The Society of Mind*, Picador.

[25] REYNOLDS, C. W. *Flocks, Herds and Schools: A Distributed Behavioral Model*, Computer Graphics, 1987, Vol. 4, No.21, pp 25-34

[26] GARDNER, M. *On Playing the New Elusis, the Game that Simulates the Search For Truth*, Scientific American, 1977, Vol. 7, pp 18-25

[27] BALLARD, D. H. *Cortical Connections and Parallel Processing: Structure and Function*, Behavioural and Brain Sciences, Vol. 9. pp 67-120.

[28] FORREST, F. *Emergent Computation*, Physica D, 1990, Vol. 42, pp 1-452.

[29] CRUCHFIELD, J. P. *The Calculi of Emergence*, Physica D, 1994, Vol. 75, pp 11-54.

[30] JOSEPHSON, B. D. HAUSER, H. M. *Multistage Acquisition of Intelligent Behaviour* Kybernetes, 1981, Vol 10, pp 11-15.

[31] FRISCH, U. *Lattice-Gas Automata for the Navier-Stokes Equation*, Physical Review Letters, 1986, Vol.56, pp 1505-1520.

[32] Baas, N. A. *Emergence, Hierarchies and Hyperstructures*, Artificial Life III, Addison-Wesley.

[33] Baas, N. A. *Hyperstructures as Tools in Nanotechnology and Nanobiology*, To be published.

[34] Langton, C. G. *Life at the Edge of Chaos*, Artificial Life II, Addison-Wesley.

[35] Gallistel, C. R. (1990) Chapter 14, *The Organisation of Learning*, MIT Press.

[36] Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison-Wesley.

[37] Goldberg, D. E. (1983) *Computer Aided Gas Pipeline Operation Using Genetic Algorithms and rule Learning*, PhD. dissertation, University of Michigan.

[38] Partridge, D. Rowe, J. (1994) *Computers and Creativity*, Intellect. Oxford.

[39] Haykin, S. (1994) Chapter 12, *Neural Networks*, Macmillan.

[40] Rumelheart, D. E. McClelland, J. L. (1986) *Parallel Distributed Processing* MIT Press, Camb. Mass.

[41] Holland, J. H. Chapter 20, *Escaping Brittleness*, Machine Learning Volume 2, 1986, Morgan Kaufmann Publishers.

[42] Park, Y. M. Moon, U. C. Lee, K. Y. *A Self-Organising Fuzzy Controller*, IEEE Transactions On Fuzzy Systems, 1995, Vol. 3, No.1, pp 75-82.

[43] Koza, J. R. Keane, M. A. *Genetic Breeding of Nonlinear Optimal Control Strategies For Broom Balancing*, Lecture Notes in Control and Information Sciences. 1990 Vol. 144, pp 47-56.

[44] Saravanan, N. *Neurocontrol Problems: An Evolutionary Programming Approach*, Journal of Systems Engineering. 1994, Vol. 4, No. 1, pp 1-12.

[45] Sutton, R. S. (1986) *Two Problems With Back-Propagation and Other Steepest-Decent Learning Procedures for Networks*, Proceedings of the Eighth Annual Conference of the Cognitive Science Society. 823-831. Hillsdale, NJ.

[46] Jacobs, R. A. Jordan, M. I. *Learning Piecewise Control Strategies in a Modular Neural Network Architecture*, IEEE Transactions Systems, Man and and Cybernetics. Vol. 23, pp 337-345.

[47] Lasky, R. E. (1974) *The Effects of Visual Feedback on the Reaching of Young Infants*, Unpublished PhD. Thesis, University of Minnesota.

[48] Brunswick, E. (1952) *The Conceptual Framework of Psychology*, international Encyclopedia of Unified Science, Vol1, No. 10, University of Chicago Press.

[49] Boden, M. A. (1994) *Piaget*, Fontana Press.

[50] Cohen, D. (1983) *Piaget: Critique and reassessment*, Croom Helm. London.

[51] Messenger, J. B. *Nerves, Brains and Behaviour*, Studies in Biology no. 114

[52] Kohonen, T. (1984) *Self-Organisation and Associative Memory*, Springer-Verlag, Berlin.