SIMULATING SENSORIMOTOR SYSTEMS WITH CORTICAL TOPOLOGY

A Thesis

by

JAMES BENNETT SAXON

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 1991

Major Subject: Computer Science

SIMULATING SENSORIMOTOR SYSTEMS WITH CORTICAL TOPOLOGY

A Thesis

by

JAMES BENNETT SAXON

Approved as to style and content by:

Bruce H. McCormick
Bruce H. McCormick
(Chair of Committee)

Wayne Shebilske
Wayne Shebilske
(Member)

Amit Mukerjee
(Member)

Richard A. Volz
Richard Volz
(Head of Department)

August 1991

ABSTRACT

Simulating Sensorimotor Systems with Cortical Topology. (August 1991)

James Bennett Saxon, B.S., B.S., University of California at Irvine

Chair of Advisory Committee: Dr. Bruce H. McCormick

This broadly oriented thesis defines different avenues into understanding brain-like intelligence. We categorize our research under the term *neurobotics*, which we have defined as the study of neurally inspired intelligent systems which causally interact with their external world. It comes at this issue from three different directions: the theoretical, the computational, and the empirical.

We first focus on robot arm/robot eye sensorimotor systems by categorizing previous work into a theoretical timescape classification. Because of the simple and causal closed-loop between the arm and the eye, this system becomes a very useful system for developing actual models to test our theories of neurobotics.

To practically address the issues raised, we develop a large brain simulation environment, called the *Neurobotics Simulation Package* (NSP) which, is capable of simulating and visualizing complex sensorimotor systems based on heterogeneous neural networks representing multiple topological brain areas.

Finally, to take us one step closer toward the empirical relevance to our theories, we explore the diverse capabilities of cortical areas in the brain by extending research on self-organizing neural networks (Kohonen, 1988; Obermeyer et al., 1990). The results of our simulations, along with physiological data, suggest that a neural paradigm can be more powerful than the self organizing abstraction because it relaxes the requirements of a stringent topological mapping and allows for degenerate, distributed, spatially-organized, but also fragmented neural mappings (Stryker, 1989).

ACKNOWLEDGMENTS

Many thanks are due in the creation of this thesis. Firstly, I thank Dr. McCormick, whose insight lead me to many of the strategic decisions in this thesis and beyond. Dr. Mukerjee is thanked not only for the global and deep discussions we've had but also for the late night candle-burning example that I have tried to attain and the midnight picnics I have tried to attend. And Dr. Shebilske, who introduced me to the other side of the loop, is also due many thanks.

I would also like to thank the researchers of the Scientific Visualization Laboratory, Hassan Charara, Sandip Barua, Niels Konrad Bauer, and especially Sheng-Yih Guan, whose comments and discussions were invaluable in making this thesis a credible, readable, and current piece of work.

I thank my parents David and Sylvia Saxon, my sister Jennifer Leslie Saxon, and all of my friends especially Cara Ellen Cohen, Randolf Solomon Mendelsohn, Cynthia Fae Berman, and Michael David Yanuck for their patience and deferral of the many plans I have sacrificed.

A very special thank you goes out to Savannah Catherine Gorrell and Deane Charba. Finally, I would like to thank Trent Edward Lange, without whose constant, vigilant, and immediate maintenance of DESCARTES and my state of being, Neurobotics would still be an idea, and I would still be "Mastersless".

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

1. INTRODUCTION

This thesis presents a theoretical framework, a simulation environment, and preliminary simulation results for the development of brainlike neural network systems. The broad focus of the research is to develop and understand the kind of intelligence that humans and animals possess, while not requiring the overhead of detailed neuronal-based modeling. Our long range goal is to identify invariant characteristics of the brain to guide the design of simulation environments for complex brainlike systems, which we can then simulate and observe.

By looking at the brain from these three different timescapes: the evolutionary, the developmental, and the immediate, we realize the intricate biological history that the human mind is built upon. Using the timescapes, we are able to categorize previous brainlike systems research into three different *types* of physical systems. Such insight makes it clear that narrow approaches to developing intelligence, such as the purely symbolic approach of artificial intelligence (AI), or using homogeneous neural networks, are too simplistic to succeed in any broad domain.

We focus here upon *sensorimotor systems*, which can be made sufficiently simple to simulate with today's computers. Much sensorimotor research combines a robot arm and a visual input tied together through some type of associating mechanism. Here we create Neurobot II, a robot arm/eye sensorimotor system based upon its predecessor, Neurobot I (Saxon & Mukerjee, 1990).

Neurobot I was based upon the insightful self-organizing networks presented by Kohonen. These networks are biologically inspired but then abstracted into a simplified form (Kohonen, 1987). Using our Neurobot II framework, we simulate a less abstracted network and find that it becomes capable of alternative types of organization as opposed to a rigid spatial topology found in Neurobot I. Specifically, we find that the network can represent a distributed organization more reminiscent of actual cortical topologies.

1.1. Research Objectives

There are three tiers of objectives in this thesis:

Theoretical: Categorize sensorimotor systems research with respect to different timescapes as a first step toward developing a theory of brainlike intelligence.

This thesis follows the form and style of *Neural Networks*.

Computational: Develop a simulation environment for implementing brainlike systems.

Experimental: Investigate self-organizing networks that allow extended flexibility similar to biological cortical areas.

"Architectures of animal nervous systems are shaped by evolution and carried by the genetic code. The essential quality of such an architecture is that it must learn from the environment in which the animal lives" (Mead, 1990). In other words, any creature on the earth is first and foremost a product of its entire evolution, secondly of its lifetime environment, and thirdly of its immediate situation. Our research brings together principles from these different brain timescapes so that we may better succeed in developing viable and robust sensorimotor systems.

Our pursuit of a brain theory begins by establishing a rudimentary timescape framework which places sensorimotor systems in an organizational scheme based upon evolution. We then create a simulation environment called the *Neurobotics Simulation Package (NSP)*, which is capable of implementing these timescape components. Finally, we model sensorimotor systems by creating Neurobot II and extending Kohonen's work.

The term *neurobotics* has been chosen to describe this line of work. We define neurobotics as study of neurally inspired intelligent systems which causally interact with their external world. Our research in neurobotics is strongly tied to neural networks and robotics, and is also related to neuroanatomy, psychology, and philosophy.

1.1.1. Timescape Categorization of Sensorimotor Systems

Sensorimotor systems combine aspects of robotics, computer vision, artificial intelligence and/or neural networks on the technological side and involve anatomical, physiological, and psychological systems on the biological side. We will characterize recent robotic sensorimotor system designs by comparing them from the perspective of three different size timescapes: the evolution-scape, the development-scape, and the thought-scape. We show that the different systems contain different proportions of the timescapes and the more versatile the system, the more fully it represents contributions from all three timescapes.

The robot arm, eye (or visual system), and control are common in sensorimotor system research and are the focus of our research and simulations.

1.1.2. Development of the Neurobot Simulation Package

Our goal here is to develop a simulation environment, called the *Neurobotics Simulation Package (NSP)*, which allows the user freedom to quickly generate and observe neurobotic models. To exercise our package, we develop Neurobot II, a model sensorimotor system that includes a simulated robot arm, a retina which encodes the visual image of the robot arm, and a control mechanism which closes the loop.

Our categorization of sensorimotor systems focuses on evolutionary development. We wish to accommodate the possibility of modeling this aspect of intelligence also. The NSP must be extensible to accommodate increasing complexity and must easily accommodate substitution and augmentation of the components as the internal and external world models becomes larger and broader.

Finally, in neural simulation, the neural activations, neural connectivity, and brain-area activities and their interrelations all provide pertinent information, both spatially and temporally. New visualization techniques for observing and presenting this type of information, such as animation and high volume density plots, must be developed.

1.1.3. Reinvestigation of Kohonen Self-Organizing Network and the Neurobot II Simulations

Preliminary results in reinvestigating Kohonen's self-organization paradigm show that it has the potential to generate more complex mappings than have been previously described (Kohonen, 1987). Figure 1 demonstrates a single activity bubble, the limitation inherent in the standard self-organizing paradigm. Neurobot I was encumbered by this same problem. By simulating a more biologically inspired network we wish to show that other types of mappings can occur in a cortical map than the self-organizing network can not manage. Figure 2 gives an example of a simulation showing that multiple activity bubbles indeed occur in our more extended model. Neurobot II reflects this extension also. Our final result demonstrates how this activity can lead to a topological *and* distributed dimensionally-reduced representation of the input space.

We also wish to observe other simulation results so that we may characterize different neural paradigms and brain-area design characteristics. This task is a step-by-step process, beginning with the simplest structures and then merging them into a

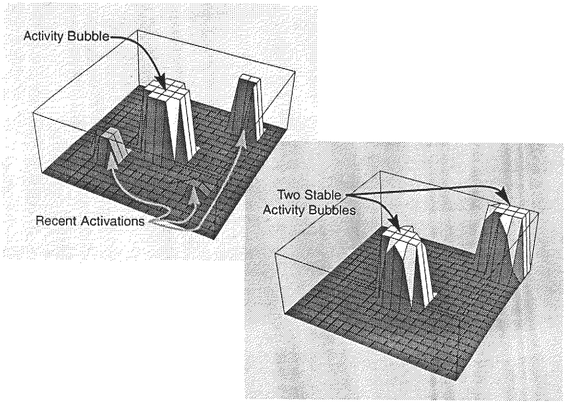


Figure 1. Activity Bubbles. An abstraction used by Kohonen in his self-organizing networks is based on a single activity bubble causing a "winner-take-all" strategy. Resimulating the same network shows that multiple bubbles are also viable, implying that multiple winners could cause a completely different mapping to occur.

larger forum. We hope by this process, to characterize some low-level components of neurobotics.

1.2. Background

The various conceptual levels inherent in this research can be seen by the three tiers of research objectives. To fully grasp the guiding theory, we need an approach based upon evolution; for our computational goals, we specify the sensorimotor systems as a useful test domain; and our experimental results emerge from simulations based upon the other two tiers. These conceptual levels are continually maintained throughout the thesis.

Concepts of interest that inspired the research include the closed-loop phenomenon Piaget observed in children (Piaget, 1963), and the self-organizational ca-

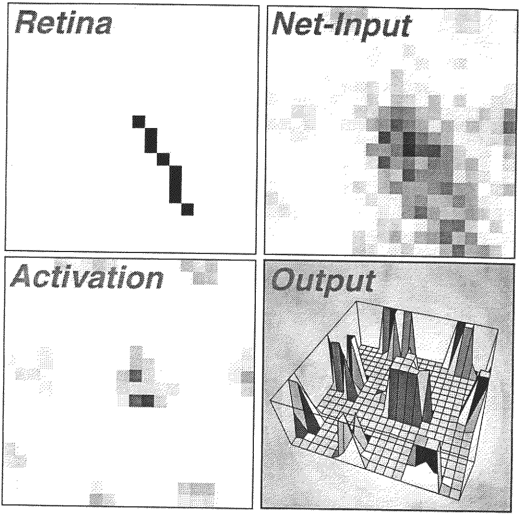


Figure 2. Multiple Activity Bubbles in a Brain Area. The strikingly even spacing of the bubbles in this network output is very similar to the Kohonen resimulation. The plots represent different stages of the neural architecture, as explained in Section 3 and Appendix B. In animated form, there is a motion to the bubbles also, which implies that they have some input-based mobility.

pabilities that Kohonen observed in neural networks (Kohonen, 1988), and the recent resurgence in neural networks as a form of Artificial Intelligence (AI) (Sejnowski et al., 1988; McClelland & Rummelhart, 1987; Feldman, 1985).

1.2.1. Bidirectional Interaction (Circular Reaction)

One of the most intriguing aspects of human and other forms of living intelligence is their continual causal relationship with the outside world. From the beginnings of self-replicating cells through societal man, the interaction between self and environment has been one of the most essential aspects of our entire evolution. A longstanding theory considers the possibility that the rapid tripling in the size of the human brain was due in part to the freeing of the hands when man became bipedal. The sudden jump in the quantity and quality of the possible interactions with the world by the hands required a massive buildup of mental material.

This *closed loop* between external world and mind was observed by Piaget in children. He called it *circular-reaction* (Piaget, 1963). Consider an infant, flailing his arms around, who accidentally rings a bell. The "ding" will divert his attention. The baby considers how the sound could have occurred and tries again to make it happen by flailing his arms around again. He may again ring the bell, perhaps with less extraneous movement, pinpointing more accurately how to make the noise occur, or internally, by refining what he was *thinking* when the sound occurred. Soon the infant will become a master at ringing the bell without any flailing at all. He has discovered the key movement that makes the bell ring, and subsequently associated the key thoughts that move his hands to the string and pull it down. In a sense, his thoughts made the bell ring. The change in the infant's world becomes a stimulus which will then alter the infants actions, which will then change the stimulus, and so on. This is the continual causal relationship between an intelligent being and his world.

A set of famous experiments involving kittens further enhance the importance of the closed loop. Held and Hein show that only when changes in visual stimulation are systematically related to self-produced movements will learning of movement occur (Hein & Held, 1963; Held, 1968; Hine, 1963). In other words, a non-causal relation between thought/action and sensory input implies something *beyond* the control of the self, something not related to the self. Without a closed loop, one can not distinguish between self and non-self, one cannot manipulate the environment, and thus, one cannot be intelligent.

With circular reaction, higher intelligence means adding deeper levels of thought, where more powerful models are developed for associating thought with action. The more accurately one can model, the better chances he has to change his environment, plan, and thus survive and procreate.

1.2.2. Dimension Reduction in the Self-Organizing Cortex

At the other end of the spectrum, near the neural level, Kohonen developed a self-organizing mapping which has its roots in biology (Kohonen, 1988). Its importance stems from the clarity with which the mapping demonstrates the topological organization of an input set. Recent research has extended the work in many different directions including developing applications (Kohonen, 1988; Saxon & Mukerjee, 1990), extending biological relevance (Obermeyer et al., 1990), and extrapolating the theory (Poggio & Girossi, 1990).

Kohonen's map is appealing to our research for many different reasons. Firstly, it is a simple model, yet its biological relevance seems strong. Simply put, the model has the capability to organize a set of vectors into a lattice that imposes a relationship between the vectors dimensionally smaller than the original dimension of the vector set. For instance, Figure 3 below shows a mapping of Danish phonemes organized with Kohonen's network. While the input space was a set of vectors of 16 dimensions where each dimension represents a different frequency component, the map of the phonemes is spread across a two dimensional sheet, similar to a cortical sheet of the brain.

1.2.3. Neural Networks as an Artificial Intelligence

Research in neural networks was, for a while, restricted to systems in which the inputs and outputs were in some way predefined. The existence of an external teacher was presumed. Further, the networks were quite simplified and limited to one or two layers. Even systems considered biologically relevant were confined to very few anatomically different areas: Mead's auditory cortex (Mead, 1990), Bower's olfactory cortex, or virtually all the vision work (Grossberg & Todorovic, 1988; Cohen & Grossberg, 1987; Cornsweet, 1970).

Recently, however, a number of attempts have been made to model robot systems whose behavior could not be derived from a few layers of homogeneous elements. Rather, these systems resemble a larger picture more accurately, a picture involving multiple brain areas and interaction with the external world. Some sensorimotor systems have been built with many different layers and different elements (Reeke, 1989; Edelman, 1989). Other developments involve closing the sensorimotor loop as a means of self-organization and self-training (Saxon & Mukerjee, 1990; Edelman, 1989; Mel, 1989; Kuperstein, 1987; Braitenburg, 1987; Brooks, 1987).

Some of these designs are computationally large. For instance one model involves 12 areas with 220,000 neurons and 8.5 million connections (Edelman, 1989). Darwin III, a recent sensorimotor system, developed a nervous system with about 50 repertoires, using 50,000 cells, and 620,000 synaptic junctions (Reeke et al., 1989). All

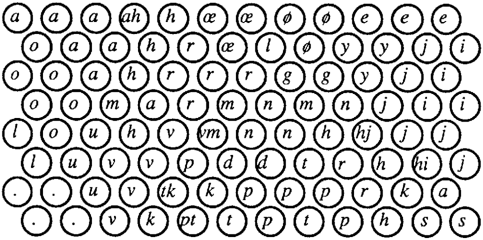


Figure 3. Kohonen's Phoneme Mapping. The circles here represent nodes of a neural network in which the neighboring nodes contain vectors *near* each other. Here, a phoneme is represented by a vector made up of sixteen different frequency detector amplitudes. Thus, the network reduces the dimensionality from a 16-space to a 2-space. Note, though, the occurrence of redundant (degenerate) phonemes such as the 'a'.

the brain areas had quite diverse connectivity schemes because they closely model the brain. This level of compliance to the brain's architecture makes the network very difficult to simulate and also to describe, especially if our interest is not restricted to understanding the brain, but also extends to mechanisms of brain*like* architecture exhibiting intelligent adaptive behavior (brainlike functionality).

On the other hand, sensorimotor systems which are overly simplified or use traditional AI as a high level controller have so far proven to be limited at developing emergent properties and adaptability, although some strides have been made at increasing complexity (Arbib, 1989; Lenat, 1991). While a neural system *evolves* representations of higher level entities and generalizations, the traditional AI *directly represents* the high level concepts. For instance, Lenat's CYC project involves storing massive amounts of common-sense knowledge such as "gravity makes things fall down" directly into a system (Lenat, 1991). A more common high level AI concept is called the *schema*, which represents knowledge of some form that can be manipulated by other schemas or related to other schemas (Arbib, 1989; Glass & Holyoak, 1986). How, though, can there be room for emergent properties if these high level entities are encoded directly? Certainly other means of creating intelligence exist.

Some developments have been made in creating a hybrid model between symbolic AI and neural designs. For instance, in the area of natural language processing using a *connectionist* (neural) network, a symbol can be represented simultaneously by a single node and also a lattice of interconnected nodes (Lange, et al., 1989; Lange & Dyer, 1989). Another form of a "distributed" schema has been developed by Arkin, in which an array of vectors represents the *gist* of the schema (Arkin, 1989).

Extending the notion of hybrids by placing both biology and psychology in the same forum under a theoretical evolutionary framework allows us to begin to study intelligence and its relationship to computers. This broad endeavor has recently been gaining popularity throughout the scientific community (Kohonen, 1988; Edelman, 1989; Churchland, 1985; Brooks, 1986)

1.3. Rationale

While this thesis does not propose to settle grand issues, there is clearly a need for our broad emphasis to have some theoretical basis. Evolution-based Meta-Brain Principles (MBPs) have been synthesized to serve this function (Appendix A). They provide an evolutionary progression of abstract building blocks for the design of intelligent media.

We study sensorimotor systems because they provide a simple model of the closed loop that has existed in our evolutionary heritage. They therefore provide an excellent forum for evolving intelligence.

Neural networks are also implicated by both evolution and their role in sensorimotor systems. Their ability to provide emergent qualities, such as the self-organizational capabilities described above, is not easily found in other modeling methodologies. They also provide an easier method of integration of multimodal signals. Neural networks promise to provide many alternate and powerful models of sensory input, motor control, and eventually cognition.

Presently, neither traditional AI systems nor neural network systems can interact with their environment across situations of great diversity. Yet, the immediate reality of creating neurally based systems better than traditional ones has not been accomplished.

Nonetheless, the emergent and integrative capabilities of neurally inspired networks, the importance and simplicity of the sensorimotor systems, their relative ease of simulation, and finally, the powerful framework of evolution make the study of neurobotics an important stepping stone in the development of robotics and a source of gaining insight into the workings of higher intelligence.

1.4. Overview of the Thesis

In the next section, we discuss a theoretical framework from which we compare the various sensorimotor systems (Section 2). We then discuss the Neurobotics Simulation Package (Section 3) and simulate a prototype sensorimotor system called Neurobot II (Section 4) using the package. Finally, we will discuss the results of our simulations and their relevance to the field (Section 5).

2. SENSORIMOTOR SYSTEMS

A sensorimotor system involves both the sensory input and the motor output of a system. If we add that its percepts are causally related to its motor effects, and that the system is neurally inspired, we create the relationship to neurobotics. The central aspects of our sensorimotor system are the bidirectional interaction, neural inspiration, and also the simplicity of a physical underpinning as opposed to higher level cognitive systems.

The various complexities of sensorimotor systems are the topic of this section. We impose a taxonomy upon these sensorimotor systems by their predominant timescape type. Basically, the interaction between an entity and the external world has different effects on the entity for the different timescapes. These timescapes are based upon the period of time necessary to make the effect. We show how these effects allow "meaning" to emerge from the closed loop (Cliff, 1990). We follow this with a description of the robot arm/vision system, including the kinematics and dynamics problems faced by robotics research. Finally, we discuss some recent sensorimotor systems.

2.1. The Timescapes of Intelligence

From the point of view of mammalian intelligence, we can segment the passage of time into three different size chunks (timescales) that elucidate different aspects of intelligence. The question to ask is, "What are the different issues and alterations that arise from sampling over different timescales, and what do they mean in terms of the actual creation of intelligence?"

Pleasure and pain would be considered contributions on the largest timescape. Neurons detecting heat or wet are physically ingrained elements of our thought processes. The brain's physical architecture is based upon evolution-scale alterations. The form of the neuron itself is also an evolutionary endowment because it is expressed very similarly throughout the body.

Learning, on the other hand, that heat causes pain, emerges over a shorter timespan because it relates to an individual's personal environment rather than an entire species'.

Finally, the shortest timespan represents the actions, perceptions, and decisions of the individual. These acts integrate all perceptual and hedonic information into an immediately applicable response.

Let's choose some appropriately spaced timescapes based upon the above breakdown. Our first timescape, the largest possible one, would be the one that built the physical brain over time, which we'll call the *evolution-scape*. The next slice, based upon the effects caused by the environment over the lifetime of a single being, or the *development-scape*. Finally, the instantaneous thoughts and actions of a being would constitute the smallest timescape, the *thought-scape*. These three timescapes represent different aspects of intelligence from a *hardware, wetware*, and *software* perspective. Figure 4 shows the three timescapes in terms of the *computation** that each timescape performs.

Table 1 summarizes the functional relationship between the three intelligence timescapes. For each timescape type we wish to look at both psychological and biological elements. For example, in the thought-scape (type III), the biological aspect refers to the neural activation and transmittal, while the psychological aspect refers to the thought processes and decisions that occur.

Using these timescapes, we wish to divide the neural network and sensorimotor research into one or another predominant timescape. Below, we begin with the thought-scape, giving an example of a neural network focused upon the immediate circumstances and then continue through the slower changing timescapes, giving examples as we go.

2.1.1. *Thought-scapes, Type III Intelligent Systems*

Biologically speaking, the type III system level is the level at which an intelligent system acts in real-time -- dealing with moment-to-moment actions such as attending an object, recognizing a scene, deciding upon something, making movements, or even consciousness. At this level, no learning is involved. An action is carried out based upon the individual's present mind state and all his inputs, both internal and external. Any resulting adaption would be considered part of the development-scape's domain.

2.1.1.1. Activations and Recognition

We begin our timescape examples with a remarkable neural thought-scape design

---

*We are not trying, here, to make a *computer analogy* to the brain. We are merely pointing out the different components of the physical mechanism, like the hardwired connections have different functions and different meanings from the maleability of those connections or the immediate activation levels of the neurons.

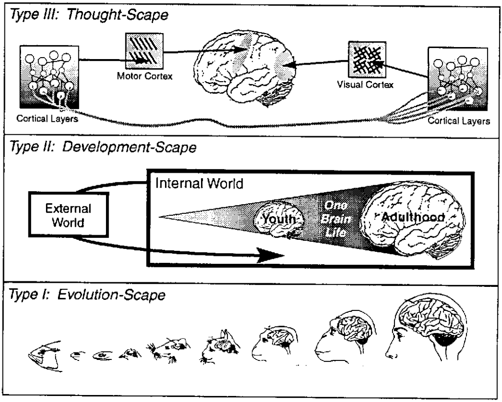


Figure 4. Three Different Brain Timescapes. The *evolution-scape* (I) views the brain from its evolutionary genetic heritage. The *development-scape* (II) captures the importance of the closed loop, including the plasticity of brain development over time, long-term memory, and learning. Finally, the *thought-scape* (III) views the different brain areas under real-time immediate conditions where aspects like short term memory and neural activation are the main topics.

called the Reentrant Cortical Integration model (RCI), which is capable of "seeing" optical illusions as humans do (Reeke et al., 1989; Edelman, 1989). It is also one of the largest neural networks simulations ever created containing multiple brainlike areas with heterogeneous connections and activation schemes and represents a perfect example of a type III network because, while it is very large, it does not contain any learning mechanism. It is based purely on the activations occurring from sensory input and the reentrant connections (see Figures 5 and 6).

The reentrant properties of the RCI model allow each neural repertoire to function independently of the others but also as *coupled-pairs* merging different aspects of the system together to achieve more complex recognition. Although the RCI model was completely "hardwired" by its designers, the neural activations alone

Table 1.

The Psychological and Biological Components of the Intelligence Timescapes Types. Type I components are based upon the entire evolution of an intelligent creature. Its actions are based upon hardwired components which provides pleasure, pain, instinct, etc. Type II components provide adaptability for a creature over a lifetime. Perhaps some aspects of emotion is a learned association between the type I components and experience while rationale is learned abstraction not associated with a visceral component. Finally type III components interact with the immediate real-time environment. Neural activations cause further neural activations in a continual interaction between type I and type II associations.

Type	Timescape	Psychological	Biological
III	Thought-scape	Thoughts/Decisions	Activations
II	Development-scape	Emotion/Rationale	Learning
I	Evolution-scape	Pleasure/Pain	Architecture

were enough to demonstrate visual illusion detecting properties, a common human capability which occurs immediately, without relying upon neural adaption.

2.1.1.2. Selective Attention

Selective attention is another aspect of a type III network. Grossberg and Fukushima have both developed recurrent layered neural networks that perform selective attention, feature detection, and object recognition regardless of object deformation and scale (Carpenter & Grossberg, 1987; Fukushima, 1988). Adaptive Resonance Theory (ART) allows multiple layers of recognition neurons in which each layer is designed to recognize the activity of the previous level (Carpenter & Grossberg, 1987). The multiple layering provides higher and higher levels of abstraction. The model has been shown to compare with biological visual systems. Similarly, Fukushima's Neocognitron uses multiple sequential layers to divide input images into larger and larger chunks of translation and deformation invariant pieces, which finally culminate in a set of abstract recognition signals. His model was inspired by the visual cortex (MacGregor, 1987). Initially the Neocognitron did not use recurrent information. The latest version, however, passes information back from the higher layers in a manner similar to ART, bestowing the Neocognitron with the capability for image enhancement and selective attention. These networks also represent a type II net-

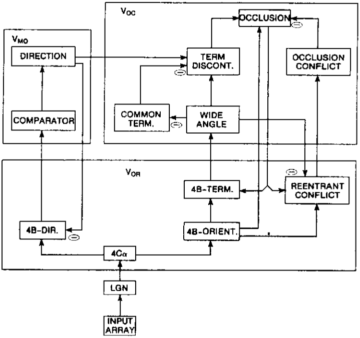


Figure 5. The RCI Model Schematic -- a Type III Network. The above model represents a large group of interacting neural areas somewhat analogous to brain areas. Every area has some specific functionality such as those elements of the visual occlusion area VOC. By providing the reentrant connectivity, the functions merged, each providing information for the other areas as shown in the following figure. From *The Remembered Present: A Biological Theory of Consciousness*, by Gerald M. Edelman. Copyright 1989. Reprinted by permission of Basic Books, Inc., Publishers, New York.

work because they not only perform selective attention and recognition, but can be made to *learn* to recognize new patterns.

2.1.1.3. Biological Correlations

While modeling gives us many interesting results, biological empirical data, if usefully trimmed and abstracted to a manageable level, approaches our goals from the other end. The Dahlem Konferenzen Group Report on Cortical Organization (Stryker, 1988) deconstructed the neocortex while trying to maintain some simplified uniform understanding of the mechanisms explored.

The group discussed differences in the functionality of cortical areas of the brain. Their summary categorized the cortex into three different types of organizational mappings. These were topological, computational, and a special map considered intercalated, also called, interdigitated, or reiterated. A topological cortical mapping

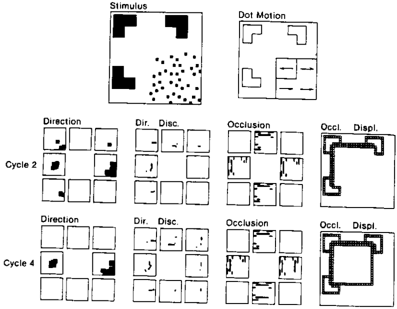


Figure 6. The Optical Illusions Perceived by the RCI Model. The areas shown in the previous figure are now shown as their activations. The input was a real time image that contained two optical illusions with a third illusion arising from their combination. The large image represents the input image. The lower right corner contains a pattern of moving dots in which a backwards "L" shape moves one direction and the rest moves oppositely. In humans, this motion provides the perception of a boundary. By combining visual motion detection with the visual occlusion, edges can be perceived by the model. This facility was combined with the imaginary occlusion that occurs when lines spanning gaps are interrupted. Where humans naturally see a square occluding a number of objects, the RCI model succeeds at this too. The perception of the model is shown in the bottom right corner. From *The Remembered Present: A Biological Theory of Consciousness*, by Gerald M. Edelman. Copyright 1989. Reprinted by permission of Basic Books, Inc., Publishers, New York.

would be a surface on the cortex in which two near points on the surface would be "near" in terms of their meaning. Parts of the auditory system have this type of mapping in which neighboring neurons will be associated with similar frequencies. Kohonen's work represents a topological mapping: consider the phoneme map (Figure 3) in which a near neuron represents a similar sounding phoneme.

The visual cortex is strong example of third type of map discussed by the group report. On a coarse scale, the map is topological, yet at a higher resolution is striated by columns of ocular dominance and also contains groups of directionally oriented neurons. Obermeyer, in extending Kohonen's work, demonstrated how maps with segregated clusters can emerge from cortical networks (Obermeyer et al., 1990). Edelman's Neuronal Group Selection paradigm also expresses the clustering phenomenon (Edelman, 1987). This approach is also taken in our research, leading to our results reported in Section 5.

With the powerful capabilities of the biological neural maps and the type III abstractions that have been developed based upon them, we are given insight into the workings of the moment-by-moment thought process. While neural networks like the RCI model are good examples of neural intelligence, it is difficult to extrapolate from such specific examples to the larger forum of intelligence. It is also difficult to design models in such detail. Thus we take a step back, and look at the brain and intelligence from larger, slower timescapes. By abstracting type III networks into a simpler form, we may be able to add type II capabilities of learning, or type I capabilities of evolutionary bias to create a more generalized format for intelligence.

2.1.2. *Development-scapes, Type II Intelligent Systems*

Table 1 defines type II intelligent systems as learning systems based upon rationale and emotion. Where the thought-scape is based on the immediacy of the present, the development-scape is directed more at acquired knowledge over time. Planning, abstract thought, and modeling the world are the essential elements of the development-scape. At this level we can see the connection between neural networks, traditional AI, and cognitive systems. Also relevant at this level is the long-term importance of circular reaction, because type III networks are basically open loop systems because of their short timescape. Ultimately, a type II network takes advantage of the changing world by learning the causal relationship between its environment and its internal states. From this learning, "meaning" can be established.

2.1.2.1. Traditional Artificial Intelligence

We first turn to the traditional AI techniques for dealing with acquired associative memory. Both Arbib and Arkin have developed approaches which consider the *schema* to be the basis of high level control of the human body (Arbib, 1989; Arkin, 1987). From cognitive psychology, the schema is considered a set of sequenced "if-then" rules guiding a level of behavior (Glass & Holyoak,1986). *Sub-schemas* can be spawned off a schema. The schema concept seems powerful. However, they are symbolic entities and in terms of both their adaptability and physical reality, require much insight to correlate with brainlike mechanisms. Further, while schemas represent knowledge and experience, how can they be extended to learn or adapt, the essential aspect of a type II system?

Nonetheless, processing and modifying symbols can not be wholly ignored. The point here is that symbolic concepts, such as schemas, should *emerge* from tiers of smaller scale structures; the interrelations between them can only be adequately described, or designed, by sub-components. Arkin's schema design bridges some of this gap with his concept of a schema.

Bearing a similarity to the hierarchical nature of Grossberg's ART and Fukushima's Neocognitron, Arkin defined a schema as a topological map of vectors. The distributed nature of the vector mapping makes Arkin's design a more neurally plausible concept of a schema, potentially allowing it the necessary learning capability. In his path planning robot, the vector map contains motor direction commands. Maps can be summed and scaled based upon a priority scheme. For instance, a vector map to avoid an object is a set of vectors, in the shape of the object, pointing outwards; the higher the danger of the object, the stronger the vector forces. To move around the obstacle, this map could be summed with a follow path map. Here, all vectors in a lane are pointing down the path while other vectors point towards the path. By summing these vector schemas, the resultant vector map may outline an appropriate path which follows the path and avoids the obstacle. While the direct connection between schemas and biology seems coincidental, Arkin's approach does demonstrate a potential connection.

The pattern recognition and selective attention of ART and the Neocognitron can be considered sensory analogs to Arkin's motor outputs. Sensory input could cause context-dependent activation in higher levels of recognition. Activations then cascade back downwards, causing entire vector sets to activate simultaneously similar to the schemas.

2.1.2.2. Neural Model Learning

The Neocognitron and ART also provide the essential component of the development-scape: learning. Unlike the customized RCI model, the much more uniform models by Fukushima and Grossberg demonstrate the capability of a neural network over time to recognize previously presented information. Perhaps with more integration of sensory inputs, reentrant connectivity, and multiple areas, type II networks like these could develop the capability to differentiate between different inputs and associate them with different internal states.

2.1.3. *Evolution-scape: Type I Intelligent Systems*

Evolution is by far the most influential contributor to a brain's intelligence. Its presence is known to us in both physical and psychical ways by the characteristic elements throughout anatomy such as the many different types of neurons that all perform the same function, and the characteristic ways in which we act and react under various circumstances such as the feeling of pain from losing a loved one, or the pleasure of eating after fasting. Evolution has given intelligent beings a strong set of physical and mental capabilities with which to continue their species. In other words, evolution provides the framework for thought and the means of supporting the framework.

2.1.3.1. Evolutionarily Defined Concepts

Physically, the means for supporting the intelligence framework are evolved invariant *concepts*. Throughout the brain, in different scales, these concepts have manifested themselves. Such concepts include the generic neuron or the complicated neural architecture common to all members of a species or evolutionary order. Neurons of similar makeup are found throughout the brain (Bullock et al., 1977). Another invariance which appears consistent across the cortex is cortical layering. For example, in the somatosensory cortex, the middle layer receives afferents while the upper layer receives projections from other cortical areas (cortico-cortical connections) and the lower layer receives and sends from/to sub-cortical areas like the thalamus (Crick, 1987).

2.1.3.2. Hedonic Contribution

While the physical components of evolution show us a useful method of modeling intelligence, another aspect is the deeply rooted *hedonic* element of the brain, the pleasure/pain principle. Edelman discusses the importance of pleasure and pain *valuation* in associating behavior. In Darwin III he uses valuations to train the robot (Edelman, 1989). These senses are considered *physical* sensations generated within the body. The importance of the hedonic senses is their intimate relationship with the survival of species. Organisms which have survived and evolved, such as humans, have developed strong psychophysical connections with the hedonic element. Pain is like a subtle hint, implying death (death of a species implying death of an individual), while pleasure is associated with health and procreation. Moravec, in speculation, considers the general pleasure/pain learning mechanism a possible explanation for emotions (Moravec, 1989). In other words, a type II system learns to associate pleasure created by a type I aspect of a system with the immediate thoughts of the type III aspect, generating emotion. For instance, losing a sexual partner is evolutionarily a painful feeling because it decreases the possibility of species survival. Thus there is a built in pain mechanism which has evolved over time. The pain gets associated with the loss. The type II aspect binds the emotion with the pain thus adapting so it can avoid the situation the next time.

2.1.3.3. Hardware Subsumption Implies Need Subsumption

Over time, as positive and negative influences are stockpiled by the evolving intelligence, levels of priority, or levels of needs, emerge. The lower needs become ingrained physically while the more recent higher needs remain somewhat less *hardwired*. For instance, the pain of hunger is a very high priority and also of very early vintage, and, as we know, the loss of food in our stomachs causes a direct sensation of pain. A lower priority would be the need for reproduction, which isn't so clearly associated with a physical sensory connection, only a feeling that it is a pleasant thing. As we might imagine, the pain of hunger would, under the right circumstances override the need to reproduce. In other words, the higher level needs are subsumed when lower needs are not satisfied.

We can see how the needs that Maslow defines in his "hierarchy of needs" could possibly emerge from the effects of pleasure and pain as humans evolved (Maslow, 1970). The hierarchy represents the emotional manifestations of the evolutionary choices that occurred in our path to humanhood (see Appendix A).

A mobile robot with an analogous hierarchy of needs was implemented by Brooks in what he called a *subsumption architecture* (Brooks, 1986). Strangely, Brooks never explicitly mentions the connection between his layered architecture and evolution. In his non-neural design, he diverged from traditional robot design methods of step-by-step modules (such as image processing, object recognition, planning, action, and motor control) for a strategy similar to Maslow's tiers. His robot began as a fully operational but simple machine. Higher functionalities were integrated (evolved) on top of the existing layers. Underlying layers were not redesigned, but augmented by higher ones which intercepted system signals and merged new signals into the outputs.

Brooks' robot's first layer simply avoided obstacles (i.e., it would move away from any ominous object nearer than a certain distance). The second layer provided for wandering (e.g., move in a random direction for a random time). By applying the second level upon the existing first level, the robot would wander as long as no obstacles became threats. A potential collision would cause the strength of the lower level's avoidance signals to increase until they inhibited the wander signals. Brooks' eight layer design interacts with the environment in increasingly complex ways. The final stage would give the robot the human capability of "logical thinking about the environment" (Brooks, 1986).

Noticeably, the hierarchy that Brooks describes has a relationship to Maslow's hierarchy. We can call Brooks' eight tiers "Brooks' Hierarchy of Robotic Needs." Although perhaps somewhere at the low end of the hierarchy, he should have included "Seek power source." Figure 7 demonstrates the parallels. Another commentator on evolution and computers was Minsky, who compared software design to evolution: one should append new code to repair software bugs rather than replace the buggy code (Brandt, 87). The concept of incrementally adding to a system, rather than redesigning, has also been discussed by MacGregor (MacGregor, 1987).

2.1.3.4. Consciousness

We close our discussion of the evolution-scape with an example of the ultimate extent of a type I network. In *The Remembered Present*, Edelman carries reentrant integration to a logical extreme based upon the architecture of the brain. He considers internal brain loops and areas that combine the human's hedonic value system with external sensations and actions to give rise to our *primary consciousness* (Edelman, 1989). The hedonic contribution, along with the extremely complicated loop architecture, is the contribution of the type I intelligence. Continuing, Edelman further considers the human capability for language, arising from Broca's and Wernicke's areas, to be the unique components giving humans *higher-order consciousness*. These language areas of the brain allow humans to surpass other mammals by applying words, through a combination of vocal, aural, and visual stimuli, into a timeless symbolic meaning for actions and objects and the other symbols of the real world (see Figure 8). Edelman's postulate is the ultimate integration of type I, II, and III systems.

Summing up Table 1 and Edelman's theory, we see that the evolution-scape intelligence provides the ingrained hedonic sensations and the neural architecture for developing speech and thought. The development-scape uses this "hardware" to associate emotions with the hedonics, symbols with perceptions, and rationale with

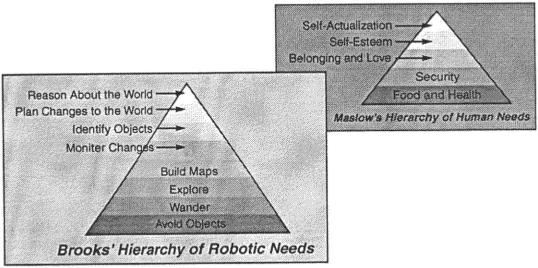


Figure 7. Hierarchies of Robot and Human Needs. An interesting parallel can be seen between Brooks' robot "needs" and Maslow's hierarchy of needs. Maslow's hierarchy emerged from evolution's continual augmentation of the brain.

the symbols based upon one's lifetime environment. And the thought-scape makes the whole system "alive".

Throughout the thesis, we will freely make use of the timescape-based type definitions with respect to intelligence, systems, neural networks, etc.

2.2. "Meaning" in Sensorimotor Systems

Maintaining a continual causal interaction with the external world in an adaptive sensorimotor system should eventually create a network of associations between the various internal and external elements. These associations have "meaning" in the sense that a set of biased weightings in one element of the system *represents* a set of biased weightings or causalities in another part of the cyclic system, like internal and external (Cliff, 1990). For instance, "I see a dog (visual impression on the retina going back to the visual cortex) ; I think 'dog' (pattern of activity further back); I pet the dog (pattern of activity at the motor cortex); I see and feel myself petting the dog (visual impression of self petting something); I think 'dog' (pattern of activity)..." ad infinitum. Unlike artificial neural networks in which both the inputs and outputs are defined by humans, we see that a closed loop system grounds its meaning only in its other elements: the elements must "model" the rest of the elements or the world.

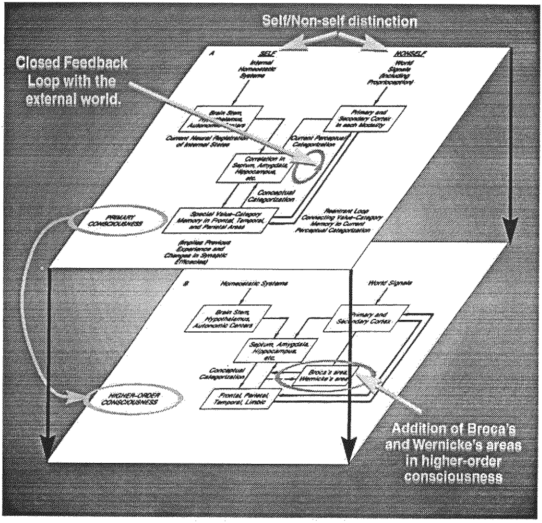


Figure 8. Levels of Consciousness -- The Ultimate System Design. Edelman's system approach extends the external world loop into a larger picture. From A to B, speech is added to the system, which makes the two levels of consciousness extremely different in terms of their capabilities. A is bound to the present by it's closed loop interaction with the external world. Through its language-augmented loop, B can replace the world signals with its own, thus detaching itself from the bounds of the present. From *The Remembered Present: A Biological Theory of Consciousness*, by Gerald M. Edelman. Copyright 1989. Reprinted by permission of Basic Books, Inc., Publishers, New York.

In designing his robot called Murphy, Mel pointed out that the meaning inherent in each node "is possibly the single most important design decision in a connectionist learning system, having a profound impact on virtually all aspects of system learning and performance behavior." (Mel, 1990)

Let's consider the infant ringing the bell. His "system equation" looks something like Figure 9. This BrainFunction can lead to a non-predictable response. Consider, for example, the sequence of numbers produced from the equation:

$$f(t) = 4\lambda f(t-1)(1-f(t-1)),$$

which, when $\lambda > 0.89248641$ becomes chaotic. And f(t) is simply based upon its previous value as shown in Figure 10 (Hofstadter, 1985). If a thought could be reduced to a single number between one and zero, a simple recurrent equation of this nature functioning within a black box would make the thoughts extremely difficult to characterize *from the outside*. Internally, though, the associations between present and past, thought and perception, can be learned by the brain because all variables are known.

PresentThoughts = BrainFunction(PreviousThoughts, NewSensoryInput)

or

$$T(t) = B(\, T(t-1),\, S(t)\,)$$

Figure 9. The Brain Function. This recurrent abstract equation which represents the brain's infinite spectrum of actions and thoughts based upon the previous thoughts and the new sensory inputs being received.

The infant is the soul entity observing the world by his own senses, so his view is inwardly unique. It only has meaning to him, in that he could not explain to somebody else how they could think the bell to ring, but only how they could move their hand to perform the same task. The baby's actual thought process for ringing the bell may be completely different from that of another baby. Churchland discusses this difference in qualia (Churchland, 1985). The thought process it takes for the infant to ring the bell is completely objective *and* completely meaningful, but only to the infant himself. Thus, while there is meaning in closed systems, it is unique to the host. To pass the information would be merely syntactic, while the semantics would be inexpressible.

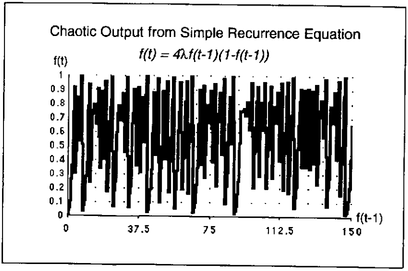


Figure 10. Chaotic Behavior of a Simple Recurrent Equation of One Variable. This shows a chaotic pattern that occurs simply by raising λ above the value .8924.

With respect to meaning, sensorimotor systems remain at the level of motor action and reaction. The emerging results are grounded in some sort of physical/spatial patterns. Thus they are easy to analyze.

2.3. Basic Components

The general components of the sensorimotor systems described below are an arm, an eye, and an associating mechanism: a "brain". While the ultimate neurobotics creation would be to replicate an infant's brain and observe the association in action, there are many steps to take before such a system is even conceivable. Every element of the closed loop is yet another module to create. Unfortunately the model is then only as strong as its weakest module. But modules need not be complicated, merely robust (Cliff, 1990), efficient, and self-consistent (Kuperstein, 1988a).

Each of these sensorimotor components by themselves are topics of ongoing research: physically-based modeling, muscular control, computer vision, associative memory, etc. An oversimplified model of the individual elements is the price paid to develop a full sensorimotor system, the focus of neurobotics. Perhaps this explains why our seemingly global viewpoint is not so heavily trodden.

The main constraint of the closed loop sensorimotor system is that each element of the system affector, effector, brain, and external environment, is capable of transforming the information of the previous level at the same resolution. Meaningless results occur if, for instance, the retina's resolution prevents differentiation between the links of the robot arm. A further constraint is that the change in any element in the system has a causal and consistent bounded level of response. Mel's Murphy had an interesting quirk: when the robot arm passed through the center of the camera's field of vision, the autofocus mechanism would change the focus of the image. Although this change in focus was unusual, it was consistent. Murphy was successful at its tasks; its resolution, causality, and consistency were adequately bounded for the malleable parts of the system to adapt successfully (Mel, 1990). We strive for this same "smoothness" of response.

2.4. Kinematics and Dynamics

The above discussion has now covered some of the higher-level issues involved in developing a sensorimotor system. Below, a number of sensorimotor designs are described. However, we must first preface their discussion with a simple definition of one of the problems in closing the loop.

In our robot scenario, a position in space can be described in two different ways: with respect to a robot arm, or with respect to the eye. From the eye's point of view, the point takes on simple Cartesian coordinates. The arm, though, in order to reach the point, must set each of its joints to some angle. Figure 11 shows the difference between the two spaces. The equations which convert joint angles to positional coordinates are the forward kinematic equations. The reverse are called inverse kinematic equations. When there is more than one link on a robot arm, the number of possible arm configurations that reach a specific point can become infinite. This is the problem of manipulator degeneracy: no simple expression exists for the inverse transformation, from a effector position in Cartesian space to a point in joint space (see Figure 12).

Researchers have attempted to solve the direct and inverse kinematics problem of the robot arm using neural networks. Throughout the systems described below, kinematics and dynamics are limited to make them manageable by the neural networks. For instance, some inverse kinematics transformations were trained to consider only a single correct inverse solution. While this artificial constraint solves the problem of degeneracy, it loses the generality so evident in human movement. Other systems encode a specific set of inverse transformations by limiting considera-

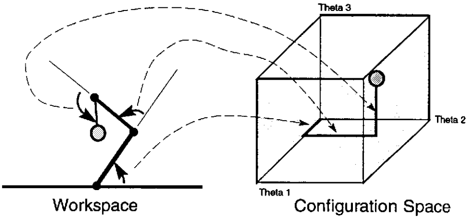


Figure 11. Difference between Cartesian Image Space and Joint Configuration Space. Forward and Inverse kinematics reflect two different ways that a robot arm can be encoded. From a visual point of view, a two dimensional robot arm with three degrees of freedom looks like a set of lines in the Cartesian Images space. Alternatively, its joint angles, represent a point in a three dimensional space called the configuration space.

tions to the simple case of the inverse differential dynamics, which is a linear approach (Mel, 1989).

2.5. Recent Sensorimotor Systems

In the following, a range of sensorimotor systems that have been developed are discussed. We find that most of the systems are type II (learning) networks which basically focus upon simple architectures. Some systems are staged to learn (type II), then perform (type III). While neural populations are used in all cases, in most, they are non-biological in their schemes. Fully connected networks, while implying the potential for some kind of global optimization, are examples that stray from biology. In general, the systems are neurally inspired and have a causal interaction with the external world. Brooks' robot managed to accommodate a type I contribution (Reeke et al., 1989; Brooks, 1986). Our final example, Darwin III, also does justice to all three system types.

2.5.1. Neurobot I

Neurobot I serves as a starting point for this thesis. It was a simple type II system that exemplified the need for a closed loop system (Saxon & Mukerjee, 1990; Cliff, 1990). It made external and internal world representations. The Neurobot "brain" was a single self-organizing cortical sheet of neurons into which two sets of coordi-

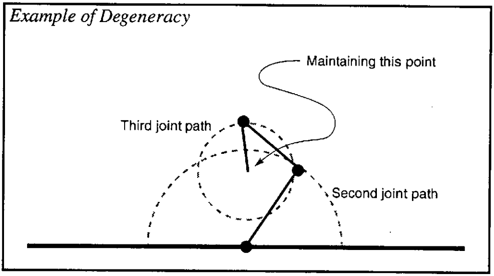


Figure 12. Degeneracy. This nonlinear mapping from image to configuration space is further aggravated by the degeneracy of the mapping. Suppose the robot has to reach a goal point. For a three degree-of-freedom arm, there may be infinitely many positions that the robot arm can assume to reach that goal position. This degeneracy is a characteristic problem associated with inverse kinematics. The third joint of the robot arm (above) can be positioned anywhere on the black circle and still maintain contact with the goal point.

nates were passed. The end effector position in Cartesian coordinates of a two degree-of-freedom robot arm constitutes the first set, while the second set was the same point represented as joint angles of the arm. Figure 13 diagrams the Neurobot I system. In this simple case, the external world was represented entirely by the two equations which relate the end effector coordinates to the joint angles, as shown in Figure 14.

The design of the original Neurobot was based upon self-organization, as described by Kohonen (Saxon & Mukerjee, 1990; Kohonen, 1988). Every node in Neurobot I's cortex received the same four dimensional input vector (the Cartesian coordinates of the end effector and the joint coordinates of the arm). The input vector was then compared with an internal weight vector in each node. The "winner" of the nodes was defined to be that node whose internal weight vector most nearly matched the input vector. As discussed above, Kohonen's abstraction provided that there was only one winner on the cortical sheet. The winner, along with its topological neighbors, are then "pointed" more directly along the input vector. This consti-

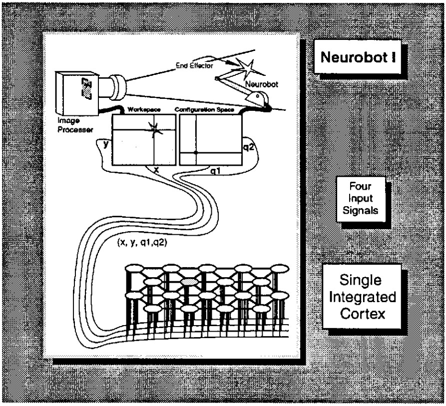


Figure 13. Neurobot I. Neurobot I is comprised of four inputs entering a Kohonen inspired self-organizing cortex. Two inputs come from a "camera" that detects only the Cartesian position of a bright light at the end effector. The second two inputs come from the joint positions of the robot arm. These signals together form the input which is eventually learned and topologically mapped to the cortex.

tuted the learning in the system. Eventually the physical connectivity of the cortex imposed an organization on the nodes so that the two dimensional cortex represented the four dimensional space in which the robot end effector could move.

The mapping, between the two coordinate spaces, learned by the Neurobot cortex as shown in Figure 15 is not only nonlinear but also degenerate. This is shown by the fold in the workspace image of the network. In the configuration space, the same net of nodes has no fold, meaning that some points in the workspace are represented by two points in the configuration space.

These constraints of using the Kohonen paradigm provide adequate reason to look into other methods of creating robotic/visual sensorimotor systems.

$$x = \text{Link1} * \cos(\theta 1) + \text{Link2} * \cos(\theta 1 + \theta 2)$$
$$y = \text{Link1} * \sin(\theta 1) + \text{Link2} * \sin(\theta 1 + \theta 2)$$

Figure 14. Forward Kinematic Equations for a Two Degree-of-Freedom Robot Arm.

2.5.2. *Ackley's Alvis*

Through a backpropagation paradigm, Ackley's approach accomplished similar goals to Neurobot I (Ackley). While Neurobot I integrated both joint and position coordinates into a single self-organizing mapping, Ackley's approach, instead, encodes a feedforward network to solve the forward kinematics problem of a multi-joint robot arm. A traditional backpropagation algorithm is used. His method of solving the inverse kinematics was to use the back-propagation paradigm to propagate an error *back* to the joint angle inputs. The outputs become the inputs for the inverse solution. Selected coordinates of the robot arm were input and their error deltas calculated, while other deltas were set to zero. The error propagated back to the inputs is now a gradient descent in the input space.

In addition, Ackley's network was based upon his concept of input-dependant "attractors" and "repellors". These were an enumerated set of units which would inhibit potential choices that previous trainings had shown either successful or unsuccessful outcomes. The repellors, controlled by the winner-take-all "don't-be" units, may function similarly to the mammalian cerebellum in providing extraneous movement inhibitors (Edelman, 1989). The drawback of this approach, as Ackley mentions, is the need for a specific number of "don't-be" and "do-be" units for every goal position.

The learning nature of Neurobot I designates it as a type II timescape model. Yet, the success of the network to categorize is limited by the abstractions that Kohonen places on it. In the degenerate cases of the mapping, two sets of nodes could potentially be winners, thus both could be activated. In higher dimensionality situations, the number of winners could increase dramatically. By avoiding the abstraction devised by Kohonen, multiple activation could allow cortical sheets with more powerful organizational capabilities.

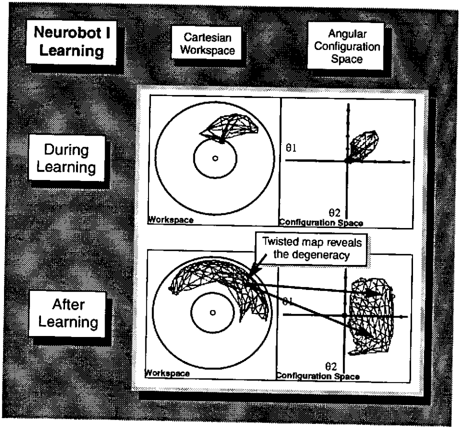


Figure 15. Neural Net Mapping of the Four Dimensional Space. The two meshes shown represent the same network spanning the Cartesian workspace and the configuration space. The seemingly tangled network in the image space is actually a proper mapping, demonstrating the capability of the cortical layer to account for the degeneracy in the mapping.

The Kohonen paradigm tries to impose a smoothness over the entire surface. As described above, the Dalhem Konferenzen has shown many types of cortical maps in which the topographic map appears to be smooth at one resolution, while at a higher resolution, the map is actually fragmented (Stryker, 1989).

2.5.3. *Kuperstein's Sensorimotor System*

Kuperstein's sensorimotor system uses a paradigm involving a set of multiple integrated repertoires, as opposed to the fully connected layers of independent neurons typical of backpropagation. His system is connected to a five degree-of-freedom robot arm, and a binocular vision system (Kuperstein, 1988a). Kuperstein's simulated robot foveates upon a cylinder in three dimensional space and then reaches for the cylinder by generating the proper muscular signal. The weights are trained by error signals representing the error in foveation, or the error in arm position. Mimicking human learning, the weights slowly acquire accuracy, first perform poorly then dramatically improve. Positional errors occurred 4 percent of the time and the error in arm orientation on the average of 4 degrees.

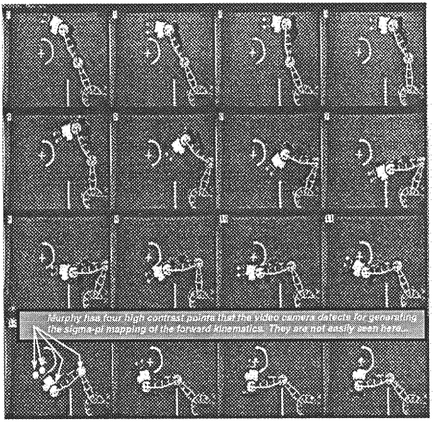
Kuperstein used two hypotheses related to bidirectional interaction to create his system. One based upon Held & Hein's kitten experiments, which he called "self-consistency", and the other extending Piaget's circular reaction paradigm (Held & Hine, 1963). Upon these two hypotheses, Kuperstein developed the system with the two sensory modalities considered. He connected the sensory modalities of binocular vision and the muscular eye position with the joint output of the arm. This was accomplished by first dissecting the two sets of inputs into smaller elements gated by large weight maps. The output of the two maps were then united in the motor map. The arm motor signals also included a random activity component.

A two step process was used to learn the appropriate motor commands. First the arm holding the cylinder was moved around the image-space, thus creating the self-consistent correlation with the image and the arm position. By changing the all weights in the two maps by a fraction of the error between the output and the real position, the arm slowly learned to accommodate all positions and orientations of the cylinder.

Similar to the robotics systems above, Kuperstein's robot does not actually see the arm of the robot in the visual input. The architecture, while being distributed, is very regimented. The correlation between the two sensory inputs and the output are specific to the situation. An obstacle infringing upon the workspace is not compatible with his model, although by augmenting the system with some form of selective attention and high level planning, a series of positions could be selected, or a path could be generated, which would guide the arm around obstacles. Below, Mel's robot arm Murphy, using a planning system and partial connectivity, achieves path planning through a field of obstacles in two dimensions.

2.5.4. *Mel's Murphy*

Murphy, a three degree-of-freedom robot arm, can navigate through a field of obstacles from a starting point to a goal point (see Figure 16). The path planning is accomplished by a heuristic based upon possible trajectories and their intersections



Murphy has four high contrast points that the video camera detects for generating the sigma-pi mapping of the forward kinematics. They are not easily seen here...

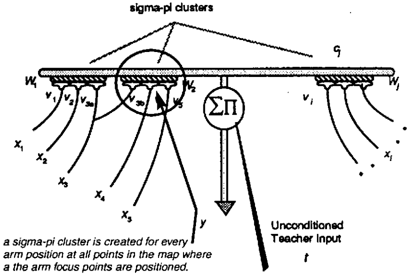
Figure 16. Murphy Navigating Through a Field of Obstacles. This series of sequential images, shows this vision-guided robot arm navigating through a field of obstacles. Part of the figure is taken from from *Connectionist Robot Motion Planning: A Neurally-Inspired Approach to Visually-Guided Reaching*, by B. Mel.

with objects. Forward and inverse differential kinematics are learned with a neural network of sigma pi neurons (Mel, 1990). One of the striking elements of Murphy is that it used an actual robot arm and video camera to close the loop with the external world. Mel further relates his work to aspects of neurology, psychology, and robotics, making the Murphy a very diversely inspired creation. From a neurobotics point of view, Murphy has type II and type III capabilities. Part of the network stops learning after learning the forward kinematics nodes. The inverse dynamics nodes, though, continue to learn.

The sigma-pi units used by Murphy are a level more complex than traditional linear threshold neural elements: a conjuncted cluster of inputs are summed with other conjuncted clusters to form a node (see Figure 17) By carefully designing a network of a small set of neural repertoires, Mel's robot arm could, exploring the entire image-space only once, store the full forward kinematic map. (Note that this direction is not degenerate). This design implied a high connectivity between areas while the training results in low connectivity. A linking strategy that integrates the various neural maps was the key to this encoding. For instance, when training the forward mapping, the joint positions were encoded in a strip of neurons; joint position would cause activity to occur in a portion of each strip. Rather than linking all nodes to the clusters, only the "winners-nodes" of each joint strip were chosen to be conjuncted together and added to the visual nodes . This reduces dimensionality and increases the speed of the network. It further implies a topological mapping because the smooth displacement of the joints transform to a smooth displacement of the visual stimulus. This *reduced* linking strategy, connecting only the most active neurons, is carried further. The training signal for each visual position, in a given arm position, was the amount of activation received by Murphy's eye. Only in the visual areas where there was stimulation would a sigma-pi cluster be added to the neuron. This system is straightforward, using an incremental one-time building of a "look-up-table" and uses kinematics rather than dynamics. Mel conjects, similarly to Bower, that constrained by memory and time, the brain will evolve to choose a simple solution (Bower, 1990). Also, while the method of developing the neural system is simplistic, Mel argues that the resultant network is similar to biological neural circuitry.

Similar to Alvis and Neurobot I, rather than use the full image of the robot arm, four landmark points were used to represent the position of the robot arm. Unlike the previous models, though, these points were placed on a full visual field. The joint angles also, as described above were encoded as a strip of neurons. Using the sigma-pi paradigm, Mel observes that the difference between encoding the full arm versus encoding the landmark points is only increased memory requirement.

Mel has correlated Murphy's learning strategy as being psychologically relevant in a number of ways including (1) Piaget's staged reaching abilities and "learning by doing", (2) the time it takes to learn visually guided reaching, and (3) the mental re-



a sigma-pi cluster is created for every arm position at all points in the map where a the arm focus points are positioned.

Figure 17. Sigma-pi Neurons. Murphy's neuronal architecture is based upon creating a new cluster of conjuncted links for every visual field neuron experiencing a signal. The visual nodes are activated by the visual input, the correlated joint angle nodes of the highest activation are placed as a cluster on the node. The cluster weight is then set to be proportional to the visual activation. Future stimulation of similar joint configurations will generate a visual activation that is appropriate. Part of the figure is taken from from *Connectionist Robot Motion Planning: A Neurally-Inspired Approach to Visually-Guided Reaching*, by B. Mel.

hearsals that go along with manipulating an internalized object (such as mental rotation).

2.5.5. *Edelman's Darwin III*

While Murphy's cross between design and biology tends toward the engineering side, Edelman and Reeke's Darwin III is by far the most biologically inspired, complex, and extraordinary representation of a robot arm/eye sensorimotor system (Reeke et al., 1989; Edelman, 1989). We see it as an example of a fully integrated type I, II, III system: it contains complex biological neural populations, it has a strong closed loop, and learns external causality, and by designing hedonic valuation into the system, it not only learns internal causality but can be considered an evolutionary (type I) component.

Consider the resultant behavior of the system in Figure 18. The work involves the most endowed system created to date: encompassing circular reaction, biological depth and breadth, emergent behavior, and epigenetic and phylogenetic develop-

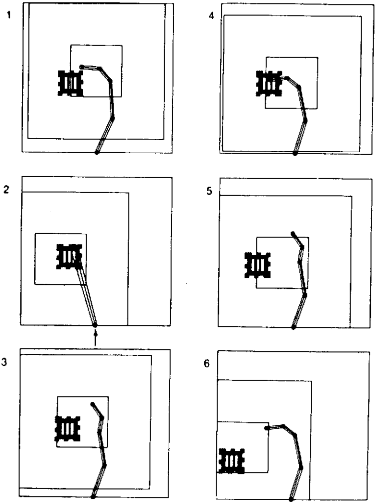


Figure 18. Darwin III's Emergent Behavior. Frames 1-6 show Darwin III foveating upon an object (the small square represents the center of vision), reach for it, examine the entire perimeter of the object, and repell it by a whacking it out of range with a rhythmic "rejection response". While the center of vision still follows the object. This surprisingly lifelike behavior is almost completely architecturally based. It involved a number of different "value" affects which, when combined with the neural architecture, give rise to the responses without direct human intervention, apart from setting up the initial conditions. Figure reprinted by permission of Basic Books, Inc from *The Remembered Present: A Biological Theory of Consciousness*, by G. Edelman. ©1989 Basic Books, Inc.

ment. Darwin III connects more than 50 different "repertoires" of brain areas and is closed by an external feedback loop.

Various aspects of Darwin III have been discussed throughout this thesis, leaving still much more that can be described. The work of Edelman et al. is so broad and yet thorough that it cannot be fully discussed or incorporated in this present scientific endeavor.

An example of the thorough design of Darwin III is described in Figure 19 which is a schematic view of the motor system that controls the motion of the robot arm above. The connectivity between the layers utilizes a global set of activation equations and learning equations. Yet, the architecture, activation, and learning aspects were all customized to reflect a thorough knowledge of the brain at all levels. For instance, the climbing activation in the motor cortex area represents the progression from gross to fine movement, as noted in mammals.

Neuronal learning was limited to specific sets of connections while the links like RCI were all hardwired The learning paradigm allowed for selection among 6561 different types of neural alteration. Once again, choices made in learning were based upon knowledge of the mammalian central nervous system. In training the robot, no stages were necessary. The system, similar to humans, grew more capable in time without needing to train one layer, than another, etc.

Darwin III's training was in addition based solely upon the inherent hedonic activatory value. Thus, being a type I system, Darwin III *liked* some activation patterns to occur and disliked others. Higher neural activity is considered a positively associating factor. Consider how Darwin III foveates upon obstacles. A brain area was created in which there were a higher number of connections coming from the center of the visual field than the edges. This gave the eye its tendency to center objects in the fovea. Other value systems throughout Darwin III facilitated lifelike behavior.

The emergent behavior of the Darwin III sensory motor system is the most extensive seen to date. Still, there are limitations worth relating. As with the other sensorimotor systems discussed, the visual representation of the robot arm was not entirely a full one. While the obstacles that Darwin III intercepted were visible, its robot arm was not. Only the hand position was encoded through the architecture. Further, while Darwin III generally learns to act based upon increased activation, the same question of how to avoid displeasure is not tackled explicitly. For example, the question of obstacle avoidance has not been considered. It is only mentioned that the rhythmic response, which is not clearly defined, is a type of repulsive reaction that causes recognized objects of dislike to be *whacked* from the image-space. In this strongly biologically inspired system, there is clearly much work to be done to understand the functioning of the brain.

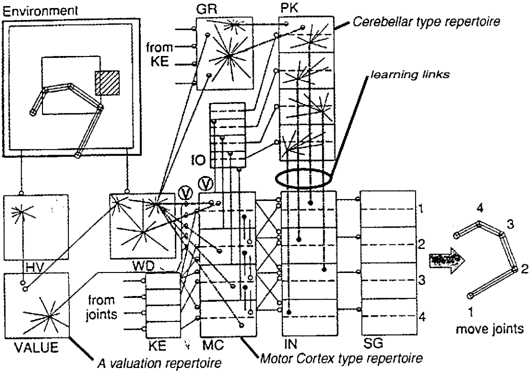


Figure 19. The Darwin III Reaching System Schematic. The many repertoires and their connectivity represent the care and precision taken in designing Darwin III. The dashed lines represent learning links; all others have fixed weights. Hedonic valuations are represented by VALUE. Note also the climbing activation and falling inhibition in area MC (motor-cortex) to cause a spread of activation towards the peripheral joints. Part of the figure is taken from from *Signal and Sense: Local and Global Order in Perceptual Maps*, G.M. Edelman, W. E. Gall, and W. M. Cowan, editors.

2.5.6. Conclusions

Mountcastle pointed out that even the most comprehensive neuronal modeling, in the long run, may be utterly useless in terms of understanding the brain. Nonetheless, each modelling attempt may bring us closer to giving us a model that will prove useful, and at least will help to develop useful applications (Mountcastle, 1990). Darwin III is an excellent example of the types of *intelligence* presently possible to create. Other sensorimotor types of systems such as mobile robots like Brooks' robot system and Braitenburg's vehicles are also eventually candidates for exploration with neural means (Brooks, 1986; Braitenburg, 1984). Combining the planned layers of evolution as Brooks has done, with the neural complexity that Darwin III contains points towards the future of neurobotics.

The remarkable and diverse set of examples of systems described here show the different ways in which a sensorimotor system can be designed and the range between the artificial and the natural that can be chosen.

Below we present the Neurobotics Simulation Package that we have developed based upon the above diversity. It has been built to addresses all three timescape types making it a useful tool to the explore the goals of neurobotics.

3. NEUROBOTICS SIMULATION PACKAGE

The Neurobotics Simulation Package (NSP) is a software system designed to support the study of neurobotics systems. The NSP is built upon an extensible, hierarchical neural network simulation program, but has been expanded to simplify the task of creating large neural systems that focus on brainlike architecture. Also, unlike most neural network simulators, the NSP focuses upon the the topological connectivity common in biological brains. Capable of simulating the internal workings of a pseudo-brain*, the NSP also allows interaction with models of the external world.

The ease of developing virtual worlds with the NSP is made possible because it is designed in CLOS (Common Lisp Object System), an object-oriented interpreted programming language. Based upon Common Lisp, CLOS proves to be quite extensible in a unique way. The multiple-inheritance capability of CLOS allows the NSP to develop programs and neural networks in a way that resembles evolutionary development. In summary, the NSP serves as a neurally inspired framework for building simulations and understanding neurobotics worlds.

To demonstrate the NSP's capabilities, we have built *Neurobot II*, a sensorimotor system comparable to its predecessors described above. Neurobot II serves as a good representative of the systems that can be designed with the NSP. The following subsections discusses the NSP. Neurobot II is described in Section 4.

3.1. Overview

The NSP is built upon a low-level neural network (connectionist) simulation system called DESCARTES (Design Environment for Simulating Connectionist ARchiTecturES) created at UCLA (Lange et al., 1988). Both DESCARTES and the NSP are developed within the Common Lisp Object System (CLOS), which is object-oriented Lisp. If the user wishes, the NSP can also use X windows to display graphical information about the system simulated entities. X windows is called through the CommonWindows facility of Allegro CommonLisp. The NSP is written in DESCARTES/CLOS and comprises about 5500 lines of code. For further information about CLOS and DESCARTES, see Appendix B.

In summary, the basic elements of the NSP are:

---

*Throughout the text we use terms such as brain and pseudo-brain interchangeably. We wish to conform only to various aspects of the mammalian brain. Thus our terms represent a brainlike architecture without sounding cluttery.

(1) *nodes and subnodes* (the neural component of our the NSP)

(2) *pseudo-brain-areas,* (PBAs)

(3) *gross-links* (which are patterned collections of neural links between two PBAs)

(4) *closed-loop cycling protocols*

(5) *visualization and storage tools.*

We will discuss these NSP mainstays, beginning with the visualization tools with which we can then discuss and demonstrate the other elements.

3.2. Visualization and Storage Tools

We begin our discussion of the NSP by demonstrating our various techniques for representing the large quantities of data that our simulations can create. There are two different aspects of visualization that the NSP uses to ease the process of analysis. First, we are able to monitor the progress of our simulation via X windows. Information can be presented about the different elements of a simulation, for instance, a robot arm view or, most importantly, the activity of the different brain areas. The second half of the simulation visualization is accomplished off-line by storing NSP information in files to be read by *Mathematica* (Wolfram, 1989). Storage methods are designed to record information about individual nodes, pseudo-brain areas (PBAs), and gross-links. Once data has been transferred to Mathematica, activation sequences, link growth, and link weights can be observed in real-time through an animation facility provided by Mathematica.

Figure 20 is an example of the simulation-time work environment. Windows for the various elements of a simulation are instantiated with the specific attributes that each element requires. Although the windows have some mouse driven capabilities, main control of the simulation is through the Lisp interpreter shown.

Other NSP windows focus on displaying the PBAs and also provide for other types of windows and further customization. With mouse selection these windows can display the net-input, activation, output, or other user-defined stages of a PBA (as described below). Further mouse controls allow the user to manipulate both functional and displayable characteristics of the PBAs, such as changing the shape of the nodes, redrawing the window, clamping the nodes to a specific output, and toggling the learning and cycling. Thus if a simulation is going awry, the user can quickly analyze where the problem rests and restart it with new parameters.

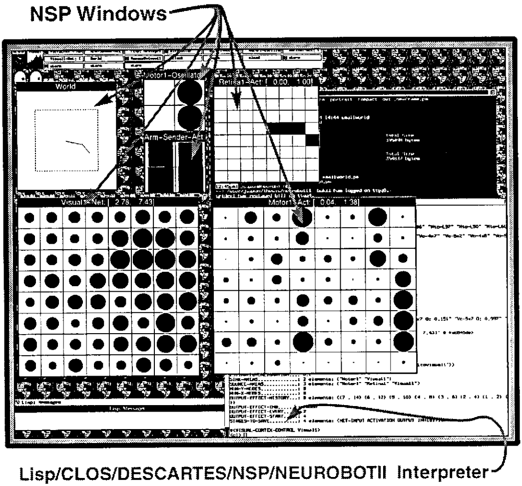


Figure 20. A Neurobotics Simulation Package (NSP) Environment. Shown here is a typical simulation environment in the midst of simulating a eye/arm closed loop containing a *retina*, a *visual-cortex*, a *motor-cortex*, a *motor-oscillator*, and a *motor-sender*. Windows can be iconized, thereby saving processing time, or scaled for detail. The *world* window displays the robot arm. The dotted box represents the visual field of the retina. Notice the scan-line conversion that occurs between the real world and the retina.

A PBA is automatically given a window upon its creation. But when the window is iconized, the areas are not redrawn, thus saving on needed simulation time. Also, windows can be scaled, and the neural elements scale with them, so that details can be observed when necessary (see Figures 20 and 21).

To clarify the distinction between our online and offline visualization tools, below in Figure 22, we show the same brain areas as viewed from within Mathematica.

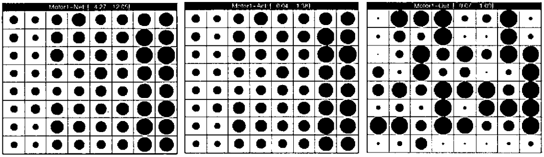


Figure 21. Views of the Same Pseudo-Brain Area (PBA). Here we show the motor-cortex of Neurobot II represented at three different neural stages, the net-input, activation, and output stages. Other stages created by the user can also be displayed.

The density plot gives a different flavor in interpreting the brain area activity. Contour plots, or three dimensional plots can also be drawn from the data. More importantly, though, once the data is stored off-line, it is possible by *animating* sequences of PBA plots, to look at the network as it would actually behave in a real-time environment. Subtle changes can be seen when the many iterations of a brain simulation are flipped through like a flip-book. A side-by-side comparison can not illuminate many of the dynamic changes that occur in a network.

Before a PBA can be characterized, it is usually necessary to characterize both the neural activation and link learning paradigms. We must determine that the nodes and links are operating as we expect. A visualization method allows us to observe single nodes and links of the NSP and plot them accordingly. Figure 23 shows a three dimensional plot representing the activation of a single node for a given range of inputs over time. The figure also shows a single slice of the plot. This type of characterization can be performed on all stages of neural activity and also on link weights.

Data saving methods for each type of element have been designed so that the data may be transferred. The analysis of the activations is an essential element for studying neural networks, especially when observing type III behavior.

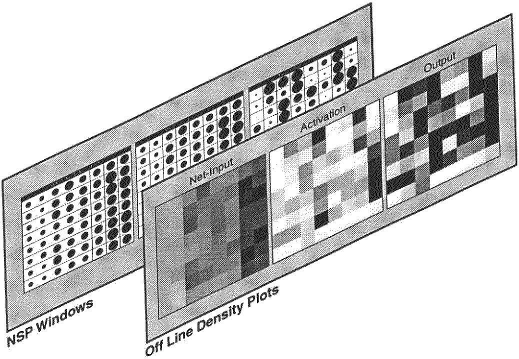


Figure 22. Neural Area Visualized with Density Plots. Here again are the same motor-cortex windows described above. We compare them with the density plots created in Mathematica.

Once design of the activatory mechanisms (type III aspect) and also learning mechanisms (type II aspect) are satisfactorily characterized, our focus turns to the varying link strengths of entire gross-links between two PBAs over time. A method of viewing the massive interconnectivity between two PBAs in an efficient manner utilizes the density plots. Consider the full connectivity of a 20x20 brain-area to another 10x10 area. This would be represent 40,000 links. If this were drawn as a set of lines connecting the two areas together with the thickness of the line representing the strength of the link, such a drawing, even in 3D, would be a tangled mess of lines. Figure 24 demonstrates a method of display which gracefully solves this problem by reorganizing the array and using the density plot.

The mapping shown is a lateral-inhibition mapping from a 20x20 brain-area to itself. It is a mesh-of-meshes which, in this case, every small mesh represents a *sink* node and the individual squares, within the small mesh, are the connectivity from the *source* area nodes. The gray level of each point represents a link strength. The

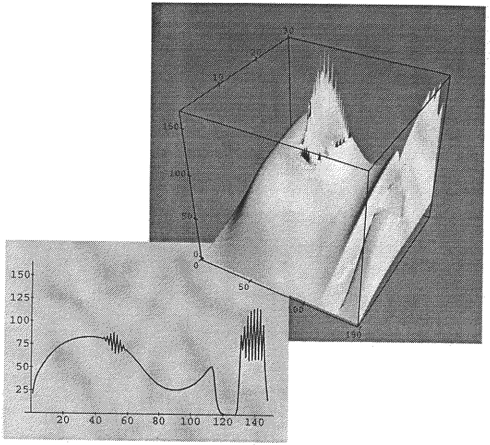


Figure 23. Single Node Characterization over a Range of Inputs. A set of nodes were simulated with a linearly varied input signal so that the activation equations used could be characterized. Both a three dimensional plot and a single slice that can be animated are shown. The oscillatory pattern demonstrates a potential instability which occurs above a certain point in the input strength.

image quite clearly shows a topological lateral-inhibition pattern which is mirrored on the edges, yet the image represents a massive set of links in a single two dimensional image. In this case the mapping is not a fully connected set of links, but even so (as we will see in section 5), a set of fully connected links, which would be 160,000 links here, can be displayed and understood. Further, animation of these maps, though memory intensive, can demonstrate the evolution of learning over time.

To accomplish the transfer between the NSP and Mathematica, methods for saving data like a massive gross-link between PBAs, was developed. Parameters that determine the cycling rules of the simulation are set to automatically save the data to a

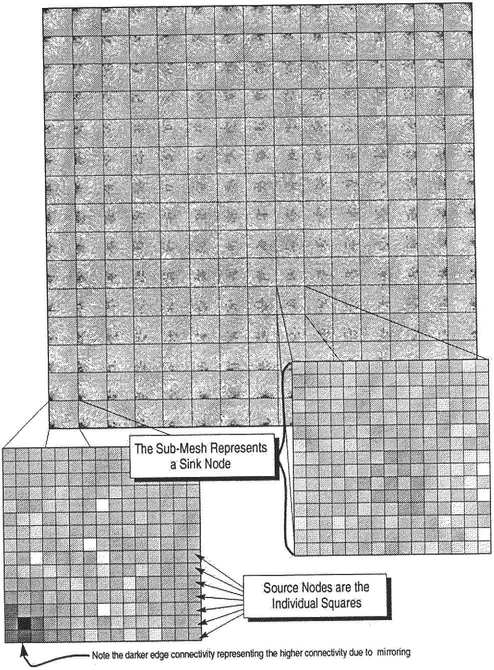


Figure 24. Representation of up to 160,000 Neural Links in a Single Two Dimensional Image. This image demonstrates the powerful visualization that can be performed using a reorganized connectivity mapping along with a density plot. The mesh-of-meshes represents the *sink* area while the individual smaller meshes represent the *source* area and each small square within the mesh represents the link weight from the source node to the sink node. This map clearly shows a topological lateral-inhibition pattern where a dark square is an excitatory link and a lighter square is inhibitory.

file or set of files. This way, a simulation can be set in motion and it is not necessary to preside over the entire event. Presently, this solution is memory intensive and also time consuming.

3.3. Nodes, Subnodes, and Links

In the connectionist paradigm, the two basic building blocks of a neural network are nodes and links. A combination of a node with a set of outgoing links can be considered an abstract representation of a biological neuron where a node is the soma of a neuron, and a link is the axonal/dentritic connections between the neurons. The links are usually given a varying weight by which they "learn". By connecting a set of these together, a neural network is formed.

A typical connectionist node (the kind utilized by DESCARTES) has three stages of activity: *net-input*, *activation*, and *output* (see Figure 25 and Appendix B). The net-input generally represents some function of the activities of the nodes linked to it. The activation is the internal state of the neuron, and the output is usually constrained further in some way to represent the axonal quality of a neuron. Using this paradigm, many different types of neuronlike components can be created. For instance, an output function can be designed that would simulate a spike train, thus resembling a neuron more accurately.

Figure 25 also demonstrates the modification that the NSP uses to provide more accurate brainlike neurons. The basic modification is that the NSP considers the connectionist node to be a *subnode*. Thus an NSP node is defined as a collection of subnodes which are governed by and linked into a main subnode. While still using the same connectionist paradigm, the NSP node is a more powerful neural processing element. Consider biological neural connectivity which has different sources of projections. Each source may have a different type of connection with a neuron. The *Purkinje* cells in the cerebellum, for instance, receive signals from four different cell types (Bullock et al., 1977). Two distinct types of its connectivity are the strong inhibitory synaptic junctions from climbing fibers and the weak inhibitory junctions from granular cells. Using subnodes, we would group granular connections in a *granular* subnode which would send a weak and possibly time delayed signal to the main node, while the climbing fiber subnode would have a strong signal sent to the main node.

Subnodes allow us to represent many different types of neural paradigms. In previous work Fukushima's Neocognitron nodes can be considered to be three subn-

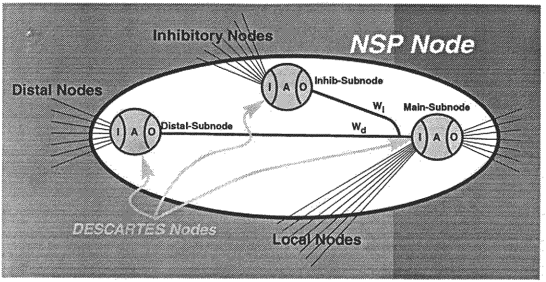


Figure 25. A Typical NSP Node. The basic DESCARTES node has a net-input (I) , activation (A), and output (O) (see Appendix B). By compounding these nodes into a single unit, a much more neurally plausible architecture can be designed. Here, subnodes perform the function of shunting inhibition and providing distal vs. local connectivity.

odes (Fukushima, 1988). A triple node combination was also used in Pearson's simulation of the surfaces of a monkey's hand (Pearson et al., 1987).

3.4. Pseudo-Brain Areas (PBAs)

Based upon the different functional areas of the brain, the PBAs are intended to parallel the types of architectures derived from the brain. Unlike the DESCARTES controller, which imposes little more than a name on the elements it governs (see Appendix B), the PBA imposes some physical form. The PBA is a node *controller* which, based upon the type of brain area, exerts a level of organization upon the set of nodes it controls. For example, a *cortical* PBA will generally be grown with some two-dimensional topological lateral-inhibition imposed upon it and will be cycled once per global cycle. Further, the border of a cortical area could be reflected or wrapped around.

There are obviously many types of PBA candidates -- such as a "globular" area which would be similar to the non-cortical brain areas, or perhaps eventually, more specific types of architectures such as a *cerebellar-area*, or *thalamic-area*. The NSP, however, has defined only one type of PBA, called the *brain-cortical* area, which stores the neurons in a two dimensional array and has the lateral-inhibition scheme of links (see Figure 24).

Figure 26 shows the textual printout of a cortical PBA from Neurobot II called "Visual1". The *display* was generated by DESCARTES primitives. Many slots* in Visual1, though, have been added by the NSP. Figure 27 elaborates upon the slots and values of Visual1 by presenting the list of classes that were mixed together to specify the capabilities of the cortex. A text output of the beginning of a simulation is provided in Appendix C.

3.5. Gross-links

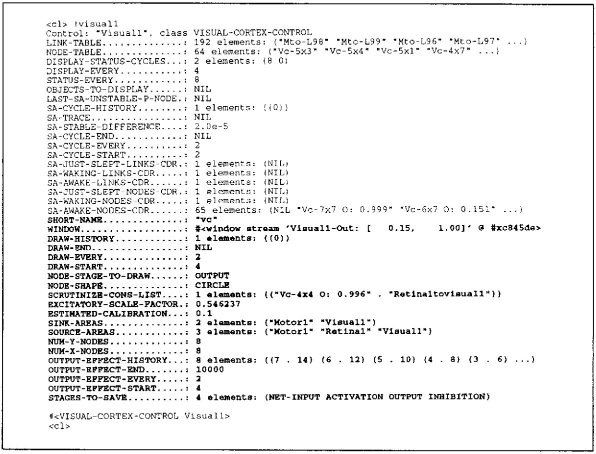
Pseudo-Brain Areas represent a higher level of neural organization. Similarly, *gross-links* are the analogous structure for links. Gross-links are controllers that keep track of a set of connecting links between PBAs. For instance, in the human body, the optic nerve connecting the retina to the lateral geniculate nucleus would be considered a gross-link while the individual neural axons would be considered normal links. Presently, gross-links are one of the most influential elements of the NSP because interconnectivity can be varied so that the same set of PBAs could solve completely different problems. Further, because of the volume of connectivity that occurs when appending a gross link, it becomes necessary to include more complex calibration to maintain some stability over the activation range of the PBAs.

We have already seen the results of using the gross-links in connecting PBAs together. Figure 24's lateral inhibition pattern was based on the difference between two 2D gaussians. The difference resulted in a "mexican hat" type function which served as a probability function to determine whether or not a link would be placed between a source node and a sink node. Further, if a link was going to be placed beyond the size of the PBA, then it was *mirrored* back to a corresponding position within the net. Figure 28 shows a graphic display of a two-dimensional lateral-inhibition probability function.

Many other types of gross-link connection types are described below in Table 2. As described in Appendix B, DESCARTES builds its neural architectures by *growth-methods*. Except for the *full-connectivity* gross-links, there is one growth-method which accomplishes the connection. In the NSP, the *cortical-interarea-control* class growth-method is the main method capable of connecting two cortical PBAs of

---

*In CLOS, an object's contents are called slots. Each slot has a name and a value.

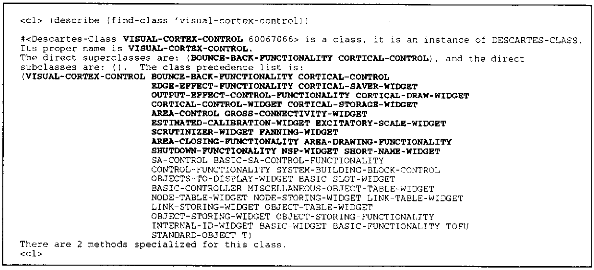


```
<cl> !visual1
Control: "Visual1", class VISUAL-CORTEX-CONTROL
LINK-TABLE..............: 192 elements: ("Mto-L98" "Mtc-L99" "Mto-L96" "Mto-L97" ...)
NODE-TABLE..............: 64 elements: ("Vc-5x3" "Vc-5x4" "Vc-5x1" "Vc-4x7" ...)
DISPLAY-STATUS-CYCLES...: 2 elements: (8 0)
DISPLAY-EVERY...........: 4
STATUS-EVERY............: 8
OBJECTS-TO-DISPLAY......: NIL
LAST-SA-UNSTABLE-P-NODE.: NIL
SA-CYCLE-HISTORY........: 1 elements: ((0))
SA-TRACE................: NIL
SA-STABLE-DIFFERENCE....: 2.0e-5
SA-CYCLE-END............: NIL
SA-CYCLE-EVERY..........: 2
SA-CYCLE-START..........: 2
SA-JUST-SLEPT-LINKS-CDR.: 1 elements: (NIL)
SA-WAKING-LINKS-CDR.....: 1 elements: (NIL)
SA-AWAKE-LINKS-CDR......: 1 elements: (NIL)
SA-JUST-SLEPT-NODES-CDR.: 1 elements: (NIL)
SA-WAKING-NODES-CDR.....: 1 elements: (NIL)
SA-AWAKE-NODES-CDR......: 65 elements: (NIL "Vc-7x7 O: 0.999" "Vc-6x7 O: 0.151" ...)
SHORT-NAME..............: "vc"
WINDOW..................: #<window stream 'Visual1-Out: [   0.15,    1.00]' @ #xc845de>
DRAW-HISTORY............: 1 elements: ((0))
DRAW-END................: NIL
DRAW-EVERY..............: 2
DRAW-START..............: 4
NODE-STAGE-TO-DRAW......: OUTPUT
NODE-SHAPE..............: CIRCLE
SCRUTINIZE-CONS-LIST....: 1 elements: (("Vc-4x4 O: 0.996" . "Retinaltovisual1"))
EXCITATORY-SCALE-FACTOR.: 0.546237
ESTIMATED-CALIBRATION...: 0.1
SINK-AREAS..............: 2 elements: ("Motor1" "Visual1")
SOURCE-AREAS............: 3 elements: ("Motor1" "Retinal" "Visual1")
NUM-Y-NODES.............: 8
NUM-X-NODES.............: 8
OUTPUT-EFFECT-HISTORY...: 8 elements: ((7 . 14) (6 . 12) (5 . 10) (4 . 8) (3 . 6) ...)
OUTPUT-EFFECT-END.......: 10000
OUTPUT-EFFECT-EVERY.....: 2
OUTPUT-EFFECT-START.....: 4
STAGES-TO-SAVE..........: 4 elements: (NET-INPUT ACTIVATION OUTPUT INHIBITION)

#<VISUAL-CORTEX-CONTROL Visual1>
<cl>
```

Figure 26. The NSP Text Display of Visual1. Every line represents a *slot* in visual-cortex-controller. The output for this method call is generated by DESCARTES. Because of the extensibility of CLOS and DESCARTES, the contributions of the NSP integrate seamlessly with the existing code. The contributions of Neurobot II and the NSP are shown in bold.

varying sizes in a topological manner. It accomplishes this with the help of a fixed set of parameters to describe the range and probability levels of the connectivity. Figure 29 shows the text display of a gross-link between the retinal PBA and the visual-cortex PBA.

Later, as we describe in Section 5, the simulation of Neurobot II clarified the need to support some higher level of control over the networks due to the diverse connectivities of a gross-link strategy. The first component added was a *fan-in* method that estimates the strength of activation that all incoming links will have on a PBA. Secondarily, a calibration factor was added, called the *estimated-calibration,* which was a user defined value to further aid in stabilizing the activations. Finally, a gross-



```
<cl> (describe (find-class 'visual-cortex-control))

#<Descartes-Class VISUAL-CORTEX-CONTROL 60067066> is a class, it is an instance of DESCARTES-CLASS.
Its proper name is VISUAL-CORTEX-CONTROL.
The direct superclasses are: (BOUNCE-BACK-FUNCTIONALITY CORTICAL-CONTROL), and the direct
subclasses are: ().  The class precedence list is:
(VISUAL-CORTEX-CONTROL BOUNCE-BACK-FUNCTIONALITY CORTICAL-CONTROL
                       EDGE-EFFECT-FUNCTIONALITY CORTICAL-SAVER-WIDGET
                       OUTPUT-EFFECT-CONTROL-FUNCTIONALITY CORTICAL-DRAW-WIDGET
                       CORTICAL-CONTROL-WIDGET CORTICAL-STORAGE-WIDGET
                       AREA-CONTROL GROSS-CONNECTIVITY-WIDGET
                       ESTIMATED-CALIBRATION-WIDGET EXCITATORY-SCALE-WIDGET
                       SCRUTINIZER-WIDGET FANNING-WIDGET
                       AREA-CLOSING-FUNCTIONALITY AREA-DRAWING-FUNCTIONALITY
                       SHUTDOWN-FUNCTIONALITY NSP-WIDGET SHORT-NAME-WIDGET
                       SA-CONTROL BASIC-SA-CONTROL-FUNCTIONALITY
                       CONTROL-FUNCTIONALITY SYSTEM-BUILDING-BLOCK-CONTROL
                       OBJECTS-TO-DISPLAY-WIDGET BASIC-SLOT-WIDGET
                       BASIC-CONTROLLER MISCELLANEOUS-OBJECT-TABLE-WIDGET
                       NODE-TABLE-WIDGET NODE-STORING-WIDGET LINK-TABLE-WIDGET
                       LINK-STORING-WIDGET OBJECT-TABLE-WIDGET
                       OBJECT-STORING-WIDGET OBJECT-STORING-FUNCTIONALITY
                       INTERNAL-ID-WIDGET BASIC-WIDGET BASIC-FUNCTIONALITY TOFU
                       STANDARD-OBJECT T)
There are 2 methods specialized for this class.
<cl>
```

Figure 27. The Class Structure of Visual-Cortex-Control. The output is a 'description' of the class that Visual1 is based upon. The class precedence list shows all the classes that were combined to produce the PBA. Again, the bold print represents the contribution of the NSP or Neurobot II.

link magnitude factor was added which gives each gross link a priority value. This allows the different gross-links to have different amounts of affect upon a PBA while still learning and activating accurately. These modifications to the NSP provided the capability to stabilize networks at the system level under many conditions.

One of the more important aspects of the NSP is the topological aspect introduced via the PBAs and the gross-links. Biologically, full-connectivity does not exist the brain and consequently is not employed in brain modeling. Our limited types of inter-area connectivities set model connectivities exhibited in biological cortical interconnectivity. There can be no *global-optimization* of functionality without full-connectivity because anything less would provide a subset of all possible combinations. However the brain must make a *constrained optimization*: physical compactness, energy conservation, and processing delays all argue for minimizing non-local connections. The NSP provides an environment for exploring this avenue of constrained neural networks above and beyond that provided by most simulation environments. Further types of topological connectivities are possible, along with different types of PBAs such as globular areas. We discuss these in our conclusion.

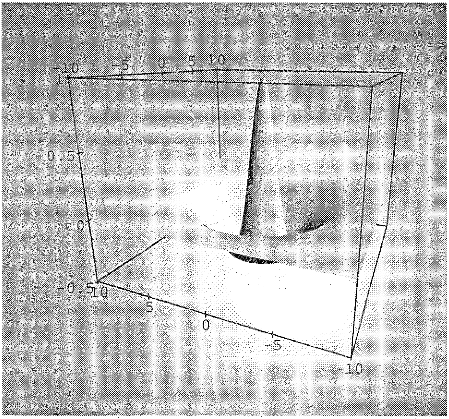


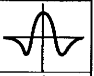
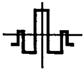
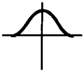
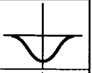
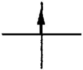
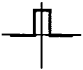
Figure 28. A 2D Mexican Hat Function. The height of the hat represents the probability of growing a link. If the probability value is negative, then it would generate an inhibitory link.

3.6. External-World and Cycling Hooks

The NSP itself has no external world per se. Rather the NSP provides the objects to integrate models of an external world into itself. This is accomplished by mimicking the cycling mechanism that DESCARTES utilizes for its spreading activation capabilities. In the NSP, we have added a number of cycling controllers, some of which can be seen above in Figures 26 and 29, the *output-effect*, *draw-cycle*, and *learn-cycle* classes are capable of being added to any object. Another cycling hook is the *external-input* class. Mixing these classes into the class definitions of PBAs or gross-links, or models of the external world, gives these objects one of these cycling capabilities. They can then be integrated with the rest of the system and coordinated.

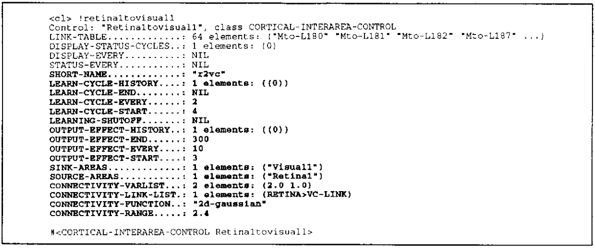
Table 2.

Different Types of Interconnectivity Provided by the NSP.

Connectivity Type	Description	Graph
Lateral-Inhibition	Difference of two 2d-gaussian functions. The peak probability and radii of the gaussians are given.	
Lateral-Inhibition-2	A list of radii are given with a probability for the range between radii. A negative number implies an inhibitory link.	
2d-Gaussian	A simple gaussian function. The height of the gaussian determines the probability of a link. The maximum probability is given.	
2d-Inhib-Gaussian	Same as 2d-gaussian except for inhibitory links.	
One-to-One	For areas of the same size, this maps a single link to the corresponding neuron of the other area. Can use segregated mapping too.	
Full-Connectivity	Connects the entire source PBA to the sink PBA. Uses a different growth method for doing this to avoid the bounce-back.	
Segmented	Segregates a set of links so that a topology is completely preserved. Used to generate specific signals, or to focus topographic maps.	

3.7. Type I Capabilities

The above components of the NSP are designed to provide the user with a set of tools for creating neurobotics simulations of basically type II and type III systems. Because of the versatility of neural areas and neural connectivities, though, we can continually append further PBAs and gross-links. Thus the NSP also allows evolutionary (type I) contribution of layered development in the same vein as Brooks' de-



```
<cl> !retinaltovisual1
Control: "Retinaltovisual1", class CORTICAL-INTERAREA-CONTROL
LINK-TABLE.............: 64 elements: ("Mto-L180" "Mto-L181" "Mto-L182" "Mto-L187" ...)
DISPLAY-STATUS-CYCLES..: 1 elements: (0)
DISPLAY-EVERY..........: NIL
STATUS-EVERY...........: NIL
SHORT-NAME.............: "r2vc"
LEARN-CYCLE-HISTORY....: 1 elements: ((0))
LEARN-CYCLE-END........: NIL
LEARN-CYCLE-EVERY......: 2
LEARN-CYCLE-START......: 4
LEARNING-SHUTOFF.......: NIL
OUTPUT-EFFECT-HISTORY..: 1 elements: ((0))
OUTPUT-EFFECT-END......: 300
OUTPUT-EFFECT-EVERY....: 10
OUTPUT-EFFECT-START....: 3
SINK-AREAS.............: 1 elements: ("Visual1")
SOURCE-AREAS...........: 1 elements: ("Retinal")
CONNECTIVITY-VARLIST...: 2 elements: (2.0 1.0)
CONNECTIVITY-LINK-LIST.: 1 elements: (RETINA>VC-LINK)
CONNECTIVITY-FUNCTION..: "2d-gaussian"
CONNECTIVITY-RANGE.....: 2.4

#<CORTICAL-INTERAREA-CONTROL Retinaltovisual1>
```

Figure 29. Gross-link Text Display. The DESCARTES component of the gross-link is very small because this is the first real step above it. Thus there are many new slots added here by the NSP. The four connectivity slots at the bottom define the type of link being created. This set of parameters designates a 2d-gaussian with a radius link scan of 2.4 from the centerpoint of the receptive field and a gaussian radius of 2.0 with the probability of a match being 1.0 at centerpoint.

sign: the ability to augment previous designs without redesign. Presently, an actual framework for *growing* evolutionary layers has not been implemented because there were too many other open issues to tackle. The freedom to develop any type of evolutionary layering is thus left solely up to the user, just as the architectural organization of the PBAs is left also to the user.

The task of developing an evolutionary layering, though, is simplified greatly by CLOS. Inheritance and especially multiple inheritance gives CLOS a step beyond most programming languages because this type of program extensibility is somewhat analogous to the real world extensibility of evolution. Objects can be designed to represent a certain concept or attribute of a design. These *mixins* can be modularized so they can be combined and customized. In the same way that evolutionary concepts spread throughout a being, so can the mixins, each time being further augmented to fit the circumstances (see Appendix B). An example is the redundancy of the brain. At one level, we see consistent connectivities throughout a brain area. In other words, a redundant replication of connections. At another level, we see brain areas over time splitting into two identical areas and eventually providing different functionalities. Using CLOS, a set of evolutionary conceptual mixins can be created which would serve as the building blocks for the evolutionary growth of neurobotics systems (See Appendix A, Meta-Brain Aspects, a suggested direction for designing these building blocks.).

In summary, the NSP has been built to provide a multitiered environment for developing neurally inspired brain-like architectures that have the ability to learn in a closed loop causal relationship with external world models. The package provides constructs such as pseudo-brain areas and gross-links capable of creating immense multidimensional topologically oriented architectures, cycling and synchronization methods, and visualization tools, which combine into an environment for testing both large neurobotics systems, or small scale theoretical concepts.

Below, Neurobot II (Section 4) represents a prototypical neurobotics system developed using the NSP, while our experimental results (Section 5) have been derived from the NSP using a small fully connected network that focuses on the diverse set of mappings and the distributed nature of a cortical area.

4. NEUROBOT II

We have simulated an arm/eye sensorimotor system, similar to those discussed above, which utilizes the many faculties of the NSP. Our ultimate goal is to build a closed-loop reentrant sensorimotor system that associates and predicts arm positions. We call our system Neurobot II. It involves a number of different PBAs and reentrant gross-links and also a closed loop interaction with its world. Figure 30 shows a conceptual view of the elements and interactions of the Neurobot II system. The NSP provides us the capabilities for building systems of this size and also much larger ones.

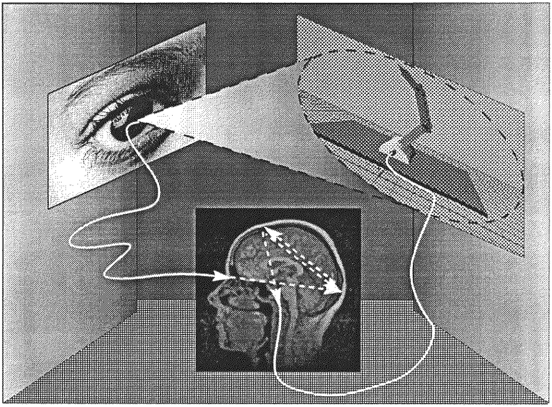


Figure 30. Basic Neurobot II System. The basic components are a (1) three degree-of-freedom robot arm, (2) a low resolution eye which can see the robot arm, and (3) the brain.

We break up our description of Neurobot II into two basic parts, the internal and the external world, where we consider the eye and the arm part of the external world and thus leave the PBAs and their gross-links as the sole components of the internal world. Neurobot II is designed using the tools of the NSP, DESCARTES, and CLOS.

Figure 31 shows a more detailed schematic of the relationships between the various controllers in Neurobot II. It also presents the addition of an external obstacle to address one of the long term goals of the Neurobot II system: to distinguish the self from the non-self. Unlike the closed loop of the arm/eye connections, the actions of the external object are not causal. We suggest that this nonembodiment, if captured adequately in a set of PBAs, would represent an ability to discern self from non-self.

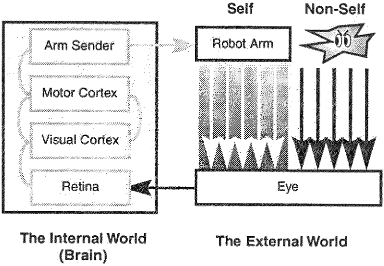


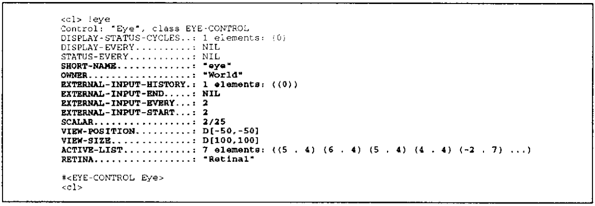
Figure 31. Schematic View of Neurobot II. Here, a component of a proposed future Neurobot II system is added, the external objects. Note the difference between components of the robot proper, linked by a closed loop which generates self-consistency, and those of the external world objects. The closed loop allows differentiation between self and non-self. Clearly, if the self were not part of the loop, we would be no more intelligent than machines.

4.1. External Elements

The external elements of the Neurobot II system consist of the *arm*, the *eye*, and the *world*. The three components are intimately connected to the Neurobot II brain through the cycling protocols provided by the NSP and the data conversion methods which the components themselves contain. For instance, the eye must convert the external world that it is viewing into the lower resolution array for the retina.

4.1.1. *The Eye*

Neurobot II's visual processor is a simple one: a singular, rectangular, two dimensional field of vision. Figure 32 is a display of the contents of Neurobot II's eye. The eye has the capability to look at different areas of the world: It can also scale and change its aspect ratio. Its main purpose, though, is to *see* the robot arm discussed below, and any objects in its field of view, and perform a scan conversion onto any size retina which is connected to the eye. Figure 33 shows an example of the eye's resulting retinal image based on its world view.



```
<cl> !eye
Control: "Eye", class EYE-CONTROL
DISPLAY-STATUS-CYCLES..: 1 elements: (0)
DISPLAY-EVERY..........: NIL
STATUS-EVERY...........: NIL
SHORT-NAME.............: "eye"
OWNER..................: "World"
EXTERNAL-INPUT-HISTORY.: 1 elements: ((0))
EXTERNAL-INPUT-END.....: NIL
EXTERNAL-INPUT-EVERY...: 2
EXTERNAL-INPUT-START...: 2
SCALAR.................: 2/25
VIEW-POSITION..........: D[-50,-50]
VIEW-SIZE..............: D[100,100]
ACTIVE-LIST............: 7 elements: ((5 . 4) (6 . 4) (5 . 4) (4 . 4) (-2 . 7) ...)
RETINA.................: "Retinal"

#<EYE-CONTROL Eye>
<cl>
```

Figure 32. Neurobot II's Eye. This display shows the contents of the interesting slots of this class. Notice the four *external-input* slots which define the cycling of the eye. The slot *external-input-every* designates that on every second cycle, a method will be called. In this case, the method converts the world image into the retinal scan-line array designated by the *retina* slot.

4.1.2. *The Robot Arm*

Similar to Kuperstein's muscular controllers, Neurobot II's architecture contains an Arm Sender that generates a pair of neural signals to define an equilibrium point for each joint. A difference in activation between a muscle pair in the Arm Sender causes the robot arm's joint to move with a certain angular velocity (see Figure 34). Further, the joints of the robot arm are not free to move anywhere in the image space, but are constrained. For instance, each joint can be limited in its movement. Once a joint reaches its fullest extent, all further velocity signals in that direction imposed by the arm sender are ignored by the arm.

The arm's class display is shown in Figure 35. Here we make use of the *output-effect* cycling tool. Further, the arm is build upon a kinematic/dynamic model of motion. The elements of this model are a set of mixin classes which, when com-

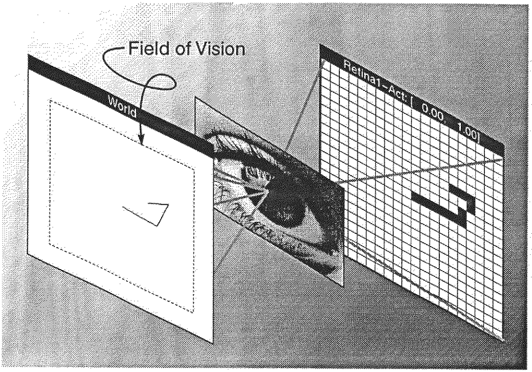


Figure 33. Retinal Imaging. Neurobot II's eye is shown here with the world image a three degree-of-freedom robot arm impinged upon it. This raw signal is then sent back to less distal areas of Neurobot II's brain. The dashed line surrounding the robot arm in the world window represents the range of vision that the eye is looking at.

bined, form a dampened, limited-movement, position guided, external world object. In this case, each *arm-link* has a rotational dynamics class mixed in with it's object definition. Figure 35 also shows the class display of the first arm-link of the robot arm.

4.1.3. The Neurobot II World

Tying the entire Neurobot II world together is the *world* controller which serves as a main growth and cycling coordinator and keeps a placeholder for all the various elements of any NSP simulation. Figure 36 shows the slots representing the arm, eye, external world, and brain. The world controller also provides the customized world window, as shown in Figure 33, which combines a world view of, not only the robot arm, but any external objects, and also the field of vision of the eye.

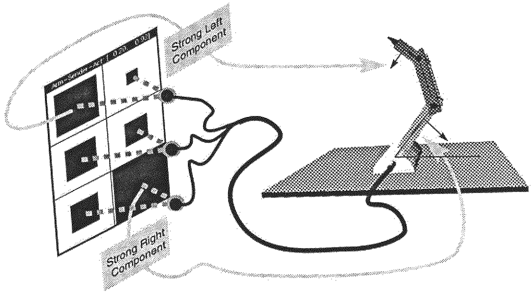


Figure 34. Agonist-Antagonist Neuromuscular Controllers. The signals sent to the robot arm from the Arm Sender are velocity based as opposed to position based. The difference in activity of the joint's agonist-antagonist pairs gives the robot arm a certain velocity. Whether the robot arm actually takes on this velocity is constrained by its movement limits and whether it may collide with itself or other objects.

4.2. Neurobot II's Internal Components

Of course, the most interesting element of the Neurobot II system is the Neurobot brain. Activation equations, learning equations, and the multi-PBA architecture with its interconnectivity of gross-links form the internal mechanisms of Neurobot II. The PBAs are organized into four different groups: afferent, efferent, automatic, and associational areas. Our single afferent PBA is the retina; the efferent area is the arm-sender, a motor-oscillator was added later which imposes an automatic cyclic pattern to the motor-cortex, and the associational areas are the visual-cortex and the motor-cortex (see Figure 31). Below, we discuss the activation and learning equations, the PBAs, and their interconnectivity.*

---

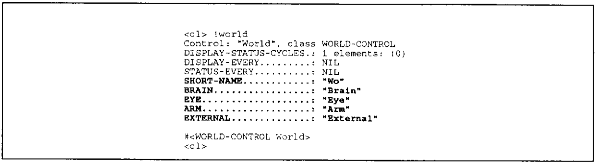
*In discussing the PBAs of Neurobot II we use brain terms such as retina, visual-cortex, and motor-cortex loosely for the sake of easing terminology confusion. There is no serious correlation with the mammalian brain here.

```
<cl> !arm
Control: "Arm", class ARM-CONTROL
DISPLAY-STATUS-CYCLES.: 1 elements: (0)
DISPLAY-EVERY.........: NIL
STATUS-EVERY..........: NIL
SHORT-NAME............: "arm"
OWNER.................: "World"
OUTPUT-EFFECT-HISTORY.: 1 elements: ((0))
OUTPUT-EFFECT-END.....: NIL
OUTPUT-EFFECT-EVERY...: 2
OUTPUT-EFFECT-START...: 4
SENDER................: "Arm-Sender"
ARM-LINKS.............: #<Arm Link Collection>
Movable: D[0,0] 'Link1'
(---Collection---------------------------
(  Line: D[-0.91500485,-1.0783172] to D[24.001534,0.9631321]
(  Movable:   D[24.916523,2.041459] 'Link2'
  (---Collection---------------------------
  (    Line: D[0.0,0.0] to D[-18.860935,-6.652932]
  (______________________________________

(______________________________________
NUM-MUSCLES...........: 2

#<ARM-CONTROL Arm>

<cl> !(get-arm-link %arm)
Object:  TOFU, Class: ARM-LINK
NAME......................: "Link1"
Y.........................: 0
X.........................: 0
PIECES....................: 2 elements: ("L[#D<-0.92,-1.08> to #D<24.00, 0.96>]"...)
OWNER.....................: "Arm"
TARGET-ANGULAR-VELOCITY..: 0.07944824
ACTUAL-ANGULAR-VELOCITY..: 0.06973513
ANGULAR-VELOCITY-DECAY...: 0.7
ANGULAR-VELOCITY-CHANGE..: 0.3
ANGULAR-CEILING-VELOCITY.: 1.0
ANGULAR-FLOOR-VELOCITY...: -1.0
ANGULAR-CEILING-POSITION.: 10.0
ANGULAR-FLOOR-POSITION...: -10.0
ARM-LINK-NUMBER..........: 1

#<ARM-LINK 60403146>
```

Figure 35. The Neurobot II Arm and Arm-Link. While there are many aspects packed into the workings of the arm, by looking at the Arm's class slots, one can only discern that it is hooked into the system cycling with the *output-effect* slots. Here, output-effect implies that on a given cycle, a connection from the *arm-sender* (a PBA) to the arm itself causes *arm-links* to alter their position. On closer inspection of the arm-links display, the rotational/translational dynamic components added to each link can be seen.

4.2.1. Neural Activation Equations

By activation equations, we mean the functions that affect the values of the net-input, activation, and output of the all the elements in the system. To ease the complication, a single rule was used to guide the entire system of nodes. The activation rules were based upon the work by Reeke et al., with reductions and modifications (Reeke et al., 1989). Whereas Reeke's original rules dealt with long-term potentiation, we have chosen not to utilize that aspect here for efficiency and simplicity. The depression based hyperpolarization of neurons, though, was maintained in a simple form. Figure 37 presents the description of the equations. The characterization of its operation is discussed in the next section.



```
<cl> !world
Control: "World", class WORLD-CONTROL
DISPLAY-STATUS-CYCLES.: 1 elements: (0)
DISPLAY-EVERY.........: NIL
STATUS-EVERY..........: NIL
SHORT-NAME............: "Wo"
BRAIN.................: "Brain"
EYE...................: "Eye"
ARM...................: "Arm"
EXTERNAL..............: "External"

#<WORLD-CONTROL World>
<cl>
```

Figure 36. The Neurobot II World. The slots of the world controller represent all the main components of an NSP simulation at the highest level. Here this implies the above listed components of the arm and eye, a future component, the external world objects, and, of course, the Neurobot II brain. The *draw-cycle* slots allow the world window to update its image as directed by its slot values. In this case, it started drawing the window on the second cycle and continued to update every two cycles.

4.2.2. *Neural Learning Equations*

The learning equations alter the weights of the links connecting the nodes. While it is possible to provide our simulations with the capability of neural plasticity (actual neural death and growth), we have chosen to consider a stable architecture for the moment. Thus *learning* means that the link weights are changed over time. The learning algorithm is also adapted from Darwin III as described in Figure 38.

4.2.3. *Pseudo-Brain Areas*

The equations above affect the PBAs and the gross-links during *cycling.* The PBAs all have the inherent capability of *sa-cycling* (*spreading-activation-cycling*), while the gross-links can have *learn-cycling.* Sa-cycling is the cycling mixin utilized by DESCARTES. It works similarly to the cycling hooks described in the previous section. It performs the functional calculations of the nodes and links within all controllers containing them. Learn-cycling is a user design issue. Some gross-links were given learning while others weren't. Usually, the retina to visual-cortex links were learning links while the other gross-links did not learn. We will discuss this in the next section. Below, the PBAs which make up Neurobot II are described by their type (afferent, efferent, automatic, or associational).

4.2.3.1. Retina

The retina receives its signals from Neurobot II's eye. The image seen by the eye becomes the sole input to the neural component. A topological gross-link connects the

Neural Activation Equation

$$s_i(t) = [A\phi_1(I_S)]\phi_1(D) + W$$

where,

$s_i(t)$ = state of cell i at time t

A = total input from specific connections = $\sum_k w_k \sum_j c_{ij}(s_{1_{ij}} - \theta_{E_k})$

w_k = scale factor that determines overall strength of connections of type k, based upon the fan-in to the PBA and also the user set calibration.

c_{ij} = strength of connection from input j to cell i

1_{ij} = index number of cell connected to input j of cell i

θ_{E_k} = excitation threshold ($s_{1_{ij}} < \theta_{E_k}$ ignored)

k = index over connection types

j = index over individual connections,

I_S = total shunting inhibition, the sum of all inhibitory inputs.

D = depression = $\upsilon_D s_i(t-1) + \omega_D D(t-1)$

υ_D = growth coefficient for depression

ω_D = decay coefficient for depression.

When $D > \theta_D$, where θ_D is a refractory threshold, then $\phi(D)$ is set to θ_D for a specified number of cycles, after which D is set to 0 and $\phi(D)$ returns to 1.0

W = decay term = $\omega s_i(t-1)$

$\phi_1(x)$ = sigmoid = $\dfrac{1}{1+\exp(\frac{\theta-x}{\varepsilon})}$, the value of ε determines the direction of the sigmoid. for example, the inhibition sigmoid ε is -.2 whereas the activation ε is .1.

Figure 37. The Neural Activation Equations. While similar to those used in Darwin III, the Neurobot II activation equations ignore the long term potentiation attribute for the sake of computational speed (Reeke et al., 1989).

retina directly to the *visual-cortex*. A typical connectivity scheme between the retina and the visual-cortex is shown in Figure 39.

4.2.3.2. Arm-Sender

In Figure 34 we see that the arm-sender's nodes correspond to the agonist-antagonist muscle strengths. The arm-sender represents Neurobot II's sole output to the exter-

Link Modification Equation

$$c_{ij}(t+1) = c_{ij}(t) + \delta\, \phi_2(c_{ij})\, (s^*_i - \theta_I)\, (m_{ij} - \theta_J)\, R$$

where,

δ = amplification factor, which adjusts the overall rate of synaptic change

s^*_i = time-averaged activity of cell i = $\lambda s_i(t) + (1-\lambda)s^*_i(t-1)$, where λ is a damping constant.

θ_I = amplification threshold relating to postsynaptic activity.

m_{ij} = average concentration of a hypothetical postsynaptic "modifying substance" produced at cell i by cell j. Rather than the complex equation, we merely use s_j as suggested by Reeke.

θ_J = amplification threshold relating to presynaptic activity.

R = rule selector. While the paradigm allows for a multitude of different rules, Neurobot II only uses one rule. If the presynaptic and postsynaptic values are strong (above threshold), then set R = 1. If presynaptic is strong and postsynaptic is weak (subthreshold), then set R = -1. If presynaptic is weak (subthreshold) then set R = 0. This simulates a pseudo Hebbian rule with the added feature that it also decreases link strength.

$\phi_2(x)$ = radial basis clamping function = it keeps the link weight in the range of 0 to 1 by shifting x into the -1 to 1 domain (x^*) and appropriately clamping it with the equation $1 - 2(x^*)^2 + (x^*)^4$.

Figure 38. Neurobot II's Link Learning Equations. Rather than operate in a range of -1 to 1 as in Darwin III, here the operating range is from 0 to 1. While the equation is very versatile based upon the rule selector *R*, here we have designated the rule as a pseudo-Hebbian rule (Hebb, 1949; Reeke, 1989).

nal world. It receives its influence from the motor-cortex with a *segregated* gross-link (see Table 2). The gross-link separates the motor-cortex into sections that are mapped to only one of the arm-sender nodes, thus, the motor-cortex represents a topographic mapping of the "body" of Neurobot II.

4.2.3.3. Motor-Oscillator

Another segregated gross-link is utilized by the motor-oscillator, an automatic PBA which generates a cyclic and realistic swaying for the robot arm. There were many different places in the system where such an addition could have been made, for instance, directly to the arm-sender. Here, though, the motor-oscillator nodes (one for

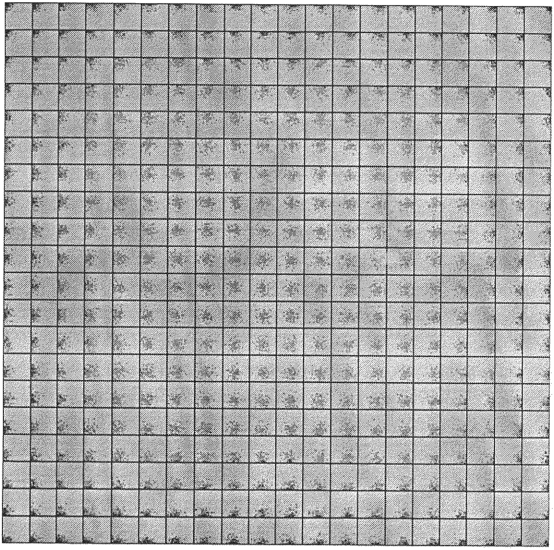


Figure 39. Retina to Visual Cortex Connectivity Scheme. The gross-link density plot shows a typical connectivity scheme between the two PBAs. It is a topological mapping of a narrow gaussian spread. The above connectivity represents a 20x20 retina mapping into a 20x20 visual-cortex.

each muscle) were expanded to cover an exclusive portion of the motor-cortex, the same nodes that then segregate back down to the arm-sender.

4.2.3.4. Visual-Cortex and Motor-Cortex

Finally, we reach the most abstract components of Neurobot II. The visual-cortex specifically also stands as the main focus of our next section which deals with characterizing cortical organization. Here, though, both associational PBAs are very similar. They have both input and output gross-links and also a self-connecting gross-link with lateral-inhibition (see Figure 32).

The motor-cortex receives a gross-link input from the visual-cortex. Its output is topographically mapped to the arm-sender. Our intention behind Neurobot II's motor-cortex is to associate the visual-cortex patterns with some motor output.

The most complex element of the Neurobot II system is the visual-cortex, which receives its inputs from both the retina and reentrantly from the motor-cortex. It outputs back to the motor cortex. The visual-cortex represents the most abstract component because, depending upon which sets of gross-links are given the capability of learning, potentially complex associational patterns, similar to those described in the Dalhem Konferenzen and by Obermeyer, or Pearson, could eventually emerge, given the proper architecture (Stryker, 1989; Obermeyer et al., 1990; Pearson, 1987).

4.3. Summary of Neurobot II Design

By designing appropriate PBAs and gross-links, building the internal components of Neurobot II is a simple task. The external world, on the other hand, requires careful modeling of the desired aspects, which in this case consist of physical models of a robot arm with dynamic and kinematic constraints and the ability to transform its image to the retinal PBA. The results which arose from this system are discussed in the next section.

In summary Neurobot II demonstrates the implementation of a brainlike sensorimotor system using the NSP. It comprises about 1700 lines of NSP/DESCARTES/CLOS code. Neurobot II's actual architecture is only a small representative of what can be designed in the NSP environment. The immense task of processing even a small number of simulated brain areas on the available hardware (a Sun 4/330) left much freedom for software generalization but not so much for simulation. In the future, the increase in speed provided by parallel computation will allow a many fold increase in size of feasible Neurobotics type simulations (Lange, 1990).

5. EXPERIMENTS AND RESULTS

While at first glance, the correlation between Neurobot I and Neurobot II may seem rather minimal, it is actually a smooth step in the same direction. In designing Neurobot II we banked upon the same principles that allowed Neurobot I to learn its mapping. Here, though, we extended the mappings into a more complex domain and range by generating networks that are more natural than the uniform Kohonen network in Neurobot I (Kohonen, 1988). Our results demonstrate a self-organizational capability in our networks which is more general than the winner-take-all system.

In the following sections, we discuss a set of experiments beginning with neural characterization tests, continuing with some of the "interesting" patterns that emerged as we reduce the size of our system, and finally the onset of generalized self-organization.

5.1. System Tests

We have seen two basic levels of equations which guide the entire Neurobot II system. While it is possible to encode any type of activation and learning into the nodes and links throughout the system, here, single equations were chosen for the activation and learning (see Figures, 37 and 38). This proves to be a worthwhile decision because while the models of neuronal activity and learning are important, they are not the focus of this research. The essential factor was that the models of neuronal activity and learning simply be powerful enough to accommodate a wide range of architectures. Fortunately, using models similar to Darwin III's has proven its versatility (Reeke et al., 1989). The models were efficient and powerful enough to provide our final results.

5.1.1. Activation Characterization

We characterize activation equations, to determine the ranges the neurons operate within. Two different simulation tests are performed. First, we designed the activation equations on an independent simulation package to test various strategies. Once we came up with useful neural activation equations, we encoded them into the system equations of Neurobot II. At this point a comparison was done to demonstrate that not only the neural activity was the same, but also that the simulation environment itself was behaving appropriately. Fortunately, as Figure 40 shows, this was the case.

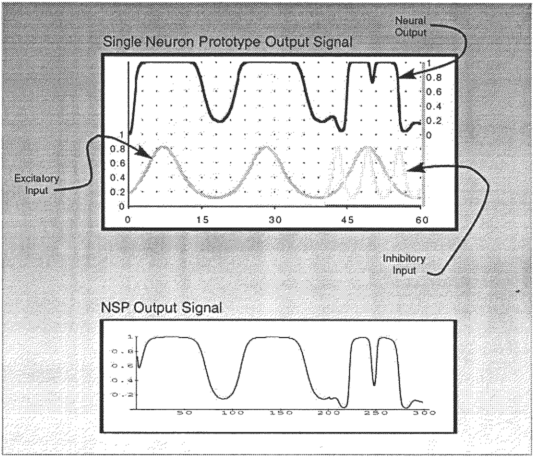


Figure 40. Comparison Between a Stand-Alone Single Neuron Simulation and its Replication within the NSP in Neurobot II. The two graphs demonstrate that the NSP is simulating our neural activation equations appropriately. The top graph was taken from an independent simulation package. The lower graphical output from the NSP obviously matches.

5.1.2. *Learning Characterization*

Characterization of the learning algorithm followed the neural activity characterization. While the learning and activation equations are very similar to Reeke's, there are numerous ways in which the variables of the equations can be altered to provide completely different results. Secondly, the architecture defined by the gross links also has a large effect upon the type of activation and learning that occurs.

Now that the types of activation and learning have been discussed and characterized, we look at the types of output that occurred from simulating elements of Neurobot II.

5.2. Summary of Neurobot II System Results

The entire Neurobot II system proved to be an encumbrance entity to manage, even at the type II level. In other words, when the entire system was connected and simulated, the networks tended towards some form of instability. While the instabilities of the networks lead to implementing more effective means of calibrating the nodes, links, and PBAs, eventually it was necessary to reduce the size of the system, so that a concrete example could be demonstrated. In the process, though, some interesting aspects occurred which, if completely characterized, may prove to be important contributions to our understanding.

In general, the following examples were based upon simulations of Neurobot II's components. Because of their generally unstable nature, they could only be considered as type II systems. The connectivity, activation and learning parameters varied throughout. Every case was unique. The robot arm was usually "self-guided" which, although random, was useful enough to study the effects of the movement on the pseudo visual-cortex.

5.2.1. Temporal Tracing

One of the earlier outputs produced a pattern that when animated was recognizably interesting. We define it as *temporal tracing* because an impression of past activations fades over time thus allowing the past to merge with the present. Figure 41 shows four sequential frames of the activation of the Neurobot II visual-cortex. As the robot arm rotated counter-clockwise, the "image" was impressed on the cortex for a period after the arm had moved on. The potential importance of the temporal aspect is based upon the limitations of many neural network models which ignore feedback. This result shows that under appropriate learning conditions, a temporal association can be formed.

5.2.2. Activity Bubbles

Another example emerged from Neurobot II's network when a lateral-inhibition scheme was combined with a gross-link from the retina to the visual cortex with a moderately sized two-dimensional topological spread. A set of moving activity bubbles formed. In other words, while the visual-cortex images did not resemble the retinal image, the combination of the retinal projection with the lateral inhibition created an evenly spaced set of activation patches, that over time had some slight mobility. Figure 2 shows a three dimensional plot of the output stage of the visual-

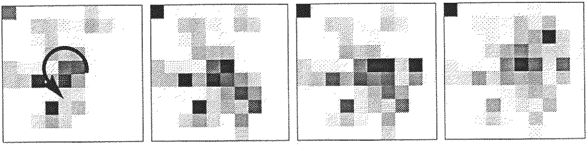


Figure 41. Temporal Tracing of a Visual Image. The set of frames above were taken from the Neurobot II visual-cortex. The robot arm, while moving counter-clockwise left an impression on the visual-cortex that faded slowly with time. It can be seen clearly in an animated presentation. This type of activation, induced by feedback, hints at the possibility for a temporally association which would eventually allow some form of predictive capability.

cortex. There is a striking similarity between Kohonen's activity bubble paradigm and the bubbles appearing in this network. The difference being the singularity of Kohonen's bubble versus the multiple evenly spaced bubbles occurring here. This lead to further results described below.

5.2.3. Increased Calibration Necessary

To this point, our simulation results have only been based upon type III systems. The activations alone gave the resultant patterns; and no learning occurred on the links connecting the structures. The NSP evolved over time as Neurobot II simulations acquired various levels of success so that learning could also be integrated into the system. Based upon the continual results provided by Neurobot II simulations, controlling factors were added so that the simulation would maintain stability over the long term at both the type III and type II levels.

The largest problem in simulating the full Neurobot II system was over- and under-activity of the nodes. In other words, it was difficult with our activation and learning equations and the diverse gross-connectivity to determine the correct multiplicative values for gross-links so that the neural activations would not only remain in the proper ranges but also have an appropriate level of effect. For instance, in many simulations, while the visual-cortex was receiving signals from both the retina and the motor-cortex, the feedback from the motor-cortex would override the visual input of the retina. With learning, eventually, the retinal effect became insignificant. Thus it became necessary to modify our systems further at the NSP level.

5.3. Self-Organizational Evidence

Once we were able to break into the type II simulation, we could begin to explore the organizational patterns which must somehow occur at each PBA to provide meaning to our sensorimotor systems. We found that a neurally connected map is capable of representing an input image in a distributed fashion, as opposed to both the traditional one-position/one-node recognition schemes (Fukushima, 1989; Kohonen, 1988; Carpenter & Grossberg, 1987) and also the holographic type encoding (Pribram, 1971). The activity bubbles that emerged from our type III simulations first prompted our further focus on Kohonen's work. Here, we summarize briefly the previous work.

5.3.1. Kohonen's Abstraction

In studying a single layer, Kohonen observed that the lateral inhibition caused by interneurons in a layers of cortex material forced the emergence of an activity bubble in which a cluster of neighboring nodes would become active while an inhibitory ring was imposed around the cluster. He hypothesized that learning takes place where activity is high and thus created an abstraction based upon the winning node of a map of neurons. The node with the strongest activity forces a *neighborhood* around the winning node. Using this abstraction, he was able to generate self-organized, topological, dimensionally-reduced representations of the input space to occur such as the phoneme map shown in Figure 3 (Kohonen, 1988)

Kohonen's algorithm was limited in the sense that it allowed only one winner. In other words, only one activity bubble was chosen. Referring back to Figure 1, we notice that it is reminiscent of Figure 2. In the left half of Figure 1, we see a single activity bubble. This represents the Kohonen neighborhood. The results of relaxing this constraint (reverting to neurally connected lateral inhibition) are shown in the right half of Figure 1. The interesting aspect of the original network as opposed to the abstracted algorithm is that the one winner constraint is not placed upon the system. The emergence of multiple activity bubbles as seen in Figure 2 concur with this aspect.

Multiple activity bubbles imply *multiple traces* of the same input pattern, or perhaps a piecewise breakdown of it. A redundant pair allows mappings which are not required to be one-to-one and can be propagated to other layers, perhaps distributing and disecting the original signal further.

Consider Neurobot I's degenerate conditions where the same end effector position could be found with two sets of joint positions (Figure 17). With Kohonen's abstraction in place, not only does it prevent networks from easily representing degenerate conditions, but it also imposes a strict topology on the network. And as discussed, a neurological mapping, such as the visual cortex, is multifaceted thus, not necessarily strictly topological (Stryker, 1989).

Finally, as we have pointed out, a gross-link should not be fully-connected. A fully-connected set of links connecting one area to another is clearly non-biological, where low connectivity is a space saving benefit. Certainly, the topological aspects found in areas of the brain such as the visual-cortex or the somatosensory cortex are mostly based upon architecture (genetics) rather than environment This leaves room for another sort of organization to take place at a smaller level which is perhaps where the self-organization belongs.

Kohonen's mapping limits a network to being a feature detection map as opposed to possibly a distributed recognition environment. (Saxon & Mukerjee, 1990). This degeneracy proves to be a valuable neural construct to further empower a neural sheet with diverse capabilities (Edelman & Mountcastle, 1978; Obermeyer et al., 1990; Stryker, 1989).

5.3.2. Distributed Self-Organization

To focus on our new goal, a smaller NSP system was developed, and nicknamed *self-org*. Its purpose is to look specifically into the principles of self organization without the Kohonen abstraction, and perhaps, further, without full-connectivity.

The same robot arm was used as input. In this case, an set of 16 8x8 retinal images was stored and cycled through. Each containing a very low resolution image of the robot arm in a different position. These are shown in Figure 42. The retina was then connected to the visual-cortex in a fully-connected random weighted set of links. Later, we return to topological connectivity. The visual cortex itself was the most variable element because many different lateral inhibition schemes were attempted.

Out of the many simulations run. Two types of non-discretionary results were the: single non-moving activity bubbles that could not discriminate between any of the images. And non-bubble type networks, that would scatter different levels of non-discriminating nodes throughout a network. Examples of these nets are shown in Figure 43.

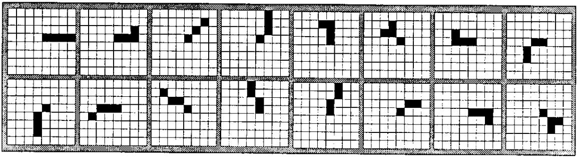


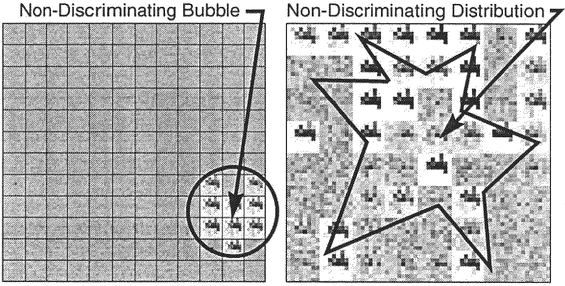
Figure 42. Retinal Image of the Sixteen Robot-Arm Positions. These positions were used to train the self-organizational networks.

Certain networks had an interesting spread of activity over time. First the learning would occur at a certain node and then slowly spread out over the network. This was characteristic of the way that Kohonen's networks performed. While these networks still proved to be non-discriminating, the smooth spread of learning is a very valuable quality.

Finally, though, a network emerged which did in fact discriminate consistently. But not in the way that a typical Kohonen feature map would do it (See Figure 17 which demonstrates the smooth topological requirements of a Kohonen mapping.). Groups of patterns occurred across the link-map which have different shapes. These shapes are components of a number of the input patterns. Figure 44 shows this link mapping at two different times during the learning trials. The learning spread slowly to encompass more and more nodes in the recognition process.

As an example demonstrating that different input patterns consistently cause correlating patterns to emerge from the visual cortex, the images were passed through the network to be stored for later comparison at various time steps. Figure 45 shows two different net-input responses at 30 epoch-sets and 35 epoch-sets, where an epoch-set is a full presentation and epoch for every input image. The figure shows that even starting with a randomized activation pattern, after a few number of cycles, the net-input pattern, which is the sum of the retinal inputs and the visual-cortex lateral inhibition inputs, successfully converged to the same activation level across the visual-cortex maps.

The resulting pattern on the visual cortex is reminiscent of the directionally sensitive neuronal activity in the motor-cortices where a node is subtly directionally selective. These neurons activate to a certain level for all inputs but as the direction of


Non-Discriminating Bubble

Non-Discriminating Distribution

Figure 43. Single Non-Discriminating Bubble and Non-Discriminating Link Maps. Both of these types of patterns were common in our simulations.

the actions changes, the signal drops or raises a little (Kalaska, 1990; Anderson, 1988). Perhaps this is somewhat, the type of mapping that has occurred here.

It seems apparent that unlike the winner-take-all paradigm of a self-organizing network the distributed nature of our PBAs may allow more diversity in the types of mappings that can emerge from it. The typical self-organizing network, enforces a strictly topological mapping, organizing the set of inputs into a dimensionally reduced representation of the input space even when tears or breaks in the mapping would be appropriate (Kohonen, 1988). When we re-accept the neural link as the mechanism of neural communication we demonstrated the capability of multiple activity bubbles which caused the distributed, yet locally topological mappings. Our results are thus a step closer to reaching the diverse set of mappings found in studies of biological brains such as the intercalated, interdigitated, and reiterated maps (Stryker, 1989).

In terms of dimensional reduction, let's consider that the activity bubbles on the two dimensional surface each represent a two dimensional signal. If a given input caused the network to respond with five activity bubbles, we could say that the input signal (8x8 implying 64 dimensions) has been reduced to 12 dimensions from 64.

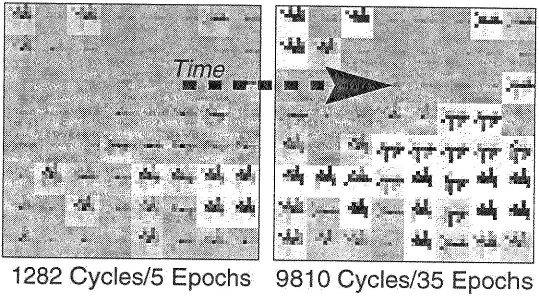


Figure 44. A Distributed Learning Link Map. The distinct differences between different nodal link inputs clearly demonstrates a discriminatory power amongst the different nodes of the visual-cortex.

Various numbers of activity bubbles could occur allowing a shifting dimensionality scale only when needed. Such versatility seems much more satisfactory and accessible than always forcing all the input signals into the two dimensions that are imposed by the standard self-organizing map.

Our results have been based upon a fully-connected network which represents the most difficult categorization that can occur. Now that we have results along these lines, we can relax the constraint and begin to organize not only the mappings, but the connectivity architecture also. We have begun testing a topological mapping between the retina and the visual-cortex. The results seem promising although no analysis has been performed.

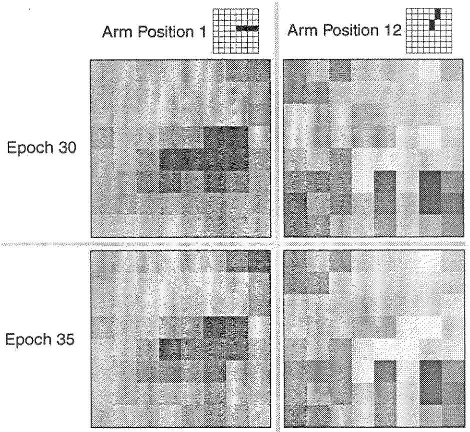


Figure 45. A Comparison of Two Input Pattern's Resultant Net-Input Response at Two Different Epoch-Sets. While not identical, it's clear that there is a similarity between the responses of two different epoch-sets and a distinction between two different presentation images.

6. FUTURE DIRECTIONS AND CONCLUSION

Understanding neurally-based intelligence is a multileveled multidisciplinary goal. In this thesis we have developed a multileveled forum using neurobotics: the study of neurally inspired intelligent systems which causally interact with the external world. Neurobotics has been developed along three basic approaches in this thesis -- the theoretical timescapes and meta-brain principles (Section 2 and Appendix A), the computational neurobotics simulation package (Sections 3 and 4), and the experimental simulations of Neurobot II (Section 5).

Our results provide evidence that by using a neurally inspired cortical map, a partially distributed representation of the input space can be encoded rather than a more constrained topological representation. By using cortical topologies as a basis, we have presented the case for continuing this line of research, fortifying our preliminary theoretical, computational, and experimental components.

Our results, though, demonstrate a considerable present gap between the scope of our theoretical goals and the scope of our experimental simulations. Still, it is desirable that we organize our higher level concepts and theories into a framework permitting direct reference to lower levels of description. This is reasonable due to the interest and quantity of research in the brain at the different levels. Added to this is the inspiring fact that the brain *works*; that it is an example of the kind of entity we would like someday to create or simulate.

By studying neurobotics we are forced to make a connection *between* all levels. Spanning from the highest conceptual level to the lowest we have: the philosophical discussions of the abstract concepts of mind such as functional consciousness and the dualism (Churchland, 1985), schemas (Arbib, 1989; Arkin, 1989), evolutionarily-layered robotic needs (Brooks, 1986), huge neurally- and brain-inspired multi-repertoire sensorimotor systems (Reeke et al., 1989; Edelman, 1989), the architecture and functioning of the real brain (Stryker, 1989; Sarat & Netsky, 1981; Bullock et al., 1977), abstract homogeneous neurally-inspired connectionist networks (Kohonen, 1988; Mel, 1990; Kuperstein, 1988b), the intricate biological neural models from clusters of neurons to single synapses (Purves, 1988; Bower, 1990; Koch & Segev, 1989; MacGregor, 1987). We have wrestled with these multiple levels in the development of a simulation strategy for neurally-based intelligence.

Beginning with our experiments, we now summarize what we have accomplished and then discuss future directions for research.

6.1. The Experimental Future

Our experimental research points out how a small scale exploration of a fully connected single layered neural network fits into the larger picture of sensorimotor systems and type I, II, and III systems.

In our main result, we found that a distributed, dimensionally-reduced representation of an input space can emerge from a fully connected laterally-inhibited neural network.

We believe that under stable closed-loop conditions, if a series of pseudo-brain areas of a similar nature were connected together, each PBA, based upon its position in the loop, its architecture, and its topological connectivity, would generate a *meaningful* distributed map, similar to our results. Thus in terms of our criteria laid out in the objectives of Section 1, we have indeed found that the mappings of a less abstracted self-organizing network can have extended flexibility.

To continue along the present course, we will require the freedom to easily integrate many different PBAs with different connectivities and physically based models of the external world. To accomplish this, there are a number of different directions that our experiments at this low level may go. For instance, the neural models of activation and learning can be made more similar to biological neurons. This approach would bring us closer to the biological validity at the neural level, but it would probably not be an immediately valuable tool when building large systems because of the increased computational burden.

A more useful approach would be to maintain simple but robust neural activation and learning paradigms, like Neurobot II's or Darwin III's, so that we may characterize the different types of cortical organizations that arise from a single PBA by varying the neural connectivities and variables (Edelman, 1989; Reeke, 1989). One advance would be to make the links somewhat plastic. Rather than a completely predefined brain architecture, a design would be given a time period over which links go through growth or attrition, thus paralleling the development-scape more accurately. By using simple and possibly plastic neurons with an accurately understood cortical characterization, we perhaps could build larger systems of integrated multiple brain areas that self-calibrate themselves so that fine-tuning by the user is not required.

Another direction is to study different types of topological models of neural connectivity aside from the simplest cortical models. Exploring *globular* (3D) areas and more complex layering of areas will require further biological information on the types of connectivities and projections existing in the brain. For instance, the thalamus is a massive subcortical controller sending projections throughout the neocortex (Pribram, 1971). By looking into these more advanced neural structures in the brain, we may be able to develop a set of well defined PBAs that abstract the real brain nuclei at a manageable level yet maintain the benefits of a neural architecture.

One final direction would be to consider architectures based upon other media. Fiber optics, for example allows freedom of computation over an extended distances unlike a neuronal-based structure which must remain in a compact volume in order to succeed. This shift in paradigm might provide a new technology for the development of intelligence.

In general, to develop a good understanding of the brain and its capabilities, we must develop good models of the brain from which consistent and dynamic empirical data can be generated. The future work discussed here can be incrementally developed while intermingling with theoretical aspects, giving *meaning* to the systems by adding the sensorimotor causal bidirectional interaction and, eventually by adding an evolutionary component, giving the neurobotics models directable *values* or skills.

6.2. The Computational Future

With the neurally inspired PBAs and interconnected neural areas, we are ready to proceed to the next aspect of study. First, we developed an abstraction of brain areas sufficient to reveal informative empirical data which also gives us a go-between language to describe both brains and networks. By building sensorimotor systems with our PBAs, we create an external feedback loop which brings the designs into the neurobotics realm. Neurobot II was our first attempt at creating such a system. We believe we are on the verge of creating large, stable, and complex neural repertoires and connectivities that can be supported and observed in a real-time environment. Once this happens, observation of our designs will lead to further insight so that our highest goals can then be sought.

Reaching this point in modeling, there are two obvious avenues for future work to pursue. Clearly, simulation of the sensorimotor systems must be the main direction. This involves experimenting with different neural architectures and different physically based models of the real world. We would begin, as we have, with very simple interactions and eventually make more complex external worlds. While the designs do not necessarily have to align themselves accurately with the brain, we would begin along those lines and perhaps work toward specific applications and also toward more accurate modeling of the brain. One clear benefit of using CLOS to support this whole investigation is that the style in which it is programmed is very extensible. By using multiple inheritance, a set of functions of a PBA can be built, where these functions can "acquire" further modifications as we improve the models (see Appendix B).

Our second avenue involves developing further simulation and visualization tools. The NSP provides useful tools for observing various aspects of the neural simulation in progress. However, many more tools can be created to clarify the behavior of the different PBAs, gross-links, and other elements. At this point, one of the most valuable tools would be a inter-area connectivity window that would be able to express some or all of the link weights between PBAs. This was left out of the present system because it was computationally too expensive to perform while simulating and will be less of a problem in the future. Another development would be to provide every type of object, whether input, output, or external, with its own cycling mechanism. This is a simple development that can be added to the NSP.

A central simulation issue involves the question of the scalability of brainlike systems. Mammalian brain sizes range over a scale of 1000:1 (human to rat). Yet they exhibit similar skill acquisition abilities. In other words, we do not see a one thousand fold improvement in physical or mental abilities of humans over rats. Multigrid simulation seeks to capture this biological scale invariance of brains (McCormick, 1988). While the NSP presently develops a single system based upon the parameters introduced, a multigrid NSP would generate parallel and interconnected systems of varying scale. This could also be a useful visualization/optimization tool. The systems of varying scale need not interact only with their own PBAs, but could have the added ability of changing their up- and down-scaled counterparts. While the complexity of the brain mappings may not allow this technique, it is a possible technique for not only speeding up the "learning" of a system, but also for visualizing its different aspects. Using this concept, a user could select the scale of each PBA to be observed and observe the relationship between counterpart PBAs. Though much thought must go into its design, a multigrid NSP might significantly increase the efficiency of the neurobotics simulations.

The NSP seems to be a viable tool for providing many different avenues of present and future research. Now that we have established a computational framework for developing our neural simulations, we can step a further level back and apply our guiding principles as to how the neural and the architectural aspects of neurobotics, can be designed and how they emerged.

6.3. The Theoretical Framework Future of Neurobotics

We would like to take a further step back and see if it is possible to place a theoretical framework upon the development of neurobotics and intelligence. Throughout the thesis, we make reference to the type I, II, and III timescapes. There seems to be a rather natural division between these different timescapes. Because we can base our theories upon the physical world in which intelligent entities evolved, we may be able to describe a set of principles that are independent concepts but, combined in a subsumption chain, characterizes the development of the brain. In Appendix A, we explore meta-brain principles (MBPs), which are loosely based upon the physical considerations. Ultimately, we should look for ways to encode these principles to make them an integral part of the design process.

Although one may argue that the timescape concept is not necessarily a discrete classification, but rather a continuum, there certainly seems to be a clear distinction between the genetic contribution of evolution to the brain (type I) and the shorter timescapes (type II and III). One could further argue for a discrete distinction between development-scape and the thought-scape based on the psychological terms of short-term and long-term memory, respectively. Rather than use these definitions, though, we have defined the thought-scape type III system to represent the present state of the brain, while the type II development-scape contribution modifies weights of the links in a more long-term manner. A type II system affects its physical connectivity strength. Changing the connectivity strength seems quite different from both the plasticity, growth, and post-natal development defined by evolution, and also the very unstable vacillation of the neural signals and dendritic spines (Gilbert et al., 1989).

Thus, we feel that it is acceptable to categorize our neurobotics system into this timescape framework, including at the evolution-scape, the highest level. In fact, we find that the inclusion of the evolution-scape is a necessary aspect of understanding the brain because it allows us to consider the full importance and complexity of the brain. Without encompassing evolution, neural network designs have a distinctly ad hoc flavor.

While our attempts at extracting meta-brain principles has been relegated to the appendix, the potential for developing such principles seems reasonable. Based upon the fact that such an evolution has empirically occurred, it may be possible to develop a set of conceptual principles that parallel evolutionary development and can be implemented in an environment like CLOS, thus giving the NSP a powerful tool for evolving brainlike intelligence.

6.4. Conclusion

In final summary, we have attempted in this thesis to not only explore a small subset of neural networks with a biological focus, but to explore the concept of brainlike intelligence by imposing a framework of intelligence. Our timescape-based taxonomy has not only revealed the importance of evolutionary contributions to intelligence, but also has allowed us to survey recent sensorimotor systems from this unique point of view and to develop a neurobotics simulation package with which to explore the multitudes of directions suggested in this thesis.

Over time the mysteries of the brain will be unlocked piece-by-piece so that we may use the knowledge to help us create tools allowing us to visualize and understand larger concepts. Most importantly, perhaps we can one day develop an alternative point of view to our own by creating a neurobotics system that successfully passes the Turing test in the most non-deceptive way, by actually having intelligence.

REFERENCES

Ackley, D. H. Associative Learning via Inhibitory Search. *Snowbird Conference on Neural Networks for Computing*. Snowbird, Utah. Private Collection. Saxon, J.

Anderson, R. A. (1988). Visual and Visual-Motor Functions of the Posterior Parietal Cortex. P. Rakic, & W. Singer (Eds.), *Neurobiology of Neocortex*. New York: John Wiley and Sons.

Aoki, C., & Siekevitz, P. (1988). Plasticity in Brain Development. *Scientific American*, **259**, 56-64.

Arbib, M. (1989). *The Metaphorical Brain 2: Neural Networks and Beyond*. New York: John Wiley & Sons.

Arkin, R. C. (1989). Path-Planning Subsumption Motor-Schema: Motor Schema-Based Mobile Robot Navigation. *International Journal of Robotics Research*, **8**(4), 92-112.

Bobrow, D. G., DeMichiel, L. G., Gabriel, R. P., Keene, S. E., Kiczales, G., & Moon, D. A. (1988) *Common Lisp Object System Specification*, X3J13 Document 88-002R.

Bower, J. (1990). Personal Communication.

Braitenberg, V. (1984). *Vehicles: Experiments in Synthetic Psychology*. Cambridge, Massachusettes: The MIT Press.

Brandt, S. (1989). *The Media Lab*. New York: Viking.

Brooks, R. A. (1986). A Robust Layered Control System For A Mobile Robot. *IEEE Journal of Robotics and Automation*, **RA-2**(1), 14-23.

Bullock, T. H., Orkand, R., & Grinnell, A. (1977). *Introduction to Nervous Systems*. San Francisco: Freeman.

Carpenter, G. A., & Grossberg, S. (1987). ART 2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns. *Applied Optics*, **26**, 4919-30.

Churchland, P. (1985). *Matter and Consciousness*. Cambridge, Massachusettes: The MIT Press.

Cliff, D. T., (1990). Computational Neuroethology: A Provisional Manifesto. Cognitive Science Research Paper. CSRP 162, University of Sussex, UK.

Cohen, M. A., & Grossberg, S. (1987). Masking Fields: A Massively Parallel Neural Architecture for Learning, Recognizing and Predicting Multiple Groupings of Patterned Pata. *Applied Optics*, **26**, 1866-91.

Cornsweet, T. N. (1970). *Visual Perception*. New York: Academic Press.

Crick, F., & Asanuma, C. (1988). Certain Aspects of the Anatomy and Physiology of the Cerebral Cortex. J. L. McClelland, & D. E. Rummelhart, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, Massachusettes: MIT Press.

Edelman, G. M. (1987). *Neural Darwinism*. New York: Basic Books, Inc.

Edelman, G. M. (1989). *The Remembered Present*. New York: Basic Books, Inc.

Edelman, G. M., & Mountcastle, V. B. (1978). *The Mindful Brain*. Cambridge, Massachusettes: The MIT Press.

Feldman, J. A. (1985). Connections: Massive Parallelism in Natural and Artificial Intelligence. *Byte*, **10**, pp. 277-284.

Fukushima, K. (1988). A Neural Network for Visual Pattern Recognition. *Computer*, **21**, 65-75.

Gilbert, C. D. (Moderator). (1988). Neuronal and Synaptic Organization in the Cortex. Group Report for P. Rakic, & W. Singer (Eds.), *Neurobiology of Neocortex*. New York: John Wiley & Sons.

Glass, A. L., & Holyoak, K. J. (1986). *Cognition*. New York: Random House.

Grossberg, S., & Todovoric, D. (1988). Neural Dynamics of 1-D and 2-D Brightness Perception: A Unified Model of Classical and Recent Phenomena. *Perception & Psychophysics*, **43**, pp. 241-277.

Hebb, D. O. (1949). *The Organization of Behavior: A Neurophyschological Theory*. New York: Wiley.

Held, R. (1968). Action Contingent Development of Vision in Neonatal Animals. In *Experience and Capacity*. Kimble, D.P. (ed.) New York: New York Academy of Sciences, pp. 31-111.

Hofstadter, D. R. (1985). *Metamagical Themas: Questing of the Essence of Mind and Pattern*. New York: Bantam Books.

Kalaska, J. F. (1990) Cerebral Cortical Neuronal Mechanisms of Reaching Movements. In *Proceedings of the International Joint Conference on Neural Networks*. San Diego. (pp. II-687 - II-692). Ann Arbor: IEEE Neural Networks Council.

Keene, S. E. (1989). *Object-Oriented Programming in Common Lisp: A Programmer's Guide to CLOS*. New York: Addison-Wesley.

Koch, C., & Segev, I. (1989). C. Koch, & I. Segev (Eds.), *Methods in Neuronal Modeling from Synapeses to Networks.* Cambridge, Massachusettes: The MIT Press.

Kohonen, T. (1987). Adaptive, Associative, and Self-Organizing Functions in Neural Computing. *Applied Optics,* **26**, 1490-18.

Kohonen, T. (1988a). The 'Neural' Phonetic Typewriter. *Computer,* **21**, 11-22.

Kohonen, T. (1988b). *Self-Organization and Associative Memory. 2 (Series in Information Sciences).* New York: Springer-Verlag.

Kuperstein, M. (1988a). An Adaptive Neural Model for Mapping Invariant Target Position. *Behavioral Neuroscience,* **102(1)**, 148-62.

Kuperstein, M. (1988b). Generalized Neural Model for Adaptive Sensory-Motor Control of Single Postures. *Proceedings of the IEEE Conference on Robotics and Automation.* (pp. 140-144). Private Collection. Saxon, J.

Lange, T. E., (1990). Simulation of Heterogeneous Neural Networks on Serial and Parallel Machines. *Parallel Computing,* **14**, 287-303.

Lange, T. E., Hodges, J. B., Fuenmayor, M., & Belyaev, L. (1989). DESCARTES: Development Environment for Simulating Hybrid Connectionist Architectures. *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society (CogSci-89).* Ann Arbor.

Lange, T. E., Dyer, M. G. (1989). Dynamic, Non-Local Role Bindings and Inferencing in a Localist Network for Natural Language Understanding. Technical Report UCLA-AI-89-01.

Lenat, D. B. (1990). *Building Large Knowledge-Based Systems : Representation and Inference in the Cyc Project.* Reading, Massachusettes: Addison-Wesley.

MacGregor, R. J. (1987). *Neural and Brain Modeling.* San Diego: Academic Press.

Maslow, A. H. (1970). *Motivation and personality, 3rd ed.* New York: Harper & Row.

McClelland, J. L., & Rummelhart, D. E. (1988). Amnesia and Distributed Memory. In J. L. McClelland, & D. E. Rummelhart (Eds.), *Parallel Distributed Processing: Explorations in the Microstructures of Cognition. II* (pp. 502-527). Cambridge, Massachusettes: The MIT Press.

McCormick, S. F. (1988). *Multigrid Methods: Theory, Applications, and Supercomputing.* New York: Marcel Dekker.

Mead, C. (1990). Time-the Essential Dimension. in M. Caudill (Ed.), *Applications Track of the Proceedings of the International Joint Conference on Neural Networks*. 2 Washington, D. C. (pp. 25). Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc.

Mel, B. (1990). *Connectionist Robot Motion Planning, A Neurally-Inspired Approach to Visually-Guided Reaching*. (B. Chandrasekaran) (Perspectives in Artificial Intelligence 7). San Diego: Academic Press, Inc.

Moravec, H. (1989) Human Culture: A Genetic Takeover Underway. in C. G Langton (Ed.), *Artificial Life*. New York: Addison-Wesley.

Mountcastle,.V. B., (1990) Personal Communication.

Obermeyer, K, Ritter, H., & Schulten, K.. (1990). A Neural Network Model for the Formation of Topographic Maps in the CNS: Development of Receptive Fields. In *Proceedings of the International Joint Conference on Neural Networks*. San Diego. (pp. II-423 - II-429). Ann Arbor: IEEE Neural Networks Council.

Pearson, J. C., Finkel, L. H., & Edelman, G. M. (1987). Plasticity in the Organization of Adult Cerebral Cortical Maps: A Computer Simulation Based upon Neuronal Group Selection. *The Journal of Neuroscience*, 7(2), 4209-4223.

Piaget, J. (1963). *The Origins of Intelligence in Children*. New York: Norton.

Poggio, T., & Girosi, F. (1990). Regularization Algorithims for Learning that are Equivalent to Multilayer Networks. *Science*, **247**, 978-982.

Pribram, K. H. (1971). *Languages of the Brain: Experimental Paradoxes and Principles in Neurophysiology*. Englewood Cliffs, New Jersey: Prentice Hall, Inc.

Purves, D. (1988). *Body and Brain: A Trophic Theory of Neural Connections*. Cambridge, Massachusettes: Harvard University Press.

Reeke, G. N. J., Finkel, L. H., Sporns, O., & Edelman, G. M. (1989). Synthetic Neural Modeling: A Multilevel Approach to the Analysis of Brain Complexity. G. M. Edelman, W. E. Gall, & W. M. Cowan (eds.), *Signal and Sense: Local and Global Order in Perceptual Maps*. New York: John Wiley & Sons.

Sarnat, H. B., & Netsky, M. B. (1981). *Evolution of the Nervous System*. Oxford: Oxford University Press.

Saxon, J. B. (1991). The Neurobotics Simulation Package: An Object-Oriented Neural Sensorimotor Simulation System. Technical Report TAMU-91-020.

Saxon, J. B., & Mukerjee, A. (1990). Learning the Motion Map of a Robot Arm with Neural Networks. In *Proceedings of the International Joint Conference on Neural Networks*. San Diego (pp. II-777 - II-783). Ann Arbor: IEEE Neural Networks Council.

Searle, J. (1984). *Minds, Brains and Science*. Cambridge, Massachusettes: Harvard University Press.

Sejnowski, T. J., Koch, C., & Churchland, P. S. (1988). Computational Neuroscience. *Science*, **241**, pp. 1299-1306.

Steele, G. L. (1990) *Common Lisp The Language Second Edition*. Bedford, Massachusettes: Digital Press.

Stryker, M. P. (Moderator). (1988). Organization of the Cortex. Group Report for P. Rakic, & W. Singer (Eds.), *Neurobiology of Neocortex*. New York: John Wiley & Sons.

Wolfram, S. (1988). *Mathematica: A System for Doing Mathematics by Computer*. New York: Addison-Wesley.

SUPPLEMENTARY REFERENCES

Arkin, R. C. (1987). Motor Schema Based Navigation for a Mobile Robot: An Approach to Programming by Behavior. *IEEE Conference on Robotics and Automation*. (pp. 264-271). Private Collection. Saxon, J.

Barhen, J., Gulati, S., & Zak, M. (June 1989). Neural Learning of Constrtained Nonlinear Transformations. Computer, 22, 67-76.

Bekey, G. A. (1990). NSF Workshop on Bio-Control by Neural Networks. *IEEE Control Systems Magazine*, **10**(5), 30-32.

Bizzi, E. Arm Trajectory Planning and Execution: The Problem of Coordinate Transformation. P. Rakic, & W. Singer (Eds.), *Neurobiology of Neocortex*. New York: John Wiley and Sons.

Bloom, F. E. (Moderator). (1988). The Integrative Function of the Neocortex. Group Report for P. Rakic, & W. Singer (Eds.), *Neurobiology of Neocortex*. New York: John Wiley & Sons.

Buffa, M. G. (November 1, 1988). Neural-Network Systems: The Key to Adaptive Robotics. *Computer Design*, **27**, 57.

Bullock, D., & Grossberg, S. (1988). Neural Dynamics of Planned Arm Movements: Emergent Invariants and Speed-Accuracy Properties During Trajectory Formation. *Psychological Review*, **95**(1), 49-90.

Carpenter, G. A. (1989). Neural Network Models for Pattern Recognition and Associative Memory. *Neural Networks*, **2**, 243-257.

Carpenter, G. A., & Grossberg, S. (1988). The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network. *Computer*, **21**, 77-88.

Carpenter, G. A., & Grossberg, S. (1986). A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine. *Computer vision, graphics, and image processing*, **37**, 54-115.

Carpenter, G. A., & Grossberg, S. (1984). A Neural Theory of Circadian Rhythms: Aschoff's Rule In Diurnal and Nocturnal Mammals. *American Journal of Physiology*, **247**, R1067-82.

Caulfield, J. H., & Armitage, D. (1989). Adaptive Resonance Theory of Optical Pattern Recognition. *Applied Optics*, **28**, 4060-1.

Churchland, P. S. (1986). *Neurophilosophy: Toward a Unified Science of the Mind-Brain*. Cambridge, Massachusettes: The MIT Press.

Drexler, K. E. (1986). *Engines of Creation*. New York: Anchor Press/Doubleday.

Feldman, J. A., Fanty, M. A., & Goddard, N. H. (1988). Computing with Structured Neural Networks. *Computer*, **21**, 91-103.

Fetz, E. E., Shupe, L. E., & Murthy, V. N. (1990). Neural Networks Controlling Wrist Movements. *International Joint Conference on Neural Networks*. San Diego. Ann Arbor: IEEE Neural Networks Council.

Fukushima, K. (1987). Neural Network Model for Selective Attention in Visual Pattern Recognition and Associative Recall. *Applied Optics*, 26(23), 4985-4992.

Gaudiano, P., & Grossberg, S. (1991). Vector Associative Maps: Unsupervised Real-Time Error-Based Learning and Control of Movement Trajectories. *Neural Networks*, **4**, 147-183.

Gaudiano, P., & Grossberg, S. (1990). A Self-Regulating Endogenous Generator of Sample-and-Hold Random Training Vectors. M. Caudill (Ed.), *Applications Track of the Proceedings of the International Joint Conference on Neural Networks*. **2** Washington D. C. (pp. 213-216). Hillsdale, N. J.: Lawrence Erlbaum Associates.

Girosi, F., Poggio, T., & Caprile, B. (July 1990). *Extentions of a Theory of Networks for Approximation and Learning: outliers and negative examples*. (A.I. Memo No. 1220, C.B.I.P Paper No. 46). Massachusettes: Massachusettes Institute of Technology.

Gleick, J. (1987). *Chaos*. New York: Penguin Books.

Grossberg, S. (1976). Adaptive Pattern Classification and Universal Recoding, II: Feedback, Expectation, Olfaction, and Illusion. *Biological Cybernetics*, **23**, 187-202.

Grossberg, S. (1986a). *The Adaptive Brain I: Cognition, Learning, Reinforcement, and Rhythm*. Amsterdam: Elsvier/North-Holland.

Grossberg, S. (1986b). *The Adaptive Brain II: Vision, Speach, Language, and Motor Control*. Amsterdam: Elsvier/North-Holland.

Grossberg, S. (1987). Neural Dynamics of Attentionally Modulated Pavlovian Conditioning: Blocking, Interstimulus Interval, and Secondary Reinforcement. *Applied Optics*, **26**, 5015-30.

Grossberg, S., & Schmajuk, N. (1987). Neural Dynamics of Attentionally Modulated Pavlovian Conditioning: Conditioned Reinforcement. Inhibition, and Opponent Processing. *Psychology*, **15**(3), 195-240.

Hein, A. (1974). Prerequesite for Development of Visually Guided Reaching in the Kitten. *Brain Research*, **71**, 259-263.

Kaas J. H. (1988). Development of Cortical Sensory Maps. Individual Research for P. Rakic, & W. Singer (eds.), *Neurobiology of Neocortex*. New York: John Wiley & Sons.

Maslow, A. H. (1959). *New knowledge in human values*. New York: Harper.

Maslow, A. H. (1968). *Toward a psychology of being*. Princeton, N.J.: Van Nostrand.

Micheli-Tzanakou, E. (1988). Neural Aspects of Vision and Related Technological Advances. *Proceedings of the IEEE,* **76**(9), 1130-1142.

Moody, J. (1989). Fast Learning in Multi-Resolution Hierarchies. D. Touretzky (ed.), *Advances in Neural Information Processing Systems*. San Mateo, California: Morgan Kaufmann.

Moody, J., & Darken, C. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation,* 1, 281-294.

Pellionisz, A. (1988). Tensor Geometry: a Language of Brains & Neurocomputers, Generalized Coordinates in Neuroscience & Robotics. R. Eckmiller, & C. von der Malsburg (eds.), *Neural Computers*. (NATO ASI Series; Series F: Computer and Systems Sciences 41). (pp. 381-391). New York: Springer-Verlag.

Saxon, J. (1990) Private Collection. Saxon, J.

Schwartz, A. B. (1990). Motor Cortical Activity During Sinusoidal Tracing. In *Proceedings of the International Joint Conference on Neural Networks*. San Diego. (pp. II-681 - II-686). Ann Arbor: IEEE Neural Networks Council.

Sekuler, R., & Blake, R. (1985). *Perception*. New York: Alfred A. Knopf, Inc.

Tawel, R., Eberhardt, S., & Thakoor, A. P. Neural Networks for Robotic Control. *Snowbird Meeting on Neural Networks for Computing*. Snowbird, Utah. Private Collection. Saxon, J.

van der Loos, H. (Moderator). (1988). Principles of Cortical Self-Organization. Group Report for P. Rakic, & W. Singer (editors.), *Neurobiology of Neocortex*. New York: John Wiley & Sons.

APPENDIX A

META-BRAIN PRINCIPLES

We are beginning to realize the crucial connection between evolution and intelligence (Edelman, 1987). Typically, these differing subjects would be considered separately, leaving the Darwinians to their species studies, and the computer scientists to their simulations. We wish, however, to base our research on a level where this combination is natural, where the relationship between evolution and intelligence is inherent. Thus we are striving for some theory which will complement both the empirical world data and our empirical simulation data.

In this appendix, a set of high level principles, called *Meta-Brain Principles* (MBPs), are proposed, which represent a set of building blocks that, when combined in a subsumption architecture, can represent many forms of intelligence such as human, robotic, or neurobotic. They are guidelines for *mind modeling*, a process somewhere in between neuronal modeling and AI and covering a scope that encompasses the evolutionary aspect of intelligence.

Each MBP may be used as a design constraint independently of the others. But here the principles are employed in a chain where each MBP is be developed within the framework of all preceding ones. This incremental foundation is meant to parallel the evolutionary emergence of intelligence.

To establish our set of MBPs, we look for invariant characteristics of the mammalian brain, aspects that relate to more than one area of the brain, or all brains, or even their evolutionary relatives. Our basic assumption is that a trait which has presented itself genetically becomes a *concept* which can replicate itself in different ways throughout the rest of evolution.*

We begin the section by presenting an evolutionary scope from which we draw insight. It is termed the Churchland/Moravec Progression after the authors whose comments it comprises.

---

*The subsumptive aspect of the MBPs and the genetic ability to replicate these concepts brings home again the utility of using CLOS as a means of modeling the systems because of its ability to "mixin" various concepts into a hierarchical structure very similarly to the process of evolution.

A.1. The Churchland/Moravec Progression

By combining two evolutionary paths outlined by Churchland and Moravec, a coherent continuum, the *Churchland/Moravec (CM) progression*, is created that describes the evolutionary path of intelligent life on this planet (See Figure 46). Combining Churchland and Moravec's progressions we come up with the following: self-replication, self-maintenance, energy-use, sex, multiple cells, death, DNA, a central nervous system, walking, learning, using tools, culture, and finally post-DNA reproduction – man's potential future (Churchland, 1985; Moravec, 1989).

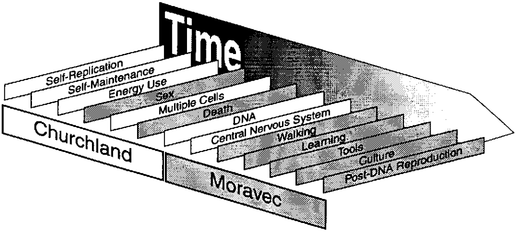


Figure 46. The Churchland/Moravec Progression. In this progression, which is a combination of Churchland's (light gray) and Moravec's (dark gray) steps of evolution, the attributes of intelligent beings are described. Each tier contains a permanent integration and continual reworking of all previous steps. There is also the potential that any step that has occurred can present itself in new ways at the micro or macroscopic levels.

Churchland places self-replication as the first of many evolutionary *choices* ultimately leading to man's emergence. "With respect to achieving large populations, the capacity for self-replication is plainly an explosive advantage." (Churchland, 1985). Unlike a virus which needs a host to replicate itself, man's evolutionary predecessors quite early on became self-replicating. This capability is permanently ingrained in all further evolutionary steps and can manifest itself where it proves helpful to the species. Every step in the CM progression, plus many smaller steps in between, all become permanent *tools* for further evolutionary choices. Each step represents a new level of biological facility. Recall Maslow's hierarchy of needs

(Maslow, 1970): each need was based upon whether all lower needs were sufficiently satisfied. The CM progression represents the evolutionary steps that build the needs.

The CM progression not only seems to parallel Maslow's hierarchy, but also Brooks' set of levels. Figure 47 shows a comparison between the hierarchies of human needs and of *robotic needs* with the CM progression. It's interesting to observe how the needs seem to *emerge* from the steps of the CM progression. The conceptual needs of Maslow, have been abstracted from the CM progression. Similarly, we hope to implement a similar progression from which our neurobotics goals emerge. For instance, sexual reproduction, is certainly not an early necessity for our robots as it is for humans. Finding energy on the other hand could be.

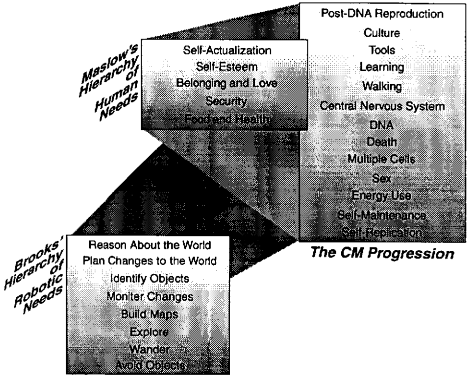


Figure 47. Comparison Between Human Needs and Robotic Needs with the CM Progression. The parallels between Maslow's hierarchy and Brooks' are related to the evolution that took place in humans following the CM progression. Perhaps a similar progression to our neurobotics goals can be created.

As for the timescapes of Section 2 (Table 1), they form an orthogonal dimension to the MBPs. Our timescapes focus on the different physical characteristics of the brain such as the architecture, learning, and activation schemes. The MBPs however are based on an abstract level. They cut across the timescape types so that the MBPs involve all three timescapes. Each MBP has its own relationship to the thought-scape (type III), development-scape (type II), and evolution-scape (type I) of intelligence.

Maslow's hierarchy of needs has a basis in the CM progression. We would like to devise a similar progression for a broader spectrum of intelligent beings that includes humans and robots. We do this by attempting to abstract the incremental contributions that occurred in the CM progression. We end up with MBPs that encompass and, in a sense, *predict* the CM progression.

A.2. The Principles

The four MBPs shown in Table 3 are meant to be independent conceptual building blocks which, when ordered in a subsumptive organization, provide a framework for developing brainlike entities regardless of the media, be it wet-, hard- or software. Figure 48 shows the relationship of the proposed MBPs. Individually, the MBPs are simple; in their combination they become complex. The difference between the MBPs and the CM progression is that the MBP is meant to describe only an independent advance, while an element of the CM progression is the sum total of all previous steps. In other words, the CM progression is a biological *integral* of MBPs. Further though, the MBPs outscope the CM progression. MBPs 1 and 2 are precursors of the progression, prescribing its existence along with similar progressions for non-humans. MBPs 3 and 4, are steps within the progression which direct any progression towards brainlike intelligence.

Table 3.

The Meta-Brain Principles. These are independent building blocks, which when placed in a subsumption chain starting with MBP 1, become very influential design constraints.

	Meta-Brain Principle
MBP 1	Bidirectional Interaction
MBP 2	Incremental Development
MBP 3	Compact Population-based Computation
MBP 4	Neurocentrism

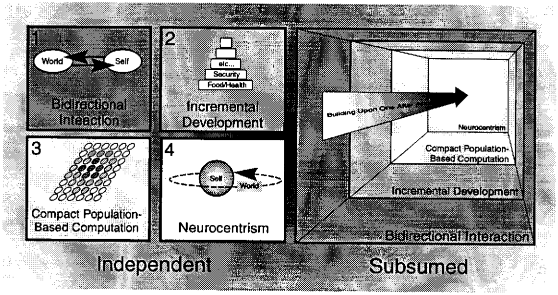


Figure 48. Independent Meta-Brain Principles Placed in a Subsumption Chain. While each MBP can be considered a separate independent descriptive element, by combining them in a chain, we provide a design environment for brainlike entities.

A.2.1. MBP 1: Bidirectional Interaction

Another term for this MBP would be "closed external feedback loop". In other words, we wish to provide a continual causal relationship between our entity and the external world. This is the basis of all vital entities, coming millions of years before self-replication of the CM progression. Without this connection, there can be no change to the entity. Thus MBP 1 describes the *world* necessary for an intelligent entity to exist.

Yet this principle is almost always overlooked because of the complexity involved in closing the loop. As we have seen, for any intelligent being to learn, an association must be created between thought and action. Piaget's Circular Reaction Paradigm and Hine and Held's experiments on kittens demonstrated this. Simply put, without this MBP, meaning can not be created. And, without meaning, there is no intelligence, merely syntax (Searle, 1984).

Bidirectional interaction between our intelligent entities and the external world applies in all three timescapes. As we have defined, a type III system is typically an open loop, implying that the actions are not tuned by the sensory perception. Nonetheless, the mechanism designed must be capable of providing for a causal loop of action/reactions. For a type II system, with learning added, an association can be formed which may then alter future interactions. In a type I system, the interaction between our entity and the external world invokes changes over time in the actual framework, or wiring, of the entity. This adds another level of complexity to the design of an intelligent entity. This level of complexity has only been scantly considered indirectly by the fields of genetic algorithms and artificial life (AL) but may also have its place in developing neurobotic intelligence. Thus it also becomes a basis for our next MBP. As our systems become more complex and more real-world like, MBP 1 begins to resemble aspects of physically based modeling.

A.2.2. MBP 2: Incremental Development

While MBP 1 provides the universe in which the CM progression exists, we consider MBP 2 to *be* the CM progression. Building upon what already exists is the basic principle of incremental development. It forks into two paths, at the macro and micro scales. Macroscopically, a brain area may eventually split into two different brain areas. Microscopically, a successful paradigm can be spread throughout the system such as the specialization of some cells into neurons, or more generally, a localized but useful brain architecture. Further, the incremental levels of MBP 2 are also subsumed so that the priority of a level changes depending on whether its requirements are satisfied.

In the extreme case, if we designed a full type I/II/III system using an artificial life technique like genetic algorithms, MBP 2 would be automatic. The bidirectional interactions of MBP 1 would alter the physical structure of the entity based upon its interactions with the world. Some entities would fail and some would succeed to be further augmented. But then, we could stop here and observe whether intelligence ever emerged from the brainlike structure that might emerge. It could be a long grueling process, though, considering that evolution took billions of years in real-time and it is doubtful that our simulations could ever run that quickly.

It is more likely that we would wish to lend a hand in orchestrating the incremental development, as Brooks has done by imposing layers, subsumption, and a priority scheme for the layers (Brooks, 1986). We can classify his robot in terms of the MBPs as a type III robot in which MBP 1 was based upon the robot's real world sensory data and actual motion-producing wheels, and the subsumption architecture representing MBP 2 was custom designed in. We shall see that MBP 3 represents the implementation tools used to create the architecture.

In general, MBP 2 represents an affect on the type I aspect of a robot. With Brooks' robot this is also the case. While there was neither automatic evolution nor automatic learning, his layered design was based upon the success of the more primitive layers. As far as intelligent brainlike behavior, though, this is as far as Brooks robot can go because his system lacks the ability to abstract, his system signals are symbolic and thus, incorporating learning would be a difficult feat. A neural type structure encapsulated in our next MBP, would allow a more robust emergent behavior.

A.2.3. MBP 3: Compact Population-based Computation

The next step in this meta-evolution takes us beyond the physiological requirements of humans, as defined in the CM progression, and straight to the centrally controlled aspect of intelligence. Since we wish to develop intelligences not necessarily based on the human body, MBP 1 and MBP 2 give us the capability to design any type to allow interaction with the external world (e.g. using the physical architecture such as robot arms, wheels, sensors, etc). MBP 3 provides the physical architecture for the intelligent entity. As opposed to symbolic processing (such as Brooks' Lisp modules), we have chosen processing elements that are much more brainlike. By compact population-based computation, we mean that the processing of information is more than just parallel and distributed.

Firstly, *population-based computation* has high granularity and is more statistically based than traditional parallel concepts. Unlike most parallel computers, which have discrete components and specific tasks, the quantity of *processing elements* (in our case neurons) is so large that the failure of an individual component is statistically negligible. The Dahlem Konferenzen discussion on cortical organization, along with our main result of Section 5, demonstrate that information may be stored in many different ways with a populational approach (Stryker, 1989). The non-globally optimized redundant pattern of links across a brain area shows how a single element, which has a bias towards a certain input, contributes to many different types of output. Population-based computation seems to be a much more powerful processing style than a parallel processing system or a holographic organization, the opposite extreme.

Secondly, population-based computation is biologically efficient, or *compact*. The wrinkled appearance of the neocortex is evidence that evolution is attempting to optimize the computation/size ratio. This involves getting the most out of the little space in the cranium. Thus, as we have stated, concepts such as full-connectivity between sets of neurons seem quite outlandish. Although links between neurons are very microscopic, evolution has certainly come up with other ways of organizing the brain's architecture so that it can have knowledge inherent in its *spatial* connectivities along with its learned associations. The value in Kohonen's work is that it makes the importance of spatial connectivity clear: a dimensionally reduced organization of an input space becomes a spatial/physical organization rather than an abstract high dimensional one (Kohonen, 1988). This is why we stress the importance of the topological mappings.

MBP 3 is inspired by it's relationship to the brain, and also by the promise that modeling such neural architectures will yield better results than traditional AI techniques. By defining compact types generally, there is also some room for potentially non-neural populations. For instance, another populational technique could be based upon artificial life (AL) type strategies or a more high speed connectivity strategy such as using fiber optic axons. In this thesis, though, we have basically assumed a neurally inspired design.

Bower has remarked that convergent evolution hints at the possibility that evolution has a natural tendency to optimize functionality (Bower, 1990). In other words, the massive populations of neurons compressed together with efficient connectivity between areas in the brain is a near optimal method of accomplishing the functionality the brain achieves. Considering the length of time that the CNS has remained a growing part of intelligent beings, one could presume that a neuronal architecture has been evolutionarily selected and thus is a very useful method of creating intelligence. This, of course, assumes that a higher level of intelligence implies a higher chance of survivability.

Further, as we have pointed out, schemas, while useful from an abstract point of view, are barely capable of crossing the type II barrier because learning seems too symbol specific, there can be little generalization. On the other hand, such a case would be immediately adaptable for a neurally inspired system. In general, a neurally realized type I, II, or III component is much more understandable, realizable, and integratable . The benefits of a populational implementation of intelligence are the ease of integration and learning and the possibility of having emergent developments.

A.2.4. MBP 4: Neurocentrism

Our fourth MBP represents the way in which our intelligent entities organize the world they live in. We are taking license with the term *neurocentrism* in the sense that it is in actuality independent of the neural aspect, but considering the subsumption of the MBPs in our case, the 'neuro' fits. Neurocentrism is similar to egocentrism but viewed from an organizational point of view as opposed to a psychological perspective. Basically, neurocentrism asserts the external world is viewed and organized with respect to the self.

Neurocentrism is not a conscious thing; it is more a matter of the brain's actual architecture. A simple example would be the way the brain "sees" walking towards the door. Rather than moving towards the door, the brain makes the door come to the self, by walking. In other words, as the brain develops associations with its internal feelings and its perceptions, it makes models of the world from its own perspective. The self is central and the world changes. An example of this is the body-centered coordinate system found in the hippocampus.

The basic concept of neurocentrism is a style of modeling the external world. The more accurately one can model the external world, the more accurately the world can be made to perform what one wants. This evolution driven, feedback-based, centrally-controlled property means that as time continues, models will more accurately model the external world. A fish has very little control over the world. It can basically only alter the world by swimming, thus shifting another part of the world *to* it. More evolved species are adept at manipulating the world at a higher level: Otters build dams, monkeys have combined sticks together to reach food, and finally humans, making a giant leap beyond monkeys, store symbols of the external world in their own heads, that can represent the entities as though they were present. This union of speech and imagery allows humans to generate intricate plans and to consider things in a much larger scope, changing more of the world by being able to learn how to change things. For instance, it could be said that intelligent people have a stronger modeling brain. The ins and outs of the way the world works from language, mathematics, physiology, psychology, etc. is based on being able to store and interact with the various elements so they all fit with respect to the neurocentric host, the subjective individual who is interacting with the external world.

Neurocentrism rounds out the MBPs we have developed. Individually, the MBPs can be modeled in one way or another; but when combining them into a subsumption architecture, the goal becomes much more difficult.

In general, the MBPs represent the guiding principles which caused the CM progression to emerge and take the direction it took with respect to the evolution of intelligent beings. MBP 1 and MBP 2 are the preconditions of the progression and MBP 3 and MBP 4 are considered important landmarks which allowed the progression to turn towards the evolution of intelligent beings. While these high level concepts require empirical validation, they suggest a synthesis that will eventually lead to a well grounded theory of neurobotics.

APPENDIX B

DESCARTES

DESCARTES, or the Design Environment for Simulating Connectionist ARchiTecturES, is a powerful language for defining and simulating neural networks (Lange et al., 1989). It was selected as the basis for developing the NSP because of its versatility and object-orientedness. DESCARTES' is based upon the connectionist concepts recently resurfaced by the PDP group (McClelland & Rummelhart, 1988). Thus the main structural elements of DESCARTES are the *node* and the *link*. In this appendix, we describe these elements and the various mechanisms which *grow*, *control*, and *cycle* them and allow easy development and simulation of neural network systems. DESCARTES and the NSP are both based upon CLOS, which is a very relaxed, versatile language. Therefore we begin with a brief discussion of some CLOS designs capabilities.

B.1. CLOS

CLOS is one of the most powerful object-oriented languages available. With CLOS, it is possible to create unique and varying sets of *classes* and *methods*, yet still maintain decent computational speed and understandability. CLOS allows message passing, multiple inheritance (*mixins*, classes that are designed to be merged into other classes by adding or modifying slots and methods), *before*, *after*, and *around* methods, and meta-objects. These features enable DESCARTES, and thus the NSP, with a very useful extensibility not found in traditional programming languages. Further, CLOS is an interpreted language, allowing more freedom to manipulate the elements of the system with ease.

There is much freedom in the way that a CLOS program can be developed. While there are many manuals describing CLOS (Keene, 1989; Bobrow et al., 1988), below, we begin with some CLOS basics so that we may mention some of the useful nuances of using CLOS that are important building blocks of DESCARTES.

The basic building block of CLOS is the *class*. A class is similar to the data-structure aspect of conventional programming languages. A class contains *slots* which are placeholders for values, where a value can be any data-structure (e.g. lists, cons-cells, arrays, or especially, other classes). A class defines the structure of a CLOS object. Instances of classes are the actual objects which are created by *instantiation*.

Upon the class instances, *methods* can be applied which perform some action with or to the class and its slots. Methods are the procedural part of a conventional language. The value of object-oriented method, is that it is intimately tied to the class and the same method can be used to manipulate different classes differently. For instance, in simulating a DESCARTES system, it is necessary to grow nodes, links, and controllers. To grow a controller, one would call `(grow controller-name)` and similarly, to grow a node, `(grow node-name)`. CLOS checks to see if there is a grow method for the each class. If so, the specific method for that class is called. Growing a controller may grow a set of nodes and links, while growing a node may initialize it's activation, grow more links, or grow more nodes.

The real power of CLOS, emerges in two ways. Firstly, rather than building up a strict hierarchy from a single *superclass*, classes can be pieced together with multiple superclasses, or mixins, which can each represent a specific functional or structural aspect of a class. For instance, in Figure 49, an NSP *cortical-control* is built by mixing in: *edge-effect-functionality, cortical-saver-widget, cortical-draw-widget, cortical-control-widget*, and *area-control*, where area-control contains simpler basic controller mechanisms including those of DESCARTES basics, and so on up the hierarchy (See Appendix E). Further, each mixin represents a concept, such as the draw-widget ,which gives cortical-control the ability to be drawn.

Secondly, rather than using the same name methods on *different* classes, a method manipulates a class which has been built up by inheriting the characteristics of many other classes combined in a hierarchy. This introduces the *before, after,* and *around* methods which can be made to envelope the upward and downward propagation of a hierarchy of method calls. This powerful tool can be used for incremental growth of complexity of the program or the models being created.

Finally, to demonstrate the ultimate utility of using CLOS, we turn the MBPs to hypothesize about how far CLOS can carry our neurobotics goals (See Appendix A). We notice that CLOS can become a powerful tool in two ways. Firstly, the subsumption of the MBPs themselves can be easily organized in a CLOS multiple superclass system (see Figure 50). Further, the incremental development described by MBP 2 can also be represented easily in CLOS by adding lower and lower subclasses as evolution takes place. Every new mixin in an incremental development is added to the lowest class. They can contain new slots and new methods. Thus the mixin

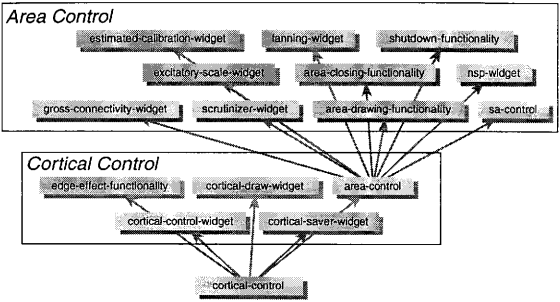


Figure 49. MixIns are Multiple Superclasses. By thinking in conceptual terms, rather than creating large objects, or classes, building blocks can be made up of *mixins* which contain a specific concept with slots and methods that can then be combined with others to form an object. In DESCARTES, usually there is one superclass (in this case *area-control*) which contains the hierarchy.

would be considered the evolutionary *step* while the new class itself would be considered the new *level*, just as in the CM progression.

B.2. DESCARTES

We can now turn our attentions to the DESCARTES program. DESCARTES provides tools for designing hybrid neural networks of any complexity with the CLOS mechanisms. Thus, the NSP is actually a DESCARTES program which has increased the level of complexity so that a higher level of management is needed. Below, we discuss the different structural elements of DESCARTES and the general functionality of the system.

B.2.1. Basics, Functionalities, and Widgets

There are two elements which make up the building blocks of DESCARTES. The *basic* classes represent the main class definitions. The objects that DESCARTES manipulates are the basic classes. The other half are the *mixins* that represent modular, generally orthogonal components, which are added to the basic classes to create the DESCARTES primitives, custom objects, and capabilities: nodes, links, controllers,

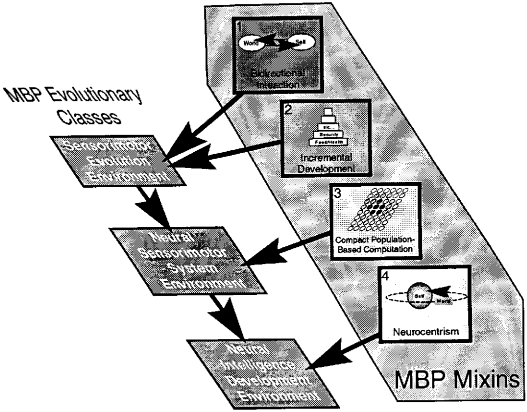
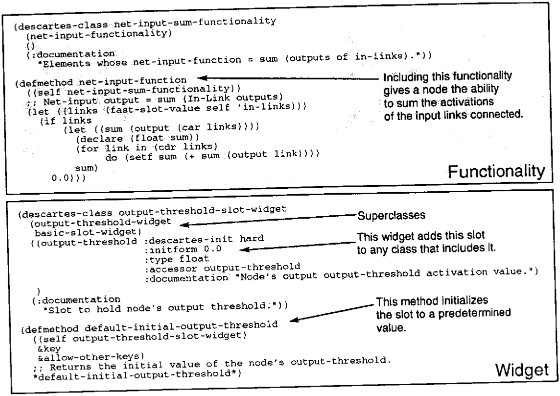


Figure 50. MBP Subsumption Using CLOS. By defining the MBPs as mixins, they can be combined together to represent their own progression similar to the CM progression (See Appendix A). Here, the levels of progress are in terms of the power of the simulation environment. Thus, by adding MBP 3, the environment becomes neural, etc.

growth, cycling, etc. Thus, there is a large number of mixins which a user can combine together to customize their design.

DESCARTES breaks up the mixins into two different types: *functionalities* and *widgets*. Their difference is based whether the mixin affects the neural processing. A functionality alters the neural signal as it is passed through the various stages, while a widget adds methods and slots not directly related to the neural signal. For example, a mixin that alters the display of various slots in an object is a widget, while one that modifies the output of a node by subtracting a bias is a functionality. Some examples of functionalities and widgets for DESCARTES are shown in Figure 51. Any object in the DESCARTES system may include various mixins and widgets into their superclasses, thus giving that class a new capability. To define these types of mixins, `(descartes-class` *`definition`*`)` is called which has added capability beyond the



```
(descartes-class net-input-sum-functionality
  (net-input-functionality)
  ()
  (:documentation
    "Elements whose net-input-function = sum (outputs of in-iinks).")) 

(defmethod net-input-function
  ((self net-input-sum-functionality))
  ;; Net-input output = sum (In-Link outputs)
  (let ((links (fast-slot-value self 'in-links)))
    (if links
        (let ((sum (output (car links))))
          (declare (float sum))
          (for link in (cdr links)
               do (setf sum (+ sum (output link))))
          sum)
      0.0)))
```

Including this functionality gives a node the ability to sum the activations of the input links connected.

```
(descartes-class output-threshold-slot-widget
  (output-threshold-widget
   basic-slot-widget)
  ((output-threshold :descartes-init hard
                     :initform 0.0
                     :type float
                     :accessor output-threshold
                     :documentation "Node's output output-threshold activation value.")
   )
  (:documentation
    "Slot to hold node's output threshold."))

(defmethod default-initial-output-threshold
  ((self output-threshold-slot-widget)
   &key
   &allow-other-keys)
  ;; Returns the initial value of the node's output-threshold.
  *default-initial-output-threshold*)
```

Superclasses

This widget adds this slot to any class that includes it.

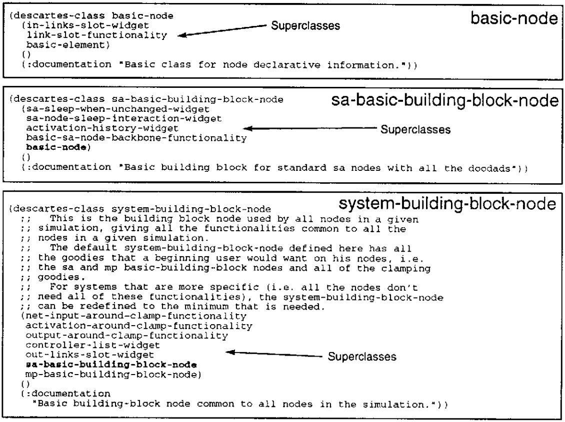
This method initializes the slot to a predetermined value.

Figure 51. Functionalities and Widgets Functionalities alter the neural signal as it passes through the various stages while widgets do not directly modify it.

standard CLOS class. Further primitives described below, such as nodes, have unique definition declarations.

B.2.2. *Nodes and Links*

The main elements of DESCARTES, the *nodes* and *links*, can be built from the functionalities and widgets. To define elements that will be instantiated, `(node-class` *`definition`*`)`, or `(link-class` *`definition`*`)` are used. The easiest way to first develop a DESCARTES network is to use the *sa-system-building-block-node* as the basis for the user's nodes because it contains many mixins operable under many different conditions. Any subclasses of system-building-block-node could be defined like `(node-class nifty-node (system-building-block-node)` *`rest-of-definition`*`)`. Figure 52 shows part of a hierarchy of this class. The relationship between functionalities and widgets and the different elements of DESCARTES are analogous to this example.



```
(descartes-class basic-node
  (in-links-slot-widget
   link-slot-functionality
   basic-element)
  ()
  (:documentation "Basic class for node declarative information."))
```

```
(descartes-class sa-basic-building-block-node
  (sa-sleep-when-unchanged-widget
   sa-node-sleep-interaction-widget
   activation-history-widget
   basic-sa-node-backbone-functionality
   basic-node)
  ()
  (:documentation "Basic building block for standard sa nodes with all the doodads"))
```

```
(descartes-class system-building-block-node
  ;;   This is the building block node used by all nodes in a given
  ;; simulation, giving all the functionalities common to all the
  ;; nodes in a given simulation.
  ;;   The default system-building-block-node defined here has all
  ;; the goodies that a beginning user would want on his nodes, i.e.
  ;; the sa and mp basic-building-block nodes and all of the clamping
  ;; goodies.
  ;;   For systems that are more specific (i.e. all the nodes don't
  ;; need all of these functionalities), the system-building-block-node
  ;; can be redefined to the minimum that is needed.
  (net-input-around-clamp-functionality
   activation-around-clamp-functionality
   output-around-clamp-functionality
   controller-list-widget
   out-links-slot-widget
   sa-basic-building-block-node
   mp-basic-building-block-node)
  ()
  (:documentation
    "Basic building-block node common to all nodes in the simulation."))
```

Figure 52. Hierarchy of Nodes. The superclasses of a node define the mixed in components that give the node its utility. Here we see that the *sa-basic-building-block-node* contains added widgets and functionalities along with its less endowed basis *basic-node*. From basic-node, other types of nodes can be created by combining it with other widgets and functionalities.

The nodes of DESCARTES have three stages of neural processing: (i) the net-input, (ii) the activation, and (iii) the output (McClelland & Rummelhart, 1988). At each stage, functionalities can be added to the node-class to alter the activity levels. Figure 51 demonstrates the basic net-input-functionality for nodes. With other mixins, this function can be further complicated. For instance, in Neurobot II, the *activation* of a node is modified by two sigmoidal functions (one is inverted) which constrain the range that the node activation resides (see Figure 32) Another sigmoid keeps the *output* of a Neurobot II node between the values of zero and one. Figure 53 provides a graphical breakdown of the three stages of the DESCARTES node.

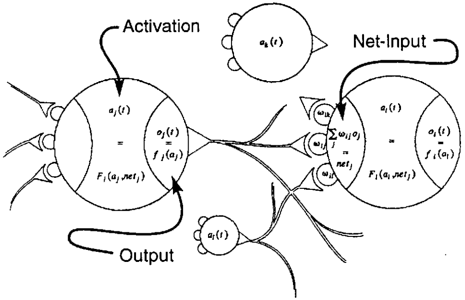


Figure 53. The Node. DESCARTES uses this three staged node as the basis for it's networks. Along with links, and CLOS extensibility, it becomes a powerful element for creating heterogeneous networks. The *net-input* receives and operates on all links, the *activation* is the internal node value, and the *output* is the axonal signal value which will be multiplied by the weight of the link before going to the net-input of another node. Parts of figure taken from *Parallel Distributed Processing* by D. Rummelhart, J. McClelland and the PDP Research Group, 1990.

Other node functionalities and widgets provided by DESCARTES give links global or specific biases, thresholds, multiplying constants, etc.

Links too, have been inspired from the PDP literature. Their basic purpose is to connect two nodes together (from the *source-node* to the *sink-node*). A DESCARTES link has a *weight* which is multiplied with the output of the source node to get the input to the sink node. Other functionalities and widgets allow the link to *conjunct* a set of links (multiply its value, rather than add), or shutdown a node completely, etc. Below, we describe the mechanisms which govern the growth and simulation (cycling) of these basic elements.

B.2.3. *Controllers*

Controllers are classes that preside over groups of DESCARTES elements including nodes, links, and other controllers. Thus, DESCARTES provides for nested controllers. For example, in Neurobot II, a large hierarchy is created. At the top is *world-control,* which contains *eye-control, arm-control, external-object-control,* and *brain-control* (see Section 4). These controllers are further broken down into smaller controllers such as brain-control, which contains the various PBA controllers for the different areas in the Neurobot II brain. Similarly, to define an instantiatable controller class, `(controller-class definition)` is used.

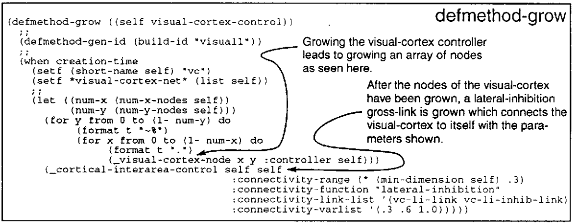
In DESCARTES, controllers are used in a rather simple way, basically to organize nodes. In the NSP, the potential of controllers is more fully utilized. For instance, while DESCARTES does not develop any controllers specifically designed to govern links, the gross-link of the NSP serves this function (see Section 3). Generalizing controllers even further, they became the building blocks for non-neural components also. For instance, the eye-control, and arm-control were completely neuronless. Connection to these elements through controllers made it possible to smoothly integrate the external world with the neural controllers when simulating. The nodes and links are organized in hierarchies of controllers so that DESCARTES can create and simulate networks as described in the next two sections.

B.2.4. Growth

In general, there are two stages in the simulation of a DESCARTES network. First, the network must first be grown, and second it must be cycled. While DESCARTES is perfectly capable of growth during cycling, this type of plasticity found in the brain has not been considered yet (Akoi & Siekevitz, 1988). In future versions of the NSP, it will be beneficial to look into this avenue for comparison studies with empirical data.

Growth is thus a major *conceptual* component of the DESCARTES nodes, links, and controllers. The difference between (1) growing a DESCARTES primitive and (2) using the standard CLOS instantiation to create one is that the user can define in a simple way, a growth method which performs many different functions. For instance, the world-control listed above grows the eye-control, arm-control, brain-control, and external-object-control (see Appendix E). Another example is a node growth method which when called, grows a node and then grows two links connected to more nodes that become its inhibitory nodes, which in turn then grow links to other nodes, etc. Further, the growth methods are intelligent, only instantiating a new element if it does not already exist. Thus, a chain of recursive node/link creations can occur with a growth methods that terminates when all nodes and links have been created.

The macro which makes the growth of nodes, links, and controllers possible is called *defmethod-grow*. It automatically generates a set of methods that are capable of performing the growth. Simply put, though, a DESCARTES element which has a growth method defined, for instance *visual-cortex-node*, would be grown with a method call: `(_visual-cortex-node 2 3)`, where this CLOS command will check to see if there exists a visual-cortex-node at position (2, 3), and if not, a *Vc-2x3* node is instantiated. An example of a defmethod-grow is shown in Figure 54 which shows the visual-cortex-control growth method for Neurobot II.



defmethod-grow

```
(defmethod-grow ((self visual-cortex-control))
  ;;
  (defmethod-gen-id (build-id "visual1"))
  ;;
  (when creation-time
    (setf (short-name self) "vc")
    (setf *visual-cortex-net* (list self))
    ;;
    (let ((num-x (num-x-nodes self))
          (num-y (num-y-nodes self)))
      (for y from 0 to (1- num-y) do
           (format t "~%")
           (for x from 0 to (1- num-x) do
                (format t ".")
                (_visual-cortex-node x y :controller self)))
      (_cortical-interarea-control self self
                                  :connectivity-range (* (min-dimension self) .3)
                                  :connectivity-function "lateral-inhibition"
                                  :connectivity-link-list '(vc-li-link vc-li-inhib-link)
                                  :connectivity-varlist '(.3 .6 1.0)))))
```

Growing the visual-cortex controller leads to growing an array of nodes as seen here.

After the nodes of the visual-cortex have been grown, a lateral-inhibition gross-link is grown which connects the visual-cortex to itself with the parameters shown.

Figure 54. The DESCARTES Growth Method. The command `(_visual-cortex-control)` would call this method to grow the visual-cortex. The basic function of this controller growth method is to grow a set of *visual-cortex-nodes* in an array and then connect the nodes in a lateral-inhibition pattern. The connection is performed by growing a gross-link called a *cortical-interarea-control*.

B.2.5. Cycling

Our final and most important stage of DESCARTES simulation is the cycling stage. After the nodes, links, and controllers are grown (created, initialized, linked), DESCARTES provides mechanisms for calculating net-inputs, activations, and outputs for nodes, spreading these values through links, and displaying any designated information on a periodic cycle. To control the order of cycling, DESCARTES provides a set of cycling parameters for its classes. The user defines the order of cycling, the beginning and ending of cycling, and when to display textual information. Figure 55 shows a simplified example of cycling.

This summarizes the capabilities and principal design aspects of DESCARTES. The freedom provided by CLOS allows DESCARTES to provide the user with a powerful set of tools for creating and simulating complicated neural networks. With the node, link, and controller basic elements and the widgets and functionalities, many

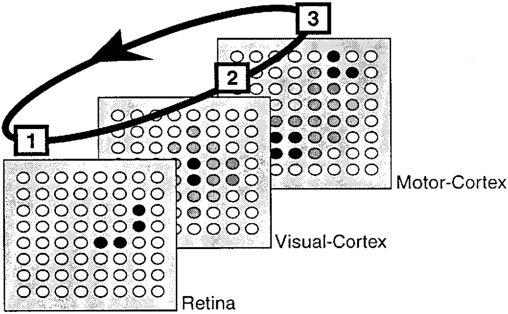


Figure 55. Cycling Through Controllers. Here is a simplified graphical description of cycling through Neurobot II. On a given cycle, only one area will have the nodes recalculate their internal values. Subsequently, the out-links of those nodes will also be recalculated, while the other sets of nodes do not get calculated, thus saving on processing. A more complicated example of cycling would demonstrate the integration of subnodes and nodes and the intricate timing requirements that occur.

types of connectivities and architectures can be designed: from completely homogeneous non-spatial, to topological and multilayered.

DESCARTES' versatility makes it easy to develop an entirely new layer of functionalities and widgets to provide the capabilities to simulate brain-areas, gross-links, and the external world. The new focus, and quantity of code necessary to shift DESCARTES' focus to this higher level, prompted us to develop separate package, the neurobotics simulation package (see Section 3).

APPENDIX C

NEUROBOT II SAMPLE TEXT OUTPUT

Below, we provide an example of the textual output generated from growing and early simulation of the Neurobot II model described in Section 4. The code provides an idea of the initialization, growth, and simulation of Neurobot II. The complete code for Neurobot II and the NSP is available in a technical report (Saxon, 1991).

```
dribbling to file "/user/jsaxon/thesis/data2/init-dribble"
 IMPORTANT VALUES:

(SETF *DEFAULT-ACTIVATION-HISTORY-CLASS* 'CIRCULAR-LIST-HISTORY): CIRCULAR-LIST-HISTORY
(SETF *DEFAULT-MAX-CIRCULAR-LIST-HISTORY-EVENTS* 5): 5
(SETF *DEFAULT-ONLY-INTERESTING-KEY* NIL): NIL
(SETF *DEFAULT-INITIAL-RETINA-NUM-X-NODES* 8): 8
(SETF *DEFAULT-INITIAL-RETINA-NUM-Y-NODES* 8): 8
(SETF *DEFAULT-INITIAL-NUM-X-NODES* 8): 8
(SETF *DEFAULT-INITIAL-NUM-Y-NODES* 8): 8
(SETF *DEFAULT-INITIAL-RETINA>VC-LINK* (CONS 0.3 0.7)): (0.3 . 0.7)
(SETF *DEFAULT-VC-LI-LINK-PRIORITY* 0.5): 0.5
(SETF *DEFAULT-MC-LI-LINK-PRIORITY* 0.5): 0.5
(SETF *DEFAULT-INITIAL-VC-LI-LINK-WEIGHT* 0.9): 0.9
(SETF *DEFAULT-INITIAL-VC-LI-INHIB-LINK-WEIGHT* 0.3): 0.3
(SETF *DEFAULT-INITIAL-ESTIMATED-CALIBRATION* 0.1): 0.1
(SETF *DEFAULT-AVERAGE-ACTIVITY-DAMPING* 0.2): 0.2
(SETF *DEFAULT-AVERAGING-THRESHOLD* 0.04): 0.04
(SETF *DEFAULT-MODIFYING-THRESHOLD* 0.1): 0.1
(SETF *DEFAULT-LINK-LEARN-RATE* 0.1): 0.1
(SETF *DEFAULT-LI-RANGE* 0.8): 0.8
(SETF *DEFAULT-VISUAL-CORTEX-TYPE* 2): 2
(SETF *DEFAULT-LI-RING-LIST* '(0.2 0.4 0.6 1)): (0.2 0.4 0.6 1)
(SETF *DEFAULT-LI-LINK-PROBABILITY-LIST* '(1.0 0.5 -0.8 -0.3)): (1.0 0.5 -0.8 -0.3)
(SETF *DEFAULT-R2VC-RANGE* 0.95): 0.95
(SETF *DEFAULT-R2VC-CONNECTIVITY* 2d-gaussian): 2d-gaussian
(SETF *DEFAULT-R2VC-LINK-LIST* '(RETINA>VC-LINK)): (RETINA>VC-LINK)
(SETF *DEFAULT-R2VC-VARLIST* '(8.0 1.0)): (8.0 1.0)

dribbling to file "/tmp_mnt/user/jsaxon/thesis/neurobotII/my-dribble"

; Loading /tmp_mnt/user/jsaxon/thesis/neurobotII/starter2.lisp.

Using an SGI? [y] y

; Loading /tmp_mnt/user/jsaxon/thesis/neurobotII/sgi.nb.

Initializing the Neurobot System...
cpu time (non-gc) 916 msec user, 217 msec system
cpu time (gc)     284 msec user, 0 msec system
cpu time (total)  1200 msec user, 217 msec system
real time  9445 msec

GENERATING WORLD COMPONENT: External...

GENERATING WORLD COMPONENT: Eye...

GENERATING: RetinalWarning: assuming lisp store in (:MOVE (:REG 2) (:IREF (:REG
```

```
0) (:REG 1)))

........
........
........
........
........
........
........
........

GENERATING WORLD COMPONENT: Arm...

GENERATING: Arm-Sender
..
..

GENERATING: Visual1
........
........
........
........
........
........
........
........

Performing gc...

GENERATING GROSS CONNECTION:  Visual1>>>Visual1
     Connectivity Function:        lateral-inhibition
     Connectivity Range:           2.4
     Connectivity Link Class List: (VC-LI-LINK VC-LI-INHIB-LINK)
     Connectivity Variables:       (0.3 0.6 1.0)
........
........
........
........
........
........
........
........

Performing gc...

GENERATING GROSS CONNECTION:  Retina1>>>Visual1
     Connectivity Function:        2d-gaussian
     Connectivity Range:           2.4
     Connectivity Link Class List: (RETINA>VC-LINK)
     Connectivity Variables:       (2.0 1.0)
........
........
........
........
........
........
........
........

GENERATING: Motor1
........
........
```

```
........
........
........
........
........
........

Performing gc...

GENERATING GROSS CONNECTION:  Motor1>>>Motor1
     Connectivity Function:       lateral-inhibition
     Connectivity Range:          4.0
     Connectivity Link Class List: (MC-LI-LINK MC-LI-INHIB-LINK)
     Connectivity Variables:       (0.3 0.6 1.0)
........
........
........
........
........
........
........
........

Performing gc...

GENERATING GROSS CONNECTION:  Visual1>>>Motor1
     Connectivity Function:       2d-gaussian
     Connectivity Range:          4.0
     Connectivity Link Class List: (VC>MC-LINK)
     Connectivity Variables:       (5.0 0.5)
........
........
........
........
........
........
........
........

Performing gc...

GENERATING GROSS CONNECTION:  Motor1>>>Visual1
     Connectivity Function:       2d-gaussian
     Connectivity Range:          4.0
     Connectivity Link Class List: (MC>VC-LINK)
     Connectivity Variables:       (5.0 0.5)
........
........
........
........
........
........
........
........

Performing gc...

GENERATING GROSS CONNECTION:  Motor1>>>Arm-Sender
     Connectivity Function:       segmented
     Connectivity Range:          4
     Connectivity Link Class List: (MC>ARM-SENDER-LINK)
     Connectivity Variables:       (T 2)
```

```
..
..

GENERATING: Motor1-Oscillator
..
..

Performing gc...

GENERATING GROSS CONNECTION:  Motor1-Oscillator>>>Motor1
     Connectivity Function:        segmented
     Connectivity Range:           4
     Connectivity Link Class List: (OSCILLATOR-LINK)
     Connectivity Variables:       (NIL 4)
..
..
cpu time (non-gc) 257233 msec user, 2667 msec system
cpu time (gc)     127367 msec user, 2550 msec system
cpu time (total)  384600 msec user, 5217 msec system
real time  480878 msec
Meta Control: "*Meta-Control*", class BRAIN-META-CONTROL
CONTROLLER-LIST..........: 18 elements: ("Motor1-Oscillatortomotor1" ...)
NODE-TABLE...............: 328 elements: ("R-6x7" "Mc-2x6" "Vc-6x0-Inhib" "Mc-2x7" ...)
LINK-TABLE...............: 644 elements: ("Mto-L90" "Mto-L91" "Mto-L94" "Mto-L95" ...)
CONTROLLER-TABLE.........: 18 elements: ("Brain" "Visual1tovisual1" ...)
GEN-ID-HEADER-TABLE......: 2 hash elements: (("Mto-L" 516) ("L" 128))
HARD-INITIAL-VALUES-TABLE.: 9 hash elements: ( ...)
SOFT-INITIAL-VALUES-TABLE.: NIL
HIGHEST-CYCLE-NUMBER.....: 0
CYCLE-NUMBER.............: 0
DEFAULT-CONTROLLERS......: 1 elements: ("World")
CURRENT-CONTROL..........: "Motor1-Oscillatortomotor1"
SA-ACTIVE-CONTROLLERS....: NIL
; Loading /tmp_mnt/user/jsaxon/thesis/neurobotII/cycling.lisp.

Saving links: /user/jsaxon/thesis/data2/vc2vc-8x8n8x8#00000
........
Saving links: /user/jsaxon/thesis/data2/r2vc-8x8n8x8#00000
........Control: "Visual1", class VISUAL-CORTEX-CONTROL
LINK-TABLE.............: 192 elements: ("Mto-L98" "Mto-L99" "Mto-L96" "Mto-L97" ...)
NODE-TABLE.............: 64 elements: ("Vc-5x3" "Vc-5x4" "Vc-5x1" "Vc-4x7" ...)
DISPLAY-STATUS-CYCLES...: 1 elements: (0)
DISPLAY-EVERY..........: 4
STATUS-EVERY...........: 8
OBJECTS-TO-DISPLAY.....: NIL
LAST-SA-UNSTABLE-P-NODE.: NIL
SA-CYCLE-HISTORY.......: 1 elements: ((0))
SA-TRACE...............: NIL
SA-STABLE-DIFFERENCE....: 2.0e-5
SA-CYCLE-END...........: NIL
SA-CYCLE-EVERY.........: 2
SA-CYCLE-START.........: 2
SA-JUST-SLEPT-LINKS-CDR.: 1 elements: (NIL)
SA-WAKING-LINKS-CDR.....: 1 elements: (NIL)
SA-AWAKE-LINKS-CDR......: 1 elements: (NIL)
SA-JUST-SLEPT-NODES-CDR.: 1 elements: (NIL)
SA-WAKING-NODES-CDR.....: 65 elements: (NIL "Vc-7x7 O: 0.072" "Vc-6x7 O: 0.089"...)
SA-AWAKE-NODES-CDR......: 1 elements: (NIL)
SHORT-NAME.............: "vc"
WINDOW.................: #<window stream 'Visual1-Act: [   0.02,    0.20]' @ #xc4508e>
DRAW-HISTORY...........: 1 elements: ((0))
DRAW-END...............: NIL
```

```
DRAW-EVERY.............: 2
DRAW-START.............: 4
NODE-STAGE-TO-DRAW......: ACTIVATION
NODE-SHAPE.............: CIRCLE
SCRUTINIZE-CONS-LIST....: 1 elements: (("Vc-4x4 O: 0.063" . "Retinaltovisual1"))
EXCITATORY-SCALE-FACTOR.: 0.53601044
ESTIMATED-CALIBRATION...: 0.1
SINK-AREAS.............: 2 elements: ("Motor1" "Visual1")
SOURCE-AREAS...........: 3 elements: ("Motor1" "Retinal" "Visual1")
NUM-Y-NODES............: 8
NUM-X-NODES............: 8
OUTPUT-EFFECT-HISTORY...: 1 elements: ((0))
OUTPUT-EFFECT-END.......: NIL
OUTPUT-EFFECT-EVERY.....: 2
OUTPUT-EFFECT-START.....: 4
STAGES-TO-SAVE.........:   4   elements:   (NET-INPUT   ACTIVATION   OUTPUT   INHIBITION)Control:
"Retinaltovisual1", class CORTICAL-INTERAREA-CONTROL
LINK-TABLE............: 64 elements: ("Mto-L180" "Mto-L181" "Mto-L182" "Mto-L187" ...)
DISPLAY-STATUS-CYCLES..: 1 elements: (0)
DISPLAY-EVERY..........: NIL
STATUS-EVERY...........: NIL
SHORT-NAME............: "r2vc"
LEARN-CYCLE-HISTORY....: 1 elements: ((0))
LEARN-CYCLE-END........: NIL
LEARN-CYCLE-EVERY......: NIL
LEARN-CYCLE-START......: NIL
LEARNING-SHUTOFF.......: NIL
OUTPUT-EFFECT-HISTORY..: 1 elements: ((0))
OUTPUT-EFFECT-END......: NIL
OUTPUT-EFFECT-EVERY....: NIL
OUTPUT-EFFECT-START....: NIL
SINK-AREAS............: 1 elements: ("Visual1")
SOURCE-AREAS..........: 1 elements: ("Retinal")
CONNECTIVITY-VARLIST...: 2 elements: (2.0 1.0)
CONNECTIVITY-LINK-LIST.: 1 elements: (RETINA>VC-LINK)
CONNECTIVITY-FUNCTION..: "2d-gaussian"
CONNECTIVITY-RANGE.....: 2.4
Control: "Visual1tovisual1", class CORTICAL-INTERAREA-CONTROL
LINK-TABLE............: 128 elements: ("Mto-L98" "Mto-L99" "Mto-L96" "Mto-L97"...)
DISPLAY-STATUS-CYCLES..: 1 elements: (0)
DISPLAY-EVERY..........: NIL
STATUS-EVERY...........: NIL
SHORT-NAME............: "vc2vc"
LEARN-CYCLE-HISTORY....: 1 elements: ((0))
LEARN-CYCLE-END........: NIL
LEARN-CYCLE-EVERY......: 2
LEARN-CYCLE-START......: 20
LEARNING-SHUTOFF.......: NIL
OUTPUT-EFFECT-HISTORY..: 1 elements: ((0))
OUTPUT-EFFECT-END......: NIL
OUTPUT-EFFECT-EVERY....: 64
OUTPUT-EFFECT-START....: 16
SINK-AREAS............: 1 elements: ("Visual1")
SOURCE-AREAS..........: 1 elements: ("Visual1")
CONNECTIVITY-VARLIST...: 3 elements: (0.3 0.6 1.0)
CONNECTIVITY-LINK-LIST.: 2 elements: (VC-LI-LINK VC-LI-INHIB-LINK)
CONNECTIVITY-FUNCTION..: "lateral-inhibition"
CONNECTIVITY-RANGE.....: 2.4
Cycle: 0.1.

2.
Run the simulation? [y] y
```

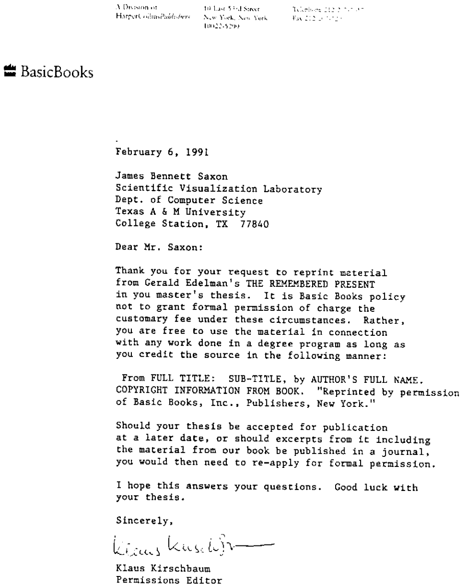
```
2.

3.

4.

5.

6.

7.


Vc-4x4 ->  Net[6.589] Act[0.413] Avg[1.000] Out[0.755] and Inh[0.014]
  RETINA>VC-LINK:
 0.3315  0.5704  0.3316  0.5418  0.5618  0.6709  0.5520  0.3156  0.4550  0.3802 0.6308  0.4477
 0.6425  0.6969  0.3031  0.3835  0.5167  0.5457  0.4194  0.4151  0.6455  0.5274 0.6097  0.6296
 0.5727  0.6274
8.

9.

10.

11.

12.
Saving links: /user/jsaxon/thesis/data2/r2vc-8x8n8x8#00012
........Cycle: 12.

13.

14.


Saving links: /user/jsaxon/thesis/data2/vc2vc-8x8n8x8#00015
........15.


Vc-4x4 ->  Net[6.368] Act[0.134] Avg[1.000] Out[0.160] and Inh[0.024]
  RETINA>VC-LINK:
 0.3315  0.5704  0.3316  0.5418  0.5618  0.6709  0.5520  0.3156  0.4550  0.3802 0.6308  0.4477
 0.6425  0.6969  0.3031  0.3835  0.5167  0.5457  0.4194  0.4151  0.6455  0.5274 0.6097  0.6296
 0.5727  0.3000
16.

17.

18.

19.

20.

21.

22.
Saving links: /user/jsaxon/thesis/data2/r2vc-8x8n8x8#00022
........Cycle: 22.

23.
```

```
Vc-4x4 ->  Net[6.302] Act[0.185] Avg[0.600] Out[0.241] and Inh[0.028]
  RETINA>VC-LINK:
 0.3315  0.5704  0.3316  0.5418  0.5618  0.6709  0.5520  0.3156  0.4550  0.3802 0.6308  0.4477
 0.6425  0.6969  0.3031  0.3835  0.5167  0.5457  0.4194  0.4151  0.6455  0.5274 0.6097  0.6296
 0.5727  0.6274
24.

25.
```

etc...

APPENDIX D

PERMISSIONS



A Division of
HarperCollins*Publishers*

10 East 53rd Street
New York, New York
10022-5299

Telephone 212 [illegible]
Fax 212 [illegible]

BasicBooks

February 6, 1991

James Bennett Saxon
Scientific Visualization Laboratory
Dept. of Computer Science
Texas A & M University
College Station, TX 77840

Dear Mr. Saxon:

Thank you for your request to reprint material from Gerald Edelman's THE REMEMBERED PRESENT in you master's thesis. It is Basic Books policy not to grant formal permission of charge the customary fee under these circumstances. Rather, you are free to use the material in connection with any work done in a degree program as long as you credit the source in the following manner:

From FULL TITLE: SUB-TITLE, by AUTHOR'S FULL NAME. COPYRIGHT INFORMATION FROM BOOK. "Reprinted by permission of Basic Books, Inc., Publishers, New York."

Should your thesis be accepted for publication at a later date, or should excerpts from it including the material from our book be published in a journal, you would then need to re-apply for formal permission.

I hope this answers your questions. Good luck with your thesis.

Sincerely,

Klaus Kirschbaum

Klaus Kirschbaum
Permissions Editor

VITA

James Bennett Saxon was born on 11 October 1964 in Los Angeles, California. Pleased with his accommodations, he has spent much of his life there pursuing a diverse set of interests. Some of these interest include building pretty little wooden cars that would win awards at the pinewood derby, programming his Apple II computer, marching in a band while holding a clarinet, getting philosophical, skiing, singing, dancing, and eating sushi.

After high school, he found himself basically comfortable going to the University of California at Irvine, where, while the orange curtain of conservatism dampened many student's fervor for silliness, he still managed to skateboard around the campus getting more philosophical, taking dance classes, taking acting classes, taking tons of electrical engineering and computer science classes, and enjoying himself.

In June 1987, after receiving two bachelors degrees in Information and Computer Science and Electrical Engineering in five years , he joined the Systems Integration and Test group at Rockwell International's Space and Satellite Electronics Division. While the employees were friendly, at some point during that year, in a flash, he suddenly realized that he was neither comfortable nor happy anymore and immediately moved to correct the situation by applying to graduate schools all over the country.

As fate would have it, his decision landed him smack in the middle of a place known to some Californians as "culturally opposite". While Zaire and a couple of other countries may also have taken this role, Texas A&M University had a better Computer Science department. Thus he found himself in Bryan/College Station, Texas studying under a Regent's Fellowship at TAMU with a newly acquired understanding that he has some measure of control of his destiny and wishes to pursue the questions he has always had in the back of his mind. These questions are beginning to be answered by his present work on this thesis.

James Saxon can be reached through Bruce H. McCormick at the Scientific Visualization Laboratory, Department of Computer Science, Texas A&M University, College Station, TX 77840-3112.