Self-organization via Active Exploration in Robotic Applications Phase II: Hybrid Hardware Prototype

Final Progress Report

Haluk Öğmen, Principal Investigator
Ramkrishna V. Prakash, Research Assistant
Department of Electrical Engineering
University of Houston
Houston, TX 77204-4793
December, 1993

Contents

1 Introduction

In many environments human-like intelligent behavior is required from robots to assist and/or replace human operators. The purpose of these robots is to reduce human time and effort in various tasks. Thus, the robot should be robust and as autonomous as possible in order to eliminate or to keep to a strict minimum its maintenance and external control. If the robot requires more human intervention than the task it accomplishes then it would be useless for most of the applications. Moreover, if the robot has to function in an uncontrolled environment where unpredictable changes can occur, and if its maintenance is kept to a strict minimum then the design requirements become more complex. In particular, direct program control or model based traditional approaches to robotic problems prove to be inadequate because they cannot cope with such uncontrolled environments. Then, what are the key issues of the design problem ? The analysis of the requirements outlined above leads to the following properties:

(i) *Fault tolerance*: This property can be achieved by use of an adequately organized distributed architecture incorporating some redundance. Fault-tolerance will let the robot to maintain an acceptable performance immediately after the occurrence of faults in the hardware or changes in its structural parameters (e.g. a change in the arm joint parameter due to mechanical fatigue).

(ii) *Self organization* (which augments fault-tolerance by completely correcting the performance) detects and analyses faults or external changes and consequently achieves the correct performance under these new conditions.

(iii) *Intelligence* is necessary to achieve the understanding required by the self-organization process and also to analyze the environment and to predict future events. Moreover, intelligence is also necessary to establish a natural communication (e.g. language) between humans and the robot.

But how can these properties be implemented in a robot ? These properties are drastically different from the ones widely used in traditional design and require a careful analysis of the underlying phenomena. A good insight can be gained by considering principles found in studies directed toward "systems" that posses all these qualities: the human ! Unlike many primitive animals which are almost completely genetically wired, human infants undergo an extensive developmental period during which they learn to control and coordinate various parts of their body. Moreover, they actively explore the environment to transform simple instincts to habits and to operational structures using novelty and complex associations which result from the interaction with the environment. It is important to emphasize how this exploratory activity is fault tolerant and self-organizing: The growing child's physical characteristics continually change (the arms become longer etc.). If the control were based on a strict model, it would

fail to function as soon as the child grows a little because none of the parameters would be the same. There is a large number of studies that outline various principles regulating this developmental stage as well as its relationship with the adult performance (e.g. Piaget 1963, 1967, 1969, 1970). These classical studies show how self-organization and intelligence emerge from active exploration. It demonstrates subtle issues underlying the transformation of instincts to habits and to operational structures. The exploratory activity requires a careful combination of internal drives and environmental cues. Until recently, these findings were limited to the realm of psychology. However, neural network theory developed tools that enable us to implement these findings for technological problems. In Phase I of this project, we developed such a neural network architecture. It captures some fundamental aspects of human categorization, habit, novelty, and reinforcement behavior. The model, called FRONTAL (in reference to the frontal lobes), is a "cognitive unit" regulating the exploratory behavior of the robot.

In the second phase of the project, we interfaced FRONTAL with off-the-shelf robotic arm and a real-time vision system. The components of this robotic system, a review of FRONTAL, and simulation studies are presented in this report.

2 The Robot

The self-organizing robotic system is shown in Figure 1. It comprises of the following four parts:

- the vision system
- the arm system
- the neural network (FRONTAL) and
- the communication protocols.

The vision system enables the robot to see its surroundings, while the arm system allows it to interact with the objects present in its field of view. The neural network FRONTAL which is the "cognitive controller" of the robot enables it to actively explore its surrounding and to adapt its behavior to changes in the environment. The vision system and the arm system of the robot communicates with FRONTAL via communication protocols. During the initial stage of the development of this robotic system a simple communication protocol using DARPA Internet protocol suite (TCP/IP) sockets was used. In the later stage this communication protocol was replaced by a more versatile protocol developed using Telerobotics interconnection Protocol (TELRIP).

The vision system comprises of a real-time image processing system called the MaxVideo 20 manufactured by DATACUBE, a grey scale camera and an object recognition software called

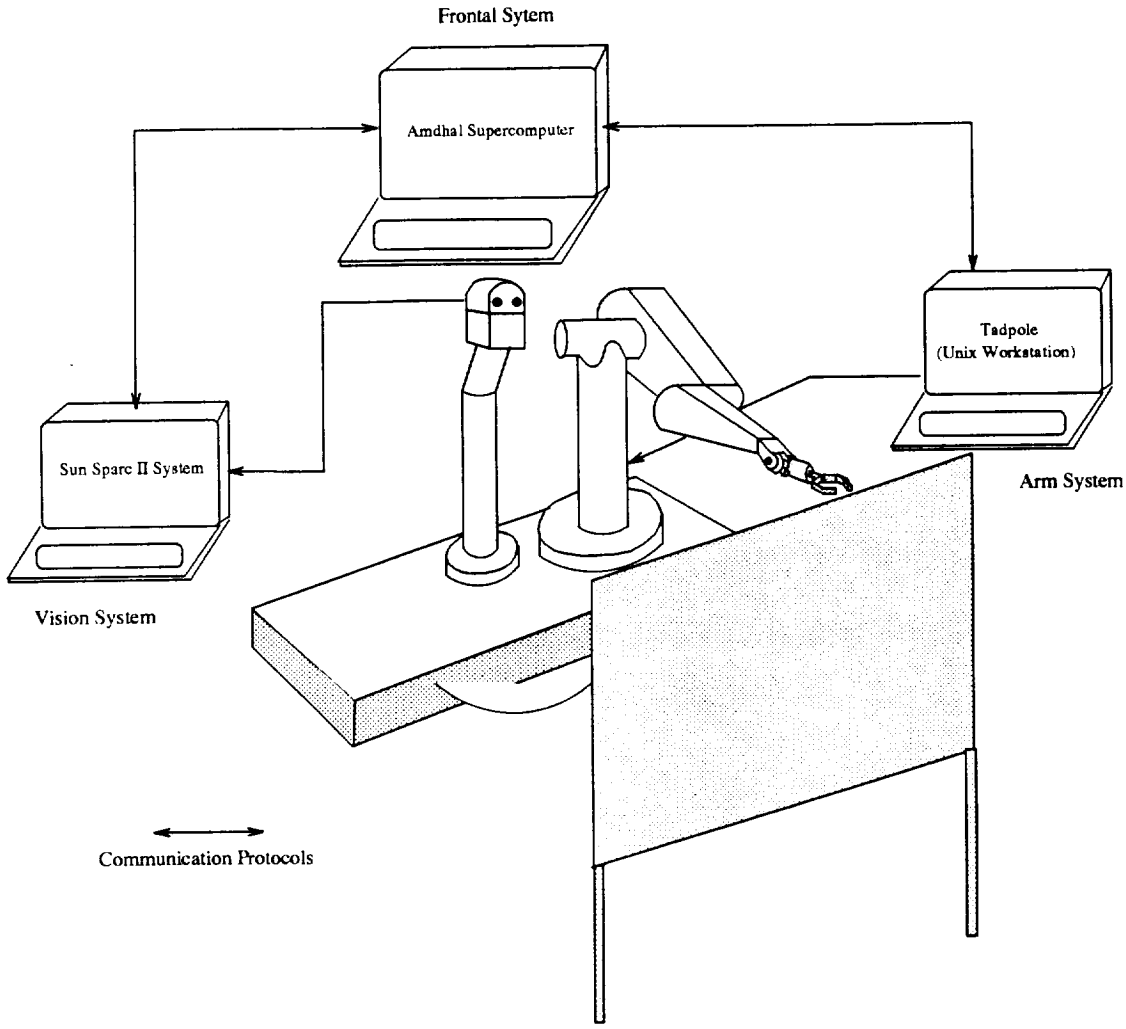

Figure 1: The self-organizing robotic system. It comprises of the following four systems: (i) the vision system (ii) the arm system (iii) the neural network based "cognitive controller" called FRONTAL and finally (iv) the communication protocols. The vision system consists of a camera which, in conjunction with the MaxVideo image processing system and object detection software, yields a real-time image processing system, capable of detecting objects in the robot's environment. The arm system consists of a PUMA 562 arm and associated software to calculate its inverse kinematics. The neural network based "cognitive controller", called the FRONTAL, is responsible for generating and co-ordinating purposeful behaviors for the robot. The various components of the robotic system communicate with each other via communication protocols developed using TCP/IP sockets. These were later replaced by an unified protocol developed using TELRIP

BLOBS. The MaxVideo 20 system is mounted on a VME cache and it communicates with a Sun Sparc II via a VME bus. The vision system was programmed to threshold the input from the camera so as to isolate objects from their background, thus accomplishing figure-ground segregation. This thresholded image constitutes the input to the robotic system. To facilitate the simultaneous viewing of the robot's environment and its inputs, the image processing system was programmed to toggle between two modes every other clock cycle. In the first mode, the raw image is captured and sent directly to the video monitor. In the second mode, the raw image is thresholded and simultaneously sent to the video monitor as well as the object recognition software BLOBS. The thresholded as well as the non-thresholded frames were simultaneously displayed on a video monitor by splitting the screen into two parts. Thus, one could monitor the input to and the output from the vision system "simultaneously". This reduced sampling of the environment was much faster than any dynamic changes that were induced in the robots environment.

The filtered images were then processed continuously by a software running on a Sun Sparc II system which generated a symbolic representation of the object's features and its location. The details of the vision system and an evaluation of its performance is presented in the later Subsection 2.1 and Section 5.

The robotic arm system [1] consisting of two PUMA 562 robot arms. A three digit Stanford/JPL dexterous hand is attached on the right arm. The left arm has a two digit gripper. In our current implementation, only the left arm with the gripper is used to interact with the environment. The PUMA 562 arms are being controlled by an Unix workstation which communicates with the arm controller via a VME bus. The controller for the robotic arms is built by Cybernetics Inc. It allows one to control as well as monitor every joint angle of each arm. Moreover, torque sensors positioned at various joints yield a measure of the force exerted at the joints. Subroutines have been written to facilitate an easy control of this robotic arm system, details of which will be presented later.

So far we have discussed the sensory and the motor systems of this anthropomorphic robotic system. Hence a brief discussion of the "cognitive controller" of this robot seems warranted. A neural network called FRONTAL (Ogmen and Prakash, 1991) controls the vision system and the arm system. This neural network is capable of identifying and selectively attending to novel as well as rewarding objects in its environment. At the same time, it is capable of actively reorganizing its behavior depending on the external reinforcement signals. This neural network is implemented on an Amdahl supercomputer and it communicates with the vision system as well as the arm system using communication protocols.

[1] This robotic system called Dexterous Anthropomorphic Robotic Testbed, (DART), is being developed by the Robotics and Automation Division of NASA, JSC.

In the following sections a more detailed descriptions of the vision, arm, FRONTAL and the communication protocol are presented.

2.1 The Vision System

The primary component of the vision sietup consists of a real-time image processing system called the DataCube MaxVideo System 20 (MaxVideo 20). The MaxVideo system comprises of various specialized hardware modules (called the MaxVideo modules) which are connected to each other via a MAXbus. This image processing system can be programmed by a host computer using an object-oriented based software called ImageFlow, which is resident on the host computer. The communication between the image processing system and the host computer is via a VME bus and thus any computer system capable of VME bus based I/O (input/output) communication can be used to control this image processing system. In the current configuration, the MaxVideo system is interfaced with a Sun Sparc II system. Figure 2 shows an overview of the setup of the vision system. The MaxVideo modules along with the MAXbus provides a 10MHz (103 nsec/pixel) synchronous pipelined DSP (digital signal processing) engine which is capable of acquiring, processing and displaying images at rate of 30 frames/sec [2]. This system is capable of acquiring images in any one of the following variety of input data precision: (i) 8/12 bit analog RS-170/CCIR (standard television), (ii) 8/12 bit asynchronous analog, (iii) 8/16 bit digital, (iv) 24-bit NTSC (video), RGB, YIQ or (v) 36-bit RGB RS-170/CCIR format. It can process these raw images with 8 or 16 bit precision [3] and store them with either 8, 16, 24 or 64 bit precision. Displaying of the processed images can be done in one of the following data precision forms: (i) 8-bit RS-170/CCIR B/W or pseudocolor, (ii) 8-bit High Resolution B/W or pseudocolor, (iii) 24-bit NTSC or RGB or YIQ or (iv) 24-bit High Resolution RGB. The MaxVideo system also provides means to add an 8-bit graphics overlay image along with processed the image being displayed.

The MaxVideo system consists of the following five modules:

- Analog Scanner (AS)
- Architectural Adapter (AA)
- Analog Generator (AG)
- Advanced Pipeline Processor (AP)

[2] This speed is for a standard 512 × 484 pixel image. The MaxVideo system is capable of processing high resolution images (4096 × 4096 pixels) but at a slower rate. However the displayable resolution of the system is only 1024 × 1024 pixels.

[3] The AU MaxVideo module is however capable of processing with 20 bit precision.

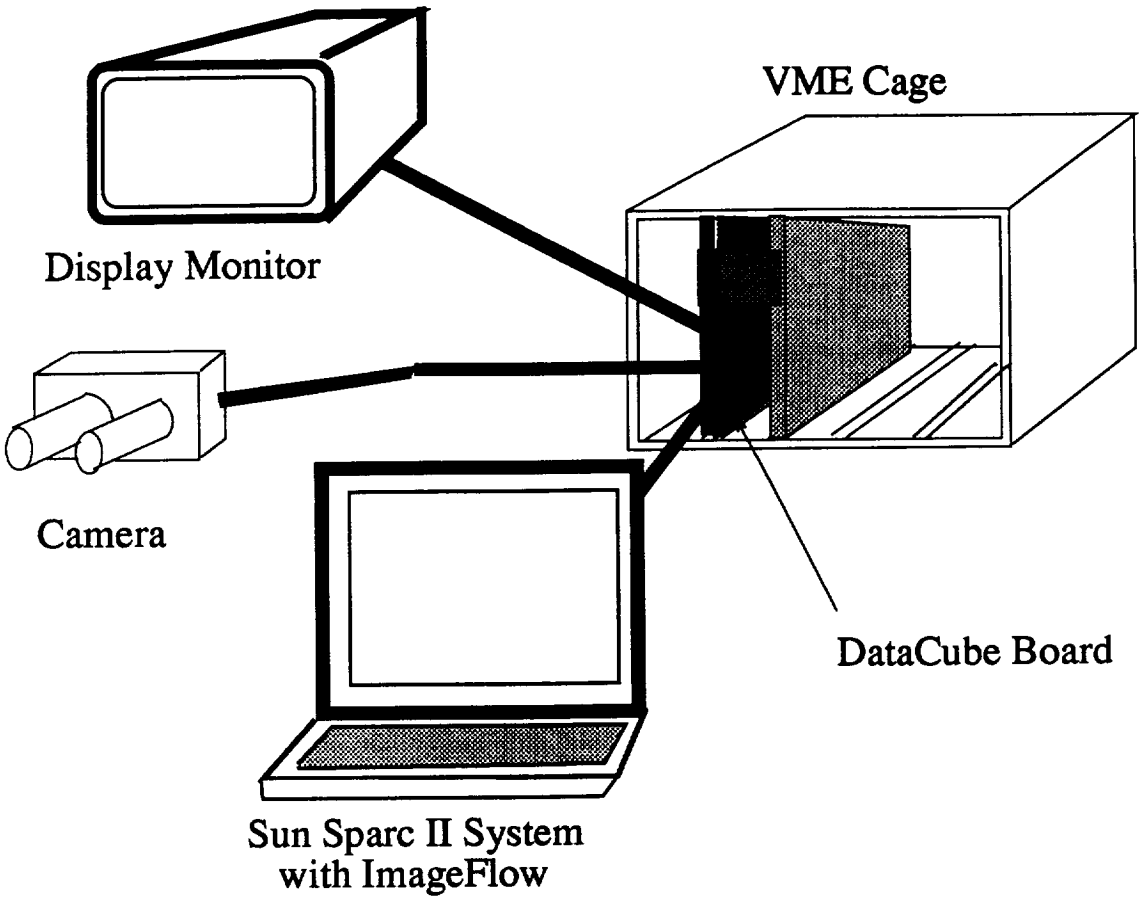

Figure 2: This figures shows a block diagram of the various components of the vision system. The MaxVideo 20 system board is placed in a VME cage which is accessible to the Sun Sparc 2 computer. The ImageFlow software which is object-oriented control software for the MaxVideo system is resident in the Sun Sparc 2 computer. Each processed frame by the MaxVideo system is displayed on a Sony monitor and simultaneously grabbed by the Sun computer to perform object recognition. The Sun Sparc 2 is in communication with Amdahl via a sockets based communication protocol.

- Arithmetic Unit (AU)

A brief overview of the various modules and their capabilities is presented in Appendix B. In the current implementation of the vision system, only the AP moudle is used in conjunction with the AS, AA and AG modules. Figure 3 shows the overall setup of the MaxVideo 20 system for thresholding images in real time. As can be seen from the figure, the MaxVideo system is configured in two different modes (pathways) called PATs. The AS module receives a multisync signal from the CCD camera and routes it alternatively through these two PATs. The first PAT goes directly from the AS module through the AA module and the AG module to the monitor. When the MaxVideo system runs in this configuration it displays the captured raw image directly on the monitor. The second PAT is from the AS through the AA via the AP and finally through the AS to the monitor. The AP module is configured to threshold the raw image by using a generic look-up table. In this configuration the MaxVideo system, thresholds the image that is captured by the CCD camera. The two paths are toggled every other clock cycle and they are alternatively displayed on an external monitor. While the MaxVideo system is in the second mode, the threshold image is read into the host computer. The threshold image is then processed via software to locate the different objects in the image. Once the objects have been detected, their location and type are identified and the information is transmitted to FRONTAL over the communication protocol.

The software used for object detection consists of a blob detection algorithm called BLOBS [4]. BLOBS groups neighboring pixels of similar color as belonging to a single bolb (or object). It also assigns pixels of a blob having only 3 neighbors as the edge pixels of the object. The area and the perimeter of the objects are detected by using the total number of pixels and the number of edge pixels respectively of the object. Three different kinds of objects (equilateral triangle, circle, stars) were required to be detected by BLOBS. A compactness measure given by,

$$compactness = \frac{perimeter^2}{Area}, \tag{1}$$

is used to differentiate amongst the three different kind of objects. An advantage of this measure is that it is a rotational invariant measure. The image grabbed by the CCD camera however is distorted because of the aspect ratio of the pixels of the camera (which is the $\frac{height}{width}$ ratio of a pixel and is equal to 0.75). This leads to an inconsistent measure of the area and the perimeter of the objects as they are rotated about an axis perpendicular to the camera. To avoid this problem BLOBS scales each pixel to take of the aspect ratio problem. Nine different types of objects consisting of three different sizes (small, medium large) and shapes (circle, equilateral triangle, star) (which were latter used for the simulations) could be reliably detected by the vision system.

[4]This software was developed in conjunction with Jeff Kowing and Bob Goode of NASA, JSC.

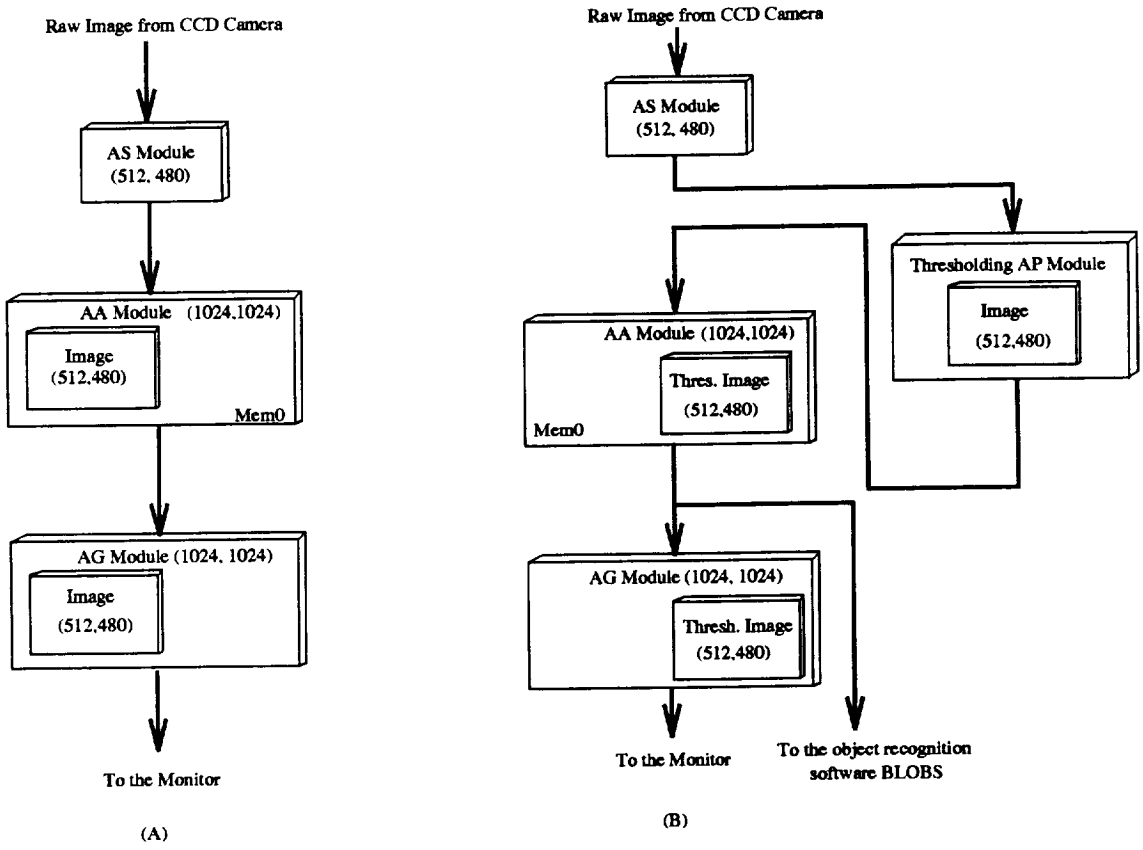

Figure 3: The two modes in which the MaxVideo system is operated are shown. (A) Shows the first mode in which the MV20 system is configured to send the raw image captured by the CCD camera directly to monitor. In the second mode the raw image is first thresholded by the AP module and then sent to be displayed on the monitor. Simultaneously the thresholded image is also sent to BLOBS which is an object recognition software. The MaxVideo 20 System is toggled between the two modes every other system clock cycle. As the system clock rates are about 1000 times faster than the dynamic changes in the environment, the processing of every other image does not effect the performance of the robotic system.

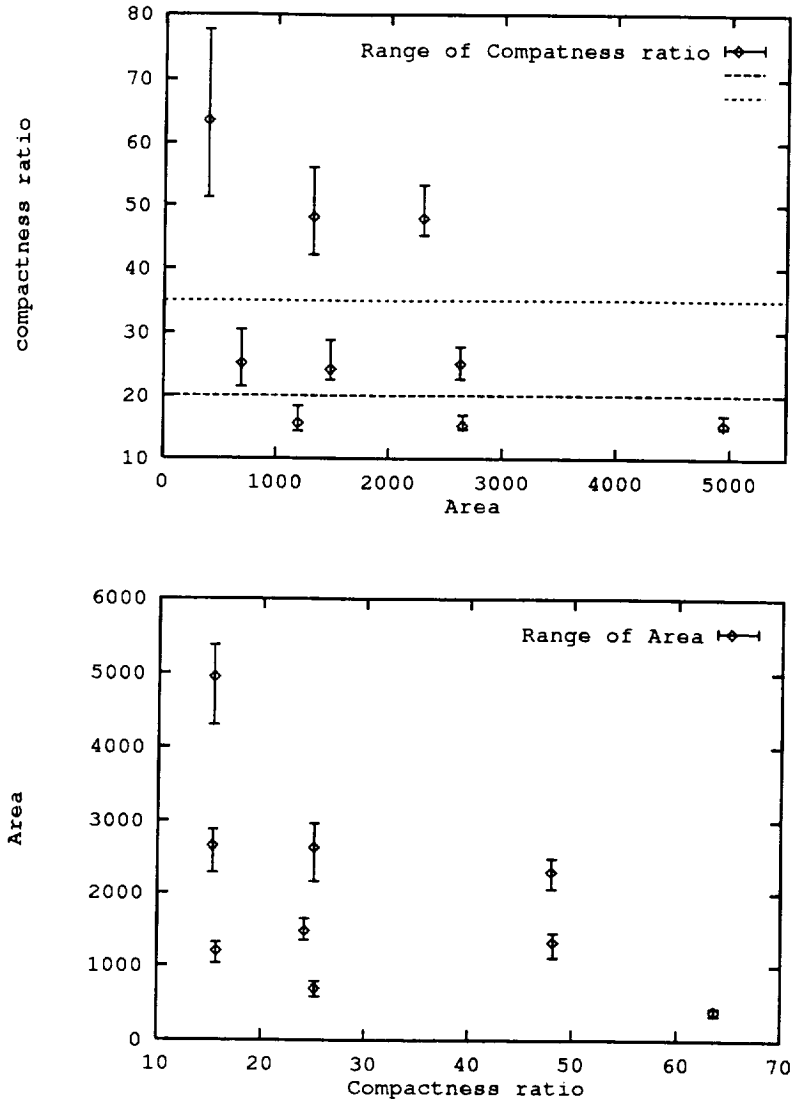

Figure 4: The above two plots summarize the performance of the vision system. The top graph plots the compactness ratio of the nine different objects against their area. Twenty objects of each of the nine kinds were presented to the vision system. The circles represent the average value of the compactness ratio for each of the objects and the bars represent the range. The objects could be grouped primarily into three types depending on their compactness ratio as indicated by the horizontal lines. The second graph is a similar plot but showing the range in the area of the nine objects. As can be seen, for a given object type (compactness ratio) the object sizes do not overlap.

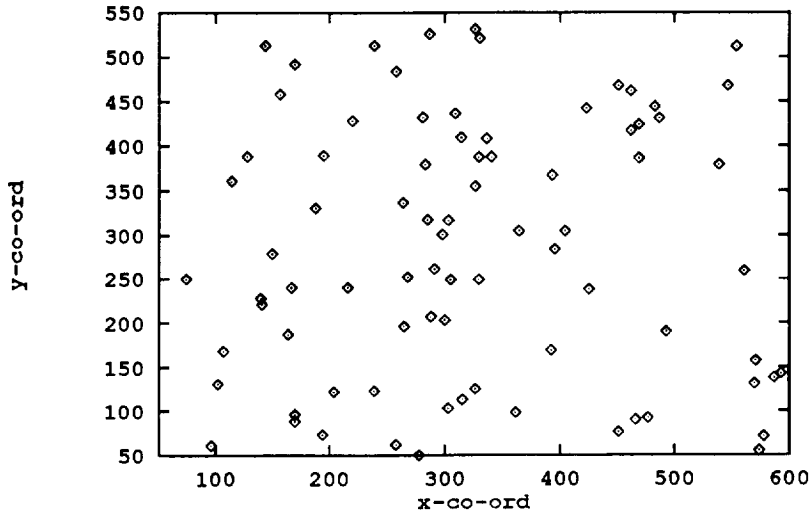

Figure 5: The the various locations in which the objects that are used for studying the performance of the vision system are placed.

Figure 4 gives a summary of the performance of the vision system. The vision system was tested twenty times for each type of object. The average compactness ratio as well as the ranges of the the three types of objects are show at the top in Figure 4. As can be seen, there is no overlap between the three types of objects. The bottom graph in Figure 4 plots the mean area and their range for all the nine objects. For any given type of object there was no overlap between the sizes. Figure 5 shows a scatter plot of the various locations in the visual field at which the objects were placed for testing the vision system's performance.

2.2 The Arm System

DART (Dexterous Anthropomorphic Robotic Testbed) is a robot developed by Automation and Robotics Division at NASA. JSC. This robot shown in Figure 6 is built with an anthropomorphic design in mind. It consists of two PUMA 562 arms, a Standford/JPL dexterous hand on the right and a gripper on the left hand. The two PUMA arms rest on a base that is controlled by a motor to enable the robot rotate around its central axis (shown by x–x' in Figure 6).

Each of the two PUMA 562 arms has 6 degrees of freedom as shown Figure 6. A Cybernetics servo controller consisting of three Central Processing Units (CPU's) controls the joints of each arm. The control of the joints are accomplished by using Position Derivative (PD) based servo-loops. Position, velocity and torque control of the arm can be achieved via the controller. The

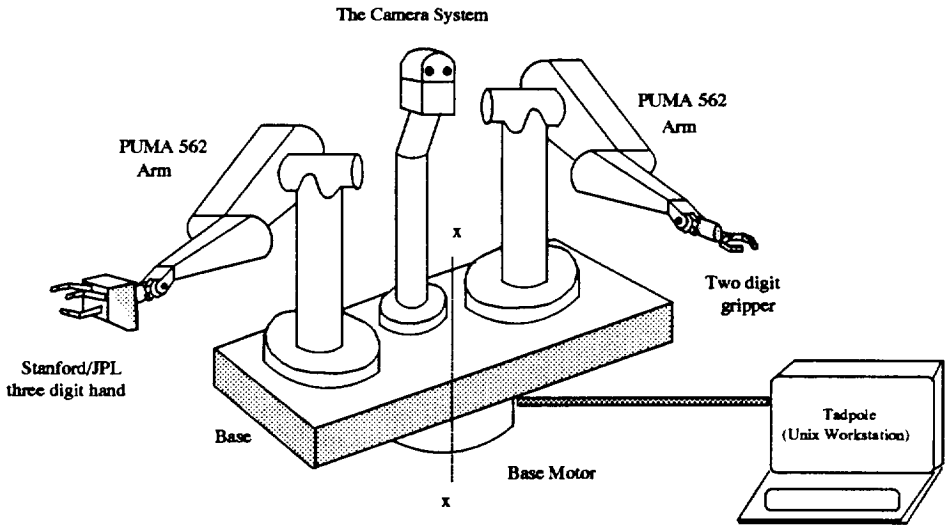

Figure 6: The DART System. It comprises of two PUMA 562 arms and a vision system. The whole assembly rests on a base which can be rotated about the central x–x' axis by a motor. Each of the PUMA 562 arms has six joints yielding a total of six degrees of freedom for each arm. The right hand consists of a Stanford/JPL three digit hand. Each of the digits can consists of three joints which can be controlled independently. The left hand of the DART comprises of a two digit gripper which is controlled by a single motor. The Tadpole Unix Workstation is used to control the two arms and their hands.

controller is also capable of applying the brakes at the joints of PUMA 562 arm. Commands to the controller can be issued by writing to a shared memory location that is read by the three CPU's of the controller. The various states such as the joint angles and the torque at each joint of the arm are written by the CPU's on to shared memory locations which the computer can read or write to. The internal states of the arm are updated every millisecond. This fast update rate enables to achieve near real time feedback control of the arm. The inverse and forward kinematic routines that were used for controlling the arm are based on the solutions for PUMA 562 arms available in standard robotic text books (Craig, 1989) [5].

The Standford/JPL three digit right hand of the robot comprises of two fingers and a thumb. The hand is controlled by a set of 12 servo-motors via a set of steel cables. Strain gauges located at the base of each finger provides tension feedback which provides information about the applied force. Also position feedback of the three digits is obtained by reading out the value of the encoder for each of the joints. The left hand comprises of a two digit gripper which work in unison as the digits are controlled by a single motor. The gripper motor is controlled independent of the motors of the arm joints[6]. Currently only the left arm and hand of the robot is used. In the future, we would like to use the three digit right hand to perform dexterous tasks.

When the robotic system wants to pick an object from its environment, it sends via the communication protocol the location of the object to the computer controlling the PUMA arm. On receiving the spatial location of the object, the computer computes the inverse kinematics for the PUMA arm. A trajectory for the arm motion is generated via joint interpolation. This interpolated set of joint-space points is written into the shared memory of the arm controller by the computer. On reaching the required location, the gripper motor is initiated to grab the object. After grabbing the object, the robot arm is then commanded to return back to its default position and releases the picked object into a bin. The performance of the arm is presented in the Section 5.

2.3 FRONTAL

The neural network called FRONTAL that controls the robot is shown in Figure 7. FRONTAL comprises of the following four parts:

- spatial novelty network,
- attentive scanning network,
- object novelty network and

[5]These routine were developed and tested by Mr. Larry Li.

[6]Details regarding the gripper operation is given in Appendix C.

- behavioral categorization network.

The spatial novelty network (shown in the bottom left hand corner of Figure 7) comprises of an array of gated dipoles which are inter-connected via a winner-take-all layer of neurons. This network enables the robot to detect a new object that enters as well as an existing object that leaves its field of view. The working of the spatial novelty circuit can be better understood by studying how a single gated dipole functions. A gated dipole network is shown Figure 8. It comprises of two parallel channels called the "ON" and the "OFF" channels respectively, which inhibit each other. Both channels receive a common arousal signal "I" while the external input signal "J" is applied only to the "ON" channel. The input signals to these channels are conveyed by depletable transmitters (marked by the square). The "ON" channel activity provides a measure of the novelty of the applied external signal. As the "ON" channel inhibits the "OFF" channel the removal of external signal "J" yields a transient reduction in the inhibition until the "ON" channel transmitter is replenished. This transient reduction in inhibition on the "OFF" yields a concomitant transient increase in the activity of the "OFF" channel. Together the "ON" channel and the "OFF" channel activities provide a measure of the novelty of an applied external input signal "J" and a signal indicating its removal. Figure 9 illustrates a neural network consisting of an array of interconnected gated dipoles capable of encoding novelty. The winner-take-all layer neurons of this novelty detection network also get input from the reward and punish neurons that encode the external reinforcement signals thus enabling the circuit to weight these signals againsts the novelty of the input. Simulations of the spatial novelty network alone and in combination with the reinforcement signals are presented later. Variations of this network are used in FRONTAL for detection of spatial novelty and object novelty. An array of these gated dipole networks, which constitute the spatial novelty circuit enables the robot to detect the introduction of a new object as well as removal of an old object from its surrounding. Each of the gated dipole correspond to a unique spatial location in the field of view of the robot. The neurons in the winner-take-all layer which receive inputs from both "ON" and "OFF" channels of their respective gated dipole.

The attentional scanning network shown in the upper left corner of Figure 7 enables the robot to scan all the objects present in its environment. This network comprises of *arousal*, and *inhibitory feedback* neurons. They play a role in temporarily disengaging the attention of the robot from the current object. This in turn allows the robot to shift it's attention to another object in its surroundings. The duration of this disengagement is controlled by *delay* neurons. The *arousal* neuron receives an inhibitory signal from the categorization network which ensures that the attention of the robot is not disengaged during categorization of the object.

While the robot attends to a particular object, the object novelty network which is in the far right of Figure 7 categories object into different types and ascertains whether that type is

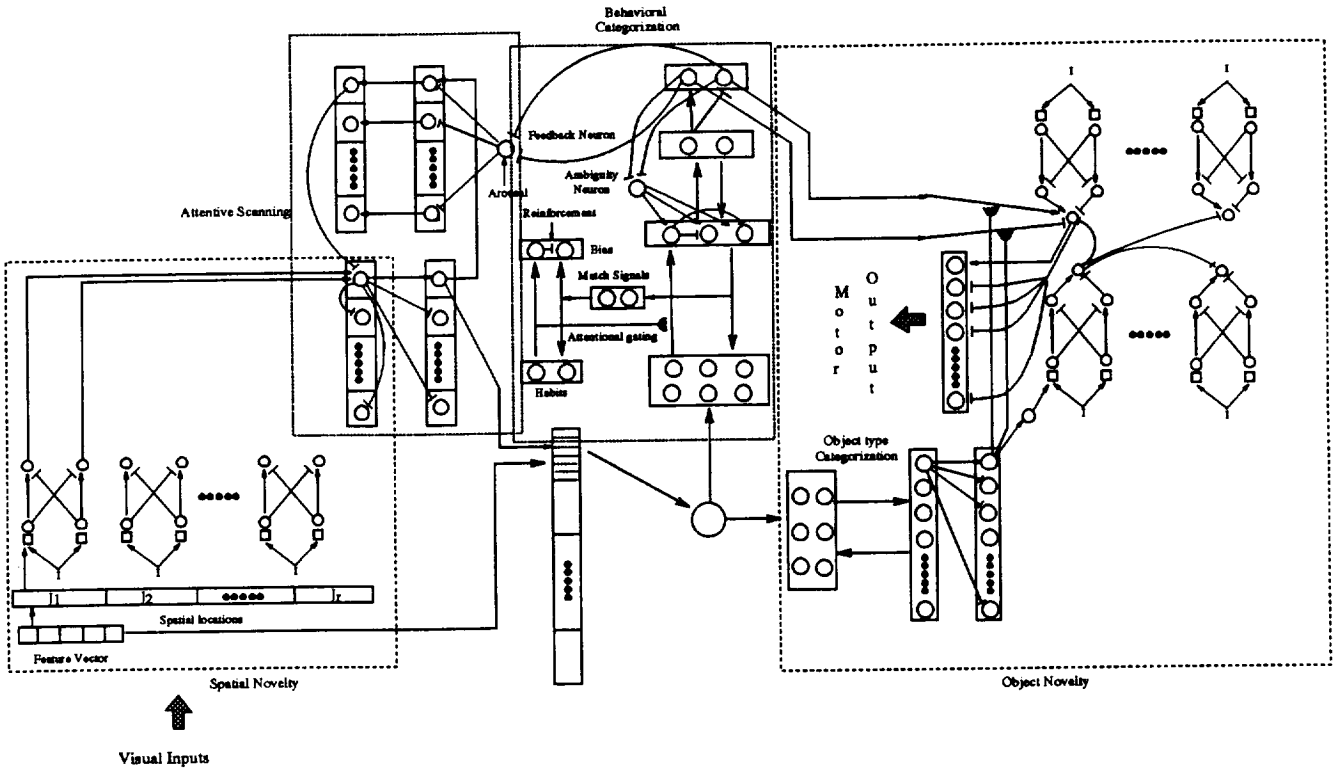

Figure 7: The "cognitive controller unit": Visual inputs shown at the bottom left of the figure are processed by spatial novelty and attentive scanning networks. The latter determines the spatial focus of attention. The features of the object present in that spatial location are sent to "behavioral" and "object-type" categorization networks. The behavioral categories consist of "good" and "bad". The outputs of the object-type categorization network are fed to object novelty network (the gated dipoles at the right of the figure). These gated dipoles are connected to a winner-take-all network which also receives inputs from the behavioral categorization network (excitatory from the good category and inhibitory from the bad category). When there is a winner in this network, a motor command signal is sent to the robot arm to initiate a visually guided reach movement towards the winning object.

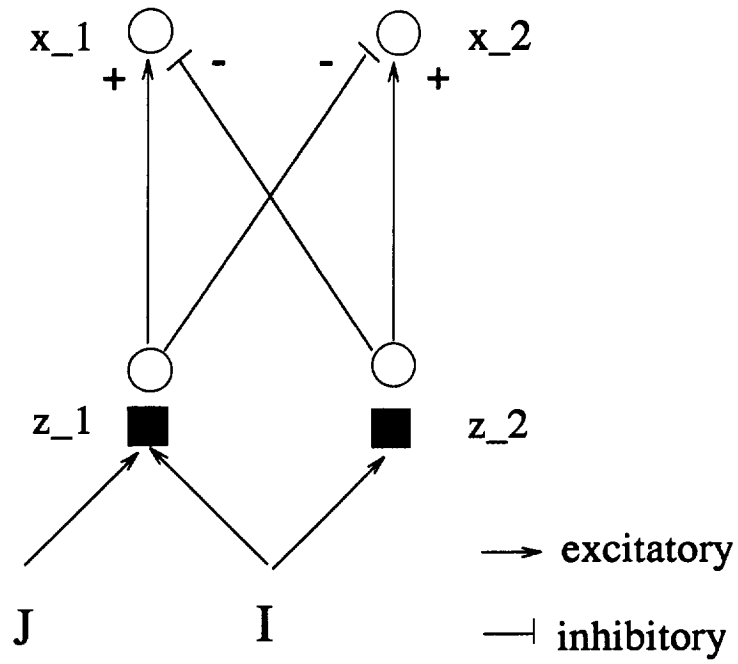

Figure 8: A gated dipole is shown in this figure. It comprises of two parallel channels each receiving a common arousal signal "I". Each of these channels has an inhibitory effect on the other. The channel receiving the external signal "J" is called the "ON" channel and the other is called the "OFF" channel. The dark squares represent synapses containing depletable transmitters. The transmitted signal in each channel depends on the total input received and the amount of transmitter present in that channel. Initially, when no external input is applied, the activities in "ON" and "OFF" channels are the same since they receive the same input. On application of the external signal "J", the "ON" channel has a larger activity than the"OFF" channel. Since the rate of depletion of the transmitter is dependent on the input, the longer the signal "J" is on the greater is the depletion of the transmitter in the "ON" channel synapse. This causes the activity of the "ON" channel to slowly decay and thus gives a measure of the novelty of input "J".

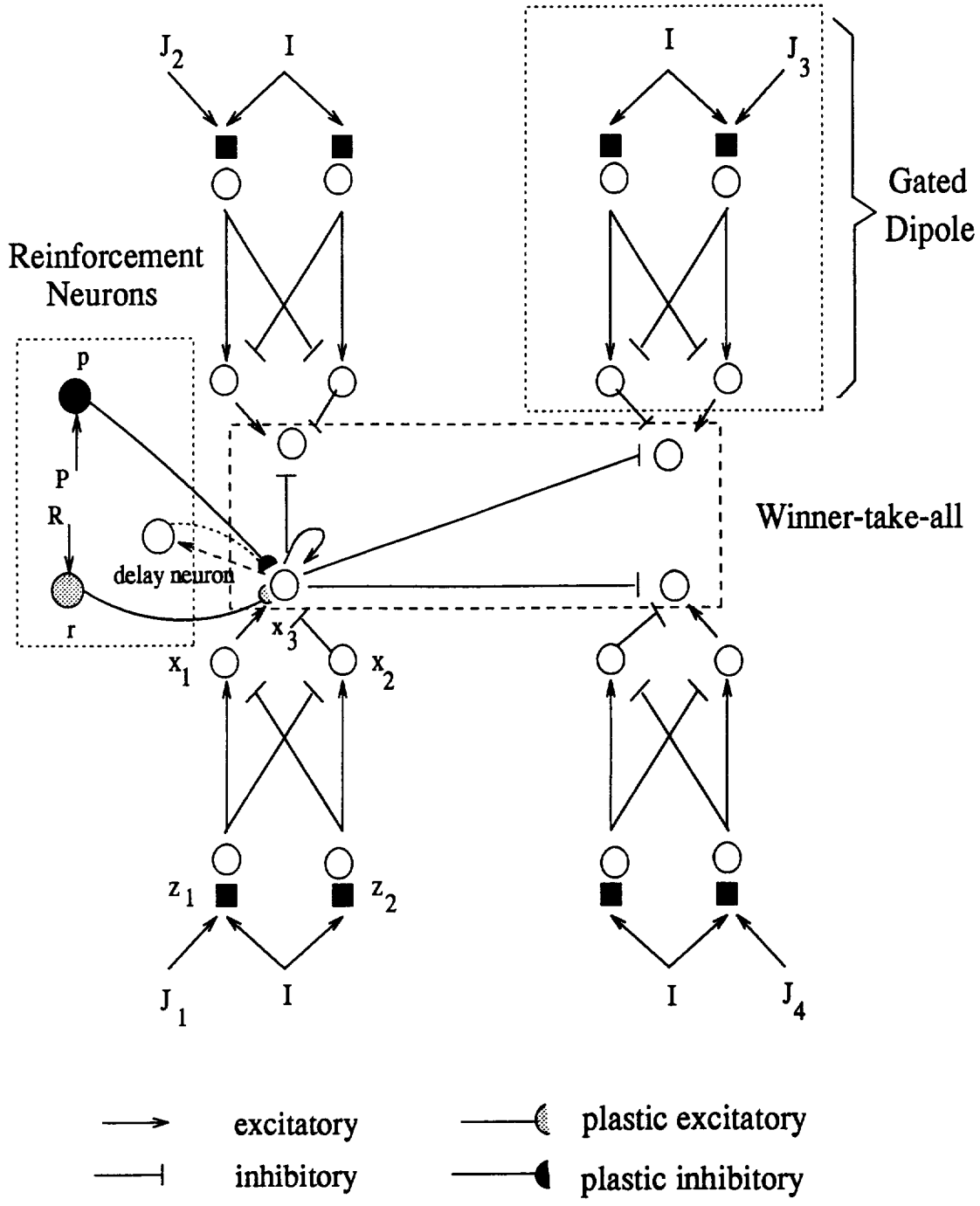

Figure 9: A neural architecture for novelty detection comprising of an array of gated-dipoles. These gated dipoles are connected to a winner-take-all network. The reinforcement signals also influence this decision of the novel stimulus by gating the neurons in the winner-take-all layer via additional "reward", "punish" and "delay" neurons. (Modified from Levine and Prueitt, 1989.)

novel. The behavioral categorization network which is in the center of Figure 7 categorizes the object according to its behavioral significance (i.e good objects are those associated with positive reinforcements and bad objects are those associated with negative reinforcements).

The object novelty network comprises of two parts, the object-type categorization network and the novelty detection network. The object-type categorization network is an ART network that categorizes the input objects to different types depending on its features. The output of the categorization layer is then fed to a novelty detection network comprising of gated dipoles which determine whether the object type is novel. The output of these gated dipoles are fed to a winner-take-all network. This winner-take-all network also receives inputs from the behavioral categorization network. The combined object novelty as well as behavioral categorization networks signals is used to drive the robot's arm.

The behavioral categorization network in the center of Figure 7 comprises of an ART network. This ART network is modified to dynamically change its internal criterion for categorization. Figure 11 gives a more detailed view of a network having similar properties as the behavioral categorization network of the FRONTAL. In this network, there are three features and four categories. To understand how the behavioral network categorizes an object, consider Figure 10. The input object is shown at the bottom of the figure. The two objects at the top represent the templates for the "good" and the "bad" categories. It can been seen from the figure that the categorization of the input results in an ambiguity if the criterion to be used in the categorization is not know. The habit and the reinforcement signals guide the network in its choice of the categorization criterion. The *reinforcement* neuron encodes the externally issued reinforcement signals to the robot. This non-specific signal is correctly assigned to the network's current choice of internal criterion by the *match* neuron. Both the *reinforcement* and *match* neurons are shown in the behavioral categorization network of Figure 7. The *habit* neurons at the bottom of the behavioral categorization network memorize the past experience of the network. The *bias* neurons combines reinforcement and habit signals to generate the appropriate internal criterion to be used to categorize input objects. Thus this network dynamically modifies its internal criterion for categorization depending on its past experiences and the reinforcement signals it receives.

The *ambiguity* neuron (shown at the top of the behavioral categorization layer) enables the network to assign the input object to one of the behavioral categorizes in ambiguous situations. The *ambiguity* neuron accomplishes this by biasing one of the category neurons. The *decision* making neurons filter the transients generated by *category* layer of neurons (i.e the F_2 layer of ART) during competition. This suppression of spurious transients and passing of steady state signals enables this network to be interfaced with other networks in a continuous non-algorithmic manner. Simulations of the working of this network are presented later.

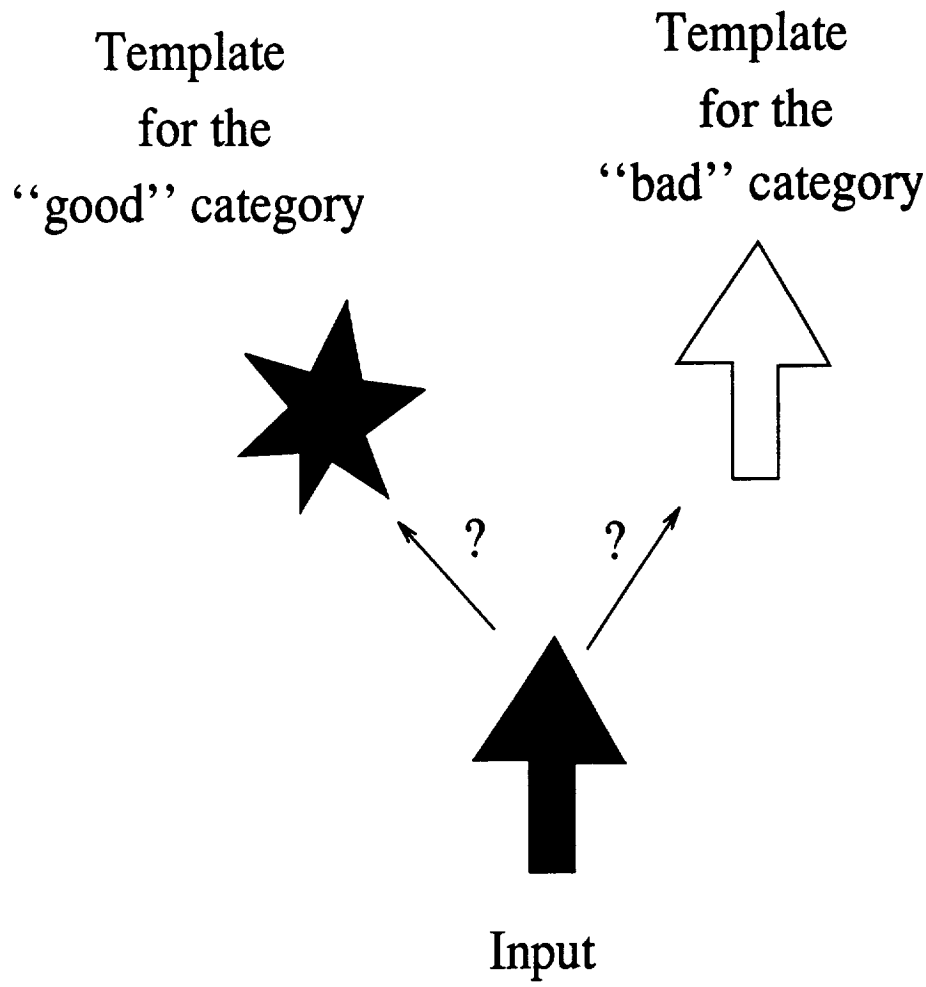

Figure 10: The input object shown at the bottom of the figure has to be categorized into one of two categories whose templates are shown at the top of the figure. The template at left may be for example for "good objects" (the system will then pick this object) and the template at right may be for "bad objects" that the system learned to avoid through reinforcement signals. The categorization here is ambiguous in that if color is taken as criterion then the input is a good object but if shape is taken as criterion then the input is a bad object.

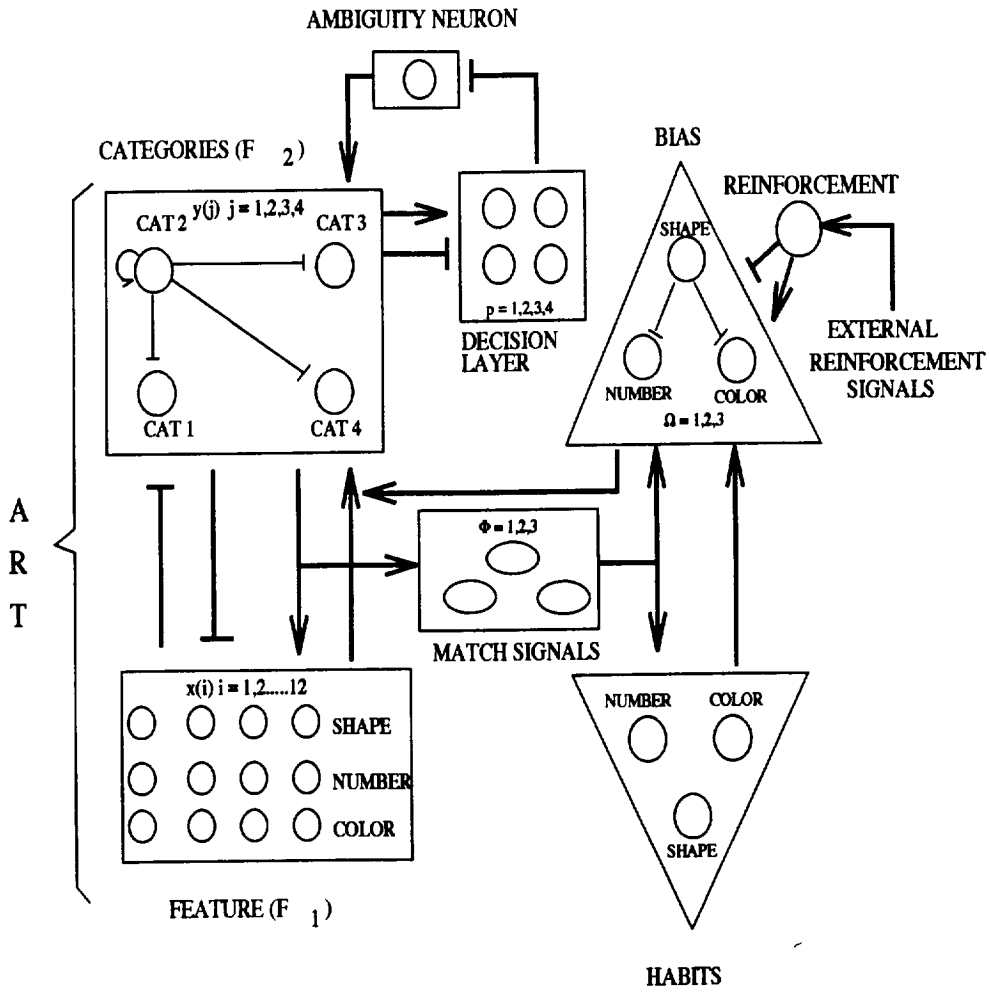

Figure 11: A neural network architecture capable of dynamically modifying its internal criterion (shape, number, or color) for categorization: The reinforcement signal is encoded by the *reinforcement* neuron. The *habit* neurons memorize the number of times a given internal criterion was used for categorization. The *bias* neurons combine reinforcement signal and habit signals and modulate the internal criterion of the network. The match neurons encode which criterion is currently being used for classification. This plays an important role in gating the non-specific reinforcement signal with a particular internal criterion. The decision and the ambiguity neurons are introduced for self-contained, continuous, non-algorithmic functioning of the network. Spurious transients that could arise in the F_2 layer of ART due to competition are filtered by the decision neurons. The ambiguity neuron is involved in the selection of one of the possible categories in situations when an object can be categorized to more than one category. (Modified from Leven and Levine, 1987).

A similar type of network, used in FRONTAL, enables the robot to decide whether the object it is looking at is a good object (and hence pick it) or a bad object (and hence not pick it). Good objects for the robot are those that have been correlated with positive reinforcements and bad ones are those which have been correlated with negative reinforcements.

The frontal network shown in Figure 7 thus enables the robot to scan for objects in its environment and to categorize these objects by picking the good and novel ones and by refraining from the bad ones. FRONTAL also provides the robot with the ability to modify its internal representation of the environment dynamically by interacting with it environment. In conclusion FRONTAL enables the robot to self-organize in a dynamic environment.

2.4 Communication Protocol: Sockets

With the advent of cost-effective fast dedicated-processors task-specific computers are now widely used. Many applications require the development of firmware to communicate between these computers. Various standards are available for development of these communication interfaces. In this implementation, we initially developed a communication interface using the TCP/IP sockets protocol [7]. The communication interface was designed to perform in the simplest manner, communication of information by Amdahl supercomputer with the Sun Sparc 2 (running the vision system) and the Unix Workstation controlling the arm Figure 12. The sockets approach was used instead of the datagram approach so as to ensure reliable communication between the computers. The overall strategy was to allow each of the computers controlling the peripheral systems (i.e the vision and arm) to independently interact with Amdahl where the FRONTAL (brain) system is running. When a new object is introduced in the visual space of the system, this information is communicated to the FRONTAL by the vision system via a dedicated socket. On receiving this information, a confirmatory signal is sent back by FRONTAL. In a similar manner, when FRONTAL decides to initiate a grasp it communicates with the Arm system which in turn executes the grasp. On completion of the grasp, the arm system issues a "success" signal to FRONTAL. Two different types of message packets are used by FRONTAL to communicate with the vision and the arm systems. The message package communicated by the vision system comprises of variable size data segments depending on the number of objects present in the environment. The message stream is terminated by an end of line terminator as shown in Fig 13. Each message stream contains data segments comprising of the following information for an object in the environment: the x, y and z co-ordinates of the centroid of the object; its shape (whether it is a triangle, square or a circle); and its size (whether it is small medium

[7] A comprehensive discussion of the various communication protocols as well as the TCP/IP protocols is given in (Stevens, 1990)

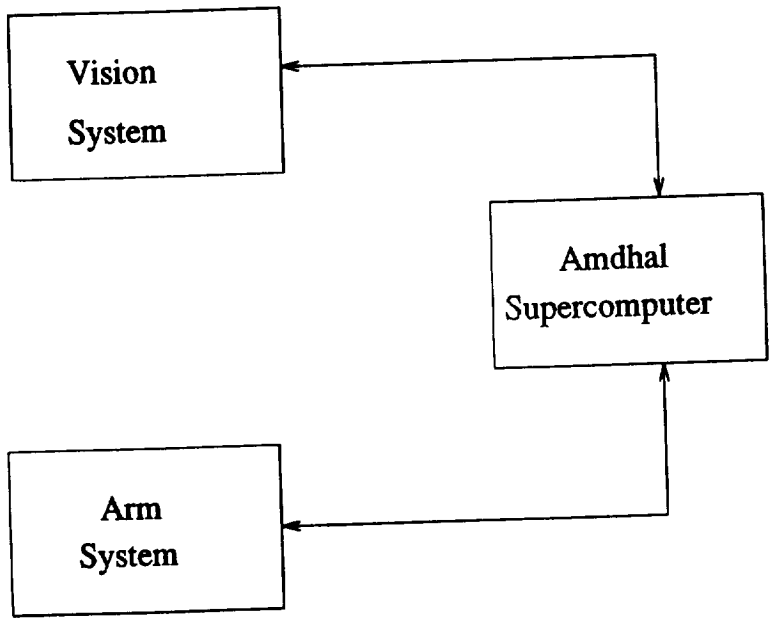

Figure 12: The computers involved in controlling the robotic system are shown. The arrowed lines indicate the communication network between the computers. The Amdahl computer communicates with both the Sun Spare Station and the host computer controlling the robot arm.

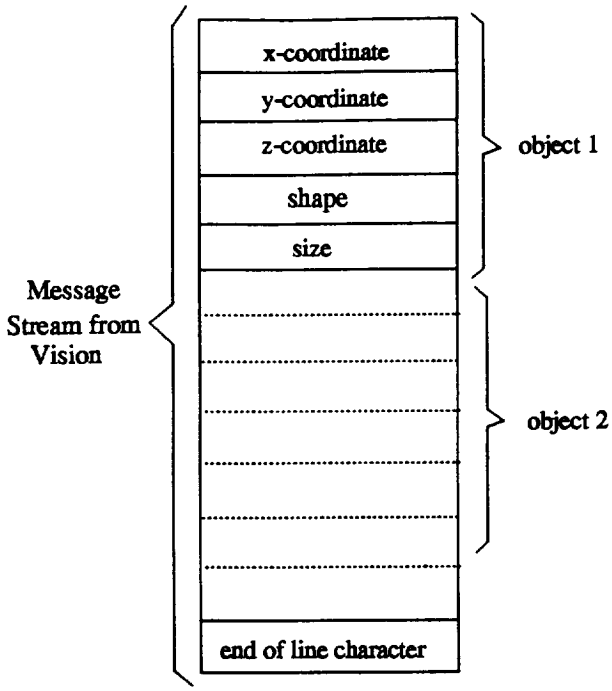

Figure 13: The message stream from Vision system to FRONTAL

or large) [8]. The message stream from the FRONTAL system to the arm system however is a fixed length message. It has three fields which give the respective x, y and z co-ordinates of the centroid of the object to be grabbed. Here too the message stream is terminated by a new line terminator as shown in Figure 14. Care has been taken to ensure that the basic unit of a message packet is a string of arbitrary length. This enables any structure to be passed as a message stream across the system. This initial implementation of the communication protocol provided a means to test the robotic system and its constituents parts. Later a more sophisticated communication protocol was implemented using TELRIP. TELRIP is a NASA software package that provides interprocess communication protocols between processes running on different Unix platforms. TELRIP enabled us to segregate the processes controlling the robotic system from those responsible for communication.

[8]The size and the shape of the object are represented by bytes taking one of the values 1,2 and 3. Thus, for a medium square the last two bytes would be 2 2 respectively

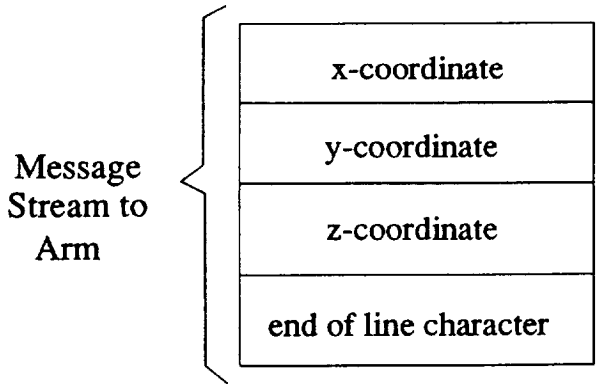

Figure 14: The message stream from FRONTAL to the Arm system

3 Simulations

In this section we present simulations of various neural architectures which elucidate the functioning of the various components that constitute FRONTAL. All of these neural architectures have been simulated using the Amdahl supercomputer. A numerical ODE solver (the Runge-Kutta-Fehlberg 4-5 method) developed by Oak Ridge Labs is used for solving the ordinary differential equations representing these neural architectures. The equations for the neural architectures and the values of the parameters are presented in Appendix A.

The user interface of the simulation enables the modification of external signals (introduction or removal of objects and external reinforcement signals) by interrupting the program as and when needed. On interruption, all the state variables of the network are pushed onto the stack of the computer and the interrupt is handled. On returning back from the interrupt, these state variables are reloaded back and the network equations are solved from the same internal state of the network before the interrupt occurred.

3.1 The novelty detection network

Simulations demonstrating the capability of the neural network to recognize novelty are shown in Figures 15 and 16. Since the same type of network, namely the gated dipole, is used for both spatial and object type novelty, the simulations apply to the former when the input signals come from the spatiotopic locations and to the latter when they come from the category layer of the categorization network. In this simulation, the network shown in Figure 9 (described by the equations presented in Appendix A. 1) is implemented. The plots in Figure 15 represent the temporal sequence in which four inputs are presented to the neural network, a high signal implying the presence of the input and a low signal implying its removal or absence. The plots

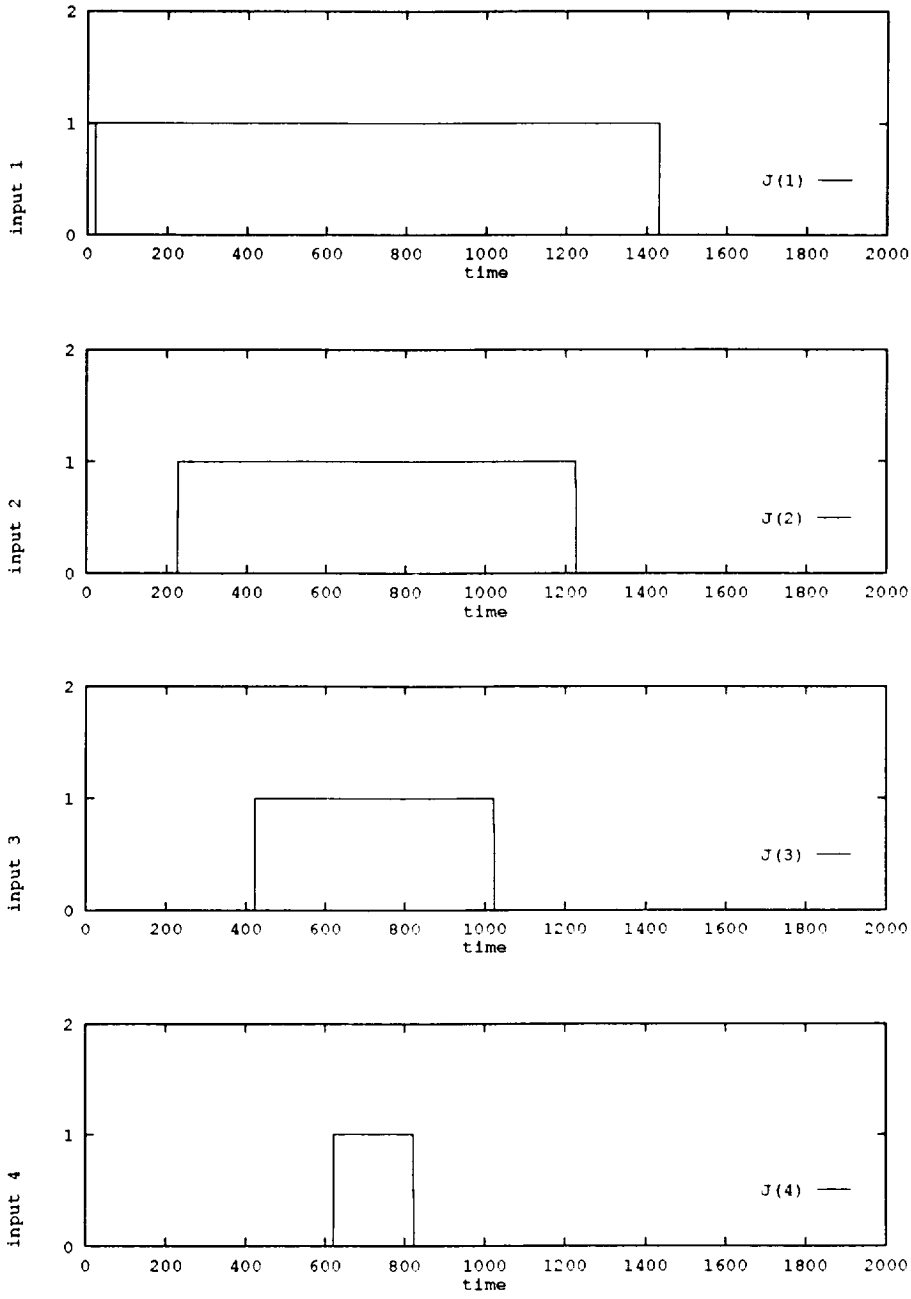

Figure 15: This figure along with Figure 8 demonstrates the novelty detection capability of the network shown in Figure 2. The four panels in this figure represent the sequence in which four inputs are presented and removed from the network's environment. A high signal implies the presence and low signal indicates the absence of the input. The response of the network is shown in the next figure.

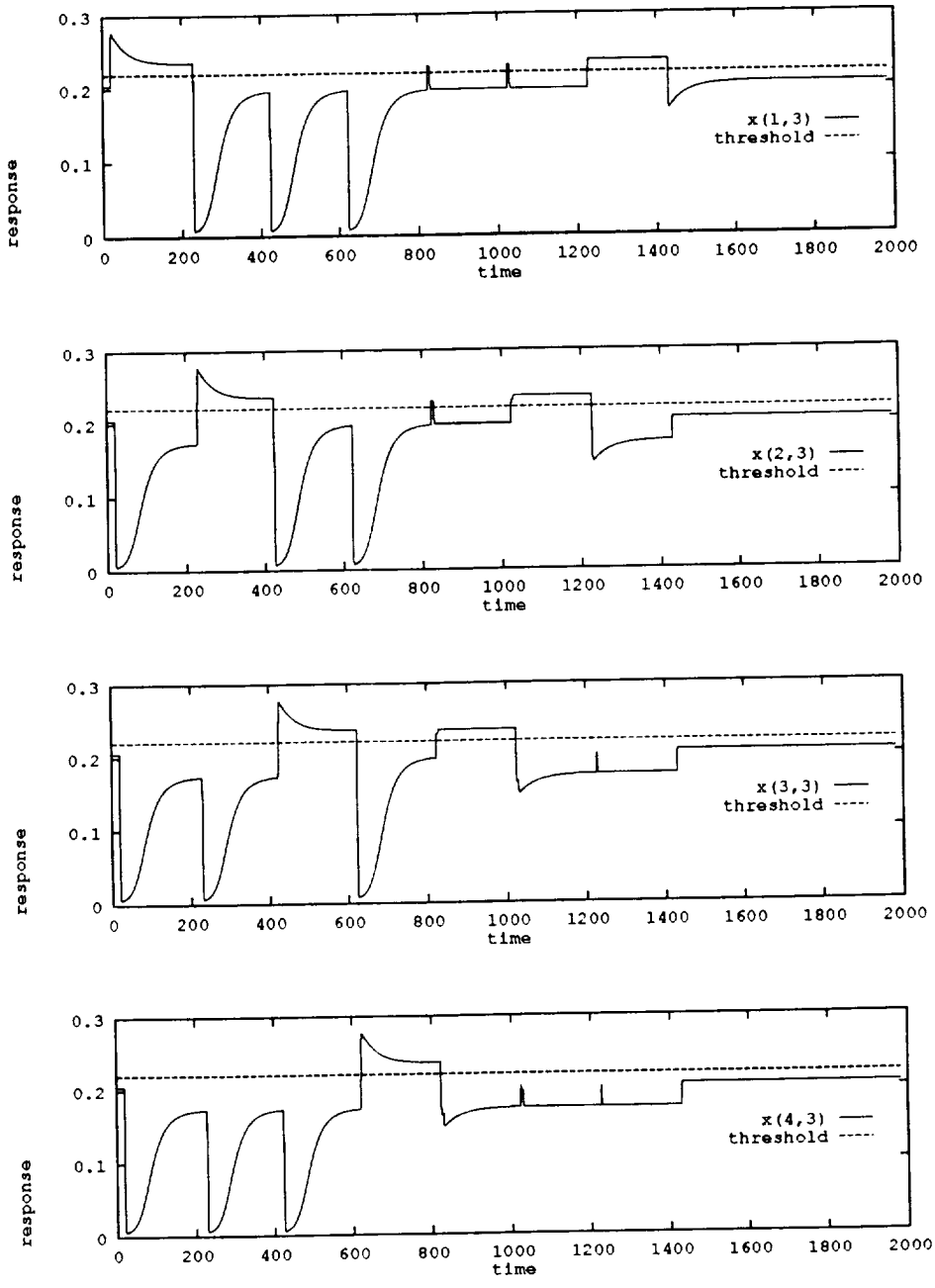

Figure 16: The panels plot the activity of the four neurons in the "winner-take-all network" (see Figure 2). The sequence of inputs presented to the gated dipole circuits that project to these competing neurons is shown in the previous figure.

in Figure 16 graph the temporal activity of the x_3 neurons of the winner-take-all network (see Figure 9). The horizontal line indicates a threshold value above which the activity of any x_3 neuron implies that the network focuses its attention on the corresponding input. As can be seen from the plots of Figure 9, the $x_{1,3}$ (this notation $x_{i,3}$ refers to the x_3 neuron (of the winner-take-all network) which receives excitatory input from the "i^{th}" gated-dipole network sampling input i) neuron's activity is above threshold until the second object arrives around 227 *time units* which causes the $x_{2,3}$ neuron's activity to rise above threshold. This in turn causes the activity of $x_{1,3}$ neuron to go below threshold. Similarly the arrival of objects three and four causes the activity of $x_{3,3}$ and $x_{4,3}$ neurons to respectively go above threshold. Thus, other things being equal, novelty guides the attention of the network.

3.2 Reinforcement versus novelty

Reinforcement can bias the attention of the network as demonstrated by the simulation result presented in Figure 17. Initially input 1 is presented to the network, to which the network immediately responded by activating the $x_{1,3}$ neuron above threshold. Following this, input 2 is presented and the network attends to it on account of its novelty. When input 2 is removed, the network attends back to input 1. Now, while the network attends to input 1, a positive reinforcement is delivered for about 20 *time units*. After this reinforcement, when input 2 is introduced again, the attention of the network remains on input 1 despite the fact input 2 is relatively more novel. This is the result of the previously delivered positive reinforcement that the network associated with input 1.

Negative reinforcement on the other hand yields opposite effects as shown in Figure 18. When an input is associated with punishment, the network learns to avoid that input. Even when the input reappears much later, its novelty is not strong enough to bias the network's attention towards it. Thus the network learns to avoid punishing inputs even though they could be relatively novel. The effects of both the punishment and the reward fade away with time if further reinforcements are not issued and eventually novelty dominates.

3.3 The delay neuron

In case of positive reinforcement, the encoding of the STM of the *reward* neuron into LTM follows the classical Hebbian learning rule with decay. This is possible because the reward node is connected via excitatory connections to neurons in the choice layer. As a result, when reward is delivered this reinforces the activity of the choice neuron which is supra-threshold (thereby crediting reward to the current choice). This creates a temporal correlation of pre- and post-synaptic activities as required in a Hebbian learning term. However the "punish node"

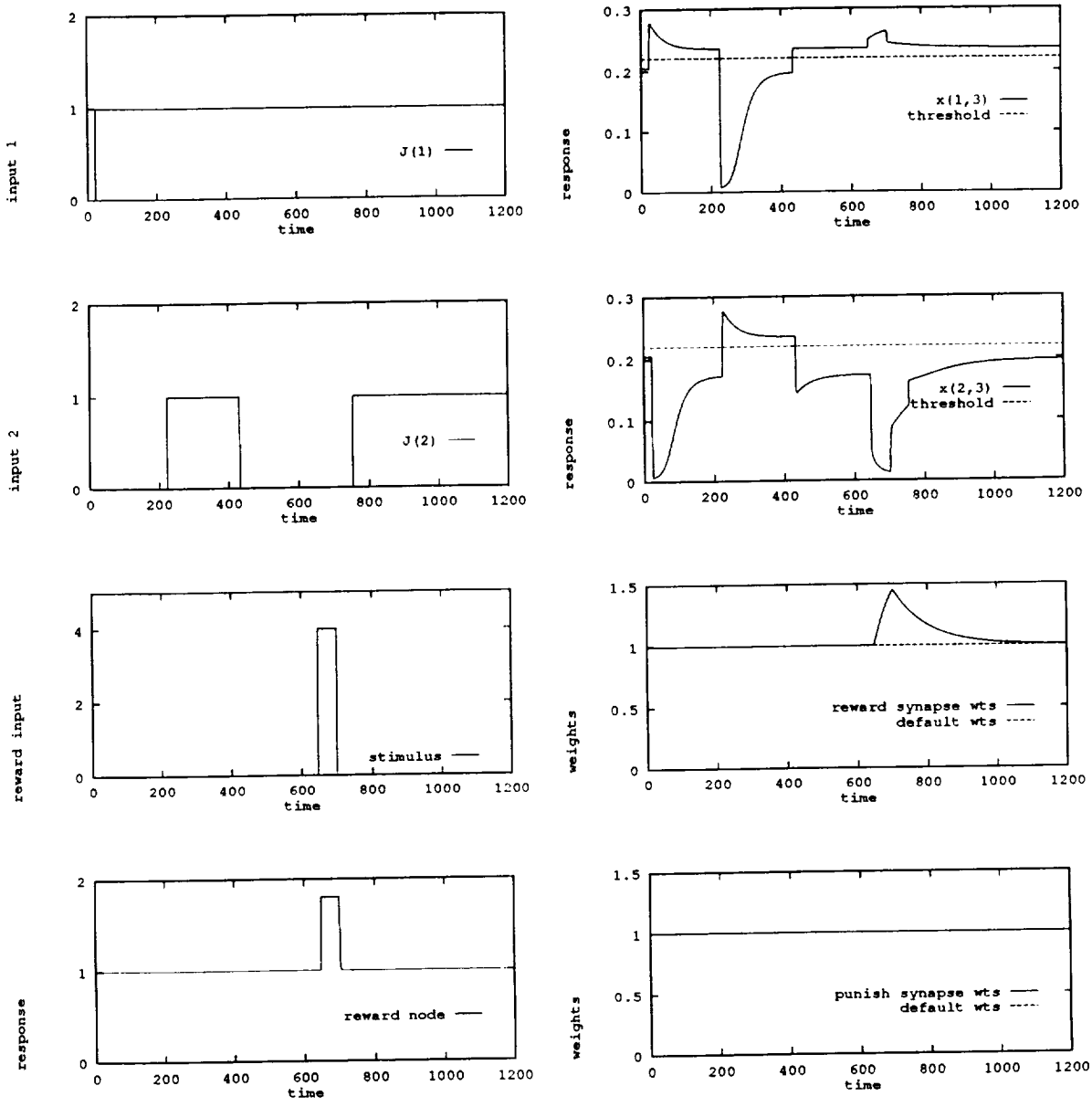

Figure 17: The effects of positive reinforcement: Initially the network attends in order to inputs 1 and 2 due to their novelty. However the application of a reward signal (positive reinforcement) to the network while it is attending to input 1 causes the network to ignore object 2, even though it is relatively novel due to its reintroduction. The network associated the positive reinforcement with input 1 and this outweighted the novelty of input 2. The encoding of the temporal association between the activities of the *reward*and *decision* neurons into LTM (i.e into the *reward* synaptic weights) is shown in the activity of the reward synapse.

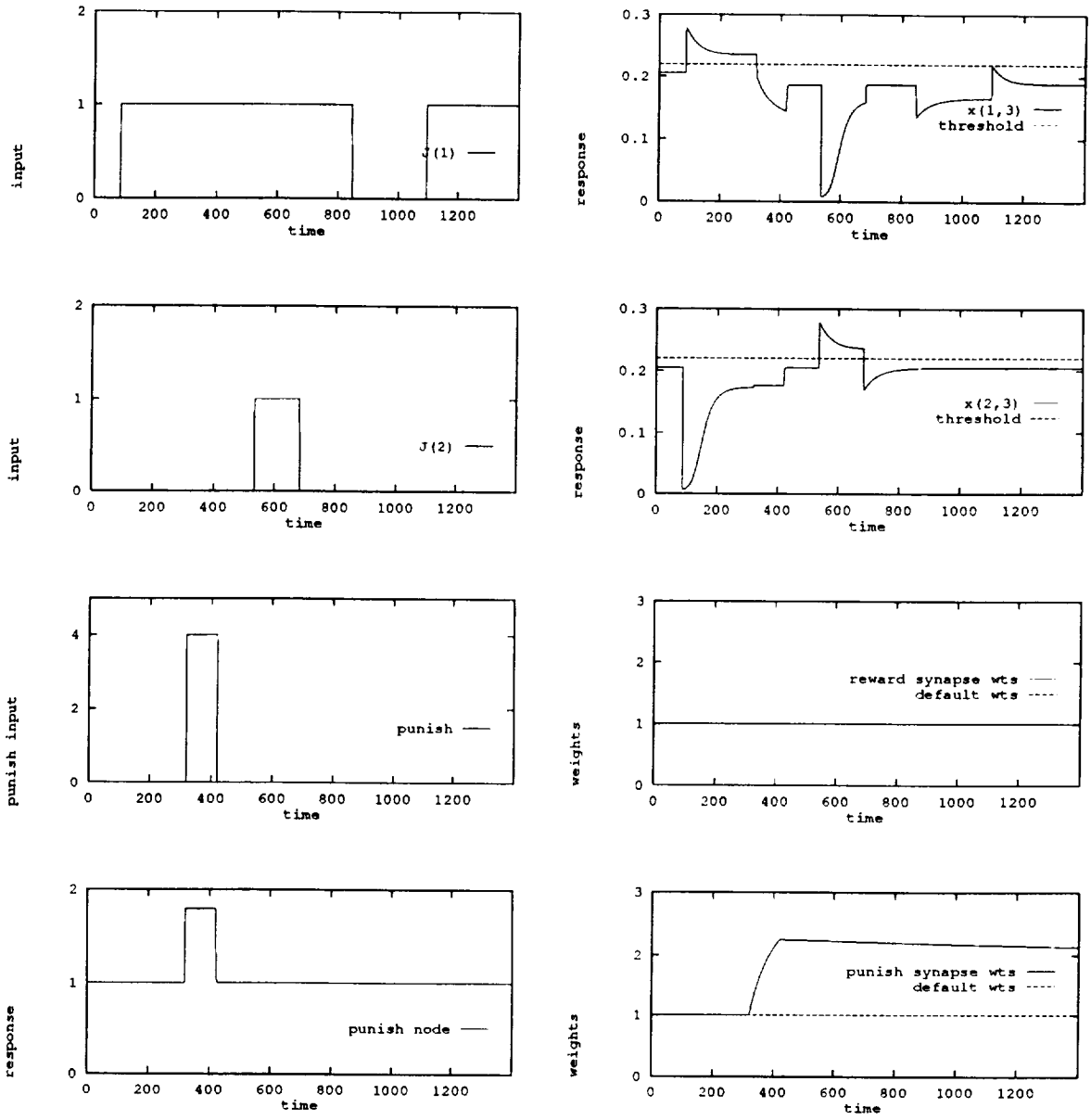

Figure 18: The effect of negative reinforcement: Initially the network attends to input 1. On application of negative reinforcement, the network shifts its attention away from input 1. The arrival of input 2 causes the network to attend to this novel input. The reappearance of input 1 after the removal of input 2 is not sufficiently novel to outweigh the effects associated with its punishment. Hence the network avoids input 1. The lower right panel plots the encoding of negative reinforcement signals on the *punish* synaptic weight of the network.

has inhibitory connections with neurons in the winner-take-all circuit in order to depress the activity of the neuron which is supra-threshold and thereby rapidly force the robot to avoid that particular object. As a result of this inhibitory effect, pre- and post-synaptic activities remain simultaneously active for a very brief time period (see Figure 18). This leads to an ineffective coding of the *punish* neuron activity into LTM via a Hebbian term which requires a temporal correlation of pre- and post-synaptic activities. To avoid this problem, the *delay* neuron whose STM trace follows a delayed version of the x_3 neuron is introduced along with a modified learning rule which is given in the Appendix A.

3.4 Variable criterion categorization

In complex environments, it is often necessary to modify criteria used in classification according to prevailing conditions. For example, while color may be an adequate criterion to separate good and bad apples during certain period of the year, during other times color may be misleading while the size of the apples may be more adequate (cf. example in Figure 10). Figure 11 describes a network capable of changing its categorization criterion based on reinforcement signals. In order to avoid noise in reinforcement signals, the network forms "habits" that encode the frequency of behaviors. The criterion of the network is modulated by combining evidence from reinforcement and habit signals. Simulations demonstrating this property are shown in Figures 20, 21 and 22. The upper three panels of Figure 20 describe graphically the input presented to the network at different time instants. Each input possesses three features. Each feature has four distinct values (types). For example, a feature can be color and the four types can be red, blue, yellow, and green. Thus a total of $4 \times 4 \times 4 = 64$ distinct inputs can be presented to the network. Each of the three panels describes one of the features of the input. The four distinct types of each feature are represented by the different stylings of the "bars". A set of bars, one from each panel, starting at the same time represents a particular input presented to the network. The width of the bars represent the time taken by the network to categorize this input. The category layer of the ART network used in the simulation has four neurons. Hence, the inputs are categorized into one of four possible categories. Figure 21 shows the category chosen by the network for a given input at different times. Each panel represents the activity of a neuron in the category layer of the ART. The supra-threshold activity in a given panel indicates that the input is categorized to that particular category. Category neuron activities have similar styling as the four possible types of each feature. At any instant of time, at most one category has a "bar" indicating that the network classified the input object to that category. The feature used by the network to categorize the input can be easily identified by comparing which of the first three panels of Figure 20 has a similar bar as the category panel at the given instant. For example the first input presented to the network is of type 2 of feature 1, type 1 of feature 2, and type 3 of feature 3. The network

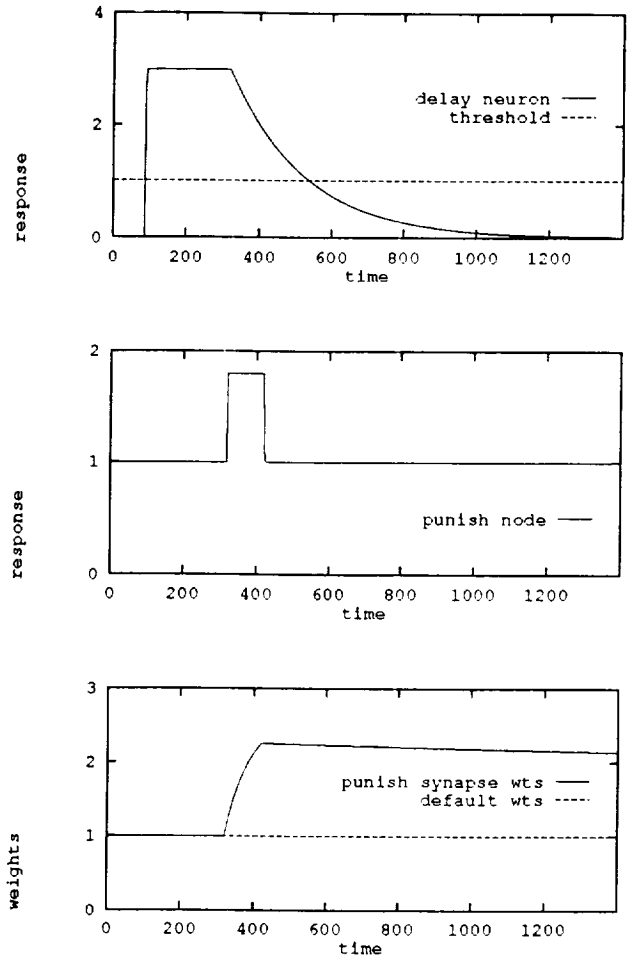

Figure 19: The three panels shown in this figure demonstrate the activities of the *delay*, *punish* neurons and the *punish* synaptic weights after the application of negative reinforcement signal to the network as discussed in Figure 9. The *delay* neuron follows the activity of the x_3 neuron in a delayed fashion so that the STM of the *punish* neuron can be encoded into LTM by the *punish synaptic* weights.

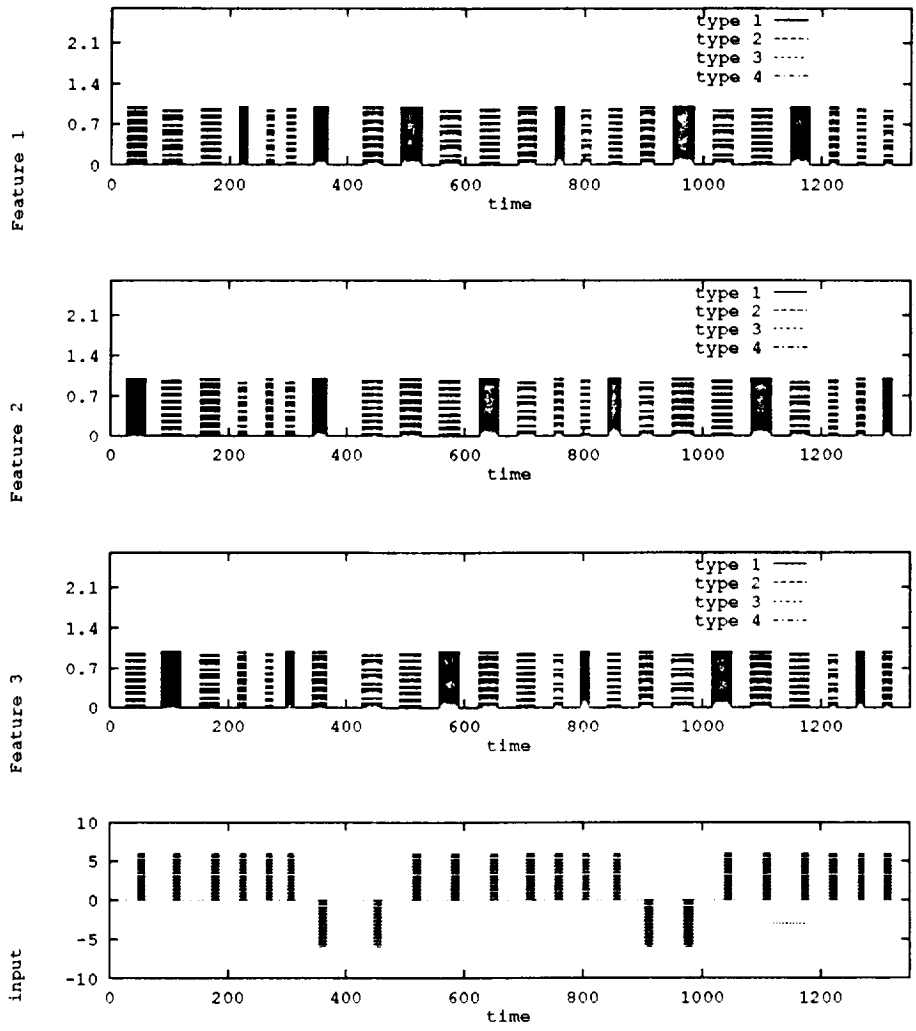

Figure 20: The first three panels describe the features of inputs presented to the network at different time instants. Each input possesses three features (e.g shape, color and size) and each feature can take one of four possible values (types). Hence 64 different inputs can be presented to the network. Each of the first three panels represents a feature. The four different styles of bars in each panel represent the four different types of a given feature (e.g. for color they may correspond to white, blue, yellow, and red). At any instant of time, the bars represented by the three panels describe the properties of the input presented to the network e.g., the first input is of type 2 of feature 1, type 1 of feature 2 and type 3 of feature 3. The width of the bars represent the time the network took to categorize that object. The categorizations performed by the network are presented in Figure 13. The last panel describes the reinforcement signals delivered to the network in response to its categorization of the object.

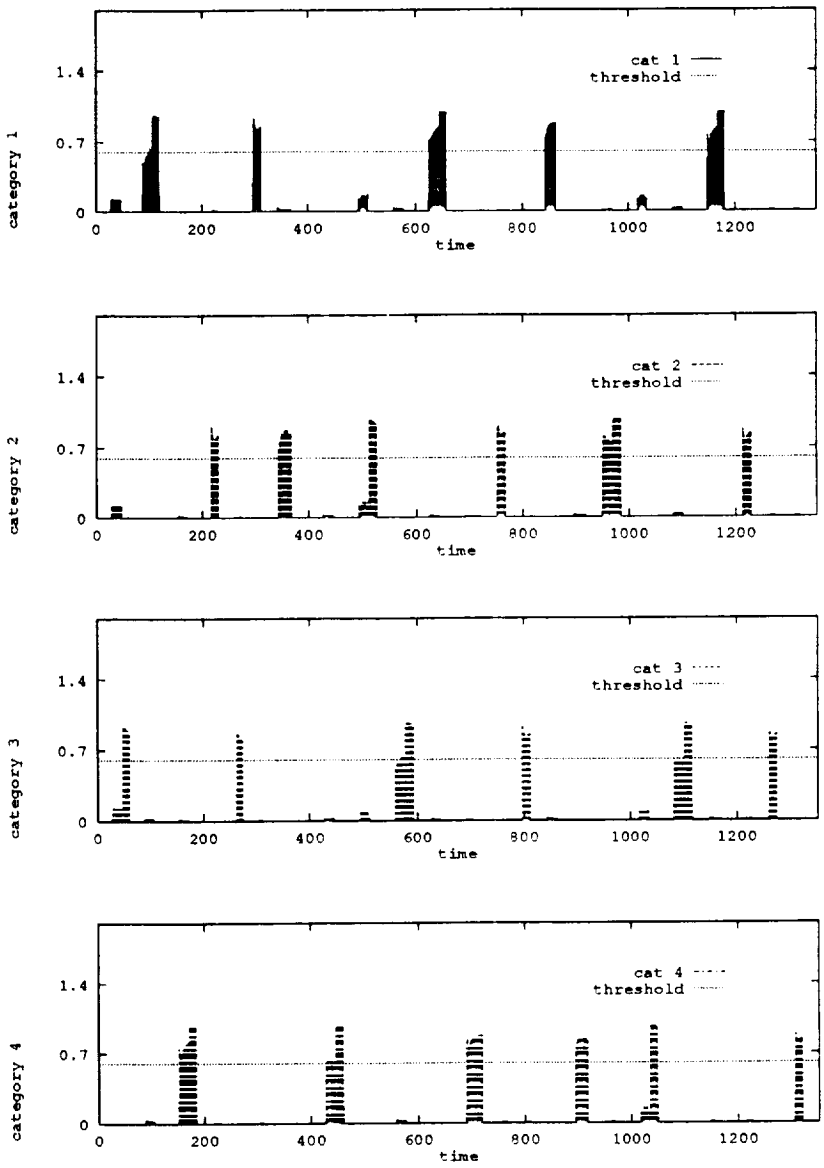

Figure 21: The activities of categorization layer neurons of the ART network are presented in this figure. Each of the panels represent the activity of a single *category* neuron as a function of time. At any given instant of time at most one category neuron is active. The four different types of bars represent the four types (e.g. red, blue, yellow and green) of each feature (say color). By comparing the styling of bars of the category neuron with that of the features in Figure 12, the criterion used by the network to categorize the input at that instant of time can be ascertained.

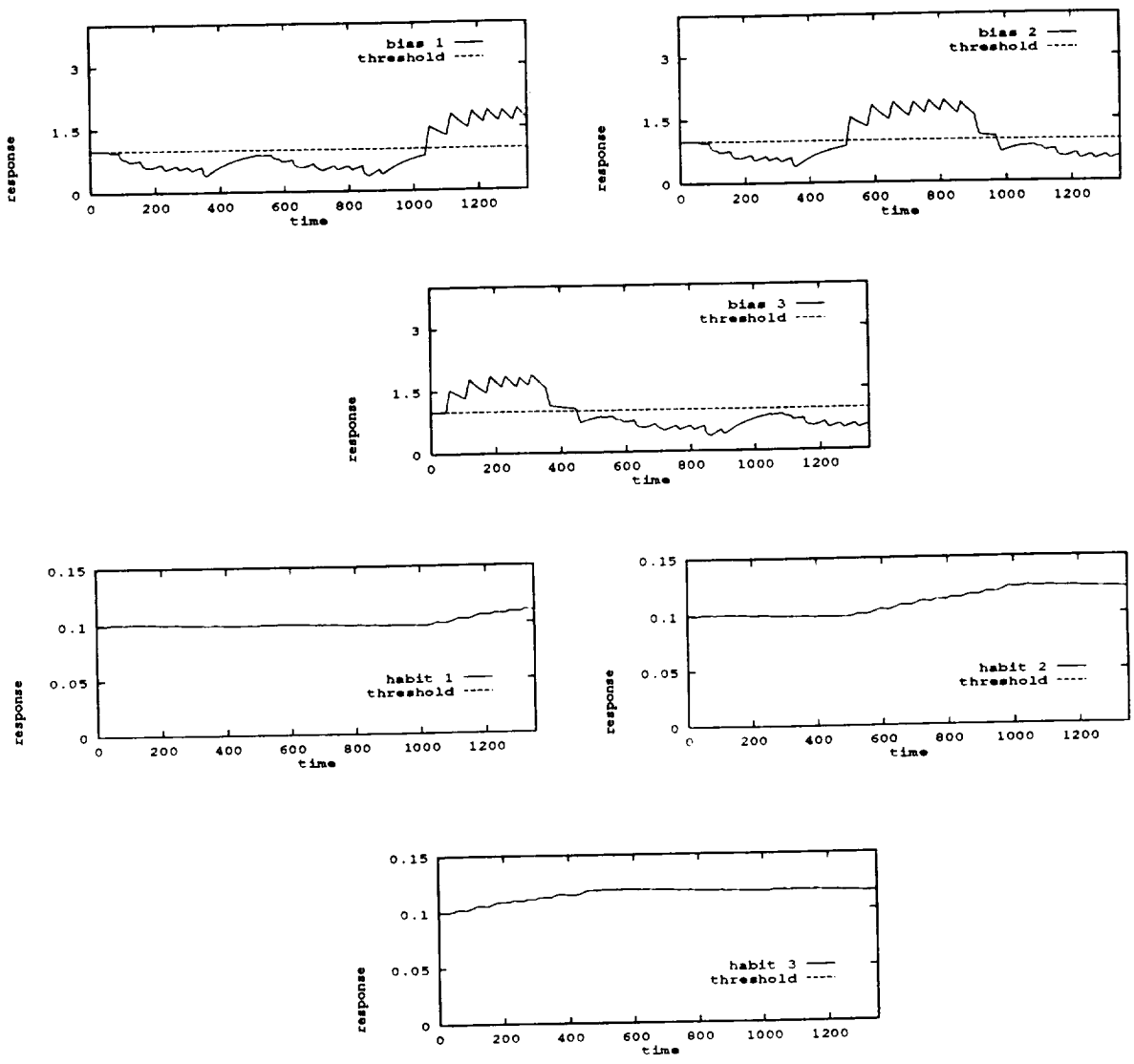

Figure 22: The top three panels plot the activities of the *bias* neurons. Initially the network categorizes using feature 3 as the criterion which is indicated by the activity of *bias* neuron 3 being above threshold. As a consequence of receiving negative reinforcement signals at a later time the internal criterion of the network changes. This is illustrated by a drop in the *bias* neuron 3 activity and in the increase in the activity of *bias* neuron 2. The internal criterion of the network is further changed to feature 1 by issuing negative reinforcement signals at a later instance. The bottom three panels plot the activities of the *habit* neurons. As the number of times a particular criterion is used by the network to categorize the input object the activity of appropriate *habit* neuron increases. As can be seen from the three plots initially the activity of *habit* neuron 3 increases followed by *habit* neuron 2 and finally *habit* neuron 1.

3.5 Spatial novelty and attentive scanning in FRONTAL

The self-organizing autonomous robotic system discussed in the section 2.3 consists of four subsystems: the behavioral categorization system, the object novelty system, the spatial novelty system and the attentive scanning system. The behavioral categorization system and the object novelty network function in a similar manner as the networks discussed in the previous subsections, hence simulation pertaining to these are not presented. In this section simulations dealing with attentive scanning and the spatial novelty system are presented. These two systems together enable the robot to "explore" for novel objects, as well as scan attentively, the objects present in the environment. The lower panel in Figure 23 shows the sequence of presentation of inputs to the network. Three inputs, placed at three different spatial locations, are presented successively to the network. The upper three panels show the activities of neurons representing these three spatial locations in the layer where the suprathreshold activity of a neuron indicates the spatial focus of attention of the robot. Following the introduction of the first input, the robot starts to scan this input. When the second and third inputs are introduced, the robot's attention sequentially scans all three inputs. As one can see from the simulation results, after some time the novelty of inputs vanishes and the robot stops scanning the inputs.

4 User-Interface

Two different user interfaces are provided for interacting with the robot. The first of these two user interfaces is a menu driven interface that can be invoked from any standard terminal. The interface is evoked when the FRONTAL simulation receives a user generated interrupt signal (Cntrl-C is the default interrupt signal). The menu provides means to change reinforcement signals as well as to monitor various variables of the simulation. The second user interface is a X-based interface providing a graphics based environment. The X-based user interface consists of three windows: two for displaying the states of the system and one which enables interaction with the system. One of the two output windows displays the visual input to FRONTAL as well as the object that FRONTAL is currently attending to for categorization. This window also displays how FRONTAL categorized the object. The second output window displays the processing stages of FRONTAL as it scans, selects, and categorizes the object in its environment. The input window is similar to that discussed in non X-window based user interface. It too provides a menu driven interface that can be invoked by a user generated interrupt.

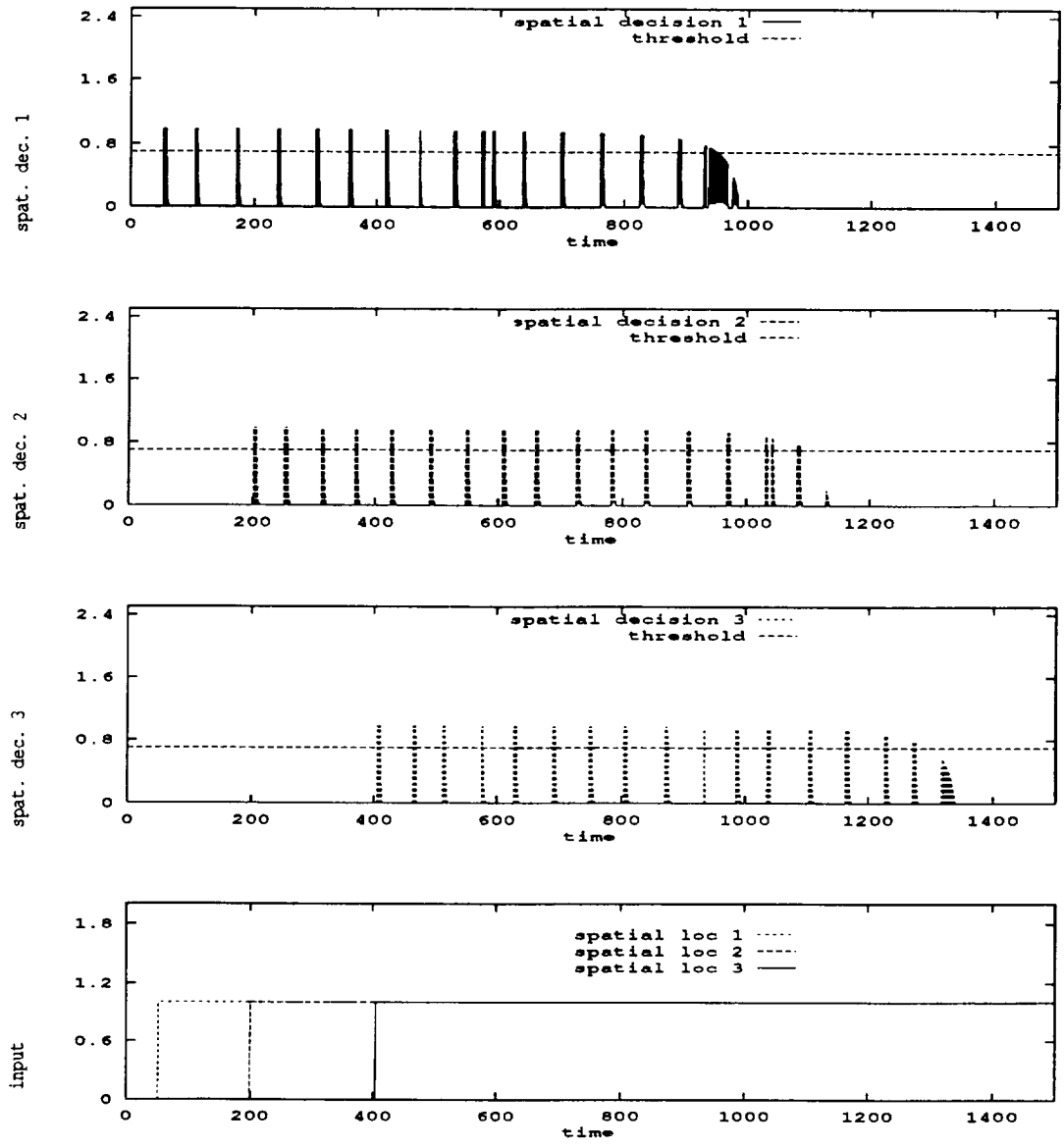

Figure 23: The top three panels illustrate the activity of three *spatial decision* neurons which is indicative of the attention of the network. The last panel indicates the temporal sequence in which three objects are introduced at different spatial locations. The network intermittenly scans these three objects until their novelty wears out.

5 Limitations

The robotic system discussed above had some shortcomings due to the vision and robotic arm system. The vision system was very sensitive to light intensity and the position of the light source. Shadows cast by the object due to different directions of incident light cause BLOBS to error in detecting the object type. Moreover the light intensity also effects the performance of BLOBS.

A major limitation of the robotic arm system was the necessity to recalibrate the arm every time the robotic system was started. This is due to the drift in the potentiometers that calibrate the motors of the arm. Hence for a given spatial target location, different set of arm joint angles are required for the robot arm to reach the target every time the arm system is shut down. This leads to an inconsistent visuo-motor map [9] in the robotic system. Another limitation of the robotic arm is a drift observed in the z-direction as the arm moves along the y-direction. This z-coupling associated with the movement in the y-direction of the arm is shown in Fig 24. As can be seen from the three experimental data shown in the graph, there is about a 0.5 inches drift in the z-direction as the robotic arm moved linearly in the y-direction. Attempts to try to model this non-linear z-coupling did not yield satisfactory results. As the centroids of the objects were more than 3.0 inches away, this z-coupling did not cause problems in realizing which object the robot was intending to grab. However the actual grabbing of the object was not always successful.

6 Conclusion and future work

In this report we presented the details of hardware implementation of a robotic system driven by a adaptive neural network. The main weakness of the robot resides in the traditional algorithmic vision and arm control modules. Our future work consists of replacing these modules by adaptive neural network modules.

[9] The visuo-motor map refers to mapping of a spatial location identified by the vision system to the joint angles required for the robot arm to reach that location.

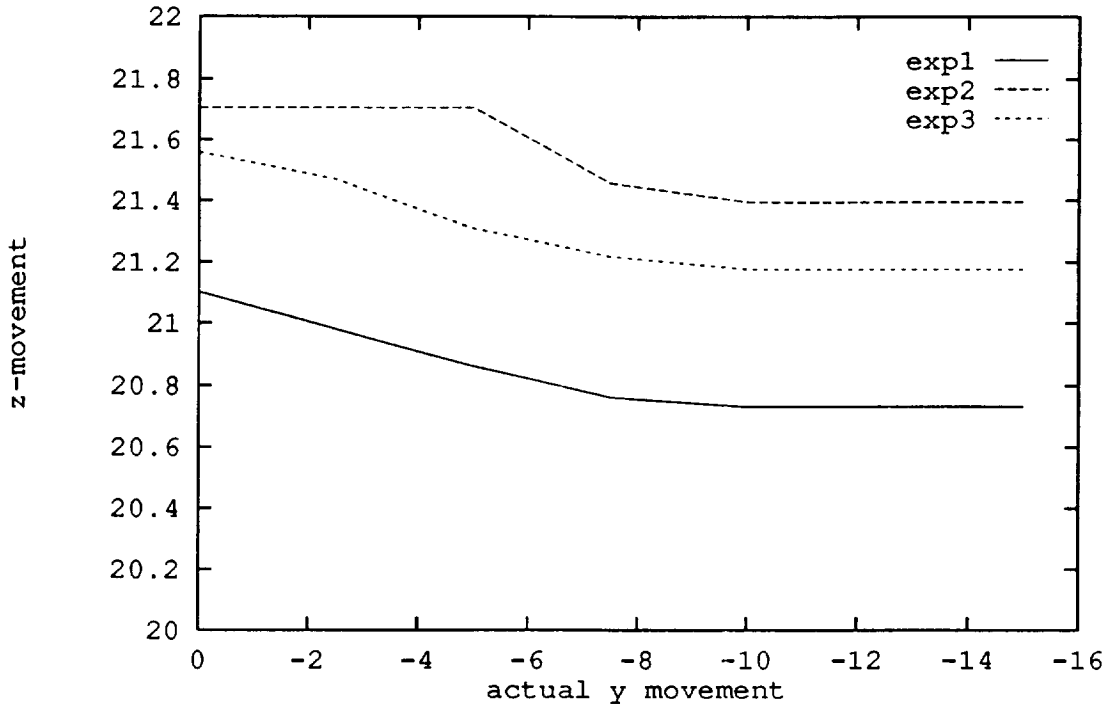

Figure 24: The three figures show the vertical drift of the robot arm as it moves in the horizontal direction. The three graphs represent three different experiments.

A Appendix A

The equations and the parameters used for the various networks are given in this appendix under the various subsections.

A.1 Reinforcement-novelty detection network

The amounts of transmitter in the "ON" and "OFF" channels of the i^{th} gated dipole are described by z_{i1} and z_{i2} respectively (see Figure 9) whose dynamics are given by

$$\frac{dz_{i1}}{dt} = \alpha(B - z_{i1}) - \gamma(I + J_i)z_{i1}. \quad (2)$$

$$\frac{dz_{i2}}{dt} = \alpha(B - z_{i2}) - \gamma I z_{i2}. \quad (3)$$

where α is the transmitter replenishment rate, B is the maximum amount of transmitter, γ is the rate of transmitter depletion, I and J_i are arousal and specific inputs respectively, x_{i1} and x_{i2} are respectively the "ON" and "OFF" channel neurons of the i^{th} gated dipole. They follow the shunting equations

$$\frac{dx_{i1}}{dt} = -Ax_{i1} + (B - x_{i1})(I + J_i)z_{i1} - x_{i1}Iz_{i2}. \quad (4)$$

$$\frac{dx_{i2}}{dt} = -Ax_{i2} + (B - x_{i2})Iz_{i2} - x_{i2}(I + J_i)z_{i1}. \quad (5)$$

where A is a passive decay rate, B is the upper saturation level. The "ON" and "OFF" channel outputs are combined by the neurons in the winner-take-all layer (see Figure 9):

$$\frac{dx_{i3}}{dt} = -Ax_{i3} + (B - x_{i3})(x_{i1} + G_1 w_{i3} r + G_3 f(x_{i3} - \theta)) - x_{i3}(x_{i2} + G_2 w_{ip} p + H \sum_{j \neq i} f(x_{j3} - \theta)). \quad (6)$$

with

$$f(x) = x^2 a(x) \quad (7)$$

where G_1, G_2, G_3, H and θ are positive constants, $a(x)$ is the unit step function. The activities of the reward and punish neurons are given by

$$\frac{dr}{dt} = -A_1 r + (B_1 - r)R + C_1. \quad (8)$$

$$\frac{dp}{dt} = -A_1 p + (B_1 - p)P + C_1. \quad (9)$$

where R and P are reward and punishment inputs respectively and C_1 is a positive constant. The reward weight $w_{i,r}$ and punish weight $w_{p,i}$ follow

$$\frac{dw_{i,r}}{dt} = -A_2(w_{i,r}-1)+(M_r-w_{i,r})B_2 g(r-\theta_1)g(x_{i,3}-\theta). \quad (10)$$

$$\frac{dw_{i,p}}{dt} = -A_2(w_{i,p}-1)+(M_p-w_{i,p})B_2 g(p-\theta_1)g(y_i-\theta_2) \quad (11)$$

with

$$g(x) = xa(x) \quad (12)$$

where A_2, M_r, M_p, θ_1 and θ_2 are positive constants. y_i is the activity of the delayed neuron which follows the following shunting dynamics.

$$\frac{dy_i}{dt} = -A_3 y_i+(B_3-y_i)\gamma_1 g(x_{i,3}-\theta). \quad (13)$$

where A_3, B_3 and γ_1 are positive constants.

A.2 Reinforcement based classification

The dynamics of the *feature* neurons of the modified ART network (see Figure 11) follows the shunting equation and is given by

$$\frac{dx_i}{dt} = -Ax_i+(B-x_i)(I_i+\sum_{j=1}^{1} f(y_j)z_{j,i})-x_i(\sum_{j=1}^{1} f(y_j)\mathcal{I}). \qquad i=1,2,.....12. \quad (14)$$

with

$$f(x) = \frac{1}{1+e^{-150x-0.5}} \quad (15)$$

where A is the decay constant, B is the upper saturation level, I_i is the input applied to the network and $\mathcal{I}$ is the reset signal. $z_{j,i}$ is the top down weight from the *category* neuron y_j to the *feature* neuron x_i as shown in Figure 11. The activity of y_j is as follows

$$\frac{dy_j}{dt} = -Ay_j+(B-y_j)(f(y_j+[v-\theta_2]^+ c_j+\sum_{i=1}^{12} g(\Omega_{[\frac{i+1}{4}]}x_i)z_{i,j}) - y_j(\sum_{r\neq j} f(y_r)+\mathcal{I}). \qquad j=1,2,3,4. \quad (16)$$

with

$$[x]^+ = xu(x) \tag{17}$$

where A, B, θ_2 are positive constants. $z_{i,j}$ is the bottom up weight of the ART network. Ω_i is a *bias* neuron and n is the *ambiguity* neuron with a random weight c_j to the *category* neuron y_j. The *bias* neuron activity which integrates both habits and external reinforcement signal is given by

$$\begin{aligned}\frac{d\Omega_k}{dt} &= -E(\Omega_k - \theta_3) + \{(F - \Omega_k)([h_k - \theta_1]^+ + \alpha[R]^+ + g(\Omega_k)) \\ &\quad -\Omega_k(\alpha[R]^- + G\sum_{r\neq k} g(\Omega_r))\}f(\Phi_k) \qquad k = 1, 2, 3.\end{aligned} \tag{18}$$

with

$$[x]^- = -xu(-x) \tag{19}$$

where E, F, G, α, θ_1, θ_3, are positive constants and R is the external reinforcement signal. The dynamics of the h_k the habit neuron and the Φ_k the match signal is as follows:

$$\frac{dh_k}{dt} = Hh_k\{(J - h_k)[\Phi_k - \theta_2]^+ - [\Phi_k - \theta_2]^-\} \qquad k = 1, 2, 3. \tag{20}$$

$$\begin{aligned}\frac{d\Phi_k}{dt} &= -A\Phi_k + (B - \Phi_k)\{\sum_{j=3k-3}^{3k}\sum_{i=1}^{I} I_j g_0(p_i - \theta_1) z_{i,j}\} \\ &\quad -\Phi_k \mathcal{I} \qquad k = 1, 2, 3.\end{aligned} \tag{21}$$

where H, J and θ_1 are positive constants. I is the reset signal. The neurons in the decision layer following the categorization layer (see Figure 11) are known as p_i and their activities are given by.

$$\frac{dp_i}{dt} = -A_1 p_i + (B_1 - p_i)Wy_i - p_i W\sum_{j\neq i} y_j \qquad i = 1, 2, 3, 4. \tag{22}$$

where A_1, B_1 and W are positive constants. Finally the dynamics of the ambignity neuron which plays a role in biasing one of the category neurons y_j under ambignous situations is as follows

$$\frac{dn}{dt} = -An + (B - n)\{\sum_{i=1}^{12} I_i\} - n\{\Upsilon\sum_{j=1}^{4} g_0(p_j - \theta_1) + I\} \tag{23}$$

with

$$g_1(x) = a(x - 0.05) \tag{24}$$

where A, B, and Υ are positive constants.

A.3 The combined FRONTAL network

As the FRONTAL is a combination of the two networks described so far, the differential equations for various parts of FRONTAL are similar to those presented above. These similarities will be referred to, so as to reduce repetition. The *spatial novelty* network (see Figure 7) comprises of an array of gated dipoles similar to novelty network described before. The differential equations for this network are as given below

$$\frac{dcx_{i,1}}{dt} = -Acx_{i,1} + (B - cx_{i,1})(I + J_i)cz_{i,1} - cx_{i,1}Icz_{i,2}. \tag{25}$$

$$\frac{dcx_{i,2}}{dt} = -Acx_{i,2} + (B - cx_{i,2})Icz_{i,2} - cx_{i,2}(I + J_i)cz_{i,1}. \tag{26}$$

$$\frac{dcz_{i,1}}{dt} = \alpha(J - cz_{i,1}) - \gamma(I + J_i)cz_{i,1}. \tag{27}$$

$$\frac{dcz_{i,2}}{dt} = \alpha(J - cz_{i,2}) - \gamma Icz_{i,2}. \tag{28}$$

where the prefix c in the variable name imply that these gated dipole neurons are related to visual novelty network. Since the introduction and the removal of objects from the robots environment constitute a novel event both the "ON" and "OFF" channels are presented as excitatory inputs to the winner-take-all layer. Further more the winner-take-all layer receives inhibitory input from the *attentive scanning* system (I_i) to enable attentive scanning. The dynamics of the winner-take-layer neurons are as given below

$$\frac{dcx_{i,3}}{dt} = -Acx_{i,3} + (B - cx_{i,3})(C_1cx_{i,1} + C_2cx_{i,2} + G_3f(cx_{i,3} - \theta)) - cx_{i,3}(MI_i + H\sum_{j \neq i} f(cx_{j,3} - \theta)). \tag{29}$$

$$\tag{30}$$

where A, B, M, G_3, H and θ are positive constants. The *decision* layer neurons p_i filters transients in the winner-take all layer and the activity of a neuron in this layer is as follow

$$\frac{dp_{i,1}}{dt} = -Ap_{i,1} + (B - p_{i,1})Wcx_{i,3} - p_{i,1}W\sum_{j=i} cx_{j,3} \qquad i, j = 1, 2...15. \tag{31}$$

where A, B and W are positive constants.

The *attentive scanning* layer consists of the *arousal* neuron f, the *inhibiting* layer neurons s_i and the *delay* layer neurons l_i. The dynamics these neurons is as follows

$$\frac{df}{dt} = -A_f f + (B - f).Arousal - fH \sum_i g_i p_{i,2} \qquad j = 1, 2. \tag{32}$$

$$\frac{ds_i}{dt} = -A_s s_i + (B - s_i)G_1 g(p_{i,1} - \theta_1) - s_i H_1 g(f - \theta_2) \qquad i = 1, 2 \ldots 15. \tag{33}$$

$$\frac{dl_i}{dt} = -A_l l_i + (B - l_i)G_2 g(s_i - \theta_2) \qquad i = 1, 2 \ldots 15. \tag{34}$$

where A_f, A_s, A_l, B, H, G_1, G_2, θ_1, θ_2 and $Arousal$ are positive constants.

The *behavioral categorization* network which categorizes the input objects is similar to the modified ART model discussed earlier. The equations for this network are as follows

$$\begin{aligned}\frac{dbx_i}{dt} &= -Abx_i + (B - bx_i)(I_i + \sum_{j=1}^{1} f(by_j)bz_{j,i}) \\ &\quad -bx_i(\sum_{j=1}^{1} f(by_j) + \mathcal{I}). \qquad i = 1, 2 \ldots 6.\end{aligned} \tag{35}$$

$$\begin{aligned}\frac{dby_j}{dt} &= -Aby_j + (B - by_j)(f(by_j) + \sum_{i=1}^{12} g(\Omega_{\lceil \frac{i+3}{3} \rceil})bx_i)bw_{i,j} + [a - \theta_1]^+ bc_j) \\ &\quad -by_j(\sum_{r \neq j} f(by_r) + \mathcal{I}). \qquad j = 1, 2, 3.\end{aligned} \tag{36}$$

$$\begin{aligned}\frac{dbz_j}{dt} &= -Abz_j + (B - bz_j)(f(bz_j) + \sum_{j=1}^{3} g(by_j)bw_{i,j}) \\ &\quad -bz_j(\sum_{r \neq j} f(bz_r) + \mathcal{I}). \qquad j = 1, 2.\end{aligned} \tag{37}$$

$$\begin{aligned}\frac{d\Omega_k}{dt} &= -E(\Omega_k - \theta_3) + \{(F - \Omega_k)([h_k - \theta_1]^+ + aR^+ + g(\Omega_k)) \\ &\quad -\Omega_k(aR^- + G\sum_{r \neq k} g(\Omega_r))\}f(\Phi_k) \qquad k = 1, 2.\end{aligned} \tag{38}$$

$$\frac{dh_k}{dt} = Hh_k\{(J - h_k)[\Phi_k - \theta_2]^+ - [\Phi_k - \theta_2]^-\} \qquad k = 1, 2. \tag{39}$$

$$\begin{aligned}\frac{d\Phi_k}{dt} &= -A\Phi_k + (B - \Phi_k)\{\sum_{j=3k-2}^{3k}\sum_{i=1}^{3} I_j g_i(p_i - \theta_1)z_{i,j}\} \\ &\quad -\Phi_k \mathcal{I} \qquad k = 1, 2.\end{aligned} \tag{40}$$

$$\frac{dp_{i,2}}{dt} = -Ap_{i,2} + (B - p_{i,2})bz_i - p_{i,2}(\sum_{j=i} bz_j + \mathcal{I}) \qquad i = 1, 2 \tag{41}$$

$$\begin{aligned}\frac{da}{dt} &= -Aa + (B - a)\sum_k p_{k,3} - a(\Upsilon \sum_j p_{j,2}\mathcal{I}) \\ &\quad j = 1, 2. \qquad k = 1, 2, \ldots, 9.\end{aligned} \tag{42}$$

where A. B. $\mathcal{I}$. E. F. G. Υ. R. θ_1 and θ_2 are positive constants. b_{ij} is the further classification of the categorized inputs into "good" and "bad" objects. The decision layer neurons $p_{2,j}$ inhibit the ambiguity neuron if a behavioral decision is achieved.

The object novelty network comprises of an ART network coupled to a novelty detection network via a layer of slowly integrating neurons q_i. The dynamics of this network are given below. The equations for the ART network are as follows

$$\begin{aligned}\frac{df x_i}{dt} &= -A f x_i + (B - f x_i)(I_i + \sum_{j=1}^{12} f(f y_j) f z_{j,i}) \\ &\quad - x_i(\sum_{j=1}^{12} f(f y_j) + \mathcal{I}). \qquad i = 1.2....6. \end{aligned} \tag{43}$$

$$\begin{aligned}\frac{df y_j}{dt} &= -A f y_j + (B - f y_j)(f(f y_j) + \sum_{i=1}^{6} g(f x_i) f z_{i,j}) \\ &\quad - f y_j(\sum_{r \neq j} f(f y_r) + \mathcal{I}). \qquad j = 1.2.....12. \end{aligned} \tag{44}$$

$$\frac{dp_{i,3}}{dt} = -A p_{i,3} + (B - p_{i,3}) b y_j - p_{i,3}(\sum_{j \neq i} b y_j + \mathcal{I}) \qquad i = 1.2.....12. \tag{45}$$

$$\frac{dq_i}{dt} = -A_q q_i + (B - q_i) p_{i,3} \qquad i = 1.2.....12. \tag{46}$$

where A. B. I_i. $\mathcal{I}$i. $z_{i,j}$ and $z_{j,i}$ positive constants. The differential equations for the novelty network are as follows

$$\frac{dcx_{i,1}}{dt} = -A cx_{i,1} + (B - cx_{i,1})(I + q_i) cz_{i,1} - cx_{i,1} I cz_{i,2}. \tag{47}$$

$$\frac{dcx_{i,2}}{dt} = -A cx_{i,2} + (B - cx_{i,2}) I cz_{i,2} - cx_{i,2}(I + q_i) cz_{i,1}. \tag{48}$$

$$\begin{aligned}\frac{dcx_{i,3}}{dt} &= -A cx_{i,3} + (B - cx_{i,3})(cx_{i,1} + G_3 f(cx_{i,3} - \theta) + G_3 p_{1,2}) \\ &\quad - cx_{i,3}(cx_{i,2} + H \sum_{j \neq i} f(cx_{j,3} - \theta) + G_3 cp_{2,2}). \end{aligned} \tag{49}$$

$$\frac{dcz_{i,1}}{dt} = \alpha(J - cz_{i,1}) - \gamma(I + J_i) cz_{i,1}. \tag{50}$$

$$\frac{dcz_{i,2}}{dt} = \alpha(J - cz_{i,2}) - \gamma I cz_{i,2}. \tag{51}$$

$$\frac{dp_{i,1}}{dt} = -A p_{i,1} + (B - p_{i,1}) W cx_{i,3} - p_{i,1} W \sum_{j \neq i} cx_{j,3} \qquad i, j = 1.2...4. \tag{52}$$

where A. B. G_1. G_3. H. I. J_i. θ and W are constants.

The values for the various parameters used in the above differential equations a re given in the following

B Appendix B

The MaxVideo system consists of the following five modules:

- Analog Scanner (AS)
- Architectural Adapter (AA)
- Analog Generator (AG)
- Advanced Pipeline Processor (AP)
- Arithmetic Unit (AU)

The Analog Scanner Module (AS module) is the video input device for the MaxVideo system. It comprises of three sections: (i) analog section, (ii) digital section and (iii) timing section. The analog section enables the imaging system to select from 4 possible input sources. It is capable of DC or AC coupling the input signal and low pass filtering it to avoid aliasing. This section can also adjust the gain and offset of the signal as well perform DC restoration. The digital section digitizes the preprocessed analog input signal with 8-bit resolution at rates upto 26MHz. The digitized images are output through 3 8-bit ports to the AA module. The timing section of this module is responsible for synchronizing the working of the other two sections. The synchronizing clock signal for this section can come from one of three possible sources. An external clock signal generated by a camera or a sensor, or the horizontal or composite sync from the camera, or any arbitrary clock can be used.

The Architectural Adapter Module (AA module) is the mother board of the MaxVideo system. It is the only board which connects to the VME bus directly. It is responsible for routing the raw digitized image via the various modules for processing and displaying. The AA module is thus capable of both data path control as well as intermediate storage of the image between processing. The 6 memory modules part of the AA module acts as source and sink locations for images being processed. The crosspoint switch, whose 32 input connections can be connected to 32 output connections, enables the appropriate routing of the image stored in the memory modules via MaxVideo modules and back to the memory modules. The appropriate connection can be programmed using the ImageFlow software. Images stored in various memory locations can be transparently accessed over the VME bus during the acquisition or display of the image.

The Analog Generator module (AG module) of the MaxVideo Video system is responsible for converting the processed digital data to a variety video format. This module accepts digital data in one of five image display modes depending on the output data precision (the output data precisions supported by the MaxVideo system have been stated earlier). The five image display modes that can be selected are as follows.

- *8-bit Greyscale.* Image Memory Modules 0. 1. or 2.
- *8-bit Pseduocolor.* Image Memory Modules 0.1. or 2.
- 24-bit RGB (8:8:8). Image Memory Modules 0. 1 and 2.
- 8-bit True Color (3:3:2). Image Memory Modules 0.1. or 2 and
- 15-bit True Color (5:5:5). Image Memory Modules 0 and 1.

The two bracketed quantities represent the manner in which the data stored in the Image Memory modules are mapped to represent the color-values. As each Image Memory module can store only 8 bit planes. 16 and 24 bit plane images require the use of more than one Image Memory module. The Display Timing Generator generates the appropriate sync and blanking signals for the variety of display video output.

The Advanced Pipeline Process module (AP module) consists of three processing devices which enable the modules to peform a variety of operations on images. The first of these devices is a statistical processor which is capable of providing 24-bit histogram results on 8-bit plane image data. This device is also capable of detecting up to 512 features in a 512 × 512 pixel image and perform a modified Hough transform on a image to find locations having features with a given angle. Four (8 × 8) bank look-up tables are provided which need to be used in conjunction with the latter two tasks to store the features and the angles to be detected in the image. Also the four banks can be used for generic look-up table. The second device called the NMAC can be used in two modes. In the first mode. it performs a neighborhood 8 × 8 multiply and accumulate which can used for performing convolution of the image with a preset mask. In the second mode. the NMAC can be used as a 2 separate 8 × 4 NMAC's. This split mode in conjunction with a LUT table can be used to perform Sobel edge detection in near real-time. The third and final device is a 16 × 16 LUT that can perform morphological operations on a 3 × 3 binary neighborhood. This device is capable of producing a 16 × 16 bit output that consists of a 3 × 3 neighborhood of all the pixels around the current pixel in the input binary image.

The fifth and final module of the MaxVideo system is the Arithmetic unit (AU device). This AU device has of five sections: (i) Input section. (ii) Binary Crosspoint section. (iii) Grey Scale Crosspoint section. (iv) Output section. (v) Linear Processor section and (vi) Non-Linear Processor section. The Input section takes 8/16 bit two's complement data and converts it to 10/20 bit two's complement data. This 10/20 bit data can then be routed to the Linear and Non-Linear Processor via the Binary and Grey Scale Cross point sections depending on whether the image is a binary or grey scale. 20 bit images are handled by routing 2 (10-bit) paths. The Linear Processor can perform a variety of linear operations which include addition and multiplication of the 10/20 bit image data image data streams. The Non-Linear Processor consists of 6. 10-bit

ALUs or 3 20-bit ALUs and receives 4-10 bit from the Grey Scale Crosspoint section. The Binary Crosspoint section connects to the Non-Linear Processor enabling binary images to control the selection of ALU operation on the 10/20-bit images. This important property of the Non-Linear Processor enables certain binary properties of the image to regulate the processing that needs to be done on it.

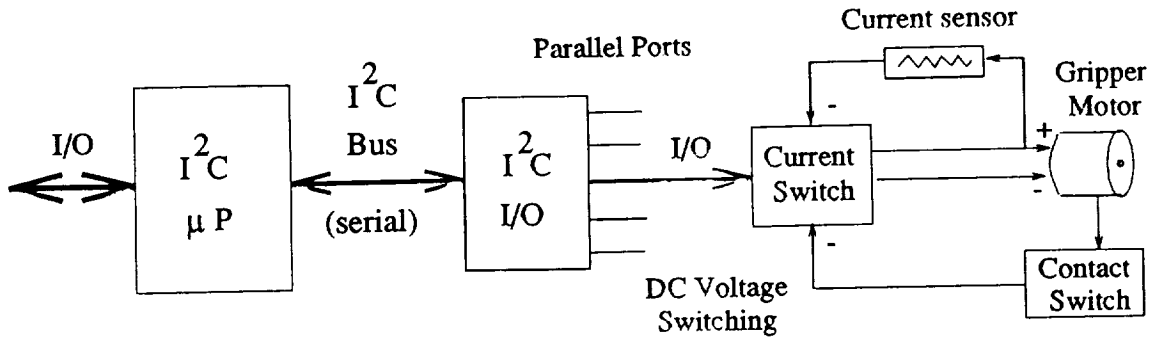

Figure 25: The schematic of the control of the gripper. On receiving a command to close, the motor starts to close the digits of the gripper. As the gripper grasps the object the current sensor circuit ensures that an adequate grasp of the object is achieved without damaging the object, by limiting the amount of current allowed. When the motor is commanded to release the object the motor starts moving the digits of the gripper until the contact switch feedbacks a stop signal.

C Appendix C

A schematic of the motor control mechanism for the two digit gripper is shown in Figure 25. The I^2C microprocessor in conjunction with current switch controls the opening and closing of the gripper by applying appropriate voltage polarity to gripper motor. The current switching circuit provides a feedback to the current switch of the amount of current received by the motor while it grasping an object. The contact switch on the other hand sense when the digits of gripper have reached it maximal open position and intimates the current switch. The microprocessor communicates with a request issuing computer via a serial line. On receiving a request the to close or to open the gripper, the microprocessor sends the appropriate voltage to the gripper motor via its parallel port. When a grip of an object is requested, the digits of the gripper close on to the object. As the force exerted by the digits on the object being gripped reaches a preset value the current sensing circuit prevents further force being applied by the digits by restricting the current to the preset value. Similarly during the opening of the digits, the contact switch turns off the current applied to the motor once the maximal opening is reached.

References

[1] Craig. J. J.. *Introduction to robotics: mechanics and control.* Addison-Wesley: Mass. 1989.

[2] Leven. S. J.. and Levine D. S.. Effects of reinforcement on knowledge retrieval and evaluation. *Proceedings of the First International Conference on Neural Networks. San Diego: IEEE/ICNN.* vol. II. pp. 269-279. 1987.

[3] Levine D. S.. and Prueitt P. S.. Modeling some effects of frontal lobe damage-novelty and perseveration. *Neural Networks.* vol. 2. pp. 103-116. 1989.

[4] Piaget J.. *The origins of intelligence in children.* Norton: New York. 1963.

[5] Piaget J.. *La psychologie de l'intelligence.* Armand Colin: Paris. 1967.

[6] Piaget J.. *The mechanisms of perception.* Basic Books: New York. 1969.

[7] Piaget J.. *Psychologie et épistémologie.* Denoël Gonthier: Paris. 1970.

[8] Stevens R. W.. *Unix network programming.* Prentice Hall: New Jersey. 1990.

Parameters	Novelty Network	Reinforcement Network	FRONTAL Network
A	1.0	10.0	1.0
B	1.0	5.0	2.0
I_i		1.0	1.0
I	1.0		1.0
J_i	1.0		1.0
α	0.01		3.0e-04
hline β	10.0		10.00
γ	0.01		3.0e-04
γ_1	1.0		
C_1	1.0		50.0
C_2			100.0
G		10.0	10.0
G_1	0.05		1.0
G_2	0.5		1.0
G_3	0.5		2.0
θ			0.6
θ_1		0.25	0.9
θ_2		1.0	0.85
θ_3		1.0	1.0
A_1	1.0	0.001	
A_2	0.0001		
A_3	0.005		
B_1	2.0	1.0	
B_2	0.005		
B_3	3.0		
M			5.0
M_c	3.0		
M_p	3.0		
W			500.0
A_f			1.0
$Arousal$			1.0
H	95.0	0.001	10.0
H_1			10.0
A_s			1.0
A_t			0.005
$\mathcal{I}$		10.0	10.0
E		0.01	0.01
F		3.0	3.0
J		3.0	3.0
Υ		10.0	10.0