Robotica (1998) volume 16, pp. 127–141. Printed in the United Kingdom © 1998 Cambridge University Press

Self-organization via active exploration: hardware implementation of a neural robot

Ramkrishna V. Prakash and Haluk Öğmen

Department of Electrical and Computer Engineering, University of Houston, Houston, Texas 77204-4793 (USA)
E-mail: rprakash@uh.edu, ogmen@uh.edu.

(Received in Final Form: July 16, 1997)

SUMMARY

In a previous paper we presented a neural network called FRONTAL that attempted to capture the ability of infants to actively self-organize in a non-stationary environment. This paper describes the procedure whereby FRONTAL is interfaced to a robotic system called DART. DART comprises of a PUMA-562 arm and a real-time image processing system. The implementation procedure consisted of first connecting FRONTAL to a simulated DART robot. After successful operation, FRONTAL was connected to the real hardware DART. The ability of this robotic system to actively explore its surroundings and modify its behavior based on experience and environmental feedback confirms previous simulation results.

KEYWORDS: Neural robot; FRONTAL; DART; Exploration; Behavior modification; Self-organization.

1. INTRODUCTION

In general, two techniques have been adopted for the design of autonomous robots. The first method, mainly used in Artificial Intelligence, is based on the philosophical and psychological tradition of *a priorism.* According to this formalism, knowledge is innate and therefore the robot is designed using *a priori* knowledge of its structure and its environment.[1,2] This approach, though adequate for a well-defined environment, fails to cope with complex or non-stationary environments wherein a fixed and *a priori* description of the domain is not possible.

The second method, used by Artificial Neural Networks, incorporates in the robot the ability to learn through experience. This view finds its roots in the philosophical and psychological tradition of *empiricism,* which states that knowledge is acquired through experience in particular via associations between events that occur in the environment. Simple neuron-like computing elements, capable of forming such associations are used in this methodology and hence the name Artificial Neural Networks. Examining these systems more closely reveals that they do not learn entirely through experience and that there exists an underlying designer who is involved in its structuring. In a complex environment such a passively associating system would be overwhelmed by the combinatorial explosion that can result from possible stimulus combinations in time. Hence, the designer's role in such systems is to choose appropriate strategies to structure the network which may consist of, incorporating *apriori* bias nodes, turning off learning at certain time, or selecting the number of hidden nodes so that the system functions effectively. Such systems however, are essentially continuous memories that are only capable of interpolating and extrapolating between the structures that they encode. Therefore, this approach is not suitable for autonomous systems that need to dynamically modify their internal structure based on their interactions with the environment.

The impetus for our neural network model comes from studies of a system best known for self-organizing: a child. As a child grows, it constantly self-organizes to account for changes in its own structure as well as those of its environment. Developmental studies suggest that the ability of infants to dynamically interact with their environment and self-organize stems partly from their capability to actively explore and partly from their capacity to incorporate environmental feedback into their reasoning.[3–6] Our neural network model, which visually explores its surroundings for novel objects and is capable of dynamically changing its decision criterion based on environmental signals it receives, attempts to capture this ability of infants.

In a previous paper[7] we presented details of this neural network model along with a quantitative analysis of its performance. However, extensive evaluation of the system requires testing the neural network model in a complex non-stationary environment either through software emulation or by interfacing the neural network to a real hardware robotic system. Though software emulation provides the first step towards testing the model, modeling the physical aspects of a non-stationary environment is very complicated and requires enormous computational power. Hence, interfacing the model with a hardware robot provides a means to investigate its performance in a non-stationary environment.

We interfaced our neural network model with a robot consisting of a PUMA 562 arm and a real time vision system. In this paper, the details of this implementation procedure are presented. First, we interfaced our model with a graphical simulation version of the robot system. The functioning of this simulated system was, then,

https://doi.org/10.1017/S0263574798000472 Published online by Cambridge University Press

tested. Next, the model was connected with the hardware robotic system. We then investigated if this robotic system could actively explore its surroundings and dynamically change its decision criterion based on signals it receives from a non-stationary environment. Before we present the details of software and hardware implementations of this neural robot, a brief overview of the neural network, called FRONTAL, is presented. A detailed description along with equations can be found in reference 7.

2. FRONTAL

To actively interact with its environment, a robotic system should to be able to (visually) explore its surroundings, be capable of detecting novel events and also be able to use prior experience to guide its future behavior. These three components, necessary for active self-organization, form the basis of FRONTAL. FRONTAL was designed to actively search and collect "good" objects from its surroundings. Good objects were those that had been correlated with positive reinforcement. Figure 1 presents a schematic of FRONTAL. It consists of the visual exploration, the good-bad categorization, and the object identification subsystems.

The visual exploration subsystem consists of a non-homogeneous retina with a high resolution fovea. Hence, the neural architecture for visual exploration consists of the attentive scanning network that enables the eyes to scan various objects in the surroundings and maintain fixation on a particular object while determining its characteristics. In addition, the spatial novelty network enables the visual exploration subsystem to detect new objects in the surroundings.

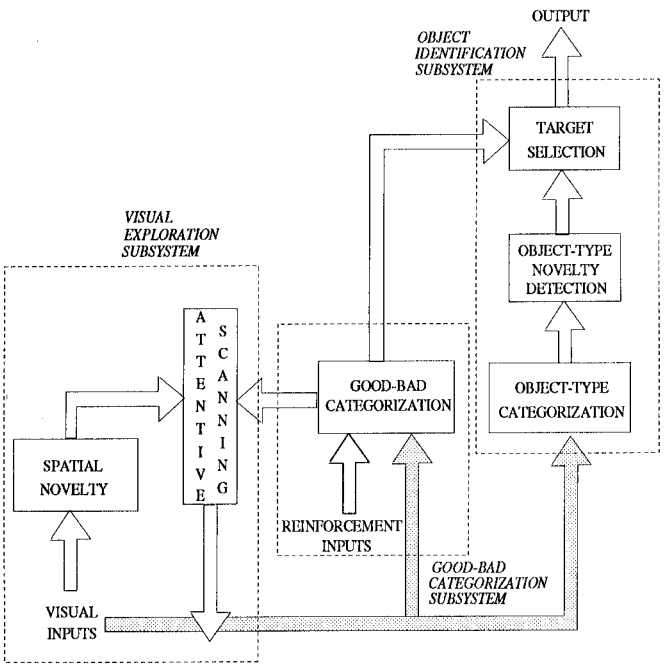


Fig. 1. Block diagram of FRONTAL. It comprises three subparts which are the visual exploration subsystem, the good–bad categorization subsystem and the object identification subsystem. The visual exploration subsystem is involved in locating objects in the environment while the object identification subsystem identifies object-type. The good–bad categorization subsystem evaluates the virtue (good or bad) of the object based on previous experience and together with the object identification subsystem decides on the robots actions (pick or refrain).

Along with the capability to visually explore objects, FRONTAL has the ability to generalize sensorial signals using the good-bad categorization subsystem. This subsystem is a modified ART network[8] capable of dynamically categorizing objects into different stable categories, depending on the object's features and on the "non-specific reinforcing" signals it receives from the environment. Since reinforcement signals, like any signal in a non-stationary environment, can be noisy, certain circuits involved in the formation of habits have been added to the modified ART to filter out the noise in reinforcement signals.

Finally, the object identification subsystem, which consists of the object-type categorization, object-type novelty and target selection networks, is involved in processing object types. The object-type categorization network categories objects present in the robot's environment into different types based on its features, while the object-type novelty network detects *new* types of object. Signals from the good-bad categorization subsystem as well as the object-type novelty network are pooled together at the target selection network, which decides whether to collect a particular object. Thus, FRONTAL, which comprises of networks that enable the robot to explore for novel objects in its surrounding and act based on prior experience and reinforcement signals, provides the basic tenets necessary for an autonomous system. In the following subsection specific details of the three subsystem of the FRONTAL are presented.

2.1 Visual exploration subsystem

Figure 2 is a schematic of the spatial novelty network which consists of an array of gated dipole networks.[9] Each gated dipole network senses inputs from a discrete area in a two-dimensional space. The gated dipole network produces both sustained and transient output signals. The sustained signal indicates the presence of an object in the surroundings, while the transient signal notifies the introduction or the removal of an object from the robots visual environment. The input signals that feed into the gated dipole network are gated by transmitter nodes (the dark boxes in Figure 2).

The gating strength of the transmitter nodes at each spatial location is inversely related to the duration for which an object has been presented at that location. Thus, recent objects generate stronger responses compared to older ones. The combined responses from all the gated-dipoles feed into the "winner-take-all" layer of the attentive scanning network whose schematic is shown in Figure 3. The "winner-take-all" layer, wherein each neuron excites itself and inhibits all the others, ensures that the fovea at any given time focuses on only one object in the environment. The feedback pathway via additional layers of delay and inhibitory neurons,

https://doi.org/10.1017/S0263574798000472 Published online by Cambridge University Press

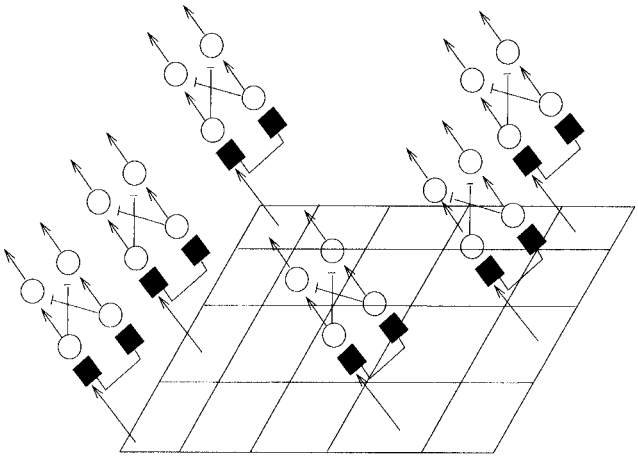


Fig. 2. The spatial novelty detection network consists of an array of gated dipoles each sampling a distinct spatial position. For simplicity, only some of the gated dipoles are shown.

enable the eyes to shift foveation to other objects. Thus, the visual exploration subsystem enables the eye to scan different objects in the surroundings as well as detect new objects that enter its view.

2.2 Good-bad categorization subsystem

The good-bad categorization subsystem, whose schematic is shown in Figure 4, generalizes the objects present in the robot's surroundings into different categories based on the object's features using an ART network (layers F1 and F2).[10] The second ART network (layers F2 and F3) further categorizes these objects as either "good" or "bad". The categorization of objects into good or bad categories is influenced by prior experience of the network, encoded in the habit nodes and the

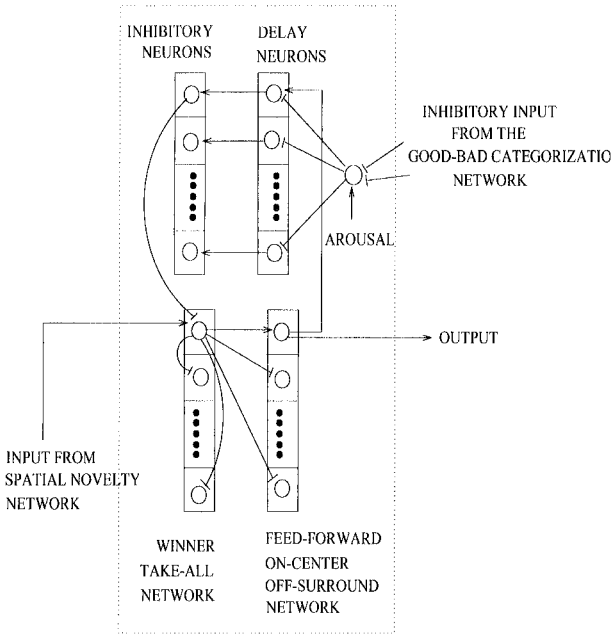


Fig. 3. Attentive scanning network. This network enables the vision system to selectively attend to individual objects in the environment. The inhibitory input from the good–bad categorization network prevents the shift of attention while object is being categorized.

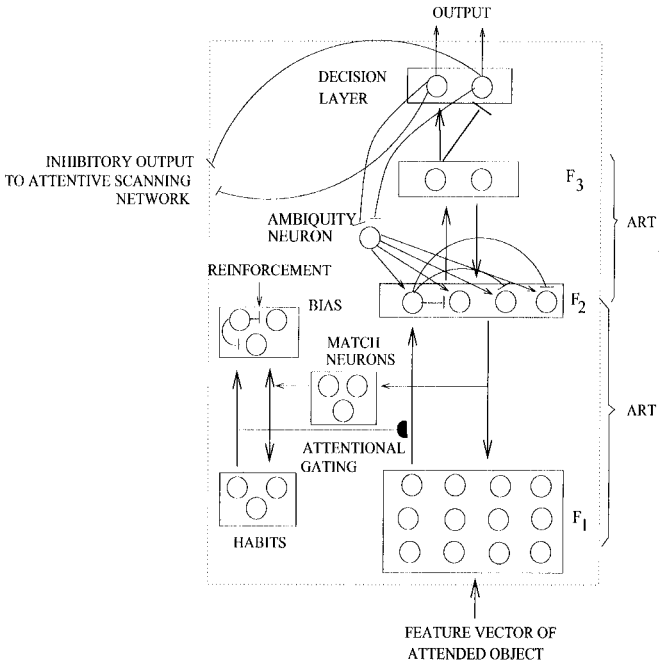


Fig. 4. Good–bad categorization network. Modified from reference 10.

reinforcement signals received from the environment. The reinforcement signals are non-specific and hence, additional nodes are necessary for appropriate credit assignment of reinforcement signals.[10] The match nodes, which encode the criterion that the network used to categorize an object, along with the reinforcement signals enable the network to dynamically modify its internal criterion for categorizing object via the bias nodes. However, certain ambiguous situations may result while the network attempts to categorize an object. The habit nodes along with the bias nodes resolve these ambiguities. To understand this, consider the situation illustrated in Figure 5. The two figures at the top of Figure 5 illustrate the templates of the "good" and "bad" objects. Good objects are those that have been correlated with positive reinforcement (which FRONTAL is supposed to gather), and bad objects are those that have been correlated with negative reinforcement (which FRONTAL should avoid). Before reaching for the object, which is shown at the bottom of Figure 5, FRONTAL should determine if the object is good or bad. Categorizing the object based on any feature alone would result in ambiguity. In order to resolve this ambiguity, the network combines habit and reinforcement signals. Habit nodes encoding the number of times a given feature have been used helps to arrive at a decision. Bias nodes generate activities favoring those features that have been associated with greater number of positive reinforcements during previous interaction of the system with the environment. These two nodes guide the network in selecting the appropriate feature for categorization. If no prior experience regarding the salient feature of an object exists, the "ambiguity" neuron categorizes the object by randomly assigning it to one of the categories. The resulting reinforcement

https://doi.org/10.1017/S0263574798000472 Published online by Cambridge University Press

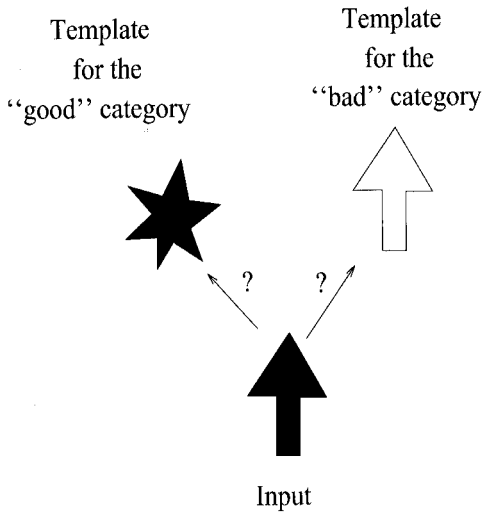


Fig. 5. The input object shown at the bottom of the figure has to be categorized into one of two categories whose templates are shown at the top of the figure. The template at left may be an example for "good objects" (the system will then pick this object) and the template at right may be for "bad objects" 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. Adapted from reference 10.

feedback from the environment due to this action is then used to resolve future ambiguities.

During categorization, competition amongst the nodes in layer F3 can generate transients. If the output from these nodes drive other subsystems, then these transients, could lead the system to unwarranted states. Hence such transients are undesirable for a system which functions in a continuous dynamic environment. A layer of decision neurons (see Figure 4) filters these transients and ensures that a stable output is reached prior to propagating the output signals to other subsystems. The decision layer also feeds into the attentive scanning network so that attention shifts only after the characteristic of the foveated object has been identified.

The good-bad categorization network is thus involved in encoding FRONTAL's experience with its surrounding and in guiding its future behavior.

2.3 Object detection subsystem

The object detection subsystem is involved in detecting novel objects in the environment. Figure 6 is the schematic of this subsystem that consists of the object-type categorization, object-type novelty, and target selection networks. The object-type categorization

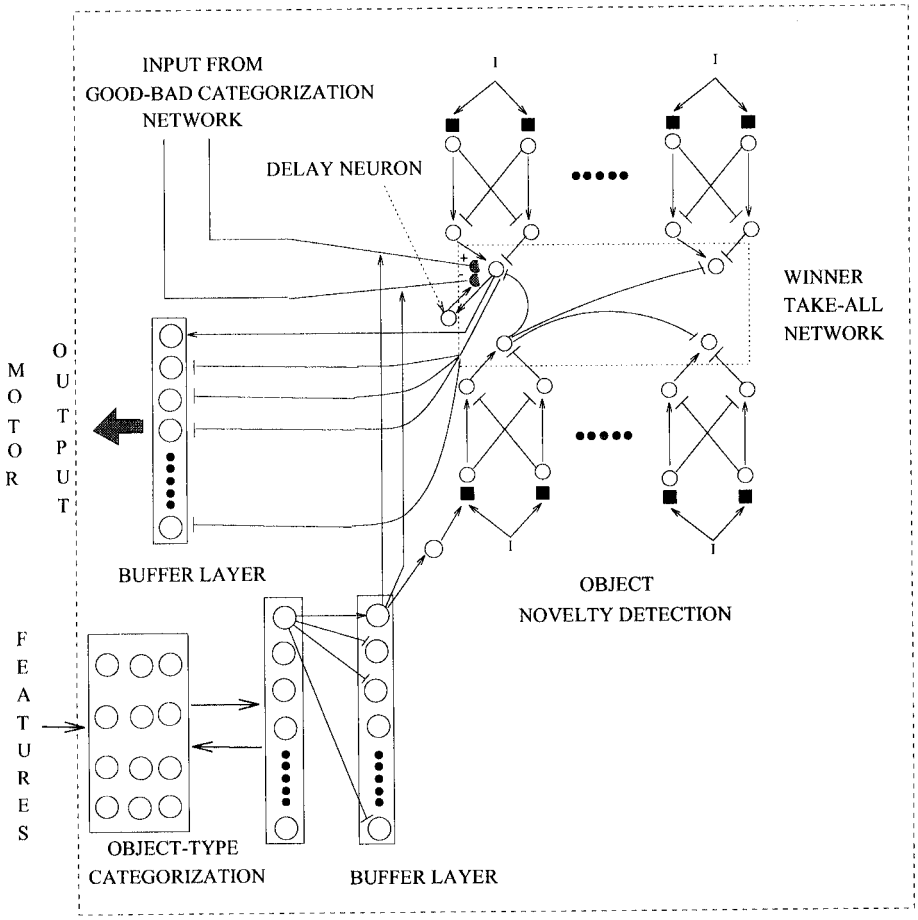


Fig. 6. The object detection subsystem. It comprises of the object-type categorization, object-type novelty, and target selection architectures. This network is adapted from reference 11.

https://doi.org/10.1017/S0263574798000472 Published online by Cambridge University Press

network, which is an ART network, categorizes the different kinds of objects present in the environment. The output from the category nodes of the ART network in turn feed into an array of gated-dipole networks which form the object-type novelty network.[11] Each gated-dipole network represents a type of object present in the environment. The transmitter node that gates the input to the gated-dipole encodes the duration a particular type of object has been present in the robot's surroundings. Hence, novel types of objects yield stronger response compared to familiar objects. Thus, similar to the spatial-novelty network of the visual exploration subsystem, the object-type novelty network identifies new object-types in the environment. The target selection network which is the final part of this subsystem is a "winner-take-all-network".[12] This network weighs the object-type novelty of an object against its characteristic determined by the good-bad categorization network and arrives at the decision to either collect or refrain from the object. This decision is then conveyed to the robotic arm along with the co-ordinates of the objects. The object-type categorization, object-type novelty and target selection networks are thus involved in deciding whether to pick a particular object that FRONTAL saw in its surrounding by weighing the novelty of object's type against FRONTAL's previous experiences with that object. These subsystems that constitute FRONTAL, are simple ingredients that are necessary for a system to explore and actively self-organize in a non-stationary dynamic environment. In the following section, we elaborate on the first step that was undertaken to further evaluate FRONTAL. This involved interfacing FRONTAL to a simulated robotic system and qualitatively studying its behavior in a virtual environment.

3. SIMULATED ROBOTIC SYSTEM

To analyze if FRONTAL can actively self-organize in a dynamic non-stationary environment, we first interfaced it with a simulated robotic system. In the following subsections we present the details of the simulated environment and our simulation results.

3.1 The simulation environment

The simulated robot, called DART (Dexterous Anthropomorphic Robotic Testbed), consisted of a PUMA-562 arm and a pair of cameras which enabled it to view its virtual surroundings. Figure 7 illustrates the simulated robot in its virtual environment where various kinds of objects were periodically presented to it. The arrow between the two eyes illustrates the robot's line of sight. The task that DART performed was to collect "good" objects from its surroundings and refrain from "bad" ones.

The entire virtual environment was simulated on a Silicon Graphics VGX320 work-station. DART as well as the various objects in its surroundings were built using a solid modeling package called Solid Surface Modeller

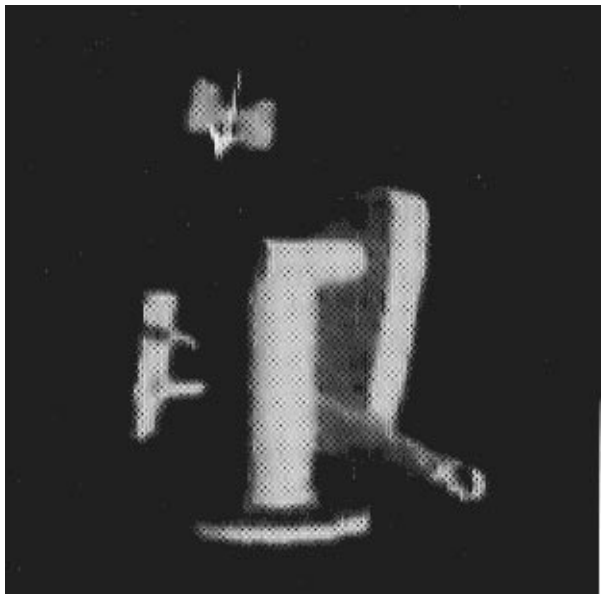


Fig. 7. Simulated DART robot.

(SSM).[13] The animation of the virtual environment itself was generated using a software called Tree Display Manager (TDM).[14] The visual field in front of the robot was segmented into discreet locations using a two-dimensional grid. Each of these discreet locations mapped to a specific gated-dipole node of the two-dimensional spatial novelty network of FRONTAL (cf. Figure 2). The simulated camera provided FRONTAL with the features and the location of the object that was currently being focussed on by DART.

FRONTAL ran on an Amdahl supercomputer and communicated with the simulated robotic system using sockets.[15] Figure 8 presents the overall schematic of the various processes involved in the simulation and their interdependence. FRONTAL, was modeled by a system of ordinary differential equations (ODE) which were solved using a numerical ODE solver, based on Runge-Kutta-Fehlberg 4–5 method developed by Oak Ridge Labs. These equations along with their parameters can be found in reference 7.

A graphical user interface (GUI) developed using X-window was also incorporated to allow user interaction with the simulated environment and

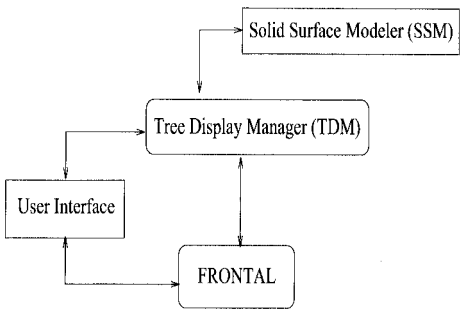


Fig. 8. A schematic illustrating the various interacting components associated with the qualitative study of FRONTAL with the simulated DART robot.

https://doi.org/10.1017/S0263574798000472 Published online by Cambridge University Press

FRONTAL. On interrupting a simulation all state variables of the network were pushed onto the stack of the computer while the interrupt was handled. One could then introduce or remove objects from the virtual environment or provide external reinforcement signals to the system. On returning from the interrupt, the previously stored state-variables were reloaded and the network continued from the internal state prior to the interrupt. Thus, the FRONTAL system and the virtual environment were modeled to mimic a continuous real-time dynamic system.

3.2 Simulation results

In this simulation study, the robotic system categorized objects as "good" and "bad" based on two features: color (red or yellow) and kind (tool or non-tool). It was assumed that red colored objects were good while yellow objects were bad, if color was the criterion for classification, while tools were good and non-tools were bad, if the criterion for classification was the kind of object. Though the simulated environment was in color the photographs illustrating the results of this simulation (Figures 9–13) are in black and white. Note that the yellow objects can be distinguished from the red objects by their lighter appearance. Also when punishment or reward signals were issued to the robot for its action, the tip of the robot's arm turned red or green, respectively. This change in the color of the tip of robot has been indicated by text in the relevant (black and white) Figures.

Initially, the robotic system had no notion of the appropriate criterion for categorizing objects. Hence, novelty of the objects caused the robot to reach for them and explore their properties. Figure 9 illustrates one such instance when the robot reached for a yellow (light gray) tool in its surrounding. In this and the following figures, the arrow in between the two eyes of the robot shows the robot's direction of sight. On reaching for the yellow (light gray) tool the robot got punished as shown in Figure 10a. The red coloring at the tip of the robot arm (pointed out by the arrow) indicates punishment while green coloring of the tip represents reward. Another object, a red (dark gray) tool, when presented caused the robot to reach for it as well due to its novelty. However, on reaching for the red (dark gray) tool, the robot got rewarded (see Figure 10b and note that the tip of the arm is green). Subsequently, when a red (dark gray) non-tool and a yellow (light gray) tool was introduced, the robot reached for the red (dark gray) non-tool, but refrained from the yellow (light gray) tool (see Figure 11). Thus, the robot learned that color was the criterion for categorizing objects as good or bad and hence it categorized red objects as good and yellow objects as bad.

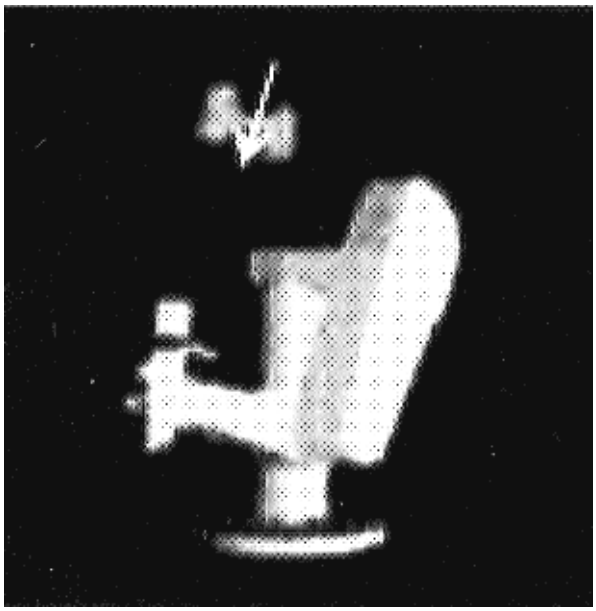


Fig. 9. Novelty driven reaching.

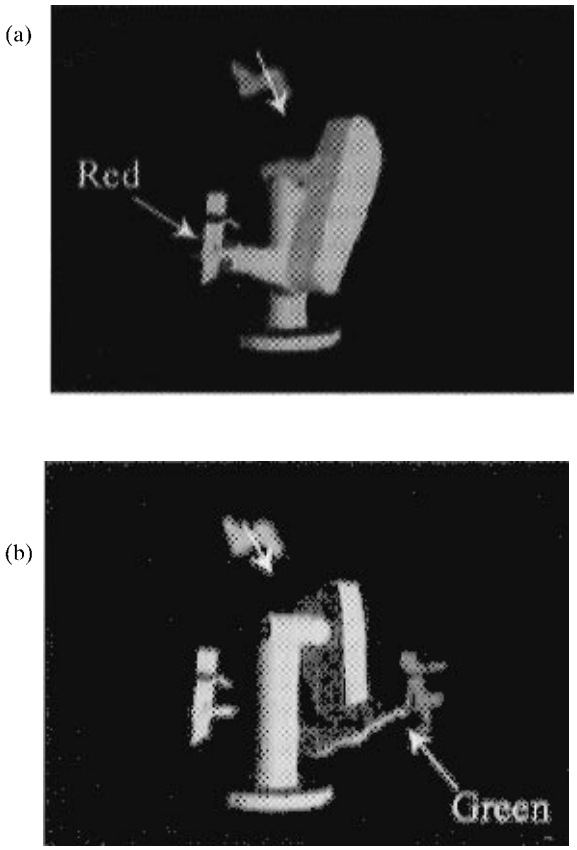


Fig. 10. Issuing "reward" and "punishment" signals causes DART to select color as the criterion to categorize objects as good or bad.

After the robot had successfully learned to categorize objects based on color, we wanted to study if the robot was capable of changing its classification criterion based on the reinforcement signals it received. Hence, from then on we rewarded the robot only when it picked up tools and punished it when it picked up non-tools. Therefore, when the robot reached for the red (dark

https://doi.org/10.1017/S0263574798000472 Published online by Cambridge University Press

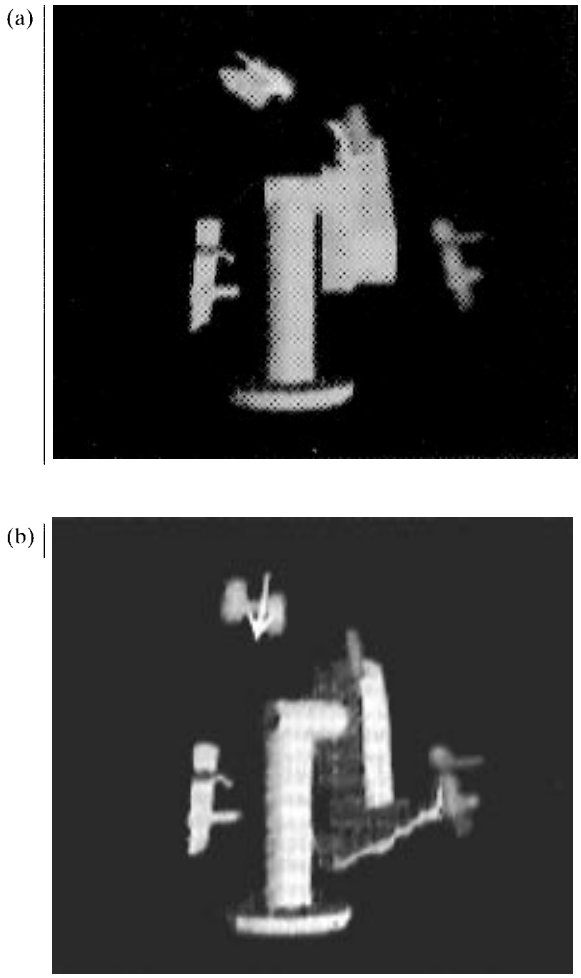


Fig. 11. DART reaches for the red rod but refrain from the yellow object illustrating that it has learnt to use color as the categorization criterion.

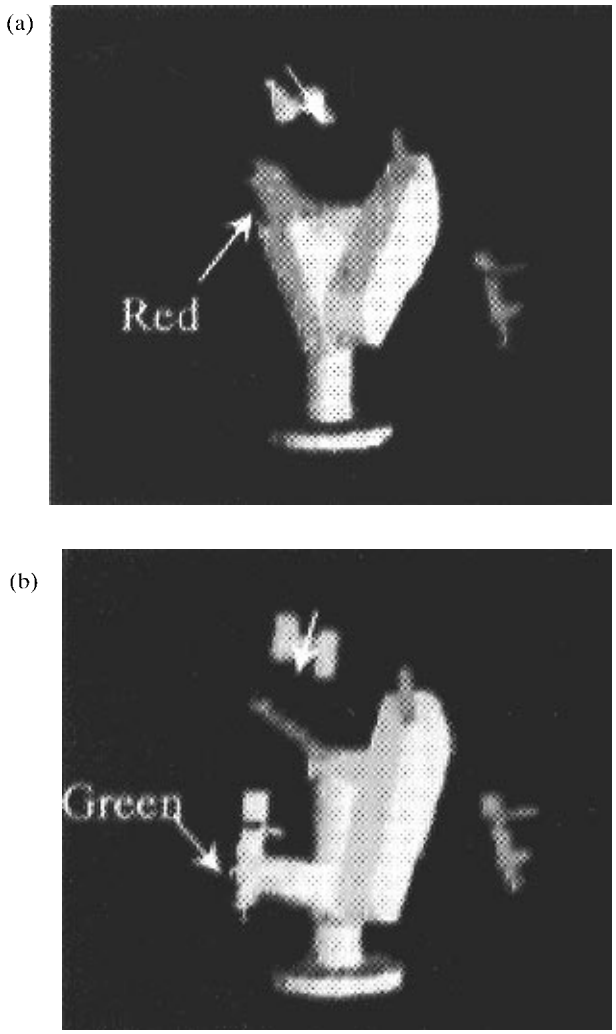


Fig. 12. Assignment of reward and punishment causes DART to change it decision criterion from color to kind.

gray) spring it got punished (see Figure 12a). After a few attempts to reach for non-tool objects (where it was punished) the robot reached for the yellow (light gray) tool for which it was rewarded (see Figure 12b). After five reinforcing sequences, where the robot was rewarded if it reached for tools and punished if it reached for non-tools, the robot changed its categorization criterion from color to kind. Figures 13a and 13b demonstrate this, where the robot refrained from the red (dark gray) bar, but on seeing the red (dark gray) tool the robot reached for it.

The initial inertia of the robot to change its internal categorization criterion from color to kind, when it received punishment, was due to FRONTAL's capacity to weigh new environmental signals against prior experience. As FRONTAL realized that the environmental signals were consistent, it modified its internal criterion for categorization to adapt to the environment.

After the successful testing of the *simulated* robotic system, the next step we performed was to interface FRONTAL with the hardware version of DART robot. In the next section we present the details of this operation along with a description of the hardware system and software protocols used.

4. THE HARDWARE IMPLEMENTATION

The overall schematic of the hardware robotic system is shown in Figure 14. It consists of FRONTAL, the DART robot hardware, the communication protocols that enabled FRONTAL to communicate with DART, and a X-window based GUI (similar to the one discussed in the previous section), that enabled the user to provide external signals to the robotic system.

In the following subsections a description of the hardware and results of the simulation are given.

4.1 The hardware robot

In this subsection, a description of the vision and arm subsystems of the DART robot are given. The communication protocols used to interconnect DART and FRONTAL are also described.

https://doi.org/10.1017/S0263574798000472 Published online by Cambridge University Press

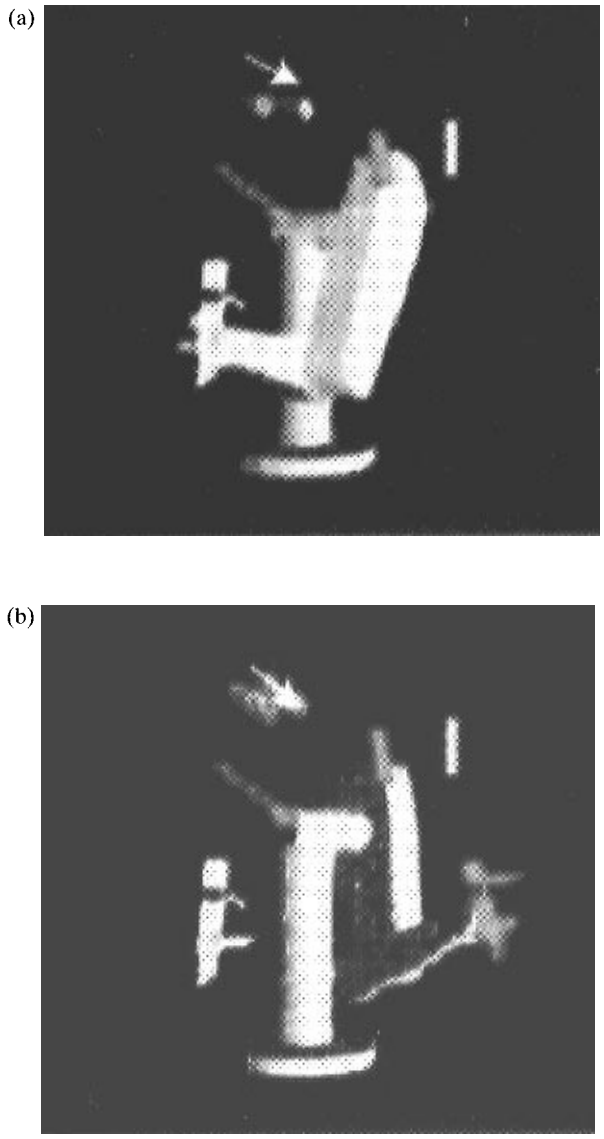


Fig. 13. (a) Refraining from the red rod, and (b) Reaching for the red tool. Thus DART has successfully modified its classification criterion from color to identifying if the object is a tool or a non-tool.

4.1.1 Vision System. The vision subsystem consisted of two primary components: A real-time image processing system, called Datacube MaxVideo System 20, and a real-time object recognition software called, BLOBS. The MaxVideo system, capable of acquiring, processing and displaying images at a rate of 30 frames/sec, was programmed using a Sun Sparc 2 host computer. An object-oriented software, called Image-flow, resident on the host computer, was used to program the MaxVideo system to grab and threshold images in real-time. These thresholded images were simultaneously made available to BLOBS and were also displayed on a television monitor simultaneously.

The MaxVideo system consisted of five modules. The first module, called the Analog Scanner (AS), was capable of capturing raw images in a variety of formats. The second module called the Architectural Adapter (AA), was used to store raw as well as processed images. Two other modules called the Advanced Pipeline Process (AP) and the Arithmetic Unit (AU) were used for processing the raw images. A variety of operations ranging from thresholding to convolution of raw images with a predefined mask could be accomplished, in real-time, using these modules. The final module called the Analog Generator (AG) was responsible for displaying the processed images, in real-time, in a variety of formats.

In our implementation of the vision system, only the AP module was used in conjunction with the AS, AA and AG modules. Figure 15 shows the overall setup of the MaxVideo 20 system. As can be seen from the figure, the MaxVideo 20 system was configured in two different modes (pathways) called PATs show in Figure 15A and 15B respectively. The AS module, which received a multisync signal from the CCD camera routed the raw image alternatively through these two PATs. The first PAT, shown in Figure 15A, consisted of the raw image going directly from the AS module through the AA module and the AG module to the monitor. In this configuration MaxVideo 20 system displayed the captured raw image directly to the monitor. In the second PAT configuration the raw image was first thresholded by the AP module and then sent through the AA and AG modules to be displayed on the monitor. The AP module was configured to threshold the raw image using a look-up table. The raw image was intermediately stored by the AA module during processing. In this configuration, the MaxVideo system thresholded the raw image it captured from the CCD camera before displaying it on the monitor. The MaxVideo 20 system was configured to toggle between these two PAT's every other 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 affect the performance of the robotic system. When the MaxVideo 20 system operated in the second mode, the threshold image was read into the host computer from the AA module. Simultaneously, the thresholded image was also sent to BLOBS, the object recognition software. The thresholded image captured by the computer was then processed by BLOBS to locate the different objects in the image in real-time. Any object that was detected, was identified and its location and type information were transmitted to FRONTAL.

The object detection software, BLOBS, used a blob detection algorithm;* which grouped neighboring pixels of similar value into a single blob (or object). It also detected the edge pixels of a blob. The area and the perimeter of the objects were computed by the total number of pixels in the blob and the number of edge pixels respectively. Three different kinds of objects had to be identified by BLOBS hence a compactness measure given by,

$$compactness = \frac{perimeter^2}{area}, \qquad (1)$$

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

https://doi.org/10.1017/S0263574798000472 Published online by Cambridge University Press

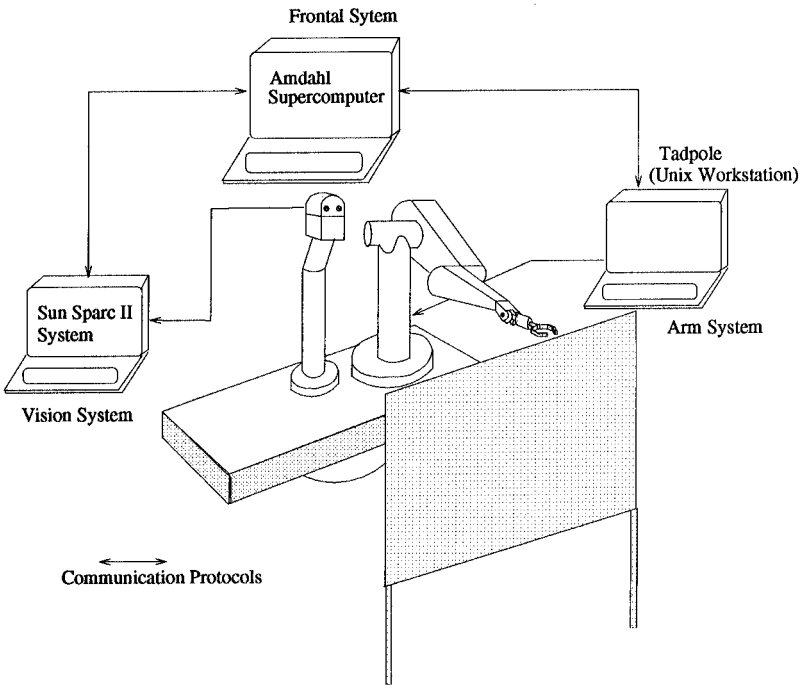


Fig. 14. The hardware robotic system. It comprises of the following four systems: (i) the vision system (ii) the arm system (iii) FRONTAL implemented on an Amdahl computer (iv) the communication protocols indicated by arrows and (v) the X-window based GUI. The communication protocols and the GUI's are incorporated in the software associated with the Vision, Arm and FRONTAL systems.

was used by BLOBS to differentiate them. An advantage of this measure is its invariance to scale. The image captured by the CCD camera was distorted due to the aspect ratio (the height to width ratio) of the pixels of the camera was equal to 0.75. To avoid any inconsistencies in the measurement of the area and the perimeter of the objects, as they were rotated about an axis perpendicular to the camera, BLOBS scaled each pixel by the aspect ratio. Nine different objects of three different sizes (small, medium, large) and shapes (circle, equilateral triangle, star) could be reliably detected by the vision subsystem. Figure 16 gives a summary of the

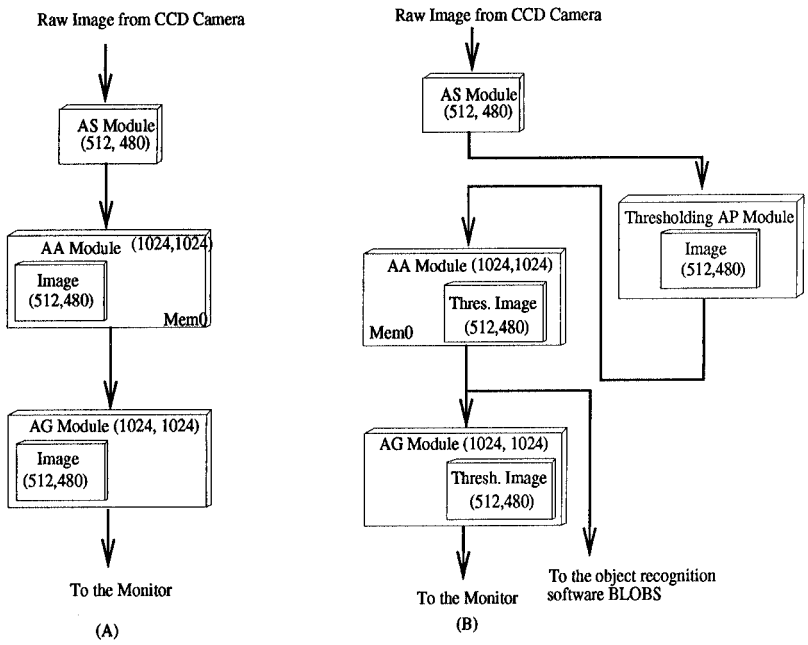


Fig. 15. The two modes of operation of MaxVideo 20 which part of the Vision subsystem represented by figures (A) and (B) respective. In mode (A) the raw image is directly output to the monitor while in mode (B) the raw image is thresholded and then displayed on the monitor. Both can be processed by each module. The size of our images were (512, 480).

https://doi.org/10.1017/S0263574798000472 Published online by Cambridge University Press

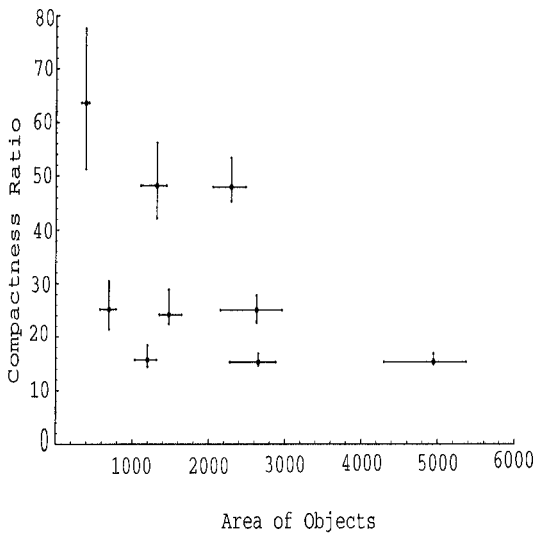


Fig. 16. This figure plots the mean (dot) and variances (lines) of the area and compactness ratio of the different objects presented to the robot. The plot shows that the vision system was able to uniquely identified nine different types of objects based on area and compactness ratio.

performance of the vision subsystem. The vision subsystem was tested twenty times for each type of object. The average compactness ratio and the area of the object, as well as their ranges are plotted in Figure 16. For any given type of object, there was no overlap between the sizes and their compactness ratio. Thus, the vision subsystem can uniquely identify the nine different types of objects presented in the robots environment.

As previously stated in Section 2.1 the objects presented to FRONTAL by the vision subsystem were from a two-dimensional visual space which was at a fixed depth from the robot. To investigate whether the vision subsystem performed adequately we gathered the above presented data from various points in this visual space. Figure 17 shows a scatter plot of the various locations in the visual field at which the objects were placed for

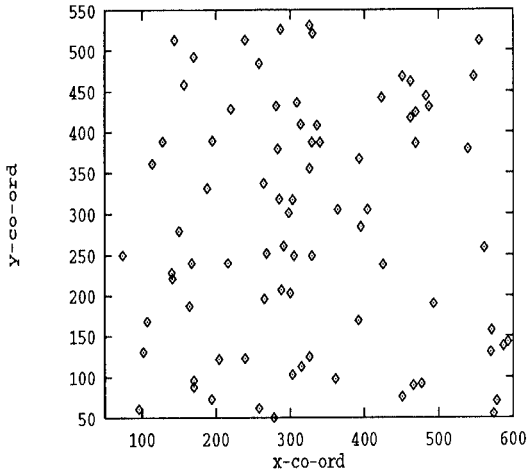


Fig. 17. The various locations at which objects were placed for studying the performance of the Vision subsystem. The plot shows that the identification of the object type by the vision system was independent of the location at which the object was presented.

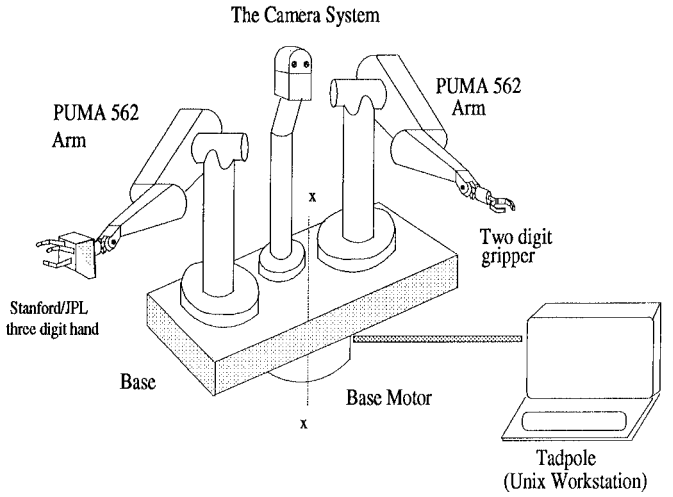


Fig. 18. The Arm System comprised of two PUMA 562 arms and was controlled via an UNIX workstation called Tadpole.

testing the vision system's performance. The random scatter of the locations suggests that the vision system can reliably identify the type of object independent of its spatial location.

4.1.2 Arm System. The Dexterous Anthropomorphic Robotic Testbed (DART), developed a NASA-JSC[16] and shown in Figure 18, consists of two PUMA-562 arms, a 3-digit Stanford/JPL dexterous hand on the right arm, and a gripper on its left arm. Each of the two PUMA-562 arms has 6 degrees of freedom. A Cybernetics servo controller consisting of three CPUs controls the joints of each of the arm. The joints are controlled using Position Derivative (PD) based servo loops. The controller provides position, velocity and torque control of each joint. It is also possible to hold the robot arm at any desired joint configuration. Commands to the controller can be sent via a software running on an UNIX workstation called Tadpole. This software, called ARMS, communicates with the controller through shared memory. The state-variables such as the joint position and torque at each of the joints of the robot arms are written every 10 millisecs, into the shared memory, by the controller. ARMS constantly monitors the contents of the shared memory thus enabling a near real-time feedback control of the robot arm. ARMS also computes the inverse and forward kinematic solutions for the PUMA 562 arm in order to move the arm to specific locations in space as instructed by FRONTAL.

In our present study, only the left hand and gripper were used. When FRONTAL arrived at the decision to pick a certain object, the position of the object was conveyed to ARMS. ARMS, then used inverse kinematics equations for a PUMA-562 Arm and computed the joint angles the robot arm should assume in order to reach the object. The trajectory that the arm would follow to reach the target was obtained by generating a set of intermediate values for each joint. These values were determined by interpolating between the current and final positions for each joint. ARMS then

https://doi.org/10.1017/S0263574798000472 Published online by Cambridge University Press

wrote these values successively into the shared memory. On reaching the desired final location, ARMS activated the gripper to grab the object. After grabbing the object ARMS moved the arm back to its home position and released the object it had picked.

Effective functioning of the self-organizing robotic system necessitated the need for reliable and efficient communication between FRONTAL, the vision subsystem and the arm subsystem. In the next subsection, details of the communication software that enabled this are presented.

4.1.3 Communication Protocols. With the advent of cost-effective, fast, dedicated processors, task-specific computers are now being widely used. Hence, many applications require the development of firmware for communication between these computers. Various standards are available for the development of these communication interfaces. In our implementation we developed a communication software based on Berkeley sockets and TCP/IP (Transmission Control Protocol/Internet Protocol).[15] The software was designed to provide simple but reliable communication between the various subsystems of the robotic system as shown in Figure 19. As can be seen from the figure, parts of the communication software were resident in each of the subsystems. Sockets, which establish dedicated channels for the communicating processes were used to ensure reliable communication between the subsystems. The overall strategy was to allow the vision and arm subsystems to independently interact with FRONTAL.

When a new object was introduced into the robot's view, the vision subsystem immediately conveyed this information to FRONTAL via a socket. On receiving this information, the communication protocol associated with FRONTAL sent back a "confirmatory" signal. Similarly, to initiate a grasp, FRONTAL communicated its decision, along with the co-ordinates of the object, to the arm subsystem. On completing a successful grasp of the object, the communication software associated with the arm subsystem sent back a "success" signal to FRONTAL.

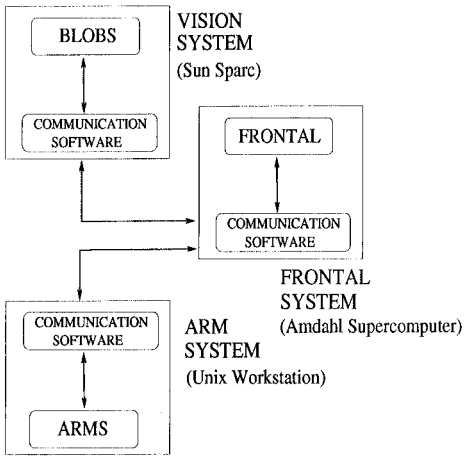


Fig. 19. A schematic of the computers and softwares involved in establishing the communication between the various the robotic subsystem.

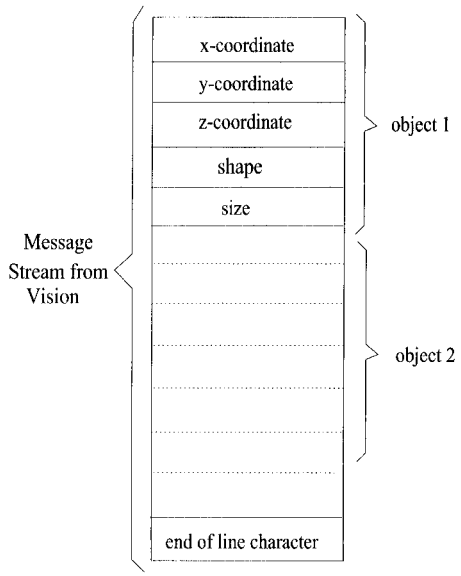


Fig. 20. The message stream from Vision subsystem to FRONTAL.

The message packets used for communication among the subsystems were character strings of arbitrary length, terminated with an end-of-line character. The message packet used by the vision subsystem to communicate with FRONTAL was a concatenation of substrings containing the co-ordinates of and information about the features of objects present in the robots environment. Figure 20 is a schematic of one such message packet of the vision subsystem. Each substring describing an object in the environment comprised of five fields, the x, y and z co-ordinates, the shape and the size of the object respectively. The message packet used by FRONTAL to communicate with the arm subsystem comprised of three fields the x, y and z co-ordinates of the object to be grasped. Figure 21 is a schematic of the message packet sent by FRONTAL to the arm subsystem. The "confirmatory" and "success" signals were a message packet of one byte length whose bits were all set to ones. This initial implementation of the communication software between the various subsystems provided a means to test the functioning of the hardware robotic system. Later, a more sophisticated software package called TELRIP (Telerobotics Internet Protocol) was used to establish interprocess communication.[17]

TELRIP consisted of three processes, the server, the client and the router that ran concurrently. The server subprocesses distributed information sought by the

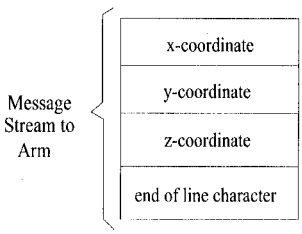


Fig. 21. The message stream from FRONTAL to the Arm subsystem.

https://doi.org/10.1017/S0263574798000472 Published online by Cambridge University Press

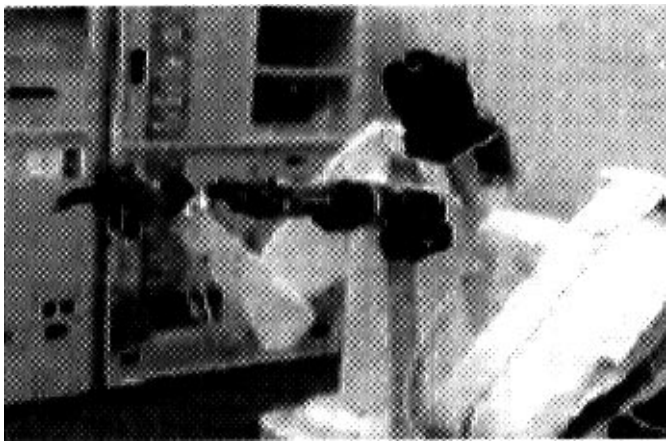


Fig. 22. The hardware DART robotic system.

various client subprocesses via the router subprocess. Thus, TELRIP provided a near real-time environment to transfer data across various processes without concern about the physical machine on which these processes were running. The distributed communication environment of TELRIP thus provided a more realistic implementation of a real-time robotic system.

In this subsection, we summarized the various components of the self-organizing hardware DART robot. In the next subsection we present qualitative simulation results demonstrating the ability of the hardware DART robotic system to reach for novel objects in its environment.

4.2 Simulation results

First, we present a brief overview of the simulation environment. Figures 22, 23 and 24 illustrate the hardware DART robot, the graphical display of the vision subsystem and FRONTAL respectively. The display representing the functioning of the vision subsystem had two parts (see Figure 23). The first part, to the left of the screen, displayed the thresholded image generated by the MV20 image processing hardware. The second part, shown in the right half of the display, presented the results of the object recognition system which identifies the type of the object. The grid represents the spatial resolution of the two-dimensional visual novelty detection network of FRONTAL. Figure 24, represents the GUI which demonstrated the functioning of FRONTAL. It comprised of a schematic of various networks that constituted FRONTAL. We selectively highlighted these networks to indicate the current state of FRONTAL as it processed objects in its environment.

To qualitatively investigate the functioning of the hardware robot, objects were placed on a black board in front of DART as shown in Figure 25. The array of small horizontal and vertical dots placed on the board were used to determine the parameters for the inverse kinematic solution and compensations for the drifts that occurred in the joint counters of the robot arm. These dots were ignored by the visual subsystem. When an object was presented to the robotic system, as shown in Figure 25, the vision subsystem immediately identified the object. Figure 26 illustrates the output that resulted from the MV20 image processing system and the object recognition software BLOBS. The results from the vision subsystem when conveyed to FRONTAL activated the spatial novelty network, as shown in Figure 27a, which in turn resulted in the robot attending to the object. This was indicated, to the observer, on the visual subsystem display (see right half of Figure 26) by placing a square ring around the object. As the robot attended to the object the behavioral categorization network and the

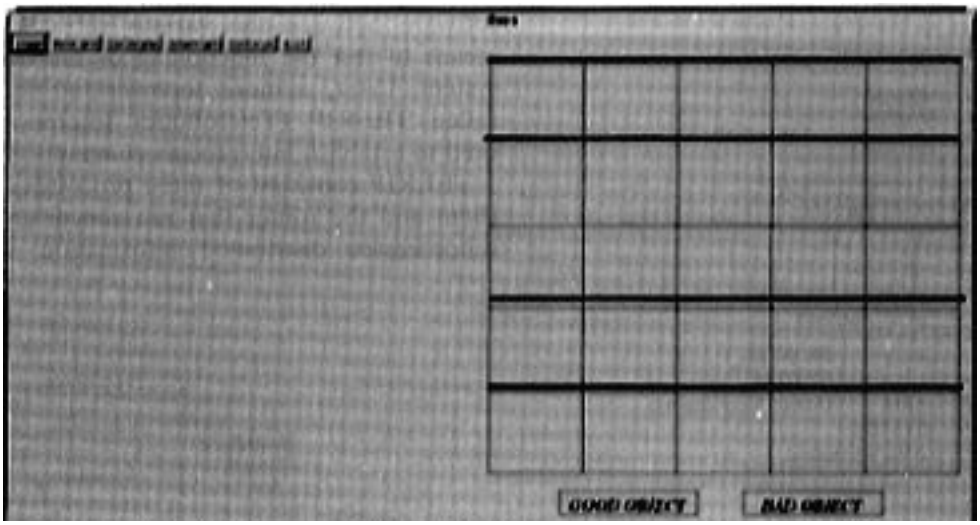


Fig. 23. The graphical display of the Vision subsystem. The left half is the thresholded image output from the MV20 image processing system while the right half displays the output of the object recognition system BLOBS. No object is present in this display.

https://doi.org/10.1017/S0263574798000472 Published online by Cambridge University Press

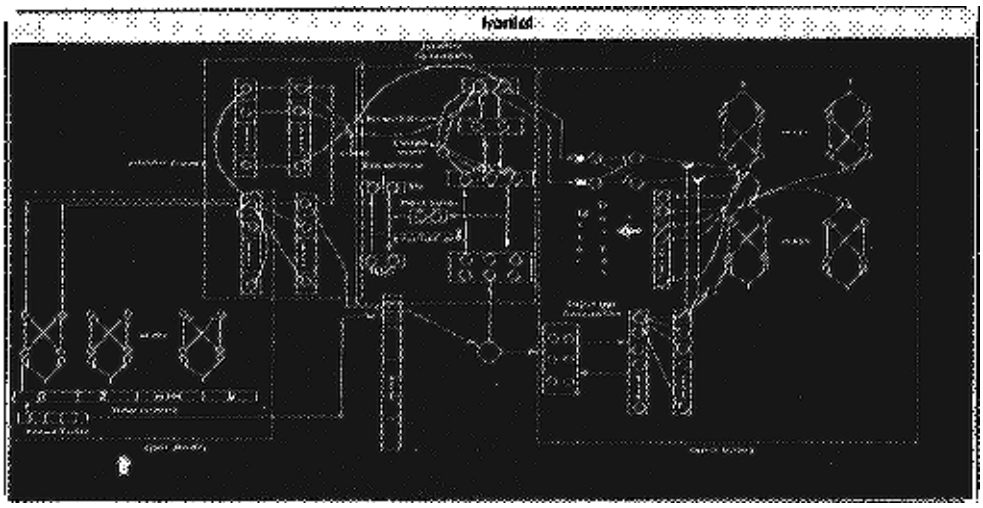


Fig. 24. The graphical display demonstrating the functioning of FRONTAL. The currently active subsystem is highlighted.

object detection subsystems simultaneously processed the characteristics of the object. This is illustrated in Figure 27b where these networks are highlighted. After the characteristics of the object were processed by these two subsystems, FRONTAL decided to grasp the object and sent the relevant message to the arm system. Figures 28a and 28b illustrate the grasp and subsequent release of the object once it returned to its home position, by the robot. As no *a priori* knowledge of the object's characteristics existed, the decision by FRONTAL to reach for the object was based only on the novelty of the object.

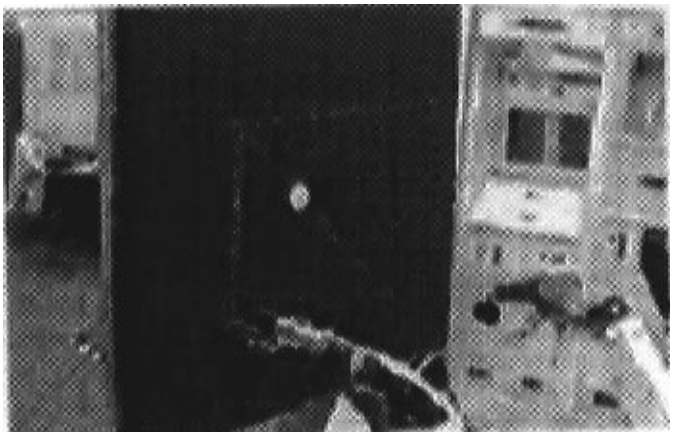


Fig. 25. This figure illustrates the situation when a new object is presented in the robot's environment.

Subsequently, we tested the capability of the *hardware* DART robotic system to change its decision criterion between shape and size based on external reinforcement signals. In our simulations circular objects were deemed as good objects when shape was the criterion and small objects were considered good when size was the as criterion feature respectively for categorization. Table I summarizes our results in which FRONTAL changed its categorization criterion a total of 50 times. The average number of negative reinforcement cycles needed for changing FRONTAL's criterion from size to shape and vice-versa were 7.5 and 8.2 respectively. Prior to deciding to change FRONTAL's criterion for categorization,

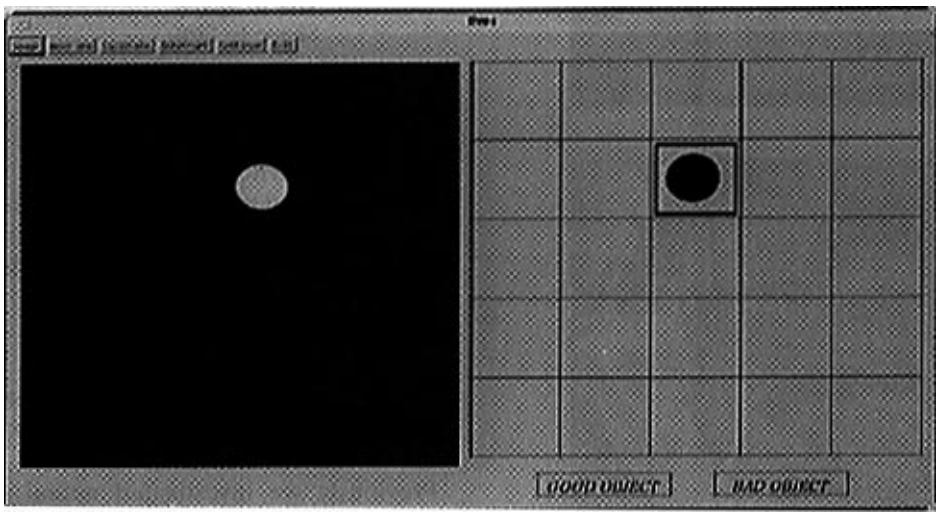


Fig. 26. The Vision subsystem processed the objects present in the robot's environment and sent this information to FRONTAL. Currently the vision subsystem has identified a large circular object in its environment. The left half illustrates the threshold output of the MV20 image processing system while the right half illustrates the output of BLOBS. The solid square ring indicates that the robot is currently focussing on this object.

https://doi.org/10.1017/S0263574798000472 Published online by Cambridge University Press

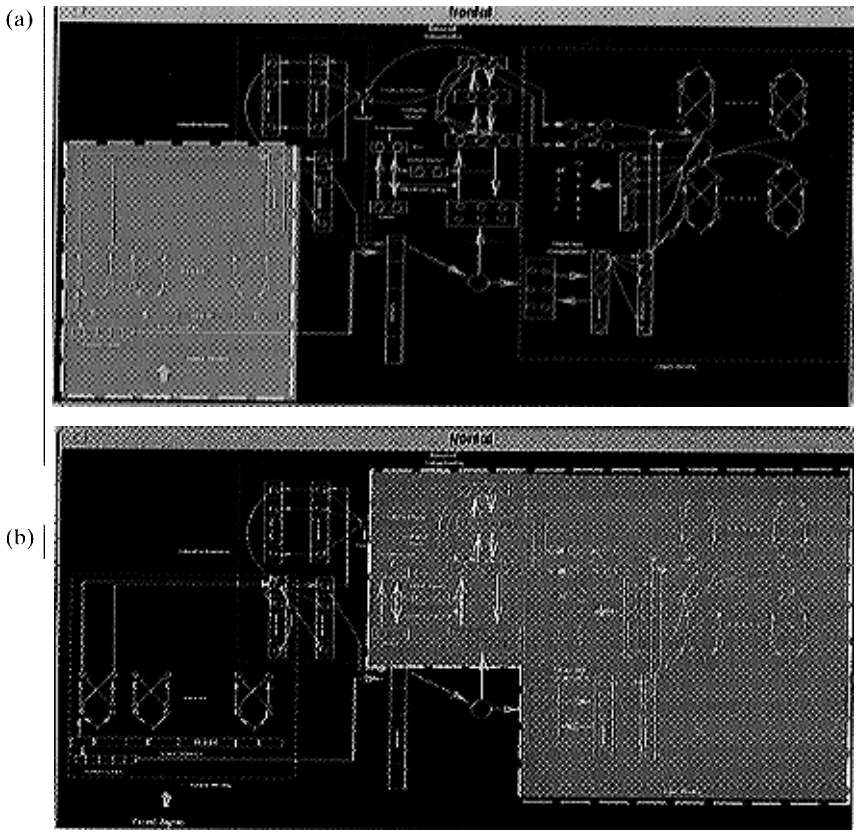


Fig. 27. Once the information on the objects in the robot's environment reach FRONTAL the visual exploration subsystem gets activated causing the robot to focus on the object (a). Subsequently the good–bad categorization subsystem and the object identification subsystems process this object leading to a decision to grab the object (b).

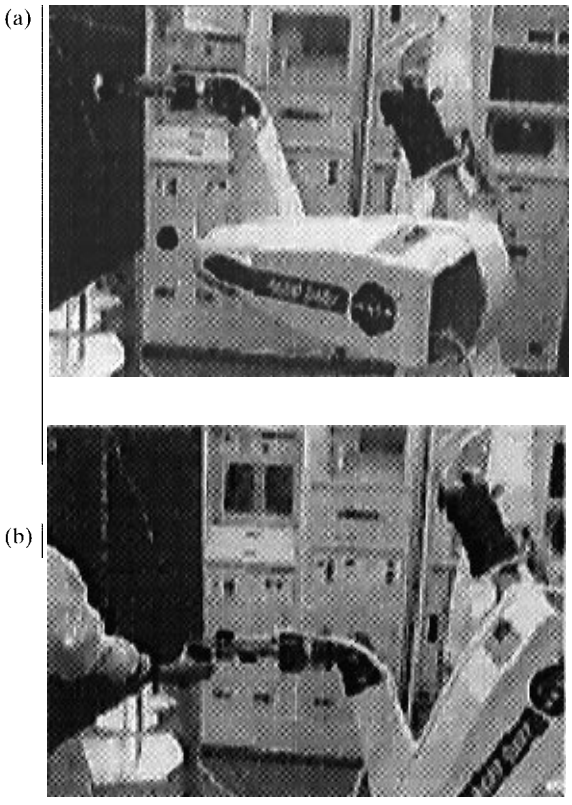


Fig. 28. When ARMS receive the signal from FRONTAL to grab the object, the PUMA arm reaches to grab the target (a). After grabbing the object the arm returns back to its home position where it releases the object (b).

Table I. Summary of criterion change simulations. The number of reinforcement cycles needed to change the criterion feature for categorization. Three positive reinforcement signals were issued after FRONTAL correctly determined the criterion feature.

Change in criterion	Number of negative reinforcements
Size to Shape	7.5 ± 1.7
Shape to Size	8.2 ± 1.2

FRONTAL was rewarded thrice for correctly identifying the relevant criterion.

These simulations thus illustartes the ability of the self-organizing hardware DART robotics system to identify novel objects in its surroundings and dynamically changing its behavior.

5. CONCLUSION

The ability to self-organize is necessary for any system to operate successfully in a non-stationary environment and this should be based on an active exploration process. Previously we investigated neural architectures having novelty sensitivity, selective attention, reinforcement learning, habit formation, flexible criteria categorization properties and analyzed the resulting behavior (consisting of an intelligent initiation of exploration) by computer simulations.[7]

https://doi.org/10.1017/S0263574798000472 Published online by Cambridge University Press

In this paper we described the implementation of this self-organizing neural network, called FRONTAL, with an actual hardware robotics system, called DART. Our first step was to interface, FRONTAL to a graphically simulated non-stationary environment and DART robot. Simulation results demonstrated that the graphically simulated robot was capable of identifying new objects and modifying its decision criterion based on the external reinforcement signals it received. In the second step, we interfaced FRONTAL with the actual hardware DART robot that comprised of a PUMA arm and a real-time vision system. Simulation results, confirming previous quantitative studies of FRONTAL, show that this hardware robotic system too was capable of detecting novel objects and dynamically modifying its behavior based on environmental feedback. Thus, in this paper we presented an integrated robotic system which is our initial step towards the development and evaluation of intelligent self-organizing autonomous systems based on active-exploration.

6. ACKNOWLEDGEMENTS

We would like to thank B. Cox, R. Davis, R. Goode, J. Kowing, L. Li, H. Nguyen, C. Price, B. Rogers, and S. Shelton from the Robotic and Automation Division of NASA-JSC, for their help and suggestions on various aspects of this work. This material is based in part upon work supported by the Texas Advanced Technology Program under grant No. 003652023 and by a grant from NASA-JSC.

References

1. E. Charniak and D. McDermott, *Introduction to Artificial Intelligence* (Addison-Wesley Publishing Company, Massachusetts, 1985).
2. R.J. Schilling, *Fundamentals of Robotics Analysis and Control* (Prentice Hall, New Jersey, 1990).
3. J. Piget, *The Origins of Intelligence in Children* (Norton, New York, 1963).
4. J. Piaget, *La psychologie de l'intelligence* (Armand Colin, Paris, 1967).
5. J. Piaget, *The Mechanisms of Perception* (Basic Books, New York, 1969).
6. J. Piaget, *Psychologie et épistémologie* (Denoël Gouthier, Paris, 1970).
7. H. Öğmen and R.V. Prakash, "A developmental perspective to neural models of intelligence and learning" **In:** *Optimality in Biological and Artificial Networks* (D.S. Levine and W. Elsberry, Eds.) (Lawrence Erlbaum, New Jersey, 1997) pp. 363–395.
8. S. Grossberg, "Adaptive pattern classification and universal recording: I. parallel development and coding of neural feature detectors" *Biological Cybernetics* **23,** 121–134 (1976).
9. S. Grossberg, "A neural theory of punishment and avoidance, II: Quantitative theory" *Mathematical Bioscience* **15,** 253–285 (1972).
10. S.J. Leven and D.S. Levine, "Effects of reinforcement on knowledge retrieval and evaluation" **In:** *Proceedings of the First International Conference on Neural Networks.* IEEE/ICNN Conference Publication 10111, (1987) **vol. II,** pp. 269–279.
11. D.S. Levine and P.S. Prueitt, "Modeling some effects of frontal lobe damage-novelty and perseveration" *Neural Networks* **2,** 103–116 (1989).
12. S. Grossberg and D. Levine, "Some developmental and attentional biases in the contrast enhancement and short term memory of recurrent neural networks", *J. Theoretical Biology* **15,** 253–285 (1972).
13. Integrated Graphics and Operations Analysis Laboratory, "Ssm – solid surface modeler user's guide, version 3.0", *Tech. Rep.* (NASA/Johnson Space Center, Houston, TX, 1991).
14. F.S. Taylor and K.P. Mancus, "Tdm – tree display manager, user manual, version 4.0", *Tech. Rep. TM-5.24.12-04* (McDonnell Douglas, Houston, TX, 1992).
15. R.W. Stevens, *Unix Network Programming* (Prentice Hall, New Jersey, 1990).
16. L. Li, B. Cox, S. Shelton and M. Diftler, "Telepresence control of a dual-arm dexterous robot" **In:** *Proceedings of the Third International Symposium on Artificial Intelligence, Robotics, and Automation for Space* (1994) **vol. 1,** pp. 313–316.
17. J. D. Wise and L. Cison, "Telrip distributed applications environment operators manual", *Tech. Rep. 9103* (George R. Brown School of Engineering, Rice University, Houston, TX, 1992).

https://doi.org/10.1017/S0263574798000472 Published online by Cambridge University Press