A Robotic Haptic System Architecture

MS-CIS-90-51
GRASP LAB 224

Mario Campos
Ruzena Bajcsy

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104

August 1990

ACKNOWLEDGEMENTS:
This research was supported in part by Airforce Grant AFOSR 88-0244, AFOSR 88-066, Army/DAAG-29-84-K-0061, NSF-CER/DCR82-19196 Ao2, NASA NAG5-1045, ONR SB-35923-0, NSF INT85-14199, ARPA N0014-88-K-0630, NATO Grant No.0224/85, DuPont Corp., Sandia 75 1055, Post Office, IBM Corp. and LORD Corp.

A ROBOTIC HAPTIC SYSTEM ARCHITECTURE

Mario Campos and Ruzena Bajcsy*
GRASP Laboratory
University of Pennsylvania
Department of Computer and Information Science
200S, 33rd Street, Philadelphia, Pa, 19104

August 8, 1990

Abstract

In order to carry out a given task in a unstructured environment, a robotic system must extract physical and geometric properties about the environment and the objects therein. We are interested in the question of what are the necessary elements to integrate a robotics system that would be able to carry out a *task*, i.e pick-up and transport objects in an unknown environment. One of the major concerns is to insure adequate data throughput and fast communication between modules within the system, so that haptic tasks can be adequately carried out. We also discuss the communication issues involved in the development of such a system.

[1] Acknowledgements: This research was supported in part by Airforce Grant AFOSR 88-0244, AFOSR 88-0966, Army/DAAG-29-84-K-0061, NSF-CER/DCR82-19196 Ao2, NASA NAG5-1045, ONR SB-35923-0, NSF INT85-14199, ARPA N0014-88-K-0630, NATO Grant No.0224/85, DuPont Corp., Sandia 75 1055, Post Office, IBM Corp. and LORD Corp.

1 Introduction and Motivation for this work

In order to carry out a given task in a unstructured environment, a robotic system must extract physical and geometric properties about the environment and the objects therein. We are interested in the question of what are the necessary elements to integrate a robotics system that would be able to carry out a *task*, i.e move around and transport objects in an unknown environment. These elements, which will be considered here are: sensors and their associated data acquisition/treatment procedures, manipulators/end-effectors and their control procedures. In a good scientific approach, one is looking for the smallest set of physical and geometrical properties, which we call here *primitives*, so that other more complex properties could then be described by or composed of these primitives.

The physical properties to be measured by our system are *distance*, *temperature*, *mass*, *force*, *position* and *time*. The derived physical properties are then: weight, velocity, acceleration, etc ... The geometric properties that we will be using is the subject of a separate work [1]. We just like to mention here that we use parametric representation up to second order models both for volume and surface as well as for contour. We assume that these geometric primitives are able to describe a given object (and its parts, if it contains them). If an object is composed of more than one part, it is represented explicitly by a graph. Parts can be in fixed or movable spatial relationship with respect to each other. The movable part/whole relationship will yield variable configurations of the whole which needs to be recognized.

Once the primitives are defined, a robotic system also has to be equipped with procedures that control data acquisition. These procedures help to ultimately determine where to look or touch as well as to extract (compute) the aforementioned primitives from the measurements.

Inspired by the work of Lederman and Klatzky [2], we name the system's haptic, manipulatory and visual procedures as *Exploratory Procedures* (or EP's). In order to test whether the proposed primitives are necessary and/or sufficient to describe the environment for object intrinsic mobility and manipulation, and whether the EP's are realizable and able to deliver the required data primitives, we built a robotic system to serve as a test bed.

We have been working for a few months building a modular robotic system with special focus on its ability to perform haptic tasks. One of the major concerns was to insure adequate data throughput and fast communication between modules within the system. Fast communication is especially important for "force control". For example, when sensors in the gripper's fingers are sensing large forces (that may very well be too large already) the system has to be able to react promptly to avoid a catastrophy (i.e. damage to the object/environment being touched or to the end-effector itself). Therefore, if sensor data throughput is not adequate, one is not able to perform simple interactions within the environment.

In what follows, we will describe the architecture of our haptic robotic system. In section 3 we discuss the communication issues and the data flow that guarantees a fast control of the manipulator and which uses the force/torque information provided by the sensors in the end-effector (gripper).

2 Architectural Issues

We begin to describe the system with the top most level of the architecture which we call the *task level.* Either the manipulatory or the haptic task has two basic components differentiated by the sensor which is being applied. We differentiate between manipulatory and haptic in that we consider manipulatory those actions which involve grasping an object and moving it around. Haptic comes into play when one is now concerned with extracting data and other information from the object itself, or the environment. As an example of manipulation we have *pick and place* whereas for haptics would be to *categorize* objects with respect to a given dimension such as size, volume, weight, objects with movable parts as oppose to rigid parts, etc ...

In order to explore an unstructured and unknown environment containing objects of unknown mechanical and material properties, we will be using vision as well haptics and manipulation. Thus our system can be naturally sub-divided into **Haptic Task** and the **Visual Task**. A top level diagram of the architecture is depicted in Figure 1.

At a higher level, the system is only concerned whether the task is accomplished or not. The task is subdivided into sub-tasks, which are further broken down into distinct modules. The main sub-tasks are shown in Figure 2 and Figure 3 respectively. Following the inspiration from Lederman and Klatzky's work, we have integrated a robotic system to perform the **Haptic Task** and the **Visual Task**.

Klatzky and Lederman in their work have identified five fundamental modules: Motoric, Sensorial, Property, Exploratory Procedures and Object modules. Our robotic architecture, however, requires a somewhat different partitioning. The *Object Module* in their work corresponds to the

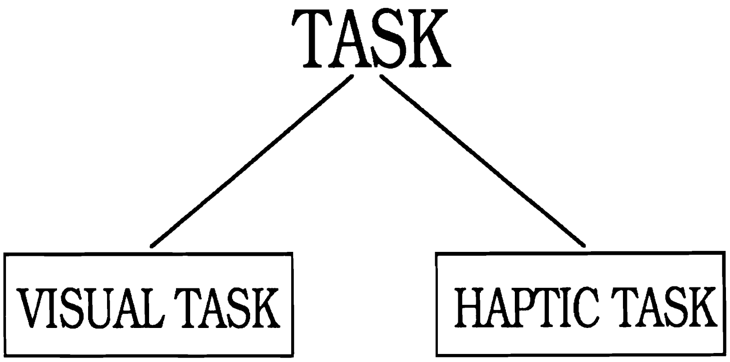

Figure 1: Functional Block Diagram

Haptic Task Description in ours. Our Haptic Properties and EP's are very similar to theirs. Their *Motoric Module* is mapped into two parts in our system: one is the *Arm Controller* or robot controller and the other is the *Gripper Controller.* Their *Sensor Module* is in our case the *Force/Tactile and Position Sensor Modules.* Physically, the Force/Tactile and Position Sensors are located, one on each finger of the Lord Gripper. The Sensor Module together with the Gripper Controller Module are combined into Gripper Primitives.

The motoric tasks are carried out by two disjoint systems: the manipulator (robot) and the gripper. In our implementation of the system, we used the PUMA 560 from UNIMATE and its associated controller. The gripper is the LORD Experimental Gripper (LEGS). This gripper is a parallel jaw gripper, with independent control of each finger. It also has the LTS-200 Force/Torque and Tactile sensors on each finger [3]. This sensor is composed of a force/torque sensor which provides information force/torque information on each direction X, Y and Z as well as a tactile sensor which is composed of a 10 x 16 array of tactile sites.

The software is composed of several libraries. These libraries are structured with respect to servo. sensors and communication. The gripper primitives are composed of other primitives from other libraries, but can be (and will be) easily replaced by other end-effector primitives such as *Hand Primitives.* Unlike the human haptic system, where the motoric and sensory system is "given" we spent a considerable amount of time in integrating these several sub-systems. The Motoric Module is composed of several (in our case two) copies of a Motoric Controller. This has the advantage that the software part is flexible and for any specific hand or some other end effector, only the

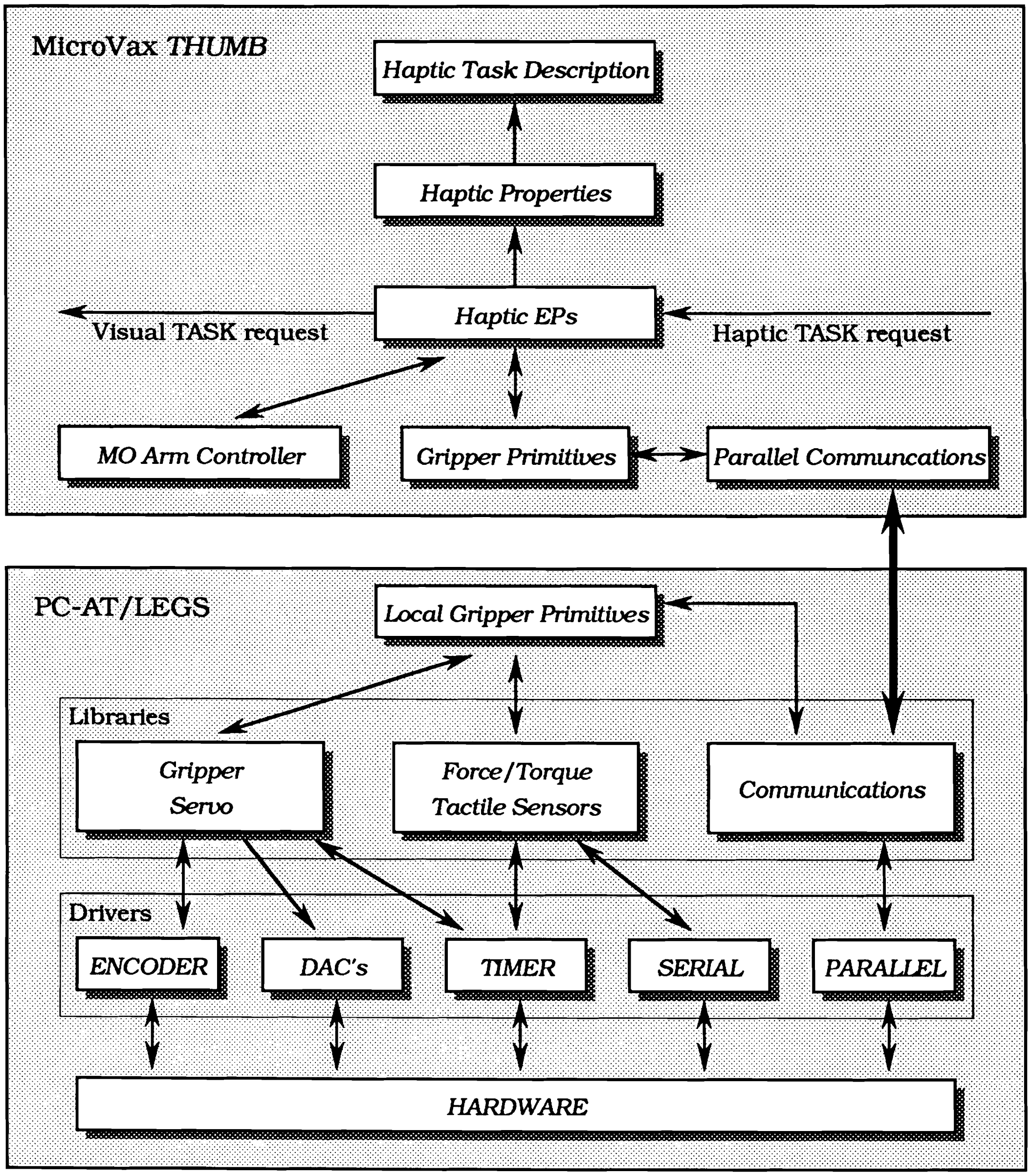

Figure 2: The Haptic Task

specifics about motors and encoders need to be changed. This is also true for the sensory module. In both cases, the modules include the models (geometry, electrical and mechanical properties) of the particular motor or sensor respectively. In section 4 we will elaborate more on the details of the motoric module and on the haptic EP's.

The Visual Task, similarly to the Haptic Task, is subdivided into modules, which is shown in Figure 3.

As the Haptic Task, the Visual Task is also composed of **Visual Properties**, which are extracted by Visual Exploratory Procedures. These Visual EP's control the position of the head, neck, the focus and vergence of the eyes, opening and closing of aperture (iris), similar to the implementation of Krotkov [4]. The Visual task also determines what resolution/detail, as well how many views and how much data should be acquired and what features need to be extracted.

Interaction between the Visual Task and the Haptic Task in this implementation at the physical level is via Ethernet. The architectural description that follows will be focused on the haptic system architecture.

3 Communication issues in this Architecture

One cannot truthfully consider performing Haptic Tasks if the Haptic System which is to perform the task does not have force control capability. One of the driving motives behind this architecture, therefore, was providing the the robotic system with the cability of performing "force control". To achieve this force control at a reasonable "haptic speed", the force/torque sensors on the end effector have to provide the force control procedure with fairly large data throughput. Clearly, the loop: "sense force – control procedure – manipulator/gripper action" has to happen at a compatible haptic speed. In order to fully understand the implications of the aforementioned issues, we first present the first architecture that was implemented in order to connected the Lord Gripper and the Unimate Puma 560 controller in an attempt to solve the problem. See Figure 4.

This architecture had the following characteristics:

- In this configuration, the UNIMATE controller was running VAL II [5], [6], and the Lord Gripper controller was running UNIX V.

- VAL II allows only position control and did not provide access to the controller hardware like

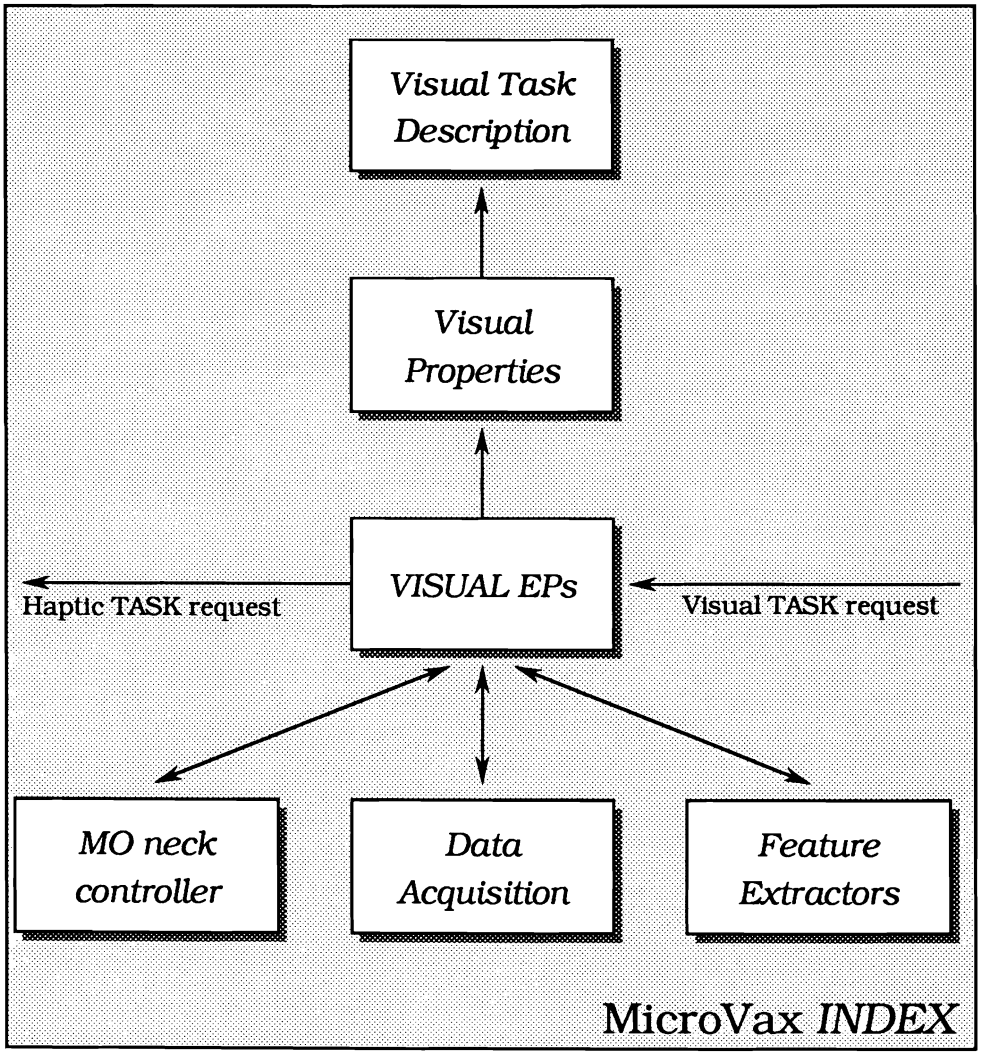

Figure 3: The Visual Task

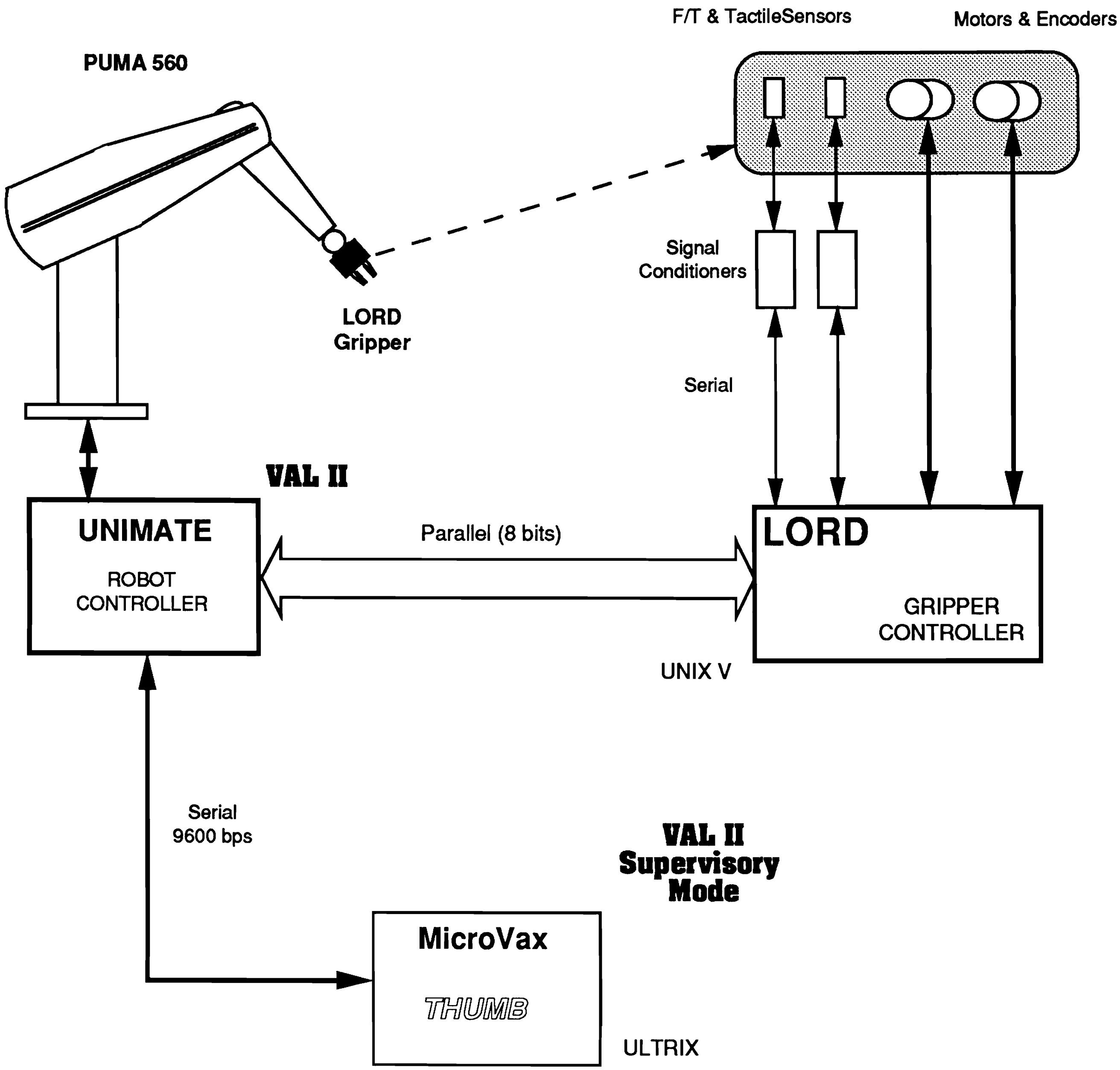

Figure 4: Robot Control using VALII

encoder values and DAC values, hindering modifications/inclusions/substitutions of control procedures. Hence, since VAL II did not provide force control to begin with, it was not possible to have this control mode.

- The rate of force sensing at the gripper and the corresponding actuation by the manipulator, would take hundreds of milliseconds. The problem with large delays is that a small displacement of the manipulator may generate large forces/moments in the gripped structure/object. In this architecture, the robot controller is not able to check for forces in the end effector "during" the movement, but only at the end of the requested move.

- The MicroVax is not able to know forces at the gripper, since the sending of data back from the UNIMATE controller would be impossible. Therefore, the MicroVax is not able to actually control the manipulatory action.

- Since all the control is performed by programs written in VAL II, the programming environment is very limited, and awkward. Therefore, gripper commands are sent over the 6-bit data parallel connection. This was system used by Tsikos [7].

As we could see, this organization was not adequate for our application. Therefore, an immediate improvement in the architecture was needed, and it is shown in Figure 5.

It differs from the architecture depicted in Figure 4 only in the introduction of a serial communication line between the MicroVax and the the Lord Gripper Controller [8]. This serial line was added so that the gripper now could receive commands directly from, and send force/torque information to the MicroVax. A library of C routines was available in the MicroVax, that included commands such as grasp, read position, etc... This architecture was still inadequate in two important aspects. First, even though it was possible now to control the gripper from the MicroVax, the limitations of VAL II still persisted. Therefore, it was still impossible to monitor force/torque information from the gripper within the motion time. Second, the force/torque data throughput was slow due to the delay of two UNIX systems back to back (LORD Gripper controller and the MicroVax).

The solution for the first problem prompted the controlling of the robot from the MicroVax as well. With RCI/RCCL [9], [10] now available and running on the MicroVax, it was now feasible to control both the gripper and the robot from the MicroVax. In order to minimize the second

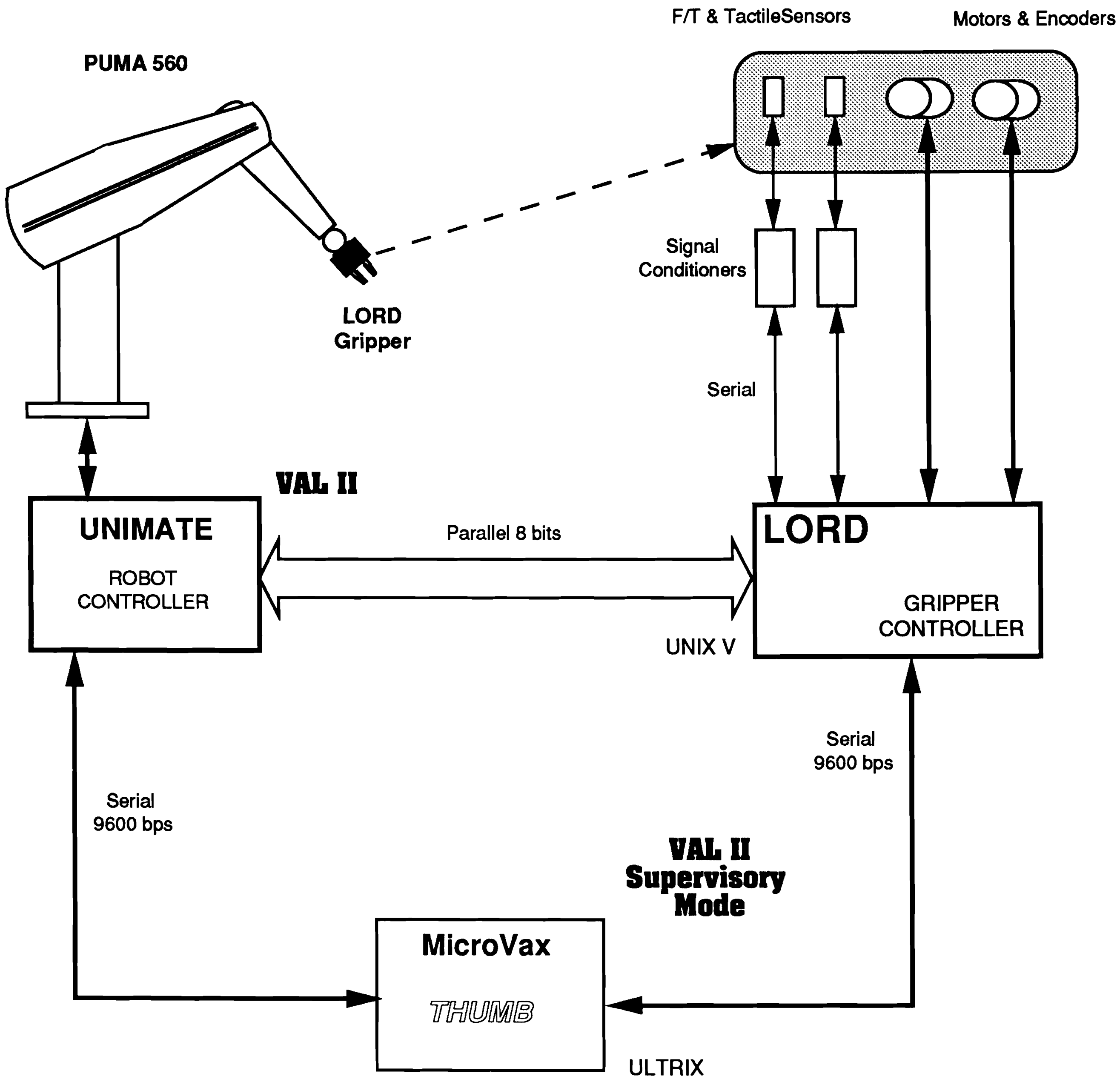

Figure 5: Robot Control using VAL-II in the supervisory mode

FEATURES	VAL II	RCI/RCCL
Position Accuracy	good	low
Repeatibility	0.1 mm	1.0 mm
Reliability	very good	poor
Sampling Rate	1 ms	28 ms
Force Control	difficult	easier
Documentation	good	fair
Extensibilty	low	good
Source Code	unavailable	available
Programming Environment	poor	very good
Programming Support	none	fair
Library of Functions	none	good

Table 1: VAL-II and RCI/RCCL contrasted

problem, the serial communication between the MicroVax and the Lord Gripper Controller was replaced by a 32 bits parallel communication port. The next architecture, that included these modifications is shown in Figure 6.

As mentioned before, this architecture has the following main characteristics:

- The serial connection between the LORD box and the MicroVax was replaced by a 32-bit parallel communication,

- VAL II was replaced by RCI/RCCL (Robot Control Interface Robot Control C Library) [10] which enabled the y control the robot from MicroVax. We can summerize the pros and cons of VAL II versus RCI [11]:

The improvements achieved with these changes were not as much in communications speed as it was in *flexibility* and *programmability* of the overall system. In terms of speed, we still had the LORD Controller running UNIX and its device drivers. Since we did not have access to the kernel's source code, any modifications of the servo code would not be viable. This was a major obstacle, since we wanted to implement "force control" at the gripper level.

Based on this analysis, we designed our system, which is depicted in Figure 7.

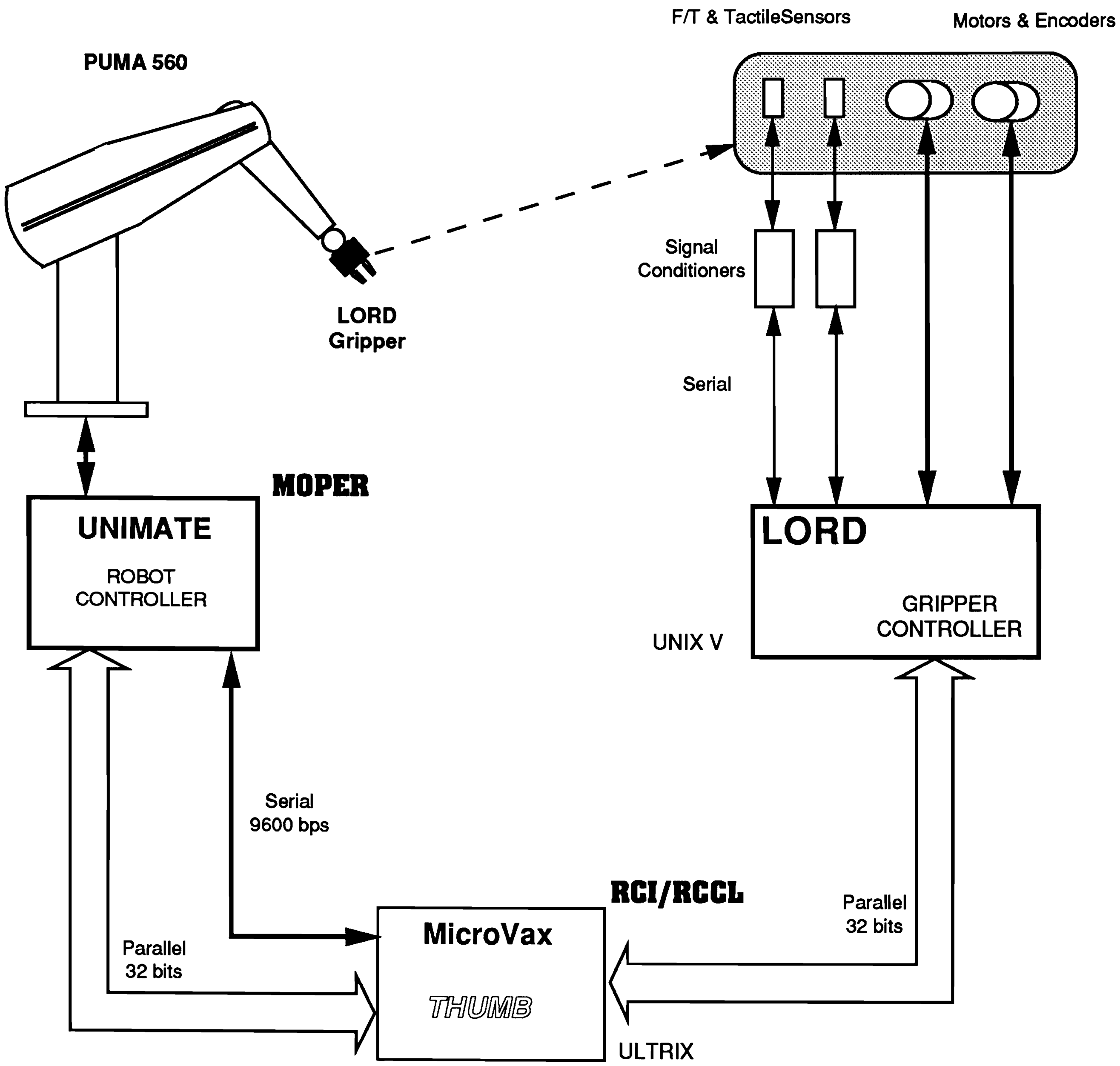

Figure 6: Robot Control running RCI/RCCL

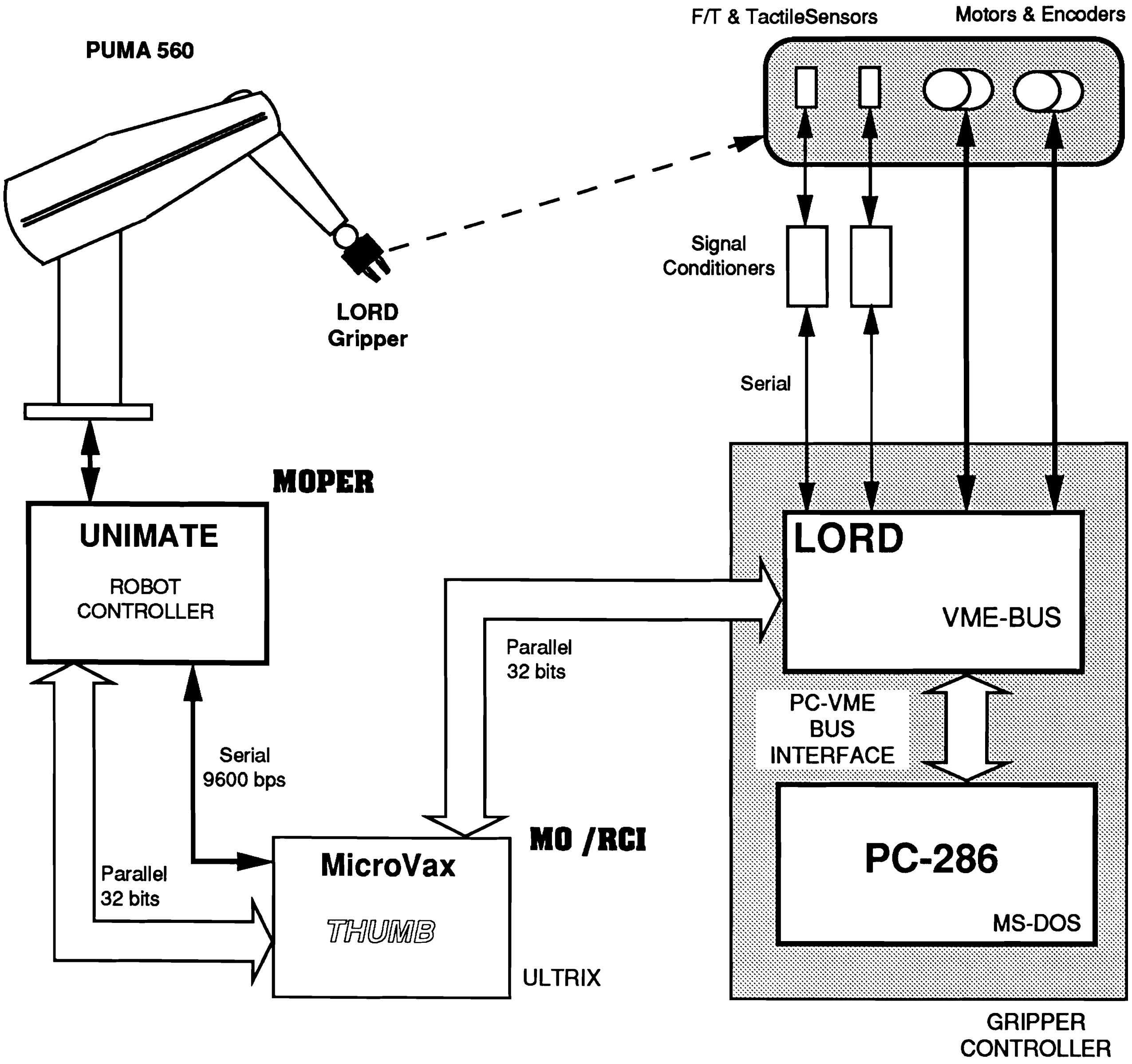

Figure 7: Current System

As it can be seen in Figure 7, the gripper controller is now a 286 based microcomputer running MS-DOS 3.3. The 286 machine was connected to the LORD Controller bus via an off-the-shelf PC-TO-VME bus interface from Bit 3 [12]. This bus interface allowed the VME bus on the Lord Controller to be mapped into the PC memory address space. The CPU board on the VME bus was removed, so that the PC would become the only bus master. This approach simplified greatly the implementation of this new system, since all current amplifiers, encoders interfaces, serial communication to the sensors, DACS and power supplies were the ones in the LORD Controller. This implied, however, in a new implementation for the gripper control software and sensors interface software.

On the other hand the system is now more flexible, since modifications to the gripper control software were now a trivial task, involving only the recompilation of the servo routine, and not of an entire operating system's kernel. Also, the parallel communication would not suffer from latencies, making the access to the sensors' data more predictable, which is indispensable for force control.

In summary, this new organization has the advantage of being flexible, predictable, cheaper, faster than the previous ones. More than that, because of the guaranteed latency of the parallel input, we will be able to work with the manipulator/gripper system in "pseudo-force control" mode, that is, we are able to perform exploratory tasks based on force feedback from the gripper sensors, which was our goal in building the current system.

4 HAPTIC EXPLORATORY PROCEDURES

Exploratory Procedures Design

We distinguish two categories of Exploratory Procedures, The first one which can extract the desired property just by using the hand/gripper and its built-in sensors. The second one, which needs some extension of the capabilities of the hand by using another tool.

4.1 Exploratory Procedures using only the hand/gripper

Every such Exploratory Procedure has at least three tasks that translate into commands:

1. *Reach to the object/surface.* This is position controlled, point to point path movement, carried out in cartesian coordinate system. The desired position is either a priori given or determined

by vision system.

2. *Make static contact.* In this command we depend on the contact sensory feedback that indicates the the task (contact in this case) has been accomplished.

3. *Carry out the particular Exploratory Procedure.*

What follows is an elaboration of the second task for particular EPs.

Hardness

There are two ways how one can implement this EP:

1. By *pressing.* This is possible if the mechanical hand can extend its fingers so that the can press the surface. This has been reported by Stansfield [13].

2. By *squeezing.* This is when the hand will GRASP and then PRESS.

In both cases the contact between the finger(s) and the surface must be such that the force exerted is normal to the surface. The control is force-feedback based. The outputs are : Force and displacement. The range of measurable hardness properties is given by: the sensitivity of the force and displacement value at one end, and either the maximum load that the robot can withstand without breaking down or the maximum force applied to the material without crashing it.

A question remains on how to determine the breaking point of the material; perhaps in absence of actually breaking the material by the use of some a priori knowledge about materials, like a handbook of materials. Hardness, however , depends also on the form factor of the object, so that the breaking point will have to be normalized with respect to the size/shape which in turn can be measured from vision (non-contact measurement).

There are two dimensions that we get as the output properties: soft-to-hard materials plastic to elastic materials. The soft to hard dimension is obtained by varying the reactive forces of the material under constant displacement (assume that sizes are the same for all samples). The dimension of plastic to elastic can be obtained by applying constant force and observing the displacement and its recoverability after removal of the force. Again we assume the same geometry for all samples.

Weight

The weight EP is composed of two parts:

1. **Grasp the object**. During the Grasping operation one needs to monitor the Grasping forces which must be equal in the opposing fingers. The magnitude is determined by the hardness EP.

2. **Lift the object**. If the lifting motion is carried out only as parallel to the force of gravity then the control is based on force feedback measured at the wrist. Simultaneously, the forces and tactile sensors on the fingers need to be monitored to prevent slip. The output is the force readout.

The derived entity from weight is the density of the material. This can be obtained by measuring in addition the geometry of the object.

4.2 Exploratory Procedures Using Tools

The second category of Exploratory Procedures is the one where the hand uses another tool in order to obtain the desired measurement. We consider two such EPs: One is to extract hardness by using a tool to be grasped by the gripper such as a rod, and the other would be to measure the material's t thermo-diffusivity by applying a the thermo-diffusivity sensor grasped by the gripper.

Both of these EPs have one additional step, which is to grasp the adequate tool, before one can perform the actual exploration.

Hardness

Extracting the hardness by means of the rod is basically the same as extraction hardness by using direct gripper pressure. The only difference is that the gripper will not be pressing the object directly, but through the rod. The other parameters of this EP are then the same as those of hardness using direct gripper pressure.

Thermal Diffusivity

Thermal-diffusivity of a material can be obtained by making maximal contact of the probe with surface of the object. This EP is force feedback controlled (move until contact) and the output is temperature variation over time.

We plan to test each of the above EP's in a series of systematic experiments. For example: hardness will be tested by taking several objects made out of different materials such as wood, steel, copper, etc... but with the same geometric properties (shape and dimensions). The results we are anticipating are families of curves for the tested objects (materials): soft-to-hard plastic to elastic. We then wish to take samples of several thickness from the different materials and obtain similar curves. These curves will be calibration curves for classifying materials with respect to hardness and deformability. A similar procedure will be also carried out for thermal diffusivity. Finally weight will also measured for different materials/sizes.

4.3 Integration of Properties

The studies on human subjects suggest that while a given EP is being carried out, other properties are obtained like in the case of grasping, hardness, texture and global shape can also be obtained [14]. However the sensitivity or resolution of these other properties is low when compared with when they are the primary EP being invoked. The question that we expect to answer here is whether this is the case for robotic systems. From the classification curves, we suspect we should be able to predict this unless the current grasping force does not belong to the set of grasping forces existent in the curves of rigid vs. elastic/plastic. The grasping force (if we neglect slip) will clearly be different when two handed grasp is performed. On the other hand the discrimination between objects based on small differences in all dimensions or bigger differences in a given dimension is an open question.

5 Conclusions

The objective of the system we have described is to support the *task*; that is to explore the unknown environment with the purpose of manipulation and mobility. The Task is decomposed in to Haptic and Visual subtasks, that these must interact. In section 3. we have described an evolution of the architecture that allows us to carry out the Tasks and subtasks listed in section 4. In section 4. we have outlined some of the features that the Haptic and Visual subsystem must extract from the environment. The main features of the implemented architecture are:

1. A flexible programming environment for interaction between sensing and actuation,
2. A high throughput inter-module communication necessary for dynamic exploration.

We wish to study how potent vision is in the haptic/vision cooperation scenario. To do so we can measure the extra steps that the haptic task takes as visual information degrades. One example of such degradation is that one reduces spatial resolution of the visual information up to the point where only the center of gravity is known.

6 Acknowledgements

The authors would like to thank Gaylord Holder, University of Pennsylvania, for the suggestions and for the implementation of the parallel communication package used in the final architecture. We also wish to thank Fil Fuma and John Bradley, University of Pennsylvania, for their suggestions and help throughout the development of the system.

References

[1] A. Leonardis, A. Gupta, and R. K. Bajcsy. *Segmentation as the Search for the Best Description of the Image in Terms of Primitives.* Technical Report MS-CIS-90-30, University Of Pennsylvania, Department Of Computer and Information Science, School of Engineering and Applied Science, Philadelphia, PA 19104-6389, May 1990.

[2] S.J. Lederman and R.L. Klatzky. Hand Movements: A Window into Haptic Object Recognition. *Cognitive Psychology*, 19:342–368, 1987.

[3] *Installation and Operations Manual for LTS 200 Tactile Sensor.* Lord Corporation, Cary, NC 27511, October 1985.

[4] E. P. Krotkov. *Active Computer Vision by Cooperative Focus and Stereo.* Springer-Verlag, New York, 1989.

[5] A. Izaguirre. *Implementing Remote Control of a Robot Using the VAL II Language.* Technical Report MS-CIS-84, University of Pennsylvania, Department of Computer and Information Science, School of Engineering and Applied Science, Philadelphia, PA 19104-6389, 1984.

[6] *Programming Manual - User's Guide to VAL II Version 2.0 - 389AG1.* Unimation Incorporated, Danbury, Connecticut, February 1986.

[7] C. I. Tsikos. *Segmentation of 3-D Scenes using Multi-Modal Interaction between Machine Vision and Programmable, Mechanical Scene Manipulation.* PhD thesis, University Of Pennsylnania, 1987.

[8] A. Koutsou. *Object Exploration Using a Parallel Jaw Gripper.* Technical Report MS-CIS-88-48, University of Pennsylvania, Department of Computer and Information Science, School of Engineering and Applied Science, Philadelphia, PA 19104-6389, July 1988.

[9] J. Loyd. *Implementation of a Robot Control Development Environment.* Master's thesis, McGill University, Montreal, Quebec, Canada, December 1985.

[10] Vincent Hayward. *RCCL User's Guide.* Computer Vision and Robotics Laboratory, McGill Research Centre for Intelligent Machines, Montreal, Quebec, Canada, April 1984.

[11] *PUMA Mark II Robot - 500 Series Equipment Manual for VAL II and VAL PLUS Operating System - 398U1.* Unimation Incorporated, Danbury, Connecticut, August 1985.

[12] *IBM PC/AT - VME Adaptor Model 403.* Bit 3 Computer Corporation, Minneapolis, MN 55431-1393, May 1989.

[13] S. Stansfield. Haptic Perception with an Articulated, Sensate Robot Hand. To appear in International Journal of Robotics Research.

[14] C. Reed R. L. Kltsky, S. Lederman. Haptic Integration of Object Properties: Texture, Hardness, and Planar Contour. *Journal of Experimental Psychology: Human Perception and Performance*, 15(1):45–47, 1989.