November 1990

HEAP: A Sensory Driven Distributed Manipulation System

Sanjay Agrawal
University of Pennsylvania

Marcos Salganicoff
University of Pennsylvania

Michael Chan
University of Pennsylvania

Ruzena Bajcsy
University of Pennsylvania

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-90-90.

HEAP: A Sensory Driven Distributed Manipulation System

Abstract

We address the problems of locating, grasping, and removing one or more unknown objects from a given area. In order to accomplish the task we use HEAP, a system of coordinating the motions of the hand and arm. HEAP also includes a laser range finer, mounted at the end of a PUMA 560, allowing the system to obtain multiple views of the workspace. We obtain volumetric information of the objects we locate by fitting superquadric surfaces on the raw range data. The volumetric information is used to ascertain the best hand configuration to enclose and constrain the object stably. The Penn Hand used to grasp the object, is fitted with 14 tactile sensors to determine the contact area and the normal components of the grasping forces. In addition the hand is used as a sensor to avoid any undesired collisions. The objective in grasping the objects is not to impart arbitrary forces on the object, but instead to be able to grasp a variety of objects using a simple grasping scheme assisted with a volumetric description and force and touch sensing.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-90-90.

Department of Computer & Information Science

Technical Reports (CIS)

University of Pennsylvania *Year* 1990

HEAP: A Sensory Driven Distributed Manipulation System

Sanjay Agrawal
University of Pennsylvania,

Marcos Salganicoff
University of Pennsylvania,

Michael Chan
University of Pennsylvania,

Ruzena Bajcsy
University of Pennsylvania,

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-90-90.

HEAP: A Sensory Driven Distributed Manipulation System

MS-CIS-90-90
GRASP LAB 243

Sanjay Agrawal
Marcos Salganicoff
Michael Chan
Ruzena Bajcsy

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104-6389

December 1990

HEAP: A Sensory Driven Distributed Manipulation System

Sanjay Agrawal, Marcos Salganicoff, Michael Chan, Ruzena Bajcsy
GRASP Laboratory
Department of Computer and Information Science
University of Pennsylvania

November 2, 1990

Abstract

We address the problems of locating, grasping, and removing one or more unknown objects from a given area. In order to accomplish the task we use HEAP, a system of coordinating the motions of the hand and arm. HEAP also includes a laser range finder, mounted at the end of a PUMA 560, allowing the system to obtain multiple views of the workspace.

We obtain volumetric information of the objects we locate by fitting superquadric surfaces on the raw range data. This volumetric information is used to ascertain the best hand configuration to enclose and constrain the object stably. The Penn Hand used to grasp the object, is fitted with 14 tactile sensors to determine the contact area and the normal components of the grasping forces. In addition the hand is used as a sensor to avoid any undesired collisions. The objective in grasping the objects is not to impart arbitrary forces on the object, but instead to be able to grasp a variety of objects using a simple grasping scheme assisted with a volumetric description and force and touch sensing[1].

[1] Acknowledgements:This work was in part supported by: Airforce grant AFOSR F49620-85-K-0018, Army/DAAG-29-84-K-0061, NSF-CER/DCR82-19196 Ao2, NASA NAG5-1045, ONR SB-35923-0, NSF INT85-14199, NSF DMC85-17315, ARPA N0014-88-K-0632, NATO grant No. 0224/85, DEC Corp., IBM Corp.,

1 Introduction

In designing a manipulation system that could function autonomously in performing simple grasping tasks, we attempted to maximize the use of sensory input and mechanical intelligence in accomplishing the task. In doing so we were able to reduce the number of assumptions we would make about the environment we operate in, the size, shape and material the objects being manipulated, and any obstruction that might be present in reaching the object.

In order to get an initial estimate of the contents workplace of the hand, within which the object to be manipulated is expected to be located, we use a laser-range finder mounted on a six degree of freedom Puma arm. The workplace of the hand is defined as the intersection of the volume which can be reached by the hand, mounted on a robot arm, and the volume that can be scanned by a range finder mounted on an adjacent robot arm.

The range image of the workplace thus obtained is then segmented using algorithms developed by Gupta[Gupta89]. The purpose of the segmentation, is to be able to identify distinct objects which can be then be grasped and removed from the scene. We use superquadrics [Sol87] to model the objects extracted from range data and obtain suitable object descriptions. The assumption made here is that the object rests in a stable manner on a planar surface.

The grasp planning and execution is done once we obtain a list of surface descriptions of objects that need to be grasped. The grasp planner works on the list sequentially, creating a queue of motions for the hand and arm as each object is analyzed. The planner is told what must be done with each object, once it has been stably grasped and lifted away from the resting surface. Once a successful grasp and disposal has been accomplished, the hand-arm system goes on to perform planning for the high level subgoal in the queue, i.e. the removal of the next object.

In any autonomous system we expect to find some amount of uncertainty and error, both in our sensory data and our planning. The use of tactile pads mounted on the fingers of the hand and covered with a compliant silicon elastomer, allows us to use the fingers as probes to detect unexpected contacts. In addition using the the three fingers as non-co-linear contact points, we can determine the surface orientation and follow a large contour, while approaching the object to be grasped.

So far we have built up a framework to perform a limited range of manipulatory tasks, in partially structured environments, making only the minimal assumptions about the objects we grasp. Work will be done to continuing to enhance both our abilities to attain more precise data about the environment, as well as increase the set of tasks that the system can plan and execute autonomously.

1.1 Previous work

Robot systems exist in many varied forms, each focused on a specific aspect of manipulation. There are only a few systems which attempt to grasp objects using primarily sensory information, without prior object modelling.

Stansfield [Sta89, Sta90] attempts to first explore and extract the physical and geometric properties of the object, such as the hardness, weight, size and shape of the object. This is done via built-in so called Exploratory Procedures, that are motoric and sensate procedures used to extract

the above mentioned properties. The size and shape is extracted from visual-range sensor data. Stansfield converts these measurements into linguistics labels, such as heavy, soft, large, small etc. which then in turn are used for planning and executing grasp strategies. Allen [All89] uses a hand mounted on PUMA560 to extract information from the environment by using tactile sensors mounted on the fingertips of the hand. In Allen's case the position of the object is known a priori.

Other complex hands have been integrated into systems to perform manipulation and grasping tasks for many years though and among these are Geschke's early system to perform robotic manipulation tasks [Ges83] Kuniyoshi et.al. [Kun90] in building an integrated robotic teaching and learning systems, Salisbury's integrated hand/tactile system [Sal85, Sal85a], and work at USC in integrating the Belgrade/USC hand into an active perceptual environment [Liu89, Rao88].

Several research groups working on the MIT/Utah hand have also build integrated systems where the developers of that hand [Jac86, Nar86] provide the low level control system, and a software environment to utilize the low-level control functions. Many researchers have focused on grasping strategies using multifingered hands. The focus here has been to use the multiple degrees of freedom in the fingers of the hand to be able to impart arbitrary forces and moments to a given object. Early work was done by Salisbury and Okada[Sal82, Oka82], where they developed control laws to utilize multi-jointed, multi-fingered grippers. Other seminal ideas were put forth by Fearing [Fea84],Cutkosky [Cut85] and Jacobsen [Jac85] in the area of multi-fingered grasp planning. Of late there have been many papers which address issues in using both intrinsic and extrinsic tactile feedback, to determine the kind of contact, as well as the contact location. Here the object model is known a priori, and the exact orientation and position of the object in the grasp remains to be determined [Hon90, Par90, How90, Bic89, Ngu90]

2 System Overview

The system consists of four main modules, the image coordinator, segmenter, grasp planner and run-time controller, see fig. 2 for details. The interaction of these is arbitrated by the central coordinator process. The hardware used to implement this scheme is show in fig.2.5.

There is a front-end to the segmenter and hand/arm modules that allows the sensory information from each module to be mapped into the current global task framework. In addition each front-end accepts commands for its respective module, and translates these commands to a format that the module can parse. As a final task, the front-end ensures that all messages are relayed back and forth in a reliable and consistent manner.

2.1 System Coordinator

The coordinator is responsible for the overall sensing and manipulation strategies and satisfying the ultimate task which is the removal of all objects. It schedules top-level actions for the task such as the invokation of perceptual actions in the visual and tactile modes and provides the parameters for these actions. Actions may be either perceptual, such as to take a scan or segment the output of a scan, or motoric such as move the hand/arm system to a given position in a given mode.

During the completion of a task, the hand-arm system iterates through several global modes of

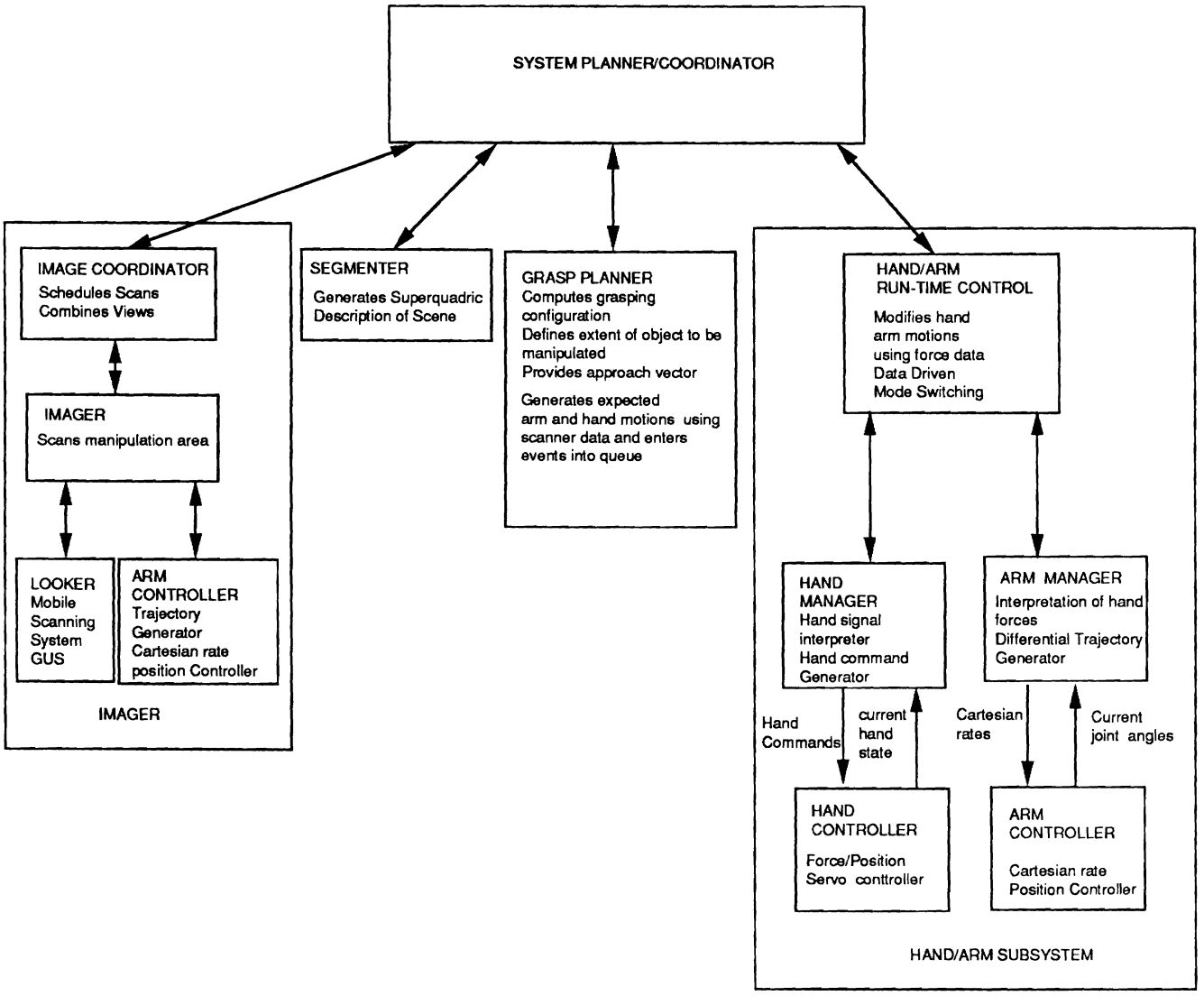

Figure 1: System Command and Data flow

operations. These modes comprise of set of modes for the hand, arm and scanner. Each mode for the arm, requires the controller to interpret the force feedback from the hand in a distinct manner. The hand modes are based on the current function the hand is performing. The mode for the hand determines under what conditions the hand executes the next command. Transition from one mode to occurs via sequential queues that are set up for both the hand and the arm. The transition between modes is decided by the respective queue managers. The queues incorporate dependencies between the desired motions of the hand and the arm, as well as the requirement for new images to be obtained via the laser range scanner.

The coordinator is primarily data driven. It has three queues, one queue specifies the position and orientation of regions of to be scanned by the range scanner and preconditions for these scans. The other queues are motion oriented and contains movement events for the hand and arm systems. Each queue has dependencies between future and current sensing actions/motions among the modules. The coordinator has queue managers to monitor the current elements in each queue, though the coordinator can, if neccessary, modify or disable any queue.

System state is defined as the position and orientation of objects within the robot workspace, their corresponding surface descriptions, the configurations of the robot arm, the hand and the scanner/arm subsystem. It is assumed that the state of the system may only be changed by actions initiated by the hand/arm subsystem, and that system state will remain static otherwise (i.e. there are no external agents)

2.2 The Image Coordinator

The image coordinator is responsible for realizing the range scanning of an arbitrarily oriented rectangular patch of the workspace. Since the mobile scanner can only scan a fixed width swath of workspace, the image coordinator is charged with the task of decomposing this commanded region into subscans which satisfy the hardware constraints. The subscans consist of arm trajectories and velocities. The coordinator plans these sub scans and merges them together into a single data structure for the entire region as necessary and passes them back to the coordinator.

2.2.1 The Mobile Laser Range Imaging System

Our Laser Range Imaging System consists of two components: The *LOOKER* and the *GUS* processing unit. The *LOOKER* is composed of a laser stripe generator and SONY XC-39 camera which generates video signal of the images obtained under the illumination of the laser stripe, and the *GUS* unit [Tsik87] processes the continuous sequence of laser images and generates a range image of the scene in real time.

2.2.2 Operation of the System

The *LOOKER* is called by its name because it can easily be mounted on the tip of a Puma 560 robot and can be made to "look" from different direction of a scene.

In operation, it moves linearly at a known constant velocity under robot control, thereby scanning the scene we are interested in. By geometry, it can be shown that the position of the laser

stripe as observed by the camera is an a measure of height of the nearest object intercepted by it. This video signal is sampled at a rate of $60Hz$ by the *GUS* processing unit and the range image is produced in real time.

Synchronization between scanning motion and image generation is ensured by the ability to send a triggering command along a serial line connecting the host computer controlling the robot and the *GUS* processing unit.

The imaging volume of a single scan and the resolution of the range image are summarized as follows:

Axis	X (width)	Y (length)	Z (height)
Imaging volume (mm)	135	164	172
Resolution (mm /pixel)	.23	.485	.672

Since the size of an image is limited by the imaging volume of *LOOKER* for a given resolution, multiple number of scans are needed in order to cover whole workspace we are interested in. Having the scanner under manipulator control allows us the flexibility of variable resolution in the Y direction. Noting that the resolution in the Y direction (the scanning direction) is a function of velocity of the scanning motion, it is often useful to obtain a coarse large area scan (scanning at a higher velocity) in order to locate approximately where the object(s) is/are. This first coarse scan can be used decide which regions should be further scanned to get the finer details and which will provide little additional information.

In surface regions where the laser stripe cannot illuminate or the camera cannot "see", pixel values of zero are assigned. Multiple number of scans of the *same* scene from *different* direction are needed to recover the occluded part of the scene as much as possible.

Another limitation of the imaging system is that orthographic projection is assumed in the generation of the range image . Software compensation is employed to counteract errors of this kind, especially for tall objects.

2.3 Grasp Planning Based on Superquadrics Surface Descriptions

In our grasp planning, we have adopted the following three grasping primitives:

- *Spherical grasp*
- *Cylindrical grasp*
- *Pinch grasp*

With the superquadric representation, we know the size of an object along its three major axes as well as the position and orientation, which can be characterized by a single homogeneous transform (the object frame). However, the 15-parameter superquadric representation of an object is not always unique [Sol87]. For instance, two different row-pitch-yaw combinations can represent the same object, but with the positive z-axis pointing in opposite directions. We make the representation deterministic by constraining each positive axis of the object frame to point towards a predetermined half of an imaginary sphere enclosing the object.

Based on the kinematics of the PUMA 560 robot and that of the PENN Hand, together with this superquadric representation of the object concerned , the grasp planner decides the approach vector, the desired grasping orientation, and which grasping configuration is most appropriate for the task. If a successful grasp is found, an approach vector is determined based on the additional constraint that one cannot easily change the wrist configuration of a PUMA560, since that would imply going through a singularity since we move in cartesian space.

Using analytical geometry we use the superquadric parameters described below to obtain the grasp parameters:

- a_1, a_2 and a_3 are the measures of the size of the object along its three principal axes. If the magnitude of these values and their relative sizes are within some predetermined range based on the geometry of the Hand, a *spherical grasp* which will enclose the object is considered first.
- If a *spherical grasp* is not possible, a *cylindrical grasp* around the major axis of an object and with the fingers closing on the shorter of the minor axis is preferred.
- If both of the above are not possible because the object is too small, a *pinch grasp* will be generated.

The position which the hand should approach from, the desired and final position and orientation of it are calculated and are formulated as homogeneous transforms respectively. These positions are obviously not unique, for instance, one can approach from one side of a rectangular block as supposed to the opposite side. However, the one which minimizes the arm movement in cartesian space is always preferred. These two transforms are then passed as key frames to the central coordinator for trajectory planning.

2.4 Planning Hand/Arm Interaction in Grasping

The first framework is setup by using the feedback from the visual sensor to create motion queues for the arm and hand, specifying among other points, an approach point, a expected contact point and a release position. The planner looks at the list of positions created by the segmenter, and determines the point at which the hand needs to start preshaping so as to be able arrive at the approach in the required hand configuration. The hand should preshape as late as possible, to allow it to remain in the *comply* position as long as possible. The *comply* position is achieved by placing the three finger-tips in an equilateral configuration around the palm which best allows the system to track or contour surfaces.

Thus, while moving to the approach position, the arm moves along the shortest possible path towards the approach point. If an obstacle is encountered the system reacts in different ways depending on which hand/arm interaction mode is currently enabled. In the *stop* mode, the dorsal fingertip sensors are monitored, and forces are complied with. This is a useful behavior when the hand is carrying a payload and collides with an obstacle. Alternatively, the *comply* mode is available in which the arm will attempt to zero out any forces it encounters while continuing towards its goal position. In *contour* mode the arm attempts to contour follow along the surface.

The *comply* algorithm reads the forces on the three fingertip sensors and uses a predetermined stiffness constant to compute a displacement for the reading. This displacement subtracted from the forward kinematics computation for the finger position for each of the fingers, gives a plane with reference to the tool frame of the robot.

In order to comply with the surface, the system attempts to find the surface normal, and then orient the approach vector of the tool frame along the same direction. This ensures that the hand is aligned to the surface normal in the static case. In the dynamic case, the fingertips tend to have varying forces, which change faster than the servo rate of the arm. In order to ensure that the arm does not get unstable, the fingers of the hand comply to reduce the force, by increasing the aperture. The force are still transmitted to the arm, which can now comply at a much lower rate, and with a much smaller gain. The fingers return to the fixed *comply* position as soon as the arm has moved sufficiently to allow them back again. Once the hand is in free space the arm reorients itself along the desired approach vector.

When an unexpected collision takes place, or we need to find a object lying on some surface, we would like to be able to track the surface, as opposed to constantly bumping into the surface, this is the purpose of *contour* mode. The aim is tracking of the surface, with no desire to model it or to extract much more than a the global surface normal of the encapsulated area. The three fingertip sensor serve as three compliant contact points. The sensors give us a reading of the normal contact force, and allow us to determine an expected slope. This scheme will work best when surfaces have continuity of curvature. The same algorithm, can be used to find discontinuities on objects, but this is useful, only if we are attempting to grasp the object.

On reaching the approach point the arm moves along a straight line until one of the fingers or the palm makes contact with the object. If a finger makes contact, the arm can be moved along the object, until all the fingertips contact the object at which point they open and then enclose the object. If the palm contacts the object then the fingers simply close around the object. If no contact has been made, once the failure point is reached, the arm backs off and attempts to get the next object.

Once the object is enclosed the hand maintains the desired force on the object. At present we assume that objects are rigid, and are more stiff than the elastomer that covers the contact areas of the hand. The grasping strategy relies heavily on using the frictional, adhesive, and compliance attributes of this elastomer to keep objects in stable prehension. The compressibility of the skin provides resistance to very large tangential forces even without a large normal force [Cut85, Wes84]. We monitor the normal forces from the contacting sensors, to ensure that net applied forces on the object are balanced, so as to avoid imparting undesired moments on the object. These measurements are not precise, since we do not have spatial resolution on the contact sensors, thus the actual contact point cannot be measured. Since the task to be accomplished is to grasp the object, and not to impart any torques or moments, the above scheme should suffice in all cases, where a suitable grasping configuration and orientation can be found.

2.5 Hand/Arm Run-Time Control Module

The hand-arm run-time controller model monitors the current motions, enforces dependencies between the hand and arm queues as well as the image coordinator queue, and constantly revises the

next motions based on current tactile sensing feedback.

The run-time controller system consists of three circular event queues which sequence the actions of the system. The queues control and coordinate the actions of the Penn Hand, the Puma Arm which on which the hand is mounted, and the Scanner/Arm Subsystem.

In general, a queue entry consists of an action, an appropriate set of parameter values for the action, a context set of prerequisite variables which must be true before the entry action is initiated and a status word which describes the current state of the given queue entry action. Implicit in the context list is the successful fulfillment of necessary actions which define an appropriate current state context in which the given action will execute. These context can include information gathered from tactile, kinesthetic or visual (scanner) means.

The hand queue elements consists of an hand action primitive, parameters for the primitive, a list of symbolic event name preconditions for arm events that must have been completed for the action to take place, and a status word. Currently, the primitives include a spherical grasp, a cylindrical grasp, and a pinch grasp. Grasp modes specify what actions the hand should take based on the values of force sensors on the fingertips and palm. The prerequisites for a hand motion consist of symbolic event names in the arm and scanner queue. The context consists of conjuncts of symbolic event names which must have been completed previously. The process controlling the hand queue monitors the queues for symbolic event names specified in the precondition, as well as the hand state which consists of a desired set of angles and force for the current given command. When all preconditions have executed successfully (including previous hand commands) the next hand queue action is fired with the appropriate parameters.

The arm queue consists of arm primitives and parameters for the primitive, a mode specifier, a precondition list, and a status word. Arm actions consist of desired tool position and orientation specifications.

The scanner queue consists of actions which may be taken by the scanner system. In general the consists of scanning a given location in the robot workspaces with a given orientation and a given resolution in the z direction. The scanner/arm subsystem then takes care of decomposing this fairly high level command into a series of subscans which must taken and merged in order to supply the desired data. This queue also has the normal prerequisite context which is used to ensure that scans are taken in a timely way, and in a fashion that will minimize the possibility of collision.

Queues are monitored in a round robin fashion and actions fired as soon as their preconditions are known to be satisfied.

2.6 Hand Module: Software and Hardware

The hand module is run on a PC-AT and linked to the coordinator by a 16 bit parallel bus. The module consists of a graphical user front-end and a communication front-end for the coordinator. Thus the user can type any of the commands via the keyboard that are normally sent over the parallel bus. The PC has 8 full sized slots that allow for motion controllers, D/A cards, A/D cards for the tactile sensors and encoder decoder cards to be plugged in. The motors of the hand are driven by externally mounted power amplifiers. All the data and actuation signals run from the PC to the hand, via a single 112 braided and shielded cable. This allows us to eliminate any heavy

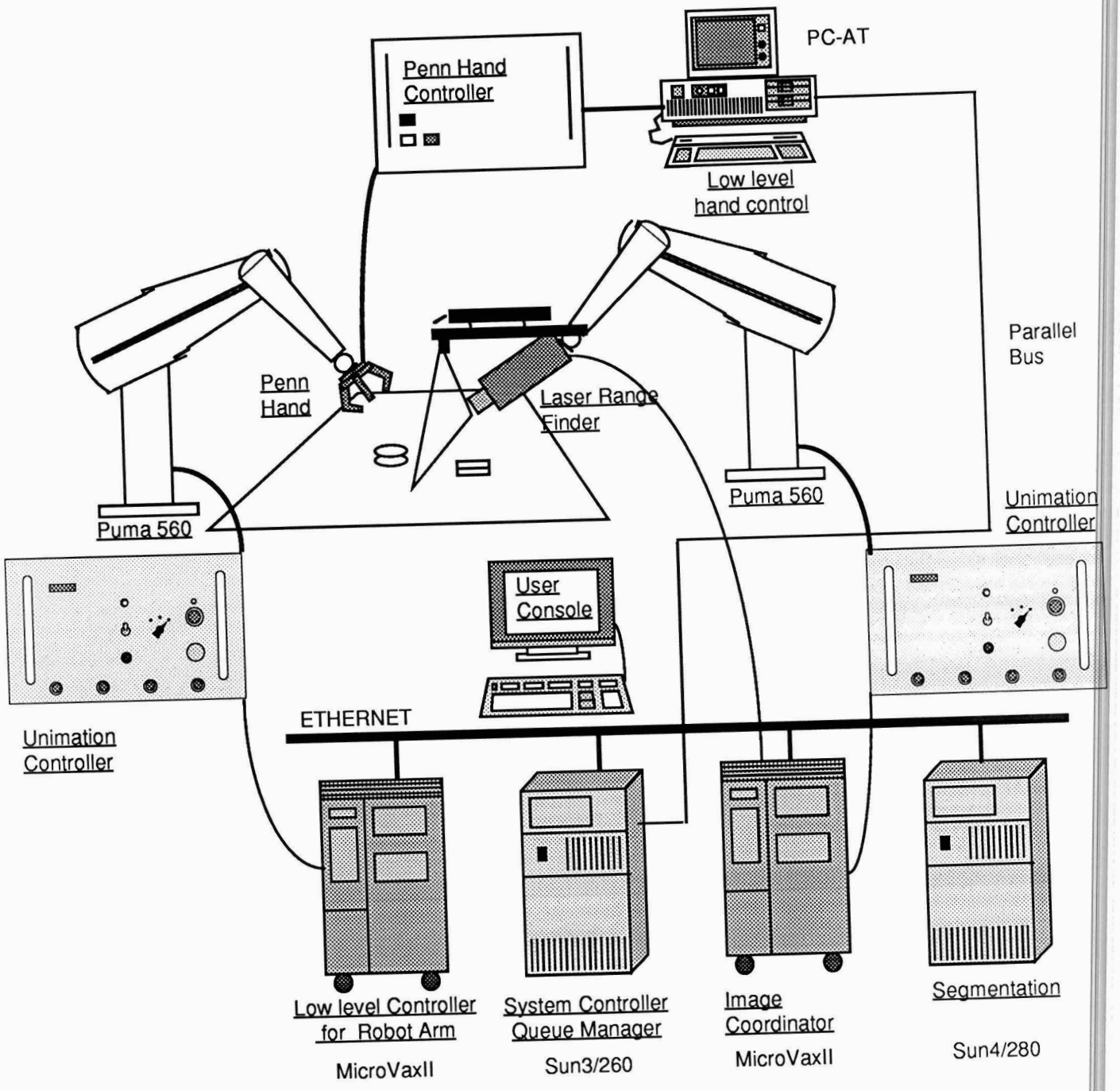

Figure 2: System Hardware Configuration

actuator packages from having to be mounted on the arm. The Penn Hand [Ulr88] itself weighs only 1.5 kgs and is mounted on the robot flange via a quick release mechanism.

2.6.1 Hand Command Description

At any given time, the hand controller has a current command, and a transition command. Since the hand communicates with the coordinator at a low bandwidth as compared with its servo rate, which can be varied between 400 and 700 hertz, the controller in the hand module must have an intermediate command to servo on when the current command has been accomplished, and the next command is sent down.

Commands are divided into 5 categories, servo_immediate, servo_transition, parameter, calibration and mode. In addition, commands can be addressed one or more joints. Joint dependencies can be specified. Dependency on a joint requires that the previous command for that joint be completed, before the the command sent is valid. Servo commands provide a desired joint angle, a combination of desired sensors to servo on. (Each finger as four spatially distinct sensors) Not all combinations are valid, for instance, one cannot expect forces on both the front and back sensors of the fingers.

Servo_immediate command causes the joints addressed to instantly switch to the given command.

Servo_transition command allows specification of other joint commands as dependencies for command execution

Parameter command allows you to specify the finger stiffness and joint velocity.

Calibration commands lets you reset the joint encoders and set the zero value of the sensors.

Mode command allows you to specify the algorithm the force servo uses.

The hand module acknowledges each command, after parsing it, and verifying the fact that it is a legal command. When the module is not parsing a command, it is sending out the the current joint positions, and the forces from the 14 tactile pads. Figure 3 shows the location of these sensors.

2.6.2 Hand Servo Control

The joints of the hand are controlled using a closed PD loop. Since a desired force may also be specified the controller can switch between monitoring the position or the force. Until a force is encountered on the sensor/s that the controller is currently monitoring, the controller is in the position servo loop. If the desired position is reached, the command is completed. If a force is encountered before or after the desired position is reached, the servo switches to force control, and tries to servo on the desired force, moving back to the desired position if the force is removed. Thus the controller switches between these two modes till a new command is sent down.

2.6.3 The Sensorized Penn Hand Description

The Penn Hand was designed to be a medium complexity end-effector, by which we imply, the ability to attain a wide set of hand configurations and grasps while at the same time requiring

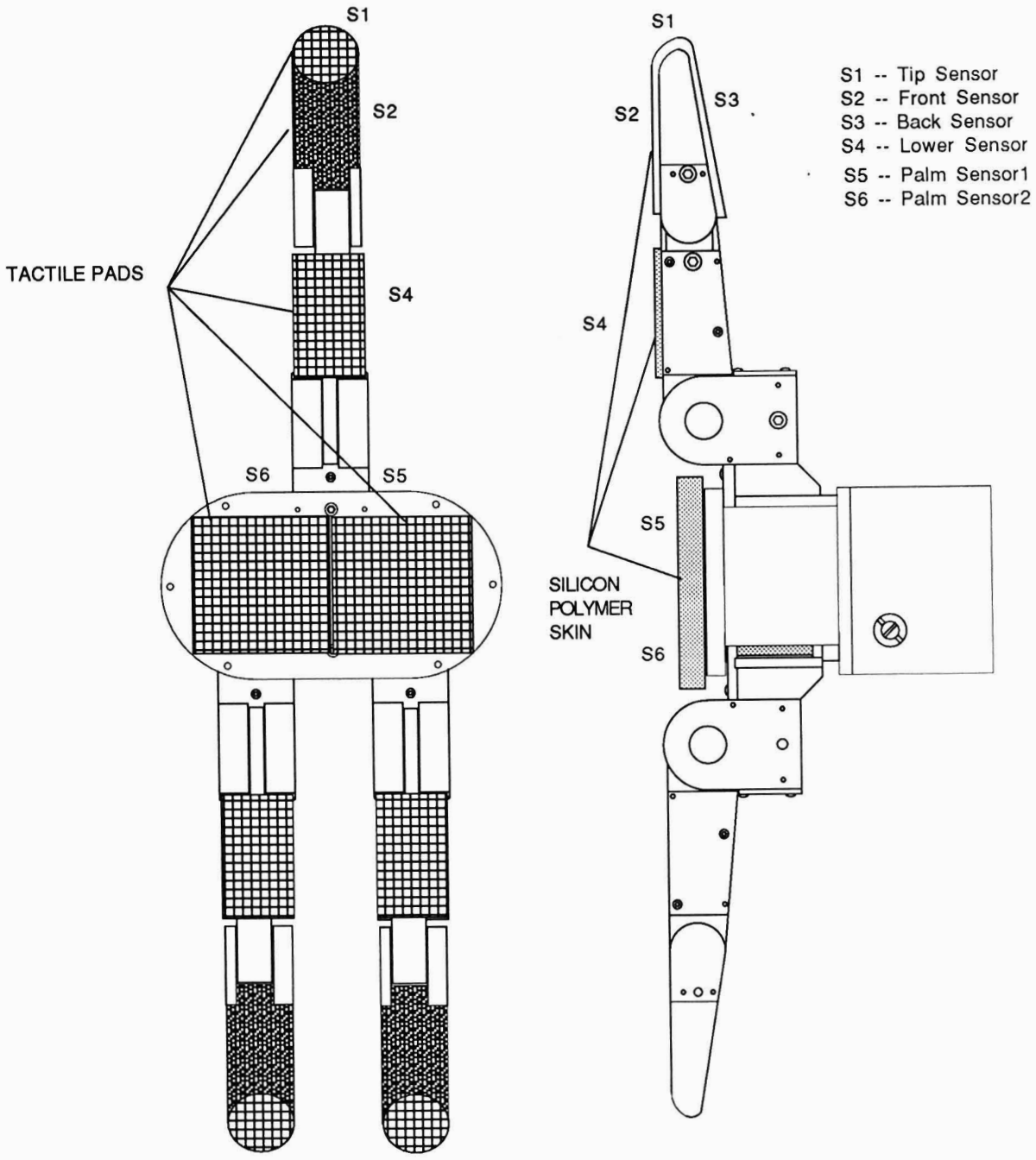

Figure 3: The Sensorized Penn Hand

minimal computational resources and a simple control scheme. The mechanics of the hand and its grasp modes are described in [Ulr88].

The hand is primarily a enveloping gripper, with the coupled fingers allowing us to enclose a variety of shapes and sizes. At the same time we have a pinch grip that allows us to pick up objects that are not suitable for enclosure grasps. The other distinctive provision of the Penn hand is the presence of a palm. Two of the fingers can perform a coupled rotation around the palm. The palm has a compliant skin over a set of Interlink sensors. The palm can be used as both as a support platform for holding objects, without actually having to grip them, as well as provide a large contact surface when we need to constrain the object beyond simply two or three finger contacts.

An important design consideration for the hand was the requirement that one should be able to furnish its surfaces with as many sensors as possible. The palm has two large planar tactile sensors, that cover the entire area of the palm. The fingers, which have two links each, have a total of four sensors on both links together. The lower link has one sensor covering the the palmar region, and the upper link which is roughly an ellipsoidal cylindrical, with an anthropomorphicly shaped tip. This link has three sensors, two covering the entire cylindrical area, and another sensor at the tip. Since these sensors are not arrays sensors, multiple sensors serve to provide us with very coarse spatial resolution.

The other important consideration in the design of the palm and fingers is the need for a compliant surface skin. The skin gives us the ability contact stiff surfaces without causing large interactive forces in the now coupled system. In addition the skin can be lubricated, to provide varying friction properties, depending on the task requirements. The skin can also withstand temperature of up to 300^o centigrade, allowing the hand to function in a variety of environments.

2.7 Arm Module: Hardware and Software

The arm servo subsystems for both the hand and the module run on two MicroVaxIIs respectively, which are interfaced to the Unimate robot controller via a parallel line that is tied in to an interrupt line. The operating system is DEVBUS a modified unix kernel, allowing for time critical operations to be performed.

The arm modules communicates with their coordinators via ethernet using AF_INET stream sockets that allow reliable communication between the coordinator and the controller. The arm controller which accepts differential cartesian rates[Cor89], runs at about 35 Hertz. These cartesian rates are run through the inverse Jacobian to obtain the joint rates which the controller then sends to the *Unimation* box. The arm module relays the current position and orientation of the tool, as well as the current configuration over to the coordinator. If the module does receive a new differential cartesian change from the coordinator within a fixed number of cycles, the controller will be asked to cut the joint rates to zero, until the next rate is sent down.

2.8 Software design and Algorithms

The software is designed with two fundamental requirements in mind. One is the need for the system to operate independently of the other manipulators and sensors, and two the ability to integrate

other sensors and manipulators, without modifying any of the existing modules. The only module in the system that must know about all the sensors and manipulators in the system, is the central coordinating module, which must initialize and startup each module. Since the communication package is a standard unix interface, any external program can communicate with the system once the coordinator is alerted to its presence.

All other process communication takes place over the 10Mbit/second Ethernet using a socket stream based reliable TCP/IP protocol. The ethernet system allows approximately 200 packets per/second to be transferred between the MicrovaxII and Sun workstation. A Sun4/280 is used to segment the range images. The systems uses a general purpose communication library for the explicit purpose of establishing XDR based connections between various hosts with a minimum of coding effort. The various different data representations are handled transparently by use of the standardized external data representation (XDR) which allows for simple transport of floating point data between the various internal representations of the different architectures.

3 Conclusion

What we demonstrate in this system is the need for integration of various sensory and manipulatory modules that can function in a coordinated manner.

The system represent several features in flexibility of active perception via the use of an active imaging system and mechanical exploratory procedures available with the sensorized Penn Hand. The hand represents a fairly sophisticated haptic probe in its own right as well as a competent manipulation device. The mobility of the scanner allows for minimizing the amount of shadowed area of scene that is inherent in structured light type range scanners. By varying the scan rate of the scanner via controlling the rate of the robot arm, variable resolution and control of scan size can be achieved.

The variable resolution control of the scanner permits the region of interest to be used to achieve a coarse to fine scene representation, where regions which have insufficient resolution or too many shadows may be rescanned or explored haptically to better characterize them. This flexibility is achieved at the cost of increasing the planning requirements for the system since more than one arm is now present in the workspace. However, unless the system has the ability to integrate the information obtained from the various sensors, and uses it to control the motions of all the manipulators, albeit in different strategies, we will not be able to work in environments that are partially structured.

References

[All89] Allen, P. K., Michelman,P., Roberts,K.S., An Integrated System for Dextrous Manipulation. Proceedings of 1989 IEEE International Conference on Robotics and Automation, Scottsdale, Az pp.612-617.

[Bic89] Bicchi, A. and G. Buttazi, Robotic Tactile Sensing: Skin-like and Intrinsic Approach. Intelligent Robotic Systems: Analysis, Design and Programming, Ed. Spyros Tzafestas, Marcel-Decker Inc, NY 1989

[Cor89] Peter I. Corke and R. Paul, *Video-Rate Visual Servoing for Robots.* GRASP Lab, University of Pennsylvania Technical Report MS-CIS-89-33

[Cut85] Cutkosky, M. Grasping and Fine Manipulation for Automating Manufacturing. Ph.D. Thesis, The Robotics Institute Carnegie Mellon University 1985

[Fea84] Fearing, R.S., Simplified Grasping and Manipulation with Dextrous Robot Hands. A.I. Memo 809, MIT Artificial Intelligence Laboratry, 1984

[Ges83] Geschke,C.C., A System for Programming and Controlling Sensor-Based Robot. Manipulators. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-5,No.1., pp.1-7, January 1983

[Gupta89] Gupta, A. *Part Description and Segmentation Using Contour, Surface and Volumetric Primitives.* Dissertation Proposal, MS-CIS-89-33, Grasp Lab 180, Department of Computer Science, University of Pennsylvania, 1989

[Hon90] Hong, J., Lafferriere, G., B. Mishra and X. Tan, Fine Manipulation with Multifingered hands. Proceedings of 1990 IEEE International Conference on Robotics and Automation, Cincinnati, Oh, pp.1568 - 1574,1990

[How90] Howe,R.D., Popp,N., Akella,P., Kao, I., and M.R. Cutkosky, Grasping, Manipulation, and Control with Tactile Sensing. Proceedings of 1990 IEEE International Conference on Robotics and Automation Cincinnati, Oh pp.1258-1263

[Jac85] Jacobsen, S.C., Wood J.E., Knutti, D.F., Biggers, K.B., and E. K. Iversen, The Version I Utah/MIT Dextrous Hand. Robotics Research: Second International Symposium, ed. H. Hanufsa and H. Inoue, MIT Press, 1985, 39-54

[Jac86] Jacobson,S.C., Iversen,E.K., Knutti,D.F., Johnson, R.T., and B. Biggers, Design of the Utah/MIT dextrous hand. Proceedings of 1986 IEEE International Conference on Robotics and Automation, San Francisco, CA, pp.1520-1532.

[Kun90] Kuniyoshi,Y., Inoue,H., and M. Inaba, Design and Implementation of a System that Generates Assembly Programs from Visual Recognition of Human Action Sequences. Proceedings of IEEE International Workshop on Intelligent Robots and Systems, IROS '90, pp. 567-574.

[Liu89] Liu.H., Iberall,T., Bekey,G.A., The Multi-dimensional Quality of Tasks Requirements for Dextrous Robot Hand Control. Proceedings of 1989 IEEE International Conference on Robotics and Automation Scottsdale, Az pp.452-457.

[Nar86] Narasimhan, S., Siegel,D.M., Hollerbach,J.M., Biggers, K., and G.E. Gerphiede, Implementation of control methodologies on the computational architecture for the UTAH/MIT hand. Proceedings of 1986 IEEE International Conference on Robotics and Automation San Francisco, CA, pp.1884-1889.

[Ngu90] Nguyen T. and H. Stephanou, A topological Algorithm for Continuous Grasp Planning. Proceedings of 1990 IEEE International Conference on Robotics and Automation Cincinnati, Oh pp.670 - 675

[Oka82] Okada T., Computer Control of Multijointed Finger system for precise handling. IEEE Transactions on Systems, Man and Cybernetics, Vol SMC-12, No 3 May 1982, pp.289-299.

[Par90] Park, Y., and G. Starr, Optimal Grasping Using a Multifingered Robot Hand. Proceedings of 1990 IEEE International Conference on Robotics and Automation Cincinnati, Oh, pp.689-695.

[Rao88] Rao, K., Medioni, G., Liu,H., Bekey,G.A., Robot Hand-Eye Coordination:Shape Description and Grasping. Proceedings of 1988 IEEE International Conference on Robotics and Automation Philadelphia, PA, pp.407-411, 1988.

[Sal82] Salisbury, J.K., Kinematic and Force Analysis of Articulated Hands. Ph.D. Thesis, Stanford University, July 1982

[Sal85] Mason, M.T., and Salisbury,J.K., Robot Hands and the Mechanics of Manipulation. MIT Press, Cambridge. 1985

[Sal85a] Salisbury, J.K. Brock, D. and Chu,S., Integrated Language, Sensing, and Control for a Robot Hand. Proceedings of International Symposium on Robotics Research, Gouvieax, France, MIT Press.pp.389-397. 1985. O.Faugeras and G.Girault (Eds.)

[Sol87] Solina, F., Shape Recovery and Segmentation with Deformable Part Models. Ph.D. Thesis, University of Pennsylvania, 1987 Technical Report MS-CIS-87-111.

[Sta89] Stansfield, S.A., *Robotic Grasping of Unknown Objects: A knowledge-Based Approach.* Sandia Report, SAN89-1087 UC-32, 1989

[Sta90] Stansfield, S.A., Haptic Perception with an Articulated Sensate Robot Hand. Sandia Report SAND90-0085, UC-406, 1990. Sandia National Laboratory

[Tsik87] Tsikos, C.I., *" Segmentation of 3-D Scenes Using Multi-Modal Interaction Between Machine Vision and Programmable, Mechanical Scene Manipulation"* Ph.D. Dissertation, University of Pennsylvania, Department of ComputerScience, 1988

[Ulr88] Ulrich, N., Paul, R. and R. Bajcsy, A Medium Complexity Compliant End-effector. Proceedings of 1988 IEEE International Conference on Robotics and Automation Philadelphia, Pa, pp.434-437

[Wes84] Westling G. and R.S. Johansson, Factors Influencing the Force Control During precision Grip. Experimental Brain Research 1984, Vol 53 pp.277-284