

RESEARCH DIRECTIONS IN ROBOTICS¹

D. Atkins	E. Leith
E. Delp	H. McClamroch
W. DeVries	T. Mudge
G. Frieder	A. Naylor
E. Gilbert	G. Ulsoy
R. Howe	R. Volz
K. Irani	J. Whitesell
Y. Koren	K. Wise
G. Lee	T. Woo

October 1982

CENTER FOR ROBOTICS AND INTEGRATED MANUFACTURING

Robot Systems Division

COLLEGE OF ENGINEERING

THE UNIVERSITY OF MICHIGAN

ANN ARBOR, MICHIGAN 48109

¹This work was supported in part by the Air Force Office of Scientific Research/AFSC, United States Air Force under AFOSR contract number F49620-82-C-0089 and the Robot Systems Division of the Center for Robotics and Integrated Manufacturing (CRIM) at the University of Michigan, Ann Arbor, MI. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the funding agencies.

RESEARCH DIRECTIONS IN ROBOTICS

PREFACE

The Center for Robotics and Integrated Manufacturing has recently received a major contract from the Air Force Office of Scientific Research. A large majority of this funding is designated for the Robot Systems Division. This report is an excerpt from the proposal to AFOSR which describes the work to be performed in robotics under that contract. It does not describe all of the activities in the Robot Systems Division, however, particularly as several new staff have joined the Division. Nevertheless, it does give a representative view of a broad spectrum of division activities.

RESEARCH DIRECTIONS IN ROBOTICS

The principal long term goal of the research in the Robot Systems division of CRIM is the architecture of intelligent sensor based robot systems. This goal is interpreted in its broadest sense, encompassing comprehensive research on subsystems which will comprise advanced robot systems, algorithms for intelligent use of sensor information, and the implementation of robot systems of evolving complexity (beginning with enhancements to current robot systems). Important research topics range from highly sophisticated and accurate control algorithms for arm motion, development of new types of sensors, use of advanced sensor information, and algorithms for collision avoidance, to higher level languages for robot control, and integration of robot systems with CAD databases.

This comprehensive view of research is necessary to achieve our long term goals; there are many significant problems to be overcome. At a low level, substantial additional work is needed on fast highly accurate control of manipulative mechanisms because current robots, when adjusted for high speed operations, exhibit considerable low frequency vibration which can cause difficulty in assembly of small components. Expertise on sophisticated aerospace control techniques will be joined with robotics expertise in the application of advanced control principals to robots.

This ability to see and feel will be central to expanded robot applications, e.g. to assembly operation, in the future. Solid state sensor development experience in Michigan's Electron Physics Laboratory will be directed toward new robot sensors, such as tactile, environment and proximity sensors. New arrangements for existing sensors, e.g. stereo vision, will be explored, and new algorithms for utilizing the information provided these sensors will be developed. Particular emphasis will be given to the development of algorithms for real time sensor feedback.

A unique feature of our research capability is the ability to implement computationally intensive algorithms (which are expected to result in several of the above areas) in the form of special purpose computer architectures or even special purpose VLSI chips. Indeed, as evidenced by the award winning paper [MUD81] this approach has already been applied.

In order to develop an intelligent robot system which can be readily used by a broad spectrum of users, suitable higher level languages must be developed. They must include, in convenient form, the ability to utilize and process the information provided by a wide variety of sensors. An appealing approach here is the use of interactive graphics coupled with information from Computer Aided Design databases. Geometric information about the parts being assembled can be coupled to a two dimensional visual recognition system to provide near real time three dimensional information, and simple display of the parts may be coupled to an interactive graphic programming method.

At a still higher level there are questions of the application of artificial intelligence and knowledge based systems to robotics task specification. It is very important that user oriented ways be found for the specification of the tasks robots are to perform and that these be translated automatically into appropriate robot motions.

The following sections will present in greater detail a set of problems to be addressed by this proposal over a long period of time, and a set of specific tasks

to be pursued during the first year of the proposed activity.

1. The Control of Mechanical Manipulators

(W. DeVries, E. Gilbert, R. Howe, Y. Koren, G. Lee, H. McClamroch, T. Mudge, G. Ulsoy, J. Whitesell)

1.1. Introduction

A mechanical manipulator can be modelled as a kinematic chain with several rigid bodies (links) connected in series by either revolute or prismatic joints driven by actuators. Current manipulator applications such as material-handling (pick and place), spray painting, spot/arc welding and the loading and unloading of numerical control machines place only moderate demands on control technology. Future tasks such as small part assembly require that much more attention be given to manipulator structure, dynamics and control. The current treatment of each joint as a simple servomechanism is inadequate because it neglects gravitational loading of the links, the dynamic interactions among the links and elasticity. The result is reduced servo response speed and damping, which in turn limits the precision and speed of the end-effector. Significant gains in manipulator performance require the consideration of structural dynamic models, sophisticated control techniques, and the exploitation of advanced computer technology. Major research effort will be directed toward the following areas:

- Accurate modelling of robot arm motion: different formulations of rigid-body equations, vibrational dynamics, discontinuous nonlinearities, actuator effects.
- Computer simulation: development of special software packages, utilization of ultra-high-speed computers, graphics display techniques.
- Optimization of robot arm motion: various constraints on control and state variables, different optimization criteria, numerical methods, practical implications, good suboptimal motions.
- Nonlinear multivariable control: decoupling methods, stability improvement, treatment of discontinuous nonlinearities, computer-aided design.
- Adaptive control: feedback gain modifications, load sensing, process modelling and identification.
- Computational structures: implementation of complex control laws, special purpose processors, distributed versus centralized control, reference-pulse and sampled signals.

The intent of the research is to improve the overall capability of computer-controlled robots and to influence the design of new, more advanced robots.

1.2. Background

The movement of a manipulator (robot arm) usually involves in two distinct control phases. The first is a gross-motion control in which the arm moves from an initial position/orientation to the vicinity of a desired position/orientation along a planned trajectory. The second is a fine-motion control in which the end-effector stays in the neighborhood of the target

object. During fine-motion the arm completes the desired task by dynamic interaction with the object using sensory feedback.

Several control schemes exist for the gross-motion phase, all neglecting structural dynamics: (i) near minimum-time control of a restricted arm [KaB71], (ii) trajectory/path control [Pau72, Pau75], (iii) resolved-motion rate control [Whi69], (iv) cerebellar model articulation control (CMAC) [Alb75, Alb75b], and (v) suboptimal and adaptive control [SaL79, Hot80]. In order to use these control schemes, an appropriate set of differential equations must be derived to model the dynamic behavior of the robot arm. Various approaches have been used to formulate the rigid-body motion of a robot arm such as the Lagrange-Euler [Lew74], the "Recursive-Lagrange" [Hol80], the Newton-Euler [LWP80], and more recently the "Gibbs-Appell" [HoT80] formulation. During the motion of the arm, the actuator inputs must be determined for every set point on a precomputed arm trajectory.

These control schemes are based on a dynamic model of the manipulator [Uic67, Pie68] in which both joint position-dependent and time-dependent terms arise. It takes about eight seconds (FORTRAN simulation) to compute the move between two adjacent set points for a six jointed Stanford arm, using a PDP-11/45 minicomputer [LWP80], and involves approximately 2,000 floating-point multiplications and 1,500 floating-point additions per joint. To improve the speed of computation, simplified sets of equations have been used by other investigators. One set was formulated by Paul [Pau72] and is an approximation of the complete Lagrangian formulation. It requires between 83 and 258 floating point operations per joint (depending on the joint structure of the arm). These operations together with input/output management operations can then be performed once every 1/60 second [Pau72, Lew74]. Their "approximate" solutions are based on models which simplify the underlying physics by neglecting second order terms [Pau72, Bej74]. At high speeds the neglected terms become significant, making the accurate positioning of the arm impossible. This has been observed in the PUMA robot of our Laboratory.

An approach proposed by Paul [LWP80], based on a Newton-Euler vector formulation, yields a set of recursive forward and backward equations which can be applied to the links sequentially. Because of the nature of the formulation and the method of systematically computing the torques, it is possible to achieve a short computing time. Together with the time taken for other relevant computations and the input/output management, this algorithm takes 5.5 ms to compute the input torques per set point using a PDP 11/45 computer. This is fast enough for real-time control if one does not need to process the sensor feedback signals.

Since fine-motion control depends on the nature of the task being performed and generally involves sensor feedback, it is a more demanding problem than the gross-motion control. Generally, task-related issues such as sensor configuration and higher level control strategy have been the primary concern [BoP73], and the performance specifications of contemporary computer-based robots have been sufficiently low that simple control systems have been adequate. Needs now arise for higher performance in more complex tasks performed over wide ranges of motion and with varying payloads. In order to achieve high performance it is necessary to now consider more precise and advanced control schemes, and in some cases the structural dynamics of the manipulator.

1.3. Modeling and Simulation in Robotics

It is clear that computer simulation can play an important role in the design, development and testing of robot systems. The need for comprehensive simulation as part of the research into various robot designs is obvious. Elastic effects, coulomb friction and backlash, complex nonlinear and discontinuous controllers, actuator and sensor dynamics all present challenging problems in modelling and simulation. By including such effects it is possible to evaluate the dynamical performance of a robot, using such performance criteria as dynamic stress and strain, overshoot, tracking error and vibrational resonances. Real-time simulation is essential for hardware-in-the-loop testing, as well as providing a tool for research into human/robot interaction and graphic displays. Major portions of laboratory test equipment may even be replaced by computer simulation. Such directions will be pursued.

The College of Engineering has developed considerable expertise in digital simulation using both general purpose and special purpose digital computers. In the general purpose computer category, a number of software packages have been developed for simulation of rigid-body motion of robotics systems; this includes DRAM for simulation of two-dimensional systems and ADAMS for simulation of three-dimensional systems. Further development of such software, including addition of capability to simulate both rigid body motion and elastic motion, will be pursued.

Of particular significance for simulation with special purpose devices is the availability of an AD-10 pipelined multi-processor computer interfaced to PDP 11-34 host computer. The AD-10 architecture has been optimized for the solution of nonlinear differential equations and can perform up to 30,000,000 additions and multiplications per second. It has been applied extensively by the aerospace industry for simulation of missiles, aircraft and helicopters; AD-10s are currently being used by NASA for real-time simulation of the space-shuttle remote manipulator arm. This computer facility permits cost-effective simulation of complex robot dynamics, especially in real time. The facility includes a high speed A/D and D/A interface for external hardware tie-in, including graphic displays. It is proposed to make substantial use of the AD-10 facility for simulation of robot dynamics.

In addition to utilization of digital simulation techniques which have already been developed using our AD-10, it is proposed to continue the development of improved techniques for representing mechanical nonlinearities (e.g., friction, backlash, etc.), discontinuous controllers, sensors, actuators, real-time integration algorithms, and other areas of obvious importance in robot design and simulation. High-level software to simplify modelling is also an important area of development.

1.4. Optimization of Robot Arm Motion

Optimization of robot arm motion is a technologically difficult but potentially rewarding problem. The potential rewards include: improved performance of existing hardware, future performance gains by effective exploitation of the developing computer technology, generation of efficient command signals in the presence of practical constraints. The basic mathematical problem is one in optimal control with complex nonlinear equations of motion, active control constraints (force and torque limits), state-variable equality and inequality constraints (velocity limits and interference restrictions), and a number of possible criteria of optimality (such as minimal time or minimal peak acceleration of gripper).

Initial efforts will focus on the formulation and solution of a variety of optimal control problems associated with the optimization of robot arm motions from one initial configuration to a final configuration. Study of such optimization problems should prove effective in improving the current point-to-point control techniques. Since the computational difficulties in solving nonlinear optimization problems with constraints are great, attention would be given to reduced model complexity (keeping dominant nonlinearities and constraints), development of computationally efficient algorithms and the use of special high speed computers. Implementation of on-line optimal control requires almost immediate generation of the optimal motions. Approaches which would be studied include the use of simplified suboptimal motions, piecing together of special optimal solutions and use of vast computer memory resources for storage of a large family of precomputed optimal motions.

1.5. Adaptive and Advanced Feedback Control Techniques

The feedback control of arm motion can be viewed as large-scale, nonlinear, multivariable control problem. For the most part current design practice treats each joint as a conventional servomechanism problem. Effects of small-scale discontinuous nonlinearities (e.g., hysteresis and friction), changes in arm geometry with motion, interactions between the control loops and other complications are considered, but only in a crude way. Increased demands on position, speed and accuracy will require more sophisticated control techniques. The applicability of current theories and methods would be explored both in general terms and in the laboratory. Where state-of-the-art results prove inadequate, new ones will be investigated.

A fundamental issue is the control of rigid body arm motion in the presence of changes in arm geometry and loading. Experience at Michigan with the control of nonstationary systems, such as CNC machine tools, has shown that adaptive control systems can be effectively applied. In robot-arm control adaptive systems may be applied to individual arm segments or to control all arm segments simultaneously. The design of adaptive control systems for robot arms would involve the following subtasks: (1) the design of an effective conventional feedback control system without consideration of the effects of parameter variations. (2) the design of a system for on-line estimation of the time varying parameters. (3) the development of an effective variable-gain controller-tuning or adaptation strategy. Interaction of the feedback control with the computer developed path would also be addressed.

An alternative to adaptive control would be the scheduling of feedback gains as a function of arm geometry and (known) load. Specific methods for determining effective schedules would be examined and trade-offs between controller complexity and control system performance would be considered.

An entirely different approach to the control of arm motion is the use of nonlinear decoupling theory. This would transform the large-scale nonlinear multivariable control problem into a new problem which could be treated as several small-scale problems. Advantages include feedback generation of complex arm motions and the synthesis of complex nonlinear control laws effective in all arm geometries. Since the known nonlinear decoupling theories have not been applied previously to such complex systems, both practical and theoretical questions must be studied. For example, complex calculations (differentiation, function inversion) involving nonlinear functions of several variables are required and efficient computer mechanization of them will have to be developed.

Small-scale nonlinearities have a profound affect on the accuracy of fine-motion control. For example, errors due to joint friction may be reduced by high feedback gains, but this may lead to control system instability. Various methods for analyzing and treating such problems in the multivariable context, including the use of accurate simulation techniques, will be studied. The resulting knowledge will reduce the need for stringent tolerances on mechanical/hydraulic/electrical components.

The use of redundant actuators and sensors has been considered in fields such as flight control to improve reliability and control accuracy. For instance, actuator or sensor failures may be detected and acceptable control maintained in spite of the failure. In addition, it may be possible to estimate important variables that are not directly sensed. The implications of such advantages would be studied as they relate to the design of feedback controllers for robot arm motion.

Because of the complexity of robot-control systems, computational aids are essential. Although computer-aided control-design packages are currently available, it would be desirable to modify them or develop new ones to meet special needs as the research program proceeds. Objectives would include automated computation of necessary design data and effective graphical displays.

1.6. On-Line Identification and Prediction

As an alternative to the use of general purpose sensors for fine-motion control, the use of feedback information obtained directly from the manufacturing process should be investigated. Seam tracking in arc welding serves as an example. Arc length could provide faster tracking speeds than current computer vision technology and would eliminate the need for extraneous transducers. To use such an approach, a mechanistic dynamic model would have to be developed, that could determine the arc length from readily measured arc characteristics, viz, arc voltage, current, wire-feed and welding speed.

In addition to developing a mechanistic model for the process, on line smoothing and signal enhancement would be needed, to account for stochastic variations in the measured process characteristics. The noise structure, as characterized by time series models could be used for this purpose. This would require the development of on-line model identification and prediction algorithms.

1.7. Structural Dynamics and Control

Most attention has been focused on control of rigid-body robot motions. But as the weight of the robot arm links is decreased flexibility of the links is increased and the potential for undesirable vibrational motion must be examined. There are substantial difficulties in development of mathematical models which describe the general motion of flexible links. In most cases the vibrational motion is superimposed on the rigid body motion; such decoupling is especially convenient for design and analysis of control systems. Suitable mathematical models for the vibrational motion can be developed using finite element techniques or other modal approximation techniques.

Control of the rigid body motion of the robot arm links is essential; if additional accuracy on the motions is required it may be desirable to control the bending or vibrational motion of the arm links as well. Passive damping of the vibrational motion is often achieved through the structural design of

oversized arm links. Such a weight penalty may be avoided by more sophisticated control. Passive damping can be obtained by nonstructural means such as fluid dampers. Or active damping of the vibration motion can be obtained using feedback control based on existing or additional actuators and sensors. Issues such as the number and location of actuators and sensors, as well as suitable forms for the control laws, will be addressed. Whether active vibration controls are used or not, it will be important to develop filtering methods to reduce or eliminate the spillover effects of uncontrolled modes on the sensor measurements and of unobserved modes on the actuators torques.

The including of flexibility effects in the models used for control system design will represent a substantial advance in the state-of-the-art of control design for multi-arm robots.

1.8. Computational Structures for Enhanced Robot Performance

The need for multi-robot systems working in cooperative tasks is arising, especially in the "factory of the future". In such multi-robot configurations, a centralized control may be inefficient and unreliable. A better solution may be to use a decentralized control with distributed local controllers, one for each robot. A configuration which is being studied in the laboratory at Michigan is shown in Figure 5.1. Research with this system will focus on the coordination schemes for the supervisory computer, the control architecture of the local controller and the control implementation technique in the local controller.

One problem which would be studied is the detailed structure of internal servo loops. Current industrial practice employs reference-pulse or sampled-data techniques in the servo control loop of the robot. With the first technique, a local controller (usually a microcomputer) produces a sequence of reference pulses for each axis-of-motion (e.g. ASEA robots), each pulse generating a motion of one increment of axis travel. The number of pulses represents position and the pulse frequency is proportional to the axis velocity. With the sampled-data technique, the servo loop is closed through the computer itself. The control program compares a reference set point from a planned trajectory with the feedback signal to determine the position error (e.g. PUMA robots). This error signal is fed at a fixed time interval to a D/A converter, which in turn supplies a voltage which commands axis velocity. It would be of interest to evaluate the performance of Reference-Pulse and Sampled-Data techniques for various robot applications. In the Sampled-Data technique, the effect of the sampling-rate on the dynamic performance will be investigated. The dynamic performance of a system which includes one control computer vs. a system which contains a microprocessor for each individual control loop will also be analyzed.

As mentioned earlier, due to the nonlinearity and complexity of robot arm dynamics, real-time computations of the control functions impose several requirements on control computers. A viable approach is to design a special purpose processor dedicated to arithmetic-intensive control computations. We are working in this area of computer architectures using a single chip VLSI implementation. Circuit densities commensurate with levels of integration projected for the mid-1980s are assumed. The proposed processor, termed the Numerical Processor (NP), is a 32 bit floating point machine using an MOS technology.

The design philosophy of the NP is oriented towards a dedicated arithmetic-intensive application, in particular, the control of a robot arm

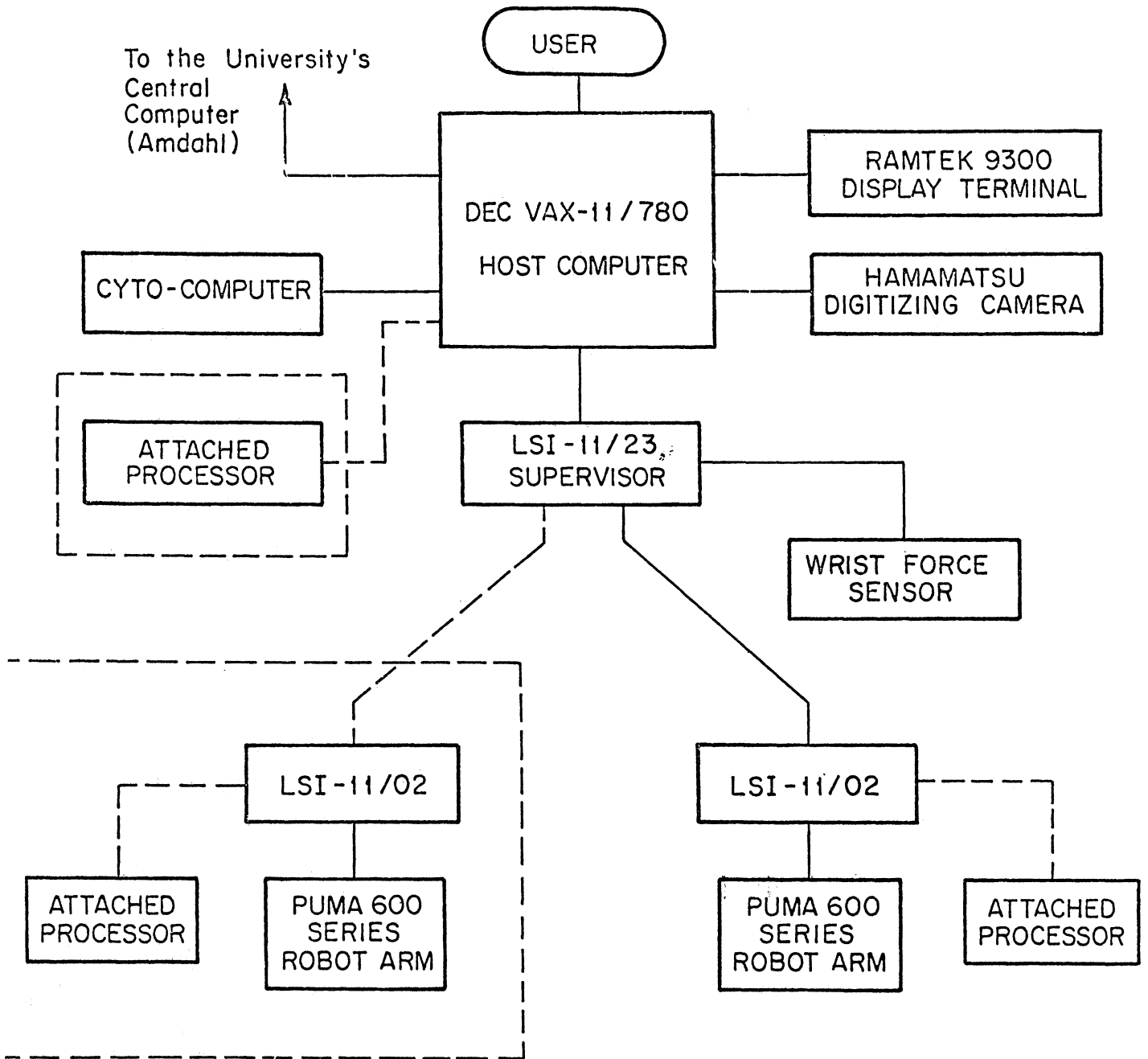


Figure 2. Physical System Structure for the Proposed Research.

where the arm response is limited by the complexity of the control computations. The specific robot arm in mind is the PUMA 600 series robot arm. The overall control strategy for a robot arm involves five basic stages. These stages are:

- (1) A path planning stage.
- (2) Orientation matrix calculation.
- (3) Trajectory transformation.
- (4) Gross motion control.
- (5) Fine Motion (Accommodation).

Simulations with the SPICE program of gross motion control for the PUMA robot arm have recently been performed to verify our design. Assumptions about gates delays were based on SPICE simulations using parameters measured in our nMOS process. Based on these (conservative) figures we obtained processing rates of about 2.7 MFlops for the NP. Joint torques can be computed every 500 microseconds. This encouraging performance indicates that the NP would be a valuable component for the design of real-time systems. Our proposed research will continue and include:

- (1) hardware versus software tradeoffs,
- (2) further simulation with more complex control tasks,
- (3) improved designs of NP based on this work.

2. Sensors and Sensor-Guided Control (Lee and Wise)

2.1. Introduction and Background

The rapid and continuing advances in microelectronics has contibuted significantly to the recent surge of interest in robotics worldwide. The availability of low-cost, high-performance microprocessors and semiconductor memory is making robotics feasible and practical to an extent not previously possible. No matter what the sophistication, of the control complex, the computer will have to depend on a sophisticated array of sensors and actuators to interact with its environment efficiently. The lack of such sensors is a major problem preventing the implementation of closed-loop control strategies, and as computer technology improves, it is certain to rapidly become even more of a problem. Classes of sensors needed for robotics include:

- Force/torque
- Tactile
- Electro-optical imaging
- Proximity
- Range-imaging

This section proposes the development of several sensors and control strategies to use them future robotics systems.

2.1.1. Solid-State Sensor Research at Michigan

If the recent past is any indicator, the needed devices will be based on solid-state process technology and will present a marriage of custom silicon interface circuits with silicon and non-silicon transducers on the same chip. Such devices take advantages of the extensive process technology developed for integrated circuits and are characterized by small size, high reliability, low cost, and relatively high performance. Many of these devices are still

exploratory but they are evolving at an increasing rate. A recent collection of papers in this area is found in the references [MeW79].

During the past few years, sensor research at Michigan has concentrated on [CIW79; BoW79; LeW82], infrared [LaW80], and on composite sensors for pressure, temperature, and pH. All of these devices depend on the precise shaping of three dimensional microstructures. Improved shaping techniques [JTW81] have been developed which promise to make these sensing structures practical. Work is also underway to develop powerful computer-based simulation programs for these devices, the first of which will soon be published [LeW81].

In the pressure sensing area there are currently two competing approaches. The first is based on a full bridge of pyroresistors diffused into a thin silicon diaphragm formed by selective chemical etching. These devices are small, highly linear, and have a relatively high pressure sensitivity; however, they also have a relatively high temperature sensitivity. The origin of this temperature sensitivity is now relatively well understood and the devices promise to see wide use in coming years. The second approach utilizes the thin diaphragm as the movable plate of a variable gap capacitor. The main advantages of these devices are their high pressure sensitivity (more than 1000 ppm/mmHg) and very low temperature sensitivity (less than 0.1 mmHg/ ° C). The capacitance values involved are very low, however, demanding on-chip circuitry.

A similar structure with quite a different application is being developed. Here the diaphragm is very thin (formed using a diffused boron etch-stop) and supports an array of thermocouples. Incoming infrared radiation heats the diaphragm area and develops an output voltage proportional to the incoming radiation level. Such devices have exhibited a responsivity of 10 volts/watt and a time constant of less than 15 msec. They have advantages over photon-based devices in that they possess a very wide spectral response, are low in cost and do not require cooling. Both bismuth-antimony and polysilicon-gold thermocouples have been studied.

2.1.2. Closed-Loop Sensor-Guided Control

The research problem in the closed-loop sensor-guided control is to find a control strategy which utilizes the force feedback signals from a sensor mounted in the wrist to appropriately servo the arm to track the desired force and position trajectories in completing the assembly tasks.

The need for researching more effective force control strategies can be seen from the following:

- Most common manipulator task failures occur during the terminal accommodation/compliance phase of manipulation [Bej76].
- The fraction of time spent to complete the terminal accommodation/compliance phase of manipulation is much longer than the fraction of time spent to perform the gross transfer motion of a manipulator.
- It is more cost-effective to use a less accurate, low cost manipulator with external sensors than a highly-accurate and expensive manipulator to perform high-tolerance assembly tasks.

Past work in force control has been performed at various institutions using joint sensors, pedestal sensors and even torque sensing by monitoring the armature currents of the motors. The first sensor-controlled manipulator was demonstrated by Ernst [Ern62] at MIT in 1961. The

computer-controlled mechanical hand MH-1 was equipped with tactile sensor which could "feel" blocks and stack them without assistance from the operator. In 1962 Tomovic and Boni [ToB62] developed a prototype hand equipped with pressure sensor which sensed the object and supplied an input feedback signal to a motor to initiate one of the two grasp patterns. These basic sensor control schemes were heuristic in nature and the control algorithms for the hand were crude.

In 1973 Bolles [BoP73] demonstrated the assembly of a water pump by a computer-controlled Stanford arm using both visual and force feedback. Force feedback techniques together with a heuristic circular search were used successfully in locating the holes for assembly. The sensing of forces was done via monitoring the armature currents of the joint motors. Though the assembly of water pump was successful, the fraction of time spent to perform force feedback control was much too long. About the same time, Will [WiG75] and his associates at IBM developed a computer-controlled manipulator with touch and force sensors to perform mechanical assembly of a 20-part typewriter. Inoue [Ino74] at the MIT Artificial Intelligence Laboratory worked on the artificial intelligence aspect of force feedback. A landfall navigation search technique was used to perform initial positioning in a precise assembly task.

In 1976 Whitney [Whi76] presented a force feedback strategy called accommodation. The method was simple and the strategy was embedded in the force feedback gain matrix (or compliance matrix). At the same time, Paul and Shimano [PaS76] extended the work at the Stanford Artificial Intelligence Laboratory by using compliance. His analysis showed that the computation load exceeded the desired sampling period. An "approximate" solution was devised for the compliance control. Shimano [Shi78] implemented the resultant compliance control method on the AL system. His control scheme was open-loop and consisted of motions controlled by external forces to comply along the desired joint coordinate axes. Nevins, et al [Gro72, NeW74, Dra77] investigated the amount of information from the compliance of the environment. This work developed into the instrumentation of a passive compliance device called remote center compliance (RCC) which was attached to the end of the manipulator for close parts-mating assembly. Although the assembly time is cut to affordable limit, its added cost is questionable to its popularity. Moreover the device's sensitivity is coupled to the mechanical deflection, an undesirable effect.

Almost all the force feedback control schemes work in conjunction with some heuristic search techniques, and heuristics usually leads to undesirably long assembly time. Moreover, these control strategies are limited in scope due to their inability to improve their performance based on past experience. One of our major goals is to design an active compliance control that builds on these strategies and incorporates self-improvement by utilizing pattern recognition and learning techniques.

In addition to developing improved strategies that will allow faster and more accurate execution of the arm's fine motion control, research effort will be directed toward developing the force control structure/architecture (or control hierarchy) to support the computations for these strategies in order to ensure the speed required for real-time control of the overall arm motion.

2.2. Proposed Research

Research is proposed in four specific areas relevant to robotics: (1) infrared imaging array, (2) an improved wrist force/pressure sensor, (3) a

tactile imaging array, and (4) sensor-guided control strategies. Each of these is described briefly below.

2.2.1. Infrared Imaging Arrays

There is high interest in the development of improved infrared imaging arrays for a variety of applications. In robotics, such devices would add another dimension to the vision process and would allow shadowing effects to be overcome. While such arrays would not be useful in all applications, they would likely be valuable for many.

Most approaches to infrared imaging today are concentrating on photon-based devices which merge silicon circuitry with non-silicon detectors to realize hybrid focal plane arrays [CRS82]. This approach has many advantages but it also requires fairly complex fabrication technology and the resulting arrays must normally be cooled to liquid nitrogen levels to provide attractive signal-to-noise levels. Robotics applications probably preclude such cooling but also probably do not require great sensitivities or large array sizes. Infrared will most likely be used as a supplement to the visible imagers and most as a stand-alone high-resolution device.

In this portion of the research, we propose to evaluate the thermal detectors (thermocouple devices) mentioned in the last section in terms of their suitability for use in small infrared arrays. This research will proceed in the following steps:

- (1) Based on the models and data already developed for silicon thermopiles, the performance of infrared detector arrays will be calculated as a function of the window (diaphragm) size. Windows having center-to-center spacings of 1 mm and up will be evaluated and compared to the expected performance of hybrid photon-based devices. These comparisons will include the anticipated effects of on-chip circuit noise.
- (2) If the thermopile devices are found to be viable for the anticipated needs of robotic systems, a complete detector chip (thermopile plus amplified plus access switch) will be built and tested.
- (3) Based on the above tests, a hybrid array of the detectors will be constructed so that the use of a simple infrared array can be evaluated for use with a robot. The anticipated array size here is expected to be either 8x8 or 16x16. The use of "smart" computer algorithms for image enhancement will also help to define the required size of such arrays.

2.2.2. Improved Force/Pressure Wrist Sensors

The wrist sensor holds an important place in robotics and amounts to a single point monitor of the forces on the hand (gripper) with the ability to resolve them into both magnitude and direction. Such sensors must be rugged, reliable and have a high dynamic range. The monitoring processor must be able to separate applied forces from forces associated with the dynamics of motion.

For the implementation of the wrist sensor, the use of strain gauges is common. While the temperature drift associated with these elements can presumably be compensated by the microprocessor during known non-contact periods (dynamic recalibration), the attachment of these elements to the bending members leads to unpredictable long-term drift and uncertain reliability. For non-silicon gauges, the output voltages (sensitivity) is also very low so that noise becomes a concern.

Improved force sensors may be possible using piezoresistive cantilever-beam transducers or pressure sensors mounted directly on the wrist. Such devices would possess a high sensitivity and would allow improved directionality in resolving the force components. High dynamic range could be preserved along with high sensitivity by using several devices in each direction, each with a different beam/diaphragm thickness and hence sensitivity range.

In this portion of the research, we would propose to first thoroughly evaluate the Scheinman force sensor using strain gauges to examine the problems with this approach. In parallel with this effort, designs using cantilever structures and pressure cells will be developed and compared with the strain-gauge approach. Depending on the results of the Scheinman tests, the most promising of the alternative designs will be implemented and characterized. The goal will be to not only develop a reliable wrist sensor but also to determine the extent to which such devices can be used to successfully monitor what the gripper is doing. A typical example is whether wrist feedback alone is adequate as a feedback technique to allow pegs to be properly guided into holes.

2.2.3. A Tactile Imaging Array for Robotics

Here we propose to develop a tactile sensor for robotics. Such a device would provide feedback on the position of the workpiece in the gripper, its shape, and, perhaps, its texture. There is general agreement that such devices are needed but no reliable structures presently exist.

The tactile imager would consist of a matrix of points, spaced one to two millimeters apart, each capable of resolving the pressure/force on that area to a level of probably 6 bits (1 part in 64). The outputs would be read out serially (or as several (parallel) serial channels) much as in a visible imager. Hence the tactile imager is similar to a low resolution visible array with pressure as input.

The tactile imager has a number of requirements, including a resolution of at least six bits, cell size of less than $2 \times 2 \text{ mm}^2$, and insensitivity to temperature. Most of all it must be rugged and reliable. The outer covering over the hand must be replaceable without disturbing the sensor array, since this skin will wear and require frequent change in some applications. The development of such an application is challenging but within the range of present technology.

There are at least three approaches to developing a practical tactile imager. Approaches involving arrays of springs or other moving parts are not considered since they are unlikely to meet the reliability requirements of this application. Here, a sensing array is mounted (electrostatically via a hermetic glass-to-silicon seal) on a gall plate which in turn is mounted on gripper surface. Above the array is a plate containing holes which access each sensing cell. Above this plate is a deformable pad, which could either flat or contoured. The pad would be partially slit to uncouple the various cells. Above this pad and bonded to it would be an outer skin. The skin, and perhaps also the plate and pad, would be removable so that they could be replaced periodically to overcome expected wear. This would likely require an automatic recalibration of the array but should not present a problem in robotic applications.

The structure shown discussed is based on the detection of pressure with an array of capacitive pressure sensors and on-chip circuitry. The pressure change would occur as the pad deformed in response to pressure against the

workpiece. Thus this approach depends on the formation of an array of small pressure cells. The sensitivity of the structure would depend on the characteristics of the pad and the silicon diaphragm, both of which should be highly uniform. The structure could be very pressure sensitive (if desired) or could be designed for a lower sensitivity and higher dynamic range. This could be implemented in the sensing array, but might be better implemented through a choice of the pad material. Over-pressure is no problem since the glass support provides a strain relief when the plates touch. The capacitive approach promises a high performance array since the structure is known to possess a very high pressure sensitivity combined with very low temperature sensitivity. The formation of the on-chip circuitry permits a high-level bussed output but is technologically very advanced.

Alternative schemes for tactile imaging are also possible. Most approaches retain the pad and skin and rely on an alternative sensing array. If we allow the pad to fill the access chamber above the diaphragm and eliminate the front (lower) reference cavity so the diaphragm does not move, then the structure can be used to sense the resistance of the pad and the change in this resistance as it deforms. For this purpose, a carbon-impregnated pad would be used. The upper surface of the cavity would be metallized and access to on-chip amplification/multiplexing circuitry would be via a diffused feed-through to the front (bottom) side of the chip. In this approach the readout circuitry and overall structure is simpler and less challenging but the sensitivity is entirely a function of the pad. There are questions of pressure sensitivity, temperature sensitivity and stability which must be answered for this approach.

A second alternative structure would use the pad to transmit applied pressure to the cell, where it would be detected as the deflection of a piezoresistive cantilever structure. The cantilever would be formed as an alternative to the diaphragm. Again the glass plate would provide strain relief. This structure could be expected to offer intermediate pressure and temperature sensitivity and reduces the dependence on the pad characteristics. However, a suitable coupling arrangement between the pad and the cantilever beams must be developed.

In developing a useful tactile imager, we would propose to first investigate the resistive pad approach outlined above. This is potentially the simplest approach if its performance is acceptable. A small array would be tested over temperature, pressure, and time. Electronics would be mounted off chip. In parallel, paper designs for the cantilever and capacitive diaphragm approaches would be worked out along with processes and anticipated characteristics. The most promising of these would then also be implemented and characterized. Our final goal would be to assemble and characterize an array of tactile cells on 2 mm centers with an array size of at least 16x16 points. Such an arrangement should allow a meaningful evaluation of tactile imaging for robotics.

The solid-state sensor area is a vital part of robotics and industrial automation. While the area is relatively new and many sensor requirements remain to be defined, the needed solutions appear to be within the range of present technology.

2.2.4. Active Compliance Control for Sensor-Based Robots

The primary objectives of the force feedback control focus on designing an active compliance control strategy that (i) effectively utilizes the force sensory feedback information to control the arm in fine motion, and (ii)

incorporates pattern recognition and learning techniques so that the manipulator can improve its performance in fine motion based on past experience. In general, the force feedback control strategy utilizes the resolved force measurements (three orthogonal forces and three orthogonal torques with reference to the hand coordinate system) from the force sensor to determine error actuating signal for each joint actuator to control the arm in completing the task. This simple control strategy is quite limited in scope due to its inability to utilize past experience to improve the performance of the arm in fine motion. With this in mind, the proposed force control algorithm incorporates (i) pattern recognition techniques which have the ability of recognizing similar recurring control situations, and (ii) learning techniques in which past experience is used to improve the performance of the arm. Similar approach of using a combination of pattern recognition and modern control theory has been used successfully in other areas [Fu70, Fu71, Skl66].

The pattern recognition techniques embedded in the proposed control strategy require that the control space be partitioned into groups of "control situations" in which appropriate best control law will be used to control the arm. The fact that the manipulator performs differently during different portions of an assembly task (e.g. gross motion control and fine motion control are different and the control law for each control situation is different) strengthens the idea of partitioning the control space in fine motion control into groups of "control regions". The learning portion of the control strategy then improves the performance of the arm by updating/modifying the feedback gains of the appropriate control law in each of the control situation.

An initial approach to solve the force control problem is to model the manipulator in fine motion control as a self-organizing system with an on-line learning controller. The self-organizing manipulator system performs its assembly task in a partially known environment and evaluates its performance iteratively during successive intervals of time so that the performance of the manipulator in fine motion is improved. The approach calls for the design of a force control algorithm that extracts "features" from the measurement space, partitions the feature space into groups of features called control situations, and then learns the best control law for each control situation. In essence, the self-organizing manipulator system with an on-line learning controller considers the force feedback control problem as a multi-state (or multi-region) control problem. It controls the manipulator in each control situation with its corresponding best control law and learns the feedback gains in the control law so that the performance of the manipulator in fine motion control is improved. Hence, the self-organizing manipulator system will outperform a manipulator controlled by a conventional force control scheme due to the fact that it recognizes the similar recurring control situations and uses and improves the best previously obtained control law for this control situation.

Preliminary investigation of the proposed force control strategy [Lee80] indicates that the control algorithm is performed in four distinct phases of operation:

- (a) *Feature Extraction.* This operation chooses/extracts features from the arm's measurement space which consists of the resolved force measurement from the force sensor, and the joint positions and velocities from the electric motors. The objective of the feature extraction is to select an "optimal" number of features from this

- measurement space that best describe/represent the current status of the arm in the fine motion control phase.
- (b) *Mapping the Features into Control Regions.* This operation uses pattern classification techniques to identify/classify the features obtained in the previous operation. The objective is to identify the control situation (or control region) in which the arm is found (based on the extracted features) and retrieve the necessary feedback gains from the computer memory for the fine motion control in the new region.
 - (c) *Learning.* This operation involves adding machine intelligence to the manipulator for the fine motion control by implementing several hill-climbing techniques to update/modify the feedback gains of the on-line controller.
 - (d) *Control.* This operation involves converting the control effort from the numerical calculations based on the dynamic model of the arm to the specific power drive unit of the arm to control its fine motion, such as obtaining the armature voltage values to generate the necessary torques for the arm.

Besides these four operations, the manipulator must go through an initial period of training to establish the nominal feedback gains for each control situation. This assures the initial smooth control of the manipulator in fine motion, and the proposed force control algorithm will improve its performance by updating the feedback gains in appropriate control situations. This training can be done either by leading or by programming the manipulator through all the possible sequences of fine motion control.

The proposed active compliance control for fine motion raises some interesting and challenging research issues that have to be investigated before it can be implemented on our PUMA arm. These issues heavily reflect the performance of the manipulator system in fine motion:

- (1) Into how many control situations should one partition the control space?
- (2) How many features are needed from the measurement space to sufficiently or uniquely represent/describe each of the control region? Too many features will slow down the control process, while too few will result in an inadequate representation which may cause large recognition error and lead to incorrect classification.
- (3) What is the percentage of the recognition error of the control situations of the system using various pattern recognition methods and how can one reduce/minimize this recognition error?
- (4) How does the manipulator system behave if a wrong recognition occurs and how can one correct and "re-guide" the manipulator to complete the task?
- (5) What is the convergence rate of the hill-climbing technique used to update the feedback gains of the control law?
- (6) What is the stopping criterion for learning (i.e. further learning will not improve the performance of the system)?
- (7) Is the robot using force feedback control method stable under real-time computer control?

3. Vision for Robotics (Delp, Leith, and Frieder)

Because visual feedback is an integral component of the gross motion control of the robot arm, a fundamental goal of the vision research covered by this proposal is to develop a system for the detection and recognition of workpieces to be handled. However, our long term objective is to develop a vision system capable of (i) performing segmentation based on a set of features relevant to a wide range of applications (ii) supplying real-time sensory feedback information and (iii) functioning effectively in the noisy environments encountered in industrial applications. Therefore, besides providing a tool to support the other areas of research described in this proposal, we will be using the development of the vision system as a framework for investigating techniques which are much broader in their implications. We shall focus on three areas of research: (i) the development of shape descriptors based on range data, and the use of range information as a feature, (ii) optical processing to allow increase in processing speeds (iii) the identification of objects directly using a broad spectrum of feature information.

3.1. Background

Past work in computer vision area has resulted in several experimental systems for industrial robots as well as a few cost-effective commercial systems [SIR80, CAI79, Hol80, Agi77, Ler80]. The major problems in the design of these systems have been segmentation, object location, and object recognition.

A typical vision system consists of the following processing states:

- (1) Image acquisition of one or more views of the object area. Both visible light and infrared sensors are used. Range information is obtained via stereo views, ultrasonic sensors or laser sensors.
- (2) Image preprocessing to remove salt-and pepper noise, filtering to correct uneven lighting, some forms of enhancement, etc.
- (3) Feature extraction , typically segmentation operations. The features used include gray-level edges, texture edges, shape descriptors such as a Fourier shape descriptors, area of objects, perimeter, depth from stereo views, depth from structured lighting, depth from ultrasonic or laser ranging, shape-from-shading, and many others. The features extracted are highly application dependent.
- (4) Object identification, i.e. position/orientation. In the past most decision procedures were based on pattern recognition concepts using either statistical or synthetic pattern recognition.

These techniques have limitations with regard to performance and ability to handle a large number of features. There has been considerable interest on using artificial intelligence concepts to interpret the features. This is the problem of "image analysis" that DARPA has been addressing [DA80]. We believe that AI concepts will have to be used to address this very complicated problem. The use of "expert systems" is now being very much considered for the general scene interpretation problem.

An important issue in computer vision for industrial robots is that of computation speed. It has been estimated [ReH79] that processor speeds on the order of 1 to 100 billion operations per second will be required to solve some of the current problems in computer vision. While the current trend towards "massively parallel" architectures for vision affords a solution (see next section), it raises the issue as to what algorithms can be implemented on

such architectures. Algorithms such as relaxation labeling techniques, correlation matching and histogramming, and edge thinning are highly parallel operations and have been implemented on various architectures. Our Proposed research in the vision area will have a component directed towards the simultaneous development of vision algorithms and architectures.

Because of these high processing requirements we are proposing the use of optical processing to perform some of the vision operations particularly those in states 1 and 2 above. This should reduce the number of computations that must be performed digitally which in turn will allow a speed-up of the recognition operations.

3.2. Real Time Recognition of Surfaces and Shapes

The surface recognition problem comes in a number of varieties, which depend on the goals of the recognition. We shall deal with four of these, all of which will be addressed, and solved, by the system which is described in this section of the proposal.

The first possible statement of the problem is: Given a three-dimensional scene, find out the surface (map) of the scene as seen from a given direction in a precise, numerically accessible format. In this variety of surface recognition, the emphasis is on the acquisition and not on the recognition problem. The goal is, indeed, to produce a series of triplets (x,y,z) describing the height z at any point x,y within the scene. As we are dealing in a finite scene and a reasonably manageable apparatus, we shall restate the problem to be: Given a surface, find the height of the surface in arbitrarily many but limited number of points, as seen from a given direction.

Note again that in this statement the surface is not recognized, nor is there any need for it in the statement problem. This is so because in this approach the important goal is to produce a surface measurement only. The nature of the surface is immaterial. In this formulation of the surface acquisition task, it can be applied to any measurement, from the problem of mapping the palate of a child through the question of precise measurement of a turbine blade, to the determination of a relative spatial position of a body in space. For future reference we refer to this formulation as the limited measurement problem.

Second formulation is similar to the first but the clause "given direction" is omitted. Thus, we are not interested in a surface as seen from one direction. We are interested in a complete map of the surface, including information about parts which may be *hidden* when viewed from a given direction.

This is an important generalization of the previous problem and will be referred to as the complete measurement problem. Its uses vary from those described for the limited measurement problem to the matching problem, i.e., comparison of a measured surface to another set of sample surfaces. The comparison problem is of utmost importance in numerous areas, ranging from automatic manufacturing to guidance problems.

Third formulation is: Given a known three-dimensional object, find its surface and its position in space relative to a given, predefined position. In this formulation, one assumes a given predefined map of the surface, i.e., a given set of triples (x, y, z) . The stipulation is that the scene currently under view contains the same objects albeit in possibly a rotated or displaced way. The surface recognition goal is to find out the amount of rotation or displacement.

The solution of this formulation, referred to as object position recognition or surface identification, is used for positioning of automated handlers and automatic assembly, as a first step in positioning an object for automated quality control and precise size measurements for selection of one among many, etc.

The last formulation to be discussed is: Given a three-dimensional scene containing a known object in a known position, find deviations from the recorded shape of the object. This is very similar to the restricted measurement problem, the difference being the prerecorded knowledge about the object. The use of this formulation is paramount in quality control, casting and molding processes and in numerically controlled machining. There is no need to discuss this formulation further, as it is clear that its solution is straightforward once the previously stated formulations can be solved. Indeed, we introduce this formulation as an example as to how the combination of the previous formulations imply other formulations which can be solved automatically once we achieve the solution to the basic problem.

Surface Recognition - Outline of Solution

We propose a solution to the surface acquisition and recognition problem which avoids the need to perform any preliminary pattern recognition or any other object oriented manipulations. Thus, our method deals directly in x,y,z coordinates of the surface, without recourse to the nature of the surface. When dealing with the measurement problem, either in its limited or general form, the nature of the surface is obviously irrelevant - all we want is the precise measurement. In other formulations, where the surface is given, it is presented as a set of coordinates, again without any need to know what the surface actually represents. The proposed solution, therefore, avoids the problems of various vision systems in which the nature of the scene, referred to as "the world" has to be known.

The proposed solution was originally conceived by B.C. Altschuler [AAT79]. Further mathematical developments were done by M. D. Altschuler and J. L. Posdamer [PoA], and recently some systems based on the methodology were proposed by Altschuler, Altschuler, Frieder, Posdamer and Toboada [APF80].

The basic idea behind the surface mapping recognition, measurement and replication method is in the realization that a set of coordinates of the surface suffice to describe the surface in full, provided that the coordinates are given in adequate number of points on a predefined but otherwise arbitrary grid, and provided that the coordinate values are given in adequate precision. The meaning of "adequate", i.e., the degree of precision required, will vary from application to application. The method, therefore, provides for an arbitrary degree of precision, meaning that the precision can be adjusted up to the physical limits of the apparatus being used. There is no inherent limitation in the method itself.

The coordinates of the surface are computed from a digitized image of the object, which is illuminated by a series of dot patterns, created by a laser based electro-optic system. Each point x,y,z on the surface of the object, which is illuminated by the electro-optic system, will create an illuminated image $x'y'$ in the digitized, two-dimensional picture. It is impossible, from a single such image and the knowledge of the values $x'y'$ to deduct the values of x,y , and z . However, if the same object is illuminated by a properly designed sequence of illumination patterns, then from all patterns together one can deduct the values of x,y,z and thus map the surface. Hence we are able to obtain the depth information z by using structured lighting. The

mathematical details are presented in [AAT79]. Here we only present the basic ideas and procedures of the solution.

The illumination patterns are designed so as to enable independent determination of the origin of an image point $x'y'$ in an $n \times n$ square grid pattern which was projected onto the digitizing camera. To this end, we provide $1 + \log n$ different patterns. The patterns are based on a binary search procedure of the columns of the illumination grid. A point $x'y'$ in the scene will be present in some patterns if the actual surface point x,y,z is visible from the illumination direction, will not appear if x,y,z is not visible (i.e., covered by other features of the surface). However, it is important to establish two points. First, the surface recognition procedure, in whatever formulation, starts by creating an illumination pattern of $n \times n$ points, and the second is that for an $n \times n$ grid there are $k = 1 + \log n$ patterns.

The acquisition of these patterns has two parts; these are the creation of the pattern and the digitization of it.

The creation is done by a laser-optic method, utilizing a high-speed electronic shutter. A model experimental shutter was developed via a contract with USAF-SAM, and was used in some sample computations that showed the feasibility of this approach, but also the need for some basic development. The acquisition is done by digitizing cameras, which typically create a digitized picture which is between 256×256 and 4096×4096 pixels.

Once the digitized picture is acquired, the position of the illuminated dots has to be determined, and the dots stored. This process is then repeated for all k patterns. The (x',y',k) data elements are used to determine the x,y,z positions of the surface. This is a relatively simple computational task which is outlined in [PaA].

This is now the point of departure, from which different procedures are required for different problem formulations. All are, however, based on the same basic computational procedure which is:

1. Creation of an $n \times n$ illumination grid;
2. Selection of one of $k = 1 + \log n$ patterns;
3. Object illumination;
4. Scene digitization;
5. Isolation of the illumination points (computation of the illuminated dot position $x'y'$);
6. Repeat parts 2-5 k times, once for each pattern;
7. From the points $(x'y'k)$ compute the surface position x,y,z ;
8. Further processing according to problem formulation

Surface Recognition - Utilization and Areas of Importance

It is our intention to use this area of endeavor in the following industrial applications:

1. Detection of structural deformation.
2. Assembly line support-detection of unusual conditions, faults, positioning and timing problems.
3. Precision casting and molding.
4. Replication Technology.

All the applications listed above will benefit, indeed be improved, by the ability to quickly ("in real time") map three dimensional surfaces. For the first year, we will start two basic phases which form the basis for any future utilization of the real time surface acquisition.

Phase 1: Basic Support: Software and Hardware

The first phase involves the generation of all basic software support. This includes subroutines for geometric phantom generation, simulation of laser grid projections on phantom data, noise simulation, filtering and noise elimination, image enhancement, fuzzy points support, self calibration support, laser-grid and object database management support and complete graphics system. This part also involves a complete integration of all currently available graphic and other software packages into a multi-user integrated system.

A second part in this phase is a thorough study of available laser-shutter systems for possible acquisition and use in the developments of following projects. The study of computer controlled movable camera-laser systems will be continued in all forthcoming projects and the design and acquisition for a geometry system for a movable camera-laser will be developed for a movable object configuration.

Phase 2: Real Time Geometry Acquisition

This phase encompasses the implementation of our solutions to all basic formulations of the surface measurement problems, using real time acquisition computation equipment and the basic software system support that is developed in Phase 1. In this phase we shall address problems such as real time digitization, shutter control, real time filtering, surface acquisition for multiple laser-camera systems, surface display techniques, etc. We will use various approaches to this phase including database management and artificial intelligence techniques.

3.3. Three Dimensional Shape Analysis

The previous section we discussed a structured lighting approach to obtaining range information, z , in a given scene. Once the depth information is obtained we will need various approaches to describe and hence classify objects or shapes. The approach used in currently available binary vision processors is to measure such features as major and minor axis, areas, perimeter, etc. While these shape features are adequate for some applications they will not work with very complicated shapes. Very often it is impossible to use these shape features due to inadequate thresholding or occluded (or hidden) parts. The shape descriptors we will investigate are based on gray-level images and not binary images.

A very powerful shape descriptor that has been in use for many years is the Fourier shape descriptor (FSD) [ZaR72]. This descriptor is based on tracing the contour of the object on the complex plane and then taking the Fourier transform of the coordinate points. The resultant "frequency" domain contains a complete description of the shape such as area, perimeter, center of gravity and various other properties. These types of descriptors are "complete" in the sense that knowing the FSD one can obtain the object shape exactly. Other shape descriptor approaches include model-based descriptors such as Perkins approach based on modeling the shapes as line segments [Per78].

Shape analysis for three dimensional object identification will be investigated. Wallace [WMF79] has shown that a local shape descriptor based on Fourier shape analysis can be used effectively even when only partial segmentation of the object is possible. This method is very fast and in some cases linear interpolation can be used in the feature space for object

classification. We propose to further investigate the shape analysis problem with particular emphasis on parallel algorithms that use range data. One question that immediately arises is how the range data, once it is obtained, can be used in our shape descriptor. We propose to address this problem in two phases:

- (1) the investigation of three-dimensional shape descriptors which extend the classical Fourier descriptors. This is the "partial" descriptor that is obtained when parts are occluded or have very complicated shapes.
- (2) the use of the range data to "compete" the shape description

The problem of describing very complicated shapes which often are associated with parts used in an assembly task is still a non-trivial problem. We propose to use the range for shape analysis. This information will undoubtedly have to be used in the database of parts and will have impact on the geometric modeling for database structures discussed in a later section of this proposal. We propose to approach the shape by extending Perkins [WA78] work on model-based shapes and adding extensions to Wallace's partial shape descriptors. These extensions include such things as mapping various partial shape descriptors together to obtain a more complete shape. By using the range data we hope to take into account regions where only a partial segmentation was obtained. We shall assume adequate range data has been obtained either by the methods of the previous section or other methods such as stereo views.

This approach will lead to the use of artificial intelligence techniques for our shape descriptors. This will be a knowledge-based approach using apriori information, i.e. in an assembly task there can only be a finite number of parts, we would not expect a headlight part to appear as a possible part for engine assembly. By incorporating this information into our shape descriptors we will be able to eliminate ambiguous shapes [Bat81].

3.4. Application of Optical Processing to Robotic Vision and Intelligence

We propose to use optical processing techniques to help reduce some of the computational overhead required to perform the vision tasks. We see several potentially fruitful ways to apply optical processing to the robotic vision and intelligence problem:

- (1) The use of incoherent light for the optical processing.
- (2) The use of hybrid optical-digital processors
- (3) The use of more sophisticated recognition and classification algorithms than the matched filter.

3.4.1. Incoherent Light

There are two problems with the use of coherent light in optical processing for robotics: the need for an incoherent to coherent transducer and the inherent noisiness of coherent light.

We illustrate the first problem with an example. In a simple scenario, the robot performs the relatively easy task of looking for parts coming down a production line. There may be only one part, of a standard size, and the robot must recognize the orientation and location of the part. A further complication may be in the presence of other objects, which are not of interest and must be discriminated against. This simple situation can be handled by a matched filter; no more sophisticated optical processing element is needed. Gara [Gar76] has already described this situation. A very major drawback to the optical implementation is that the image formed by

light reflected from the object cannot be used in the matched filtering system; coherent light is required. Therefore, an incoherent to coherent transducer, such as a liquid crystal display device, is needed. The purely optical approach described by Gara worked perfectly well; however, the additional complication of a real time incoherent to coherent transducer made the optical method non-competitive with the purely digital method.

The second problem is that coherent light is inherently noisy, a fact long known to optical researchers and often the chief reason for the failure of potential applications of coherent optical processing. Incoherent light is far superior in terms of SNR. We suggest that, because of the noise problem, optical processors that use coherent light cannot be competitive with all-digital methods except in some special, relatively simple situations, such as the example cited above, and even there, the optical method failed for other reasons.

The greatest possible contribution to optical processing for robot vision and intelligence, in our opinion, would be the use of ordinary spatially incoherent white light for the optical processing. Two major improvements could result. First, the processing could be done directly with the light reflected from the object and the awkward, expensive, and rather limiting incoherent to coherent transducer avoided. Second, a large improvement in SNR could result.

We and others have used incoherent methods for optical processing. For example, we have developed a chromatic optical processors that use white light, but still behave like coherent systems and thus offer the great flexibility of the coherent systems [LeS81, Yu78, MoG80]. The experimental results show enormously better SNR than for purely coherent light. We and others have had highly successful results using spatially incoherent light in several applications [YaL81, Loh77, Loh78].

In view of these two major advantages for incoherent light, a significant part of our proposed effort would be to do the optical processing with light of reduced coherence, either with spatially coherent white light, with monochromatic, spatially incoherent light, or with completely incoherent light. With the former we can achieve better SNR, with the latter two we achieve this and also avoid the incoherent to coherent transducer problem.

3.4.2. Hybrid Systems

A second promising area for optical processing is the hybrid optical processor, in which optical processing and digital processing are combined into a single system. We have had previous experience with such hybrid processors in connection with synthetic aperture radar data processing, and the experience can be readily extended into the robotic vision and intelligence area.

3.4.3. More Sophisticated Filters

A third area for optical processing is the use of complex filters other than the conventional matched filter. Following our development of the spatial matched filter in 1961, we made a number of variations. For example, to reduce the extreme sensitivity of the filter to the exact form of the object and to slight size and orientation variations, we produced desensitized filters, in which the higher spatial frequencies were demphasized. Also, average filters were developed, in which the object from which the filter is made is a composite, or average, of a number of objects that together constitute the class of objects we desire to recognize. Also constructed were idealized

objects, which contained the essence of the objects to be detected, that is, contained aspects that were common to all members of the class to be detected.

In recent years, more sophisticated filtering methods have been developed, in which features are extracted from images, allowing the images to be placed in one of several classes. For example, Lee [Lee81] has described the Least-Square Linear Mapping Technique and the Fukunaga-Koontz transform for accomplishing this.

Certainly, the possibilities for pattern recognition by spatial filtering are considerable, limited only by the requirement that the operation be linear, since non-linear operations are not easily done optically.

3.4.4. Proposed Program

Our proposed program is therefore:

- (1) To apply the previous results of optical processing with incoherent light to the problem of robotic vision and intelligence.
- (2) To develop architectures for the combined use of optical and digital processing in robotic systems.
- (3) To develop for robotic systems complex spatial filtering techniques that are more general than the conventional matched filtering methods, and that can overcome the basic limitations of the matched filter.

4. High-Performance, Special Purpose Computer Structures

(D. E. Atkins, G. Frieder, T. N. Mudge)

4.1. Introduction

This long standing research project addresses the need for significant increases in computer processing power for the solution of problems where the quality of the answer is proportional to the amount of computations that can be performed. It is aimed at discovering theory and methodology for the automatic design, synthesis and analysis of high-performance digital computers which are optimized for a given application, i.e., for application-directed computer design. Components of this broad-based project include

- (1) development and formal specification of arithmetic-intensive algorithms;
- (2) computer-aided synthesis of concurrently executing systems of processors to meet specific functional and performance specifications; and
- (3) the design, fabrication, and characterization of advanced integrated circuit components to serve as computational building blocks in highly parallel systems.

Data rates for computer vision are estimated to range up to 108 bits/sec and require 109 operations/sec for processing. This computation together with speech and high-speed arm control will be achieved in a cost-effective manner only by the synergistic design of hardware and algorithms. A methodology for mapping algorithms into customized, VLSI circuits is an area of present activity in the College but requires much more emphasis and support in order to meet the computational demands of advanced robot systems.

The Application-Directed Computer Architecture (ADCA) group at the University of Michigan has for the past seven years been working on projects

aimed at the synergistic design of machines and algorithms rather than the fitting of algorithms to available machines. This is the major theme of our research and in this spirit we have participated in the design and implementation of numerous application-directed parallel machines many of which are in the area of image processing. In order to support the algorithms for arm control and vision so that the former can be performed in real-time and the latter in relevant-time, we are proposing to research and develop attached processors with appropriate application-directed architectures.

4.2. VLSI Processor for Robot Arm Control

This work has developed a preliminary specification for a VLSI implementation of a single chip processor for dedicated arithmetic-intensive applications [TuM81]. Circuit densities commensurate with levels of integration projected for the mid-1980s are assumed. The proposed processor, termed the Numerical Processor (NP), is suitable for real-time control where sophisticated control strategies require very large numbers of high precision arithmetical operations to be performed for every input/output transaction. In particular, the NP is intended for the real-time control of a robot arm. The NP functions as an attached processor of a general purpose minicomputer. Conceptually, it lies between Floating Point Systems' AP120B [Flo79], a high performance numerically oriented attached processor, and the Intel 8087 [Pal80], a single chip numerically oriented attached processor in the Intel 8087 family of components [Int79]. All three work with floating-point numbers. The NP differs from the AP120B by being much simpler, less flexible, slower, and by having a smaller word size (32 bits versus 38 bits). It differs from the 8086 by having its own on chip program memory, input/output buffers to facilitate real-time applications, and two independent function units. However, the 8087 has a more flexible number format, and can deal with several variants of the IEEE floating point standard up to and including the 80 bit format.

The preliminary study reported in [TuM81] assumed the NP will be implemented in nMOS because of our present expertise is in this technology (see section 4.2.5). However, our eventual aim is to investigate the design of the NP in a faster technology that still has the density of integration associated with nMOS. A prime candidate is the I^3L (Isoplanar Integrated Injection Logic) technology developed by Fairchild, however, the n-well CMOS process presently under consideration by us (see 4.2.5 again) is also a possibility.

As mentioned earlier, the design philosophy of the NP is oriented towards a dedicated arithmetic-intensive application, in particular, the control of a robot arm where the arm response is limited by the complexity of the control computations. The specific robot arm in mind is the Unimation PUMA 600. It is a six link arm having all revolute joints.

4.3. VLSI Processor for Robot Vision

Robot vision is an area in which a considerable amount of research has been accomplished. In spite of this the capabilities of present systems are quite primitive. One of our major interests is to investigate the possibility of applying VLSI technology, in the form of special-purpose computers, to improve the capability of present vision systems.

Past work in robot vision has resulted in numerous experimental systems for industrial robots as well as several cost effective commercial systems [SIR80, CAI79, Hol80, Agi79]. The major problems with which these systems

have had to deal are segmentation, object location, and object recognition. Typical approaches to segmentation have been based on thresholding [Ohl75, Bai79], region growing [PNK79], outlines derived from edge maps [Per79], and contour information derived from laser range finding and planar lighting [Hol80, Ris77]. Laser range finding and other structured lighting techniques have also been used for object detection and location as applied to visual servoing [Van79]. Features such as line and curve lengths, connectivity, areas, and moments of inertia, as well as template matching have also been successfully used in object identification, even in the case where sections of the objects have been obscured [Per78]. Many of these techniques have become standardized and are beginning to appear in stand alone systems such as the vision module developed by the Stanford Research Institute [Nit79]. It has been noted however [TBB79], that current approaches do not generalize well for the solution of some of the more difficult problems facing industrial vision systems, such as those encountered in batch assembly tasks and bin picking. In particular, what is needed is a more general and powerful set of primitive features and a more effective approach for segmentation in noisy environments.

Segmentation and object location based on stereo vision¹ offers several advantages over other approaches. Most notably, because surface boundaries are detected on the basis of range data and not on local changes in illumination and surface characteristics, edges caused by shadows, uneven lighting, texture, and other ancillary surface features do not present a serious problem. In fact, certain algorithms which are used in three dimensional reconstruction depend on the existence of random features for their operation [Mar76, Ols80]. Also, as has been noted by Tenenbaum et al. [TBB79], even relatively simple segmentation techniques work well when used in conjunction with range data. Along with "surfaces", another powerful and general feature which can be derived from the use of stereo vision is that of "objects" per se. In this case an object is recognized on the basis of its disparity relative to a ground plane. Such an approach has been used in the design of a vision system for an autonomous vehicle [Gen79], and we are currently pursuing it in the context of industrial vision for robotics.

A further important issue in industrial vision is that of speed. It has been estimated [ReH79] that processor bandwidths on the order of 1 to 100 billion operations per second will be required to solve some of the current problems in robot vision. While the current trend towards "massively parallel" architectures [Bat80] for vision affords a solution it raises the issue as to what algorithms can be implemented on such architectures (see also [Rev81] for a discussion of this question). Our research is aimed at providing an answer to this question and further providing a candidate special-purpose architecture capable of meeting the computing requirements of industrial robot vision.

Vision is part of our robotics research and involves the use of multicamera stereovision to supply the relevant-time feedback for gross motion control of the robot arm. As part of that program we are presently investigating three dimensional reconstruction algorithms for use in depth ranging, as well as various types of shape descriptors for use in segmentation. In the specific area of special-purpose processors for vision our initial research has been to catalog those algorithms that a consensus of researchers considers useful, and then to consider various architectures appropriate for those algorithms. This initial work will appear in [MuDB2]. Our proposed

¹The term stereo vision is loosely applied in the literature to any technique which extracts range data or 3-D information from a scene, not with necessarily the use of two cameras.

research will proceed as follows:

- (1) Investigate the effectiveness of the Cytocomputer [LMS80, LoM80, Ste80] in vision applications. Our work with the Cytocomputer to date indicates that it performs well at "low level" operations such as edge detection and thinning. As one might expect these operations are ones that rely only on local or nearest neighbor information from the 2D image field. However, it does not appear suitable for operations that require "global" information. For example, forming Fourier shape descriptors of contours, or reconstructing 3D information.
- (2) Investigate multiprocessor architectures suitable for operations that require global information. Consideration will be given to incorporating these architectures into a system having a Cytocomputer as front-end. Alternatively, consideration will be given to integrating features of the Cytocomputer into these architectures.

5. The Design and Analysis of a VLSI Crossbar Switch

The common denominator among the special-purpose computers that we are researching are:

- (1) A computation environment in which the majority of the operations are arithmetic.
- (2) The need for multiple processors, or at least multiple ALU's, to support the high operation rates.
- (3) A high bandwidth memory to supply operands rapidly enough to keep the processors busy.

The research to develop improved VLSI arithmetic units is described elsewhere in this proposal (section 4.2.2). To support the tightly coupled multiprocessor environment implied by points 2 and 3 we are investigating the design and analysis of the VLSI implementation of a crossbar switch [MaM81a, MaM81b]. This work is being done in collaboration with the Fairchild Camera and Instrument Corporation. Our eventual aim is to construct a VLSI implementation of a crossbar.

The environment that we envision for a crossbar is depicted in Figure 5.2. N processors (Proc. in the figure) with their private memory (PM in the figure) are connected through a crossbar (ICN--interconnection network in the figure) to M data memories (DM in the figure). The crossbar allows any one of the processors to establish a connection with any one of the memories. In the case of conflict, i.e., when two or more processors request the same output, only one of the requests is granted. In our preliminary design [MaM81a] a simple daisy chain priority is used to resolve the conflict. In our research we have developed bandwidth measures for crossbar switches [MaM81b]. Our preliminary findings indicate that requests have about a 65% chance of getting through the crossbar on their first try regardless of the size of N and M, and even if each processor makes a request every system cycle (an unrealistically high request rate). When the request rate drops the chance of getting through increases dramatically. In addition, in the case where an input has a favorite output (a frequent occurrence) we have an even greater chance of getting through, or in other words the bandwidth increases.

Past proposals for tightly coupling multiple processors to multiple memories have steered away from crossbar switches because of their component complexity which grows as $O(N^2)$ assuming $N = M$. A whole range of ingenious networks have been proposed ranging from the early ideas of Clos and Benes [Clo53, Ben65] through the Omega network of Lawrie [Law75] and the

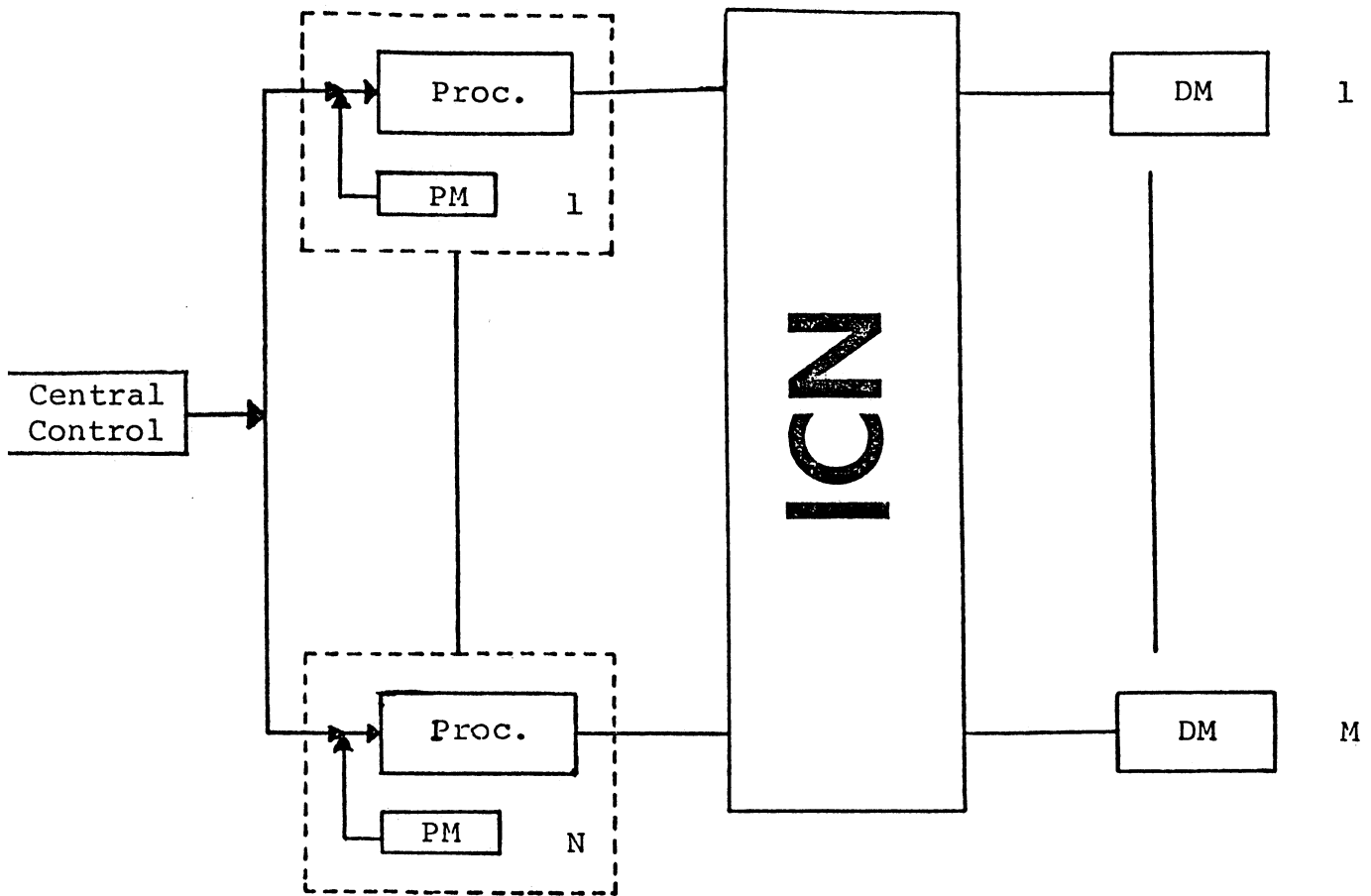


Figure 5.2. System with Crossbar.

Shuffle Exchange network of Stone [Sto71] which have many of the connectivity properties of a crossbar without the component complexity. For a good survey of the state-of-the-art in networks see Siegel [Sie80]. These networks all grow in complexity as $N \log_2 N$ rather than N^2 . With the advent of VLSI technology it is no longer clear that the reduced component complexity is an advantage. For example, it doesn't appear to translate into more efficient space utilization in an IC layout. Furthermore, although the reduced complexity networks preserve some of the connectivity properties of the crossbar they do not preserve bandwidth. For these and other reasons we have decided to explore crossbar switches more fully.

Our research will proceed to:

- (1) Continue with our research to develop more complete models of the behavior of crossbar networks. In particular, we are in the process of developing a probabilistic analysis that models the effect of request queues.
- (2) Investigate the design of a crossbar in our n-well CMOS process (section 4.2.5) as well as high speed technologies such as I^3L . The present design [MaM81a] assumes our nMOS process which results in network delays estimated at about 100 ns for $N = 16$. We hope to get a better idea of how delay scales with N .

6. Improved Robot Programming (Naylor, Volz and Woo)

The ingenious idea that led to modern programmable industrial robots was programming by teaching. In early applications it was easy to do, and almost anyone could do it. Computer programmers were not required. Now, however, there is growing interest in finding other ways to program robots. These ways are usually grouped under the heading of "off-line" programming. There are a number of reasons that this has happened. First, for complex operations such as assembly, and in particular operations such as insertion or screwing motions, programming by teaching can be tedious and difficult. Moreover, programming by teaching does not provide mechanisms for making adjustments based upon sensor inputs. Also, robot applications with complicated branching and exception logic are very difficult to program using only teach mode.

Recent approaches to off-line programming have been based upon the development of procedural languages for robot programming. One of the early languages, WAVE [Pau77], provided the programmer with the ability to specify point to point robot motions. A related language, AL [Fin74], provided many advanced features such as coordinated motions between two robots, conditional motions based upon sensors such as force, smoothed trajectory calculations, and compile time planning values, the latter being used for AI-like high level task specification. There are several other experimental procedural robot programming languages, such as TEACH [Ruof79], and PAL [Pau79]. However, there is as yet only one truly high level robot programming language commercially available, VAL from Unimation [Shi79]. VAL embodies the simplest essential features from WAVE and AL. The MCL language [MacD80] currently under development by McDonnell Douglas for the ICAM [ICAM78] project attempts to manage all of the devices in an advanced manufacturing cell, including both machine tools and robots, in a single language by extending APT to include robot operations.

Up to a point, programming in one of these languages is similar to programming in an ordinary procedural language such as FORTRAN or PASCAL. However, there is the obvious difference that here one is always programming with respect to an evolving geometric situation, that is, the robot and its environment. Also, some of these languages allow concurrent programming and other advanced constructs mentioned above.

In theory, these languages solve many of the problems perceived in programming by teaching. However, some problems persist, and new ones arise. These include:

- (1) Obstacle avoidance and trajectory calculation. The robot must be aware of and avoid all objects in its workspace. This becomes particularly difficult in multiple arm robot systems.
- (2) Complexity. As indicated by Ruoff [Ruof80] it is not difficult to find robot applications that require very large programming efforts. In other words, while programming languages can handle more complex problems than teaching, they also require a higher level of skill on the part of the programmer.
- (3) Multiplicity of devices. With the addition of sensors, such as force, touch and vision, and particularly multiple arms, the device communication and control operations become quite complicated.
- (4) Acceptance by operating factory personnel. Introduction of even a relatively simple language such as VAL into the factory is not a welcome step. Most manufacturers would prefer to avoid it, desiring something

directly usable by factory personnel.

To address these problems we introduce three coordinated ideas which hold promise for simplifying and improving robot programming systems:

- (1) Coupling computer aided design (CAD) geometric databases to robot programming languages.
- (2) Use of computer graphics in robot programming.
- (3) Examination of a distributed operating system to manage the multiplicity of devices which must work together in an advanced robot-based manufacturing cell.

As evidenced by systems newly offered or under development by vendors such as Applicon, Calma, Computer Vision and MDSI, the use of computer graphics based CAD systems is going to play an important role in future product design. It is widely believed that the use of graphics afforded by these systems will be the key to their success. We believe that it will be possible to extend this simplifying notion of programming by graphics to robot programming as well. Information contained in the geometrical modelling databases which are part of these systems has the potential for either simplifying procedural robot languages or forming the base for a graphical programming language. For example, once the location of an object is known, say by teaching or from a vision system, relative locations of holes, protuberances, etc., may be determined from the database. In fact, the relative location of points of interest can be determined beforehand at compile time, thus reducing the amount of run time computation needed. If coupled with a labeling of appropriate points in the database so that points may be referenced by name, this may simplify procedural programming or form the base of graphic programming. In other applications, obstacle avoidance algorithms may reference data in the geometric modelling database.

To realize the benefits of the approaches mentioned above, research must be carried out in several directions. In addition to an expected improvement in robot language capabilities, the study of a coupling between CAD geometric modelling databases and robot programming may provide useful feedback on possible improvements to the CAD system themselves. The following sections discuss in greater detail specific tasks to be pursued.

6.1. Facilities Available

The facilities available for the entire scope of this proposal are described in some detail in Section 9. However, it is useful to outline here the facilities which provide the context in which the robot language work would be conducted.

The Robotics Laboratory has a Unimation PUMA 600 series robot arm. This has been coupled to an LSI 11/23 in a manner such that VAL commands can be sent to the PUMA from the 11/23. Thus, one has the extended general programming capabilities of PASCAL, which runs on the LSI 11/23, and the robot commands of VAL available for use. PASCAL programs which execute on the 11/23 can thus provide interfaces to advanced sensors or databases, passing VAL robot commands down to the robot as needed.

Communication is also being developed between the LSI 11/23 and the VAS 11/780 and is expected to be complete before this project would start. A rudimentary vision system utilizing the Hamamatsu digitizing camera has already been developed. Thus, it will be possible to incorporate two

dimensional binary vision systems into robot programming languages. Furthermore, a recently delivered GE TN2500 camera will be interfaced directly to the LSI 11/23 and the vision software installed on it within the timeframe of the proposed research, thus making a more direct use of visual feedback possible.

The Ramtek image display system includes a joystick cursor positioning device which may be used in the graphical robot programming work. Also, there are several other graphics terminals (Tektronix 4010's and 121's) with graphics tablets available within the University.

6.2. Computational Geometrics

In a typical robot manipulation environment, geometric computations are involved in sensory data processing, trajectory calculation and collision avoidance. Potentially, many of these may be based upon information in a geometric design database describing the parts to be manipulated. This task is directed toward determining the basic geometric support needed to effectively utilize a geometric modelling database. The solutions may, in actuality, partially involve the database aspects (storage and retrieved schemes) of the system (see next task).

We assume that if a robot is to perform some sequence of operations on a set of parts, $\{P_1, \dots, P_n\}$, there is information in the geometrical database describing the geometry of each P_i in the set. We also assume that with each P_i there is a reference point and orientation, R_i stored in the database. Finally, we assume that the position and orientation of each P_i in the world coordinates of the robot, i.e., a homogeneous transformation to the reference point R_i , is determinable (for example by teaching or visual recognition). The goal is to determine a variety of useful geometric information about the parts which will be useful in robot programming.

First, operations on the database useful for robot programming will be classified. The simplest operations will be transformations from a point on an object, e.g. a pickup point, to the object's reference point. More complex will be feature related information extraction such as the axis of a hole, e.g. for bolt insertion, or the axes of an elliptically shaped opening. Also complex will be the determination of features such as perimeter, area, or centroid (when the object is viewed from a given perspective) which could be used to avoid the training phase in a visual recognition system.

Another important class of geometric computations can be classified as the set membership problem, namely,

M: "Is a set X in another set Y?"

For processing sensory data, the set X may be points acquired from some sensor and the set Y may be the contour of an object. In trajectory calculations, the set X may be a set of points to be smoothed by a spline function Y. For collision avoidance, the set X may be the trajectory in the form of a curve and Y may be obstacles.

The set membership problem can be posed as a a geometric query or as a topological query. In its geometric form, the query may be asked as:

G: "Does A curve C intersect a surface S? If so, compute the point of intersection :".

In its topological form, the query may be asked as:

T: "Does a curve C bound a surface S? If so, retrieve an end point p."

After identifying the operations needed, an investigation into techniques for realizing the requirements determined will be carried out. Point operations should be straightforward. Feature related operations may be much more complex. An important issue will be the suitability of the information typically stored in geometric databases, or possible changes which would make the needed computations more readily calculable.

We plan to base our investigations on an explicit modelling language rather than an implicit approach as this is the direction being taken by the major commercial CAD systems. Four geometric modelling systems for object description are available at the University of Michigan. They are BUILD (developed by Braid at the University of Cambridge), SYNTHAVISION (a commercial system marketed by MAGI), 3DFORM (developed by Woo at Michigan), and ARCHMODEL (developed by Borkin at Michigan). We expect to use the first two systems more extensively because they include a capability in handling analytic surfaces.

6.3. Geometric Database Structures

The world model of an intelligent robot consists of (i) axiomatic knowledge of the work space as geometric data and (ii) procedural knowledge of what information to access, when to access them, and what to do with them. The way in which the data is organized impacts the efficiency of the procedural queries. The interplay between storage structure and query structure becomes especially critical if the robot is to perform in a real time environment. (Elsewhere in the proposal, effectiveness via higher speed computation hardware is addressed in dependability of the operating hardware.)

It may be noted that G and T of the previous section illustrate a dichotomy between allocation in storage and time. If everything is stored, then there is no need to compute. In other words, there will only be topological queries that thread through data structures for retrieving stored answers. The opposite may be said if no data is stored. In reality, a delicate balance must be obtained between geometric and topological queries, hence storage and time.

Recent research by Woo at the University of Michigan has revealed that there are nine "bedrock" queries for geometric data abases. They serve as the interface between the data base and the world outside (consisting of users and programs). It is also discovered that there are a total of 502 or (k) possible schemata, $k = 2, 3, \dots, 9$, for structuring data to answer the queries. Some of the schemata are storage expensive. But they offer fast retrieval and updating time. Others behave in an opposite way. What is particularly revealing in the finding is that a certain schema requires only twice the storage of another schema but provides an order of magnitude in savings in retrieval time. This task will explore the interface between a programming system and geometric databases. Of particular interest will be answers to the following questions:

- (1) What is the significance of the different schemata for a geometric database? Of the 502 schemata, we have explored only a few. We anticipate to reduce the complexity by finding classes of equivalent schemata.

- (2) What are the functional requirements of the geometric database in a robot environment? On the 9 bedrock queries, does sensory data processing create a different demand than path calculation or collision avoidance does?
- (3) What is the optimal schema for a geometric database in a robot environment? This question is expected to be partially answered by results from the previous two steps in the investigation.

6.4. Graphic Based Robot Programming

The principal work in this task will be to begin development of an interactive graphics system for programming robots. The first phase will be to develop a system capable of programming current robots such as a Unimation's PUMA, Cincinnati Milacron's T3, ASEA's 6 or the Trallfa for simple motions with simple objects. Since the Robotics Laboratory already has a PUMA, this will be the first target robot.

The principal purpose of the first phase of the activity will be to demonstrate the feasibility and simplicity of robot programming via relatively simple graphics. Accordingly, the initial system will display only very simple geometry, and will not incorporate a three dimensional solid modeler. It will be similar to current CAD systems that produce programs for NC machines. Any objects that need to be represented will be shown as wireframes. The system user will have to recognize collision problems. Usually the robot's tool point will be the only part of the robot shown. The user will obtain an understanding of the three-dimensional situation by changing viewpoint. This will be done in discrete steps rather than continuously so that a relatively simple display may be used. The output of the system will be a VAL program to run the PUMA.

Specifically the following subtasks will be performed:

- (1) Determine functionally how graphic programming would operate, i.e., develop a preliminary functional specification for a graphic programming language.
- (2) Implement simple wire-diagram graphics system to support graphics robot programming language.
- (3) Implement rudimentary graphics robot programming language based upon 1 and 2 above.

The geometric features used most will be points. Although it will be possible to enter point locations explicitly while programming, the usual approach will be specify some points qualitatively and, then, locate each other point relative to one of the qualitatively specified points. The exact locations of these latter points will be given later, usually by teaching. It is believed that this approach will lead to a simple, practical system.

A key problem will be the determination of the best way to handle the nongeometric parts of the program graphically. For example, conditional operations, or representation of forces will require some interesting solutions.

Subsequent phases of the project will include both coupling to various sensors being added to the robot (e.g. tactile, wrist force, or enhanced vision), and coupling to CAD geometric modelling language databases. The coupling to the geometric modelling database will provide much more detailed information about the individual parts. Position and orientation information about subsection of a part (e.g. holes, protuberances, etc.) will be available relative to some reference point for the object as a whole. Thus, teaching (or extracting via vision) the position of the object will result in the information

needed to access many points on the object. Thus, it will be possible to program complex assembly operations with only a minimum of teaching or visual recognition required. The long range goal of the project is to see how far it is possible to go with interactive graphics programming of robots.

A number of groups have considered aspects of using interactive graphics in robotics. Probably the most germane work has been that done by Stanford [Bin81] and McDonnell-Douglas for ICAM [Bin81].

6.5. Distributed Operating System Considerations

A large number of languages for programming robots off-line have been proposed. It is not clear which if any of these will become an industry-wide standard. Nor is it clear if any will be able to evolve with robotics as robots become more intelligent. On the one hand, there are simple languages such as Unimation's VAL, which are relatively straightforward "movement" languages. In the middle are more powerful languages such as MCL, the recent extension of APT. Their typical characteristic is that they allow concurrent programming of widely differing devices. Then there are high-level languages such as AUTOPASS [Lie77]. Typical characteristics of these are that they allow goal-oriented commands and incorporate an explicit world model.

Some of these languages are intended for programming robots only, while others are aimed at more than just robots. For example, MCL is applicable, it is claimed, to the entire manufacturing cell. Still other languages, such as languages for programming NC machines and languages for programming programmable controllers, address parts of the manufacturing system other than the robot. In other words, there are a number of languages applicable to manufacturing and they can vary in level and scope.

Clearly, then, a key question is what is the proper level and scope for a language or languages? Should there be one general language for integrated manufacturing? Or should the current multi-language situation be encouraged? Industry is obviously coming to a crossroads. As long as each piece could be treated as a largely separate system, overall system view was not so necessary. However, as integrated manufacturing systems come into being, they must be looked at as total systems, not a collection of parts. The question is: How is this best done? The purpose of this project is to answer this question.

More accurately, the purpose of this project is to investigate one type of answer to this question. In particular, it is assumed that a single general purpose language is not the best approach, and a hierarchically based system would be superior. At top of the hierarchy would be an operating system which manages the entire integrated manufacturing system. Below this would be either existing languages or new ones as found necessary. The issue is to design such a system so that it is applicable to most integrated manufacturing systems and can evolve with them.

The first step of the project will be to survey the operating systems for integrated manufacturing that currently exist. For example, there is Kearney and Trecker's System Gemini [Kla81]. Almost all these are ad hoc design with no claim to generality. However, they do constitute partial solutions to the problem. One general approach is that of Albus et al [ABN81].

The second step will be to design the system. In particular, the operating system will be designed with careful attention being given to the systems it must control, its distributed nature, and communication needs. New languages needed at lower levels of the branch will be specified. The result of

this second step will be a functional specification of the total software system.

7. Knowledge Representation (K. B. Irani)

7.1. Introduction

It is, by now, universally recognized that intelligent robots will play a central role in any manufacturing system of the future. However, much research remains to be done before the current robot can be equipped with sufficient intelligence for it to perform its envisaged tasks correctly, safely, and efficiently. The power of an intelligent robot will depend primarily on the quality and the quantity of the knowledge it has about the tasks and its ability to utilize the stored knowledge as well as its ability to acquire new knowledge. In recognition of these facts about an intelligent system, much of the focus of the artificial intelligence research has shifted to a knowledge-based paradigm as evidenced by the survey done by Brachman and Smith and reported in [BrS], as well as the chapter on Knowledge Representation in [BaF] and the work reported in [DaL]. Most of the current literature on Knowledge Representation has been referenced in these reports.

We propose to conduct research in the area of Knowledge Representation for intelligent robots. Our past experience in database research [DaL, Mi175, Be177, IPT79, Khi79, Kh181, Ch1, Mu1] especially qualifies us to provide a fresh perspective for this related area of Knowledge Representation. The similarities between the research done on knowledge representation in artificial intelligence on semantic database models in databases and on data abstraction in programming languages have been recognized [BrZ], and channels have now been opened up for researchers in one field to move comfortably into the research in another field with the view to enrich and enhance these fields. That is not to say that the knowledge and experience gained in one area is directly applicable to the other area. There are some important differences between the two fields of knowledge representation for intelligent robots and for databases. These differences must be appreciated. To mention just one example, the scope and the grain size of the data to be represented is usually known to a database designer whereas these are two of the crucial parameters to be carefully determined by the designer of the knowledge representation scheme for the correct and efficient operation of an advanced robot.

7.2. Discussion

An intelligent robot may be defined as one which is capable of performing its tasks correctly, safely, and efficiently even under circumstances unanticipated during its explicit instructional stage. In the context of intelligent robots, we are using the term "Knowledge Representation" in its broadest sense. To us, therefore, the issue of knowledge representation involves both the epistemological aspects as well as its heuristics. Under the heading of epistemology, we include the study of the following questions:

1. What facts about the external world are available to the robot?
2. How should these facts be internally represented and to what detail? The answer to this question determines to a large extent the complexity of the data structure and the efficiency of the algorithms. Currently several representation schemes have been available. Among them are logic, procedural representations, semantic networks, production systems, and direct representation. At present there is no theory which makes it possible to determine if one representation scheme is better than the

other.

3. If some facts are not internally represented but are made available to the robot, what internal knowledge and what decision procedures should the robot use to determine if the available facts are worthy of consideration for the execution of the task at hand and/or worthy of assimilation for future use? If these new facts are to be assimilated, how should they be integrated with the current internal knowledge?
4. Not all facts may be explicitly represented. However, if the method of representation is properly selected, it may permit the derivation of other facts. This represents reasoning on the part of the robot. Formal reasoning using first predicate calculus and procedural reasoning have been largely used. Human beings also reason using analogies, abstractions and generalizations. Currently there are no available methodologies which enable these types of reasoning to be incorporated into robots.

The heuristics are concerned with processing and utilizing knowledge. Good guesses and plausible reasoning make good heuristics. There require meta-level reasoning. Knowledge about the extent or the limitations of its knowledge would enable the robot to logically take short cuts during its process of reasoning. Thus epistemology and heuristics are intimately interrelated. In designing a knowledge-based system for intelligent robots, both the data structure as well as the procedures which interpret and use these data structures need to be considered simultaneously. We shall refer to the two together as a knowledge representation scheme.

As mentioned above, in the past, several types of knowledge representation schemes have been proposed. Each type has its advantages and disadvantages. For example, logic is natural, precise, flexible, and modular. However, since the representation and the processing parts are separated, it does not meet head-on the most difficult part of knowledge-based systems, namely, how to use the knowledge stored in the system. At the present time semantic networks are a popular representation scheme in artificial intelligence. However, with large amounts of data to be explicitly represented, the computational problem becomes very complex. Besides, the simple idea of representing objects, concepts or events by nodes and relations by arcs, often leads to some subtle difficulties.

It should be clear by now that none of the existing knowledge representation schemes is perfect and that one scheme may be better suited in a given situation than the other.

7.3. Proposed Research

It is apparent that a need exists for a formal methodology for the design of an "optimum" knowledge representation scheme for a knowledge based system. We propose to take the first step towards such a methodology for the case of an intelligent robot working in a given environment and performing a given set of tasks.

To begin with, we will develop a general model for the "total knowledge" available in an environment. This general model will be derived by abstracting the essential features of several instances of "environment." We expect that this model would represent the "primitives" of the environment as well as various hierarchies of abstractions and the redundancy of both inter-level and intra-level relationships. Depending on the frequency of the tasks to be performed and the expected response time of the robot to each task, it is expected that in the eventual scheme, some of these relationships will be

represented explicitly while some others will have to be derivable by "reasoning." Not all of the relationships represented in the general model may need to appear either explicitly or implicitly in the ultimate scheme. The type of reasoning to be employed would be determined by the "optimum" scheme.

The ultimate measures of performance should be correctness, safeness, and efficiency. While the correctness and efficiency measures may be intuitively obvious, it may not be clear how safeness is to be measured. One of the approaches could be to introduce unforeseen situations into the normal environment and to determine the extent of the system response under each of the representation schemes. This type of safety measure, therefore, would depend on the ability of the system to select and utilize new knowledge presented to it.

We expect to consider all the existing representation schemes as well as their feasible hybrids and also some new ones. We will test our methodology by designing instances of representation schemes and simulating them.

7.4. References

[AAT79]

Altschuler, M. D., B. R. Altschuler, J. Taboada: Measuring Surfaces Space Coded by a Laser Projected Dot Matrix, in *Imaging Applications for Automated Industrial Inspection and Assembly* (R.P. Kruger, editor) SPIE 1979.

[Agi77]

G. J. Agin, "Servoing with Visual Feedback," *Seventh International Symposium on Industrial Robots*, Tokyo, Japan, Oct. 1977.

[Alb75]

Albus, J. S., "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)," *Journal of Dynamic Systems, Measurement and Control, Transaction ASME, Series G, Vol. 97, No. 3, Sept. 1975* pp 220-227.

[Alb75b]

Albus, J. S., "Data Storage in the Cerebellar Model Articulation Controller (CMAC)," *Journal of Dynamic Systems, Measurement and Control, Trans. ASME, Sept. 1975*.

[APF80]

Altschuler, M. D., J. L. Posdamer, G. Frieder, M. J. Mantley, B. R. Altschuler, and J. Taboada, "A Medium Range Vision Aid for the Blind," *Proceedings for the 1980 International Conference on Cybernetics*, IEEE 1980.

[BaF]

Barr, A. and Feigenbaum, E.A., *The Handbook of Artificial Intelligence*, Volume 1.

[Bat81]

Barron, H. G. and J.M. Tenenbaum, "Computational Vision," *Proceedings of IEEE*, Vol. 60, pp. 572-595, May 1981.

[Bej74]

Beyczy, A. K., "Robot Arm Dynamics and Control," *Technical Memo 33-669*, Jet Propulsion Laboratory, February 1974.

[Bel77r]

Berelian, E. and Irani, K.B., Evaluation and Optimization of Database Structures. *Proceedings of the International Conference on Very Large*

Databases, 1977.

- [BoP73]
Bolles, R., R. Paul, "The Use of Sensory Feedback in a Programmable Assembly System," *Stanford Artificial Intelligence Laboratory Memo AIM-220*, Stanford University, October 1973.
- [BoW79]
Borky, J.M. and K.D. Wise "Integrated Signal Conditioning for Silicon Pressure Sensors," *IEEE Transactions on Electron Devices*, vol. 26, pp. 1906-1910, December 1979.
- [BrS80]
Brachman, R. J. and Smith, B.C., *1980 SIGART Newsletter 70* (Special issue on knowledge representation).
- [BrZ]
Brodie, M.L. and Zilles, S.N., *Proceedings of the Workshop on Data Abstraction, Databases and Conceptual Modellings*.
- [CAI79]
Proceedings of the 6th International Joint Conference on Artificial Intelligence Tokyo, Japan, August 1979.
- [ChI]
Chung, C. and Irani, K.B., *Query Processing in Distributed Database Systems*. To be published.
- [CIW79]
Clark, S.K. and K.D. Wise, "Pressure Sensitivity in Anisotropically-Etched Thin Diaphragm Pressure Sensors," *IEEE Transactions on Electron Devices* vol. 26, pp. 1886-1896, December 1979.
- [CRS82]
Chow, K. J.P./ Rode, D. H. Seib, and J. D. Blackwell, "Hybrid Infrared Focal-Plane Arrays," *IEEE Transactions Electron Devices*, vol 29, January 1982 (to be published).
- [Da80]
Proceedings of DARPA Image Understanding Workshop, 1977, 1978, 1979, 1980.
- [DaL]
Davis, R. and Lenat, D.B., *Knowledge-Based Systems in Artificial Intelligence*, McGraw-Hill.
- [DeH55]
J. Denavit, R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *Journal of Applied Mechanics*, June 1955.
- [Ern62]
H. A. Ernst, "MH-1, A Computer-Operated Mechanical Hand," *1962 Spring Joint Computer Conference*, San Francisco, May 1-3, AFIPS Proceedings, pp. 39-51.
- [Fin74]
R. Finkel, et al., "AL, a Programming System for Automation," *Memo AIM-243*, Stanford Artificial Intelligence Laboratory, Nov. 1974.
- [Fu70]
K. S. Fu, "Learning Control Systems --- Review and Outlook," *IEEE Trans. on Automatic Control*, April 1970.
- [Fu71]
K. S. Fu, "Learning Control Systems and Intelligent Control Systems" An

Intersection of Artificial Intelligence and Automatic Control," *IEEE Transaction on Automatic Control*, vol. AC-15, February 1971.

[Gar76]

Gara, A., *Appl. Opt.* 15, 510, (1976).

[Gro72]

R. C. Groome, "Force Feedback Steering of a Teleoperator System," *SM Thesis*, MIT, Department of Aeronautics and Astronautics, August 1972.

[Hol80]

Holland, S. W., et. al. "A Vision Controlled Robot for Part Transfer," *Publication GMR-312L*, General Motors Research Laboratory, Warren, Michigan.

[HoT80]

Horowitz, R., M. Tomizuka, "An Adaptive Control Scheme for Mechanical Manipulators-Compensation of Nonlinearity and Decoupling Control," *Dynamic System and Control Division of the ASME*, Winter Annual Meeting, Chicago, Ill., Nov. 1980.

[ICAM78]

"The ICAM Program Report", issued periodically by AFML/LTC, WPAFB, Ohio, 45433, - first issue, Vol. 1, no. 1, June, 1978.

[Ino74]

H. Inoue, "Force Feedback in Precise Assembly Tasks," *MIT Artificial Intelligence Laboratory, Memo 308*, MIT, August 1974.

[IPT79]

Irani, K.B., Purkayastha, S. and Teorey, T.J., A Designer for DBMS Processable Logical Database Structures. *Proceedings of Fifth International Conferences on Very Large Databases*, 1979.

[KaB71]

Kahn, M. E., B. Roth, "The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains," *Transaction of the ASME, Journal of Dynamic Systems, Measurement, and Control*, Sept. 1971 pp 164-172.

[KhI79]

Khabbaz, N. and Irani, K.B., A Model for a Combined Network Design and File Allocation for Distributed Databases. *First International Symposium on Distributed Computer Systems*, 1979.

[KhI81]

Khabbaz, N. and Irani, K.B., A Combined Communication Network Design and File Allocation for Distributed Databases. *Second International Conference on Distributed Computing*, 1981.

[JTW81]

Jackson, T.N., M.A. Tischler, and K.D. Wise, "An Electrochemical P-N Junction Etch-Stop for the Formation of Silicon Microstructures," *IEEE Electron Device Letters* vol 2, pp. 44-45, February 1981.

[KTS80]

Korke, N., I Takemoto, K. Satoh, S. Hanamura, S. Nagahara, and M. Kubo, "MOS Area Sensor; Design Considerations and Performance of an n-p-n Structure 484x384 Element Color MOS Imager," *IEEE Transactions on Electron Devices*, vol 26, pp 1676-1681, August 1980.

[LaW80]

Lahiji, G.R. and K.D. Wise, "A Monolithic Thermopile Detector Fabricated using Integrated Circuit Technology," *Digest of Technical Papers*, 1980 International Electron Device Meeting, Washington, D.C., pp. 676-679,

December 1980.

[Lee80]

C. S. G. Lee, "Design Methodology of Force Control for the Sensor-Controlled Manipulators," Presented at the *101st ASME Winter Annual Meeting*, Chicago, Nov. 17-20 1980.

[Lee81]

Lee, S., paper delivered at *NASA Conference*, Hampton, VA. Aug. 18-19, 1981.

[Ler80]

Lerner, E. J., "Computers That See," *IEEE Spectrum* Vol. 17, No. 10, October, 1980.

[LeS81]

Leith, E. and G. Swanson, *Appl. Opt.* 20, 381, (1981).

[Lew74]

Lewis, R. A., "Autonomous Manipulation on a Robot: Summary of Manipulator Software Functions," *Technical Memo 33-679, Jet Propulsion Lab.*, March, 1974.

[LeW82]

Lee, K.W. and K.D. Wise, "SENSIM: A Simulation Program for Silicon Pressure Sensors," *IEEE Transactions Electron Devices*, vol 20, January 1982 (to be published).

[LeWi2b]

Lee, Y.S. and K.D. Wise, "A Batch-Fabrication Capacitive Pressure Sensor with Low Temperature Sensitivity", *IEEE Transaction Electron Devices* vol 29 January 1982 (to be published).

[Lie77]

Lieberman, L. I., and Wesley, M. A., "AUTOPASS: An Automatic Programming System for Computer Controlled Mechanical Assembly", *IBM Journal of Research and Development*, July 1977, pp. 321-333.

[Loh78]

Lohmann, A., *Appl. Opt* 16 265 (1977).

[LoR78]

Lohmann, A., and W. T. Rhodes, *Appl. Opt.* 17, 1141 (1978).

[LTB81]

Lee, T.H., T.J. Treadwell, B.C. Burkey, J.S. Hayward, T. M. Kelly, R. P. Khosla, and D. L. Losee, "A Novel Solid-State Image Sensor for Image Recording at 2000 Frames per Second," *Digest of Technical Papers, International Electron Devices Meeting*, 1981 (paper 19.4).

[LWP80]

Luh, J. Y. S., M. W. Walker and R. P. C. Paul, "On-line Computational Scheme for Mechanical Manipulators," *ASME Journal of Dynamics Systems, Measurement and Control*, Vol. 102, June 1980, pp. 69-76.

[MacD80]

MacDonald Douglas Corporation, "MCL Language Definition, version 1.0," *Preliminary Information Release*, November, 1980.

[McC68]

J. McCarthy, et al., "A Computer with Hands, Eyes, and Ears," *1968 Fall Joint Computer Conference, AFIPS Proceedings*, pp. 329-338.

[MeW79]

Meindl, J.D. and K.D. Wise, eds., "Special Issue on Solid-State Sensors,

- Actuators, and Interface Electronics," *IEEE Transactions on Electron Devices* vol 26, pp 1861-1978, December 1979.
- [MiI75]
Mitoma, M. and Irani, K.B., Automatic Database Schema Design and Optimization. *Proceedings of the International Conference on Very Large Databases*, 1975.
- [MoG80]
Morris, G.M. and N. George, *Opt. Lett.* 5, 446 (1980).
- [MuI]
Murad, O. and Irani, K.B., *Distributed Database: Selection and Allocation of Data Elements*. To be published.
- [NeW74]
J. Nevins, D. E. Whitney, et al., "Exploratory Research in Industrial Modular Assembly," *C. S. Draper Laboratory Report*, NSF project reports 1 to 4, covering the period 1974 to August 1976.
- [PoA]
Posdamer, J.L., M. D. Altschuler, *Surface Measurement by Space Encoded Projected Beam Systems*, accepted for publication in *Computers in Industry* (copies available in the University of Michigan from G. Frieder).
- [PaS76]
R. Paul; B. Shimano, "Compliance and Control," *Proceedings of Joint Automatic Control Conference*, Purdue University, July 1976.
- [Pau72]
Paul, R., "Modeling, Trajectory Calculation, and Servoing of a Computer-Controlled Arm," *Stanford Artificial Intelligence Laboratory Memo AM-177*, Nov. 1972.
- [Pau75]
Paul, R., "Manipulator Path Control," *1975 IEEE International Conference on Cybernetics and Society*, San Francisco, California.
- [Pau77]
Paul, R. et al., "Wave: A Model-Based Language for Manipulator Control," *The Industrial Robot*, Vol. 4, No. 1, pp. 10-17.
- [Pau79]
Paul, R., "Evaluation of Manipulator Control Programming Languages", *Proceedings of the IEEE Conf. on Decis & Control*, Fort Lauderdale, Fla., Dec. 12-14, 1979. pp. 252-256.
- [Per78]
Perkins, W. A., "A Model-Based Vision System for Industrial Parts," *IEEE Trans. on Computers*, Vol. C-27, pp. 126-143, 1978.
- [Pie68]
Pieper, D. L., "The Kinematics of Manipulators Under Computer Control," *Computer Science Department, Stanford University, Artificial Intelligence Project Memo No. 72*, Oct. 24, 1968.
- [ReH79]
Reddy, D. R. and R. W. Hou, "Computer Architectures for Vision," in *Computer Vision and Sensor-Based Robots*, Edited by G. G. Dodd and L. Rossol, Plenum Press, New York 1979, pp. 169-186.
- [Rob63]
L. G. Roberts, "Machine Perception of Three-Dimensional Solids," *Lincoln Lab, Report No. 315*, May 1963.

- [Ruof79]
Ruoff, C. F., "TEACH - A concurrent Robot Control Language", *Proceedings of CAMSAC 79*, Chicago, Illinois, November 1979, pp. 442-445.
- [SaL79]
Saridis, G. N., C.S.G. Lee, "An Approximation Theory of Optimal Control for Trainable Manipulators," *IEEE Transaction on Systems, Man and Cybernetics*, vol. SMC-9, no. 3, March 1979 pp 152-159.
- [Shi78]
B. E. Shimano, "The Kinematic Design and Force Control of Computer Controlled Manipulators," *Stanford Artificial Intelligence Laboratory, Memo AIM-313*, March 1978.
- [Shi79]
Shimano, B., "VAL: A Versatile Robot Programming and Control System", *Proceedings of COMPSAC 79*, November 1979, pp.878-883.
- [SIR80]
Proceedings of the 10th International Symposium on Industrial Robots, Milan, Italy, March 1980.
- [Sk166]
J. Sklansky,, "Learning Systems for Automatic Control," *IEEE Trans. on Automatic Control*, Vol. AC-11, Jan. 1966.
- [TML80]
Turney, J., T. N. Mudge, C. S. G. Lee, "Equivalence of Two Formulations for Robot Arm Dynamics," *SEL Report 142, ECE Department*, University of Michigan, Dec. 1980.
- [Uic67]
Uicker, J. J., "Dynamic Force Analysis of Spatial Linkages," *Jam*, June 1967, pp. 418-423.
- [Whi69]
Whitney, D. E., "Resolved Motion Rate Control of Manipulators and Human Prostheses," *IEEE Transaction on Man-Machine Systems*, vol. MMS-10, no. 2, June 1969 pp 47-53.
- [Whi76]
D. E. Whitney, "Force Feedback Control of Manipulator Fine Motion," *Proceeding of Joint Automatic Control Conference*, Purdue University, July 1976.
- [WiG75]
P. Will, D. Grossman, "An Experimental System for Computer Controlled Mechanical Assembly," *IEEE Transactions on Computer*, Vol. C-24, No. 9, Sept. 1975.
- [Wis82]
Wise, K.D., ed., "Special Issue on Solid-State Sensors, Actuators, and Interface Electronics," *IEEE Transactions on Electron Devices*, vol 29, January 1982 (to be published).
- [WMF79]
Wallace, T. P., O. R. Mitchell, and K. Fukunaga, "Three Dimensional Shape Analysis Using Local Shape Descriptors," *Proceedings of the IEEE Conference on Pattern Recognition and Image Processing*, Chicago, June 1979.
- [YanL]
Yang, G. and E. Leith, *Opt. Comm.* 36, 14 (1981).



[Yu78]

Yu, F.T.S. Appl. Opt. 17, Letter 5, 446 (1980).

[ZaR72]

Zahn, C. T. and R.Z. Roskies, "Fourier Descriptors for Plan Closed Curves,"
IEEE Trans. on Computers, Vol. C-21, pp. 269-281, 1972.