Tenth Conference on Production Research and Technology

P-128

February 28 - March 2, 1983
Detroit, Michigan

Hosted by:
SAE The Engineering Resource For Advancing Mobility

Advanced Control for Multirobot Assembly Systems

C. S. G. Lee and T. N. Mudge
Department of Electrical and Computer Engineering
The University of Michigan
Ann Arbor, MI

ABSTRACT

The proposed research is directed toward developing methodologies for the control and coordination of multirobot batch assembly systems using distributed computer-based control. Components of this broad-based research include the investigation and development of force feedback control strategies with statistical decision-making capabilities for fine motion; the investigation of advanced control strategies using modern control, estimation and identification concepts; vision analysis and recognition of three dimensional objects; and design and implementation of low-cost high performance computation structures for robot arm control and vision analysis. Special emphases are placed on improving the dynamic performance of the overall robotic systems in the areas of real-time control, generalized system architecture and the use of sensory information for the control of senor-guided robots for manufacturing and assembly tasks. The long term outcome of this research will be a working experimental multirobot work-station with vision and sensors under the supervision of a modularized software control package capable of performing high-tolerant batch assembly tasks.

PROGRAM ACHIEVEMENT

The grant was received in January, 1982. A significant portion of our past research effort has been in studying and understanding the operation of computer-controlled sensor-based industrial robots, particularly in the areas of robot arm dynamics (1,2,3,4)*, gross motion control (5,6,7,8), pattern recognition methods for force recognition in insertion process (9), and the control architecture for PUMA robot arms (10,11). Our short-term goal is to extend this research to the development of a high performance sensor-based robot capable of performing assembly tasks.

RESEARCH RESULTS SINCE JANUARY 1982

Last year, major research efforts focused on five areas:

ADAPTIVE CONTROL FOR MECHANICAL MANIPULATORS - Given the equations of motion of a manipulator, the purpose of robot arm control is to maintain a prescribed motion for the arm along a desired arm trajectory by applying corrective compensation torques to the actuators to adjust for any deviations of the arm from that desired trajectory. The current approach to robot arm control system design treats each joint of the robot arm as a simple servomechanism. Such modeling is inadequate because it neglects the motion and configuration of the whole arm mechanism. These changes in the parameters of the controlled system are significant enough to render conventional feedback control strategies ineffective. The result is reduced servo response speed and damping, which limits the precision and speed of the end-effector. Any significant performance gain in this and other areas of robot control require the consideration of elaborate dynamic models and sophisticated control techniques, and the exploitation of computer architecture.

An adaptive control based on the perturbation equations in the vicinity of a desired trajectory has been studied (6). The highly coupled nonlinear dynamic equations of a manipulator are expanded in the vicinity of a preplanned joint trajectory to obtain the perturbation equations. A linear quadratic controller is then designed for the perturbation equations about the desired trajectory. The joint torques consist of the nominal torques computed from the Newton-Euler equations of motion and the variational torques computed from the perturbation equations. The recursive least square identification scheme is used to perform on-line parameter identification for the unknown parameters in the perturbation equations. The parameters of the perturbation equations and the feedback gains of the linear quadratic controller are updated and adjusted in each sampling period successively to obtain the necessary control effort. This adaptive control strategy reduces the manipulator control problem from a nonlinear control to controlling a linear control system about a desired trajectory. Furthermore, a clear advantage of such formulation is that the nominal and variational torques can be computed separately and simultaneously. A control block diagram of the adaptive control system is shown in Figure 1. The complexity of computing the variational torques is tabulated in Table 1.

Computer simulation studies of a three-link PUMA robot arm are performed on a VAX-11/780 computer to evaluate the validity of the use of the perturbation equations and the performance of the adaptive controller. We simulated the adaptive control strategy and

*Number in parentheses designate references at end of paper.

compared it with the conventional position plus derivative control (PD) method for various loading conditions for a given trajectory. The performance of both controllers are compared and tabulated in Table 2 for three different load conditions: (a) No-load and 10% error in inertia tensor matrix, (b) Half of maximum load and 10% error in inertia tensor matrix and (c) Maximum load and 10% error in inertia tensor matrix. In each case 10% error in inertia matrices means ± 10 % error about its measured inertial values. Plots of position errors for case (a) for both controllers are shown in Figures 2 and 3. The applied torques computed from the adaptive controller for case (a) is shown in Figure 4. A more detailed discussion and derivation of the adaptive control with computer simulation can be found in (6).

NONLINEAR FEEDBACK CONTROL - The use of nonlinear feedback components to minimize the effects of the nonlinear coupling terms in a nonlinear control system is not new to control practitioners, but it is a good approach to control multi-joint robot arms. There is a substantial body of nonlinear control theory which may allow one to design a near-optimal control for mechanical manipulators. We have been looking at this problem for the last six months and had designed a suboptimal control with nonlinear feedback. Basically the control system consists of two servo loops. The inner loop consists of nonlinear feedback to minimize the nonlinear effects while the outer loop controls the quasi-linearized control system with switching functions that stablize the control system. In the next sections, we briefly describe the reformulation of the robot arm dynamic model to accommodate the nonlinear feedback and the linear quadratic controller.

The Lagrange-Euler equations of motion of a manipulator can be written as (12):

$$\tau_i = \sum_{k=1}^{n} C_{ik}\ddot{q}_k + \sum_{k=1}^{n}\sum_{m=1}^{n} C_{ikm}\dot{q}_k\dot{q}_m + C_i \; ; \; i = 1,2,\ldots,n \qquad (1)$$

where:

$$C_{ik} \stackrel{\Delta}{=} \sum_{j=\max(i,k)}^{n} Tr\left[\frac{\partial \mathbf{A}_0^j}{\partial q_k} \mathbf{J}_j \left(\frac{\partial \mathbf{A}_0^j}{\partial q_i}\right)^t\right] \; ; \; i\,,k = 1,..,n$$

$$C_i \stackrel{\Delta}{=} \sum_{j=1}^{n}\left[-m_j \mathbf{g}\frac{\partial \mathbf{A}_0^j}{\partial q_i}\bar{\mathbf{r}}_j\right]$$

$$C_{ikm} \stackrel{\Delta}{=} \sum_{j=\max(i,k,m)}^{n} Tr\left[\frac{\partial^2 \mathbf{A}_0^j}{\partial q_k \partial q_m}\mathbf{J}_j\left(\frac{\partial \mathbf{A}_0^j}{\partial q_i}\right)^t\right]$$

Since

$$\frac{dC_{ik}}{dt} = \sum_{m=1}^{n} C_{ikm}\dot{q}_m + \sum_{m=1}^{n} C_{kim}\dot{q}_m$$

the equations of motion can be rewritten as:

$$\tau_i = \frac{d}{dt}\left[\sum_{k=1}^{n} C_{ik}\dot{q}_k\right] + C_i - \sum_{k=1}^{n}\sum_{m=1}^{n} C_{kim}\dot{q}_k\dot{q}_m \; ; \; i = 1,..,n \qquad (2)$$

It should be noted that $C_{ikm} \neq C_{kim}$. If we let q_i^d be the desired angular position, and the angular position error as $r_i = q_i - q_i^d$, by substituting r_i and $\dot{r}_i$ into Eq. (2), we have

$$\tau_i = \frac{d}{dt}\left[\sum_{k=1}^{n} D_{ik}\dot{r}_k\right] + D_i - \sum_{k=1}^{n}\sum_{m=1}^{n} D_{kim}\dot{r}_k\dot{r}_m \qquad (3)$$

where

$$D_{ik}(r_1, \cdots, r_n) \stackrel{\Delta}{=} C_{ik}(q_1 - q_1^d, \cdots, q_n - q_n^d)$$

$$D_i(r_1, \cdots, r_n) \stackrel{\Delta}{=} C_i(q_1 - q_1^d, \cdots, q_n - q_n^d)$$

$$D_{kim}(r_1, \cdots, r_n) \stackrel{\Delta}{=} C_{kim}(q_1 - q_1^d, \cdots, q_n - q_n^d)$$

In other words, the control system is to be driven to the origin $r_i = 0 \; , \; 1 \leq i \leq n$.

Now consider the input torque τ_i to be consisted of a feedback σ_i and a control u_i as shown in Figure 5. That is, $\sigma_i + u_i = \tau_i \; , \; 1 \leq i \leq n$. The feedback control component σ_i can be chosen as:

$$\sigma_i = D_i - \sum_{k=1}^{n}\sum_{m=1}^{n} D_{kim}\dot{r}_k\dot{r}_m \qquad (4)$$

With this feedback control, the control u_i can be found to be:

$$u_i + \sigma_i = \tau_i \; ; \text{ and } u_i = \frac{d}{dt}\left[\sum_{k=1}^{n} D_{ik}\dot{r}_k\right] \qquad (5)$$

If we define the generalized momenta p_i as:

$$p_i = \sum_{k=1}^{n} D_{ik}\dot{r}_k \; , \; 1 \leq i \leq n \; ; \; or\ \mathbf{p} = \mathbf{D}\dot{\mathbf{r}} \qquad (6)$$

then Eqs. (5) and (6) become:

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{u} \\ \dot{\mathbf{r}} &= \mathbf{D}^{-1}\mathbf{p} \end{aligned} \qquad (7)$$

where

$$\mathbf{p} = (p_1,..,p_n)^t \; ; \; \mathbf{u} = (u_1,..,u_n)^t \; ; \; \mathbf{r} = (r_1,..,r_n)^t$$

Eq. (7) can be expressed in a matrix differential equation form as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \qquad (8)$$

where

$$\mathbf{x} = (p_1, \cdots, p_n, r_1, \cdots, r_n)^t = \begin{bmatrix} \mathbf{p} \\ \mathbf{r} \end{bmatrix} \; ; \; \mathbf{x} \in \mathbf{R}^{2n}$$

$$\mathbf{A}_{2nx2n} = \left[\begin{array}{c|c} 0 & 0 \\ \hline \mathbf{D}^{-1} & 0 \end{array}\right]_{2nx2n}$$

$$\mathbf{B}_{2nxn} = \begin{bmatrix} 1 & 0 & 0 & \cdot & 0 \\ 0 & 1 & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & 1 \\ \hline 0 & 0 & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & 0 \end{bmatrix}_{2nxn} = \begin{bmatrix} I_n \\ 0 \end{bmatrix}$$

Using the recursive computation technique in (13), the nonlinear feedback in Eq. (4) can be computed in O(n) time. Hence the implementation of the nonlinear feedback is straight forward.

As a result of the nonlinear feedback, the control system becomes quasi-linearized as in Eq. (7). Treating Eq. (8) as a linear system, a linear quadratic controller with switching functions for stability can be designed. A more detailed discussion of this controller design can be found in (7).

PATTERN RECOGNITION METHODS FOR FORCE RECOGNITION IN INSERTION PROCESS - The primary objectives of 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 methods for force recognition and guidance control so that the manipulator can improve its performance in fine motion. The pattern recognition techniques embedded in the force 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's end-effector. The guidance control then improves the performance of the arm by updating/modifying the feedback gains of the appropriate control law in each of the control situation. Similar approach of using a combination of pattern recognition and modern control theory has been used successfully in other areas (14,15,16).

The following sections briefly present the concept of applying pattern recognition techniques to force recognition in insertion process.

On observing the resolved force sensor measurements, a decision rule based on minimum probability of the error can be written as $d(X) = P(\omega_i \mid X) > P(\omega_j \mid X)$ for all $j \neq i$ and $X \in \omega_i$, where X denotes the resolved force sensor vector which is treated as a random variable and ω_i are the control situations (or various contact configurations) for $i = 1,\ldots,m$. That is, if the a posteriori probability $P(\omega_i \mid X) = \frac{f(X \mid \omega_i)P(\omega_i)}{f(X)}$ is the greatest for class i then the hand is "classified" as in the control situation ω_i. The problem then is to find the a priori probability $P(\omega_i)$ and the conditional probability density function $f(X \mid \omega_i)$ for $i = 1,\ldots,m$. The a priori probability $P(\omega_i)$ can be estimated from $P(\omega_i) = \frac{N_i}{N}$ where N_i are the samples from ω_i and $N = \sum_{i=1}^{m} N_i$.

If the functional form of the conditional probability density function $f(X \mid \omega_i)$ is known up to a set of unknown parameters, then the problem is reduced to a parameter estimation problem. If the functional form of $f(X \mid \omega_i)$ is not known, then the problem is the estimation of a probability density function. Usually the functional form of $f(X \mid \omega_i)$ is not known in the insertion process, the k-nearest neighbor approach can be used to estimate the probability density function based on N samples from the distribution.

Unfortunately one of the disadvantages of the k-nearest neighbor method for estimating the density function is that it is totally *empirical (experimental)* and it does not contain any information about the geometric structure and constraints of the problem. A statistical pattern recognition method based on force sensor measurements due to the geometric constraints and the quasi-static equilibrium conditions of the insertion process must be devised to estimate the probability density function $f(X \mid \omega_i)$ *analytically* and not experimentally. The contact points of a peg with a hole are random but there are geometric relations between these contact points which make these random contact points *dependent* rather than independent.

To illustrate the method, consider the control situation in Figure 6. Let R_B and R_A be the reaction forces at contact points B and A (in coordinate frame C_F) respectively, and R be the radius of the hole. From Figure 6, the geometric constraints of the forces are stated but not derived here. (A more detailed discussion of the geometric constraints and its quasi-static equilibrium conditions for various control situations of a peg in a hole insertion process can be found in (9))

$$r_B = r_o + T_F^H X_B$$

$$r_A = r_o + T_F^H (X_A - h_A k)$$

$$X_B = R\,Cos\vartheta_B\, i + R\,Sin\vartheta_B\, j$$

$$X_A = R\,Cos\vartheta_A\, i + R\,Sin\vartheta_A\, j$$

where ϑ_B and ϑ_A are random angles; r_A and r_B are the position of the contact points A and B with respect to the coordinate frame C_F respectively; T_F^H is the transformation matrix between coordinate frames C_H and C_F $(C_F = T_F^H C_H)$

$$T_F^H = \begin{bmatrix} R_F^H & | & r_o \\ - & | & - \\ 0 & | & 1 \end{bmatrix}$$

The orientation submatrix of T_F^H is fixed, because neither C_F nor C_H changes orientation. The only component that changes as the insertion process proceeds is r_o.

From Figure 6, R_A and R_B, the reaction forces at the contact points are considered random and have distribution f_{R_A} and f_{R_B} independent of the contact condition, that is $f_{R_A}(R_A \mid \omega_i) = f_{R_A}(R_A \mid \omega_j)$ for all i and j. Furthermore we can assume $f_{R_A}(R_A) = f_{R_B}(R_B)$ that is the density functions are identical (but R_A and R_B are independent random vectors). $f_{R_A}(R_A)$ can be easily obtained from the experimental data.

Let X_F and X_M be the force and moment components of the resolved force measurement respectively, then the resolved force vector X can be expressed as:

$$X = \begin{bmatrix} X_F \\ X_M \end{bmatrix} \;;\; X_F = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \;;\; X_M = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}$$

The quasi-static equilibrium conditions for Figure 6 require that:

$$\begin{aligned} X_F &= R_B + R_A \\ X_M &= r_B \times R_B + r_A \times R_A \end{aligned} \tag{9}$$

With the resolved force vector calculated from Eq. (9), the conditional probability density function $f(X \mid \omega_i)$ can be estimated from the Synthetic Sampling (Monte Carlo) technique:

$$X_F = g_F(R_A, R_B) = R_A + R_B$$

$$X_M = g_M(R_A, R_B, \vartheta_A, \vartheta_B, r_o) = r_B \times R_B + r_A \times R_A$$

In summary, the pattern recognition method for force recognition is based on a structural approach to estimate the probability density function $f(X|\omega_i)$ to be used for recognizing various contact configurations in insertion process. The idea is to generate a random contact configuration of the type belonging to class ω_i and generate "admissible" random contact reaction forces and then calculate the resolved force sensor vectors based on the geometric constraints and the quasi-static equilibrium conditions and repeat this process for $N(N>10^3)$ times and find the probability density function $f(X|\omega_i)$ from the synthetic sampling technique. The method can be summaried in Algorithm 1:

Algorithm 1:

1. Set $k=0$
2. Generate $\mathbf{r}_o$
3. Generate admissible random contact points belonging to class ω_i
4. Generate admissible random reaction forces (i.e. $\mathbf{R}_A$ and $\mathbf{R}_B$ depending on the class ω_i)
5. Calculate the resolved force sensor vector based on the geometric constraints and the quasi-static equilibrium conditions
6. If $k>N$, then find the distribution from data generated, else goto step 2.

At this time, we are at the stage of performing computer simulation of the insertion process on our VAX-11/780 computer. (i.e. try to verify the validity of using pattern recognition method for recognizing resolved force vectors for various contact situations.) We have completed instrumenting our PUMA robot arm with a wrist force sensor from Robot Technology Inc. Sensor calibration and the related software are being written.

COMPUTATION STRUCTURE FOR ROBOT ARM CONTROL - As stated in the adaptive control design, the nominal torques are computed from the Newton-Euler equations of motion. Normally such numerically intensive computations can be performed by mainframe computers or microprocessors. We argue that a better solution is the use of a special purpose processor (APAC--Attached Processor for Arm Control) to compute the nominal joint torques. We had designed such an APAC and performed computer simulation on it (See Figure 7). The APAC controls the robot arm as a whole system and performs the dedicated functions of servo control. The following sections present the preliminary specification for a very large scale integrated circuit (VLSIC) implementation of our proposed APAC, a single chip processor for dedicated numerically intensive control applications. A more detailed discussion and design of the APAC can be found in (10,11).

The APAC functions as an attached processor of a general purpose minicomputer. It operates on 32 bit floating point data. Conceptually, it lies between Floating Point Systems' AP120B, a high performance numerically oriented attached processor, and the Intel 8087, a single chip numerically oriented attached processor in the Intel 8086 family of components. All three work with floating-point numbers. The APAC 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. This preliminary study assumes the APAC will be implemented in nMOS. However, our eventual aim is to investigate the design of the APAC in a faster technology that still has the density of integration associated with nMOS. A prime candidate is the I3L (Isoplanar Integrated Injection Logic) technology developed by Fairchild Corporation. The major components are as follows:

1. A 32 bit floating point adder unit (AU).
2. A 32 bit floating point multiplier unit (MU).
3. A 256x32 register file (RF).
4. A 32x32 bit input buffer (IB).
5. A 32x32 bit output buffer (OB).
6. A 1Kx50 bit program memory (PM).
7. A 4x10 bit program counter stack (PCS).
8. A 1x50 bit program memory data register (PMDR).
9. A 16 bit loop counter (LC).
10. Condition code logic (CC).

To illustrate the effectiveness of the APAC a functional simulation was performed using APL. The forward and backward recursive equations for computing the nominal actuator torques were used as a benchmark, since they are the major computational task in the joint torque computation. These were programmed for the APAC. A listing of the program showing how the function units can be efficiently scheduled can be found in (10). The APAC is operating at its maximum rate when both function units are in streaming mode. In this mode it is producing the results of two floating-point operations every M-cycle, i.e., it is operating at a rate of 4 MFLOPS. Our simulation showed that about 73% of the time the function units produced results, that is, the APAC was operating at an average of 2.93 MFLOPS for this benchmark. This corresponds to a torque computation (i.e., actuator signals for all six joint motors) in about 250 μs. To achieve this considerable time was spent hand optimizing the program. Scheduling two pipelined function units is time consuming. Support software to help with this aspect of program preparation would be a necessity in a production environment.

COMPUTER VISION - The objective of this research project is to develop an algorithm using grey-scale computer vision techniques to recognize two overlapping objects in their overlapping state and determine the position and orientation of the overlapping objects. More specifically, the task is to detect the outside boundaries of the objects, construct both the overlapping edges and the hidden edges, and determine the position and the orientation of the top object and then the bottom object. The vision algorithm was used to recognize two specific overlapping objects (door latches for automobiles, see Figure 8).

The basic system was divided into two phases. The first phase involved training the computer to recognize each object in its non-overlapping stable states. The second phase involved the actual recognition and reconstruction of the internal edges and the objects in their overlapping state. This second phase was divided into six stages as follows: (1) image acquisition, (2) edge detection, (3) labeling, (4) feature selection and recognition, (5) template matching, (6) objects recognition and reconstruction.

The image acquisition stage converts the analog image obtained from a single T.V. camera into a 256x256x8 bit grey-scale image of the overlapping parts. The edge detection stage converts the grey-scale image into a binary image while locating as many edge pixels as possible. The labeling technique segments the binary image by labeling its various internal and external components. The feature recognition stage searches the labeled image for local features such as holes, and determines the position of their corresponding objects. The template matching stage computes a radial template, compares it with templates generated in the Training phase, and determines the orientation of the objects. And finally the objects recognition and reconstruction stage reconstructs the objects and determines which one is on top. The only requirements are that the edge detection stage generate a "good" internal edge, the objects have some clearly distinguishable features such as holes, and that at least one of these features be non-overlapped. For the objects used in this project, the vision system was unable to recognize only those objects whose internal hole was "hidden"; it could still recognize all of the other objects. However, to accurately determine which object is on top requires all of the holes to be completely visible. A more detailed discussion of the technique can be found in (17).

PROGRAM OBJECTIVES FOR THE NEXT YEAR

Our major efforts will be in the following areas:

- Continue our modeling and computer simulation of the insertion process for various contact configurations and implement the proposed pattern recognition method for force recognition on our PUMA robot arms equipped with wrist force sensors. The objective is to verify experimentally the equations governing the geometric force constraints and the quasi-equilibrium conditions of the insertion process and the use of pattern recognition for recognizing these configurations.
- Continue to investigate the guidance control for insertion process for PUMA robot arms.
- Investigate methodologies directed toward the problems of controlling and coordinating multirobots operating in a cooperative task environment (i.e. within a manufacturing cell). Some of the major problems associated with multirobots in such environment are: synchronizing the distributed processes performed by the multirobots; communication and coordination of the sensory feedback control signals for each robot; real-time processing of these signals for each robot; and the development of a user-oriented high-level concurrent programming language for control of the systems. (see (18) for our preliminary thinking)

REFERENCES

1. J. L. Turney, 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.
2. T. N. Mudge, J. L. Turney, and C. S. G. Lee, "Connection Between Formulations of Robot Arm Dynamics with Applications to Simulation and Control", CRIM Report No. RSD-TR-4-82, April 1982.
3. C. S. G. Lee, B. H. Lee, R. Nigam, "An Efficient Formulation of Robot Arm Dynamics for Control and Computer Simulation," CRIM Technical Report RSD-TR-8-82, University of Michigan, August, 1982.
4. C. S. G. Lee, "Robot Arm Kinematics, Dynamics, and Control," *IEEE Computer*, Vol. 15, Number 12, December 1982, pp 62-80.
5. C. S. G. Lee, M. J. Chung, T. L. Turney, T. N. Mudge, "On the Control of Mechanical Manipulators," *Proceedings of the Sixth IFAC Conference in Estimation and Parameter Identification*, Washington D.C., June 1982, pp. 1454-1459. Also issued as CRIM Technical Report RSD-TR-5-82, the University of Michigan, April 1982.
6. C. S. G. Lee, M. J. Chung, "An Adaptive Control Strategy for Computer-Based Manipulators," *Proceedings of the 21st Conference on Decision and Control*, Dec. 8-10 1982, Orlando, Florida, pp 95-100. Also issued as CRIM Technical Report RSD-TR-10-82, the University of Michigan, August 1982.
7. C. S. G. Lee, C. H. Chen, "A Suboptimal Control Design for Mechanical Manipulators," paper submitted for presentation at the 1983 American Control Conference (ACC), June 22-24, 1983, San Francisco, California.
8. T. N. Mudge, J. L. Turney, "Unifying Robot Arm Control", *Proceedings of the 1982 Annual Meeting of the Industry Applications Society*, October 1982.
9. C. S. G. Lee, S. Nemazie, "Pattern Recognition Methods for Force Recognition in Insertion Processes," Technical report, Center for Robotics and Integrated Manufacturing, the University of Michigan. (in preparation)
10. T. N. Mudge, J. L. Turney, " VLSI Implementation of a Numerical Processor for Robotics," *Proceedings of the 27-th International Instrumentation Symposium*, Indianapolis, Indiana, April 1981, pp 169-175. [Received a Best Paper Award]. Also presented at the Instrument Society of America Anaheim Conference, October 1981.
11. C. S. G. Lee, T. N. Mudge, J. L. Turney, "Hierarchical Control Structure using Special Purpose Processors for the Control of Robot Arms," *Proceedings of 1982 Pattern Recognition and Image Processing Conference*, Las Vegas, Nevada, June 14-17, 1982, pp 634-640.
12. A. K. Bejczy, "Robot Arm Dynamics and Control," Technical Memorandum 33-669, Jet Propulsion Laboratory, Feb. 1974.
13. J. M. Hollerbach, "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-10, No. 11, November 1980 , pp. 730-736.
14. K. S. Fu, "Learning Control Systems --- Review and Outlook," *IEEE Trans. on Automatic Control*, April 1970.
15. 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.

16. J. Sklansky, "Learning Systems for Automatic Control," *IEEE Trans. on Automatic Control*, Vol. AC-11, Jan. 1966.
17. S. Berman, P. Parikh, C. S. G. Lee, "Computer Recognition of Overlapping Parts using a Single Camera," *Proceedings of 1982 Pattern Recognition and Image Processing Conference*, Las Vegas, Nevada, June 14-17, 1982, pp 650-655. Also issued as CRIM Technical Report RSD-TR-6-82, the University of Michigan, April 1982.
18. T. N. Mudge, R. A. Volz, D. E. Atkins, "Hardware/Software Transparency in Robotics through Object Level Design," *Proceedings of the Society of Photo-optical Instrumentation Engineers Technical Symposium West*, August 1982.

CONTACTS

Professor C. S. G. Lee (313)-764-6894
Professor T. N. Mudge (313)-764-0203
Department of Electrical and Computer Engineering
University of Michigan
Ann Arbor, Michigan 48109.

Table 1. Mathematial Operations of the Adaptive Controller Based on Perturbation Equations

Controller based on Perturbation Equations	Multiplications	Additions
Identification Algorithm	$30n^2 + 5n + 1$	$30n^2 + 3n - 1$
Control Algorithm	$8n^3 + 2n^2$	$8n^3 - n^2 - n$
Total Mathematical Operations (excluding nxn matrix inversion)	$8n^3 + 32n^2 + 5n + 1$	$8n^3 + 29n^2 - 3n^2 - n - 1$

Table 2 Comparison of PD and Adaptive Controllers

		PD Controller			Adaptive Controller		
Various Loading Conditions	Joint	Trajectory Tracking Max. Error (degree)	Trajectory Tracking Max. Error (mm)	Final Position Error (degree)	Trajectory Tracking Max. Error (degree)	Trajectory Tracking Max. Error (mm)	Final Position Error (degree)
No load and 10% error in inertia tensor	1	0.089	1.55	0.025	0.038	0.66	0.012
	2	0.098	1.71	0.039	0.043	0.75	0.014
	3	0.328	2.86	0.121	0.068	0.59	0.020
1/2 max. load and 10% error in inertia tensor	1	0.121	2.11	0.054	0.103	1.78	0.102
	2	0.147	2.57	0.078	0.127	2.22	0.127
	3	0.480	4.19	0.245	0.095	0.82	0.095
Max. load and 10% error in inertia tensor	1	0.145	2.53	0.082	0.121	2.11	0.018
	2	0.186	3.23	0.113	0.253	4.41	0.252
	3	0.807	5.30	0.360	0.269	2.52	0.181

Figure 1 Adaptive Control Block Diagram

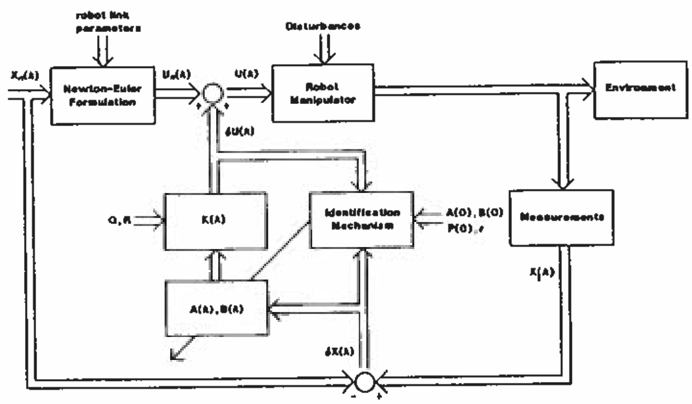

Figure 2 Position Errors of PD Controller

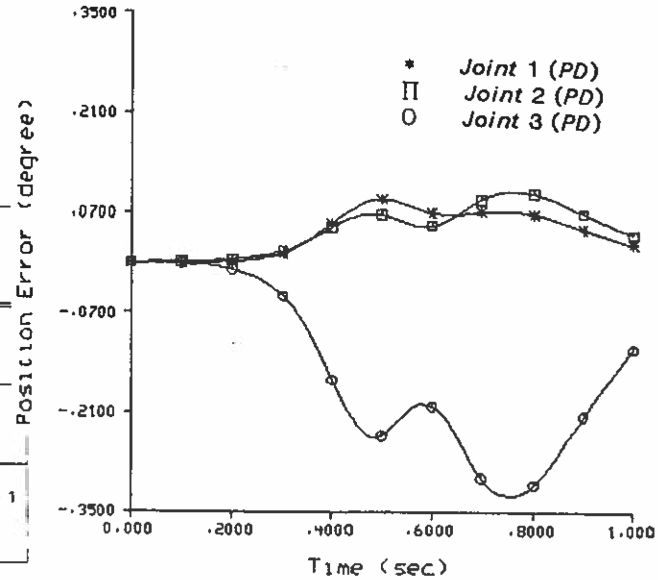

Figure 3 Position Errors of Adaptive Controller

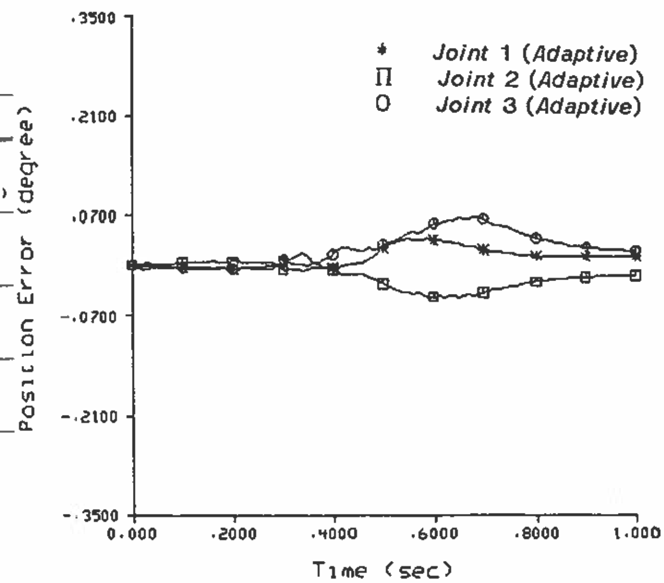

Figure 4 Torques Computed From Adaptive Controller

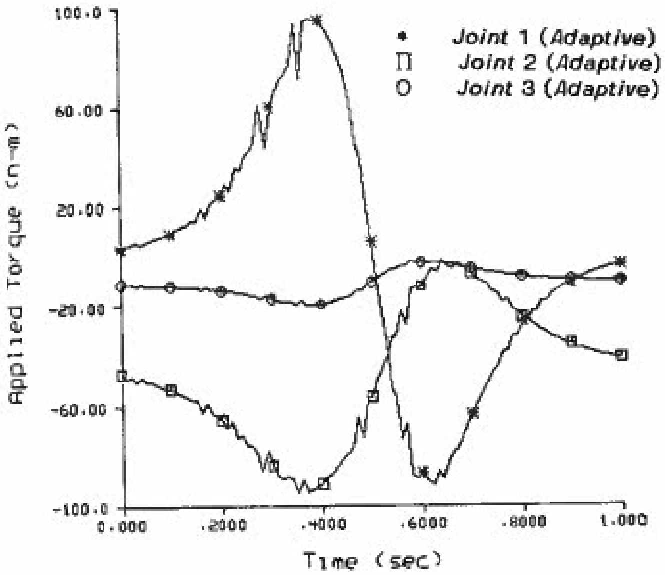

Figure 7 The Proposed APAC Block Diagram

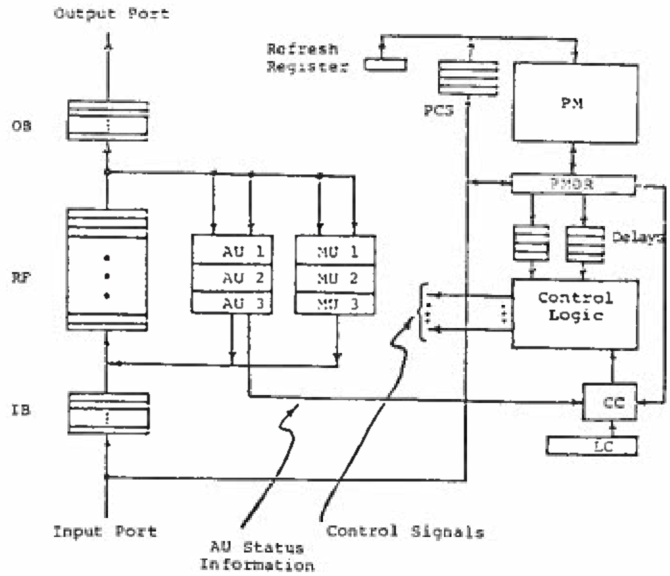

Figure 5 Nonlinear Feedback Control Block Diagram

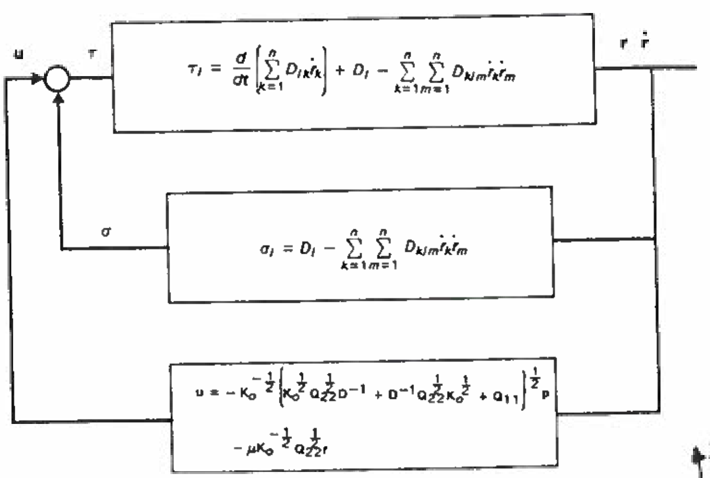

Figure 8 Objects for Computer Vision

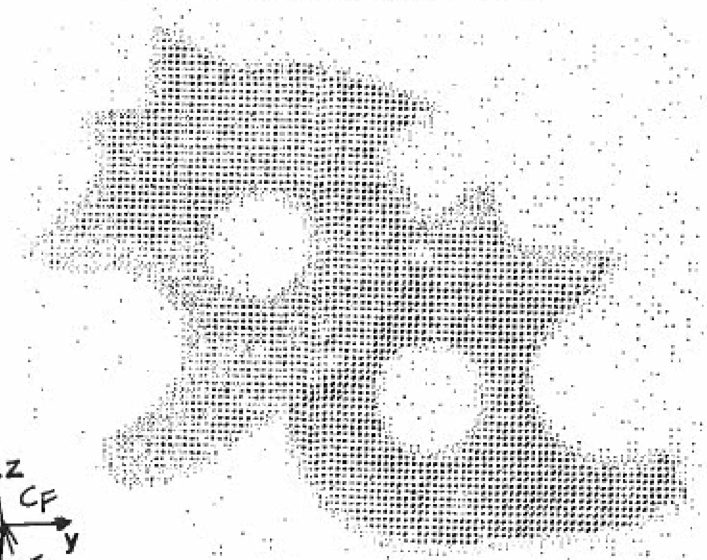

Figure 6 A Control Situation for Insertion Process