N94-14651

Design and Control of a Macro-Micro Robot for Precise Force Applications

Yulun Wang, Amante Mangaser, Keith Laby, Steve Jordan, Jeff Wilson

Computer Motion, Inc.
Goleta, CA. 93117

Abstract

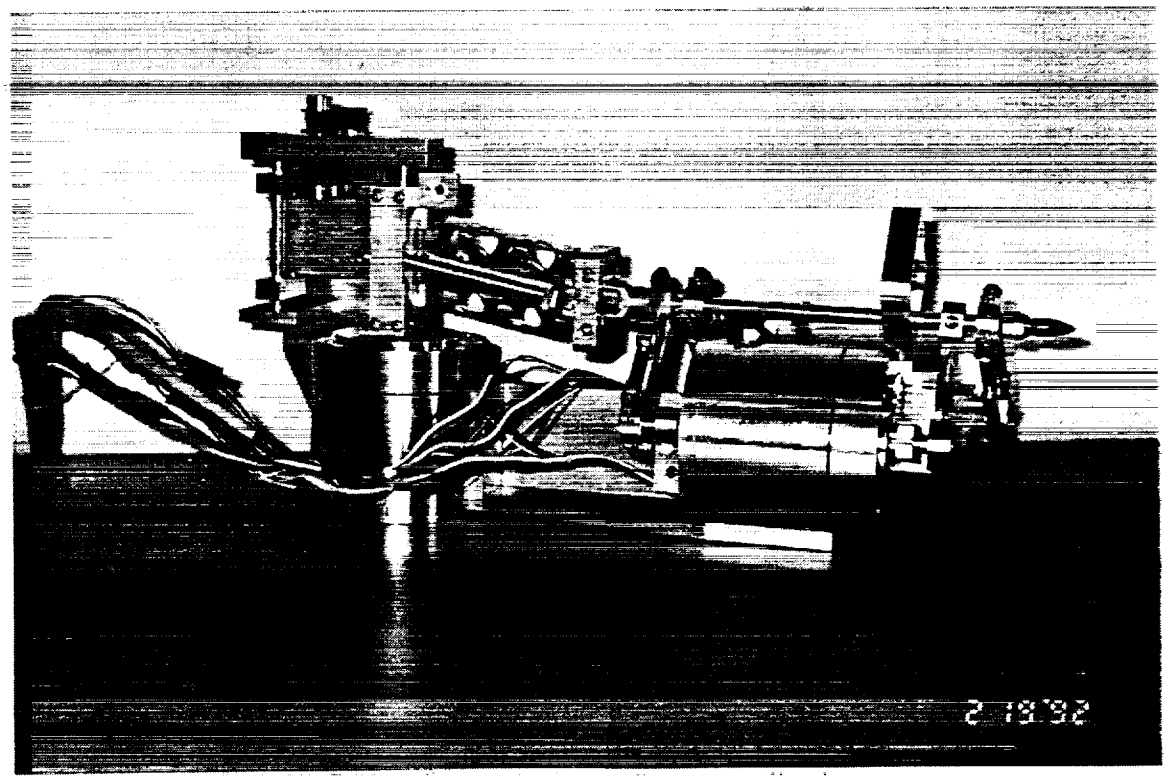
Creating a robot which can delicately interact with its environment has been the goal of much research. Primarily two difficulties have made this goal hard to attain. The execution of control strategies which enable precise force manipulations are difficult to implement in real time because such algorithms have been too computationally complex for available controllers. Also, a robot mechanism which can quickly and precisely execute a force command is difficult to design. Actuation joints must be sufficiently stiff, frictionless, and lightweight so that desired torques can be accurately applied.

This paper describes a robotic system which is capable of delicate manipulations. A modular high-performance multiprocessor control system was designed to provide sufficient compute power for executing advanced control methods. An 8 degree of freedom macro-micro mechanism was constructed to enable accurate tip forces. Control algorithms based on the impedance control method were derived, coded, and load balanced for maximum execution speed on the multiprocessor system. Delicate force tasks such as polishing, finishing, cleaning, and deburring, are the target applications of the robot.

1.0 A Force Controllable Manipulator

A manipulator capable of delicate interactions with its environment must be designed differently from today's position controlled robots. It has been shown that a high-bandwidth, low effective end-effector inertia design is helpful for precise force control [1,2]. There are two design approaches to creating such a structure. One is to design the manipulator so that the entire structure is very light. This approach can be very costly since expensive materials and tight tolerances are required. The other approach is to attach a low-inertia small manipulator to the end of another larger and heavier manipulator. This macro-micro structure results in a combined structure with the low end-effector inertia of the micro robot and the large workspace of the macro robot.

Our macro-micro design couples a 3 degree of freedom micro robot to the end of a 5 degree of freedom macro robot. A schematic and photograph of the micro design is shown in Figure 1, and a schematic of the macro design is shown in Figure 2. For the micro robot, the x and y directions are actuated with a parallel set of 5-bar-link mechanisms, one attached to each side end of the two motor shafts. The z motion is actuated by a fixed motor oriented perpendicular to the x and y motors. This motor is attached to the parallel link mechanism through a pair of universal joints. The range of motion is 2 centimeters along each axis. A fourth pneumatic motor, located furthest from the tip, rotates the tip through a series of transmissions at a constant speed for polishing, finishing, and grinding applications.



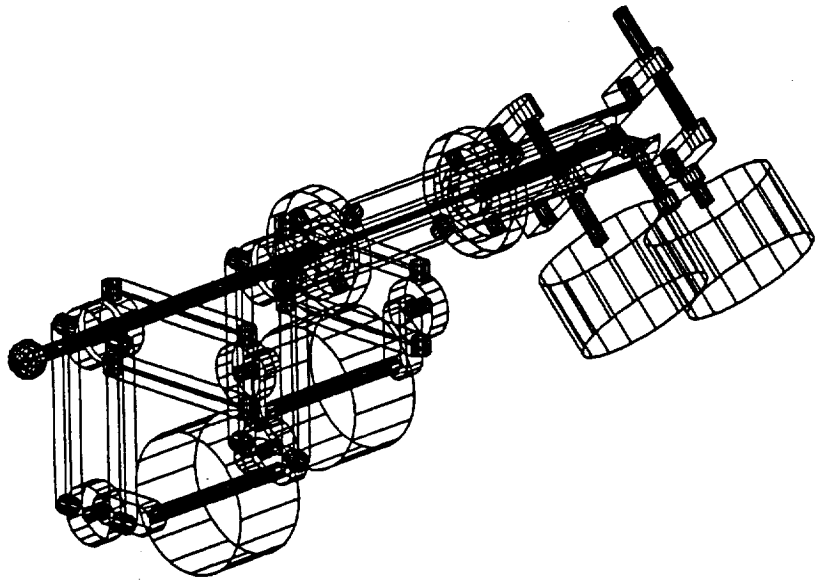


Figure 1. The Micro Manipulator

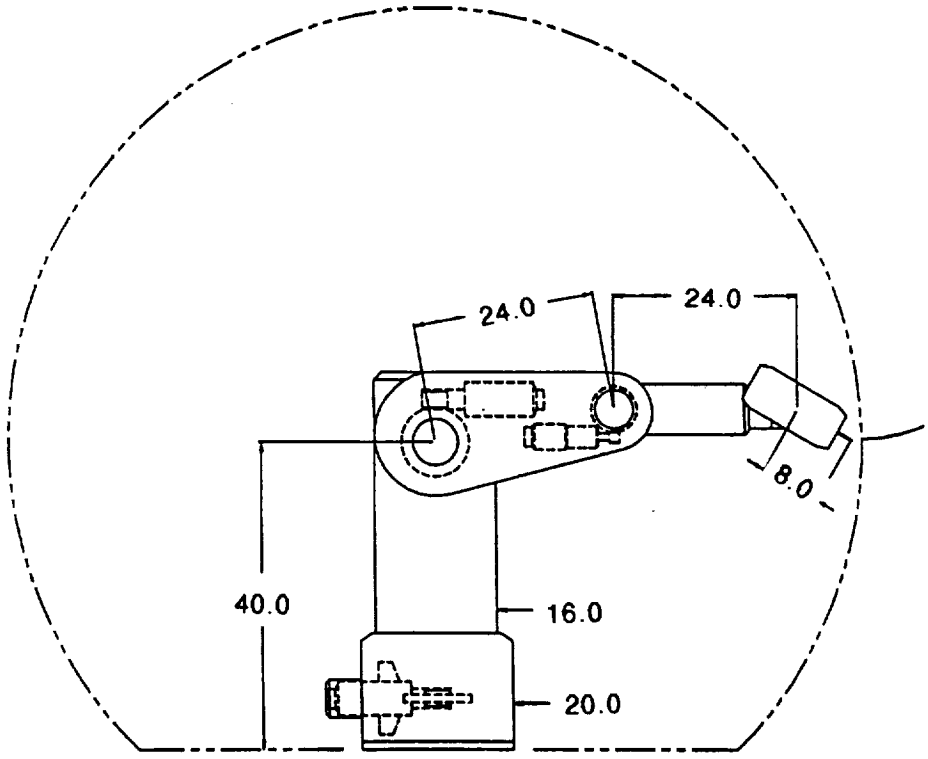


Figure 2. The Macro Manipulator

Since the macro and micro robots coordinate as a single system, many tradeoffs influence both designs. For example, the size of the micro robot's workspace influences the accuracy with which the macro robot must be able to position itself. The mass of the micro robot also influences the payload capability of the macro design. Our design strategy was to simplify the macro design by making the micro robot more capable. The main consequence of this decision is a large micro workspace, thereby allowing less accuracy and performance in the macro. However, the micro's workspace volume directly influences the overall mass and size of the design considerably. In our design, reducing travel along each dimension by a factor of two roughly reduces the size and mass of the micro robot by a factor of two.

The main objectives of the micro design were to minimize end-effector inertia, minimize joint friction, maintain tip orientation throughout the workspace, and support a maximum payload (i.e. force exertion) of 3 kilograms. The resulting tip inertia is roughly 250 gms. The joint friction was minimized by using direct-drive transmission and limited angle flex bearings at the joints. These limited-angle bearings offer virtually no friction. They do generate a spring force, however, which must be compensated for in the control law. Tip orientation is maintained by the parallel 5 bar link structure.

Secondary goals were to minimize the size and weight of the micro-manipulator. The final size is 35.5 by 19 by 17.8 centimeters, and the weight is 6.3 kilograms. Strain gages mounted on the links provide sensing for 5 degrees of freedom (as shown in Figure 1). Sensors for detecting a moment about the tip axis were not included.

1.1 Micro kinematics

The micro robot is a closed-chain kinematic structure with 3 servo actuators to produce translational motion in 3 space while maintaining constant orientation. To derive the kinematic equations we only need to consider the tip's position. This positioning mechanism can be thought of as three links which are all connected by ball joints at one end. The other ends of each link are connected to separate actuators, each through a universal joint. Since the joint angle of the actuators are always known, the end position of each link is known, and will be referred to as P1, P2, and P3 as shown in Figure 3.

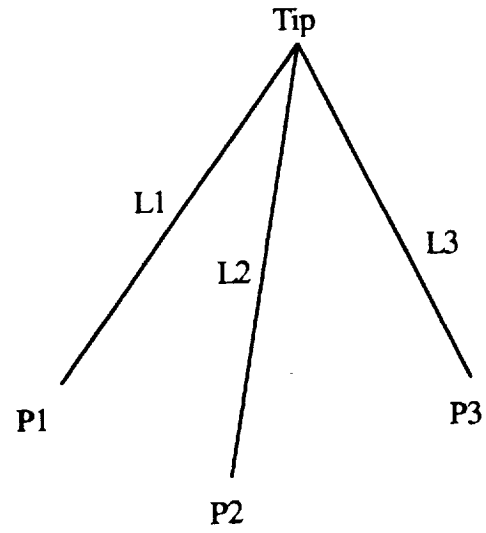


Figure 3. Kinematic Model of Micro Robot

Three simple geometric relations which can be observed are as follows:

$$| \text{Tip} - \text{P1} | = \text{L1}$$
$$| \text{Tip} - \text{P2} | = \text{L2}$$
$$| \text{Tip} - \text{P3} | = \text{L3}$$

where L1, L2, and L3 are the lengths of the 3 links. The forward kinematics problem is to solve for the Tip position given L1, L2, L3, P1, P2, and P3. P1, P2, and P3 can easily be related back to the joint angles. This solution may be visualized geometrically by imagining three spheres with radius L1, L2, and L3 centered at P1, P2, and P3 respectively. The point where all three spheres intersect is the robot's tip position.

Solving these equations for the Tip position is possible. However, the solution is a very large and complex high order polynomial. Since it is important that the micro robot is servoed at a high rate, it was necessary to develop a more computationally efficient approximate solution to the forward kinematics.

If we assume that the links rarely rotate beyond 10 degrees from the center position, we can derive a fairly accurate and simple forward kinematics solution. The angles α and β, shown in Figure 4, describe the orientation of each link and can be used to approximate the manipulator's tip position with the following equations:

$$\text{Tip}_x = A_1 \sin\theta_1 - \text{L1} [(1 - \cos\alpha_1) + (1 - \cos\beta_1)]$$
$$\text{Tip}_y = A_2 \sin\theta_2 - \text{L1} [(1 - \cos\alpha_2) + (1 - \cos\beta_2)]$$
$$\text{Tip}_z = A_3 \sin\theta_3 - \text{L1} [(1 - \cos\alpha_3) + (1 - \cos\beta_3)]$$

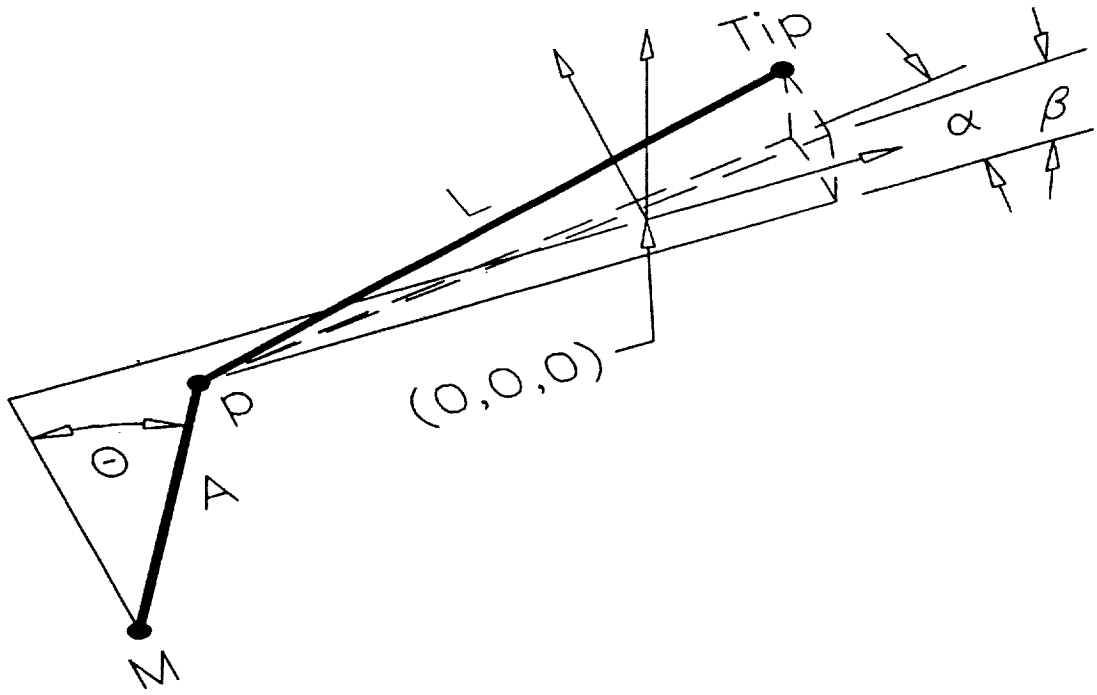


Figure 4. Geometric quantities for one branch of the micro robot

These equations are simple enough to be computed quickly so that high speed servoing can be achieved. This solution is very accurate at the center of the workspace, and error increases towards the edge of the workspace. The influence of this approximation error on the impedance control law is quantified in section 3.0.

1.2 Micro dynamics

In order to calculate the closed-form dynamic equations of the micro mechanism, we assumed a rigid body model which includes 3 independent motor inertias, and a lumped mass at the tip with different masses along each cartesian axis. Since the micro robot is carried about by the Macro robot, it is important to include the macro's motion into the dynamic equations. By using Lagrange's equation we derived the following equations of motion:

$$\tau = \vec{\dot{\theta}}_I^T \left(J_I^T M_T J_I + I_M \right) + \vec{\ddot{\theta}}_I^T \left(J_I^T + 2 J_I^T \dot{R}^T R \right) M J_I$$
$$+ \vec{X}_I^T \left(K_F + \left(\ddot{R}^T R - 2 \dot{R}^T \dot{R} \right) M_T \right) J_I + \vec{g}^T R (M_T + M_z) J_I + \vec{F}_x^T + \vec{\ddot{X}}_A^T R M_T J_I$$

where

$\vec{\theta}_I^T$ = joint angles of Micro robot

J_I^T = jacobian of Micro robot

M_T = tip mass of Micro robot

R = rotational matrix from Micro coordinate system to Macro coordinate system

$\vec{F}_x^T$ = force exerted at tip of Micro

K_F = force created by flex bearings in Micro joints

$\vec{X}_A$ = tip position of Macro robot

$\vec{X}_I$ = tip position of Micro robot

M_Z = added mass felt by z-axis motor

The macro design is a 5 degree-of-freedom articulated manipulator, as shown in Figure 2. This manipulator supports the weight and continuous force exertion capability of the micro-manipulator throughout the workspace with 1g acceleration. A 1 meter reach was chosen as a reasonable workspace. The main features of this design are high mechanical rigidity, simple kinematics, large workspace volume, and cost effectiveness.

The 5 degree of freedom kinematic structure is very similar to the first five joints of a PUMA robot [3]. A 6th joint is unnecessary because the tip of the micro robot spins continuously. Link offsets, link lengths, and structural characteristics were designed to account for the size and mass constraints imposed by the micro-manipulator, however.

We considered a variety of actuation methods. Options which were considered were direct-drive, harmonic drive, spur gear, worm gear, planetary gear, and different combinations of these. The goal was to maximize accuracy, resolution, and stiffness while staying cost effective. After various optimization procedures we decided on a harmonic drive - worm gear double reduction scheme for the first three joints. The last two joints, which carry a much smaller load, use harmonic drives.

The procedure for solving for the inverse kinematics equations of this robot is very similar to that of the PUMA robot and can be found in many of different robotics textbooks [4]. The kinematics and dynamic equations used for computed torque control can also be derived very easily using of the generalized formulations which have been developed [5]. However, because of the high reduction ratios of the transmissions, independent joint control is adequate.

2.0 A Modular Multi-processor Control System

A high performance multiprocessor system is used to satisfy the significant computational demands of controlling this robot. We designed this control system as a general purpose high performance controller with both hardware and software modularity as a key feature. The ability to easily rearrange and extend hardware and software modules to support different requirements for various tasks is particularly important in experimental projects such as this. Frequently designs are unable to accomodate even minor modifications without a major impact to the existing system configuration.

A schematic of the motion control system configuration is shown in Figure 5. The four basic units are the compute unit, the global memory unit, the position, velocity and digital I/O unit, and the A-to-D D-to-A unit.

The compute unit is based on Texas Instrument's TMS320C31 floating-point digital signal processor. In our earlier generation systems [6,7], we used a novel 3D computing processor which proved to offer much higher performance than DSPs or RISC processors on kinematic and dynamic calculations. However, due to the high cost of implementing this design using discrete datapath parts we opted to used an off-the-shelf processor. At a crystal speed of 33Mhz the TMS320C31 offers 33 MFLOPS of peak power. Each unit contains 2 Mbyte of program memory, 2 Mbyte of data memory, 2 programmable timers, interrupt capabilities for both the I/O Bus and the VME bus, and bus arbitration logic for accessing the I/O Bus. The memory is directly accessible by the host computer over the VME bus. Different levels of concurrency are provided to maximize execution speed. For example, the host may access data memory while the processor continues program execution. Programs are developed in either C or C++ on the host computer and downloaded to the appropriate unit before run time. Several libraries are provided to support program development. Remote procedure calls were provided so that UNIX services, such as printf(), scanf(), open(), and close(), are available for code development. Math functions, functions for accessing sensory data, and message passing functions for multi-processing are also provided.

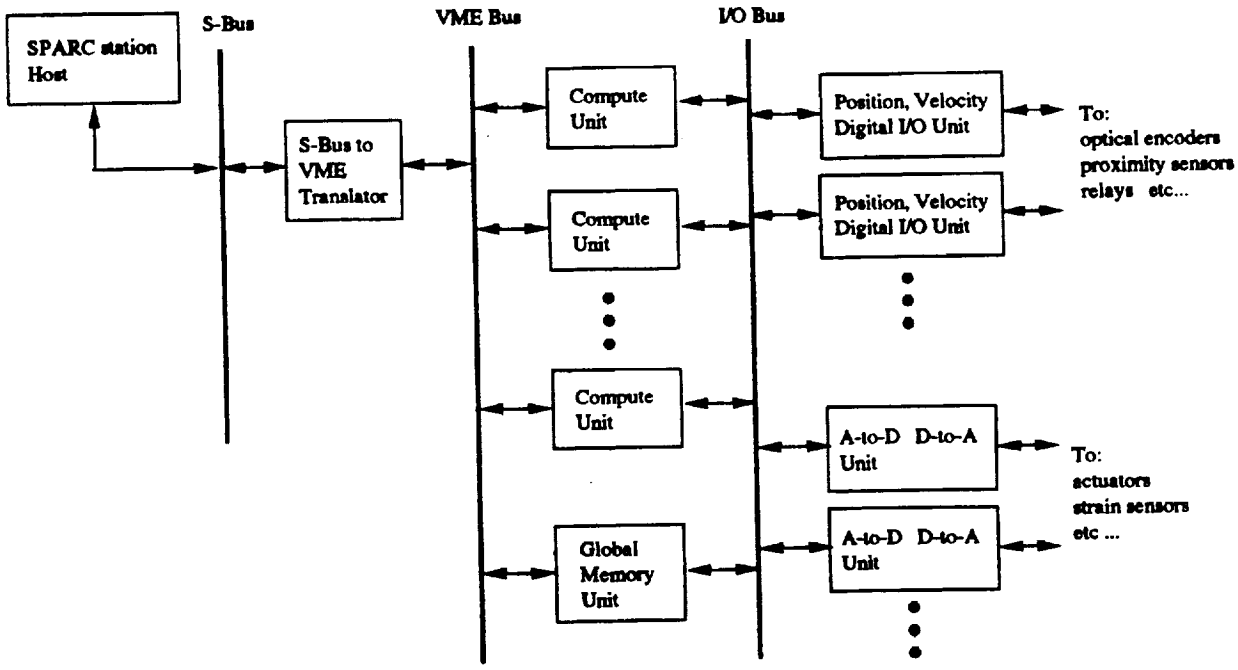


Figure 5. The Motion Control System

The global memory unit contains 2 Mbytes of memory for passing messages between compute units, to and from the host, and to store global variables shared by multiple compute units. A mailbox message passing scheme is implemented to support multiprocessor communication. Information is passed from one compute unit to another compute by first acquiring the IO Bus, then writing the message into the target compute unit's mailbox, and then interrupting the target compute unit. The target compute unit reads its mailbox, and sends an acknowledgement to the sending compute unit. Hardware interlocking and interrupt mechanisms are included to achieve high bandwidth communication. Reading or writing a message requires ~3 μs overhead and another 180ns for each 32-bit word.

The position, velocity, and digital I/O unit accepts 6 channels of 2 channel quadrature encoder input and translates that into absolute position and velocity. Each channel also supports index pulse detection, which is generally used for position homing. Position is stored to 24-bit accuracy and velocity is stored to 10-bit accuracy. Thirty-two bits of digital input and 32 bits of digital output are included for instrumenting relays, proximity sensors, or other on-off type devices.

Velocity is generated by two different schemes, depending on the velocity range. At low speeds, velocity is generated in hardware by a free running counter which measures time between successive encoder counts. At high

speeds, velocity is determined by calculating the number of encoder counts which have passed during the previous sample period. For each velocity read operation, the software automatically chooses between the two schemes by reading the velocity counter and comparing it with a threshold value. The result of this method is a more accurate velocity signal with minimized quantization effects.

Velocity is generated in hardware from the optical encoder signal by incorporating a free running counter chip which calculates the time between successive encoder pulses. Velocity is usually derived from a quadrature signal by subtracting the current position with the previous sample period's position. This subtraction may result in very quantized velocity signals especially at high sample rates, however. The hardware counter method produces a much more finely resolved velocity signal. There is still a problem, however, since at low speeds there may be significant time delay between new velocity acquisitions.

The A-to-D D-to-A unit provides 9 channels of 12-bit digital-to-analog output, and 8 channels of 12-bit analog-to-digital input. Separate digital to analog converters are provided for each output channel. A single analog-to-digital converter is multiplexed between the 8 input channels. Each channel requires 3 μs of conversion time. Software routines are provided to configure the card to only sample the channels which are in use. Conversion is performed continuously and asynchronously only on the channels being used. Therefore, the maximum delay from when the data was acquired to when it was read is 3 μs $\times$ number of selected channels.

The software structure of the operating system level software is shown in Figure 6. Note that there is a clear separation between the real-time execution environment and the non-real-time UNIX environment. The UNIX environment is used for program development, user interface, and monitoring the real-time system. Because of the UNIX front-end, the robot interface must be carefully constructed such that the integrity of the real-time system is not lost. For example, UNIX service requests by the real-time system cannot be made while servoing since a real-time response from the UNIX process cannot be guaranteed.

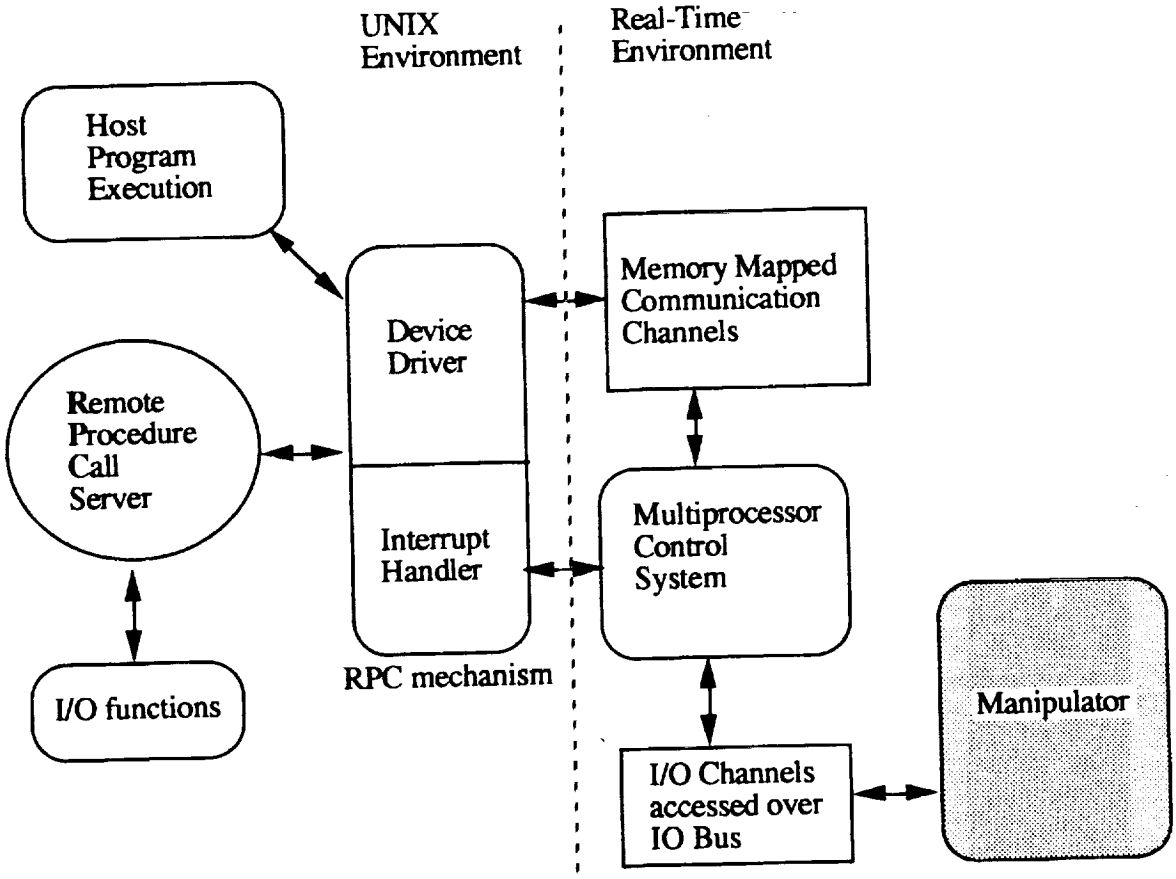


Figure 6. System Software Structure

Figure 7 shows the general hierarchy of the application software of the system. Macro calls provide fast access to the various hardware features of the system. C language routines provide the next layer, which support functions such as synchronizing multiple processes, remote procedure calls to the host, and algorithms for performing mathematic operations. At the highest level, object-oriented class libraries are supported in C++.

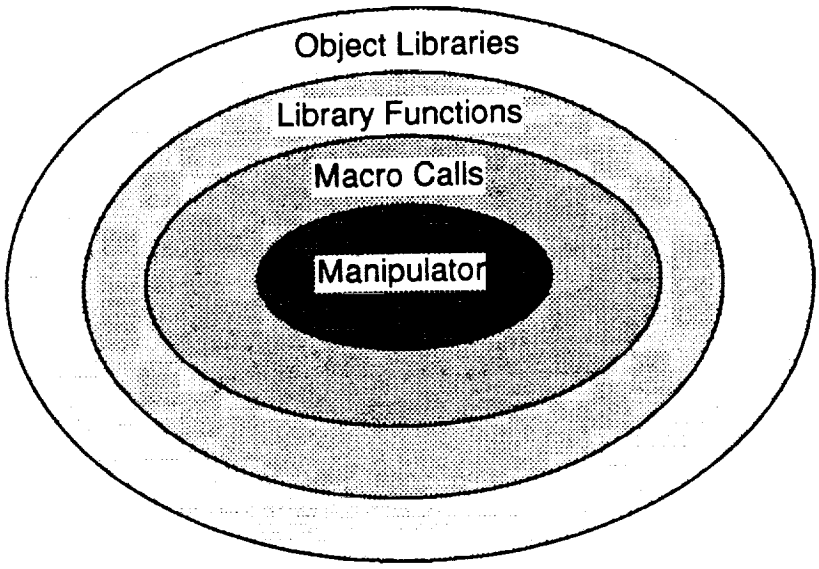


Figure 7. Software Support for Application Development

3.0 Impedance Control for a Macro-Micro Robot

The impedance control method enables a robot to interact with its environment in a well controlled and precise manner [8]. The manipulator's end-effector reacts to environmental disturbances in the same manner as a linear mass, spring, damper system. The mass, spring, and damper values are controlled electronically and can be different along different axes, and can continuously change during a trajectory.

This method is different from hybrid position/force control [9] since specific forces or positions are never specified. The control variable is the equilibrium point of the mass, spring, damper system without external forces. The advantage of this methodology is that a single control variable and control algorithm can be used to guide a robot through interactions with the environment. Hybrid position/force control, on the other hand, requires a switch in control methods and control variables whenever the robot changes the configuration in which it interacts with its environment.

Figure 8 gives an example of a trajectory specified by the equilibrium path where the manipulator comes into contact with a surface, slides across it, and then leaves the surface. Note that the nominal force exerted on the surface is proportional to the spring constant. By using the spring constant and surface location information, it is simple to calculate the equilibrium point's trajectory to produce a desired force across the surface. The force at the contact point will be influenced by contributions due to the mass and damper as well. Consequently, if precise force control is important, the smaller the mass and damper values are the better. The macro-micro design facilitates small mass values.

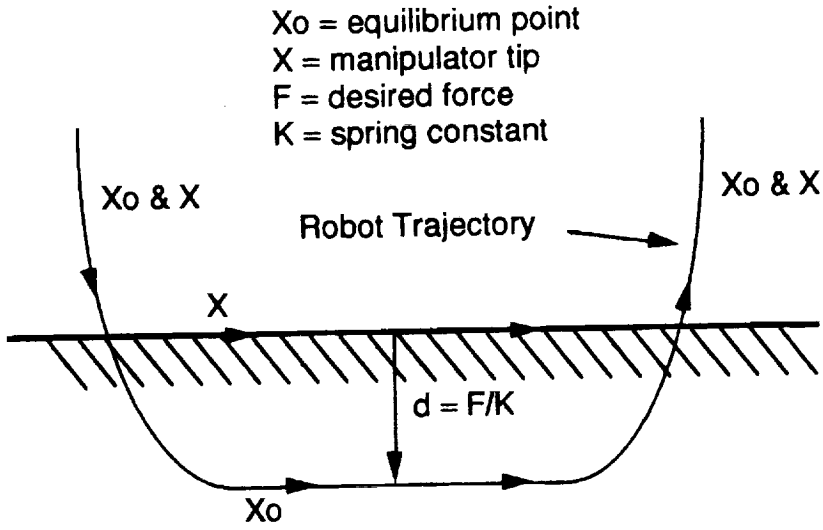


Figure 8. Manipulator trajectory specified by equilibrium point

The impedance equation can be written as follows:

$$F_{ext} = M_s(\ddot{X}_R - \ddot{X}_o) + C_s(\dot{X}_R - \dot{X}_o) + K_s(X_R - X_o)$$

where

F_{ext}	external force applied to robot tip
X_R	tip position of macro-micro robot
X_o	desired equilibrium point of macro-micro robot
M_s	desired mass constant
C_s	desired damper constant
K_s	desired spring constant

Impedance control of a macro-micro design has the added complexity of managing the manipulator's redundancy to optimize force interactions by exploiting the micro robot's low tip inertia. In other words, the redundancy should be used to keep the micro robot from reaching its workspace limit, where one or more degrees of freedom would be lost. Our robot has 3 degrees of redundancy along the translational axes. Delicate interactions for translational motion is possible because of the micro robot. Orientation is left to the macro robot and is position controlled.

A block diagram of the control structure is shown in Figure 9. The impedance control law, which outputs torques to the micro robot, is derived by combining the desired impedance equation stated above with the equations of motion of

the micro robot presented in section 1.2. Note that the servo control law for all 5 joints of the macro robot is a simple position controller without feedback from the micro robot. However, feedback from the micro robot is input into a real-time trajectory generator for the macro robot. This trajectory generator uses the robot's redundant degrees of freedom by constantly updating the macro robot's desired position such that the micro robot is centered in its workspace, and hence far from its workspace boundary. Consequently, entire manipulator can respond to external disturbances with the quick reaction of the micro robot over the entire workspace of the macro robot.

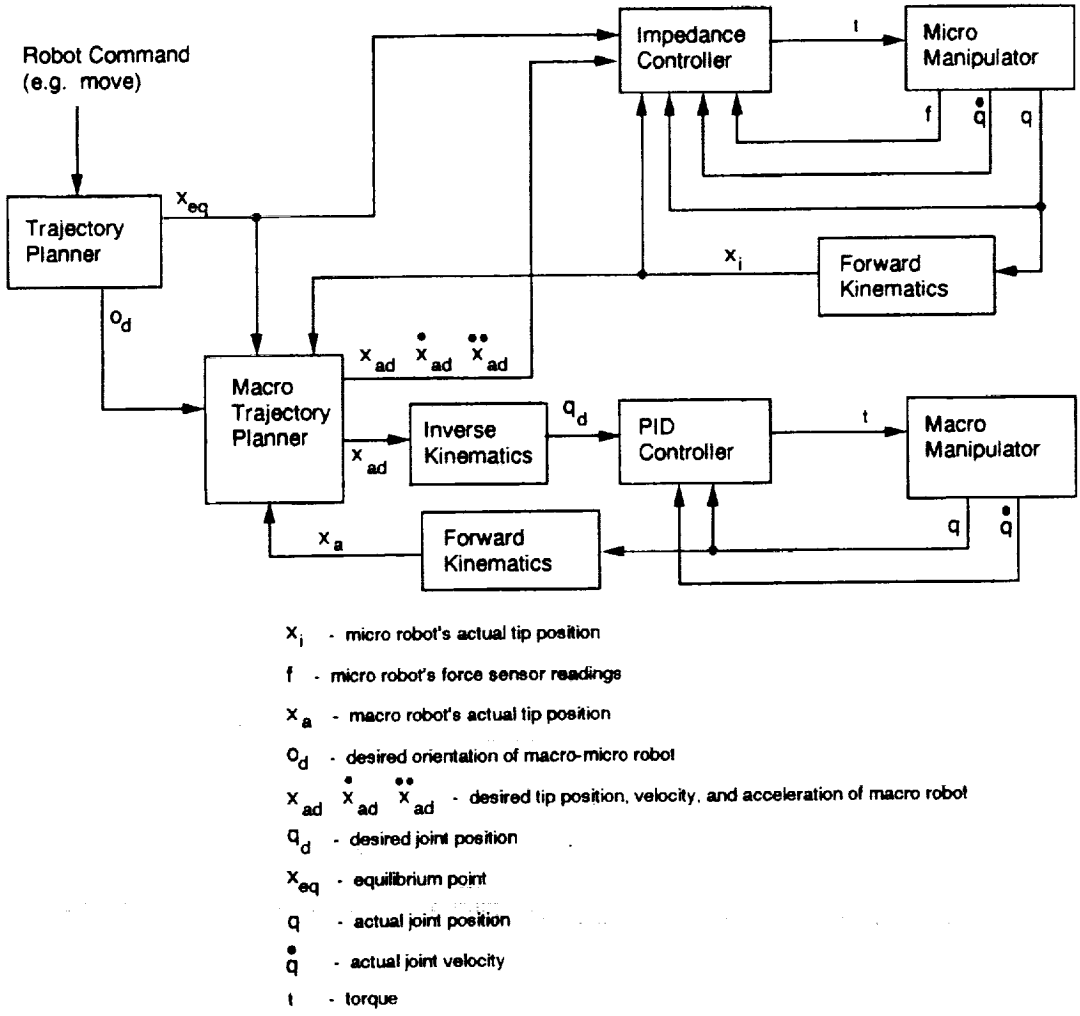


Figure 9. Impedance control of macro-micro robot

The maximum distance which the micro will deviate from its center position is a relationship which includes the ratio of the maximum accelerations of the macro and micro, the magnitude and time of the maximum disturbance, and the reaction time of the servoing system. This information is important since it quantifies the critical tradeoffs between the micro's performance versus the macro's performance. We will obtain more insight into these relationships through experimentation of the robot.

With this control strategy, since the macro robot is purely position controlled it may be possible to apply this strategy to a micro connected onto the end of a commercial robot. However, the success of this approach is dependant upon the ability of the commercial robot to accept and quickly respond to new position commands. The requirements of a commercial robot used in this manner will become clearer with more experimentation on our robot.

In order to quantify the errors in the impedance control law resulting from the approximations used in section 1.1 we evaluated the control law at several different robot configurations and compared the resulting torques with torques calculated using exact equations. Nominal values of desired mass, spring, damper were chosen, and the micro's workspace center was used as the equilibrium point. To represent the results in a more intuitive framework we multiplied the final joint torques with $(J^T)^{-1}$ to produce a single force vector. The errors are quantified by the percentage different in magnitude, and the orientation different in degrees, from the force vector produced by the exact method. Figure 10a confirms that error is small at the workspace center and increases as we move towards the edge. Figure 10b shows the resulting error at different configurations and includes non-zero velocity terms. Note that in all of these cases the error in the force vector never exceeds 2 percent in magnitude and 1 degree in orientation.

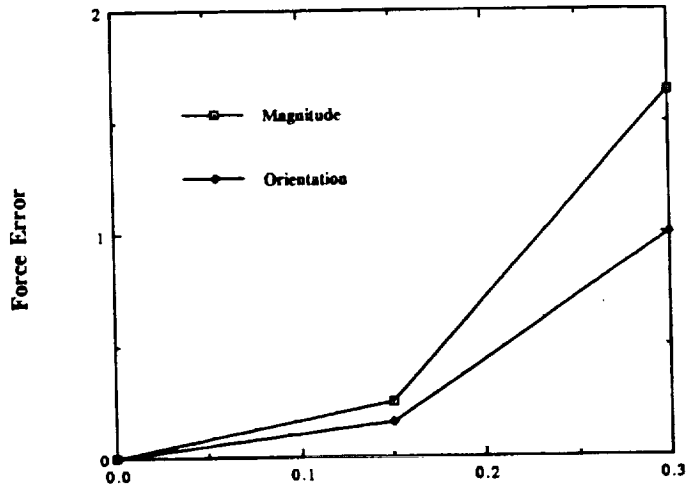
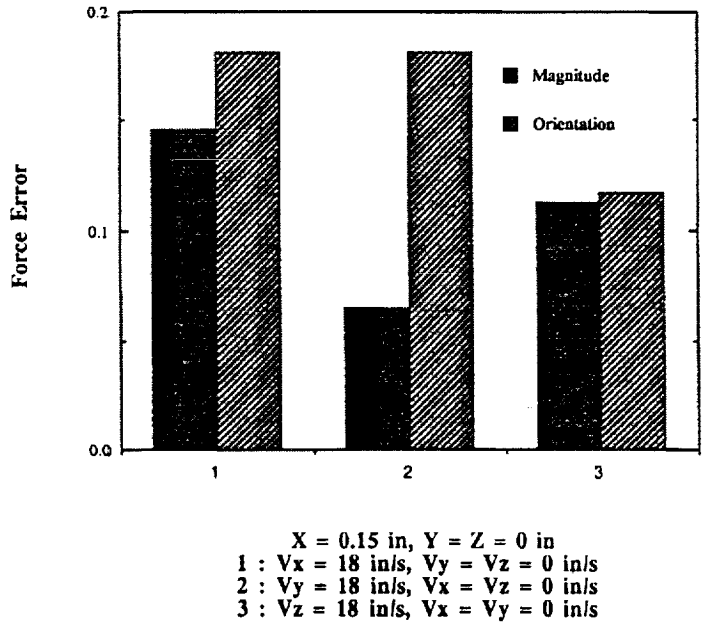


Figure 10a. Force error w.r.t. Tip offset along X-axis (velocity = 0.0, $F_x = F_y = F_z = 4$ lbs)



X = 0.15 in, Y = Z = 0 in
1 : Vx = 18 in/s, Vy = Vz = 0 in/s
2 : Vy = 18 in/s, Vx = Vz = 0 in/s
3 : Vz = 18 in/s, Vx = Vy = 0 in/s

Figure 10b. Force error with non-zero tip velocity

4.0 Control Implementation on a Multiprocessor controller

The hardware control system which we assembled for executing the impedance control method includes 4 compute units, 2 position velocity units, 2 analog and digital units, and one global memory unit. One of the compute units is used for a trajectory planner. The execution of the control algorithms are mapped across the other 3 compute units to optimize the response of the system. Even though there is a great deal of research on automating the processing partitioning a task for parallel processing , our approach has been to manually partition the problem optimizing for balanced loads and meeting the various real-time constraints.

In addition to parallel processing with multiple compute units, within compute units we nested algorithms so that variables which changed faster were evaluated more frequently than variables which changed slower. For example, calculations which include force readings are grouped together and calculated at the highest update rate, and calculations which change only with position are grouped together and calculated at a slower update rate. Figure 11 shows the update hierarchy employed in our scheme.

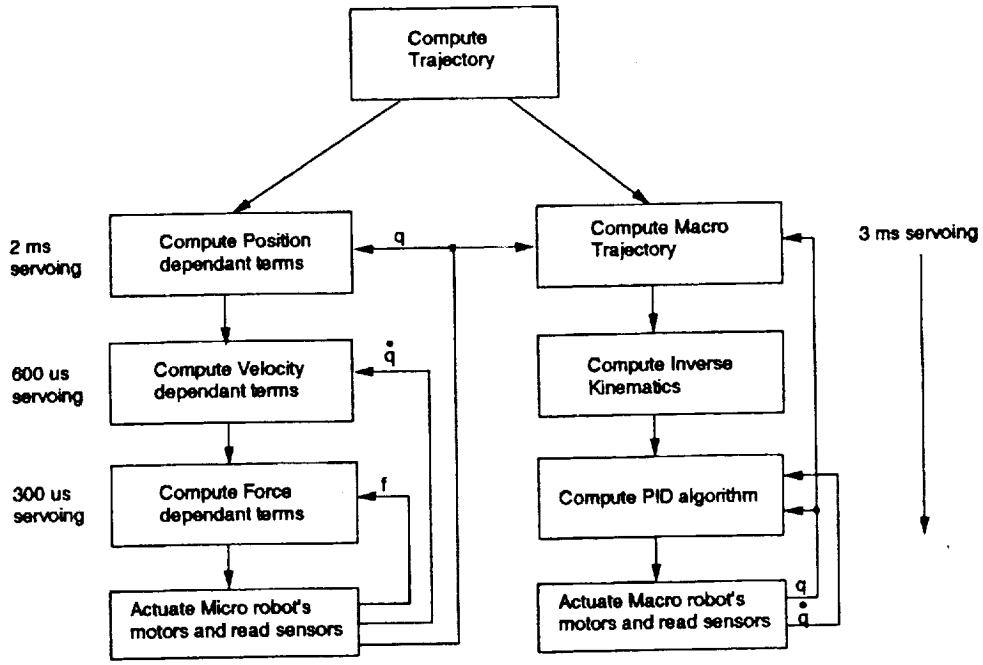


Figure 11. Hierarchical update rates

5.0 Conclusion

Building a robot which can interact delicately with its environment is a challenging task. This paper describes a robotic system designed for such tasks. An 8 degree of freedom macro-micro manipulator is controlled by an impedance-based controller, executed on a high performance multiprocessor control system. The manipulator's tip inertia is very low and can therefore react quickly to force disturbances. The control method compensates for manipulator dynamics, and can generate very precise torques. The multiprocessor offers sufficient compute power to meet the real-time demands of the control strategy. Preliminary results show that this design will be capable of precise force control. More conclusive experimental results will be available by the end of the year.

Acknowledgements

The authors would like to thank Kevin Schantz for his assistance in implementing and debugging the control software. We would also like to thank Professor Yoshihiko Nakamura for his guidance in developing the control algorithm. This work was performed under funding from the Small Business Innovative Research Program from both NASA and the National Science Foundation.

References

[1] Khatib, Oussama, "Augmented Object and Reduced Effective Inertia in Robot Systems," *Proc. of the American Control Conference*, Atlanta, Georgia, June 1988.

[2] Sharon, Andre, Neville Hogan, and David E. Hardt, "High Bandwidth Force Regulation and Inertia Reduction Using a Macro/Micro Manipulator System," *Proc. of the IEEE Conf. on Robotics and Automation*, Philadelphia, Penn., April 1988.

[3] Leahy, M.B. and et. al., "Efficient Dynamics for the PUMA-600," *Proc. of the IEEE Conf. on Robotics and Automation*, San Francisco, CA., 1986.

[4] Wolovich, William A., *Robotics: Basic Analysis and Design*, Holt, Rinehart and Winston, New York, 1987.

[5] Nakamura, Yoshihiko,"Unified Recursive Formulation of Kinematics and Dynamics of Robot Manipulators," *Proc. of the Japan - USA Symposium on Flexible Automation*, Osaka, Japan, July 14-18, 1986.

[6] Wang, Yulun, Amante Mangaser, Steven Butner, Partha Srinivasan, and Steve Jordan, "The 3DP: A Processor Architecture for 3-Dimensional Applications," *IEEE Computer*, January 1992.

[7] Wang, Yulun and Steven E. Butner, "RIPS: A Platform for Experimental Real-Time Sensory-based Robot Control", *IEEE Transactions on Systems, Man, and Cybernetics*, vol 19, no 4, July/August 1989, pp 853 - 860.

[8] Hogan, Neville, "Stable Execution of Contact Tasks Using Impedance Control," *Proc. of IEEE Int. Conf on Robotics and Automation*, Raleigh, North Carolina, 1987.

[9] Raibert, M. and J. Craig, "Hybrid Position/Force Control of Manipulators," *Journal of Dynamic Systems*, Measurement, and Control, vol. 102, pp. 126-133.