A REAL-TIME CONTROL SYSTEM FOR A MOBILE REDUNDANT 7DOF ARM

David Lim, Homayoun Seraji
Jet Propulsion laboratory
California Institute of Technology
Pasadena, CA 91109

Abstract

This paper describes the design and implementation of a real- time control system with multiple modes of operation for a mobile redundant dexterous manipulator. The manipulator under study is a seven degree of- freedom arm from Robotics Research Corporation, mounted on a motorized platform. The manipulator-plus- platform system has two degrees- of- redundancy for the task of hand placement and orientation. The redundancy resolution is achieved by accomplishing two additional tasks using the configuration control technique. The system allows a choice of arm angle control or collision avoidance for the seventh task, and platform placement or elbow angle control for the eighth task. In addition, joint limit avoidance tasks are automatically invoked when any of the joints approach their limits. The system is robust to singularities, and also provides the capability of assigning weighting factors to end- effector and redundancy resolution tasks. The motion control algorithms arc executed at $2.5ms$ on two MC68040 processors in a VME-bus environment running the Vx Works real- time operating system. The paper describes the hardware and software components of the VM E environment. Experimental results on real-time control of the Robotics Research arm using the manipulator control system are also presented in the paper[1].

1 Introduction

In October 1990, the National Aeronautics and Space Administration (NASA) initiated a research and development project on supervised robotic inspection [3] at the Jet Propulsion Laboratory (J PI,). An essential component of a remote inspection system is a dexterous robotic manipulator for sensor placement and the associated control system. This paper describes the design and implementation of a multi- mode real-time control system for a mobile dexterous manipulator used in the supervised inspection project,

The paper is structured as follows. In Section 2, the hardware used by the manipulator control system is described. The control system is discussed in Section 3. In Section 4, the graphical user interface to the system is outlined. In Section 5, we present the software architecture of the VM E environment used for real- time control of the manipulator. Experimental results on real- time control of the dexterous manipulator are presented in Section 6. Section 7 draws some conclusions from this work and discusses the directions of future research and development.

2 Hardware System Description

In this section, we describe the hardware of the manipulator control system (MCS). 'l'he hardware structure is shown in Figure 1 and consists of a Robotics Research Corporation's Model K 1207 seven degree-of- freedom (DOF) arm/control unit, a VME-based chassis with two MC68040 processor boards and additional interface cards, two joysticks, a motorized platform/control unit on which the arm is mounted, and a Silicon Graphics IRIS workstation,

The system is divided into the local and remote subsystems. Sensor information is received from an integrated sensor/end- effecter (ISEE) unit consisting of two CCD cameras, two infrared triangulation- based proximity sensors, a gas sensor, a temperature sensor, and a force/torque sensor. The robotic inspection laboratory setup is shown in Figure 2. Figure 2 shows the remote site where the inspection task is performed and

[1] The research described in this paper was performed at the Jet I'repulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

consists of the arm with the ISEE, the platform, and a one third scale mockup of part of the Space Station truss structure. The local site consists of an operator control station where the operator resides and is referred to as the "cupola". Within the cupola are the IRIS workstation, two color monitors, a stereo color monitor, and the two joysticks.

The dexterous manipulator used in this study has seven revolute joints in an alternating roll/pitch sequence beginning with the shoulder roll at the base and ending with the tool- plate roll at the hand. The arm pedestal is mounted on a mobile platform of a motorized rail which provides one additional translational degree of- freedom that can be treated as a prismatic joint. Therefore, the complete manipulator system has eight independent joint degrees- of- freedom. This system has two degrees-of- redundancy, i.e. two "extra" joints, since six joints are sufficient for the basic task of hand position and orientation in the three dimensional workspace.

The Robotics Research arm is controlled by a real-time microprocessor- based controller that uses advanced control algorithms for high- level dexterous motion control and interfaces directly with the Multibus-based arm control unit supplied by the manufacturer. The real- time controller is a VM Ebus-based system that uses two Motorola MC68040 processors along with various data acquisition, memory, and communication devices. The VME controller is linked via socket communication to the Silicon Graphics IRIS workstation, which serves as the host computer for the graphical user interface. The controller also has a shared memory interface allowing a high speed communication link with other systems. A separate image processing VM E chassis currently uses this interface to monitor the Cartesian position of the end- effector. The real- time VME chassis and the arm control unit Multibus chassis are connected via a two card VME-to- Multibus adaptor set from the 111'1'3 corporation. This allows a high speed bi- directional shared memory interface between the two buses.

The robot is mounted on a platform which is motorized. The commands to the platform control unit are communicated through a serial port. The platform control unit is capable of providing very accurate position control (0.012 mm accuracy). However, due to the serial port link, communication with the platform control unit occurs at a slow rate relative to the arm control unit.

3 Control System Description

The manipulator Cartesian control flow diagram is shown in Figure 3. The configuration control technique [6, 8] developed at J }'1, is implemented in the VME environment for the seven DOF arm plus the one DOF mobile platform. The major algorithms for the control system are the: forward kinematics and Jacobian computation, a singularity robust inverse kinematic computation, a real- time trajectory generation routine, and proximity sensor compensation.

The computation of the kinematics and Jacobian of the 7DOF manipulator utilizes Craig's interpretation of Denavit- Hartenberg (DH) parameters for frame assignment [1, 4]. This method provides direct computation of the manipulator Jacobian in the world frame of the robot. in addition, the forward kinematics and Jacobian are also computed for the obstacle avoidance task [2] .

A singularity- robust inverse kinematics algorithm is implemented. This technique is known as the damped-least- squares method [1 O, 5, 8]. Basically, the method relies on weighting large joint velocities against large task space errors. The resultant computation of the joint velocities has the following form [8]:

$$\dot{\theta}_d = \left[J^T W_t J + W_v\right]^{-1} J^T W_t \left[\dot{x}_d + K E\right] \qquad (1)$$

where W_t and W_v are the task space error wrights and joint Velocity damping weights, $E = x_d - x$ and K is a diagonal matrix with positive elements. Note that (1) can be also written as follows:

$$\begin{aligned}\dot{\theta}_d \; = \; & \left[J_e^T W_e J_e + J_c^T W_c J_c + W_v\right]^{-1} \\ & \left[J_e^T W_e \dot{y}_d + K_e E_e + J_c^T W_c \dot{z}_d + K_c E_c\right]\end{aligned} \qquad (2)$$

where the subscript c refers to the basic task of positioning the hand, and the subscript c refers to additional tasks for redundancy resolution. Cholesky decomposiion is used to solve (2). It can be seen that as the Jacobian becomes singular, the velocity weight dominates in the inverse matrix term in (2), reducing the commanded joint velocities. The reduced joint velocitics, in turn, act to retard the arm from reaching the singular configuration.

3.1 Configuration Control

The configuration control technique [6, 8] allows specification of additional tasks for redundancy resolution Currently, two choices for the seventh task are available: arm angle control for elbow placement or obstacle avoidance to reach through an opening For the eighth task, two choices are available: platform position control, or elbow angle control.

The "arm angle" is define.d as the angle between the arm plane OEW and the vertical plane passing through the line OW, where O, E and W refer to the origins of the shoulder, elbow and wrist frames, respectively [4]. in the control software, we use a simple and efficient method described in [4] for computing the arm angle and the associated constraint Jacobian. The "elbow angle" is defined as the angle between the upper- arm and the forearm [7]. The platform position is defined to be the positionof the base of the robot with respect to a given world frame. The obstacle avoidance task allows the arm to reach through an opening while using the redundancy to avoid collision [2].

Sine.ecommunication with the platform control unit occurs at a relatively slow rate, and because the platform control unit is currently used as a position controller with no velocity blending between commanded positions, the control system does not treat the platform as a true joint. instead the following scheme is used to mimic an eight DOF system. When the eighth task is platform position control, for a given user- specified trajectory, the controller creates an equal but opposite trajectory for the hand. Since the platform always provides a straight. line trajectory with a trapezoidal velocity profile, the compensating trajectory for the arm is extremely ly simple. The net effect is to maintain the hand position while moving the platform. If elbow angle control is chosen as the eighth task, instead of specifying an explicit elbow angle, an acceptable range of elbow angles is specified, The user specifics a wedge around 90°, in which the controller should maintain the elbow angle, If the elbow angle moves out of this range, the controller will bring the elbow angle back to approximately 90°. The choice of 90° for the elbow angle is to maximize the manipulability of the hand in the arm plane and is also the midrange value of the elbow angle[7]. The control system moves the elbow angle back to 90° by moving the platform, while compensating for the platform movement with an opposite trajectory for the hand. The control system performs a simplistic computation for the platform movement. Assuming a two- link approximation for the arm (see. Figure 4), it is easily shown that the hand must move by:

$$\Delta x_h = \sqrt{x_{hi} + 2 * l_1 * l_2 * cos(\phi_i)} - x_{hi}$$

where x_{hi} is the initial hand position relative to the shoulder; l_1 and l_2 are the upper- arm and forearm lengths, and ϕ_i is the initial elbow angle. The platform is thus commanded to move by $(-\Delta x_h)$. The net effect is to leave the hand position constant while placing the c] bow angle back to approximately 90°. For the K 1207 arm, inaccuracies on the order of +/- 10° is observed. For a typical cam, an elbow wedge angle of 30° is used, 60°< ϕ <120", and has been found to be adequate.

3.2 Joint Limit Avoidance.

In addition to the eight basic tasks, an additional task is added for each joint that is near its limit. This is accomplished within the framework of the configuration control scheme. When joint limits are approached, the system actually becomes "deficient" (as opposed to being "redundant"). The damped- least- squares algorithm automatically relaxes certain tasks based on their weighting factors. The joint limit avoidance task is formulated as an inequality constraint that is activated only when the joint is within its "soft" limit, and is inactive otherwise. Interestingly, the formulation of the extra task is extremely simple. Observe that $J_\zeta^T W_\zeta J_\zeta = W_\zeta$ and that $J_\zeta^T W_\zeta$ reduces to W_ζ, where ζ indicates the joint limit avoidance task, Thus computationally the joint limit avoidance task is extremely fast .

3.3 Trajectory Generation

Two independent trajectory generators are implemented in the system. The first trajectory generator produces smooth continuous cycloidal functions to make the transition from the initial value to the final value in the specified time. A second via- point blending trajectory generator is also implemented in the system [9]. The via point blending trajectory generator allows the specification of several via points. The control system generates a smooth trajectory between the points, while smoothly blending the velocities from one via point to the next.

3.4 Proximity Control

The proximity sensors can be used in three modes.

- Proximity *Off*: No proximity sensor compensation is generated.
- *Standoff Mode*: The sensor is used to maintain a minimum standoff distance from the surface.
- *Servo Mode:* The sensor is used to continuously servo the arm to maintain a constant distance from the surface.

For "standoff mode". a velocity away from the surface is generated when the sensor indicates that the arm is closer than the user- specified standoff distance. The repulsive velocity is a piecewise linear function of the sensor distance (see Figure 5). It starts at zero at the standoff distance, and grows to a repulsive velocity that is equal to the maximum possible commanded velocity at a distance of 15 cm. This is the desired minimum distance away from the surface, and corresponds to 4 cm beyond the tip of the gripper jaws of the ISEE. The repulsive velocity then increases to four times the maximum possible commanded velocity at a distance of $5cm$, after which it is constant to a distance of $0cm$. For "servo mode" the curve is similar except that an attractive velocity is provided for distances greater than the desired servo distance. Since the sensor has a finite range (approximately $5cm$ to $50cm$), the operator needs to move the arm close enough to enter the proximity sensor range, to allow the arm to be "pulled" into the servo distance.

3.5 Simulation mode

The control system implements an arm simulation mode in addition to real arm execution mode. When in real arm execution mode, the control system sends the joint angles measured by the arm control unit to the IRIS. This guarantees that the user views the real arm configuration since the measured joint angles are used. In simulation mode, the control system simply outputs the joint setpoints to a different shared memory location. The commanded joint setpoints are not sent to the arm control unit. The control system then transmits the commanded joint setpoints instead of the actual joint angles to the IRIS. The implementation of a simulation mode on the control system assures that the simulation will effectively duplicate real arm execution since the same code is executed in both cases. However, the proximity sensor data is not being simulated.

4 Graphical User Interface

The graphical user interface (GUI) enables the user to specify the command and control modes of operation, set control system parameters, and determine the manipulator status. Figure 6 shows the IRIS windows for manipulator control. The user first selects the command mode by clicking the mouse on the appropriate button on the menu. The manipulator control system has three *com m and mode* options:

- *Teleoperated Command Mode*
- *Automated Command Mode*
- *Shared Command Mode*

In the teleoperated command mode, the user employs two industrial rate joysticks to generate command inputs to the manipulator system. These joysticks are identical to the ones used by astronauts to operate the Remote Manipulator System (R MS) from the Space Shuttle bay. The first joystick has a 3- axis square handle and is used solely to control translation; the second joystick has a 3- axis rotational grip handle and is used for controlling orientation. The second joystick also has three mounted switches. The trigger is used to change the arm angle in one direction at a constant speed. The slide switch is used to move the platform in one direction at a constant speed. The momentary pushbutton switch has dual usage; it is used to change the direction of both the platform and the arm angle. The user can select the gains that map the joystick deflections into the arm displacements. in the automated command mode, the motion commands to the arm arc issued by a trajectory generator software in the VME. in this case, the user inputs on the keyboard or the slider bars the desired final values of the hand coordinates and arm angle or the target values of the joint angles, as well as the motion duration.

The system also provides shared command mode by combining the teleoperated and automated modes, where the commanded values for the arm coordinates arc read both from the joystick channel and the trajectory generator channel and added together to form the commanded arm coordinates. The shared command mode of operation is particulary useful in applications where the hand coordinates arc moved in automated mode by the trajectory generator software while the user is commanding the elbow motion through the arm angle using the joystick in teleoperated mode.

The user can also select any of the following *control modes* to operate the arm by clicking the mouse on the appropriate but ton on the I RIS screen:

- *Joint Control Mode*: Commands are issued to the seven joint angles of the arm and the platform.
- *Cartesian- World Control Mode*: Commands arc expressed relative to a fixed user- defined frame of reference (the world frame).
- *Cartesian- World Relative Control Mode:* Motion commands are in the world frame coordinates measured relative to the current world frame coordinates of the tool.
- *Cartesian- Tool Control Mode:* Motion commands are in the end- effector coordinates measured relat ive to a reference frame displaced by the tool length from the current hand frame. The tool length is defined by the user using the IRIS screen.

The operator can choose between arm simulation mode or real arm execution mode with the click of a button. The choice for the seventh and eighth tasks are also software toggle buttons. When switching from platform control to elbow angle control, the eighth task parameter in the sliders window changes from the platform position to the elbow angle. When the obstacle avoidance task is activated, the obstacle avoidance window shows the distance of the arm from the center of the opening, and the entry angle. The entry angle is defined as the angle between the arm link entering the opening and the normal to the opening. The sliders display the current values, and the maximum allowable values. When the maximum allowed values are exceeded, the obstacle avoidance task is aborted. The system allows the operator to continue to move the arm, however, the responsibility for obstacle avoidance falls on the operator. Finally, a simple proximity control window is provided to turn on/off proximity sensor compensation, to select standoff or servo mode, and to set the standoff distance.

The various command and control modes provide considerable flexibility for operation of the robotic arm. The mode of operation can be changed on-line by the user at any time based on the task at hand. The control system greatly increases the up-time of the arm by being robust to singularities and joint limits.

5 Software Architecture

In this section, we discuss the software components of the VME environment used for real-time control of the manipulator. All of the software executing on the VME environment is written in the C language. Code is developed on a SUN Spare 10 UNIX computer utilizing Wind River System's VxWorks development environment. The development environment consists of a C compiler, remote symbolic debugger, and the Stethoscope real-time -monitoring tool. The code is downloaded through Ethernet to the target processor boards for execution.

Figure 7 shows the software structure of the VME-based controller. The VME chassis hosts two Motorola MV167 MC68040 CPU cards that perform all necessary computations to provide real-time control of the manipulator and the base platform. The *ui task* interfaces with the high-level system residing in the IRIS to receive user commands and to send acknowledgement and state information after execution of the commands. The information is routed bi-directionly through Ethernet using the UNIX socket protocol. Once a command is received from the IRIS, the *ui task* parses the command and then writes appropriate command information onto the shared memory card to pass along to the other tasks. All commands from the IRIS are acknowledged by the controller, Every reply from the controller contains the state of the system which includes in formation such as sensor data, current joint angles, current mode, and Cartesian task values. The in formation also includes the current parameters that the system is using such as the elbow wedge angle, proximity standoff distance, and the hand controller gains. The state of the controller also includes information such as whether joint limit tasks are activated, whether the arm power is on, and the current seventh and eighth tasks chosen.

The *hc task* is designated to perform data acquisition. It controls the activities of the Analog- to Digital (A/D) converter boards which are used to read in joystick inputs and sensor data. The first A/D board reads in various voltage outputs of the six potentiometers on the joystick. In addition, it monitors the voltages of the three switches on the rotational grip joystick. The second A/D board reads in the sensor data from the temperature sensor, gas sensor, and the two proximity sensors.

The *ctrl task* performs real-time trajectory generation and kinematic computations. The kinematics and Jacobian computation, damped-least-squares computation, and the Cholesky decomposition computation have been timed to take approximately $2.5ms$ to complete. The system does not presently use the actual joint angles from the arm control unit in the control law computations.

The *rrs task* communicates with the arm cont rol unit at the maximum possible rate of 400112, i.e. every $2.5ms$. Each joint servo motor can be independently commanded in any of the four modes: posit ion, velocity, torque, and current. This feature enables the operation of the robot under both kinematic and dynamic control schemes, and therefore facilitates validation of a variety of arm control laws. The driver performs all necessary handshakes with the arm control unit software and conversion of data into appropriate formats. In addition, joint position and velocity limits are also checked at each cycle for safety reasons.

6 Experimental Results

in this section, we present the experimental results, on real-time control of the mobile manipulator system. The purpose of the experiment is to demonstrate how base mobility can be used to appropriate) move the arm in order to reach a target hand position.

In this experirnent,the arm is operated in Cartesian world control mode using automated command [illegible] with the via point blending trajectory generator. The units used internally are meters and radians. [illegible] rent system uses task weightings (W_t) of 1 an[illegible].

gains (K) of 1 for the end-effector and arm angle control tasks, and joint velocity weighings (W_v) of 0.005. The weighting for the joint limit avoidance task (W_ζ) is a cycloidal function that starts at O when the joint is within 10 degrees of its limit (the "soft limit"), and increases to a maximum of 10 when the joint is within 3.5 degrees of its limit (the "hard limit"). The feedback gain used for the joint limit avoidance task is 1.

Starting from the initial hand position of x =- - 3.075m, y = - 0.624m, and z = 1.200m, the hand is commanded to move to the final position of x = - 3.925m, y = - 0.624m, and z = 0.600m in 35 seconds. Note that the target hand position is beyond the reach of the arm if base mobility is *not* activated. The elbow angle control is selected as the eighth task. Figure 8 shows the experimental results for the system. The plot shows that when the elbow angle exceeds 120°, the platform starts to move and brings the elbow angle back to approximately 90°. Notice that since the hand velocity is greater than the base velocity, the elbow angle exceeds the wedge 60° $\leq \phi \leq$ 120° momentarily until the base mobility y has sufficient time to compensate for the hand motion. The control system continues to execute the end- effector task and exhibits good tracking performance. Thus, the base mobility of the arm is used effectively to prevent the manipulator from reaching its workspace boundary.

7 Conclusions

A real- time control system for a mobile redundant seven DOF manipulator is described in this paper. The manipulator is well- suited for tasks that demand positioning and pointing a payload dexterously, such as in the supervised robotic inspection project at JPL. The control system provides dexterous motion try controlling the end- point location and the manipulator posture simultaneously. 'l'his enables operation of the manipulator in the presence of workspace obstacles and provides the capability to reach inside constricted openings. This yields a general- purpose highly - flexible manipulator control system which is capable of performing many tasks requiring teleoperation and autonomous manipulation in unstructured dynamic environments in both space and terrestrial applications. In fact, although the arm control system has been designed for the robotic inspection project, it possesses generic capabilities that can be used for many diverse applications utilizing different hardware.

References

[1] J. Craig. *Introduction to Robotics: Mechanics and Control.* Addison- Wesley, Reading, Massachusetts, 1986.

[2] K. Glass, R. Colbaugh, D. Lim, and H. Seraji. On-Line Collision Avoidance for Redundant, Manipulators. *Proc. IEEE Intern. Conf. on Robotics and Automation, Atlanta, Georgia,* pages 36-43, May *2-61993.*

[3] S. Hayati, J. Balaram, H. Seraji, W.S. Kim, K. Tso, and V. Prasad. Remote Surface Inspection System. *Proc. IEEE Intern. Conf.* on *Robotics and Automation, Atlanta, Georgia,* pages 875-882, May 2-61993.

[4: K. Kreutz-Delgado, M. Long, and H. Seraji. Kinematic analysis of 7DOF anthropomorphic arms. *Proc. IEEE intern. Conf. on Robotics and* Automation, Cincinnati, *Ohio,* pages 824-830, 1990.

[5: Y. Nakamura and H. Hanafusa. Inverse Kinematic, Solutions with Singularity Robustness for Robot Manipulator Control. *ASME Journal of Dynamic Systems, Measurement, and Control, 108,* 1986.

[6] *H.* Seraji. Configuration Control of Redundant Man i pulators: Theory and Implementat ion. *IEEE Trans. on* Robotics *and Automation,* 5(4):472- 490, 1989.

[7] H. Seraji. An On-Line Approach to Coordinated Mobility and Manipulation. *Proc. IEEE Intern. Conf. on Robotics and Automation, Atlanta, Georgia,* pages *28-33,* May 2-61993.

[8] H. Seraji and R. Colbaugh. Improved Configuration Control for Redundant Robots. *Journal of* Robotic *Systems,* 7(6):897- 928, 1990.

[9] R. Volpe. Task Space Velocity Blending for Real-Time Trajectory Generation. *Proc. IEEE Intern. Conf. on Robotics and Automation, Atlanta, Georgia,* pages 680-687, May 2-61993.

[10] C. Wampler and L. Leifer. Applications of Damped-Least- Squares Methods to Resolved Rate and Resolved-Acceleration Control of Mani pulators. *ASME Journal of Dynamic Systems, Measurement, and Control,* 110:31- 38, 1988.

Figure 1: Hardware Architecture

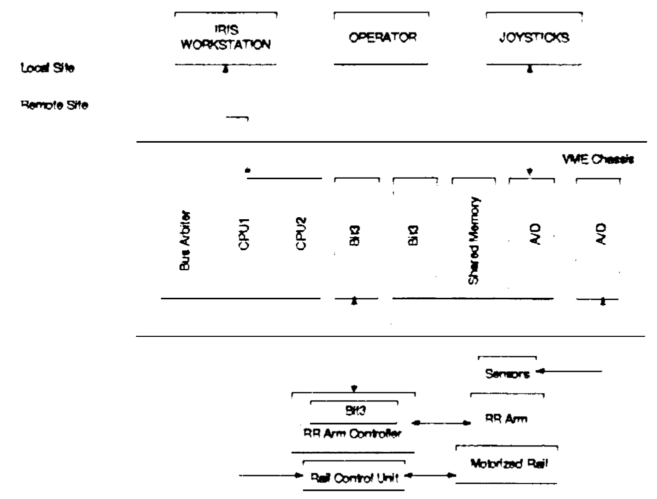


Figure 2: Remote Surface Inspection Laboratory

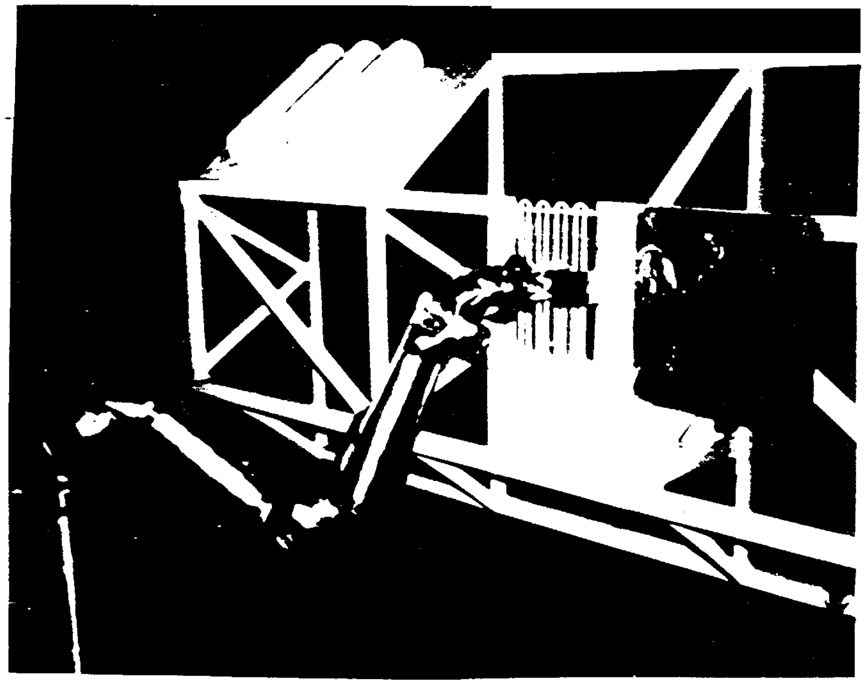


Figure 3: Control Diagram

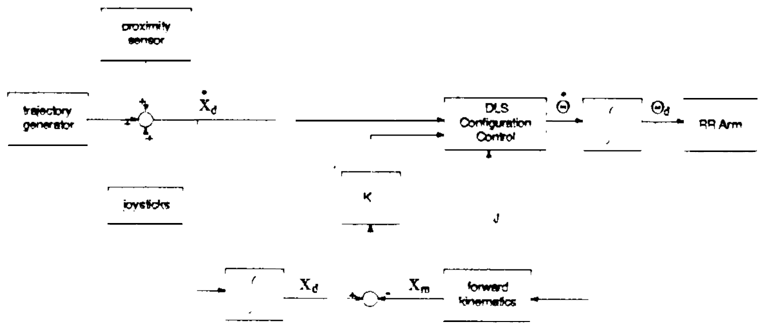


Figure 4: Two-Link Approximation

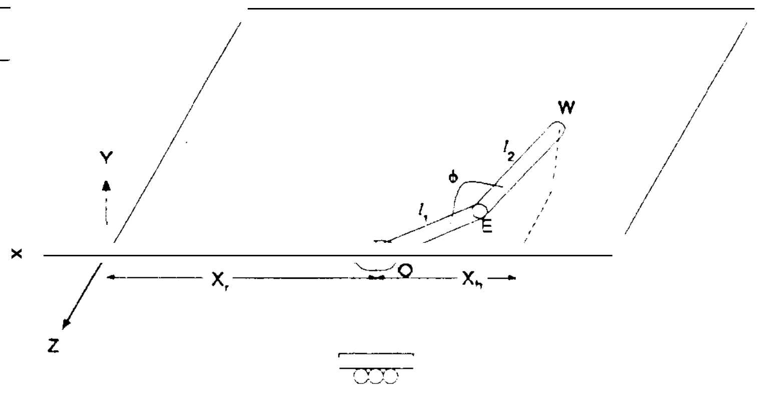


Figure 5: Proximity ed Velocities

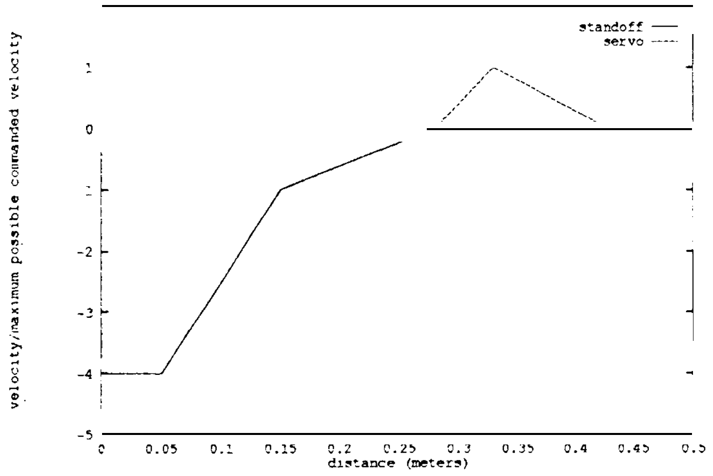


Figure 6: Graphical User Interface

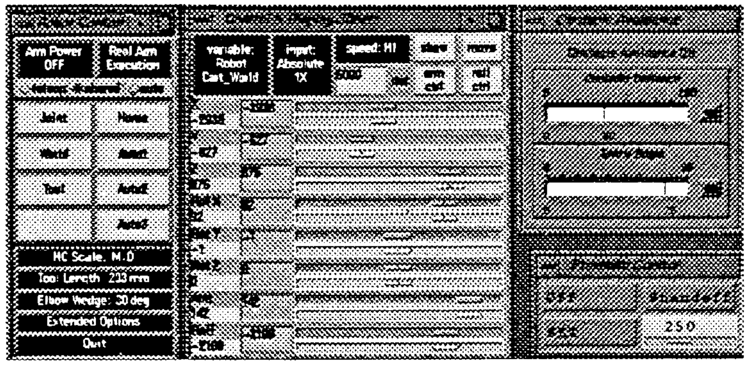


Figure 7: Ar ect

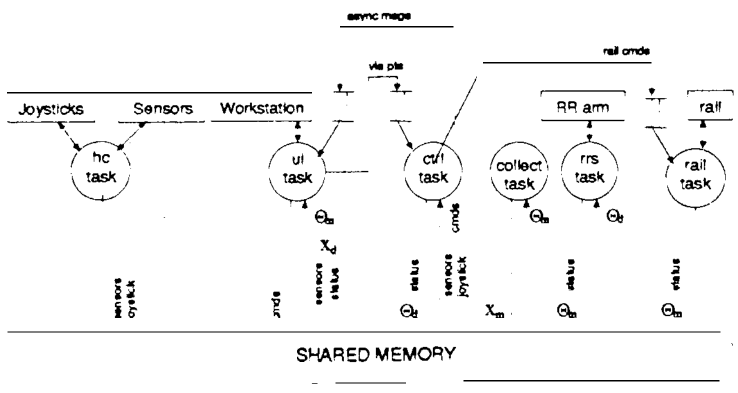
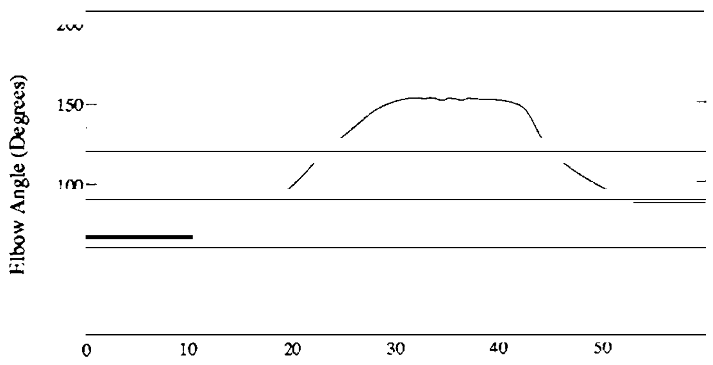


Figure 8: Experimental Data


Elbow Angle (Degrees)