A TRIDENT SCHOLAR PROJECT REPORT

NO. 187

"STEREO VISION CONTROLLED BILATERAL TELEROBOTIC REMOTE ASSEMBLY STATION"

UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND

U.S.N.A. - Trident Scholar project report; no. 187 (1992)

"STEREO VISION CONTROLLED BILATERAL TELEROBOTIC REMOTE ASSEMBLY STATION"

A Trident Scholar Project Report

by

Midshipman Robert L. Dewitt, Class of 1992

U. S. Naval Academy

Annapolis, Maryland

Adviser: Professor Kenneth A. Knowles
Weapons and Systems Engineering Department

Accepted for Trident Scholar Committee

Chair

8 May 1992

Date

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
8 May 1992

3. REPORT TYPE AND DATES COVERED
Final 1991/92

4. TITLE AND SUBTITLE
STEREO VISION CONTROLLED BILATERAL TELEROBOTIC REMOTE ASSEMBLY STATION

5. FUNDING NUMBERS

6. AUTHOR(S)
Dewitt, Robert L.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S.Naval Academy, Annapolis, Md.

8. PERFORMING ORGANIZATION REPORT NUMBER
U.S.N.A. - TSPR; 187 (1992)

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Accepted by the U.S.Trident Scholar Committee

12a. DISTRIBUTION/AVAILABILITY STATEMENT
This document has been approved for public release; its distribution is UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

The objective of this project was to develop a bilateral six degree- of- freedom telerobotic component assembly station utilizing remote stereo vision assisted control. The component assembly station consists of two Unimation Puma 260 robot arms and their associated controls, two Panasonic miniature camera systems, and an air compressor. The operator controls the assembly station remotely via kinematically similar master controllers. A Zenith 386 personal computer acts as an interface and system control between the human operator's controls and the Val II computer controlling the arms. A series of tasks, ranging in complexity and difficulty, was utilized to assess and demonstrate the performance of the complete system.

14. SUBJECT TERMS
Robot Vision; Robots - Motion; Robots - Control systems Manipulators(Mechanism); Degree of freedom

15. NUMBER OF PAGES
63

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	

ABSTRACT

The objective of this Trident research project has been to develop a bilateral six degree-of-freedom telerobotic component assembly station utilizing remote stereo vision assisted control. The component assembly station consists of two (i.e. bilateral) Unimation Puma 260 robot arms and their associated controls (slave robot), two Panasonic miniature camera systems, and an air compressor. Each Puma arm has six degrees-of-freedom and is independently controlled by a VAL II computer. In addition to the six degrees-of-freedom, the arms have pneumatic end effectors (or *grippers*). The two Panasonic cameras are positioned to provide the operator with a working view of the assembly area. The operator controls the assembly station remotely via kinematically similar master controllers. These have been designed to give the operator a comfortable feeling when in control. The master controllers provide joint angles and gripper status information necessary to control the slave robot arm. A Zenith 386 personal computer (PC) acts as an interface and system control between the human operator's controls and the VAL II computers. In order to view the assembly operation, the operator is provided with real-time stereo imaging via two closed-circuit television (CCTV) systems. The images from the two CCTV monitors are optically combined via a lightweight cap-mounted periscope mirror

assembly to permit human depth perception. A series of tasks, ranging in difficulty and complexity, have been utilized to assess and demonstrate the performance of the complete system. Alternate end effectors, compliant wrists, and tactile or force feedback are all desirable future additions which would further enhance the operator's feeling of comfort and control.

ACKNOWLEDGMENTS

When I began this project, I really did not have a clue how or where to start (I just wanted to get moving). Thanks to Professor Knowles (that's K-n-o-w-l-e-s), I was able to set my spinning wheels on the pavement and proceed in the right direction. I only felt neglected when he was letting me experience the learning process. I am extremely grateful for this and everything else he has provided me.

I would also like to acknowledge the many others who provided support to me in one way or another: Professor Clement for always helping on short notice; all the other faculty advisors for their recommendations and interest; Ralph 'the hole-in-one' Wicklund (always had something for you to do and you came through!), Carl Owen (thanks for the drive-through service!), Herb - the aluminum man (fantastic work), Bill Lowe, Clyde, Sam, the enlisted guys, etc. for their technical support if they were lucky enough to let me find them when Ralph was tied up; Captain Cass, USMC, and the rest of my chain of command for their full support; my family for not knowing what this was and saying 'terrific' anyway; the various dates that bored me (giving me extra research time); and most of all I would like to thank God for blessing me with intelligence to make it this far.

TABLE OF CONTENTS

BACKGROUND

A manipulator is defined to be a mechanical arm with a gripper at the end. The term 'robot' commonly refers to any computer controlled manipulator, but more specifically implies the use of artificial intelligence (AI). Teleoperators, on the other hand, are manipulators that are remotely controlled by a user (human). A teleoperator often incorporates the use of computers, but control originates with the operator. Modern day robots all evolved from teleoperated manipulators,[1] the first introduced in 1944 by R. Goertz. This was a basic master-slave mechanical configuration (no electronics or computers). With the progress in computers, teleoperators became able to operate more remotely and with better dexterity. This has been possible because of increased computing capabilities, which enable the system to have various external sensors to give the user a better sense of being there (telepresence). There is a significant advantage to having a user controlled robot. "The reason for man's presence stems from his ability to set strategy and to deal with the unexpected - those situations we cannot preprogram

---

[1] "Modern industrial robots essentially evolved from teleoperators and numerically controlled machine tools, both of which appeared at almost the same time." (Wolovich 2).

into a machine's memory."[2] The basic idea is to project man's dexterity and intellect into the work space.

The telerobot has many useful applications in space, underwater, nuclear research, and others.[3] The telerobot can operate in mediums that are too difficult or impossible for man, such as maintaining a nuclear reactor during plant operation. A telerobot, while being operated around the clock from a control station on Earth, might be sent into space. (Figures No.1-2 illustrate the design for a telerobot that could be used to assist an astronaut). An astronaut can only spend a limited amount of time in space, and each trip is costly. Deep sea operations are another area where telerobotics can be of great assistance. Telerobots are ideally suited for these types of applications because there are bound to be circumstances which arise that will require human intuition or wisdom. The telerobot can then be used to apply the resulting solutions. A fully autonomous robot is clearly outmatched in these situations.

A telerobot's capabilities will vary depending on the features incorporated into its design. Typical features are: multiple arm system; joystick or kinematic control; vision system with multiple views and/or stereo vision; tactile

---

[2] (Johnsen 1).

[3] "Telerobots are becoming increasingly important in several areas. One application is public safety. Teleoperated systems are used for police, fire, and bomb disposal work." (Poole 391).

and/or force feedback; and compliance. Multi-arm systems are better adapted to areas such as industrial automation (assembly) and space technology. However, there are more problems involved in collaborating multi-robot-arm systems than in single robot arms.[4] Typical problems involve simultaneous motion. This problem is minimized in telerobots by the amount of telepresence available to the operator. The more there is, the more natural it will feel to the user (it will feel more like coordinating his/her own arms). Telepresence can be provided by the type of master controllers, vision system, and sensor feedback.

There would be significant problems for the user if the controls (master controllers) were not well suited to him/her. It would require much more effort to control a six joint robotic arm with six dials than it would with a single six degree-of-freedom kinematically similar controller. A kinematic controller gives the user a better association with (feel for) the robots motion. This is especially true in anthropomorphic designs. The addition of kinematic redundancies would allow for even better control in confined space and around obstacles, but with their benefits they also bring a challenging and complex control problem.[5]

Vision feedback is the most important sensor available to the user. It is what projects him/her into the work space.

---

4 (Hemami 21).

5 (Martin 1-37).

Most telerobots use multiple views to give the operator position and performance feedback. This is especially the case for remotely operated telerobots. While this provides the user with adequate visual information, stereo imaging allows him/her the ability to perform assembly type operations with greater ease. This is because humans are provided depth perception in the work space. "In experiments with master-slave manipulators fitted with television viewing devices, it has been found that stereo viewing is superior to normal 2-D, but only if it is correctly aligned."[6]

The addition of external sensors is the final way in which the operator can better understand what is happening in the work cell. The types of sensors to be used will depend on the nature of the operations to be performed. For example, pressure and temperature might be important to a telerobot performing rescue operations. Another useful sensor would be one to measure tactile and force information and relay it to the user. "Force feedback can considerably improve an operator's ability to perform complex tasks that interact with a remote environment, such as assembly or surface following and inspection."[7] A difficult problem is that force feedback, when used as a control force, has a strong de-stabilizing effect when time delays are present (such as in this project).

Several options exist for integrating external sensors,

---

[6] (Young 193).

[7] (Niemeyer 152).

such as force feedback, into a telerobotic control station. One is to have the robot controller sample these signals for use as a control force. In addition to having de-stabilizing effects, this is extremely complex and time-consuming. A second option is to leave the external sensors completely outside the control loop. One way to achieve this is to place the operator in the force feedback loop, which can be accomplished using tone and vibration signals for the operator to interpret (pseudo-force feedback). Strain gauges and piezo-electric strips would meet this requirement. In addition, spring loaded microswitches could be used to prevent too much force from being applied in a certain direction and so provide a form of collision control. The use of more sophisticated equipment, such as range-finders, also would be beneficial, but this would require extensive resources. These methods of implementing pseudo-touch/force feedback, while beneficial, still possesses the troublesome time delay effect.

Adding compliance to the end effectors is necessary to perform many assembly operations that require exact positioning, such as putting a peg in a hole. Again, there are components on the market designed to provide this, but they are complex, expensive, and difficult to obtain for certain robot types. Addition of compliance, however simple the implementation, will improve telerobot performance.

In summary, teleoperators require telepresence to be effective. Telepresence can be provided by using a kinematically similar master controller, stereo (3-D) vision feedback, and some type of touch/force feedback to the user. The addition of compliance enables the telerobot to perform those operations requiring careful and exact positioning. Teleoperation is the projection of man's dexterity and problem solving abilities into the workspace. The remoteness provided by telerobots enables man to perform assembly or maintenance operations in unsuitable environments.

PROJECT OVERVIEW

The general idea of the Stereo Vision Controlled Bilateral Telerobotic Remote Assembly Station is illustrated in Figure No.3. This shows how the user manipulates the master controllers while observing the resulting actions from the slave robots in the work cell. The work cell and control station are linked through a personal computer (PC), which communicates with the master controllers and the slave robot arms.

The object of this research project was to develop a useful telerobotic system (using the resources available) to permit realistic equipment assembly operations to be performed remotely with substantial operator telepresence.[8] All the features explained in the BACKGROUND section were investigated, with the finding that a kinematically controlled multiple arm system with real-time stereo vision feedback, compliant wrists, and pseudo-force feedback would be the most effective.

There were several things to consider in the design of the master controllers. Not only is it desirable to have them provide a comfortable interface with the operator, but they

---

[8] "Effective teleoperation requires telepresence, meaning that the operator feels that he/she is at the remote site and is operating the robot rather than the hand controller." (Hayati 7).

also must provide accurate position information that can be used to control the slave arms. The first thing that was investigated was the method of controlling the robotic arms. There are two methods of controlling the end effector's position and orientation in space. The first involves a forward matrix solution of the master controller which determines the position and orientation of the end effector in terms of three coordinates and three angles referenced to a universal (world) coordinate system. This was found to be inadequate in that it allowed the intermediate joints of the Puma arm to have multiple angles that still placed the end effector in the correct position and orientation. Thus, the operator might not see the Puma arm in the same physical configuration as the master controller. Not only does this reduce the operator's sense of telepresence, but also it provides poor control in situations in which physical obstacles or objects must be avoided by the intermediate joints of the slave arm.

The only way to accomplish absolute positioning of the slave arm is to control each joint directly by using joint angle commands. A kinematically similar scale model design was chosen because it eliminates the complex computations necessary for the forward solution. It also ensures that all joint angles are preserved and can be used without scaling. This gives the operator a much better sense of control (because the slave arms will follow the identical movements of the controllers). The six degree-of-freedom kinematically similar master controller provides the optimum control with the most reliability, mathematical tractability, and accuracy. Figure No.4 shows the Schilling master controller and manipulator. The master controller and manipulator design for this project are illustrated in Figures No.5-6.

The second consideration for the design of the controllers was comfort during use. Both full scale and subminiature designs would be inconvenient to operate, and by looking at other designs, a one-half scale model of the Puma arms provided the most comfort.

The second feature of this telerobot is that it is controlled by stereo vision feedback. Vision feedback is essential for a remotely controlled system because it provides position information to the operator. In order to convey to the operator the manipulators relative depth to the screen, one of two (or both) methods must be used. The most common way of establishing depth perception is by using multiple views (alternate perspectives) of the manipulator's work space. This is an effective method, especially when zoom lenses are incorporated. The disadvantage, however, comes at the expense of telepresence. It is tiresome and difficult to get a feel for control when the user must constantly look from monitor to monitor and the depth perception is not a three dimensional perspective.

This project has proven the value of stereo vision.

Stereo vision is far superior to 2-D vision because it provides the depth perception the same way it is observed by man's eyes. This creates the maximum amount of telepresence possible for the user. It also eliminates the time required for looking at various screens and enables the operator to concentrate on the operation being performed. This does not mean that the operator cannot benefit from having alternate views, but these are unnecessary for depth perception when stereo vision is used. Alternate views are helpful in seeing behind obstacles.

Stereo vision for this project is achieved by using a periscope mirror assembly (Figure No.7) to correctly align each of the user's eyes with a separate video monitor. The cameras for these monitors are mounted with the robot arms to provide the same perspective and field of view that the user has when using his/her own arms.

In order to further increase the sense of telepresence, many telerobots incorporate the use of external sensors. Tactile sensations through the implementation of touch/force feedback would give this project a higher sensitivity and thus enable it to perform delicate operations. This design must give the user real-time information on the forces being encountered by the robot in the work cell. Strain gauges, piezo-electric strips, and microswitches can be used to provide tactile information to some degree. These components would give the user needed information for the performance of more precise and delicate operations. The robot (teleoperator) often must know how much force to apply. If it is too much, the object may be crushed. If it is too little it may slip out of the gripper.[9] In addition, collision information should be made available to the user. Pseudo-force feedback was chosen because it does not directly affect the control of the arms, thereby avoiding any de-stabilizing affects. Due to time limitations, the employment of this option is being left for future research.

Compliance is also an important, perhaps essential, feature for assembly robots. Compliance refers to the yielding or deformation of a component due to externally applied forces and inertia.[10] The use of compliant wrists gives the user a greater flexibility to perform assembly tasks that require precise alignment. "If any one of us were to take a drink from a glass, for instance, by using absolute positioning, we would probably spill the drink down our chin and clothes or chip either our teeth or the glass when they collided."[11] Compliance allows for careful and gentle adjustments. The degree of deformation of the compliant wrists also provide an important visual cue to the operator about the forces being experienced by the slave arms.

Using available resources, this multi-arm telerobotic

---

[9] (Robillard 71).

[10] (Dorf 79).

[11] (McDonald 83).

assembly station has been constructed to provide the operator with a high degree of telepresence. The 3-D stereo vision and the kinematically similar master controllers provided telepresence. The addition of pseudo-force feedback is left for future study. Better control over difficult positioning operations has been achieved by adding compliance to the wrist joints, which also provide secondary visual information about the forces acting on the slave arms.

SYSTEM DESIGN AND CONSTRUCTION

Many considerations interact when developing a telerobotic station for assembling components, and can be grouped into three categories: communication and interfacing; controller modeling; and vision feedback.[12] Two parts of this project involve communication and interfacing. One is the communication between the master controllers and the personal computer (PC), the other, between the PC and the Puma robot (slave). Figure No.8 illustrates the teleoperator control loop. Both parts involving communication and interfacing utilize the personal computer. *In order to be able to control* the robot it is necessary to take signals from the master controller and put them into the PC for numerical processing. These signals must be converted to commands for the Puma robot and then communicated to the Puma.

The control signals from the master controllers are voltage measurements from potentiometers, which represent the positions of the master controller's joints. These voltages are read into the PC through a DT-2801 analog to digital (A/D) converter and mapped into appropriate robot joint angles, using known offsets. These angles are then sent to the robot

[12] "All teleoperator systems share several basic features: a display to monitor the situation; a remote effector to implement decisions; and communications links to carry information." (Uttal 124).

as position commands using standard RS-232 serial communication.

The controller modeling was the most challenging and important portion of the project design. It is important to have master controllers that are reliable, accurate, and comfortable to use. The two master controllers had to be constructed to provide accurate voltage measurements to the PC. As previously stated in the PROJECT OVERVIEW section, the most reliable master controller design (for this project) is one which requires less manipulation of the signals by the computer. Therefore, reliability and accuracy dictated the six degree-of-freedom kinematically similar controller design shown in Figure No.5. Considerations of comfort were realized in the scale factor used as well as the positions of each controller relative to one another at the control station. The scale factor (1:2) used to construct the master controllers produced a design 8" tall. This scale factor was also applied to the distance between the Puma arms (32") and controllers (16").

The vision system, like the master controllers, must be reliable and comfortable to use. In order for it to be reliable it must provide real-time video feedback to the user. This has been achieved by using Closed-Circuit TV (CCTV) between the work cell and control station. It and the master controllers are the only physical interfaces between the assembly station and the operator. This vision system must provide the user with more than an adequate view of the assembly operation. It must also have some method to relay depth information. 3-D vision is provided to the operator through the two video monitors by using an adjustable depth perception device (periscope mirror assembly shown in Figure No.7). This and alternate viewing locations give the operator all the necessary position information.

This system also must be adaptable for users with different eye widths and users who wear eyeglasses. Therefore, the periscope mirror assembly was constructed with an adjustment for different eye widths and distances from the eyes. For more comfort, the cameras in the work cell have been mounted parallel to each other with an adjustable distance between them to provide the best sense of depth perception for the user (2" was found to be ideal for a 12" focal distance). One of the best features of this design (Figure No.7) is that it does not restrict the operators view, This permitting him to take notes, make keyboard entries, or view the computer monitor without removing the apparatus.

In order to perform those operations requiring exact positioning (such as putting a peg in a hole), the telerobot must have some degree of compliance. It has been possible to add some compliance by inserting a 3/16" thick piece of stiff foam rubber in the wrist joint of the robot. A 1/16" thick piece of soft rubber was placed on the inside of the end effectors (providing some compliance and a better grip). The installed compliance proved effective in all directions, including rotation. This was necessary to enable the assembly station to accomplish difficult positioning operations without applying too much torque on the wrist joints (causing them to stall).

After the various parts of this project were successfully designed, they had to be carefully integrated into a functional remote assembly station. The robot's response depends heavily on the quality of programming used to take in signals from the master controllers and communicate joint commands to the Puma robot. At the start of the project, both the Turbo C and True BASIC programming languages were being learned and utilized. Turbo C was desirable because of its superior speed, although prototype programs could be completed much faster when using True BASIC. This was because of its structure, available library routines (communications and graphics), and adequate speed. Thus, True BASIC was chosen to provide all the control from the PC.

The final program (Appendix A) reads the A/D converter, creates appropriate joint commands, and sends them to the Puma. It also permits the user to send commands from the keyboard. Joint commands are created by making a text string out of the six joints and end effector command (open, close, or stay the same) for each controller. These are then sent to the appropriate Puma using several nested subroutines (Rpts, Lpts, Terminal, Switch, etc.). These subroutines are located after the main body of the program.

Several macros were developed to reduce the amount of keyboard entry. The macros provided were: PANIC (to immediately stop both arm's motion); INITIALIZE (sets up the Pumas for receiving information); CALIBRATE (permits better accuracy); START PTS (begins sending joint angle commands to Puma); SWITCH (allows the operator to move keyboard entry to the other Puma); and READY (places the Puma arm in a position ready for storage). The screen graphics routines present this information and the communications between the PC and both Pumas in a comfortable format which is easy to understand.

Sending the joint commands is only half of the communication effort. These commands have to be received and interpreted by the Pumas' VAL II computers.[13] It was necessary to write additional programs in the VAL II language to have the arm perform the desired movements. This program is listed in Appendix B. (The procedures to start-up and run the entire system are listed in Appendix C).

The response of the manipulator is also a function of the user's coordination and dexterity. The controller design, stereo vision system, and compliant wrists were designed to facilitate the operators telepresence and consequently, ability to control effectively.

---

[13] "VAL II allows supervisory control of the robot from a remote computer. Structured programming statements have also been added." (Poole 259).

PRELIMINARY ANALYSIS AND SIMULATION

Figure No.9 illustrates the basic control flow of this project. The top portion shows a task being put into the system which results in a control signal (the resultant of that task and the video feedback). This control signal is actually the output from a joint potentiometer on the master controller. The output (in volts) from this pot is read into the computer by an A/D converter. This signal is then converted into a joint command (using a scaling factor, degrees/volt) and sent to the corresponding robot joint controller. The performance of the robot relative to the control signal is visually fed back in real time to the user (who is manipulating the master controller). Each master controller and robot is actually comprised of six joints (degrees-of-freedom). Only one joint is herein analyzed as all joint control channels are identical.

The bottom portion of the figure shows the system in more detail. Here, the computer is replaced by a sampler (T=1/36 seconds) and a zero-order data hold. The sampler represents the robot system sampling data from the computer at 36 Hz. The controlled system is a unity gain DC servomotor (transfer function), where the electrical and mechanical time constants are 0.1 and 0.01 seconds respectively. The video feedback is real time, however there is a 0.1 second time constant modeled to represent human reaction time. Both the open and closed loop transfer functions were found in the Z-domain. MATLAB was used to facilitate this analysis and obtain impulse, step, and ramp response plots for the open and closed loop systems. Z-plane Root Locus plots were also obtained. All MATLAB plots are shown in Appendix D.

Comparison of the impulse, step, and ramp responses for the open and closed loop system, reveals several differences. By looking at the step responses, one notices that the closed loop system has a faster response and roughly the same settling time as the open loop system. The impulse response plots also show this. The ramp response plots illustrate that the closed loop system has a smaller steady state error constant. The step and ramp responses also show how closing the loop attenuates the steady state output by approximately one-half (the final value in the step response, and the slope in the ramp response). This to be expected since all gains are unity and the closed loop is simply G/(1+GH), where G is the forward gain and H is the feedback gain.

The Root Locus plots for the system with (closed loop) and without (open loop) video feedback are nearly identical in that they are stable for gains up to about 18. At this high gain, the roots exit the unit circle and cause instability. Gains this high will never be approached.

There are several other factors involved in the realization of the system. Most of these minor factors have a negligible effect on the performance of the system, and they have been neglected in the analysis. One of the factors is the sampling involved in the A/D conversion. This event is much faster than 36 Hz, and is therefore considered to be real time. There is also another inner loop to the plant (robot). This is a closed loop which cycles 32 times in the 28ms window (36 HZ) with feedback to place the robot joint at the proper angle. In depth details of this are not explained in the documents for the Puma arm, but the resulting time constant is approximately 0.001 second. Therefore, the plant also is considered to be continuous.

Other considerations for this system are aliasing and position lag. Aliasing is a function of the sampling rate. The Nyquist frequency is 18 Hz for this system. This is far above anything that would be encountered with the applications of this project. Another consideration is position lag. This is dependent on what speed the robot arm is set (1-100%) and how much movement is required. There is no speed for which the arm will follow the exact path in real-time. There occurs instead, a very short 'tail chase.' This is acceptable for many applications as long as the lag is very short. To have the arm follow the exact path, a queue would have to be set up in the computer. This, however, would introduce a major perception problem for the user with the vision feedback appearing to significantly lag the controller movement. This also gets away from the teleoperator concept and is not considered to be a viable option. To reduce the position and lag problems, an optimum robot speed is used. The position lag is mostly due to the time it takes for the Puma to reach the new position as it follows the master controller motions.

From this preliminary analysis and simulation, it is seen that the high gains required to make the system unstable are much larger than those encountered in the real system. Therefore, the system will always be stable. The time lag ('tail chase') is minimized by finding an optimum speed for the robot arm.

PERFORMANCE EVALUATION

The performance of the Stereo Vision Controlled Bilateral Telerobotic Remote Assembly Station is evaluated based on its ability to execute and repeat an assembly operation (repeatability), as well as how easily the assembly task can be performed (tractability). Repeatability, accuracy, resolution, compliance, path error, and overshoot/under shoot are several factors that contribute to the overall performance. "Currently there are no standards for robot calibration or performance specification determination."[14] The selected evaluation tests ranged in difficulty from those performed with one arm (object placement) to those requiring coordinated motion.

Single arm object placement was executed several times to determine repeatability. A histogram of the forty-five trials is illustrated on page 59 of Appendix D. The following statistics were obtained: a mean of 20.5 seconds; median of 18.5 seconds; and standard deviation of 7.5 seconds. (Page 60 of Appendix D displays these times against their trial numbers). This test proved that the repeatability is unlimited, and that the operation is easy to perform. Surprisingly, operator training was not a significant factor for this test (untrained operators had similar times). Both

---

[14] (Riley 10-1).

arms were found to be equally proficient in object placement.

To test the accuracy and resolution of a single arm, tacks were placed as close to a target pinpoint as possible. Each arm was able to repeatedly place a tack within one-sixteenth of an inch of the target. This demonstrated the effectiveness of the stereo vision feedback as well as the fine motion control (resolution and accuracy) provided by the master controllers.

To further evaluate the system, coordination between the arms was investigated. This involved passing and moving objects (pencils and paper clips) using both arms. These types of operations require a significant amount of accuracy and resolution. Both resolution and accuracy were easily demonstrated by successfully inserting and extracting a refill from a pen. Compliance was observed to be very useful when handling objects with both end effectors.

The most difficult evaluation test performed involved the disassembly-assembly of an ink pen (which was broken down into four parts). The disassembly required: removing the cap from the pen; switching the pen to the other hand; unscrewing the pen's end; and extracting the ink tube from the pen body. Assembly was performed by reversing these steps. This is representative of a typical assembly operation because it combines the different levels of difficulty into one operation. Pages 61-63 in Appendix D display the results of seven performances. The following information (in minutes) was determined:

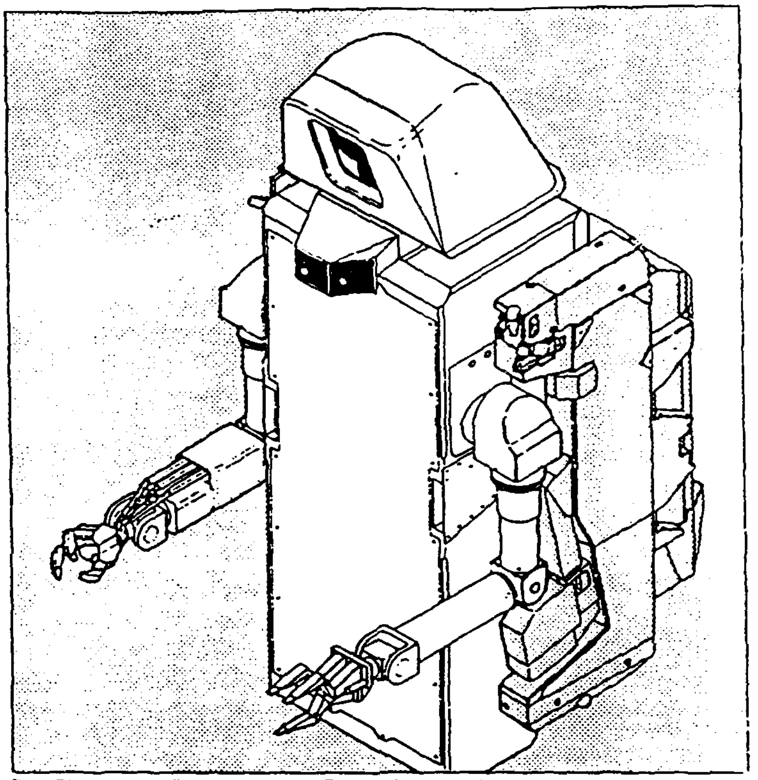
	mean	median	std	range
disassembly:	2.45	2.58	0.44	0.98
assembly:	5.06	4.92	0.83	2.00
total:	7.51	7.75	0.80	2.35

Completing this entire process in less than eight minutes (on average) is a significant accomplishment. In fact, the system's performance is far better than anticipated. The time lag ('tail chase') has proven to be completely acceptable for the scope of this project. The only path error and overshoot/undershoot effects are due to the operator's perception in the work space. These effects were found to be minimal and decrease rapidly with operator training. The overall project performance is a direct result of the degree of telepresence provided by the master controllers and stereo vision feedback.

RECOMMENDED MODIFICATIONS

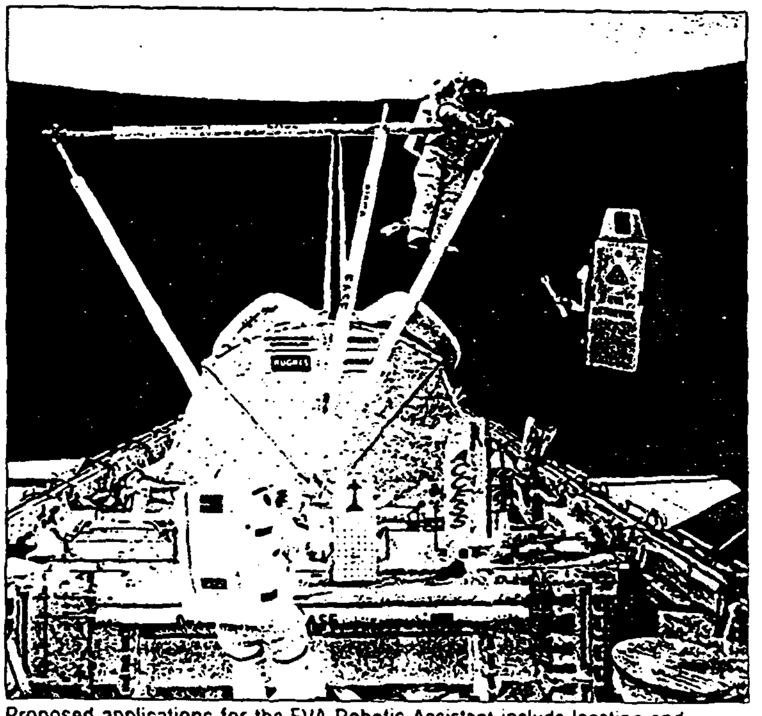
Touch and psuedo-force feedback would be a desirable addition to this project. It would provide the user with an added sense of telepresence, enabling him to perceive objects encountered and forces applied by the slave arm . Increased accuracy and resolution could be achieved by upgrading the master controller joint angle sensors to optical encoders or digital resolvers. The best way to increase the speed of performance and to eliminate the 'tail chase' effect between the master controller and slave arm is to directly access the joint motor drive controllers in the Puma, thus bypassing VAL II control. This would require more PC computing power and speed, which could be provided by upgrading to a 33 Mhz 80486 CPU. Converting the control program from True BASIC to Turbo C would increase the computational speed communication bandwidth between the Puma and PC. Any of these additions to the existing system would increase the operational performance.

ILLUSTRATIONS



The Phase II configuration of the EVA Robotic Assistant.

Figure No. 1



Proposed applications for the EVA Robotic Assistant include locating and retrieving objects. In this hypothetical situation, the EVAR retrieves and returns a wrench to the astronaut.

Figure No. 2

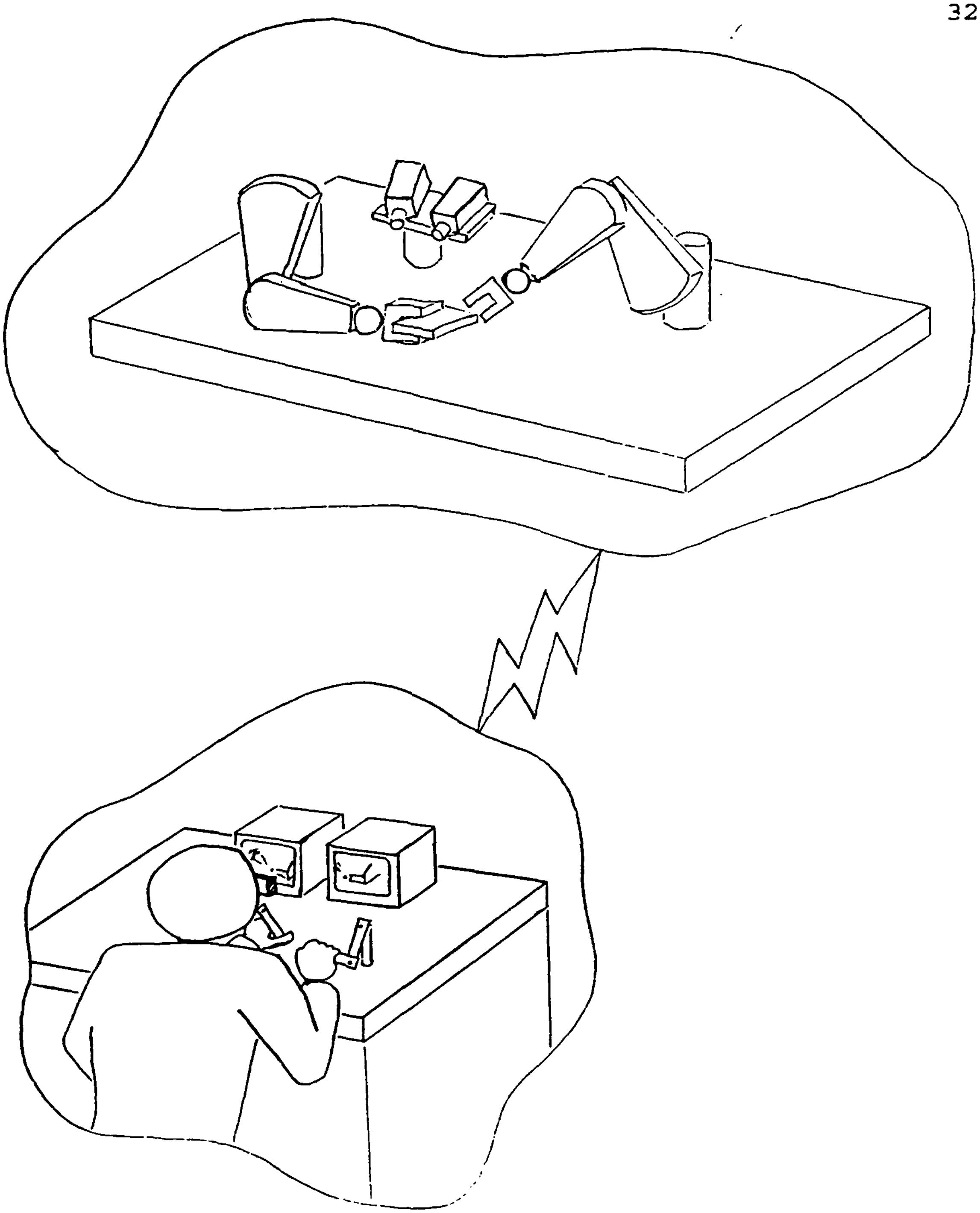


Figure No. 3

General concept of remotely operated telerobotic assembly station using stereo vision feedback.

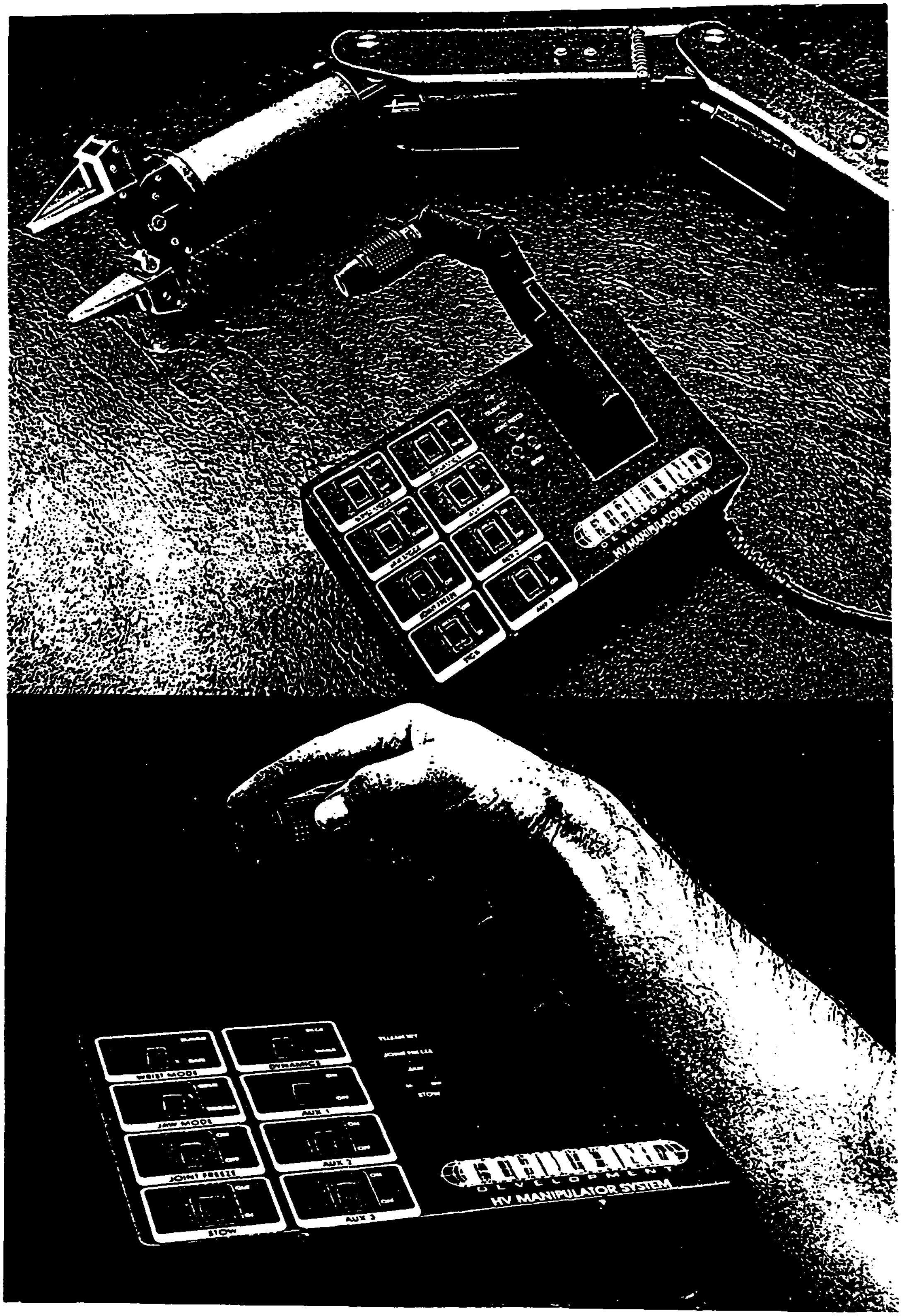


Figure No. 4

Schilling's kinematically similar master controller.

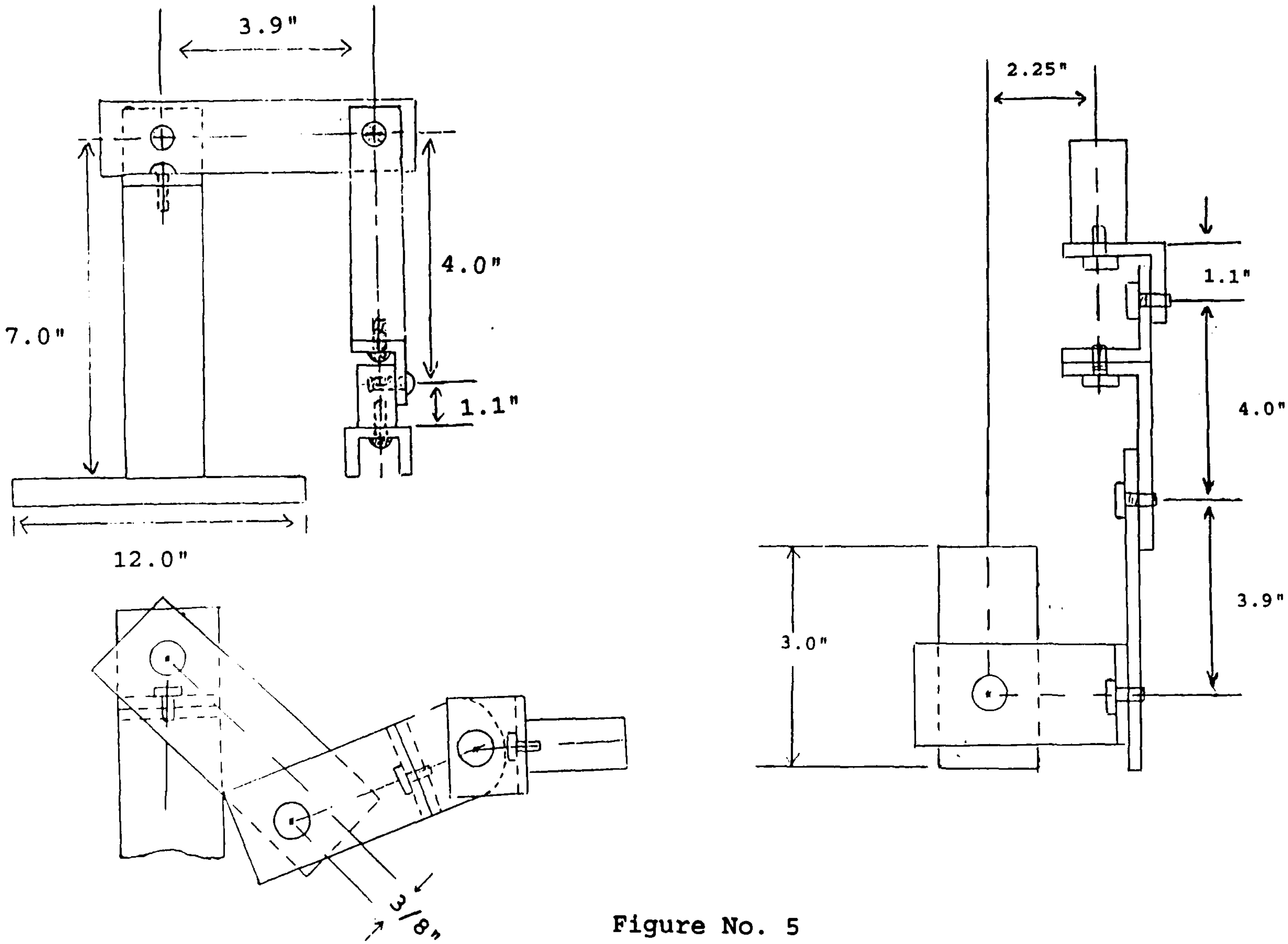


Figure No. 5
Master controller design.

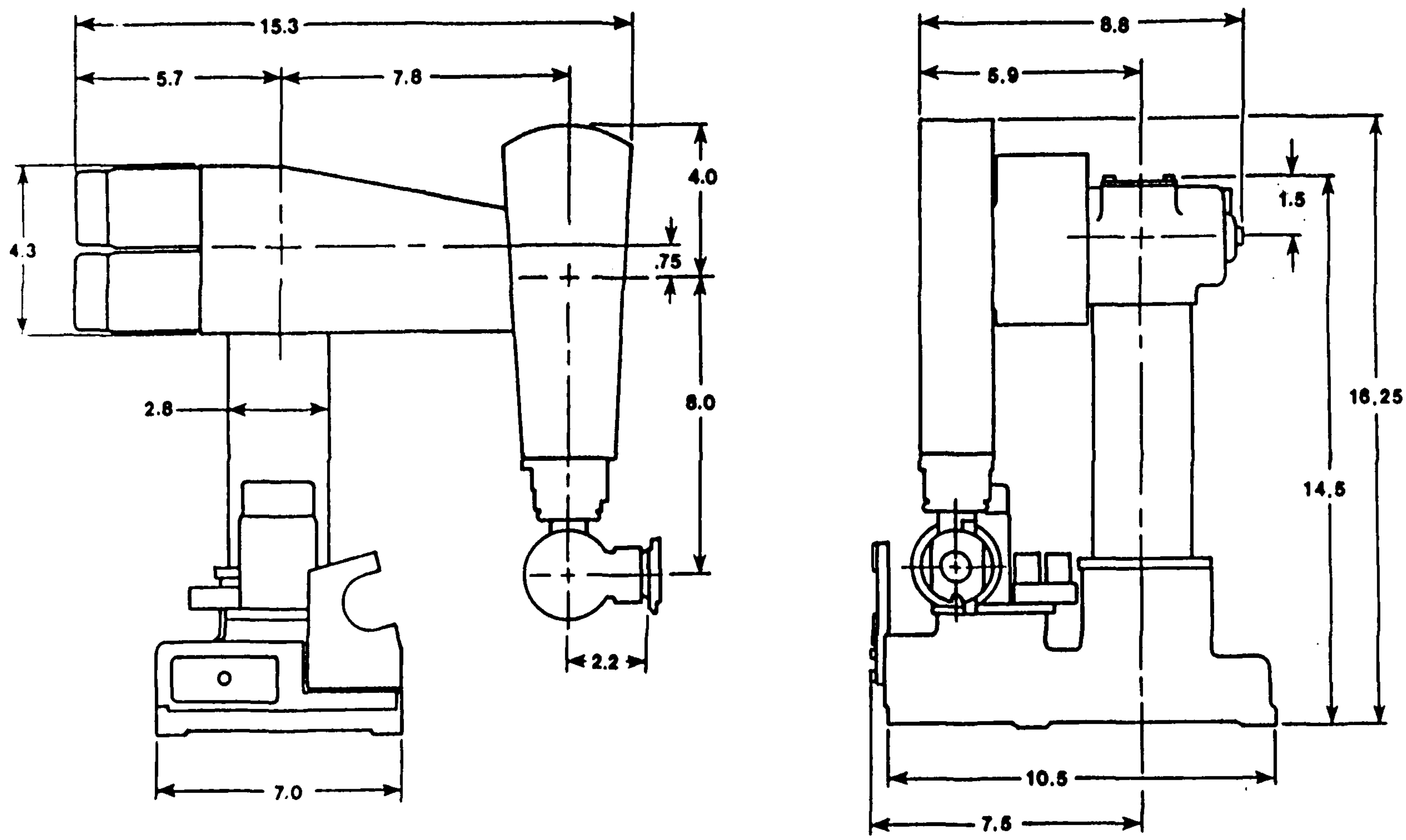


Figure No. 6 PUMA Dimensions - Installation

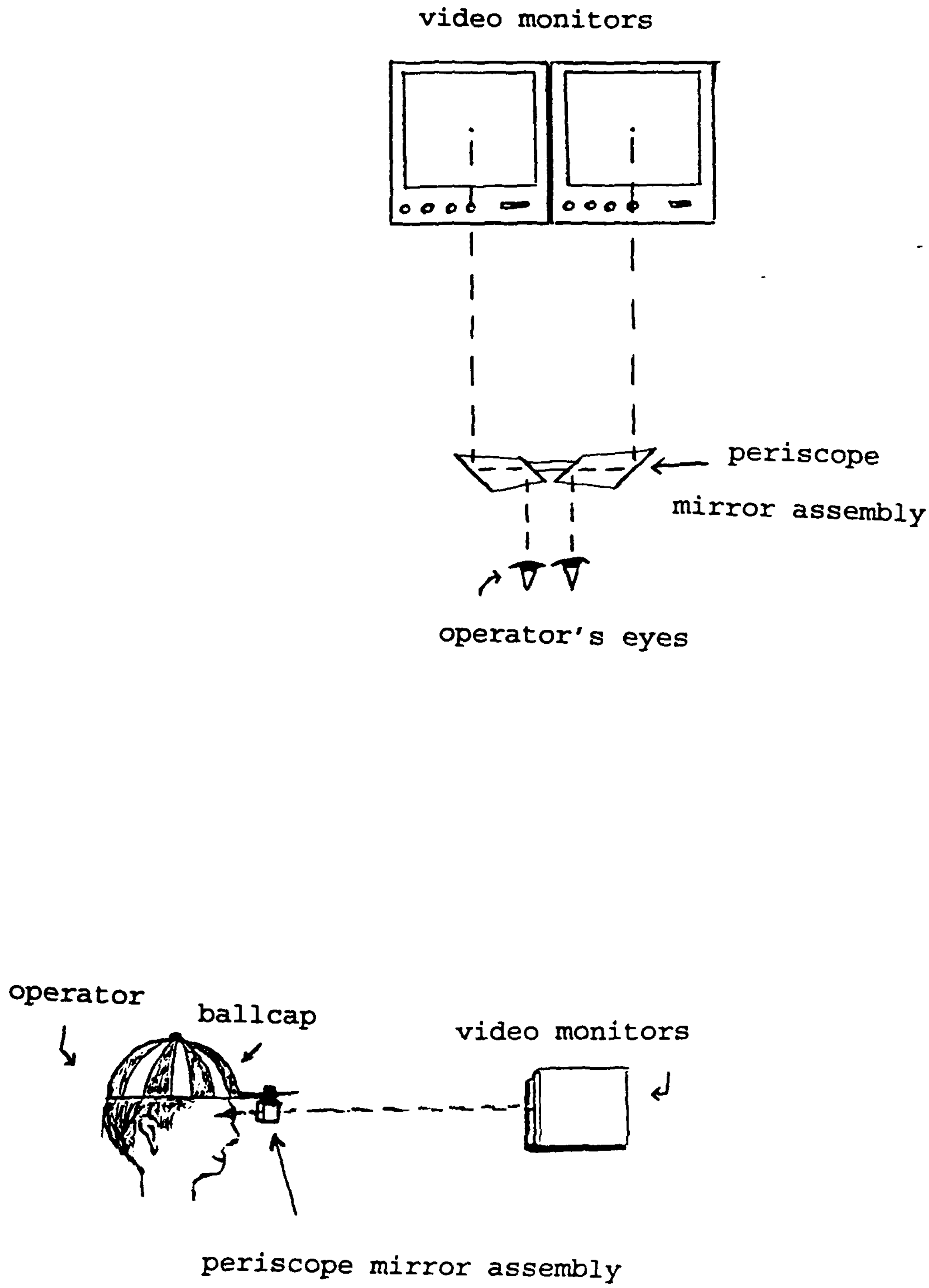


Figure No. 7
periscope mirror assembly

TELEOPERATOR CONTROL LOOP

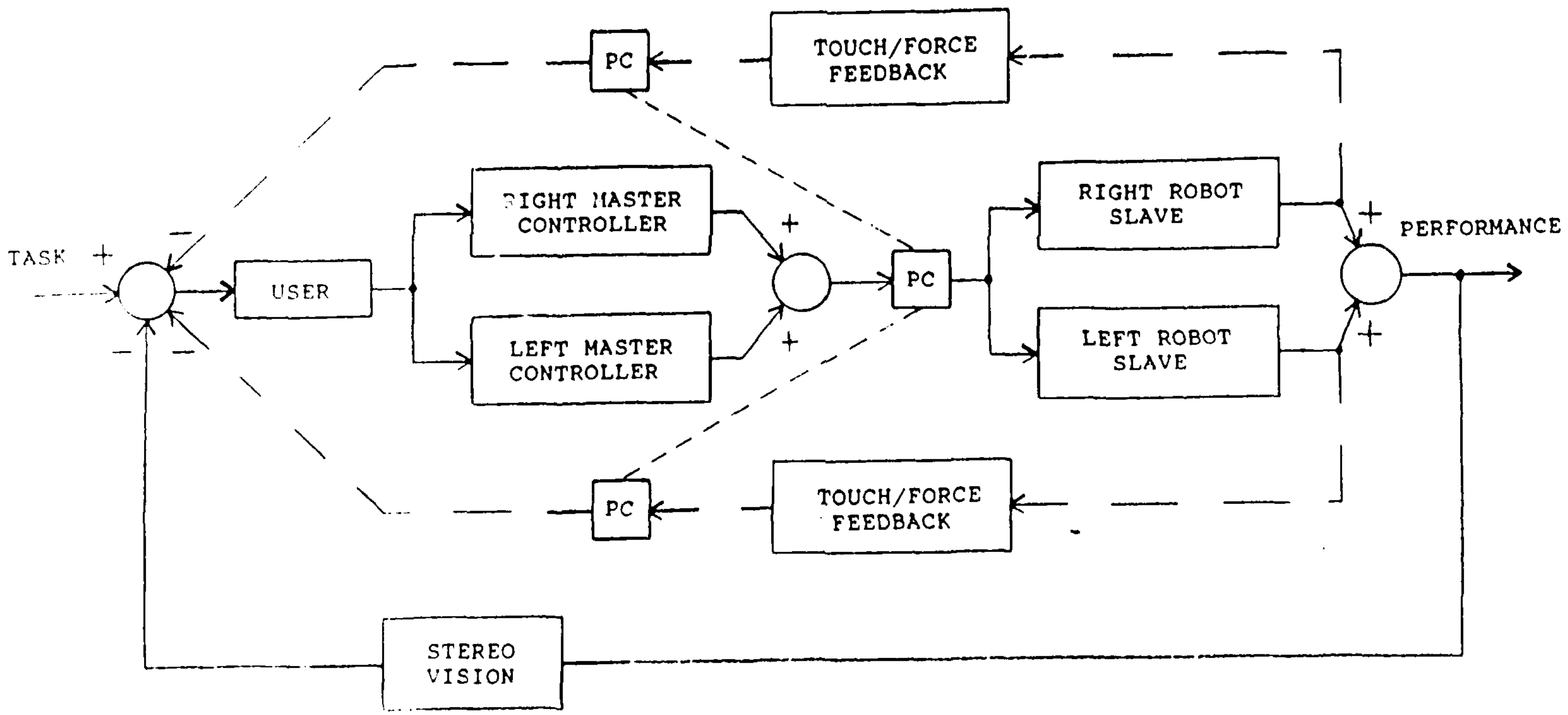
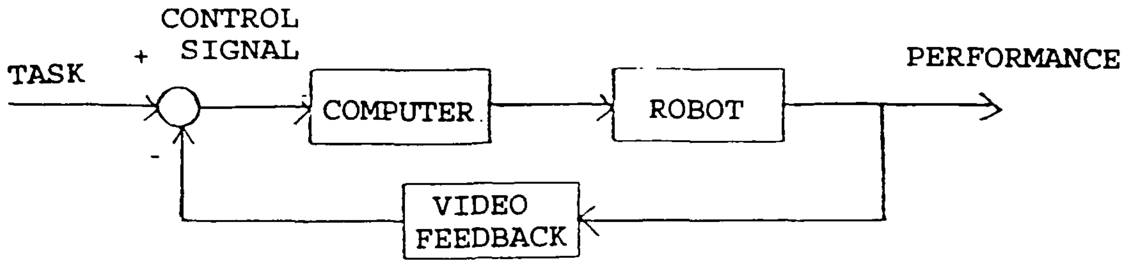
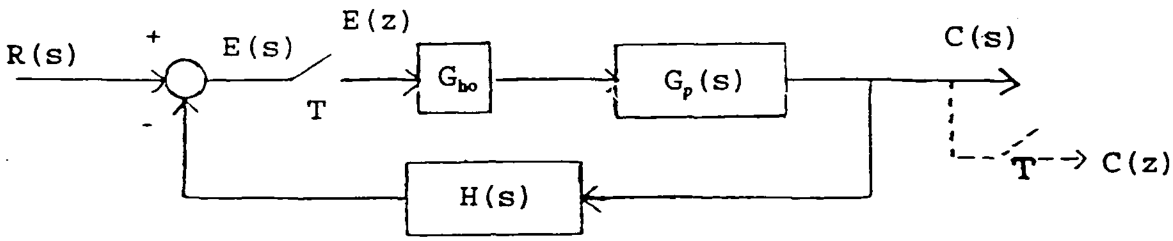


Figure No. 8

BASIC CONTROL FLOW



DETAILED BLOCK DIAGRAM



T = 1/36 second

$G_p(s)=1/(0.001s^2+0.11s+1)$

$H(s)=1/(1+0.1s)$

$G_{bo}(s)=e^{-ST}/s$

$E(z)=[R(s) - E(z)G_{bo}(s)G_p(s)H(s)]^*$

$=R(z) - E(z)\overline{G_{bo}G_pH(z)}$

$=R(z)/(1 + \overline{G_{ho}G_pH(z)})$

$C(s)=E(z)G_{ho}(s)G_p(s)$

$C(z)=E(z)\overline{G_{ho}G_p(z)}$ - open loop

$=R(z)\overline{G_{ho}G_p(z)}/(1 + G_{ho}G_pH(z))$ - closed loop

Open Loop Transfer Function

$$\frac{0.165z+0.062}{z^2-0.82z+0.05}$$

Closed Loop Transfer Function

$$\frac{0.165z^4-0.200z^3+0.012z^2+0.036z-0.02}{z^5-2.38z^4+2.03z^3-0.602z^2+0.060z-0.002}$$

Figure No. 9

BIBLIOGRAPHY

Asada, H., and J.J. Slotine. Robotic Analysis and Control. New York: John Wiley & Sons, Inc., 1986.

Beni, Gerardo, and Susan Hackwood. Recent Advances in Robotics. New York: John Wiley & Sons, Inc., 1985.

Brady, Michael, and Richard Paul, eds. Robotics Research: The First International Symposium. Cambridge: The MIT Press, 1984.

Chae, An H. Model Based Control of a Robotic Manipulator. Cambridge: The MIT Press, 1988.

Chirouze, Michel, and Phillippe Corffet. An Introduction to Robot Technology. London: Kogan Page, 1983.

Craig, John J. Introduction to Robotics: Mechanics and Control. Reeding: Addison-Wesley Publishing Company, Inc., 1986.

Critchlow, Aurthur J. Introduction to Robotics. New York: MacMillan Publishing Company, 1985.

Donald, Bruce R. Error Detection & Recovery in Robotics. Berlin: Springer-Verlag, 1989.

Dorf, Richard C. Robotics and Automated Manufacturing. Reston: Reston Publishing Company Inc., 1983.

Hayati, S.; T. Lee; P. Backes; and J. Lloyd. "A Unified Teleoperated-Autonomous Dual-Arm Robotic System." IEEE Control Systems. February '91.

Hemami, A.; F. Ranjbaran; and R. M. Cheng. "A Case Study of Two-Robot-Arm Workcell Material Handling." Journal Of Robotic Systems v8 p21-37 February '91.

Johnsen and Corliss. Human Factors in Teleoperator Design and Operation. New York: John Wiley & Sons, Inc., 1971.

Klaus, Berthold, and Paul Horn. Robot Vision. Cambridge: The MIT Press, 1986.

Koivo, Antti J. Fundamentals For Control of Robotic Manipulators. New York: John Wiley & Sons, Inc., 1989.

Koren, Yoram. Robotics for Engineers. New York: McGraw-Hill, Inc., 1985.

Martin, H. Lee and Paul E. Satterlee Jr. "Kinematic Redundancy - Key to Dexterous Manipulation in Confined Environments." Robots 11: 17th International Symposium On - Conference Proceedings. Dearborn: Society of Manufacturing Engineers, 1987.

Mason, Matthew T., and J. Kenneth Salisbury, Jr. Robot Hands and the Mechanics of Manipulation. Cambridge: The MIT Press, 1985.

Mauer, Georg F. "A Flexible PC-Based Robot Controller." Robots 13: Conference Proceedings. Dearborn: Society of Manufacturing Engineers, 1987.

McDonald Anthony C. Robot Technology Theory, Design, and Application. Englewood Cliffs: Prentice-Hall, 1986.

Niemeyer, Gunter and Jean-Jacques Slotline. "Stable Adaptive Teleoperation." IEEE Journal Of Oceanic Engineering v16.1 January '91.

Poole, Harry H. Fundamentals of Robotics Engineering. New York: Van Nostrand Reinhold, 1989.

Riley, Donald L. "Robot Calibration and Performance Specification Determination." Robots 11: 17th International Symposium On - Conference Proceedings. Dearborn: Society of Manufacturing Engineers, 1987.

Robillard, Mark J. "Robotics: Tactile Sensing." Radio-Electronics v57 p71-2 August '86.

Ruocco, S. R. Robot Sensors and Transducers. New York: John Wiley & Sons, Inc., 1987.

Scott, Peter, ed. The World Yearbook of Robotics Research and Development. 2nd Ed. London: Kogan Page, 1986.

Stokic, Dragon, and Miomir Vukobratovic. Applied Control of Manipulation Robots: Analysis, Synthesis, and Exercises. Berlin: Springer-Verlag, 1989.

Uttal, William R. "Teleoperators." Scientific American v261 p124-9 December '89.

Wolovich, William A. Robotics: Basic Analysis & Design. New York: CBS College Publishing, 1987.

Young, John F. Robotics. New York: John Wiley & Sons, Inc., 1973.

Zeldman, Maurice I. What Every Engineer Should Know About Robotics. New York: Marcel Dekker, Inc., 1984.

APPENDICES

```
! ******************************************************************
! PROGRAM TELEROB2.TRU CONTROLS TWO PUMA ROBOT ARMS  -- R. L. Dewitt
! ******************************************************************

LIBRARY "comlib.trc"

! -----------------------------------------------------------------
!  section divides screen into three sections for monitoring
! -----------------------------------------------------------------

OPEN #1: screen 0, 1, 0, 1              ! entire screen
OPEN #2: screen 0, 1, .55, 1            ! upper window
OPEN #3: screen .005, .995, .56, .99    ! inside upper window
OPEN #4: screen 0, 1, 0, .45            ! lower window
OPEN #5: screen .005, .995, .01, .44    ! inside lower window
OPEN #6: screen 0, 1, .455, .545        ! reference window
OPEN #7: screen .005, .995, .465,.535   ! inside reference window
SET MODE "egahires"

WINDOW #2                          ! upper window
SET COLOR 9                        ! bright blue
BOX AREA 0, 1, 0, 1

WINDOW #3                          ! inside upper window
SET COLOR "black"
BOX AREA 0, 1, 0, 1
SET COLOR 15                       ! bright white

WINDOW #4                          ! lower window
SET COLOR 9                        ! bright blue
BOX AREA 0, 1, 0, 1

WINDOW #5                          ! inside lower window
SET COLOR "black"
BOX AREA 0, 1, 0, 1
SET COLOR 15                       ! bright white

WINDOW #6                          ! reference window
SET COLOR 12                       ! bright red
BOX AREA 0, 1, 0, 1

WINDOW #7                          ! inside reference window
SET COLOR "black"
BOX AREA 0, 1, 0, 1
SET COLOR 15                       ! bright white

! -----------------------------------------------------------------
! initialization section - test checks if the position (pos)
!   bit is set in the data word (num)
! -----------------------------------------------------------------

OPTION ANGLE degrees

DEF test(num,pos) = mod(num,2^(pos+1))

DIM thetas(12)                     ! holds old joint angles

MAT thetas = zer(12)               ! set old joints to zero
LET sw = 0                         ! switch toggle
LET robot = 1                      ! robot indentity (default right)
LET start = -1                     ! transfer set-up toggle (current arm)
```

```
LET robot = 1                        ! robot identity (default right)
LET start = -1                       ! transfer set-up toggle (current arm)
LET start2 = -1                      ! transfer send toggle (current arm)
LET start2_L = -1                    ! data transfer toggle (left arm)
LET start2_R = -1                    ! data transfer toggle (right arm)
LET close_R = -1                     ! close RIGHT gripper
LET open_R = 1                       ! open RIGHT gripper
LET close_L = -1                     ! close LEFT gripper
LET open_L = 1                       ! open LEFT gripper
LET jump = -1                        ! initialization toggle
LET factor = 26                      ! initial guess for angle conversion
LET rtn$=chr$(13)
LET msg$=""

! -----------------------------------------------------------------
! screen graphics routine
! -----------------------------------------------------------------

WINDOW #1
SET CURSOR 13,2
SET COLOR 15                             ! bright white
LET htky$ = " <ESC>Quit <F1-F4>Panic <F5>Init <F6>Cal"
LET htky$ = htky$&" <F7>Start Pts <F8>Switch <F9>Ready"
PRINT htky$

WINDOW #5                                ! right arm (default)
CALL com_open (#8, 1, 9600, " ")         ! open com1 at 9600 baud
CALL com_open (#9, 2, 9600, " ")         ! open com2 at 9600 baud
CALL com_switch (1)

! +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!                        BEGIN MAIN LOOP
! +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

DO
   CALL POKE(-749,15)
   DO
   LOOP until (test(PEEK(-749),2) > 3)      ! looking for a 4
   CALL POKE(-749,1)
   FOR ch = 0 to 14                         ! read 15 channels
       LET index = ch + 1
       DO
       LOOP until (test(PEEK(-749),2) > 3)  ! looking for a 4
       CALL POKE(-749,12)
       DO
       LOOP until (test(PEEK(-749),1) <> 2) ! looking for a 2'
       CALL POKE(-748,0)
       DO
       LOOP until (test(PEEK(-749),1) <> 2) ! looking for a 2'
       CALL POKE(-748,ch)
       DO                                   ! looking for a 5
       LOOP until ((test(PEEK(-749),2) > 0) and (test(PEEK(-749),0) >0))
       LET low = PEEK(-748)
       DO                                   ! looking for a 5
       LOOP until ((test(PEEK(-749),2) > 0) and (test(PEEK(-749),0) >0))
       LET high = PEEK(-748)
       LET data = 256 * high + low          ! put bytes together
       IF data > 32767 then LET data = data - 65536
       LET voltage = ROUND(10*data/4096,2)  ! change to real value
       LET theta = voltage*factor           ! change to degrees
```

```
LET theta = ROUND(theta,1)          ! round for output to robot

SELECT CASE index
CASE 1
     LET theta = -(theta - 14.4)
     IF abs(theta) > 180.0 then
        LET j1R = theta - sgn(theta) * 360
        LET j1R = ROUND(j1R,1)
     ELSE
        LET j1R = ROUND(theta,1)
     END IF
     LET diff = abs(j1R) - abs(thetas(index))
     IF abs(diff) > 0.5 then
        LET thetas(index) = j1R
     ELSE
        LET j1R = thetas(index)
     END IF
CASE 2
     LET theta = theta - 316.2
     IF abs(theta) > 180.0 then
        LET j2R = theta - sgn(theta) * 360
        LET j2R = ROUND(j2R,1)
     ELSE
        LET j2R = ROUND(theta,1)
     END IF
     LET diff = abs(j2R) - abs(thetas(index))
     IF abs(diff) > 0.5 then
        LET thetas(index) = j2R
     ELSE
        LET j2R = thetas(index)
     END IF
CASE 3
     LET theta = -(theta - 222.5)
     IF abs(theta) > 180.0 then
        LET j3R = theta - sgn(theta) * 360
        LET j3R = ROUND(j3R,1)
     ELSE
        LET j3R = ROUND(theta,1)
     END IF
     LET diff = abs(j3R) - abs(thetas(index))
     IF abs(diff) > 0.5 then
        LET thetas(index) = j3R
     ELSE
        LET j3R = thetas(index)
     END IF
CASE 4
     LET theta = -(theta - 107.3)
     IF abs(theta) > 180.0 then
        LET j4R = theta - sgn(theta) * 360
        LET j4R = ROUND(j4R,1)
     ELSE
        LET j4R = ROUND(theta,1)
     END IF
     LET diff = abs(j4R) - abs(thetas(index))
     IF abs(diff) > 0.5 then
        LET thetas(index) = j4R
     ELSE
        LET j4R = thetas(index)
     END IF
CASE 5
     LET theta = -(theta - 136.0)
     IF abs(theta) > 180.0 then
```

```
            LET j5R = theta - sgn(theta) * 360
            LET j5R = ROUND(j5R,1)
         ELSE
            LET j5R = ROUND(theta,1)
         END IF
         LET diff = abs(j5R) - abs(thetas(index))
         IF abs(diff) > 0.5 then
            LET thetas(index) = j5R
         ELSE
            LET j5R = thetas(index)
         END IF
   CASE 6
         LET theta = -(theta - 177.3)
         IF abs(theta) > 180.0 then
            LET j6R = theta - sgn(theta) * 360
            LET j6R = ROUND(j6R,1)
         ELSE
            LET j6R = ROUND(theta,1)
         END IF
         LET diff = abs(j6R) - abs(thetas(index))
         IF abs(diff) > 0.5 then
            LET thetas(index) = j6R
         ELSE
            LET j6R = thetas(index)
         END IF
   CASE 7
         LET theta = -(theta + 203.0)
         IF abs(theta) > 180.0 then
            LET j1L = theta - sgn(theta) * 360
            LET j1L = ROUND(j1L,1)
         ELSE
            LET j1L = ROUND(theta,1)
         END IF
         LET diff = abs(j1L) - abs(thetas(index))
         IF abs(diff) > 0.5 then
            LET thetas(index) = j1L
         ELSE
            LET j1L = thetas(index)
         END IF
   CASE 8
         LET theta = -(theta - 129.0)
         IF abs(theta) > 180.0 then
            LET j2L = theta - sgn(theta) * 360
            LET j2L = ROUND(j2L,1)
         ELSE
            LET j2L = ROUND(theta,1)
         END IF
         LET diff = abs(j2L) - abs(thetas(index))
         IF abs(diff) > 0.5 then
            LET thetas(index) = j2L
         ELSE
            LET j2L = thetas(index)
         END IF
   CASE 9
         LET theta = -(theta - 217.0)
         IF abs(theta) > 180.0 then
            LET j3L = theta - sgn(theta) * 360
            LET j3L = ROUND(j3L,1)
         ELSE
            LET j3L = ROUND(theta,1)
         END IF
         LET diff = abs(j3L) - abs(thetas(index))
```

```
      IF abs(diff) > 0.5 then
         LET thetas(index) = j3L
      ELSE
         LET j3L = thetas(index)
      END IF
CASE 10
      LET theta = -(theta - 145.2)
      IF abs(theta) > 180.0 then
         LET j4L = theta - sgn(theta) * 360
         LET j4L = ROUND(j4L,1)
      ELSE
         LET j4L = ROUND(theta,1)
      END IF
      LET diff = abs(j4L) - abs(thetas(index))
      IF abs(diff) > 0.5 then
         LET thetas(index) = j4L
      ELSE
         LET j4L = thetas(index)
      END IF
CASE 11
      LET theta = -(theta - 142.9)
      IF abs(theta) > 180.0 then
         LET j5L = theta - sgn(theta) * 360
         LET j5L = ROUND(j5L,1)
      ELSE
         LET j5L = ROUND(theta,1)
      END IF
      LET diff = abs(j5L) - abs(thetas(index))
      IF abs(diff) > 0.5 then
         LET thetas(index) = j5L
      ELSE
         LET j5L = thetas(index)
      END IF
CASE 12
      LET theta = -(theta - 177.3)
      IF abs(theta) > 180.0 then
         LET j6L = theta - sgn(theta) * 360
         LET j6L = ROUND(j6L,1)
      ELSE
         LET j6L= ROUND(theta,1)
      END IF
      LET diff = abs(j6L) - abs(thetas(index))
      IF abs(diff) > 0.5 then
         LET thetas(index) = j6L
      ELSE
         LET j6L = thetas(index)
      END IF
CASE 13
      LET factor = 253 / voltage
CASE 14
      IF voltage < 1 and close_R = -1 then
         LET close_R = 1
         LET flagR = 1
      ELSEIF voltage > 3 and close_R = 1 then
         LET close_R = -1
         LET flagR = -1
      ELSE
         LET flagR = 0
      END IF
CASE 15
      IF voltage < 1 and close_L = -1 then
         LET close_L = 1
```

```
            LET flagL = 1
         ELSEIF voltage > 3 and close_L = 1 then
            LET close_L = -1
            LET flagL = -1
         ELSE
            LET flagL = 0
         END IF
    CASE else
         LET theta = theta
    END SELECT

    CALL terminal(msg$)
    CALL keyboard(cmd$)

NEXT ch

! ------------------------------------------------------------------
! toggling routines
! ------------------------------------------------------------------

IF jump = 1 then                          ! sends to current robot
   CALL send("dis int"&rtn$)
   CALL send("do coarse always"&rtn$)
   CALL send("where"&rtn$)
   CALL send("do intoff"&rtn$)
   LET jump = -jump
END IF

CALL terminal(msg$)

IF start = 1 then                         ! sends to current robot
   CALL send("en int"&rtn$)
   CALL send("point #next"&rtn$)
   LET start = -start
END IF
IF start2 = 1 then                        ! sends to current robot
   IF robot = 1 then
      LET start2_R = -start2_R
   ELSE
      LET start2_L = -start2_L
   END IF
   LET start2 = -start2
END IF

CALL terminal(msg$)

IF start2_L = 1 then
   CALL Lpts(j1L,j2L,j3L,j4L,j5L,j6L,flagL)
END IF

CALL terminal(msg$)

IF start2_R = 1 then
   CALL Rpts(j1R,j2R,j3R,j4R,j5R,j6R,flagR)
END IF

CALL terminal(msg$)

IF sw = 1 then
   CALL switch
   LET sw = 0
END IF
```

```
LOOP

! ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
!                        END OF MAIN LOOP
! ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
! ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

SUB Lpts(j1L,j2L,j3L,j4L,j5L,j6L,flagL)
   LET L_order$ = str$(j1L)&","&str$(j2L)&","
   LET L_order$ = L_order$&str$(j3L)&","&str$(j4L)&","&str$(j5L)
   LET L_order$ = L_order$&","&str$(j6L)&","&str$(flagL)
   IF robot = -1 then
      CALL send(L_order$&rtn$)
      CALL terminal(msg$)
   ELSE
      CALL switch
      CALL send(L_order$&rtn$)
      SET COLOR 15
      CALL terminal(msg$)
      CALL switch
      SET COLOR 14
   END IF
END SUB

SUB Rpts(j1R,j2R,j3R,j4R,j5R,j6R,flagR)
   LET R_order$ = str$(j1R)&","&str$(j2R)&","
   LET R_order$ = R_order$&str$(j3R)&","&str$(j4R)&","&str$(j5R)
   LET R_order$ = R_order$&","&str$(j6R)&","&str$(flagR)
   IF robot = 1 then
      CALL send(R_order$&rtn$)
      CALL terminal(msg$)
   ELSE
      CALL switch
      CALL send(R_order$&rtn$)
      SET COLOR 15
      CALL terminal(msg$)
      CALL switch
      SET COLOR 14
   END IF
END SUB

! ----------------------------------------------------------------
! subroutine "terminal" checks for messages from robot and then
! outputs them to the screen
! ----------------------------------------------------------------

SUB terminal(msg$)
    CALL receive(msg$)                  ! get any message from robot
    CALL output(msg$)
END SUB

! ----------------------------------------------------------------
! subroutine "keyboard" has definitions for certain 'hot keys' and
! allows the user to build messages to send to either robot
! ----------------------------------------------------------------

SUB keyboard(cmd$)
    IF key input then                   ! get anything typed by user
       GET KEY k
       SELECT CASE k
       CASE 27                          ! ESC -- quit session
```

```
            PRINT "Session terminated. <Hit Any Key>."
            STOP
        CASE 315                           ! F1 - send 'panic' to robots
            CALL send_both("panic")
        CASE 316                           ! F2 - send 'panic' to robots
            CALL send_both("panic")
        CASE 317                           ! F3 - send 'panic' to robots
            CALL send_both("panic")
        CASE 318                           ! F4 - send 'panic' to robots
            CALL send_both("panic")
        CASE 319                           ! F5 - Initialization toggle
            LET jump = -jump
        CASE 320                           ! F6 - Calibrate
            CALL send_both("calibrate")
        CASE 321                           ! F7 - Start sending points
            LET start2 = -start2
        CASE 322                           ! F8 - Switch to other robot
            LET sw = 1
        CASE 323                           ! F9 - Do Ready
             CALL send_both("do ready")
        CASE 13                            ! CR
            LET cmd$ = cmd$&chr$(k)
            CALL send(cmd$)                ! send line to robot
            LET cmd$ = ""
        CASE else
            LET cmd$ = cmd$&chr$(k)
        END SELECT
    END IF
END SUB

! ------------------------------------------------------------------
! subroutine "output" prints response string (msg$) on crt screen
! and handles CR & LF characters
! ------------------------------------------------------------------

SUB output(msg$)
    DO                                     ! first strip all CRs
       LET i = Pos(msg$,rtn$)              ! find first CR
       IF i = 0 then EXIT DO               ! none - all done
       LET msg$[i:i] = " "                 ! remove the CR
    LOOP
    DO                                     ! end line on LF
       LET i = Pos(msg$,Chr$(10))          ! find next LF
       IF i = 0 then EXIT DO               ! none - all done
       PRINT msg$[1:i-1]                   ! print each separate line
       LET msg$ = msg$[i+1:maxnum]         ! remove that line
       LET count = count + 1
       IF count = 2 then
          LET whr$ = msg$[1:i-1]           ! stores robot joint angles
       END IF
    LOOP
    PRINT msg$;
END SUB

! ------------------------------------------------------------------
! subroutine "send_both" immediately sends word$ to both robots
! ------------------------------------------------------------------

SUB send_both(word$)
    CALL send(rtn$)
    CALL send(word$&rtn$)
    CALL terminal(msg$)
```

```
    CALL switch
    CALL send(rtn$)
    CALL send(word$&rtn$)
    CALL terminal(msg$)
    CALL switch
    LET start2_L = -1
    LET start2_R = -1
    CALL terminal(msg$)
    CALL switch
    CALL terminal(msg$)
    CALL switch
END SUB

! ------------------------------------------------------------------
! subroutine "switch" switches robots
! ------------------------------------------------------------------

SUB switch
    IF robot = 1 then
       CALL com_switch (2)
       WINDOW #3
       SET COLOR 14
       LET robot = -robot
    ELSE
       CALL com_switch (1)
       WINDOW #5
       SET COLOR 14
       LET robot = -robot
    END IF
END SUB

END
```

VAL II PROGRAM - ROBRIGHT (ALSO ROBLEFT)

```
      HERE #NEXT                                 - initializes position
10    MOVE #NEXT                                 - begin matching positions
      PROMPT "?",J1,J2,J3,J4,J5,J6,X                   - get new position
      IF X>0 GOTO 20                                   - close gripper?
      IF X<0 GOTO 30                                   - open gripper?
      SET #NEXT = #PPOINT(J1,J2,J3,J4,J5,J6)               - declare #NEXT
      GOTO 10       - loop
20    CLOSEI        - close gripper
      GOTO 10       - loop
30    OPENI         - open gripper
      GOTO 10       - loop
```

Appendix C

STEREO VISION CONTROLLED BILATERAL TELEROBOTIC REMOTE ASSEMBLY STATION

START-UP AND RUN PROCEDURES:
(from True BASIC environment)

TURN ON POWER TO PUMA ROBOTS AND CAMERA SYSTEM

<F12> - automatic screen color key
<F2> - go to command window
OLD DEWITT\TELEROB2 <RTN> - retrieve program
<F9> - run program

<RTN> - start communication with right arm
N <RTN> - no, do not want to load VAL
N <RTN> - no, do not want to initialize
PRESS ARM POWER TO RIGHT ARM

<F8> - switch to LEFT arm

<RTN> - start communication with left arm
N <RTN> - no, do not load VAL
N <RTN> - no, do not initialize
PRESS ARM POWER TO LEFT ARM

<F5> - initializes both arms
<F6> - calibrates both arms

LEFT ARM |
EX ROBLEFT <RTN> - starts VAL program
<F7> - send joint commands to left arm

<F8> - switch to RIGHT arm

RIGHT ARM |
EX ROBRIGHT <RTN> - starts VAL program
<F7> - send joint commands to right arm

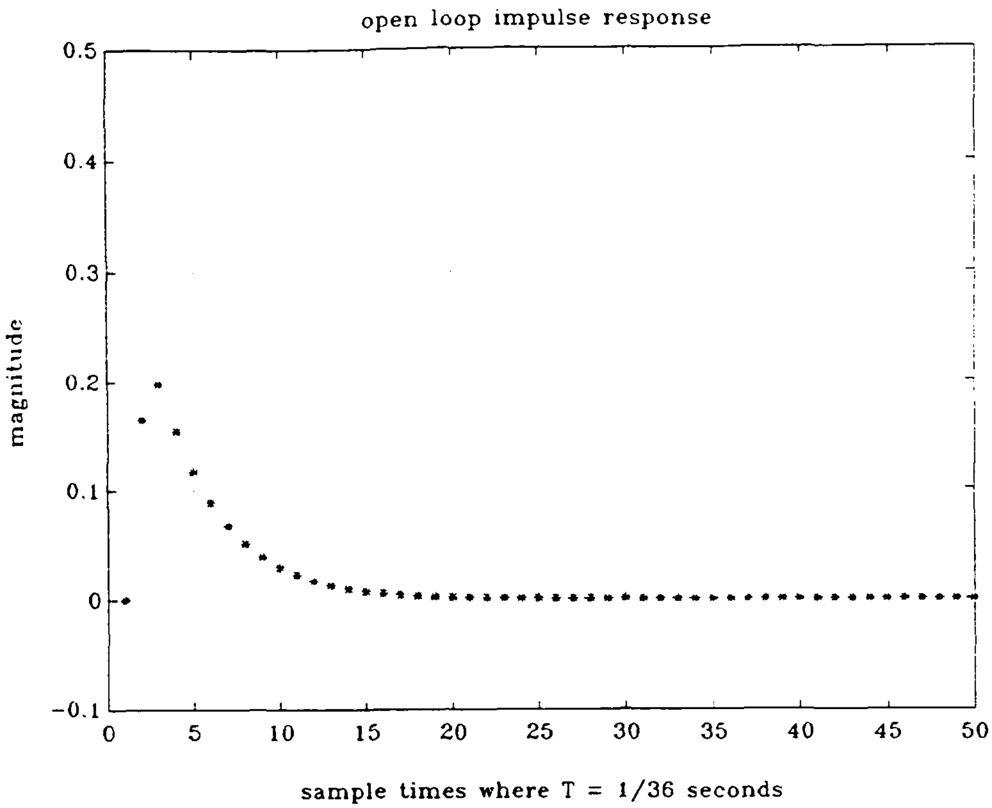
* keyboard entry will now be to <u>RIGHT</u> arm

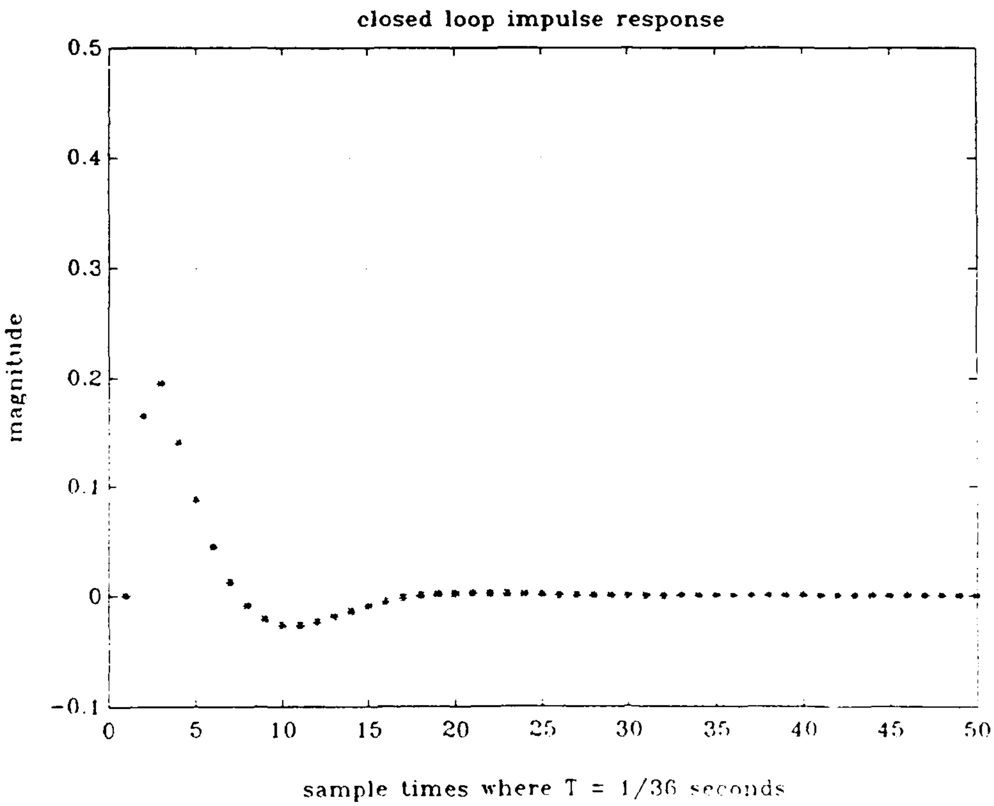
- <F7> is a toggle for sending joint commands
- <F8> will switch between arms
- <F9> will send both arms to ready position after programs have been halted.[15]

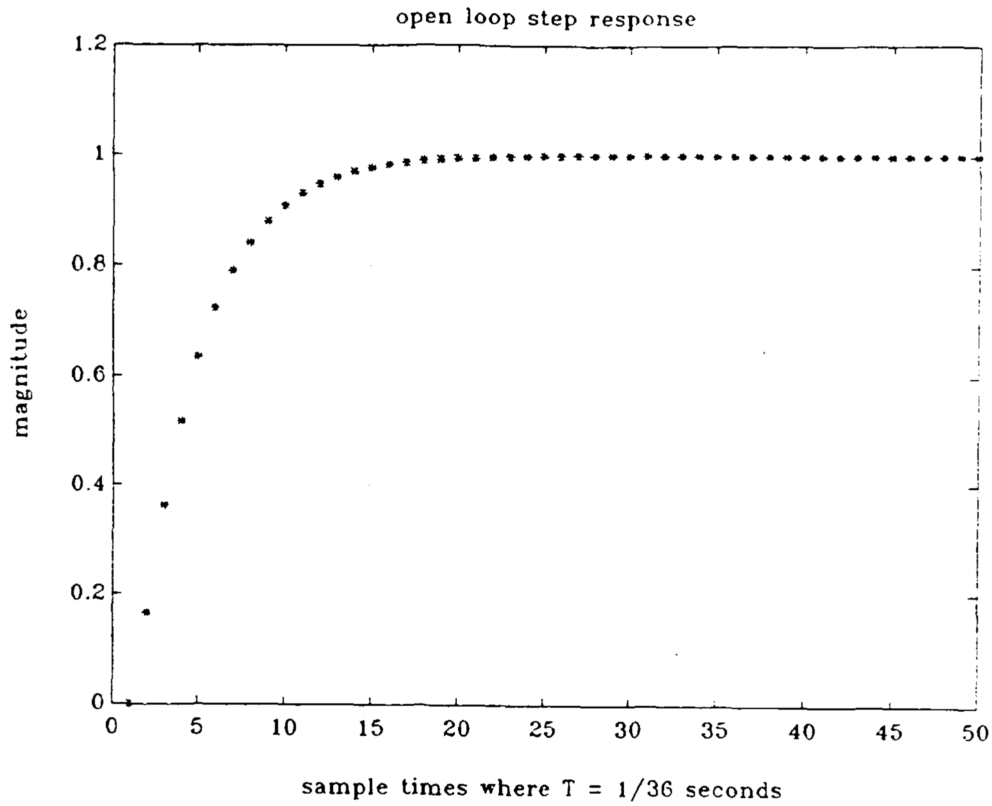
---

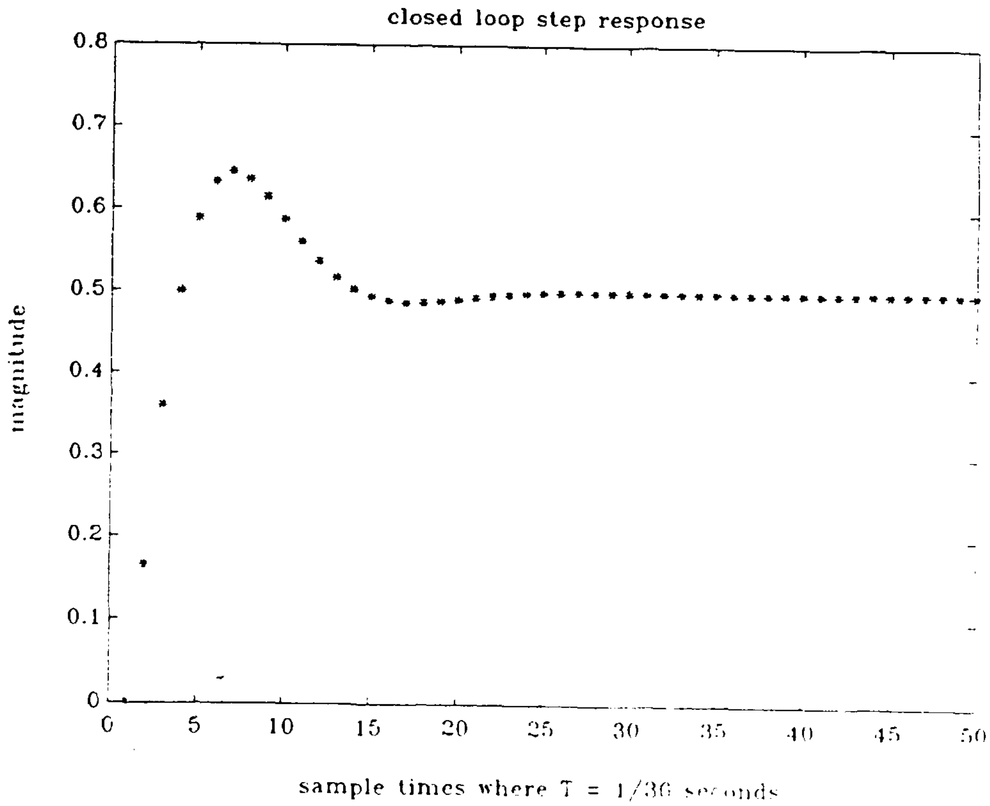
[15] *** To halt program :
- press <F7> to stop sending points
- press <a> followed by <RTN> to abort run

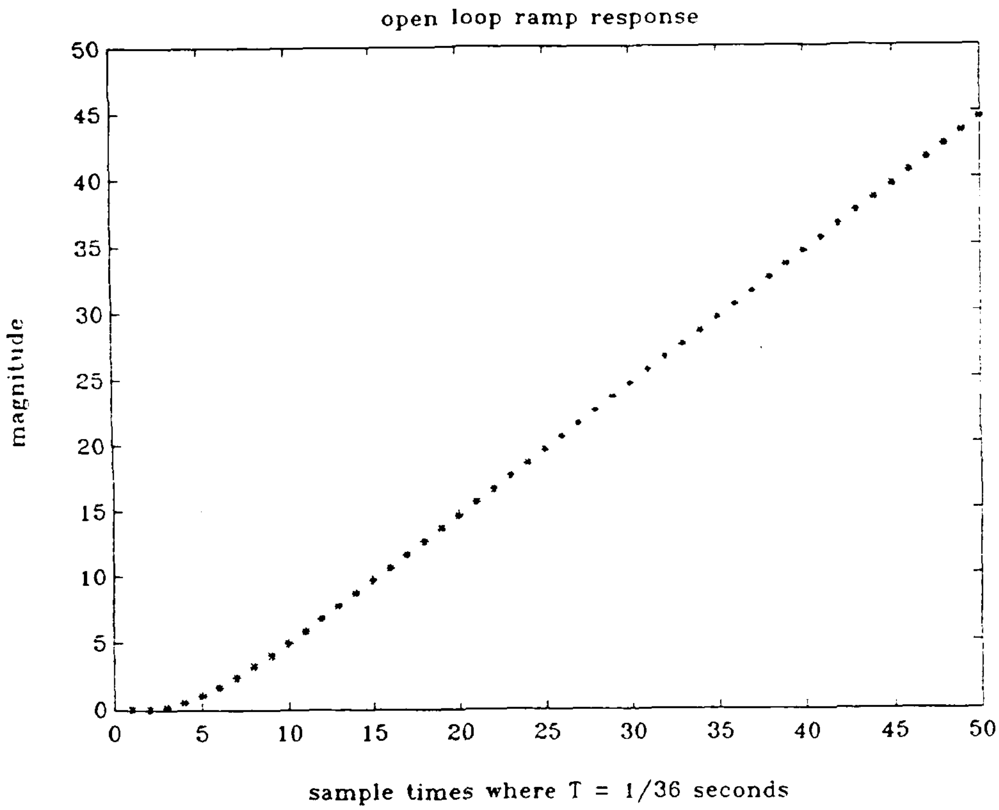
Appendix D - MATLAB Plots

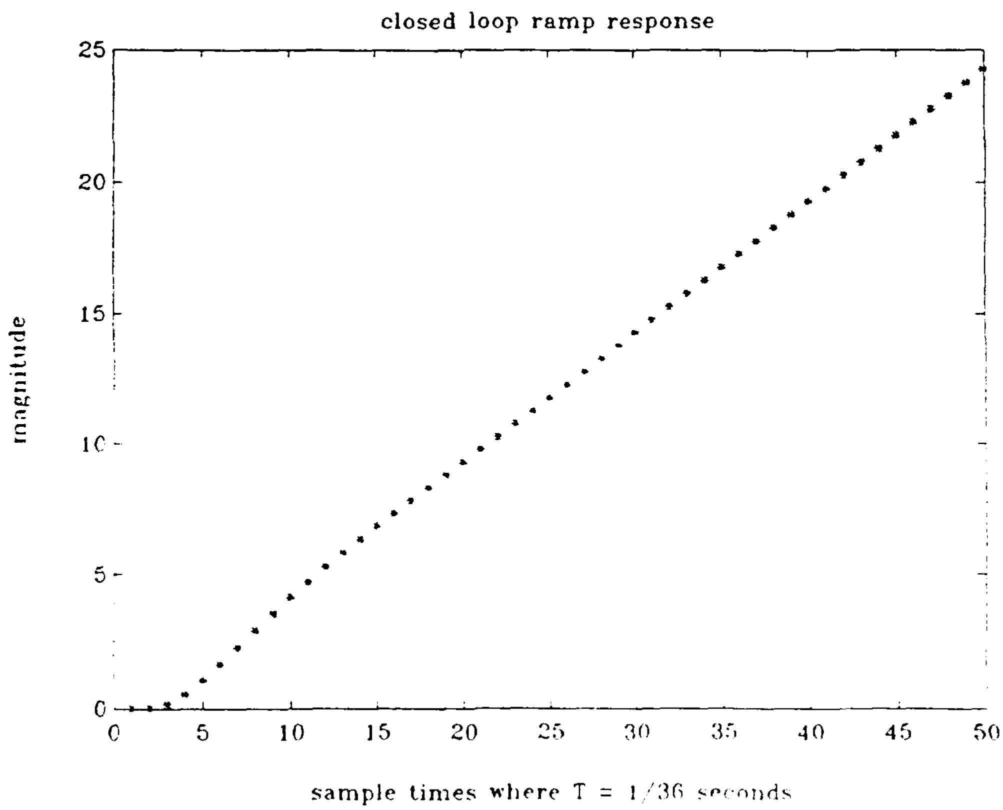

open loop impulse response
magnitude
0.5
0.4
0.3
0.2
0.1
0
-0.1
0
5
10
15
20
25
30
35
40
45
50
sample times where T = 1/36 seconds

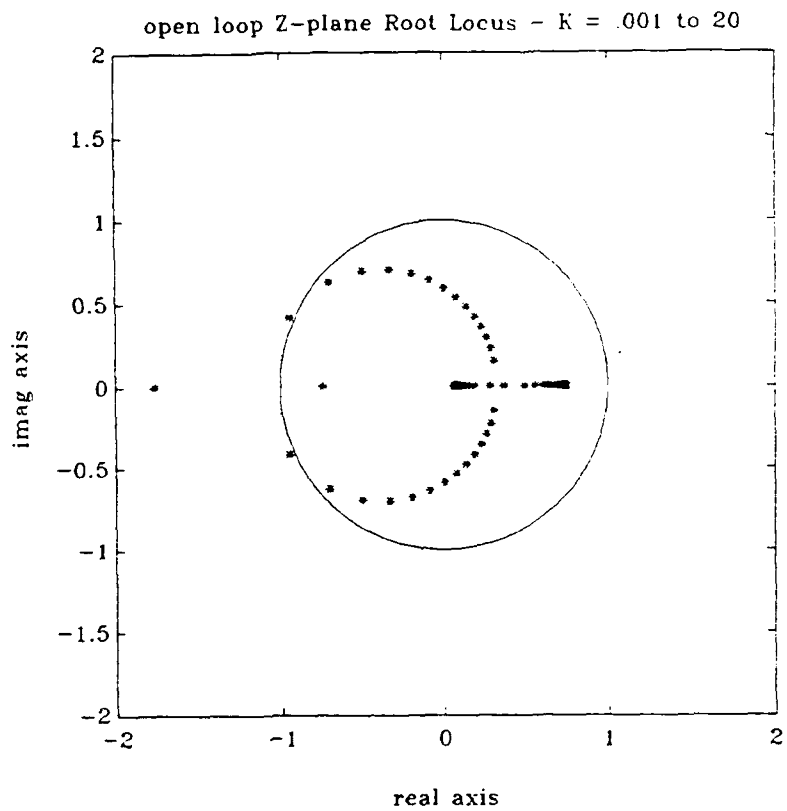

closed loop impulse response
magnitude
0.5
0.4
0.3
0.2
0.1
0
-0.1
0
5
10
15
20
25
30
35
40
45
50
sample times where T = 1/36 seconds

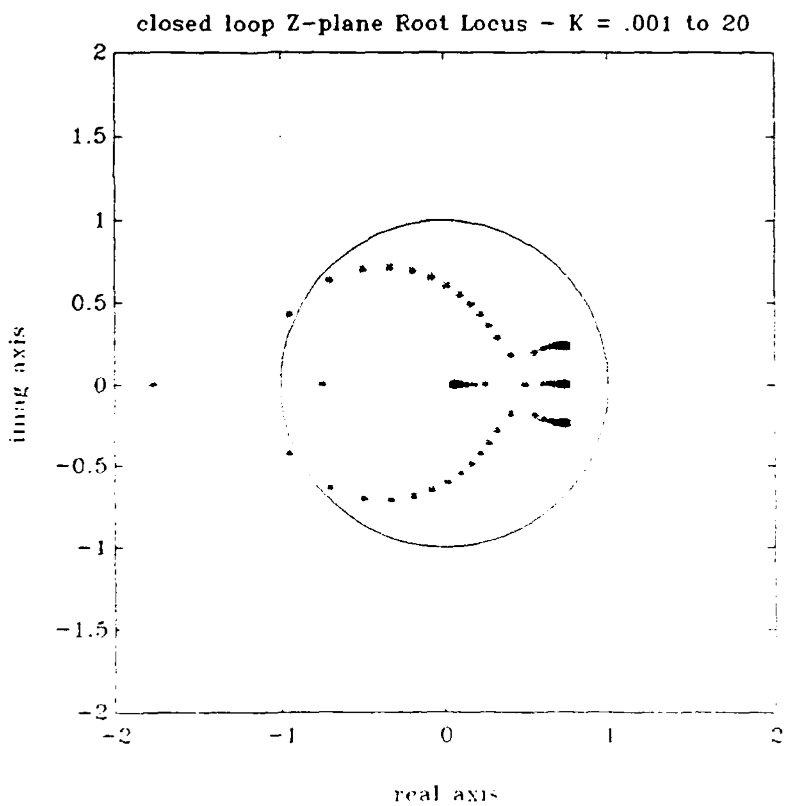

open loop step response
magnitude
sample times where T = 1/36 seconds

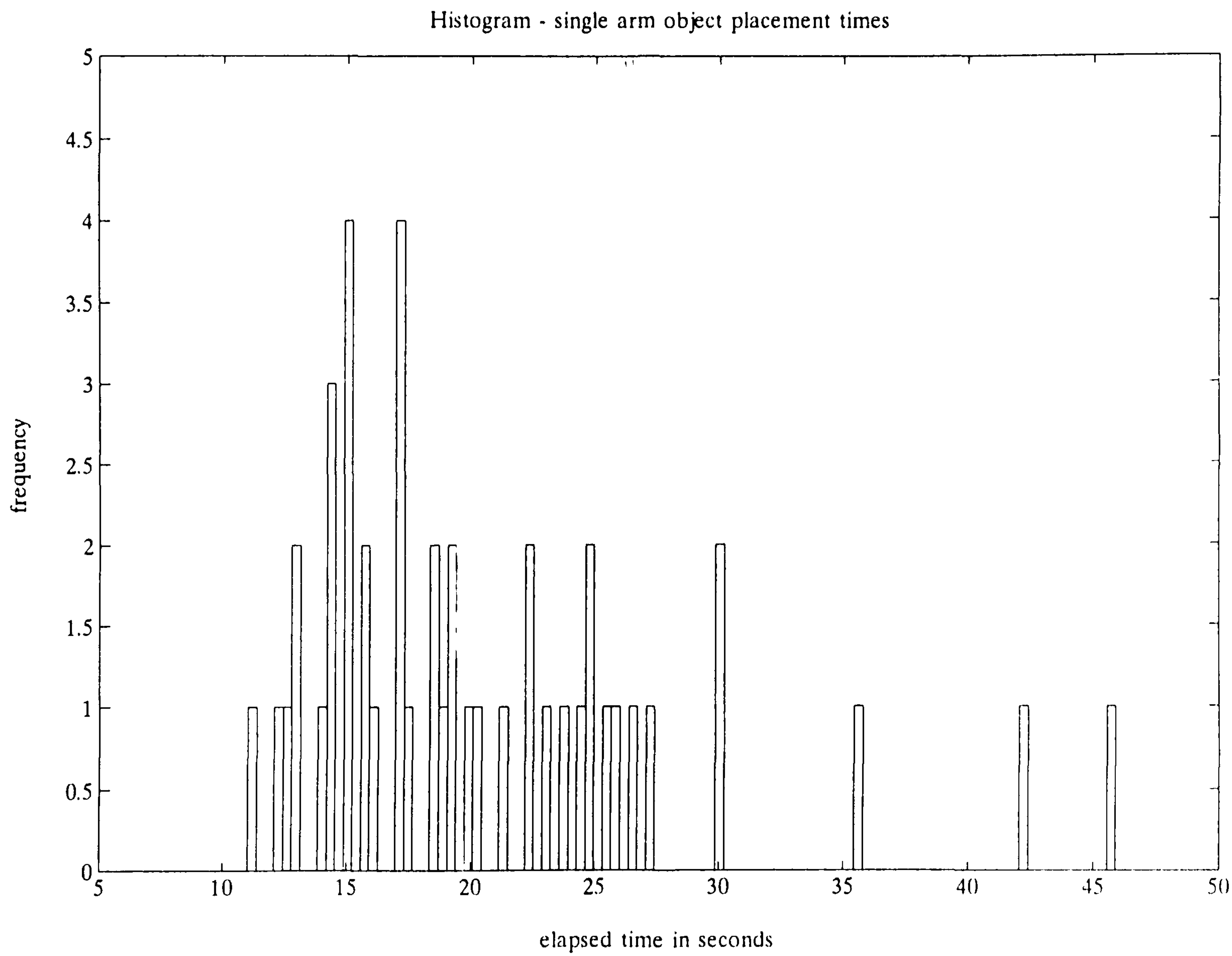

closed loop step response
magnitude
sample times where T = 1/36 seconds


open loop ramp response
magnitude
sample times where T = 1/36 seconds
0
5
10
15
20
25
30
35
40
45
50

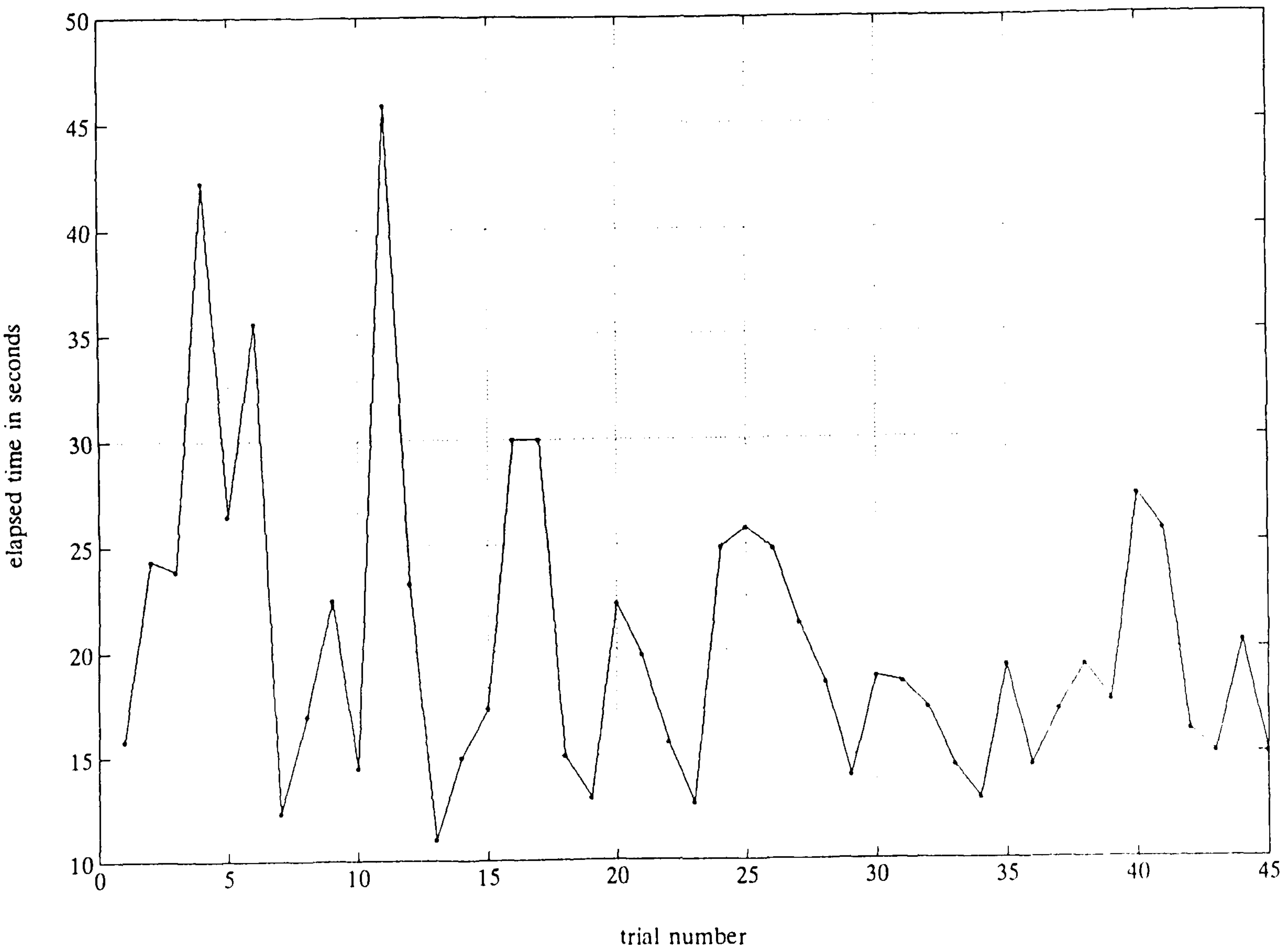

closed loop ramp response
magnitude
sample times where T = 1/36 seconds
0
5
10
15
20
25
30
35
40
45
50

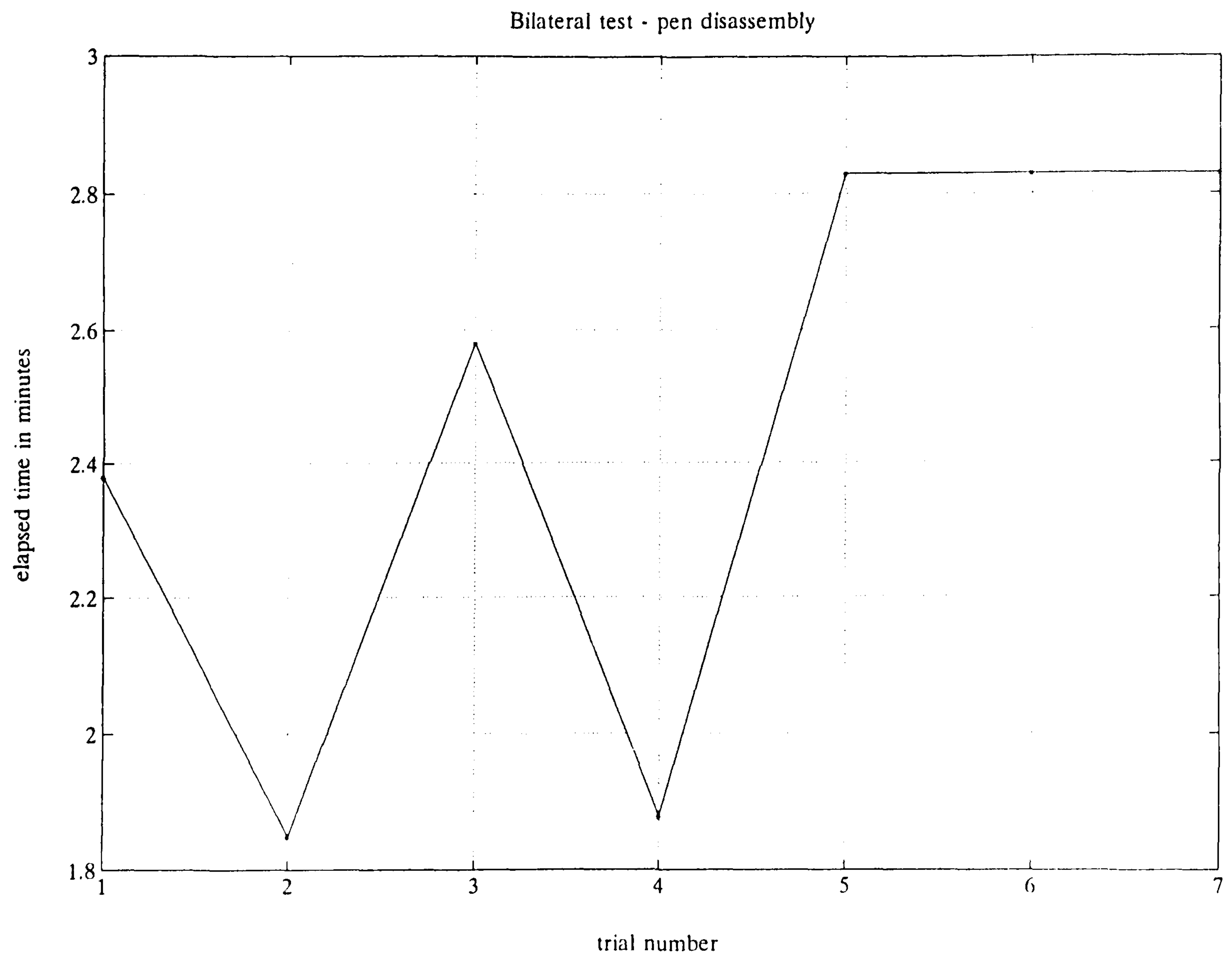

open loop Z-plane Root Locus - K = .001 to 20
imag axis
real axis
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1
0
1
2

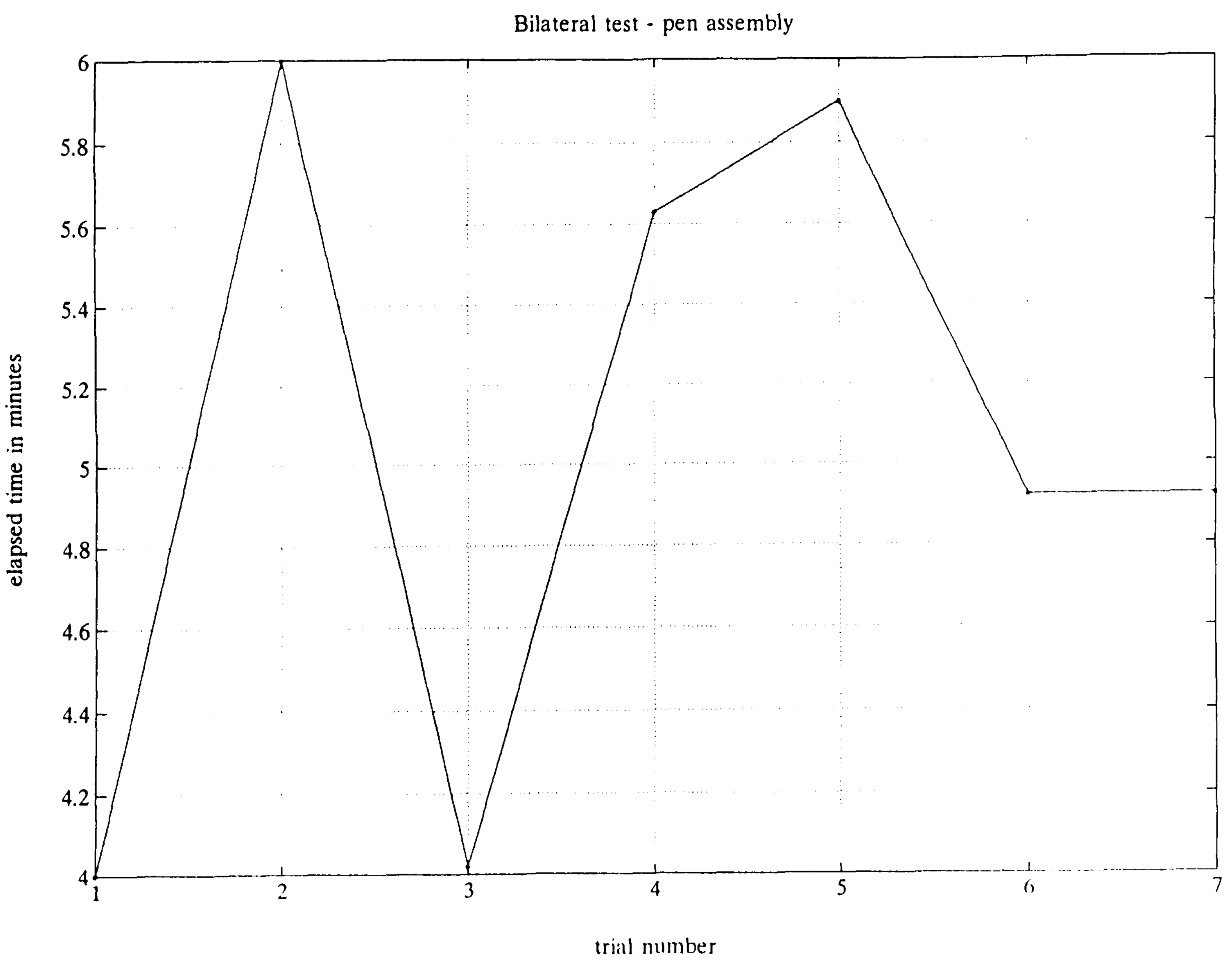

closed loop Z-plane Root Locus - K = .001 to 20
imag axis
real axis
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1
0
1
2


Histogram - single arm object placement times
frequency
elapsed time in seconds
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
10
15
20
25
30
35
40
45
50

Single arm test - placing object in cup




Bilateral test - pen disassembly
elapsed time in minutes
3
2.8
2.6
2.4
2.2
2
1.8
1
2
3
4
5
6
7
trial number


Bilateral test - pen assembly
elapsed time in minutes
trial number
6
5.8
5.6
5.4
5.2
5
4.8
4.6
4.4
4.2
4
1
2
3
4
5
6
7

Bilateral test - pen disassembly and assembly