SUPERVISORY CONTROL OF REMOTE MANIPULATION WITH COMPENSATION FOR MOVING TARGET

Kazuo Tani

MAN-MACHINE SYSTEMS LABORATORY
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

July 21, 1980

ATOMIC ENERGY BUREAU
Science and Technology Agency
Japan

Grant 04-7-158-44079 e/v1
M.I.T. SEA GRANT PROGRAM
Office of Sea Grant
National Oceanographic and
Atomospheric Administration
Department of Commerce

Contract N00014-77-C-0256
Work Unit Number NR196-152
OFFICE OF NAVAL RESEARCH
Engineering Psychology Programs

SUPERVISORY CONTROL OF REMOTE MANIPULATION WITH COMPENSATION FOR MOVING TARGET

by

Kazuo Tani

ABSTRACT

The aim of this project is to evaluate automatic compensation for moving targets in the supervisory control of remote manipulators.

An experimental system was built which consists of a master/slave manipulator, a moving table for the moving object, and a computer controlling both the manipulator and the table.

A software system was made which allows the master/slave operation with object motion compensation under computer control. The method of resolved motion rate control was adopted for the manipulator control. The computation time in this way proved practical and permitted a system sampling interval of 0.05 s.

Experiments were carried out with human operators performing manipulation tasks in the master/slave operation under computer control. Their performance was compared in three situations: no object motion, compensation for the object motion, and no compensation. The comparison of the compensation and no compensation situations showed that the compensation reduced the operation time by 26 - 41% in the peg moving task and increased the accuracy by two and a half times in rectangle tracing. In valve turning, however, a significant improvement was not observed.

Thus, it can be concluded that the compensation for target motion can improve the performance of the human operator significantly in certain kinds of tasks.

ACKNOWLEDGEMENTS

I would like to thank Prof. Thomas Sheridan, who kindly accepted my visit to the Man-Machine Systems Laboratory and offered me the field of research activity. I also wish to thank him for his guidance and advice; and, also, his help in revising my report draft.

I am also indebted to Dana Yoerger for familiarizing me with the M.I.T. way of life, Brad Deghuee for sharing the troubles of the manipulator experiment, Shunichi Odahara for the information given in Japanese, and all the members of the MMSL, including those mentioned above plus Thurston Brooks, for their cordial help and suggestions.

I appreciate the generocity of the Atomic Energy Bureau, Science and Technology Agency, Japan; the M.I.T. Sea Grant Program; and the Office of Naval Research, without which my work could never have been accomplished.

Finally I express my gratitude to my wife, Ryoko, for helping my stay in the United States.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION TO SUPERVISORY CONTROL OF REMOTE MANIPULATORS

The new environments man is working in with great hope are: Outer space, undersea, and nuclear radioactive. One problem encountered by people who are trying to perform activities in these areas is their physiological hositility, which keeps people distant from them. Either they must stay physically remote or remain isolated by special garments or capsules.

Manipulators have been developed to extend the man's capability while keeping him away from the dangerous working site. The work done with manipulators, such as repairing and maintenance of equipment, is not routine work, but varies from time to time according to the state of the equipment, and requires good judgment, dexterity, and great care.

The manipulators are usually controlled by a human operator in rate control or master/slave control. Operating manipulators in this way is a tiring job and the operator gets exhausted after a short time of work.

The use of the computer has been introduced to help the operator. Ferrell and Sheridan [1] in 1967 proposed the supervisory control of manipulation, where the computer can be to some extent autonomous, controlling the manipulator on its own, and the human operator, released from the direct control loop in which he had to take care of every tiny motion of the manipulator, can become a supervisor of the semi-autonomous manipulator or robotic device.

2. PROBLEMS IN THE MANIPULATION OF MOVING OBJECTS

Undersea tasks done by human divers are getting more and more costly and hazardous as they have to be done at increasing depth [2]. Using a manipulator mounted on an unmanned movable submersible and controlling it from the sea surface is the most desirable solution provided that the task can be performed reasonably well. But the use of the manipulator has its own problems. One of them is the relative motion between the manipulator and the object, which wasn't really a problem at all in the case of human divers [3].

Underseas a submersible with a manipulator may fix itself to the sea bottom, but the object the manipulator must handle may be moving about in the water, drifting due to the force of the current. Or, conversely, the object may be fixed and the submersible with the manipulator may be moving around; or, both can be moving.

In any of these cases there is relative motion between the manipulator and the object which the manipulator is to handle. The preprogrammed task motion may fail because the object simply isn't where it is supposed to be. If the manipulator is controlled in master/slave mode, the human operator must make the slave follow the object, i.e. he has to follow it himself on the master, and do his proper detail manipulation job in addition. As is easily imagined, this makes his work a lot harder.

If the relative motion of the object can be measured, object motion compensation can be accomplished. That is, in master/slave mode the object motion is added to the master motion to give a reference to the slave motion, and is subtracted from the slave motion to give the reference to the master motion. In a preprogrammed task mode the object motion is superimposed on the predetermined slave motion. So, in master/slave mode, despite relative motion the operator can perform the same task motion as if the object were staying still.

In this project, a method of object motion compensation is developed, and experiments are conducted about the performance of task operation in master/slave mode with moving objects, assuming that the controller can know the relative motion of the object.

3. HARDWARE FOR THE EXPERIMENT

It may help one to understand what has been done in this project if the devices used in the experiment are described briefly.

A master/slave manipulator which has been in use for a long time in the Man-Machine Systems Laboratory was used in this project (Fig. 3.1). A new PDP11/34 minicomputer was used to perform the control job. A movable table was brought back to life, offering three-dimensional (x,y,z) motion under the computer control to the object which is handled by the manipulator (Fig. 3.2. Also, see Appendix C).

The manipulator is an Argonne National Laboratory E2 master/slave manipulator. It is articulated type with six degrees of freedom (θ_A, θ_X, θ_Y, θ_L, θ_R, θ_A) excluding gripping action, and can be modeled as shown in Fig. 3.3 along with its dimensions [3]. The master and the slave have the same contruction, both being electromechanical except that the master has a grip which fits the human hand and the slave a gripper similar to a pair of tongs. The electric drive is by means of 10 Watt 115 Volt 60 Hz A.C. motors. The upper three joints (θ_Z,θ_X,θ_Y) are connected to their motors by means of gears. The lower three joints (θ_L, θ_R, θ_A) and the grip are connected to their motors by means of cables and pulleys.

A servo has been constructed to control each joint angle with its potentiometer output compared to the reference input value. In master/slave mode, in which the computer control can be irrelevant, the master servo and the slave servo are coupled to give the master/slave control loop shown in Fig. 3.4. This allows bilateral control with force feedback. To improve stability, two other loops are present: a tachometer feed-

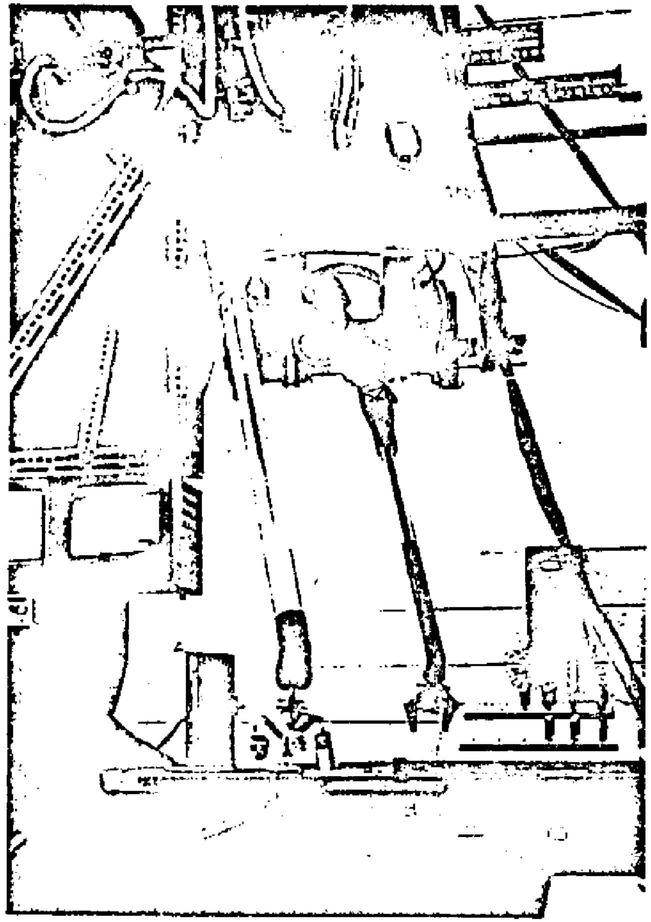


Fig. 3.1. The Manipulator

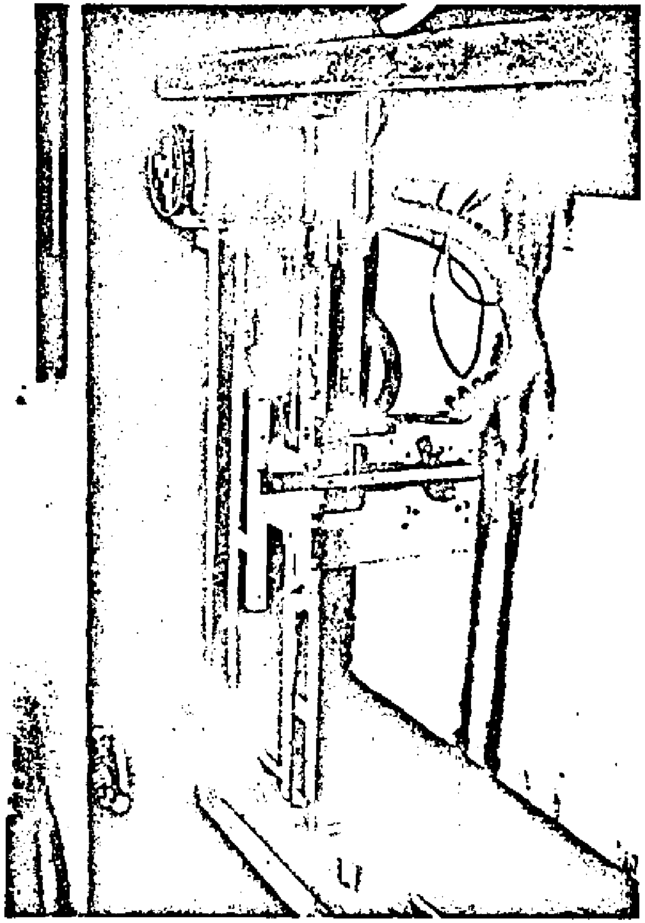
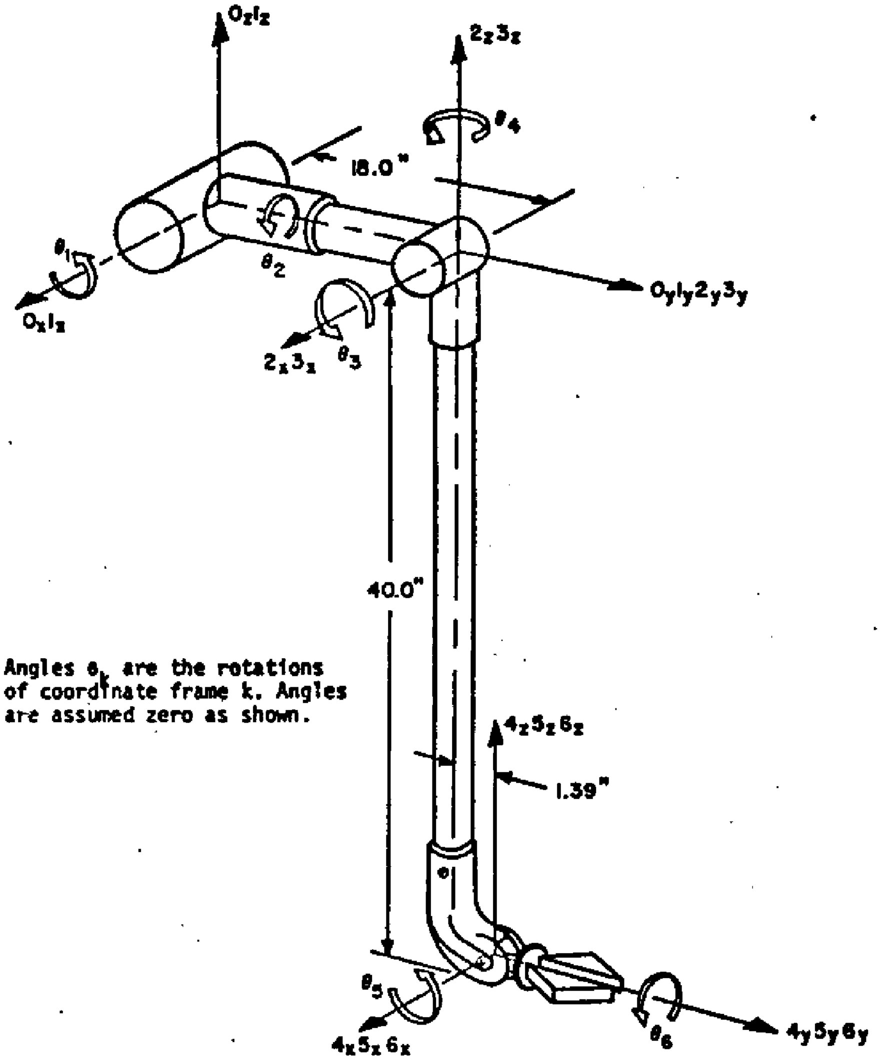


Fig. 3.2. The Movable Table



Angles θ_k are the rotations of coordinate frame k. Angles are assumed zero as shown.

Fig. 3.3. Definitions of Coordinate Systems and Rotation Angles of the Manipulator (from Brooks[3])

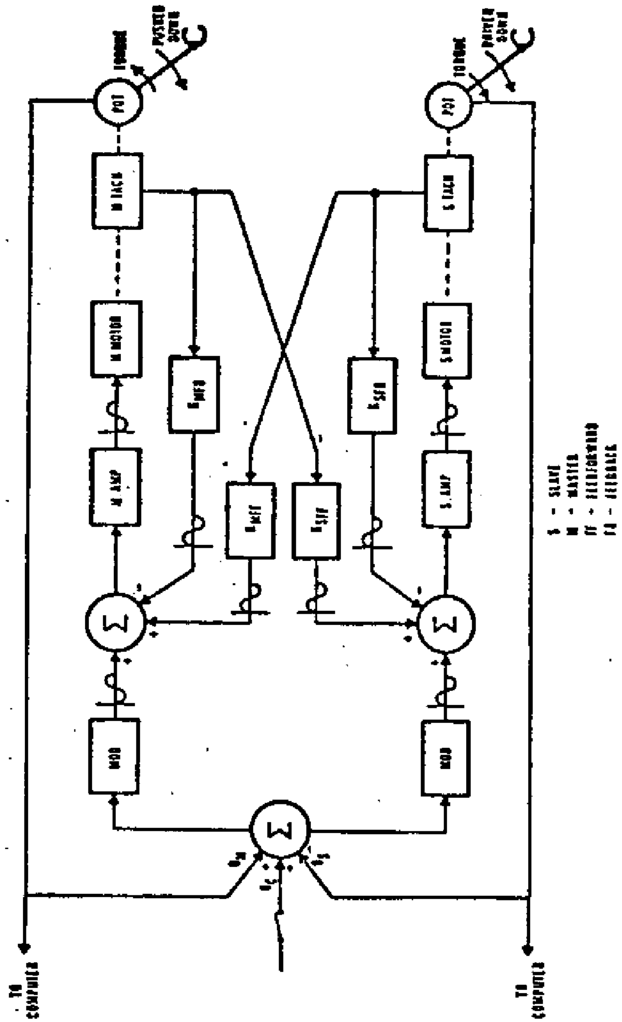


Fig. 3.4. Generalized Block Diagram of Servo Control Loop in Master/Slave Mode (Modified from Brooks[3])

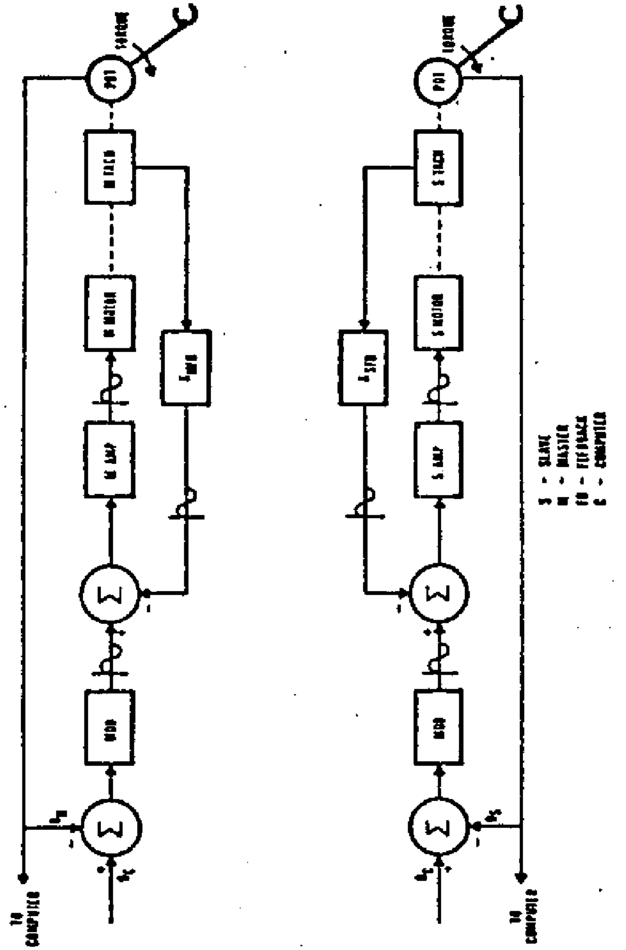


Fig. 3.5. Generalized Block Diagram of Servo Control Loop in Computer Control Mode (from Brooks[3])

forward and a tachometer feedback loop. In the computer control mode the servos for the master and the slave act independently in reference to their respective input values from the computer as shown in Fig. 3.5.

The manipulator was originally controlled by an Interdata Model 70 computer. Thus the need arose to change its computer interface completely to fit the new computer, a PDP11/34. Details of this interface are discussed in Appendix B.

A Data Aquisition and Distribution System ANALOGIC AN5400 is used to allow the computer to communicate with the experimental devices. The AN5400 is connected through a General-Purpose Interface Module DR11-C to the PDP11/34. It has an A/D convertor and a multiplexer providing 32 channels of analog input, 16 D/A converters, 32 bits of digital input, 32 bits of digital output, and a 7-digit octal dial on the panel which can be read by the computer.

The computer can take the following actions based on signals from or to the manipulator through the AN5400.

a) input the joint angles of the manipulator;

b) input the gripping force of the slave tongs;

c) switch between the two control modes: master/slave and computer control;

d) output the reference values to the manipulator servos in the computer control mode, or the angular differences between the master and the slave in master/slave mode.

The computer can also take care of the table action through the AN5400.

a) see if the carriage has hit any of the limit switches;

b) specify the speed and the direction of the carriage motion along the three axes (x,y,z).

The details of the AN5400 and the DR11-C are given in Appendix A.

4. CONTROL THEORY OF THE MANIPULATOR MOTION

The manipulator can be modeled as shown in Fig. 3.3. It has six degrees of freedom. Although the joint angles can represent the manipulator motion by themselves, cartesian coordinate systems give more convenience and flexibility to the control.

Frame 0 is defined at the manipulator base and fixed to the vehicle. Each joint of the arm is assigned a coordinate system, starting with frame 1 at the first joint out to the hand which is designated as frame 6.

The joint angles θ_k specify the rotation of the kth frame with respect to the previous frame (k-1).

According to the notation given by T. Brooks [3], the transformation from the hand frame to the vehicle frame is given as

$$^0A_6 = {}^0A_1 \cdot {}^1A_2 \cdot {}^2A_3 \cdot {}^3A_4 \cdot {}^4A_5 \cdot {}^5A_6.$$

Also the transformation from the fourth frame (wrist frame) to the vehicle frame is given as

$$^0A_4 = {}^0A_1 \cdot {}^1A_2 \cdot {}^2A_3 \cdot {}^3A_4.$$

Tables D.1 and D2 show the expression of $^0A_4$ and $^0A_6$.

Suppose one is given the following problem: given a position (x,y,z), find the joint angles θ_1 through θ_6 that place the manipulator at that position. A look at the transformation matrices $^0A_4$ and $^0A_6$ shows that θ_5 and θ_6 are irrelevant. Solving θ_1 through θ_4 given x, y, and z is a redundant system and yields an infinite number of solutions. Some constraint could be given to make the system non-redundant but the problem is that the system is analytically insolvable.

One of the methods to overcome this difficulty is Resolved Motion Rate Control (RMRC) proposed by D. Whitney [4].

$$\vec{x} = f(\vec{\theta}),$$

where $\vec{x}$ is a position vector at time t,

$$\vec{x} = (x_1, x_1, \ldots, x_n),$$

and $\vec{\theta}$ is a vector of joint angles at time t,

$$\vec{\theta} = (\theta_1, \theta_2, \ldots, \theta_m).$$

If we define a Jacobian matrix as

$$J(\vec{\theta}) = \begin{bmatrix} \dfrac{\partial x_1}{\partial \theta_1} & \cdots & \dfrac{\partial x_1}{\partial \theta_m} \\ \cdots & \cdots & \cdots \\ \dfrac{\partial x_n}{\partial \theta_1} & \cdots & \dfrac{\partial x_n}{\partial \theta_m} \end{bmatrix}$$

we get the following relation:

$$\dot{\vec{x}} = J(\vec{\theta})\ \dot{\vec{\theta}}.$$

Since this is a linear system, $\dot{\vec{\theta}}$ can be solved for given $\dot{\vec{x}}$ if m = n and $J(\vec{\theta})^{-1}$ exists.

$$\dot{\vec{\theta}} = J(\vec{\theta})^{-1}\dot{\vec{x}}$$

As for a short period of time, Δt,

$$\Delta\vec{\theta} = \dot{\vec{\theta}}\Delta t$$

$$\Delta\vec{x} = \dot{\vec{x}}\Delta t$$

$$\Delta\vec{\theta} = J(\vec{\theta})^{-1}\Delta\vec{x}$$

$$\begin{aligned} \vec{\theta}(t+\Delta t) &= \vec{\theta} + \Delta\vec{\theta} \\ &= \vec{\theta} + J(\vec{\theta})^{-1}\ \Delta\vec{x} \\ &= \vec{\theta}_r \end{aligned}$$

For the implementation of this method one variable is introduced in addition to x, y, and z. Suppose the hand direction vector ${}^6\vec{y}$ as referenced to frame 0 is (s,1,0). Since the vector ${}^4\vec{x}$ is perpendicular to ${}^6\vec{y}$ regardless of θ_5 and θ_6, the inner product of ${}^4\vec{x}$ and ${}^6\vec{y}$ should be zero.

$$({}^4a_{11}, {}^4a_{12}, {}^4a_{14})\cdot(s,1,0) = 0,$$

which can be rewirtten

$$s\cdot{}^4a_{11} + {}^4a_{12} = 0.$$

So a new variable

$$p = s\cdot{}^4a_{11} + {}^4a_{12}$$

is introduced, which is to be made zero in the control.

In this case n = 4 and

$$\vec{x} = (x,y,z,p),$$

$$\vec{\theta} = (\theta_1,\theta_2,\theta_3,\theta_4).$$

To keep vector ${}^6\vec{y}$ horizontal,

$${}^6a_{32} = 0,$$

which gives θ_{5r}.

The additional constraint of keeping the hand vector ${}^6\vec{x}$ horizontal gives

$${}^6a_{31} = 0,$$

which gives θ_{6r}.

Table D.3 shows the elements of the Jacobian matrix and the expression of the bases of the joint angles: θ_{5r} and θ_{6r}.

5. MASTER/SLAVE OPERATION WITH OBJECT MOTION COMPENSATION UNDER COMPUTER CONTROL

If the object to be manipulated is moving, it is desirable to compensate the object motion for easy operation by a human operator. By compensation is meant that the slave manipulator hand moves with the object motion and in addition makes the necessary motion to do the task on the object, while the master makes the latter motion only. In this compensated master/slave operation, the reference given to the slave will be the master position plus the object position, and the reference given to the master will be the slave position minus the object position.

Since the object position is given in the cartesian coordinate system fixed to the frame 0 of the manipulator and the drive is done on the joint angles, this control necessitates the transformation from the angle representation to the cartesian representation as well as the other way around.

The conceptual diagram of this control is shown in Fig. 5.1.

As discussed in the previous chapter, given the cartesian position of the manipulator (x,y,z) and the hand vectors ${}^6\vec{y}$ = (s,1,0) and ${}^6\vec{x}$ horizontal, the RMRC logic yields the six angles θ_{1r} through θ_{6r}. From this state any angular value θ_{6d} can be superimposed on the base: θ_{6r} without giving any effect back to θ_{5r} through θ_{1r}.

Thus the degrees of freedom of the compensated master/slave operation are five: x, y, z, s, and θ_{6d}, excluding the gripping motion, which is not included in the computer control but is retained in master/slave mode by itself. Thus, this control system has one less degree of freedom than the manipulator mechanism has. The recovery of this lost degree of freedom

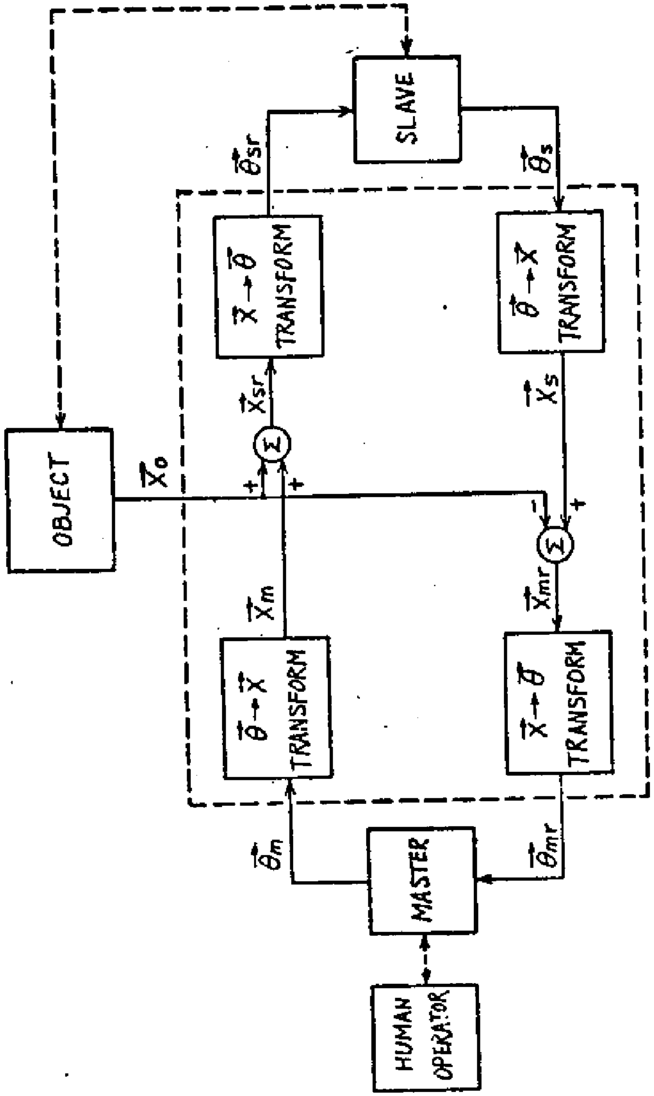


Fig. 5.1. Conceptual Block Diagram of Master/Slave Operation with Object Motion Compensation Under Computer Control

could be had at the cost of considerable amount of additional computation.

By the use of a digital computer, the master/slave control system becomes a discrete time sampling system. The sampling time interval was chosen to be one twentieth of a second (0.05 s). The interval of 0.1 s was tried but proved to produce motion which was too shaky. The system clock of the RSX-11M operating system which works on the 60 Hz line frequency is used.

6. PRACTICAL PROBLEMS IN IMPLEMENTATION

In the course of developing programs many problems have emerged. One is how to make the master operation by the human operator easy. Most of the remaining problems are concerned with stability. The measures taken are hardly academic.

6.1 Master/Slave Operation Under Computer Control

Simply using the master position as the reference to the slave and the slave position as the reference to the master has led to a swaying vibration of the manipulator. The following scheme is used to prevent this vibration.

$$\vec{x}_{sr} = a\cdot\vec{x}_m + (1-a)\cdot\vec{x}_s$$

$$\vec{x}_{mr} = a\cdot\vec{x}_s + (1-a)\cdot\vec{x}_m,$$

where $\vec{x}_s$ and $\vec{x}_m$ are the positions of the slave and the master at the sampling time t, and $\vec{x}_{sr}$ and $\vec{x}_{mr}$ are the reference values output from the computer to which the slave and the master are controlled. Experiments have shown that a value over 0.7 for a brings about the vibration.

The above scheme still has a problem in that the master is too heavy to be moved by a human operator. The use of tachometer feed-forward in the master/slave mode has suggested the following scheme [3].

$$\vec{x}_{sr} = a\cdot\vec{x}_m + (1-a)\cdot\vec{x}_s + b\cdot(\vec{x}_s - \vec{x}_{s2})$$

$$\vec{x}_{mr} = a\cdot\vec{x}_s + (1-a)\cdot\vec{x}_m + b\cdot(\vec{x}_m - \vec{x}_{m2}),$$

where $\vec{x}_{s2}$ and $\vec{x}_{m2}$ are the positions of the slave and the master taken two cycles earlier. These values, $\vec{x}_{s2}$ and $\vec{x}_{m2}$, are used because the use of $\vec{x}_{s1}$ and $\vec{x}_{m1}$, which are the positions taken one cycle earlier, has caused vibration. A value over 0.5 for b causes the self-initiated motion of the manipulator.

6.2. Compensation of the Object Motion

The compensation of the object motion has introduced new problems concerned with the dynamics of the manipulator. If the slave is following the object motion and the master is compensated, the value given from the slave to the master/slave scheme should be $(\vec{x}_s - \vec{x}_t)$ and $(\vec{x}_{s2} - \vec{x}_{t2})$, and the reference to the slave should be increased by $\vec{x}_{ta}$, where $\vec{x}_t$ is the table position at the sampling time t and $\vec{x}_{t2}$ is that two cycles earlier, and $\vec{x}_{ta}$ is the position where the table is expected to be a certain time in the future.

This lead time is understood to result from the delay time of the manipulator drive and the discrete updating of the reference signals by the computer. The proper lead time is chosen by experiment. Figure 6.1 shows how this lead time takes care of both the delay time of the manipulator and the effect of the reference signals which do not change continuously but change at the sampling time and stay constant during the cycle.

The lead time derived by experiment is:

x: 0.10 s

y: 0.10 s

z: 0.15 s.

Too small a lead time makes the master move in the opposite direction to the object motion, while too big a lead time makes the master move in the same direction. Too small a lead time also makes the manipulator follow behind the object, while too big a lead time makes the manipulator go ahead of the object.

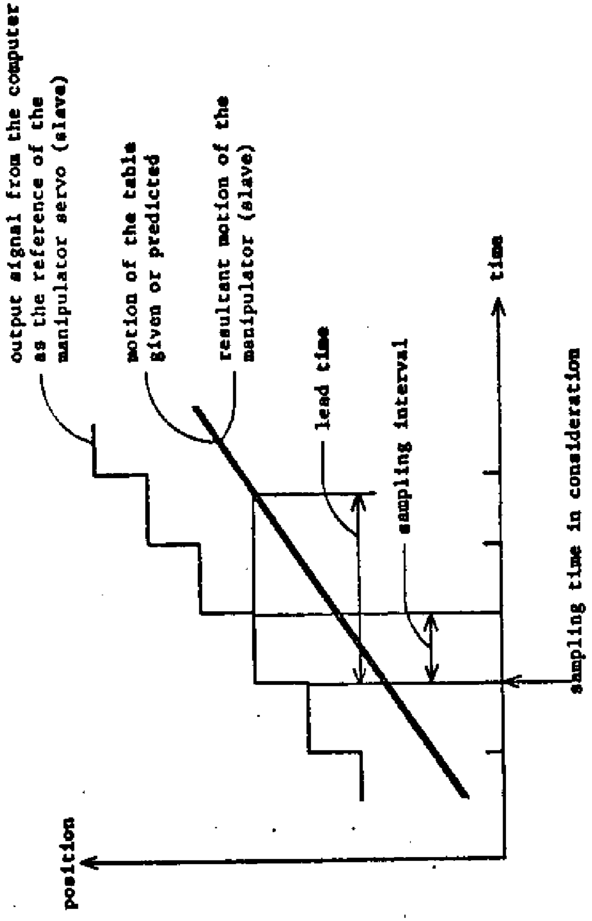


Fig. 6.1. Lead Time in the Master/Slave Operation with Object Motion Compensation

In a practical application where the object motion is measured, the prediction of the object position is necessary [5]. In this project, where the object is moved in a predetermined way, it is easy to achieve, but it may be difficult in practice.

The eventual scheme is as follows:

$$\vec{x}_{sr} = a\cdot\vec{x}_m + (1-a)(\vec{x}_s-\vec{x}_t) + b\cdot[(\vec{x}_s-\vec{x}_t)-(\vec{x}_{s2}-\vec{x}_{t2})] + \vec{x}_{ta}$$

$$\vec{x}_{mr} = a\cdot(\vec{x}_s-\vec{x}_t) + (1-a)\cdot\vec{x}_m + b\cdot(\vec{x}_m-\vec{x}_{m2})$$

6.3 Resolved Motion Rate Control

For the resolved motion rate control program the following method was used.

Given θ_1 through θ_4, x, y, and z are calculated. Given the destination position (x_d, y_d, z_d, s_d), the variable

$$p = s_d\cdot {}^4a_{11} + {}^4a_{12}$$

is evaluated. The partial derivatives are evaluated forming the Jacobian matrix

$$J(\vec{\theta}) = \begin{bmatrix} \frac{\partial x}{\partial\theta_1} & \frac{\partial x}{\partial\theta_2} & \frac{\partial x}{\partial\theta_3} & \frac{\partial x}{\partial\theta_4} \\ \frac{\partial y}{\partial\theta_1} & \frac{\partial y}{\partial\theta_2} & \frac{\partial y}{\partial\theta_3} & \frac{\partial y}{\partial\theta_4} \\ \frac{\partial z}{\partial\theta_1} & \frac{\partial z}{\partial\theta_2} & \frac{\partial z}{\partial\theta_3} & \frac{\partial z}{\partial\theta_4} \\ \frac{\partial p}{\partial\theta_1} & \frac{\partial p}{\partial\theta_2} & \frac{\partial p}{\partial\theta_3} & \frac{\partial p}{\partial\theta_4} \end{bmatrix}$$

The position increments are calculated as follows:

$$\Delta x = x_d - x$$

$$\Delta y = y_d - y$$

$$\Delta z = z_d - z$$

$$\Delta p = -p$$

Now the equation to be solved is

$$\Delta\vec{x} = J(\vec{\theta})\ \Delta\vec{\theta}.$$

The matrix is 4x4, not too large, and one element $\partial x/\partial\theta_1$ is zero. Taking advantage of this fact, an economized version of the method developed by T. Banachiewicz [6] is used instead of a general method.

The variables x, y, z, p and their partial derivatives are made of terms which consist of the combinational multiplication of the trigonometric functions of θ_1 through θ_4. At first glance their computation looks time consuming but further inspection reveals a lot of duplications of the same terms, which enables simplication of the computation. Comparison of Tables D.3 and D.4 shows how much calculation was saved by the simplification. One cycle of this program takes 8.8 ms on the PDP11/34. The control needs two cycles, one for the master and one for the slave. This computation time is quite acceptable and compared with the system sampling interval 0.05 s.

6.4 Model Angles and Mechanical Angles of the Manipulator

Due to the mechanism of the manipulator, the mechanical joint angles θ_X, θ_Y, θ_Z, θ_L, θ_R, and θ_A are not quite the same as the angles shown in Fig. 3.3. Three of the angles are the same.

$$\theta_1 = \theta_Z$$

$$\theta_2 = \theta_X$$

$$\theta_4 = \theta_A$$

With a link connection,

$$\theta_3 = \theta_Y - \theta_Z.$$

With the differential gear mechanism using 33/40 gear ratio,

$$\theta_6 = (\theta_L - \theta_R)/1.65.$$

With the interference of θ_3 over θ_5 in addition,

$$\theta_5 = (\theta_L + \theta_R)/2 + 0.27\ \theta_3.$$

7. EXPERIMENTS ON THE PERFORMANCE OF HUMAN OPERATORS

Experiments were carried out to evaluate how compensation for the moving object helps the human operator. Operator performances are compared for the three situations: no relative motion between the object and the manipulator base, relative motion between them but with compensation, and relative motion but no compensation.

The motion of the object, or the motion of the table, which actually was the relative motion between the object and the manipulator base, was arbitrarily chosen as follows:

$$x = 1.4\ \sin(6.2832/5.0)t - 5.6\ \sin(6.2832/14.5)t$$

$$y = 1.4\ \sin(6.2832/6.5)t - 5.6\ \sin(6.2832/18.5)t$$

$$z = 0.4\ \cos(6.2832/5.0)t - 0.4\ \cos(6.2832/6.5)t,$$

where x, y, and z are expressed in inches and t in seconds.

The master/slave operation under computer control was chosen for all of the three situations to provide equal degrees of freedom of the manipulator motion and equal ease of master handling, although master/slave mode without computer could have been used for the no object motion and no compensation situations.

Three kinds of experiments were conducted: peg movement, valve turning, and rectangle tracing.

In the peg movement experiment (Fig. 7.1) a peg was used which consisted of a 2.6" x 2.4" x 0.8" board to be gripped by the slave tongs and a 0.39" dia. x 1.25" pin sticking out of the board. A board was mounted on the table carriage which had eight holes of 0.44" diameter arranged in 2 x 4 array with 3" intervals. Seven of the holes had 0.1" champher and were used for the experiment. The subject operator was told to insert the peg pin into one hole and move it into the next and

Fig. 7.1. Peg Moving Task

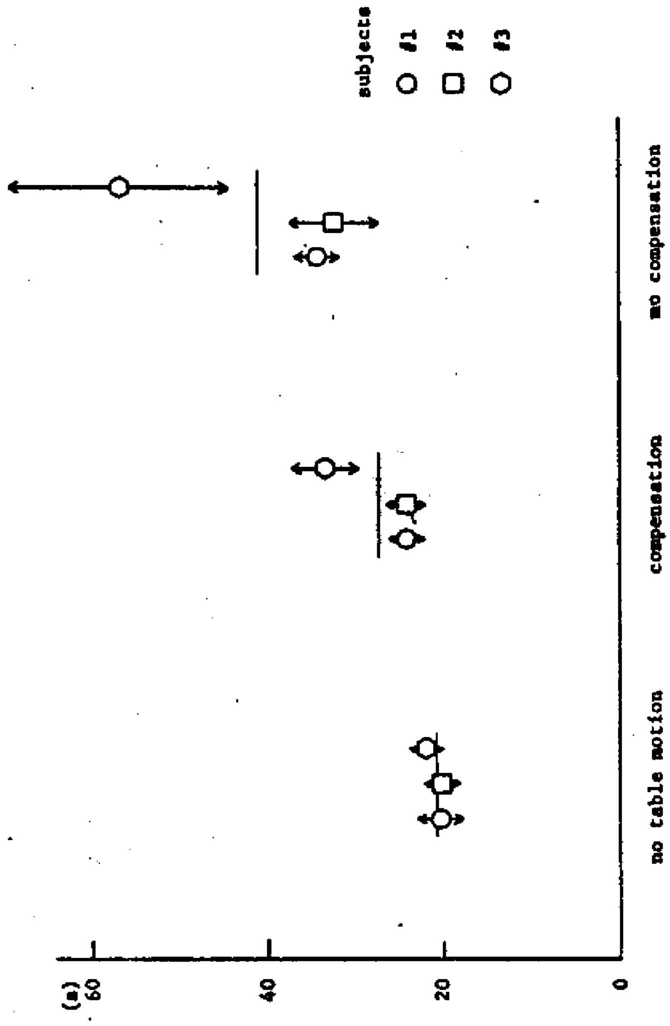


Fig. 7.2. Operation Time for the Peg Moving Task

so on around the board. The time was measured from the insertion into the first hole to the insertion into the same hole after one round of travel. Each subject made five trials for each situation.

Fig. 7.2 shows the results of this experiment. The mark shows the average time and the arrows show the standard deviation for each subject. The figure shows that with compensation the operation time is 18 to 51% (29% average) greater than with no object motion, but 26 to 41% (32% average) less than without compensation.

In the second experiment (Fig. 7.3) a valve was mounted on the table carriage. Time during which the subject rotated the valve two turns was measured. Fig. 7.4 shows the results of this experiment. This experiment did not show clear differences between situations.

The reasons for this seem to be as follows: In the peg movement task the manipulator positioning is initiated by the operator. But in the valve turning task most of the manipulator positioning is initiated by the slave which grips the moving valve firmly. The peg movement task is rather easy with no object motion at all and gets a little awkward with motion and compensation and more awkward without compensation. On the other hand, the valve turning task is as awkward in one situation as in any other.

The valve turning in this way may be an inhuman task. Positioning of the manipulator tongs by the human operator and automatic turning of the valve by the computer, which was experienced in T. Brooks' thesis [3], might show different results.

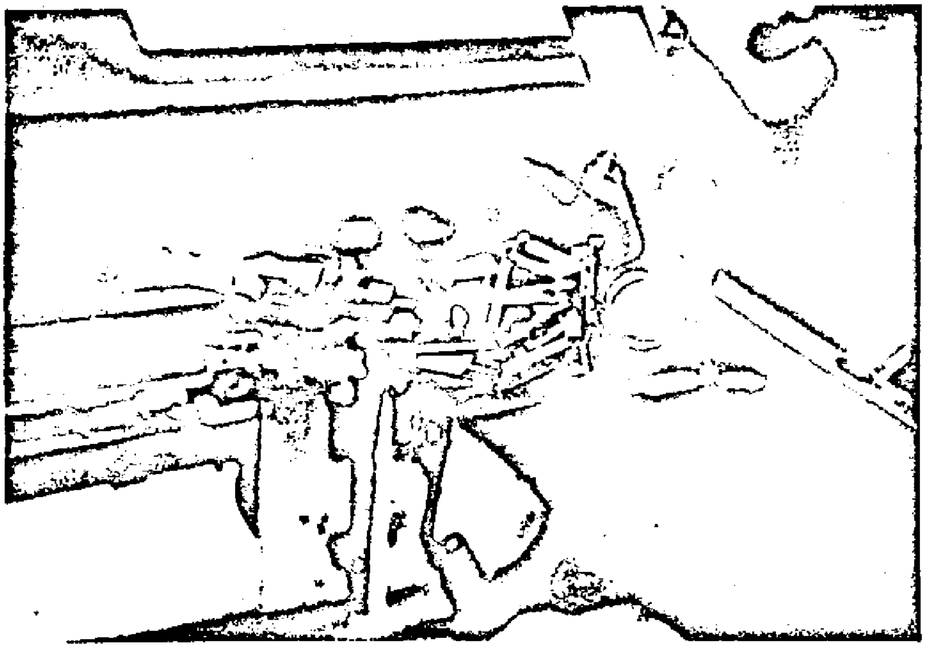


Fig. 7.3. Valve Turning Task

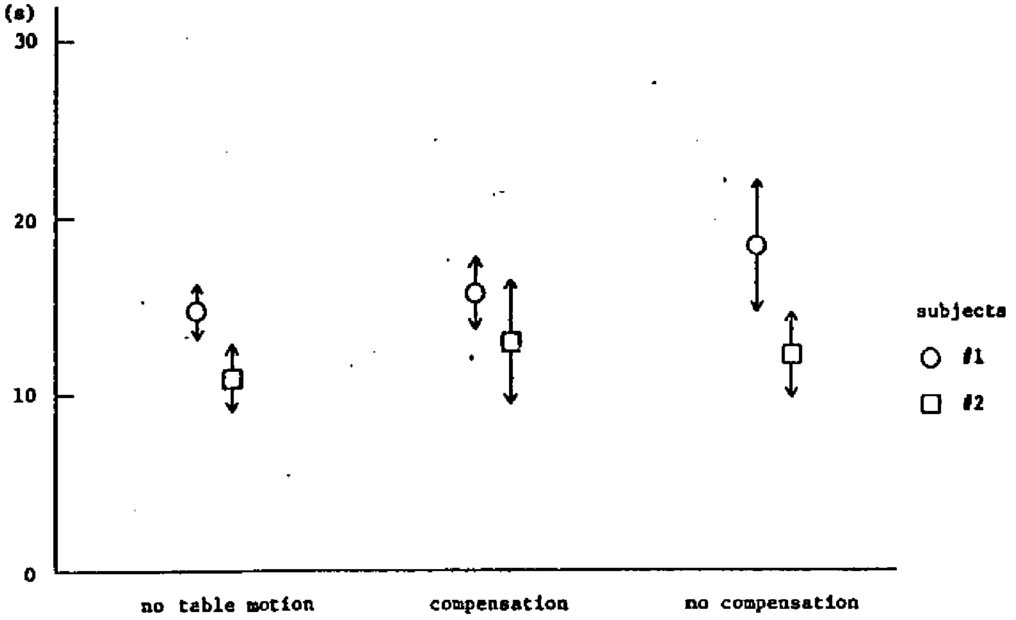


Fig. 7.4. Operation Time for the Valve Turning Task

In the third experiment a piece of paper which has concentric 4" x 6" and 3" x 5" rectangles drawn on it is placed on the carriage. The rectangle in the middle of the two is considered as the reference rectangle. The subject was requested to trace the reference rectangle with a magic marker gripped by the slave. The eight greatest errors of the trace from the reference rectangle were measured and their mean value was computed. For the selection of the eight errors the following rule was adopted:

> Take the maximum error. Discard all the errors in the same direction (inside or outside) within the neighborhood of one inch (measured on the reference rectangle) from the previous error point. Take the maximum error from the rest. Repeat this operation until eight values have been chosen.

Figure 7.6 shows the examples of the traces drawn with the manipulator. Figure 7.7 shows the comparative errors for the three situations. With compensation the average error is about twice as big as with no object motion but two and a half times as small as without compensation.

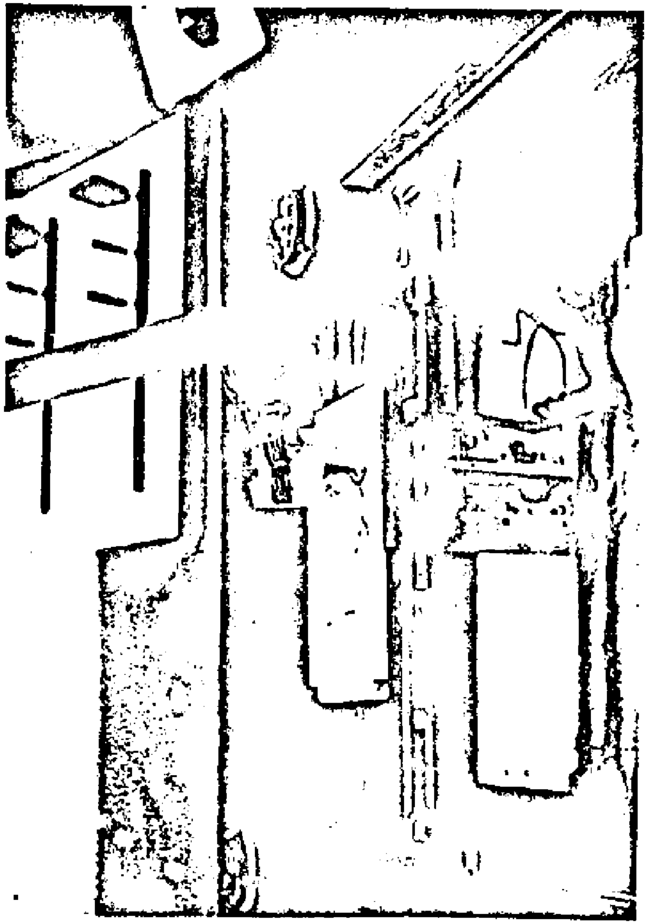
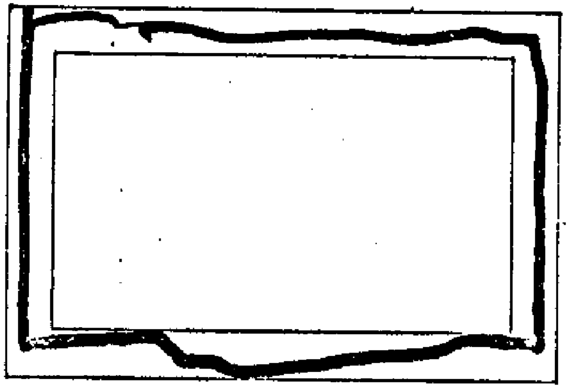
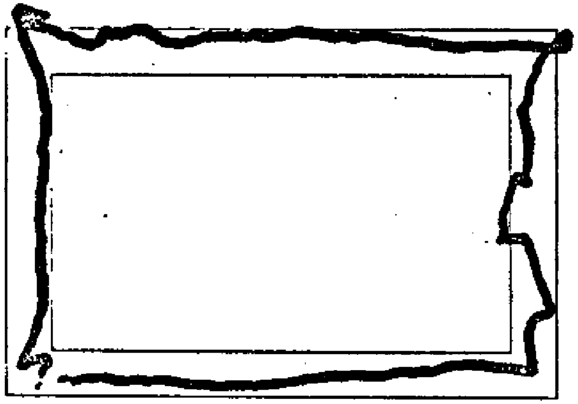


Fig. 7.5. Rectangle Tracing

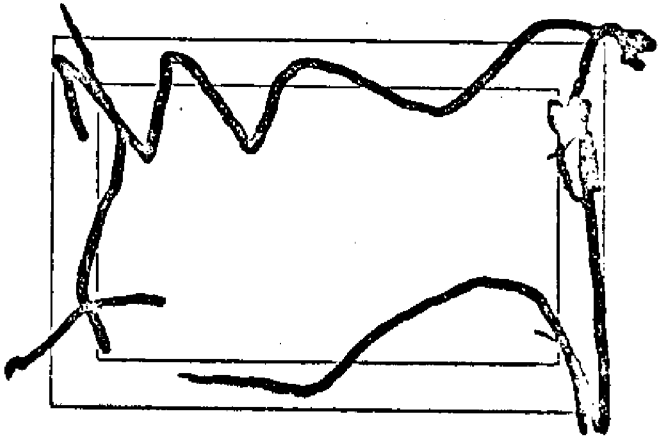


(a) no table motion (e = 0.156)

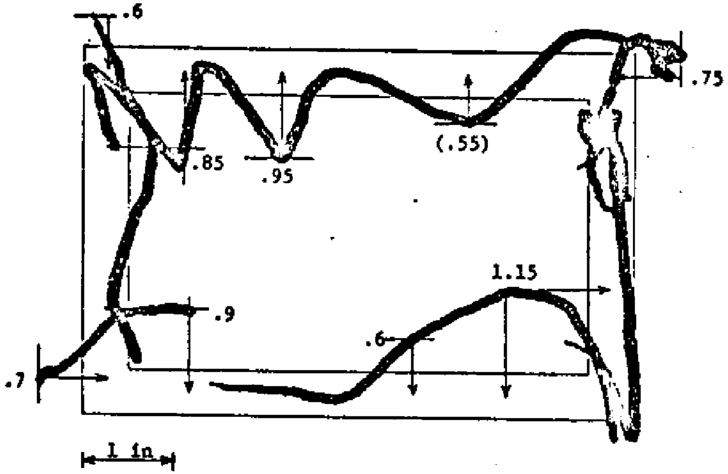


(b) compensation (e = 0.275)

Fig. 7.6. Examples of Rectangle Traces



(c) no compensation (e = 0.812)



(d) selection of errors

Fig. 7.6. (Cont.)

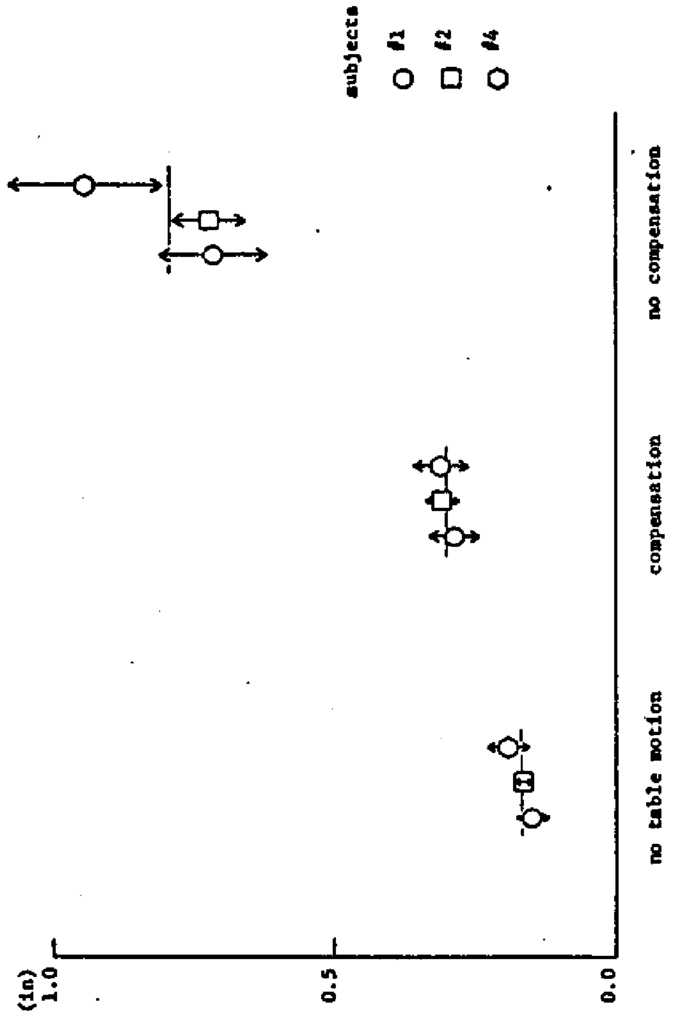


Fig. 7.7. Comparative Errors in Rectangle Tracing

8. CONCLUSIONS

The aim of this project is to evaluate automatic compensation for moving targets in the supervisory control of remote manipulators.

An experimental system was built which consists of a master/slave manipulator, a moving table for the moving object, and a computer controlling both the manipulator and the table.

A software system was made which allows the master/slave operation with object motion compensation under computer control. The method of resolved motion rate control was adopted for the manipulator control. The computation time in this way proved practical and permitted a system sampling interval of 0.05 s.

Experiments were carried out with human operators performing manipulation tasks in the master/slave operation under computer control. Their performance was compared in three situations: no object motion, compensation for the object motion, and no compensation. The comparison of the compensation and no compensation situations showed that the compensation reduced the operation time by 26 - 41% in the peg moving task and increased the accuracy by two and a half times in rectangle tracing. In valve turning, however, a significant improvement was not observed.

Thus, it can be concluded that the compensation for target motion can improve the performance of the human operator significantly in certain kinds of tasks.

APPENDIX A. ANALOGIC AN5400 AND DR11-C

The AN5400 is connected to the PDP11/34 through a DR11-C parallel interface. Jumpers are made giving the DR11-C device address: 767770, interrupt vector addresses: 300/304, and interrupt level: 5. Tables A.1 and A.2 show the cable connection between the DR11-C and the AN5400.

The connections between the AN5400 and the experimental devices are done through 20-pin flat cable cinch connectors. Tables A.3 and A.4 show the pin assignment of the connectors.

An assembly language program AN5402 is made to enable the FORTRAN programs to transfer data through the AN5400 (see Appendix E). It consists of seven subroutines: ANINIT, AIN, AINSQ, AOUT, AOUTSQ, DIN, and DOUT. For an example of the time required for the data transfer, the subroutines AINSQ requires approximately (37n + 63) microseconds to get analog data from n successive channels.

Table A.1.
Cable Connection Between DR11-C and AN5400 (1)

DR11-C P1 PIN		DR11-C P1 NAME	AN5400 GPI PIN	AN5400 GPI NAME	DR11-C P2 PIN		DR11-C P2 NAME	AN5400 GPI PIN	AN5400 GPI NAME
1	A				40	VV			
2	B				39	UU	GND	20	DIG GND
3	C	OUT00	35	BOD16	38	TT	IN00	43	BID16
4	D				37	SS	GND		
5	E				36	RR	INIT H		
6	F				35	PP	GND	(5)	
7	H				34	NN	INIT H	(9)	
8	J	GND	1	DIG GND	33	MM	GND	19	DIG GND
9	K	OUT01	36	BOD15	32	LL	IN01	44	BID15
10	L	OUT04	39	BOD12	31	KK	IN04	47	BID12
11	M	GND			30	JJ	GND		
12	N	OUT05	40	BOD11	29	HH	IN05	48	BID11
13	P	INIT H			28	FF			
14	R	OUT06	41	BOD10	27	EE	IN06	49	BID10
15	S	GND			26	DD	GND		
16	T	OUT07	42	BOD9	25	CC	IN07	50	BID9
17	U	OUT03	38	BOD13	24	BB	IN03	46	BID13
18	V	GND			23	AA	GND		
19	W	OUT08	5	BOD8	22	Z	IN08	13	BID8
20	X	OUT09	6	BOD7	21	Y	IN09	14	BID7
21	Y	GND			20	X	GND		
22	Z	OUT10	7	BOD6	19	W	IN10	15	BID6
23	AA	OUT11	8	BOD5	18	V	IN11	16	BID5
24	BB	OUT12	9	BOD4	17	U	IN12	17	BID4
25	CC	GND			16	T	GND	(3)	
26	DD	CSR1	(8)		15	S	REQ B	(7)	
27	EE	GND	(4)		14	R	GND		
28	FF	OUT13	10	BOD3	13	P	IN13	33	BID3
29	HH	OUT14	11	BOD2	12	N	IN14	32	BID2
30	JJ	OUT15	12	BOD1	11	M	IN15	31	BID1
31	KK	GND			10	L	GND	(2)	
32	LL	REQ A	4	DONE	9	K	CSR0	(6)	
33	MM	GND			8	J	GND	18	DIG GND
34	NN	OUT02	37	BOD14	7	H	IN02	45	BID14
35	PP	GND	2	DIG GND	6	F			
36	RR	OUT02			5	E	IN02		
37	SS	GND			4	D			
38	TT				3	C	D.TRANSM'D		
39	UU	GND	3	DIG GND	2	B			
40	VV	N.D.READY	34	DATA READY	1	A			

Table A.2.
Cable Connection Between DR11-C and AN5400 (2)

AN5400 GPI PIN	AN5400 GPI NAME	DR11-C PIN	DR11-C NAME
1	DIG GND	P1- J 8	
2	DIG GND	P1-PP 35	
3	DIG GND	P1-UU 39	
4	DONE	P1-LL 32	REQ A
5	BOD8	P1- W 19	OUT08
6	BOD7	P1- X 20	OUT09
7	BOD6	P1- Z 22	OUT10
8	BOD5	P1-AA 23	OUT11
9	BOD4	P1-BB 24	OUT12
10	BOD3	P1-FF 28	OUT13
11	BOD2	P1-HH 29	OUT14
12	BOD1	P1-JJ 30	OUT15
13	BID8	P2- Z 22	IN08
14	BID7	P2- Y 21	IN09
15	BID6	P2- W 19	IN10
16	BID5	P2- V 18	IN11
17	BID4	P2- U 17	IN12
18	DIG GND	P2- J 8	
19	DIG GND	P2-MM 33	
20	DIG GND	P2-UU 39	
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31	BID1	P2- M 11	IN15
32	BID2	P2- N 12	IN14
33	BID3	P2- P 13	IN13
34	DATA READY	P1-VV 40	N.D.READY
35	BOD16	P1- C 3	OUT00
36	BOD15	P1- K 9	OUT01
37	BOD14	P1-NN 34	OUT02
38	BOD13	P1- U 17	OUT03
39	BOD12	P1- L 10	OUT04
40	BOD11	P1- N 12	OUT05
41	BOD10	P1- R 14	OUT06
42	BOD9	P1- T 16	OUT07
43	BID16	P2-TT 38	IN00
44	BID15	P2-LL 32	IN01
45	BID14	P2- H 7	IN02
46	BID13	P2-BB 24	IN03
47	BID12	P2-KK 31	IN04
48	BID11	P2-HH 29	IN05
49	BID10	P2-EE 27	IN06
50	BID9	P2-CC 25	IN07

9-PIN CONN.	DR11-C PIN	DR11-C NAME
1		
2	P2- L 10	GND
3	P2- T 16	GND
4	P2-EE 27	GND
5	P2-PP 35	GND
6	P2- K 9	CSR0
7	P2- S 15	REQ B
8	P1-DD 26	CSR1
9	P2-NN 34	INIT H

Table A.3.
User Connectors Pin Assignment for DR11-C and AN5400

1. DR11-C

unused	1	6	CRS0 (output from PDP11/34)
GND	2	7	REQ B (input to PDP11/34)
GND	3	8	CSR1 (output from PDP11/34)
GND	4	9	INIT H (output from PDP11/34)
GND	5		

Device Addresses: 767770(STATUS), 767772(OUTPUT), 767774(INPUT)
Interrupt Vector Addresses: 300(REQ A), 304(REQ B)
Interrupt Level: 5

For details see DR11-C manual..

2. AN5400

2.1. DIGITAL INPUT 32 bits.
(2 connectors, 16 bits for each, addresses: 20 and 22 decimal)

2.2. DIGITAL OUTPUT 32 bits.
(2 connectors, 16 bits for each, addresses: 24 and 26 decimal)

2.3. ANALOG INPUT 32 channels.
(2 connectors, 16 channels for each, addresses: 0 - 15 and 16 - 31 decimal)
The A/D works in Single-Ended operation. But the input signal is measured with respect to Signal Return, not Analog Ground. This feature is called 'Pseudo-Differential" operation.

2.4. ANALOG OUTPUT 16 channels.
(2 connectors, 8 channels for each, addresses: 4 - 11 and 12 - 19 decimal)
Return Sense leads are omitted.
Returns are connected to Analog Ground.

Table A.4. User Connectors Pin Assignment for AN5400

Connectors are 943-1205 FCC-110-20.

DIGITAL INPUT

	unused	1	2	DIG GND	
	(MSB) 1	3	4	2	
	3	5	6	4	
	5	7	8	6	
Input bits	7	9	10	8	Input bits
	9	11	12	10	
	11	13	14	12	
	13	15	16	14	
	15	17	18	16 (LSB)	
	DIG GND	19	20	unused	

DIGITAL OUTPUT

	unused	1	2	DIG GND	
	(MSB) 1	3	4	2	
	3	5	6	4	
	5	7	8	6	
Output bits	7	9	10	8	Output bits
	9	11	12	10	
	11	13	14	12	
	13	15	16	14	
	15	17	18	16 (LSB)	
	DIG GND	19	20	unused	

ANALOG INPUT

	SIGNAL RTRN	1	2	ANALOG GND	
	0	3	4	1	
	2	5	6	3	
	4	7	8	5	
Input channels	6	9	10	7	Input channels
	8	11	12	9	
	10	13	14	11	
	12	15	16	13	
	14	17	18	15	
	ANALOG GND	19	20	unused	

ANALOG OUTPUT

LOAD0	1	2	DIG GND
OUT0	3	4	RETURN0
OUT1	5	6	RETURN1
OUT2	7	8	RETURN2
OUT3	9	10	RETURN3
OUT4	11	12	RETURN4
OUT5	13	14	RETURN5
OUT6	15	16	RETURN6
OUT7	17	18	RETURN7
LOAD4	19	20	DIG GND

For details see AN5400 Manuals.

APPENDIX B. MANIPULATOR INTERFACE

Figures B.1 through B.6 show the construction of the manipulator servos.

Figure B.7 shows the circuit for the control mode switching. Two manual switches on the panel are used to enable and disable the computer to change the modes. When the enable switch is pushed, the enable state is held by the self-holding relay circuit until the disable switch is pushed or the power is shut off. An LED light indicates the enable state.

In the enable state the computer can select the mode through a digital output, while in the disable state the manipulator is kept in the master/slave mode regardless of the signals from the computer.

Seven bits of the word of digital output are used for mode switching. Each of them can change the mode of its corresponding joint angle of the manipulator. One bit of the word is used to inhibit computer control. In normal operation this bit is set to zero. The use of this bit invalidates the spurious signal of all ones that comes when the cable is disconnected. The one kiloohm resistors placed in parallel to the relay contacts protect the contact points which switch inductive loads.

Figure B.8 shows the breadboard wiring of the circuit.

A modification is made on the comparator/modulator chassis (Fig. B.2). A switch is added to each chassis which is meaningful only in the master/slave mode. When on it allows the D/A signal from the computer to operate as the angular difference between master and slave in master/slave mode, and when off it makes the D/A signal irrelevant.

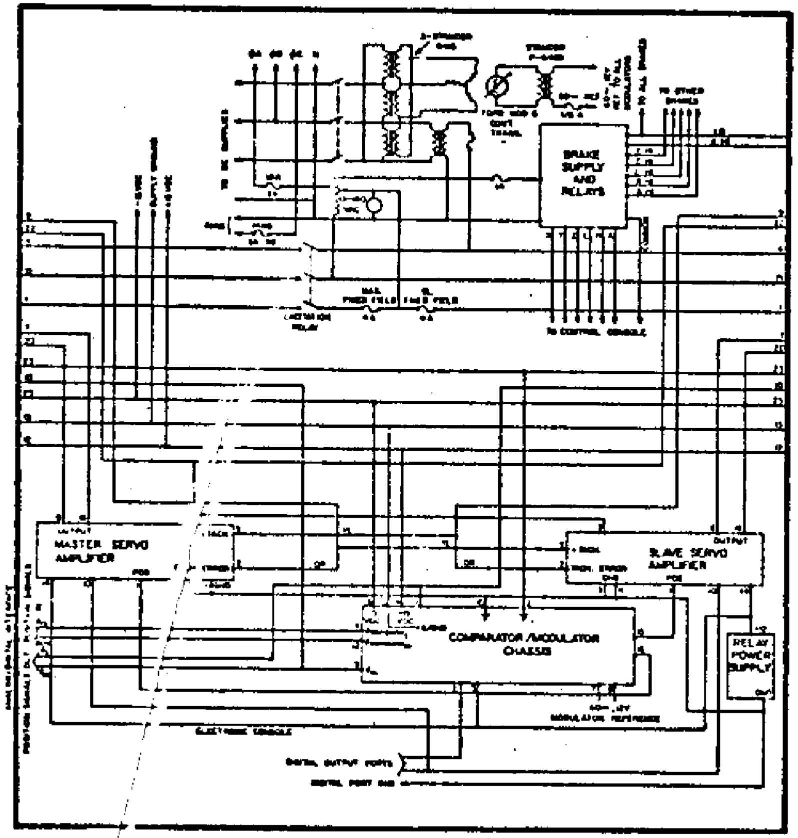


Fig. 3.1. Servo Control Rack Schematic (from Brooks[3])

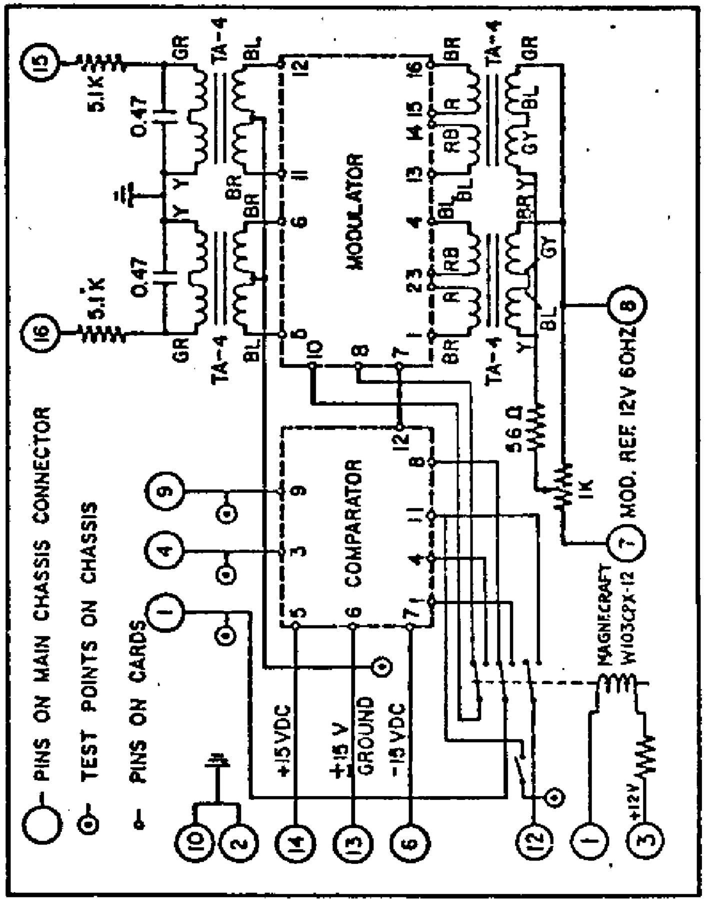


Fig. B.2. Comparator/Modulator Chassis Schematic (Modified from Brooks[3])

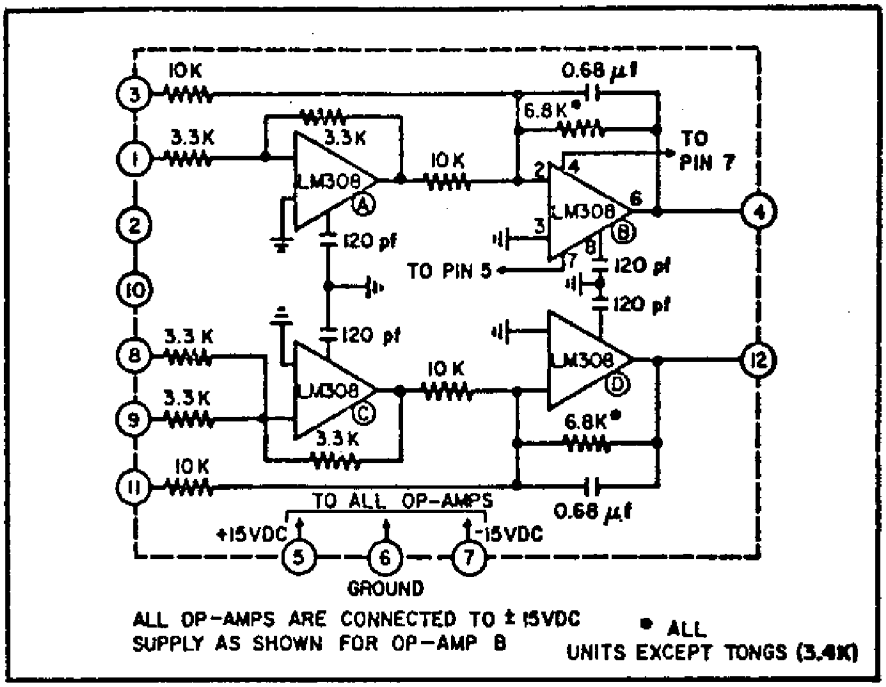


Fig. B.3. Comparator Circuit Schematic (from Brooks[3])

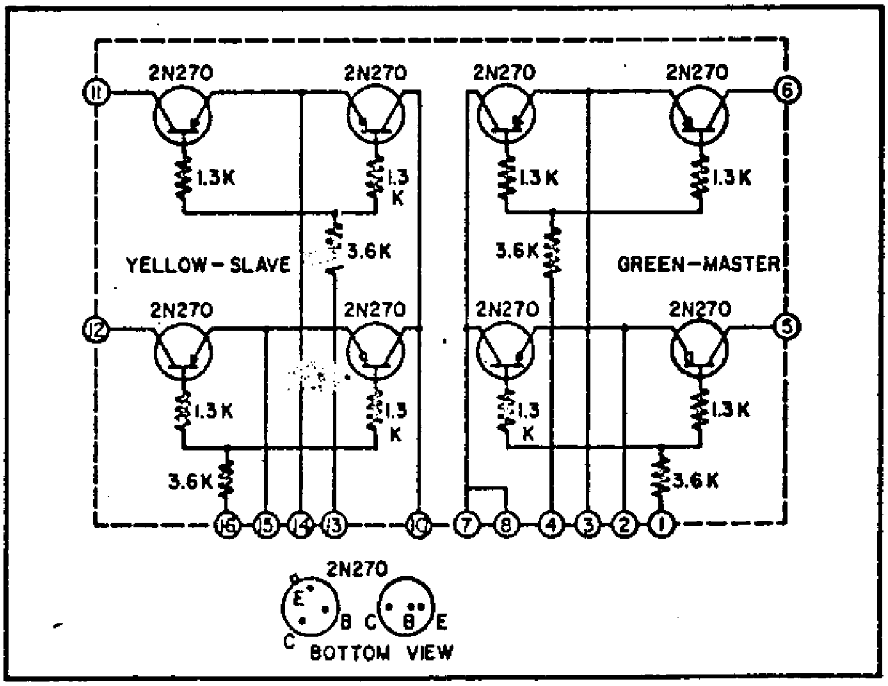


Fig. B.4. Modulator Circuit Schematic (from Brooks[3])

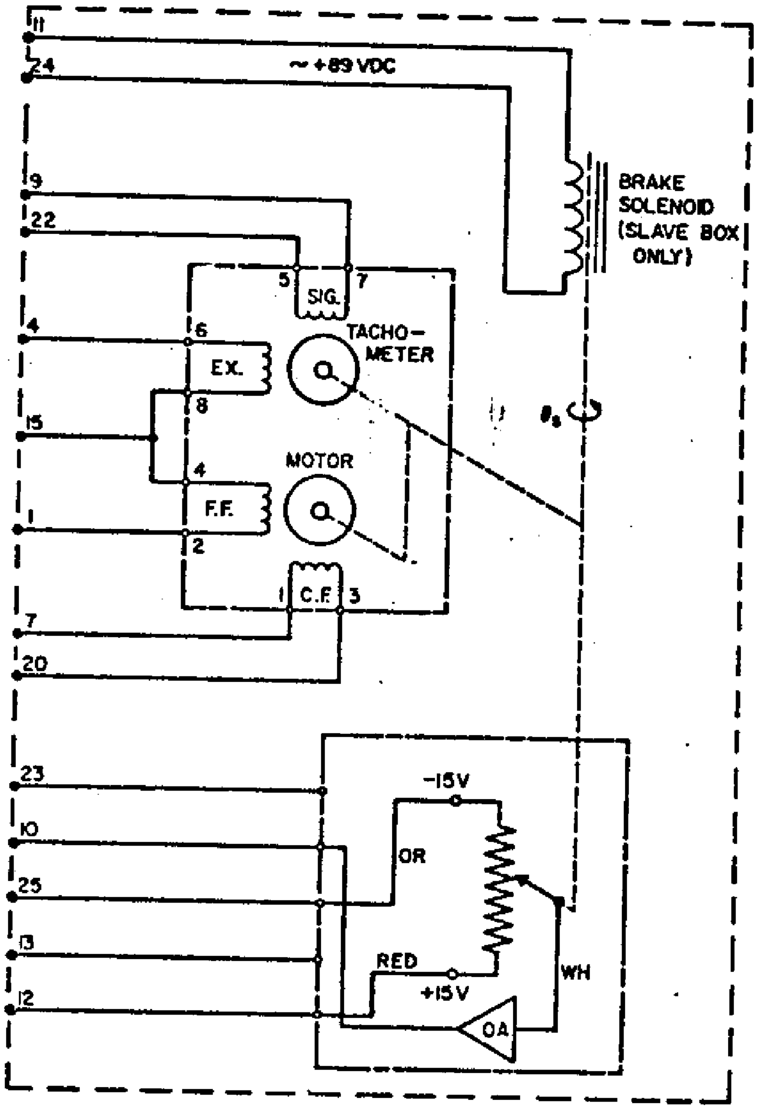


Fig. B.5. Gearbox Schematic (from Brooks[3])

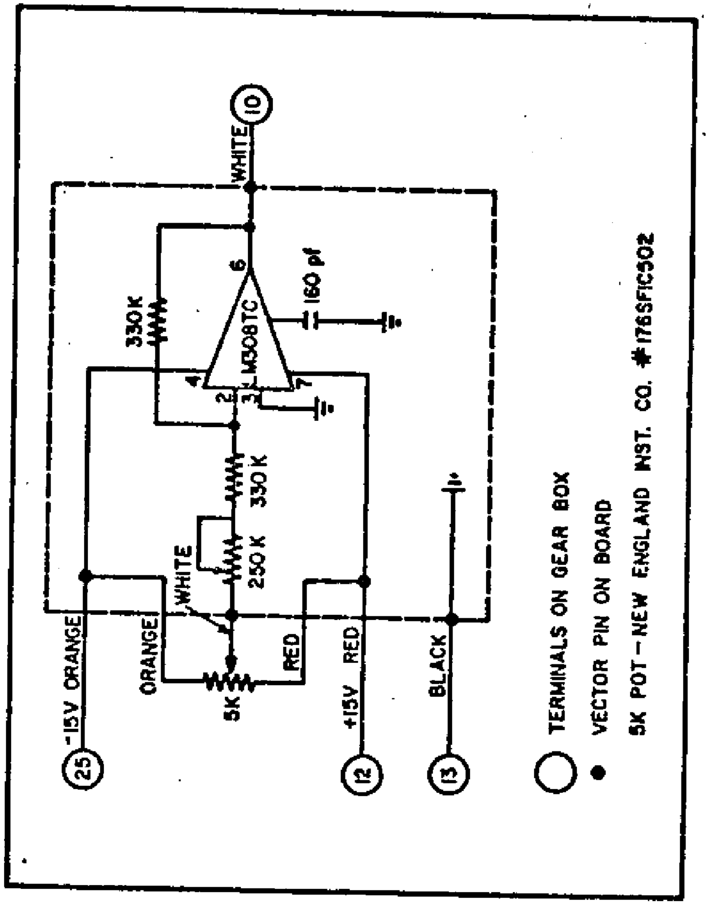
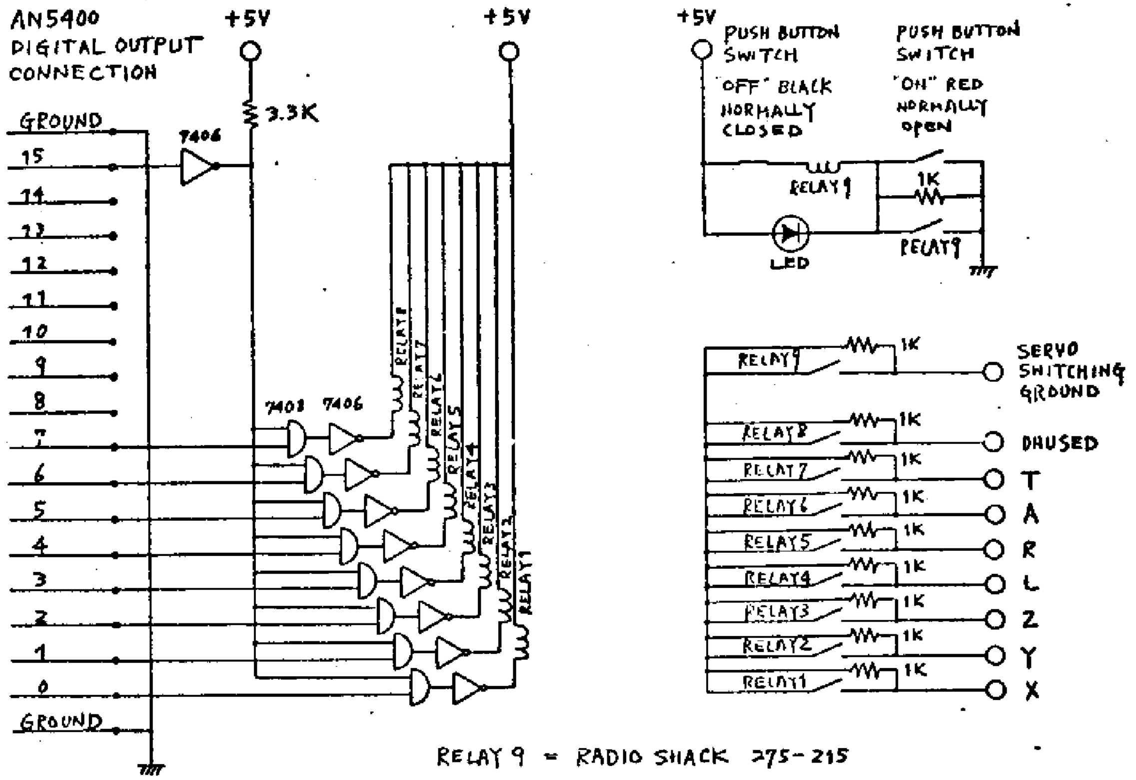


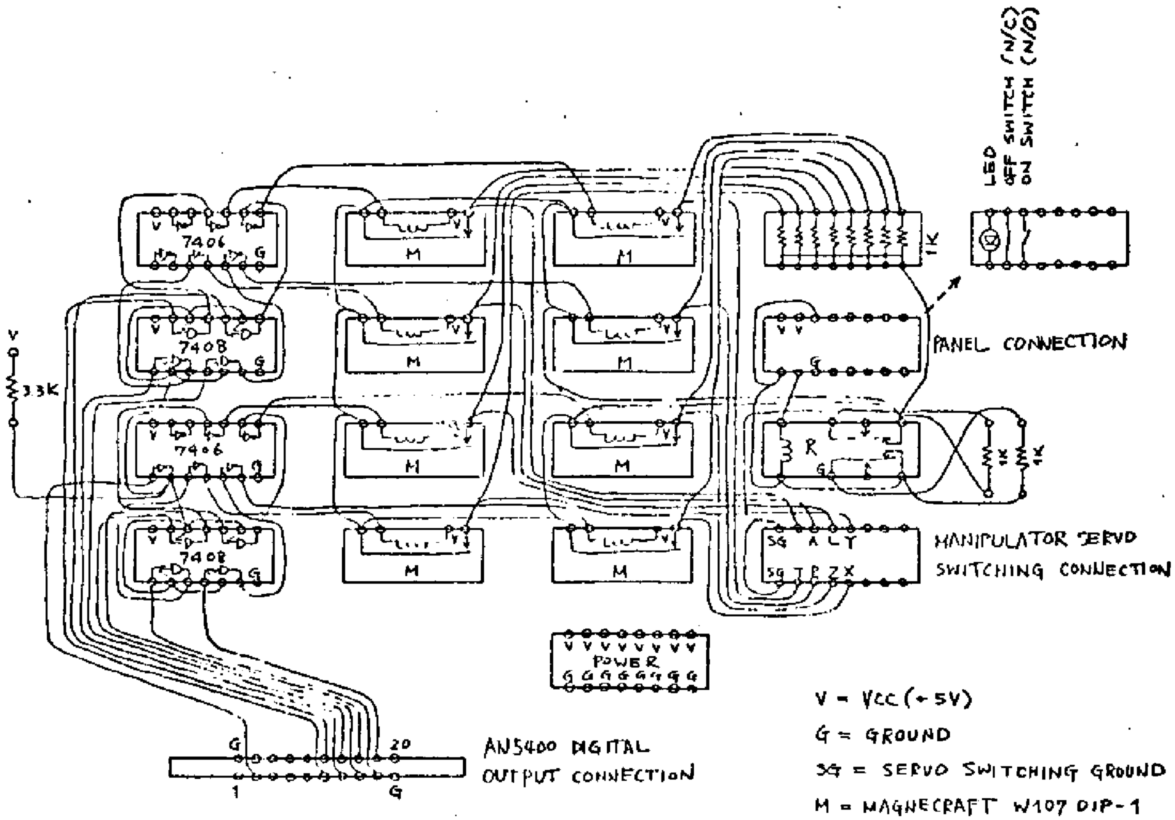
Fig. B.6. Gearbox Potentiometer and Line Driver Schematic (from Brooks[3])



RELAY 9 = RADIO SHACK 275-215

RELAY 1~8 = MAGNECRAFT W107DIP-1

Fig. B.7. Drawing of Control Mode Switching Circuit



V = Vcc (+5V)

G = GROUND

SG = SERVO SWITCHING GROUND

M = MAGNECRAFT W107 DIP-1

R = RADIO SHACK 275-215

Fig. B.8. Breadboard Wiring of Control Mode Switching Circuit

APPENDIX C. THE TABLE AND ITS INTERFACE

A table having a carriage which moves along the x, y, and z axes is provided for the project. The carriage is driven by three stepping motors, z overriding x and x overriding y. The motion is realized by a timing belt-pulley mechanism along the x and y axes and by rack-and-pinion along the z axis. The strokes are 14 inches, 16 inches, and 2 inches along x, y, and z axes, respectively. A translator is used to convert the forward and reverse pulse signals into the phase-shifting motor driving currents.

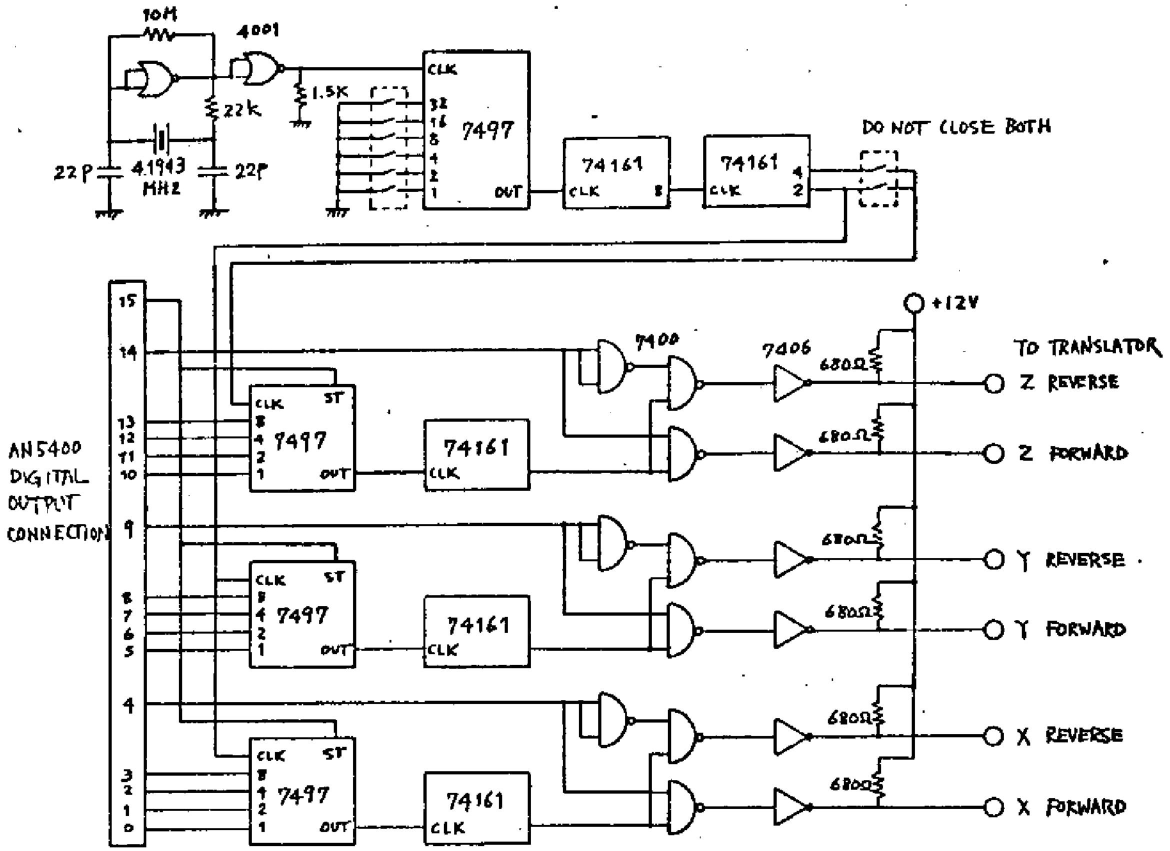
An interface circuit was made for the computer to control the table. Figure C.1 shows the circuit. The most interesting IC chip used in this circuit was the TTL 7497 Binary Rate Multiplier. Essentially, it has six bits of rate input and one clock input and one clock output, and it outputs as many clock pulses for every 64 input clock pulses as specified by the six bits (0 - 63). The crystal oscillator generates 4.1943 MHz clock pulse and the first BRM counts it down to 0 - 63 (set by the manual switches) for every 64. The next two 4-bit counters in series count it down to one sixty-fourth. Next come three BRM's, each of which counts the pulse down to 0 - 15 (set by the digital output signals from the computer) for every 64. In this sequence, the last counter can output 0, 1(1 x 1), up to 945(63 x 15) pulses per second.

The computer gives 16 bits of digital output signal to this circuit. Fifteen of the bits are given to the three axes five for each. One of the five bits specifies the direction, "one" forward and "zero" reverse, and the remaining four bits assign the sixteen levels of pulse rate (0 - 15). The last one bit inhibits the pulse output of the circuit. This invalidates the spurious signal of all ones that come when the cable is disconnected.

If the computer is working with a 20 Hz clock, the manual rate switch can be set 20, and the computer can update the digital output every twentieth of a second giving the motor 0 to 15 pulses every cycle, resulting in 0 to 300 pulses per second.

The cable from the panel, used in place of that from AN5400, allows manual test of the motor drive.

Figure C.2 shows the breadboard wiring of the circuit.



Fig, C.1. Drawing of Table Circuit

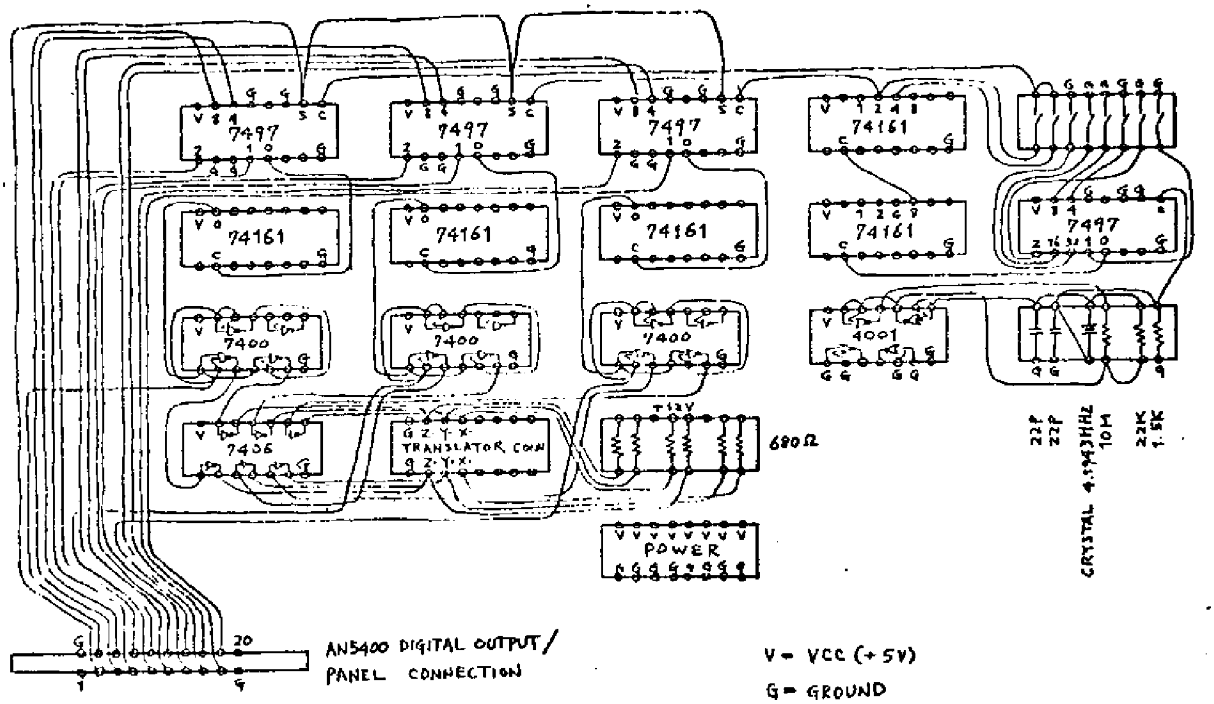


Fig. C.2. Breadboard Wiring of Table Circuit

APPENDIX D. TABLES OF EXPRESSION OF VARIABLES

Table D.1 Transform Matrix from Frame 4 to Frame 0

$$
{}^{0}A_{4} = \begin{bmatrix} {}^{4}a_{11} & {}^{4}a_{12} & {}^{4}a_{13} & {}^{4}a_{14} \\ {}^{4}a_{21} & {}^{4}a_{22} & {}^{4}a_{23} & {}^{4}a_{24} \\ {}^{4}a_{31} & {}^{4}a_{32} & {}^{4}a_{33} & {}^{4}a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

${}^{4}a_{11} = C2C4 + S2S3S4$
${}^{4}a_{12} = S2S3C4-C2S4$
${}^{4}a_{13} = S2C3$

${}^{4}a_{21} = S1S2C4 + C1C3S4 - S1C2S3S4$
${}^{4}a_{22} = C1C3C4-S1C2S3C4-S1S2S4$
${}^{4}a_{23} = -S1C2C3-C1S3$

${}^{4}a_{31} = -C1S2C4+S1C3S4+C1C2S3S4$
${}^{4}a_{32} = S1C3C4+C1C2S3C4+C1S2S4$
${}^{4}a_{33} = C1C2C3-S1S3$

${}^{4}a_{14} = x = 1.39(S2S3C4-C2S4)-40S2C3$
${}^{4}a_{24} = y = 1.39(C1C3C4-S1C2S3C4-S1S2S4)+40(S1C2C3+C1S3)+18C1$
${}^{4}a_{34} = z = 1.39(S1C3C4+C1C2S3C4+C1S2S4)+40(-C1C2C3+S1S3)+18S1$

${}^{4}\vec{x} = ({}^{4}a_{11}, {}^{4}a_{21}, {}^{4}a_{31})$
${}^{4}\vec{y} = ({}^{4}a_{12}, {}^{4}a_{22}, {}^{4}a_{32})$
${}^{4}\vec{z} = ({}^{4}a_{13}, {}^{4}a_{23}, {}^{4}a_{33})$

Table D.2 Transform Matrix from Frame 6 to Frame 0

$$
{}^{0}A_{6} = \begin{bmatrix} {}^{6}a_{11} & {}^{6}a_{12} & {}^{6}a_{13} & {}^{6}a_{14} \\ {}^{6}a_{21} & {}^{6}a_{22} & {}^{6}a_{23} & {}^{6}a_{24} \\ {}^{6}a_{31} & {}^{6}a_{32} & {}^{6}a_{33} & {}^{6}a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

${}^{6}a_{11} = (C2C4+S2S3S4)C6+[-S2C3C5+(S2S3C4-C2S4)S5]S6$

${}^{6}a_{12} = (S2S3C4-C2S4)C5+S2C3S5$

${}^{6}a_{13} = [S2C3C5+(-S2S3C4+C2S4)S5]C6+(C2C4+S2S3S4)S6$

${}^{6}a_{21} = (S1S2C4+C1C3S4-S1C2S3S4)C6$
$\quad +[(S1C2C3+C1S3)C5+(C1C3C4-S1C2S3C4-S1S2S4)S5]S6$
${}^{6}a_{22} = (C1C3C4-S1C2S3C4-S1S2S4)C5-(S1C2C3+C1S3)S5$
${}^{6}a_{23} = [-(S1C2C3+C1S3)C5+(-C1C3C4+S1C2S3C4+S1S2S4)S5]C5$
$\quad +(S1S2C4+C1C3S4-S1C2S3S4)S6$

${}^{6}a_{31} = (-C1S2C4+S1C3S4+C1C2S3S4)C6$
$\quad +[(-C1C2C3+S1S3)C5+(S1C3C4+C1C2S3C4+C1S2S4)S5]S6$
${}^{6}a_{32} = (S1C3C4+C1C2S3C4+C1S2S4)C5+(C1C2C3-S1S3)S5$
${}^{6}a_{33} = [(C1C2C3-S1S3)C5-(S1C3C4+C1C2S3C4+C1S2S4)S5]C6$
$\quad +(-C1S2C4+S1C3S4+C1C2S3S4)S6$

${}^{6}a_{14} = x = 1.39(S2S3C4-C2S4)-40S2C3$
${}^{6}a_{24} = y = 1.39(C1C3C4-S1C2S3C4-S1S2S4)+40(S1C2C3+C1S3)+18C1$
${}^{6}a_{34} = z = 1.39(S1C3C4+C1C2S3C4+C1S2S4)+40(-C1C2C3+S1S3)+18S1$

${}^{6}\vec{x} = ({}^{6}a_{11}, {}^{6}a_{21}, {}^{6}a_{31})$
${}^{6}\vec{y} = ({}^{6}a_{12}, {}^{6}a_{22}, {}^{6}a_{32})$
${}^{6}\vec{z} = ({}^{6}a_{13}, {}^{6}a_{23}, {}^{6}a_{33})$

Table D.3 Expression of the Variables Used in the RMRC

$x = 1.39(S2S3C4-C2S4)-40S2C3$

$\partial x/\partial\theta_1 = 0$

$\partial x/\partial\theta_2 = 1.39(C2S3C4+S2S4)-40C2C3$

$\partial x/\partial\theta_3 = 1.39S2C3C4+40S2S3$

$\partial x/\partial\theta_4 = -1.39(C2C4+S2S3S4)$

$y = 1.39(C1C3C4-S1C2S3C4-S1S2S4)+40(S1C2C3+C1S3)+18C1$

$\partial y/\partial\theta_1 = -1.39(S1C3C4+C1C2S3C4+C1S2S4)+40(C1C2C3-S1S3)-18S1$

$\partial y/\partial\theta_2 = 1.39(S1S2S3C4-S1C2S4)-40S1S2C3$

$\partial y/\partial\theta_3 = -1.39(S1C2C3C4+C1S3C4)+40(C1C3-S1C2S3)$

$\partial y/\partial\theta_4 = 1.39(-S1S2C4-C1C3S4+S1C2S3S4)$

$z = 1.39(S1C3C4+C1C2S3C4+C1S2S4)+40(-C1C2C3+S1S3)+18S1$

$\partial z/\partial\theta_1 = 1.39(C1C3C4-S1C2S3C4-S1S2S4)+40(S1C2C3+C1S3)+18C1$

$\partial z/\partial\theta_2 = 1.39(-C1S2S3C4+C1C2S4)+40C1S2C3$

$\partial z/\partial\theta_3 = 1.39(C1C2C3C4-S1S3C4)+40(S1C3+C1C2S3)$

$\partial z/\partial\theta_4 = 1.39(C1S2C4-S1C3S4-C1C2S3S4)$

$p = s\cdot{}^4a_{11}+{}^4a_{12} = s\cdot px+py$

$px = C2C4+S2S3S4$

$\partial px/\partial\theta_1 = 0$

$\partial px/\partial\theta_2 = -S2C4+C2S3S4$

$\partial px/\partial\theta_3 = S2C3S4$

$\partial px/\partial\theta_4 = S2S3C4-C2S4$

$py = S1S2C4+C1C3S4-S1C2S3S4$

$\partial py/\partial\theta_1 = C1S2C4-S1C3S4-C1C2S3S4$

$\partial py/\partial\theta_2 = S1C2C4+S1S2S3S4$

$\partial py/\partial\theta_3 = -S1C2C3S4-C1S3S4$

$\partial py/\partial\theta_4 = C1C3C4-S1C2S3C4-S1S2S4$

For ${}^6\vec{y}$ to be horizontal,

$(S1C3C4+C1C2S3C4+C1S2S4)C5+(C1C2C3-S1S3)S5 = 0,$

which gives

$\theta_{5r} = \arctan\,[-(S1C3C4+C1C2S3C4+C1S2S4)/(C1C2C3-S1S3)]$

(cont'd)

Page Two
Table D.3

For ${}^6\vec{x}$ to be horizontal,

$(-C1S2C4+S1C3S4+C1C2S3S4)C6$

$+[(-C1C2C3+S1S3)C5+(S1C3C4+C1C2S3C4+C1S2S4)S5]S6 = 0$

which gives

$\theta_{6r} = \arctan[(-C1S2C4+S1C3S4+C1C2S3S4)$

$/\{(C1C2C3-S1S3)C5-(S1C3C4+C1C2S3C4+C1S2C4)S5\}]$

Table D.4 Simplified Expression of the Variables Used in the RMRC

Notation

For example, $x1 = \partial x/\partial\theta_1$, $px3 = \partial px/\partial\theta_3$.

Variables beginning by a capital letter are temporary variables.

Ypart = S1C2C3+C1S3
Zpart = -C1C2C3+S1S3
X2part = C2S3C4+S2S4
Pz4 = S1C3C4+C1·X2part

px = C2C4+S2S3S4
px1 = 0
px2 = -S2C4+C2S3S4
px3 = S2C3S4
px4 = S2S3C4-C2S4

py = -S1·px2+C1C3S4
py1 = -C1·px2-S1C3S4
py2 = S1·px
py3 = -Ypart·S4
py4 = C1C3C4-S1·X2part

x = 1.39px4-40S2C3
y = 1.39py4+40Ypart+18C1
z = 1.39Pz4+40Zpart+18S1

x1 = 0
x2 = 1.39X2part-40C2C3
x3 = 1.39S2C3C4+40S2S3
x4 = -1.39px

y1 = -z
y2 = 1.39S1·px4-40S1S2C3
y3 = -1.39Ypart·C4+40(C1C3-S1C2S3)
y4 = -1.39py

z1 = y
z2 = 1.39C1·px4+40C1S2C3
z3 = -1.39Zpart·C4+40(S1C3+C1C2S3)
z4 = 1.39 py1

(cont'd)

Page Two
Table D.4

F123 = S1C3+C1C2S3
TN5 = F123·C4+C1S2S4
TD5 = C1C2C3-S1S3
θ_{5r} = arctan(-TN5/TD5)
θ_{6r} = arctan[(-C1S2C4+F123·S4)/(TD5·C5-TN5·S5)]

APPENDIX E. PROGRAM

The program consists of the main part MANEX1, resolved motion rate control subroutines MRESS and MRESM, voltage-radian conversion subroutine MANGL, and AN5400 controlling subroutine AN5402.

MANEX1 generates the object motion and controls the master/slave operation as described in Chapter 6. MRESS and MRESM are the same except that they deal with different areas of the data (MRESS for the slave and MRESM for the master). Such duplication could be avoided in assembly language coding by use of an index register to switch the data area, or still better and faster, by use of the memory management register if possible as with the PDP11/34.

The singularity of the Jacobian matrix was not taken care of in the program assuming that operators do not give the manipulator such posture that makes the matrix singular.

Figure E.1 shows the general flowchart of the program.

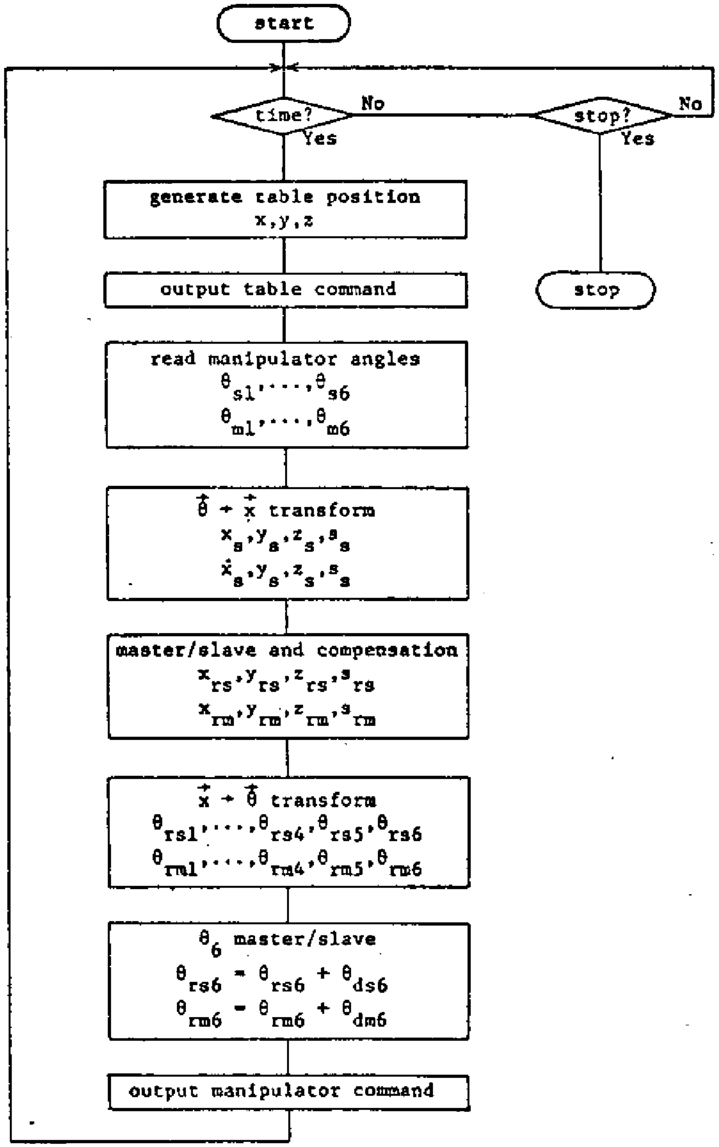


Fig. E.1. General Flowchart of the Program

```
C       MANIPULATOR AND TABLE
C       MANIPULATION OF MOVING OBJECT
        COMMON THXSI(7),THXMI(7),THXSO(7),THXMO(7)
        COMMON THS1,THS2,THS3,THS4,THS5,THS6,XS,YS,ZS,SS,XSD,YSD,ZSD,SSD
        COMMON THM1,THM2,THM3,THM4,THM5,THM6,XM,YM,ZM,SM,XMD,YMD,ZMD,SMD
        COMMON IDATA(14)
C
        THXSO(1)=0.0
        THXMO(1)=0.0
C
        CALL ANINIT
        CALL DOUT(24,0)
        CALL DOUT(26,0)
        TYPE *,'MANIPULATION OF MOVING OBJECT WITH COMPENSATION'
        A=0.7
        B=0.5
        A1=1.0-A
        TYPE *,'MANIPULATOR COMPUTER CONTROL?'
        ACCEPT *,I
        IF(I.NE.1) STOP
        CALL AINSQ(16,29,IDATA)
        CALL AOUTSQ(4,17,IDATA)
        CALL DOUT(24,63)
C
C       SINUSOIDAL MOTION OF THE TABLE
C       STOPS WHEN AN5400 DIAL IS NON-ZERO
C       STOPS WHEN ANY LIMIT SWITCH IS HIT
C       STOPS WHEN TIME FAILURE OCCURS
        TYPE *,'SET FUNDAMENTAL FREQUENCIES FX=FY=FZ=20HZ'
        ACCEPT *,I
        IF(I.NE.1) STOP
        PT1=5.0
        PT2=6.5
        PT3=14.5
        PT4=18.5
        PW1=0.31416/PT1
        PW2=0.31416/PT2
        PW3=0.31416/PT3
        PW4=0.31416/PT4
        TYPE *,'SET TABLE AT CENTER AND TURN ON TRANSLATOR'
        TYPE *,'THIS INITIATES COMPUTER CONTROLLED MASTER/SLAVE'
        ACCEPT *,I
        IF(I.NE.1) STOP
C
        T201=-20.0
        IX=0
        IY=0
        IZ=0
        IX1=0
        IY1=0
        IZ1=0
        IX2=0
        IY2=0
        IZ2=0
```

```
        IX3=0
        IY3=0
        IZ3=0
        X=0.0
        Y=0.0
        Z=0.0
        X1=0.0
        Y1=0.0
        Z1=0.0
        X2=0.0
        Y2=0.0
        Z2=0.0
        X3=0.0
        Y3=0.0
        Z3=0.0
        T1=SECNDS(0.0)+4.0
        GO TO 500
C
C       TABLE MOTION
100     IZ=ININT(34.0*COS(T20*PW1)-34.0*COS(T20*PW2))
        IX=ININT(100.0*SIN(T20*PW1)-400.0*SIN(T20*PW3))
        IY=ININT(100.0*SIN(T20*PW2)-400.0*SIN(T20*PW4))
        X=0.014*FLOAT(IX)
        Y=0.014*FLOAT(IY)
        Z=0.0117809*FLOAT(IZ)
        IVX=IX1-IX2
        IVY=IY1-IY2
        IVZ=IZ2-IZ3
        ISX=1
        ISY=1
        ISZ=1
        IF(IVX.GT.0) GO TO 120
        ISX=0
        IVX=-IVX
120     IF(IVY.GT.0) GO TO 130
        ISY=0
        IVY=-IVY
130     IF(IVZ.GT.0) GO TO 140
        ISZ=0
        IVZ=-IVZ
140     JDATA=16384*ISZ+512*ISY+16*ISX
        JDATA=JDATA+1024*IVZ+32*IVY+IVX
        CALL DOUT(26,JDATA)
C
C       MASTER/SLAVE
300     CALL MANGLI
        THS1=THXSI(5)
        THS2=THXSI(7)
        THS3=THXSI(6)-THS1
        THS4=THXSI(2)
        CALL MRESS1
        THM1=THXMI(5)
        THM2=THXMI(7)
        THM3=THXMI(6)-THM1
        THM4=THXMI(2)
        CALL MRESM1
```

```
      XSE=XS-X2
      YSE=YS-Y2
      ZSE=ZS-Z3
      XSD=A*XM+A1*XSE+X
      YSD=A*YM+A1*YSE+Y
      ZSD=A*ZM+A1*ZSE+Z
      SSD=A*SM+A1*SS
      XMD=A*XSE+A1*XM
      YMD=A*YSE+A1*YM
      ZMD=A*ZSE+A1*ZM
      SMD=A*SS+A1*SM
      IF(T20.LT.-5.0) GO TO 320
      XSD=XSD+B*(XSE-XSE2)
      YSD=YSD+B*(YSE-YSE2)
      ZSD=ZSD+B*(ZSE-ZSE2)
      SSD=SSD+B*(SS-SS2)
      XMD=XMD+B*(XM-XM2)
      YMD=YMD+B*(YM-YM2)
      ZMD=ZMD+B*(ZM-ZM2)
      SMD=SMD+B*(SM-SM2)
320   CALL MRESS2
      CALL MRESM2
C
      THSI6=(THXSI(4)-THXSI(3))/1.65-THS6
      THMI6=(THXMI(4)-THXMI(3))/1.65-THM6
      THSO6=A*THMI6+A1*THSI6+THS6
      THMO6=A*THSI6+A1*THMI6+THM6
      IF(T20.LT.-10.0) GO TO 340
      THSO6=THSO6+B*(THSI6-THSI62)
      THMO6=THMO6+B*(THMI6-THMI62)
340   CONTINUE
C
      THXSO(2)=THS4
      THXSO(3)=THS5-THSO6*0.825-THS3*0.27
      THXSO(4)=THS5+THSO6*0.825-THS3*0.27
      THXSO(5)=THS1
      THXSO(6)=THS1+THS3
      THXSO(7)=THS2
      THXMO(2)=THM4
      THXMO(3)=THM5-THMO6*0.825-THM3*0.27
      THXMO(4)=THM5+THMO6*0.825-THM3*0.27
      THXMO(5)=THM1
      THXMO(6)=THM1+THM3
      THXMO(7)=THM2
      CALL MANGLO
      CALL AOUTSQ(4,17,IDATA)
C
400   T201=T201+1.0
      IX3=IX2
      IY3=IY2
      IZ3=IZ2
      IX2=IX1
      IY2=IY1
      IZ2=IZ1
```

```
      IX1=IX
      IY1=IY
      IZ1=IZ
      X3=X2
      Y3=Y2
      Z3=Z2
      X2=X1
      Y2=Y1
      Z2=Z1
      X1=X
      Y1=Y
      Z1=Z
      XSE2=XSE1
      YSE2=YSE1
      ZSE2=ZSE1
      SS2=SS1
      THSI62=THSI61
      XM2=XM1
      YM2=YM1
      ZM2=ZM1
      SM2=SM1
      THMI62=THMI61
      XSE1=XSE
      YSE1=YSE
      ZSE1=ZSE
      SS1=SS
      THSI61=THSI6
      XM1=XM
      YM1=YM
      ZM1=ZM
      SM1=SM
      THMI61=THMI6
C
500   CALL DIN(256,JDATA)
      IF(JDATA.NE.0) GO TO 900
      CALL DIN(20,JDATA)
      IF(JDATA.LT.0) JDATA=JDATA+16384+16384
      JDATA=JDATA-JDATA/64*64
      IF(JDATA.NE.63) GO TO 920
      T20=AINT((SECNDS(T1)+0.008)*20.0)
      IF(T20.LT.T201) GO TO 500
      IF(T20.GT.T201) GO TO 910
      IF(T20.LT.0.0) GO TO 300
      GO TO 100
C
920   CALL DOUT(26,0)
      TYPE *,'LIMIT SWITCH'
      GO TO 900
910   CALL DOUT(26,0)
      TYPE *,'TIME FAILURE'
      GO TO 900
900   CALL DOUT(26,0)
      TYPE *,'TABLE STOPPED, MANIPULATOR LEFT UNRELEASED'
      TYPE *,'TURN OFF MANIPULATOR'
      STOP
      END
```

```
      SUBROUTINE MRESS1
      COMMON THXSI(7),THXMI(7),THXSO(7),THXMO(7)
      COMMON TH1,TH2,TH3,TH4,TH5,TH6,X,Y,Z,S,XD,YD,ZD,SD
      COMMON FILLER(14)
      COMMON IDATA(14)
      EQUIVALENCE (A11,X2),(A12,X1),(A13,X4),(A14,X3)
      EQUIVALENCE (A21,Z2),(A22,Z1),(A23,Z4),(A24,Z3)
      EQUIVALENCE (A31,P2,PY2),(A32,P1,PY1),(A33,P4,PY4),(A34,P3,PY3)
      EQUIVALENCE (A41,Y2),(A42,Y1),(A43,Y4),(A44,Y3)
      EQUIVALENCE (Q1,DX,DTH2),(Q2,DZ,DTH1),(Q3,DP,DTH4),(Q4,DY,DTH3)
      EQUIVALENCE (P,PY),(SD,TX)
C
C     RESOLVED MOTION RATE CONTROL
      C1=COS(TH1)
      C2=COS(TH2)
      C3=COS(TH3)
      C4=COS(TH4)
      S1=SIN(TH1)
      S2=SIN(TH2)
      S3=SIN(TH3)
      S4=SIN(TH4)
      C1C3=C1*C3
      C1S3=C1*S3
      S1C3=S1*C3
      S1S3=S1*S3
      C2C4=C2*C4
      C2S4=C2*S4
      S2C4=S2*C4
      S2S4=S2*S4
C
      YPART=C1S3+S1C3*C2
      ZPART=-C1C3*C2+S1S3
      X2PART=C2C4*S3+S2S4
      PZ4=C1*X2PART+S1C3*C4
      PX=C2C4+S2S4*S3
C     PX1=0.0
      PX2=C2S4*S3-S2C4
      PX3=S2S4*C3
      PX4=-C2S4+S2C4*S3
      PY=C1C3*S4-S1*PX2
      PY1=-C1*PX2-S1C3*S4
      PY2=S1*PX
      PY3=-YPART*S4
      PY4=C1C3*C4-S1*X2PART
C
      X=1.39*PX4-40.0*S2*C3
      Y=1.39*PY4+40.0*YPART+18.0*C1
      Z=1.39*PZ4+40.0*ZPART+18.0*S1
      S=-PY/PX
C     X1=0.0
      X2=1.39*X2PART-40.0*C2*C3
      X3=1.39*S2C4*C3+40.0*S2*S3
      X4=-1.39*PX
```

```
      Y1=-Z
      Y2=1.39*S1*PX4-40.0*S1C3*S2
      Y3=-1.39*YPART*C4+40.0*(C1C3-S1S3*C2)
      Y4=-1.39*PY
      Z1=Y
      Z2=-1.39*C1*PX4+40.0*C1C3*S2
      Z3=-1.39*ZPART*C4+40.0*(C1S3*C2+S1C3)
      Z4=1.39*PY1
      Y=Y+40.0
      Z=Z-19.39
      RETURN
C
      ENTRY MRESS2
      P=PY+TX*PX
      P2=PY2+TX*PX2
      P3=PY3+TX*PX3
      P4=PY4+TX*PX4
      DP=-P
      DX=XD-X
      DY=YD-Y
      DZ=ZD-Z
C
C     SOLVING THE RATE EQUATION
      A13=A13/A11
      A23=(A23-A21*A13)/A22
      A33=A33-A31*A13-A32*A23
      A43=A43-A41*A13-A42*A23
      A14=A14/A11
      A24=(A24-A21*A14)/A22
      A34=(A34-A31*A14-A32*A24)/A33
      A44=A44-A41*A14-A42*A24-A43*A34
      Q1=Q1/A11
      Q2=(Q2-A21*Q1)/A22
      Q3=(Q3-A31*Q1-A32*Q2)/A33
      Q4=(Q4-A41*Q1-A42*Q2-A43*Q3)/A44
      Q3=Q3-A34*Q4
      Q2=Q2-A24*Q4-A23*Q3
      Q1=Q1-A14*Q4-A13*Q3
C
C     SOLVING TH5 AND TH6
      SS1=S1+DTH1*C1
      CC1=C1-0.5*DTH1*(S1+SS1)
      SS2=S2+DTH2*C2
      CC2=C2-0.5*DTH2*(S2+SS2)
      SS3=S3+DTH3*C3
      CC3=C3-0.5*DTH3*(S3+SS3)
      SS4=S4+DTH4*C4
      CC4=C4-0.5*DTH4*(S4+SS4)
      C1C2=CC1*CC2
      C1S2=CC1*SS2
      F123=C1C2*SS3+SS1*CC3
      TH5=F123*CC4+C1S2*SS4
      TD5=C1C2*CC3-SS1*SS3
      TH5=ATAN2(-TN5,TD5)
      TH6=ATAN2(F123*SS4-C1S2*CC4,TD5*COS(TH5)-TN5*SIN(TH5))
```

```
C       NEW TH1,TH2,TH3,TH4
        TH1=TH1+DTH1
        TH2=TH2+DTH2
        TH3=TH3+DTH3
        TH4=TH4+DTH4
        RETURN
        END



        SUBROUTINE MRESM1
        COMMON THXSI(7),THXMI(7),THXSO(7),THXMO(7)
        COMMON FILLER(14)
        COMMON TH1,TH2,TH3,TH4,TH5,TH6,X,Y,Z,S,XD,YD,ZD,SD
        COMMON IDATA(14)
          .
          .
          .
        RETURN
C
        ENTRY MRESM2
          .
          .
          .
        RETURN
        END



        SUBROUTINE MANGLI
        COMMON THXSI(7),THXMI(7),THXSO(7),THXMO(7)
        COMMON FILLER(28)
        COMMON IDATA(14)
        DIMENSION COEF(7),ICONST(7)
        DATA COEF/.00048,.00142,.00173,-.00173,-.00043,-.00079,.00036/
        DATA ICONST/0,0,300,-300,0,0,0/
C
C       CONVERT ANGLE READINGS INTO RADIANS
        CALL AINSU(16,29,IDATA)
        DO 10 I=1,7
        THXSI(I)=FLOAT(IDATA(I)/16-ICONST(I))*COEF(I)
        THXMI(I)=FLOAT(-IDATA(I+7)/16-ICONST(I))*COEF(I)
10      CONTINUE
        RETURN
C
        ENTRY MANGLO
C       CONVERT ANGLE RADIANS INTO OUTPUT FORMS
        DO 20 I=1,7
        IDATA(I)=(ININT(THXSO(I)/COEF(I))+ICONST(I))*16
        IDATA(I+7)=-(ININT(THXMO(I)/COEF(I))+ICONST(I))*16
20      CONTINUE
        RETURN
        END
```

```
;CONTROL SUBROUTINES OF AN5400 THROUGH DRII-C
        .GLOBL  ANINIT,DIN,DOUT,AIN,AINSQ,AOUT,AOUTSQ
        .GLODL  STATUS,OUTBUF,INBUF
;
DIN:    TST     (R5)+
        MOV     #20000,R0       ;'READ DIGITAL DATA'
        ADD     @(R5)+,R0       ;+ ADDRESS
DIN1:   TSTB    STATUS
        BPL     DIN1
        MOV     R0,OUTBUF       ;INPUT COMMAND
DIN2:   TSTB    STATUS
        BPL     DIN2
        MOV     INBUF,@(R5)+    ;INPUT DATA
        RTS     PC
;
DOUT:   TST     (R5)+
        MOV     #110000,R0      ;'LOAD DAC DATA'
        ADD     @(R5)+,R0       ;+ ADDRESS
DOUT1:  TSTB    STATUS
        BPL     DOUT1
        MOV     R0,OUTBUF       ;INPUT COMMAND
DUUT2:  TSTB    STATUS
        BPL     DOUT2
        MOV     @(R5)+,OUTBUF   ;OUTPUT DATA
        RTS     PC
;
AIN:    TST     (R5)+
        MOV     #50000,R0       ;'LOAD ADDRESS'
        ADD     @(R5)+,R0       ;+ ADDRESS
AIN1:   TSTB    STATUS
        BPL     AIN1
        MOV     R0,OUTBUF       ;OUTPUT COMMAND
AIN2:   TSTB    STATUS
        BPL     AIN2
        CLR     OUTBUF          ;'READ A/D DATA'
AIN3:   TSTB    STATUS
        BPL     AIN3
        MOV     INBUF,@(R5)+    ;INPUT DATA
        RTS     PC
;
AINSQ:  TST     (R5)+
        MOV     #50000,R0       ;'LOAD ADDRESS'
        ADD     @(R5)+,R0       ;+ FIRST ADDRESS
        MOV     #50000,R1       ;'LOAD ADDRESS'
        ADD     @(R5)+,R1       ;+ LAST ADDRESS
        MOV     (R5)+,R2        ;DATA ADDRESS
AINS1:  TSTB    STATUS
        BMI     AINS4
        BR      AINS1
AINS2:  TSTB    STATUS
        BPL     AINS2
        CLR     OUTBUF          ;'READ A/D DATA'
AINS3:  TSTB    STATUS
        BPL     AINS3
```

```
        MOV     INBUF,(R2)+     ;INPUT DATA
AINS4:  MOV     R0,OUTBUF       ;INPUT COMMAND
        INC     R0
        CMP     R0,R1
        BLOS    AINS2
AINS5:  TSTB    STATUS
        BPL     AINS5
        CLR     OUTBUF          ;'READ A/D DATA'
AINS6:  TSTB    STATUS
        BPL     AINS6
        MOV     INBUF,(R2)+     ;INPUT DATA
        RTS     PC
;
AOUT:   TST     (R5)+
        MOV     #110000,R0      ;'LOAD DAC DATA'
        ADD     @(R5)+,R0       ;+ ADDRESS
AOUT1:  TSTB    STATUS
        BPL     AOUT1
        MOV     R0,OUTBUF       ;OUTPUT COMMAND
AOUT2:  TSTB    STATUS
        BPL     AOUT2
        MOV     @(R5)+,OUTBUF   ;OUTPUT DATA
        RTS     PC
;
AOUTSQ: TST     (R5)+
        MOV     #110000,R0      ;'LOAD DAC DATA'
        ADD     @(R5)+,R0       ;+ FIRST ADDRESS
        MOV     #110000,R1      ;'LOAD DAC DATA'
        ADD     @(R5)+,R1       ;+ LAST ADDRESS
        MOV     (R5)+,R2        ;DATA ADDRESS
AOS1:   TSTB    STATUS
        BPL     AOS1
        MOV     R0,OUTBUF       ;OUTPUT COMMAND
AOS2:   TSTB    STATUS
        BPL     AOS2
        MOV     (R2)+,OUTBUF    ;OUTPUT DATA
        INC     R0
        CMP     R0,R1
        BLOS    AOS1
        RTS     PC
;
ANINIT: CLR     STATUS
ANI1:   TSTB    STATUS
        BPL     ANI1
        MOV     #170000,OUTBUF  ;'CLEAR'
ANI2:   TSTB    STATUS
        BPL     ANI2
        MOV     #170000,OUTBUF  ;'CLEAR' AGAIN
ANI3:   TSTB    STATUS
        BPL     ANI3
        MOV     #60006,OUTBUF   ;'LOAD STATUS'
;                               ;CONV+SEQ
        RTS     PC
        .END
```

APPENDIX F. DATA

Table F.1. Operation Time for the Peg Moving Task

	TRIAL NUMBER	SUBJECT #1	SUBJECT #2	SUBJECT #3	
NO TABLE MOTION	1	20.6	22.4	22.6	
	2	24.8	20.0	24.2	
	3	19.0	22.0	20.2	
	4	18.2	18.6	20.6	
	5	20.0	18.2	22.6	
	E	20.5	20.2	22.0	20.9
	S	2.29	1.71	1.47	
COMPENSATION	1	27.4	22.8	32.2	
	2	24.0	25.4	39.6	
	3	23.8	21.8	28.0	
	4	22.4	27.0	33.4	
	5	23.0	23.0	32.8	
	E	24.1	24.0	33.2	27.1
	S	1.74	1.91	3.72	
NO COMPENSATION	1	35.2	30.8	72.2	
	2	30.8	37.2	44.8	
	3	32.8	26.4	42.8	
	4	38.0	38.4	68.8	
	5	34.4	28.6	55.0	
	E	34.2	32.3	56.7	41.1
	S	2.41	4.73	12.04	

Table F.2. Operation Time for the Valve Turning Task

	TRIAL NUMBER	SUBJECT #1	SUBJECT #2
NO TABLE MOTION	1	16.4	13.6
	2	15.0	10.6
	3	15.6	8.6
	4	12.0	8.8
	5	14.8	12.0
	E	14.8	10.7
	S	1.49	1.90
COMPENSATION	1	18.2	19.6
	2	13.8	11.0
	3	14.0	10.8
	4	17.8	10.4
	5	14.4	12.6
	E	15.6	12.9
	S	1.94	3.44
NO COMPENSATION	1	23.4	16.0
	2	15.6	11.4
	3	22.2	13.0
	4	16.4	11.6
	5	14.6	9.2
	E	18.4	12.2
	S	3.63	2.24

Table F.3. Comparative Errors in Rectangle Tracing

	TRIAL NUMBER	SUBJECT #1	SUBJECT #2	SUBJECT #4	
NO TABLE MOTION	1	0.206	0.181	0.212	
	2	0.125	0.156	0.175	
	3	0.156	0.150	0.231	
	4	0.131	0.162	0.131	
	5	0.150	0.175	0.206	
	6	0.138			
	E	0.151	0.165	0.191	0.169
	S	0.0269	0.0116	0.0350	
COMPENSATION	1	0.344	0.294	0.356	
	2	0.325	0.275	0.362	
	3	0.225	0.356	0.250	
	4	0.275	0.300	0.262	
	5	0.262	0.300	0.312	
	E	0.286	0.305	0.309	0.300
	S	0.0430	0.0272	0.0464	
NO COMPENSATION	1	0.781	0.775	1.156	
	2	0.762	0.744	0.806	
	3	0.812	0.731	0.894	
	4	0.669	0.600	1.038	
	5	0.550	0.762	0.812	
	E	0.715	0.722	0.941	0.793
	S	0.0954	0.0631	0.1360	

REFERENCES

[1] Ferrell,W.R., Sheridan,T.B., "Supervisory Control of Remote Manipulation", IEEE Spectrum, Vol. 4, No. 10, October 1967, pp. 81-88.

[2] Sheridan,T.B., Verplank,W.L., "Human and Computer Control of Undersea Teleoperators", Man-Machine Systems Laboratory Report, Massachusetts Institute of Technology, July 1978.

[3] Brooks,T.L., "Superman: A System for Supervisory Manipulation and the Study of Human/Computer Interactions", Master's thesis, Massachusetts Institute of Technology, May 1979.

[4] Whitney,D.E., "Resolved Motion Rate Control of Manipulators and Human Prostheses", IEEE Transactions on Man-Machine Systems, Vol. MMS-10, No. 2, June 1969.

[5] Sheridan,T.B., Ferrell,W.R. Man-Machine Systems, M.I.T. Press, Cambridge, MA, 1974

[6] Crandall,S.H., Engineering Analysis, McGraw-Hill, New York, 1956, pp. 26-31.