DESIGN AND DEVELOPMENT OF A MICROCOMPUTER CONTROLLED SMALL ARTICULATED ROBOT

BY

MAHMOOD HASAN

M.S. LENINGRAD POLYTECHNIC INSTITUTE, 1979

SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENT FOR THE DEGREE

MASTER OF SCIENCE

DEPARTMENT OF INDUSTRIAL ENGINEERING

KANSAS STATE UNIVERSITY

MANHATTAN, KANSAS

1983

APPROVED BY

MAJOR PROFESSOR

LD
2668
.T4
1983
H37
c.2

A11202 594098

ACKNOWLEDGEMENT

The author wishes to express his appreciation and gratitude to Dr. Muthuraj Vaithianathan for his guidance and encouragement throughout this study.

The author is indebted to Professor Jacob J. Smaltz and Dr. Ekramul Haque for serving in the graduate committee. The author also wishes to thank Mr. Bruce P. Mignano for the help and encouragement he rendered.

Last but certainly not the least, the author is indebted to his parents and his wife without whose inspiration, love, and patience this graduate program may not have materialized.

The author wishes to dedicate this work to his brother the Late Mr. Masud Nasir.

TABLE OF CONTENTS

			PAGE
ACKNOWLEDGEMENT			
LIST OF FIGURES			
LIST OF TABLES			
CHAPTER 1.		INTRODUCTION	
	1.1	HISTORICAL BACKGROUND	1
	1.2	DEFINITION	4
	1.3	CLASSIFICATION	7
	1.4	MAJOR COMPONENTS	11
	1.5	PROBLEM STATEMENT	14
	1.6	PURPOSE	15
	1.7	THE APPROACH - AN OVERVIEW	15
	1.8	EXISTING EQUIPMENT	18
CHAPTER 2.		MECHANICAL DESIGN	
	2.1	OVERVIEW OF TASK	19
	2.2	DESIGN OF ARM, HAND, AND GRIPPERS	19
	2.3	DESIGN OF Z AXIS VERTICAL DRIVE	22
	2.3.1	POWER REQUIREMENT FOR Z AXIS VERTICAL DRIVE	27
	2.4	ROTARY TABLE CONVERSION	30
	2.4.1	LEADSCREW POWER REQUIREMENT	31
	2.4.2	MOTOR ASSEMBLY FOR GEAR DRIVES	34
	2.4.3	MACHINING OF MECHANICAL COMPONENTS	37

2.5 MOTOR CHARACTERISTICS 40

2.6 TRAVEL LIMITS AND WORK ENVELOPE 41

CHAPTER 3 HARDWARE DESIGN

3.1 AC MOTORS DRIVE CIRCUIT 43

3.2 DC MOTOR DRIVE CIRCUIT 53

3.3 ELECTROMAGNET DRIVE CIRCUIT 59

CHAPTER 4 SOFTWARE DESIGN

4.1 TIMING CALCULATIONS REQUIRED FOR DELAY PROGRAMS 61

4.1.1 DELAY PROGRAM FOR X AXIS LINEAR DISPLACEMENT 61

4.1.2 DELAY PROGRAM FOR Y AXIS LINEAR DISPLACEMENT 64

4.1.3 DELAY PROGRAM FOR Z AXIS ROTARY DISPLACEMENT 66

4.1.4 DELAY PROGRAM FOR Z AXIS LINEAR DISPLACEMENT 69

4.1.5 GENERAL PURPOSE DELAY PROGRAMS 71

4.2 GENERAL OVERVIEW OF SOFTWARE 75

4.3 STEPS FOR RUNNING THE ROBOT 83

4.4 A SAMPLE PROGRAM 88

CHAPTER 5 CONCLUSIONS AND FUTURE DIRECTIONS

5.1 CONCLUSIONS 91

5.2 FUTURE DIRECTIONS 92

REFERENCES 94

APPENDIX 95

LIST OF FIGURES

PAGE

1. PATH OF NON SERVO AND SERVO CONTROLLED ROBOTS 9
2. ROBOT ARM CONFIGURATION 10
3. DEFINITION OF ROLL AND PITCH ANGLE 12
4. ARTICULATION OF ROBOT ARM 13
5. GENERAL CONCEPT OF OPERATION 17
6. GENERAL ARRANGEMENT OF ROBOT ARM 20
7. COMPONENTS OF ARM AND GRIPPING MECHANISM 21
8. Z AXIS LINEAR MECHANISM 23
9. Z AXIS LINEAR MECHANISM (VERTICAL GUIDE AND SLIDING BLOCK) 24
10. Z AXIS LINEAR MECHANISM (BASE PLATE AND TOP PLATE) 25
11. Z AXIS LINEAR MECHANISM (THREADED ROD) 26
12. SIMPLIFIED FORCE DIAGRAM OF Z AXIS 28
13. MOTOR - LEADSCREW COUPLING FOR AC MOTORS 35
14. DIMENSIONS OF A FACE PLATE 38
15. COMPLETED ROTARY TABLE WITH AC MOTORS 39
16. WORKING ENVELOPE OF ROBOT ARM 42
17. IC CIRCUIT FOR X (Y AND Z) AXIS AC MOTOR 45
18. RELAY AND MOTOR CIRCUIT FOR X (Y AND Z) AXIS AC MOTOR 46
19. DC STEPPER MOTOR PRINCIPLES OF OPERATION 54
20. DC MOTOR CIRCUIT FOR Z AXIS LINEAR MECHANISM 55
21. GRIPPER MECHANISM CIRCUIT 60
22. GENERAL ARRANGEMENT OF PROGRAM SECTIONS 76

23. LOGICAL FLOW DIAGRAM OF THE TASK TRANSFER SECTION AND THE TASK EXECUTION SECTION TOGETHER WITH ASSOCIATED INTERACTIONS 80

24. LOGICAL FLOW DIAGRAM OF THE AXIS MOVEMENT SECTION TOGETHER WITH ASSOCIATED INTERACTIONS 82

25. LIST OF TASKS IN SEQUENCE PRIOR TO PROGRAM EXECUTION 84

LIST OF TABLES

PAGE

1. GEAR SIZES FOR COUPLING DC MOTOR TO THE THREADED ROD 27
2. GEAR SIZES FOR COUPLING AC MOTORS TO VARIOUS AXES MECHANISM 36
3. DIMENSIONS OF THE WORMGEAR MECHANISM FOR ROTARY DISPLACEMENT ABOUT Z AXIS 37
4. MOTORS AND ELECTROMAGNET CHARACTERISTICS 40
5. WIRE CONNECTIONS FOR THE AC MOTORS 44
6. LOGIC TABLE FOR 4 TO 16 LINE DECODER CHIP 47
7. LOGIC TABLE FOR OR GATE 48
8. LOGIC TABLE FOR POSITIVE EDGE TRIGGERED LATCH 48
9. COUNTERCLOCKWISE CODES FOR STEPPER MOTOR 57
10. EXCLUSIVE OR LOGIC TABLE 58
11. DELAY PROGRAM FOR 1/2 INCH LINEAR DISPLACEMENT ALONG X AXIS 62
12. DELAY PROGRAM FOR 1/2 INCH LINEAR DISPLACEMENT ALONG Y AXIS 65
13. DELAY PROGRAM FOR 15 DEGREES ROTARY DISPLACEMENT ABOUT Z AXIS 67
14. DELAY PROGRAM FOR 0.3125 INCH LINEAR DISPLACEMENT ALONG Z AXIS 70
15. FIXED DELAY PROGRAM BETWEEN TASKS 72
16. DELAY PROGRAM FOR REVERSE SWITCHING 74

17. LOCATIONS OF TASK SUBROUTINES (AXIS MOVEMENT SECTION) AND THEIR ASSOCIATED DELAY PROGRAMS 78

18. CODES FOR SAMPLE PROGRAM 89

CHAPTER 1

INTRODUCTION

THIS BOOK CONTAINS NUMEROUS PAGES WITH THE ORIGINAL PRINTING BEING SKEWED DIFFERENTLY FROM THE TOP OF THE PAGE TO THE BOTTOM.

THIS IS AS RECEIVED FROM THE CUSTOMER.

1.1 HISTORICAL BACKGROUND.

The word ROBOT is derived from a Czechoslovakian word 'robota' which implies a worker. In 1932 czech playwright Karel Capek's science-fiction play 'Rossums Universal Robots' was translated to english. Therein, robots were humanoid creations of Rossum and his son, and were meant to serve and obey their masters (1). Such robots having physical resemblances to human beings are called 'android' or 'anthropomorphic' robots.

Even today, this science fiction based perception of a robot continues to dominate the thinking of a lay person. Such robots typically resemble a human being (android types) with a computer for a brain, laser for eyes, instant language interpretation, and super human strength. Additionally such robots could be easily taught to perform multiple tasks.

The industrial robot is far from the above mentioned science fiction version of a robot. The foremost version of the industrial robots were primitive. Many of them were developed in the 1930's and 1940's. They were generally mechanically controlled as opposed to being computer controlled. They had no vision and were relegated to performing a limited number of tasks. By today's definition of robots these mechanical devices cannot be called robots.

During the early 1950's another type of mechanically controlled device called the 'Teleoperator' was developed. A teleoperator was merely a mechanically controlled manipulator designed to allow an operator to perform a task from a distance. Such devices were used in dangerous or hazardous environments.

All the mechanically controlled devices then called robots had several limitations. Firstly, not being computer controlled, they could not be easily taught to perform new tasks. Secondly, the designer could accomodate only a limited number of tasks in a model.

The computer or digital control of a mechanical device was first demonstrated by researchers at Massachusettes Institute of Technology in 1953. In this demonstration, a milling machine was made to move under the direction of numerical input. Additionally, this milling machine had feedback and servo-control. This development was significant in that, it formed the basis for the development of today's industrial robots.

George Devol was one of the pioneers in the development of industrial robots. He used the above concept in the development of the first industrial robot which could perform multiple tasks with ease. He later formed a company called Unimation for the development of Industrial robots.

In the 1960's, George Devol demonstrated the first 'Industrial Robot'-- a device combining the articulated linkage of the teleoperator with the servoed axes of the numerically controlled milling machine. By 1966, seventy such hand-built universal 'Unimate' robots were in operation in the real world.

These industrial robots of the sixties compared to that of fifties, had stepped into a new era of robot technology in the sense that a computer or digital control was made possible. The thrust in the seventies however, was basically to develop the sensory devices and control aspects of the robots. The robots were made to see, feel, touch, and transfer these sensory inputs to a central control. Also they were programmed to make simple decisions based on the sensory inputs. Thus these robots had limited intelligence to react to simple changes in the environment. Considerable contribution to the development of robotic intelligence and robotic sensory devices was made at Stanford University during the early seventies.

The current thrust in the field of robotics is to further develop the primitive intelligence and sensory capabilities of the 70's to a higher degree of perfection. What may be expected of a robot in the future is the ability to take complex decisions under complicated circumstances.

1.2 DEFINITION.

The word robot is used to imply anything from the android types which threaten our existence in science fiction films to the UTD's (Universal Transfer Devices) which are used in pick and place environments.

JIRA (Japan Industrial Robot Association) classifies robots by the method of input information and teaching. The classification (2) consists of six categories.

1) Manual manipulator- not a true robot, but a manipulator controlled by an operator.

2) Fixed sequence robot- a manipulator that repetitively performs successive steps of a given operation according to a predetermined sequence, condition, and position. The set of information can not be changed easily.

3) Variable sequence robot- similar to fixed sequence, except that the set of information can be changed easily.

4) Playback robot-- a manipulator which produces, from it's memory, operations originally executed under human control (the robot is programmed either by lead through or by walk through).

5) NC robot- a manipulator that can perform a given task according to the sequence, conditions, and positions commanded via numerical data (programmed in a plug-in

manner).

6) Intelligent robot- Using sensory perception (visual and/or tactile), this robot can detect changes inthe work environment and, using it's decision making capability can proceed with it's operation accordingly.

Perhaps the most widely accepted definition is that of RIA (Robot Institue of America) stated in the spring 1980 issue of 'Robotics Today'(3).

"A robot is a reprogrammable, multifunctional manipulator designed to move materials, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks."

As per Unimation (4), the definition of a robot is as follows; "A robot should be as general in purpose as the worker it replaces. Therefore, special purpose automation would not qualify because it is built for a single purpose and is rarely adaptable to any other job. Also remotely controlled manipulators would not be classified as robots for they are merely extensions of a human operator's hand. They may be very sophisticated but they are still basically just tools used for human operations."

The definition provided by GE in 'Robotics Today'(4) is a more narrow definition.

"A robot is a commercially available device that can be easily programmed to perform autonomously manipulative job functions. It is considered to fill the gap between special purpose automation and human endeavour."

The android or anthropomorphic robot would still fit into the above definition, but strictly speaking, it would have little use in the industry. The question is then "how do we categorize the industrial robot?" What are the practical features required of an industrial robot?

1) A robot must be flexible, i.e., capable of doing more than a single job, if not in immediate succession, at least following certain adaptive procedures. Absence of such a characteristic would result in the robot being just another special purpose automatic device.

2) A robot must be easy to teach or program. This feature is rather important since time-consuming teaching or programming procedures would result in substantial loss of production time or increased manufacturing costs.

The three existing methods of teaching a robot a task are lead through, walk through, and plug-in.

Lead through : uses a remote teach box to control the robot through the desired positions and store these positions in the robot's memory.

Walk through : is physical manipulation through robot's desired path- the motions are initially recorded in the memory and the successive repetitions result due to the playback of these recorded motions.

Plug-in : means operation of a robot from a pre-recorded program. The program is usually entered into robot memory prior to the operation.

3) Two very desirable characteristics of an industrial robot are intelligence and sensory perception. Considered a luxury today, such capabilities would be required if the robots are to function in the automated factories of tomorrow.

4) The robot must be reliable, a measure of which is often indirectly expressed in downtime percentage. Estimated downtime for a human industrial worker is 9% as compared to 2% to 3% for a present day robot. The industrial robots like any other robot must possess precision, accuracy and repeatability.

1.3 CLASSIFICATION.

Robots may be generally classified as non-servo and servo controlled devices (5).

Non-servo controlled robots have just two points of control in each axis, the initial point and the end point.

These two points can be programmed to lie anywhere within the limits of it's axis movement. Referring to figure 1, X-beg. (begin) and X-end are two such points programmed for a particular program. Thus in non-servo controlled robots, stops and starts can be initiated only at these two points. Further the path between these two points is not controllable. The special characteristics of this category of robots are excellent repeatibility, reliability, ease of operation, and low cost.

The servo controlled robots are different from the non-servo type in the sense that they can stop at any number of points on their path along a single axis.

The servo controlled robots are subdivided into two groups- point to point and continous path. Paraphrasing from Kramer(5), "the point to point path robots are simply programmed to go to certain points, but the movement between these two points are not controlled." In continous path robots, the path between the two points can also be controlled.

The robots can also be distinguished based on the type of movement. There are four types of coordinate systems which are used to define a position within the working envelope of a robot- cartesian, cylindrical, polar, and revolute (figure 2). Each axis constitutes one degree of freedom, and an industrial robot usually has three to five degrees of freedom. Adding a sixth degree of freedom virtually results

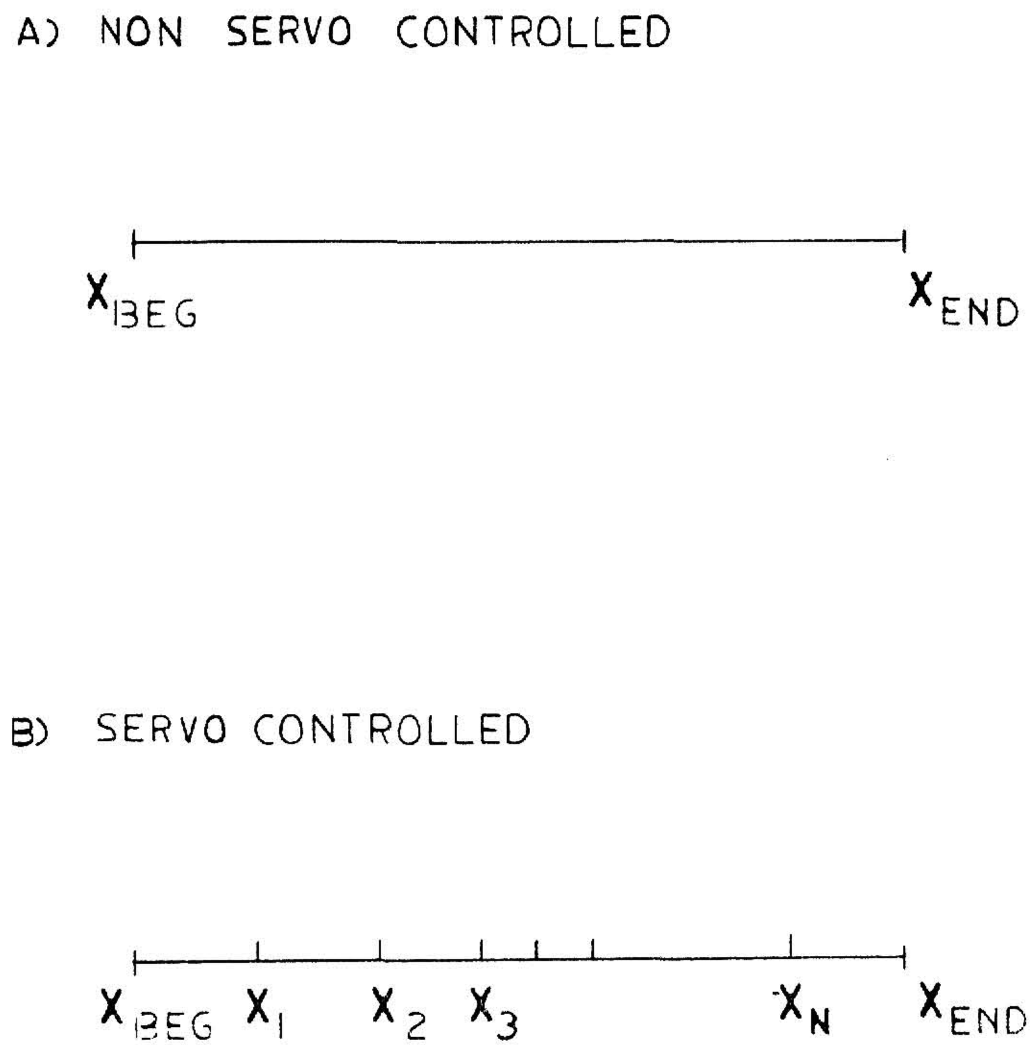
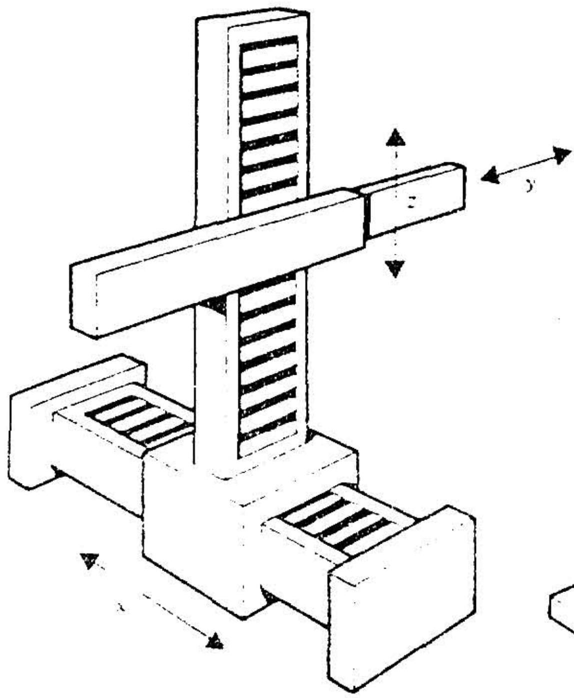
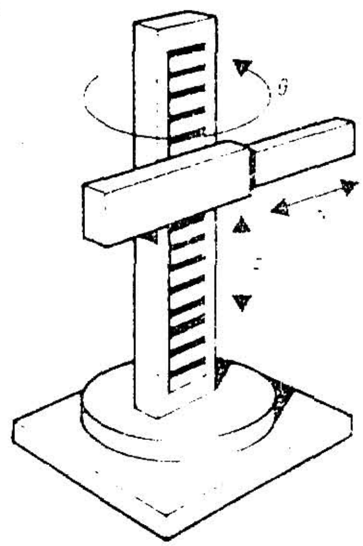


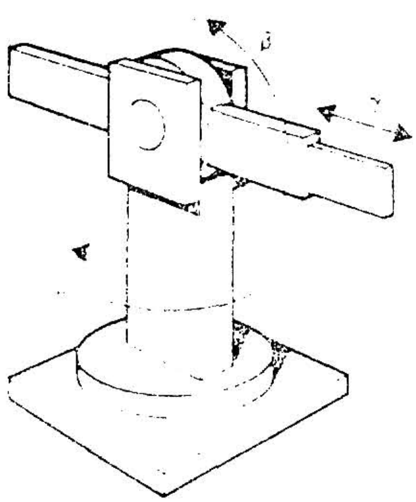
Figure 1. Path of non servo and servo controlled robots.



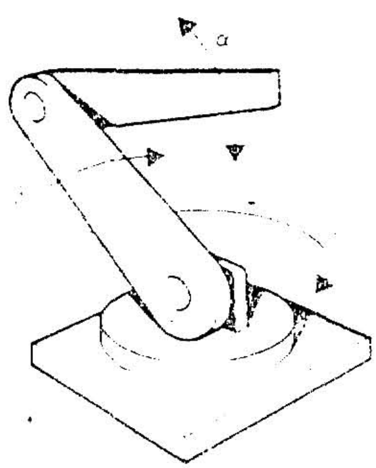
Cartesian Coordinates



Cylindrical Coordinates



Polar Coordinates



Revolute Coordinates

Figure 2. Robot arm configuration.

in repetition of one or the other axis movement and so is rarely incorporated. When it is incorporated it is generally to enlarge the work envelope of the end effector. Maximum manipulation of the wrist is achieved through three types of articulation; the swivel or roll, the bend or pitch, and the yaw (figures 3 and 4).

1.4 MAJOR COMPONENTS.

Robots are available in a wide range of capabilities and configurations. Basically however, an industrial robot consists of four major components;

1) Mechanical unit (Manipulator)- It includes fingers, elbow, wrist and grippers (end effectors), shoulder and an arm which constitute the moving parts together with the linkages. There are generally feedback devices mounted on the links of the manipulator to sense position and communicate this back to the controller.
2) Brain (Controller)- The basic function of the brain is to control the movement of the robot. This includes initiation, direction, as well as monitoring the path of a robot. The controller is also used to receive inputs from various sensors, analyze the inputs, and make decisions on movements. It comprises of either a programmable controller, minicomputer, or a microcomputer.
3) Interfacing circuit- The digital signals sent out by

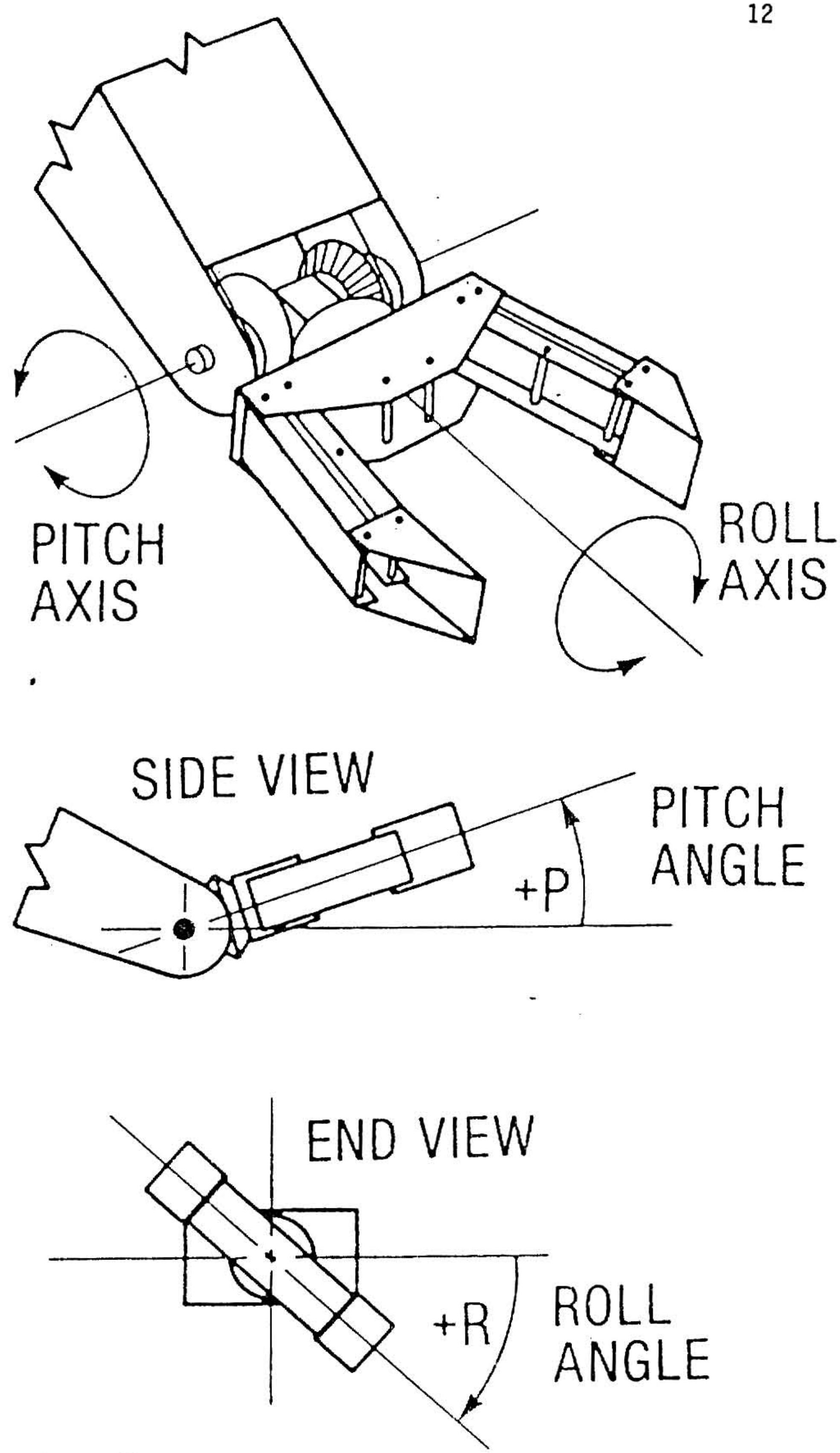


Figure 3. Definition of roll and pitch angle.

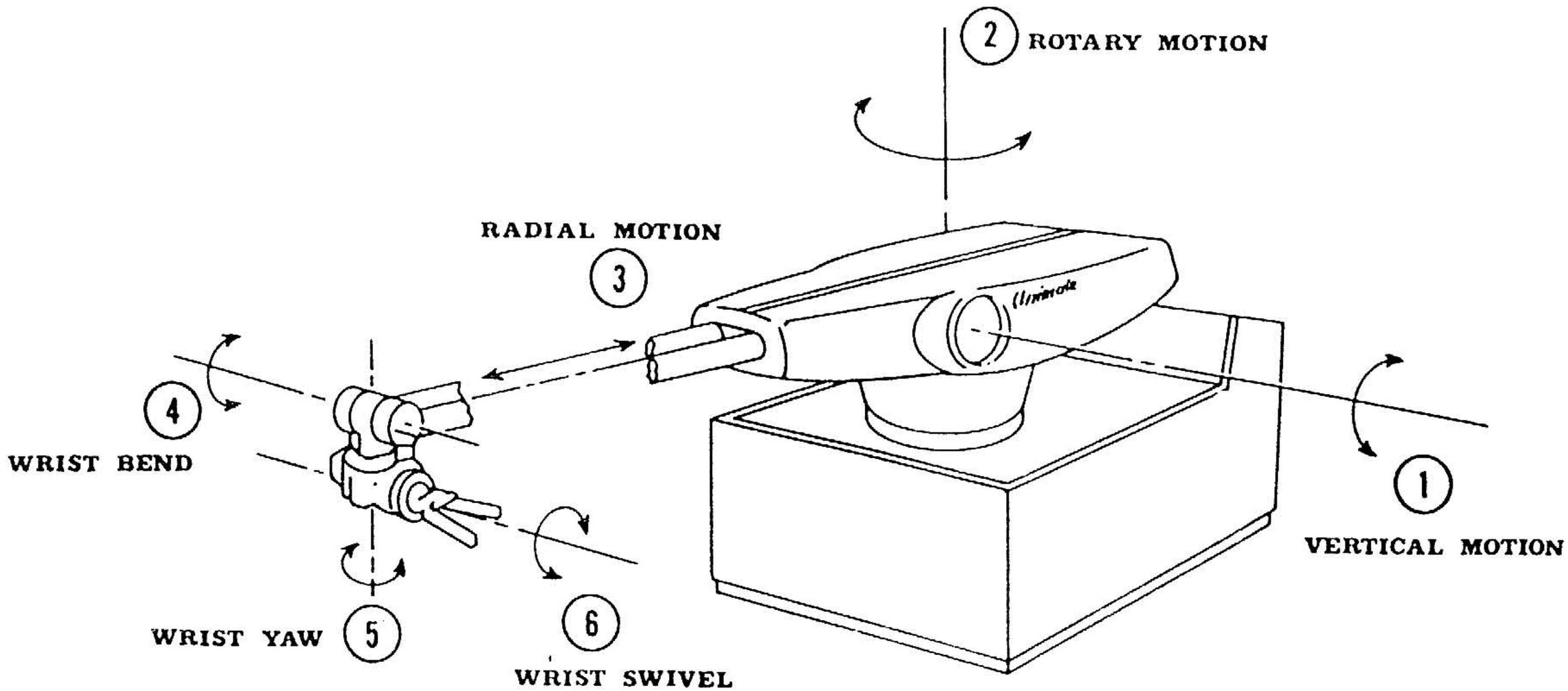


Figure 4. Articulation of robot arm.

the programmable controller or the minicomputer or the microcomputer is translated into analog signals by the interfacing circuit. The purpose of such a gyration is to provide appropriate signals to the analog devices, viz; solenoid valves, DC stepper motors, electromagnets, AC motors, etc.. The interfacing circuits are built to match the control unit and the analog devices.

4) Power supply- It provides the mechanical components with power to move along predetermined paths. The power supply can be pneumatic, hydraulic, or electric (AC/DC) (5). The pneumatic and hydraulic types use a media to power their manipulator through it's motion. In case of pneumatic types such a media is air, and for hydraulically controlled systems the media is petroleum based hydraulic fluid or a special type of flame retardant fluid.

The electrically driven systems are usually more efficient compared to the other two types particularly at heavier loads. Their limitation lies in their inability to maintain high levels of precision for very heavy loads.

1.5 PROBLEM STATEMENT.

To design, develop, fabricate and demonstrate a microcomputer controlled small articulated robot.

1.6 PURPOSE.

There are two basic purposes for this research:

1) To design, develop, and build a laboratory model of a small articulated robot with four degrees of freedom. This robot is to be an integral part of the material handling device in a laboratory model of an automated factory currently being developed in the Department of Industrial Engineering at Kansas State University.

2) To demonstrate programming flexibility and control capability of a microcomputer in unision with an interfacing circuit in controlling the articulated robot. The laboratory model can be utilized as a teaching tool in the field of robotics.

1.7 THE APPROACH- AN OVERVIEW.

The solution approach consists of three distinct phases.

1) Design and fabrication of the mechanical components.

2) Design and fabrication of Interfacing circuit (hardware).

3) Design of software to demonstrate microcontrol of robot.

The initial phase involves converting an existing rotary table with three degrees of freedom to four degrees of freedom. The three degrees of freedom are in the X, Y, and C (rotary about the Z) axis. A vertical up and down motion along the Z axis is to be introduced as the fourth degree of freedom. Additionally, a simple elctromagnet activated gripping mechanism needs to be designed and fabricated. Finally the size of the motors needed to drive the various axes has to be determined. All design has to be performed keeping in mind the limited material and machine resources available within the department.

The second phase consists of the development of a compatible interfacing circuit, which upon the commands of the controller, in this case a MMD-1 microcomputer, can control the motors and the electromagnet.

The final phase is to develop a prototype software to demonstrate the movement of the robot along different axes in sequence and thereby transfer a load from one place to the other.

The interaction and structure of the three phases is pictorially represented in figure 5. As can be seen from the figure, the software in octal code is interpreted by the microcomputer. These interpreted digital signals are analyzed by the microcomputer and necessary actions are output in digital form. This digital output is received by the interfacing circuit where it is converted into analog signals. These analog signals, via relays, control the appropriate motors or the electromagnet.

THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.

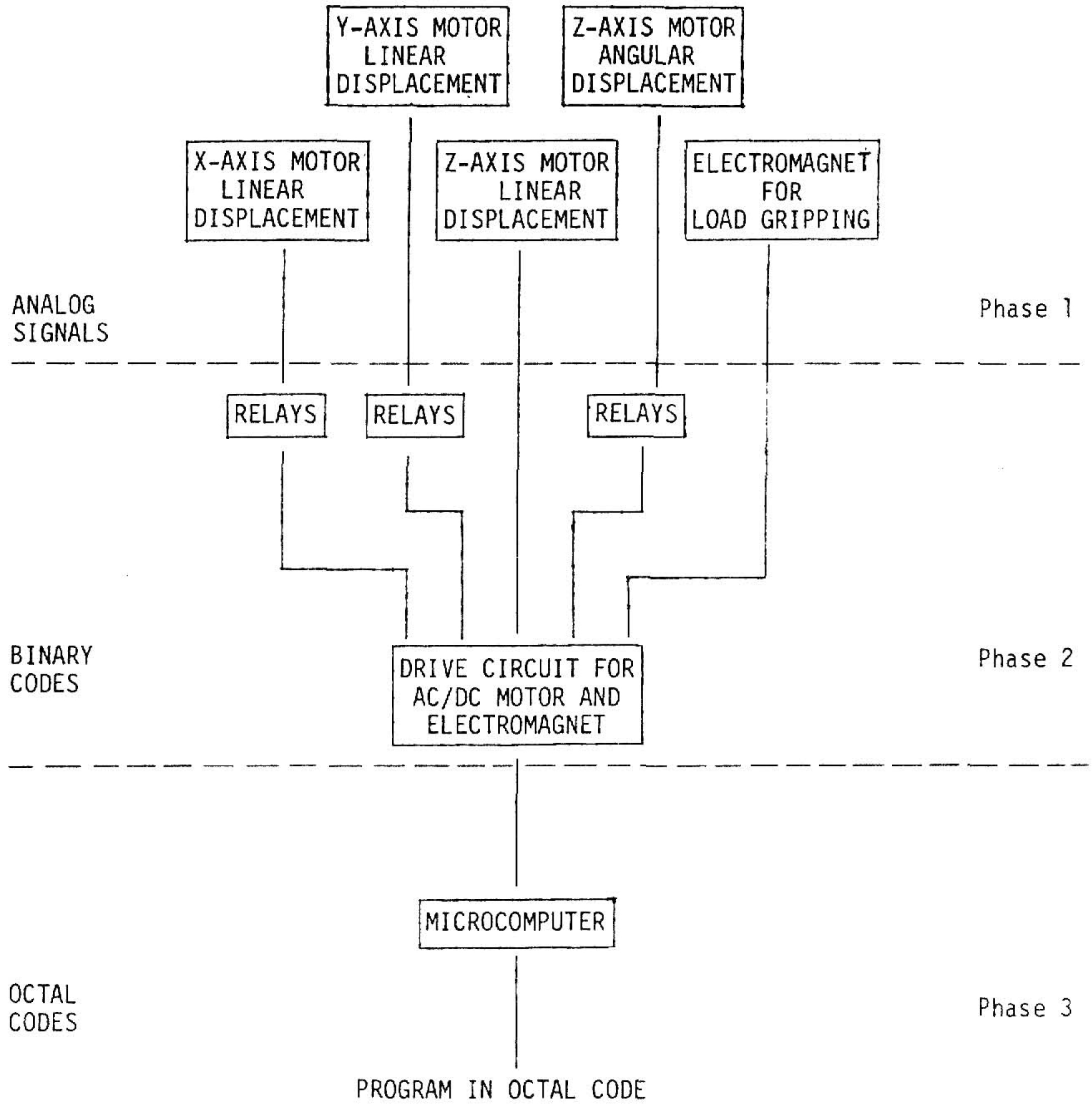


Figure 5. General concept of operation.

1.8 EXISTING EQUIPMENT.

The circular rotary table 'PALM CREN' which serves as the base for the robot arm is manufactured by Chicago Tool and Engineering Company, Illinois. The table is capable of linear displacements along X and Y axes and circular motion about the Z axis. The maximum possible displacement along the X axis is 4.15 inches, 4.25 inches along Y axis, and 360 degrees about the Z axis. The X and Y axes have simple leadscrew mechanism, and the rotation about the Z axis is achieved through a wormgear drive. The leadscrews have 10 Tpi i.e., with each complete rotation they advance the load by 0.1 inch. In a like manner, one rotation of the worm shaft moves the load through 9 degrees.

The microcomputer used is an MMD-1 manufactured by E & L Instruments, Inc., Derby. For a central processing unit the microcomputer is equipped with a 8080A microprocessor manufactured by Intel Corporation. The Random Access Memory (RAM) of this microcomputer has a capacity of 1024 bytes. Additionally, an M1 board provides a storage capacity of another 2048 bytes. The expansion board also provides an audio recorder interface through which programs can be transferred to and fro between the RAM and an ordinary magnetic cassette tape.

CHAPTER 2

MECHANICAL DESIGN

2.1 OVERVIEW OF TASKS.

The mechanical design phase can be grouped into three major tasks.

1) Design of arm, hand, and grippers - includes the gripping mechanism operated by the electromagnet.

2) Design of Z axis vertical drive - involves the design of a motor driven system which will permit vertical movement of the arm along the Z axis.

3) Design for conversion of manual table - involves the determination of appropriate size motors to drive the various axes. It also involves designing the method of coupling the leadscrew ends to the extended motor shafts.

2.2 DESIGN OF ARM HAND AND GRIPPERS.

The arm consists of a 14 inch long brass tube attached to an aluminium block. The electromagnet is fixed on an L plate to one end of the brass tube. The arm and the gripping mechanisms are attached to the other end of the brass tube. Figure 6 shows the general arrangement of the arm. Figure 7 shows the individual components of the arm and the gripping mechanisms. The fingers are normally kept open with the aid of springs. The pull torque of the electromagnet when energized overcomes the spring

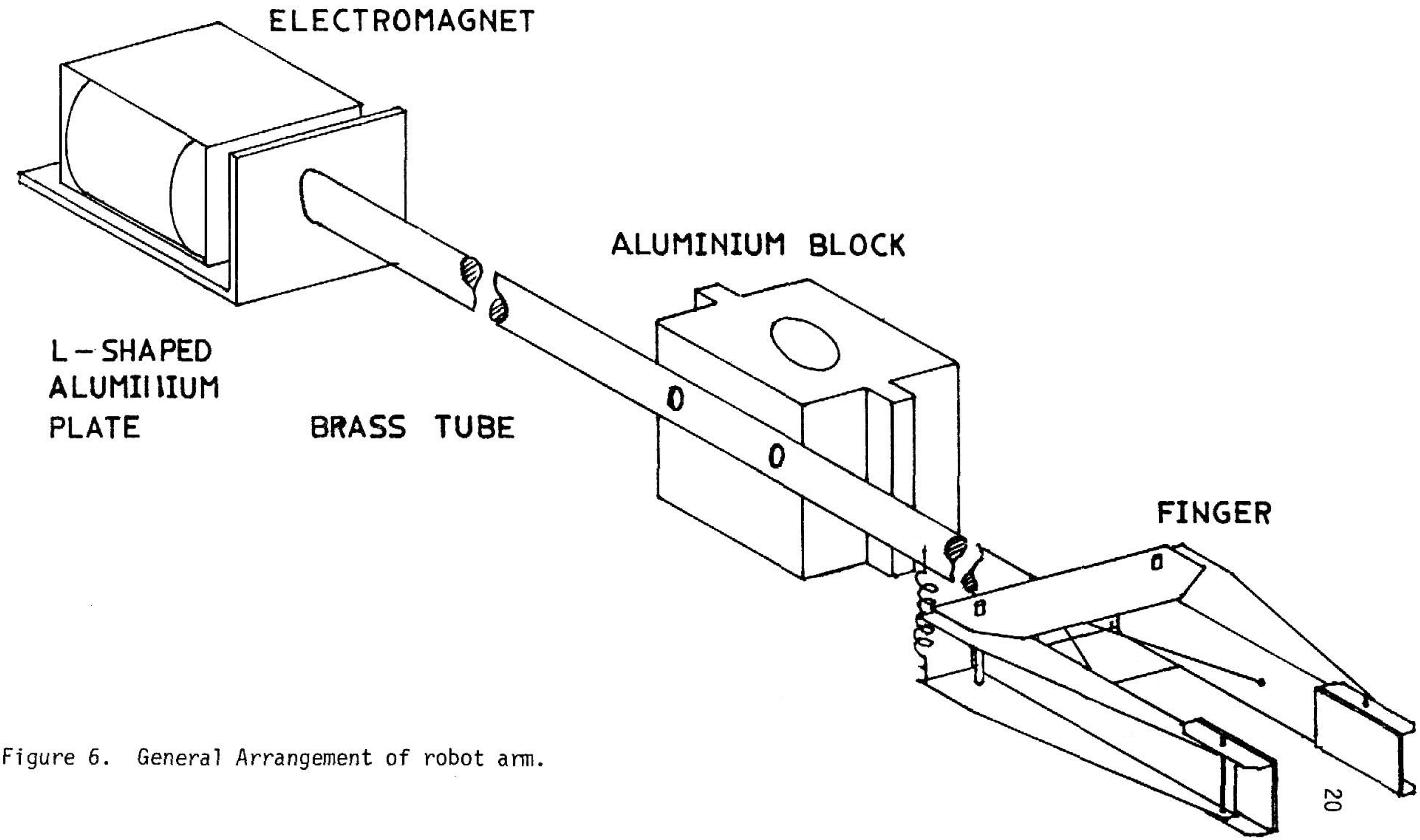


Figure 6. General Arrangement of robot arm.

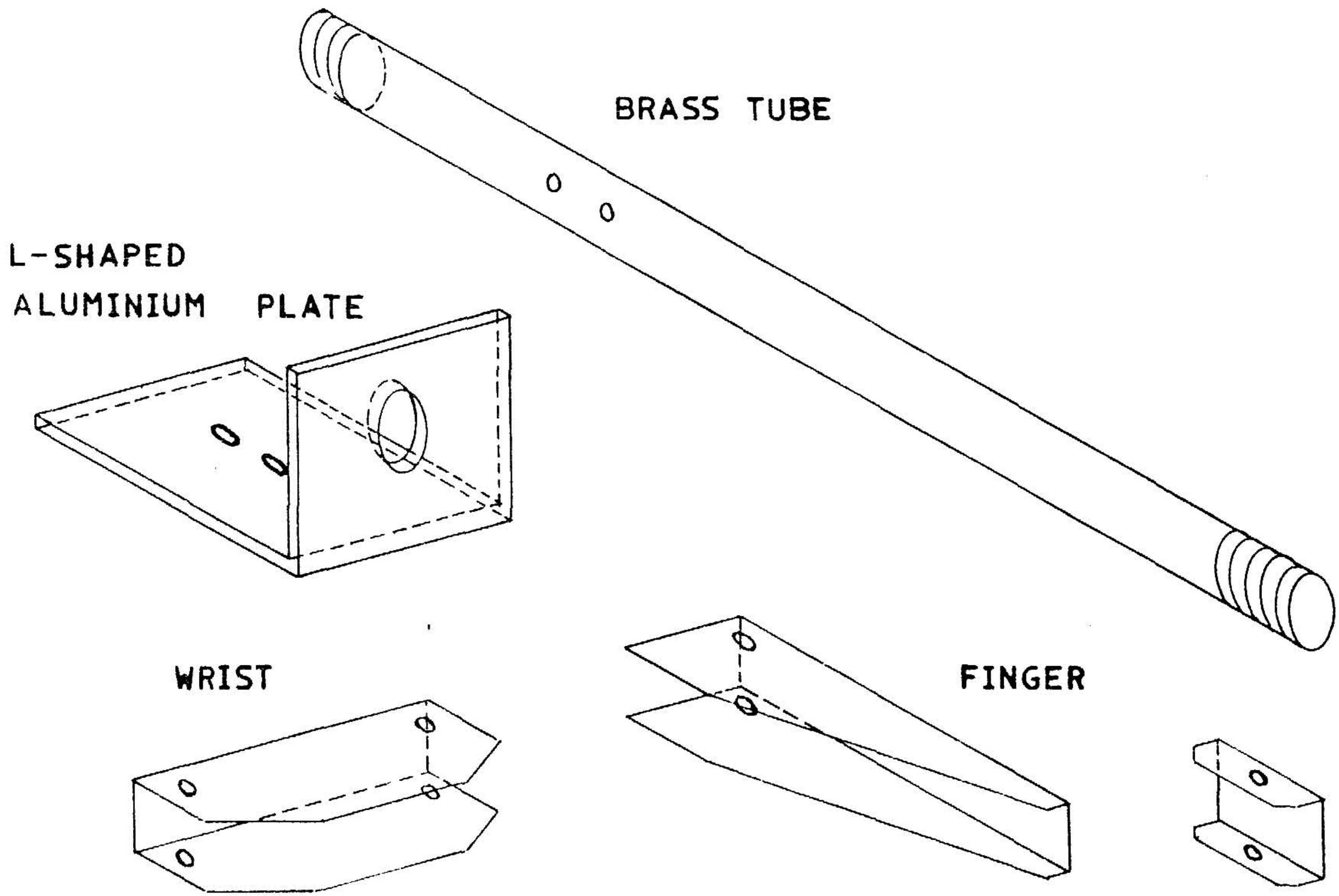


Figure 7. Components of arm and gripping mechanism.

tension and applies the gripping force on the load. The finger end is designed in such a manner that uniform pressure is applied throughout the contact surface. The fingers are made of 0.2 mm aluminium sheet to prevent deformation.

2.3 DESIGN OF Z AXIS VERTICAL DRIVE.

The different components of the Z axis mechanism, (vertical guide rail, top and bottom plate, threaded rod, block, and spur gears) along with the motor arrangement is shown in figure 8. Detailed dimensions of the components are shown in figures 9, 10, and 11.

The threaded rod passes through a brass nut that is press fitted at the center of the block. The bottom end of the threaded rod sits in the center of the ball bearing allowing the rod to rotate freely. The upper portion of the rod passes through the brass bushing that is press fitted at the top plate. The DC motor is mounted vertically on the top plate. The transfer of the rotating torque from the DC motor shaft to the projected end of the threaded rod is accomplished through a spur gear system. Depending on the clockwise or anti-clockwise rotation of the threaded rod, the block with the attached robot arm travels upward or downward along the guides.

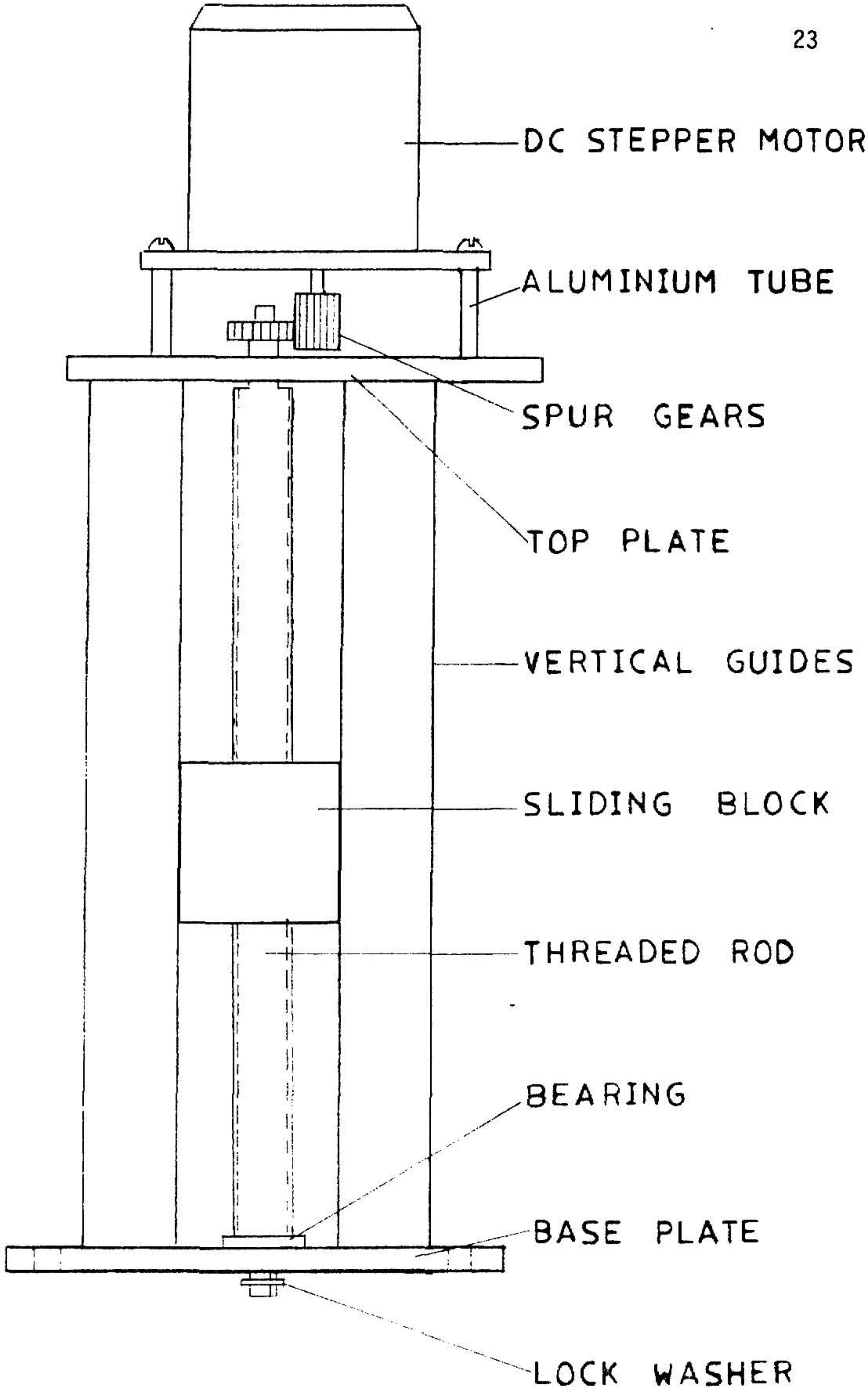


Figure 8. Z axis linear mechanism.

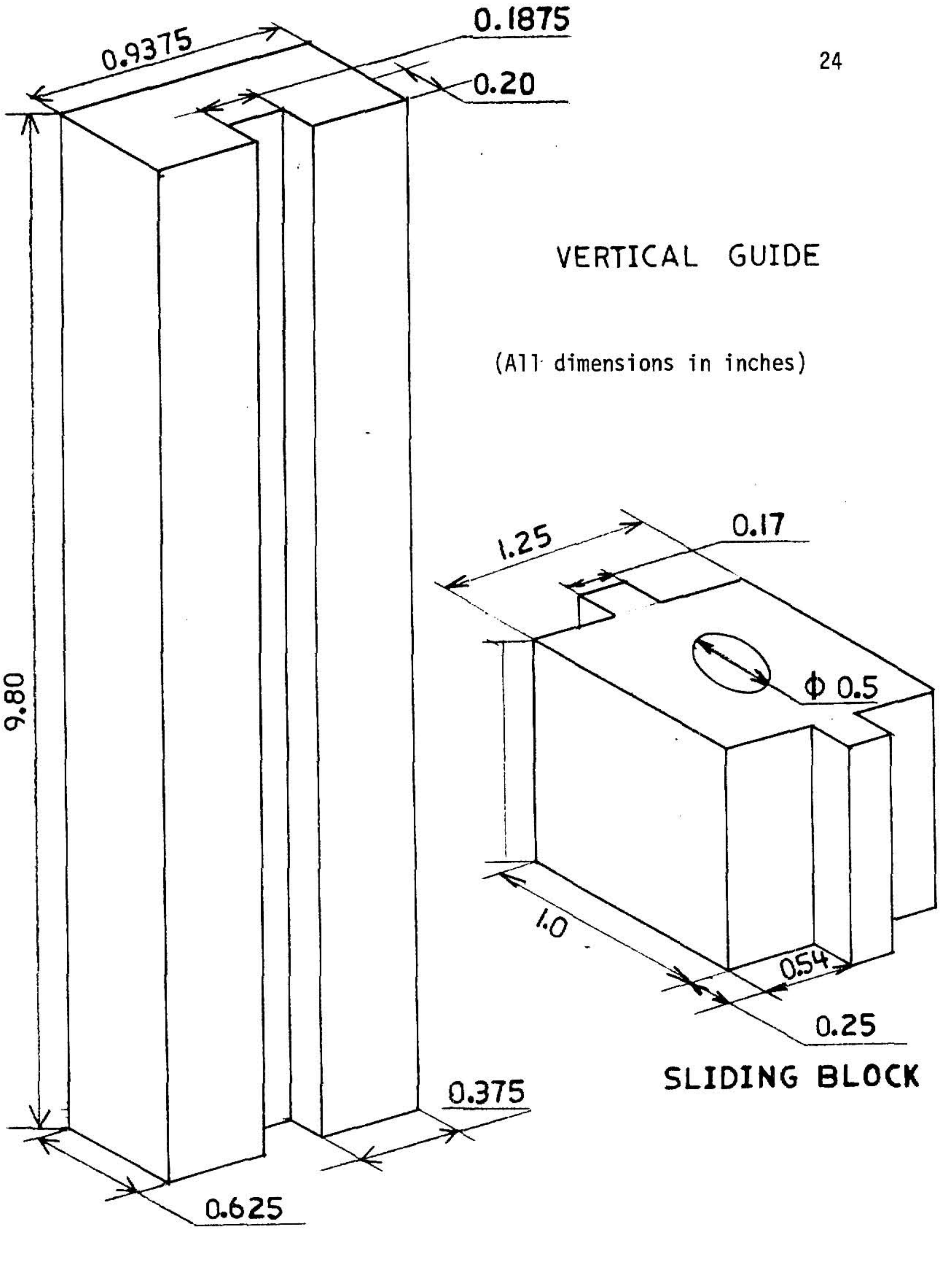
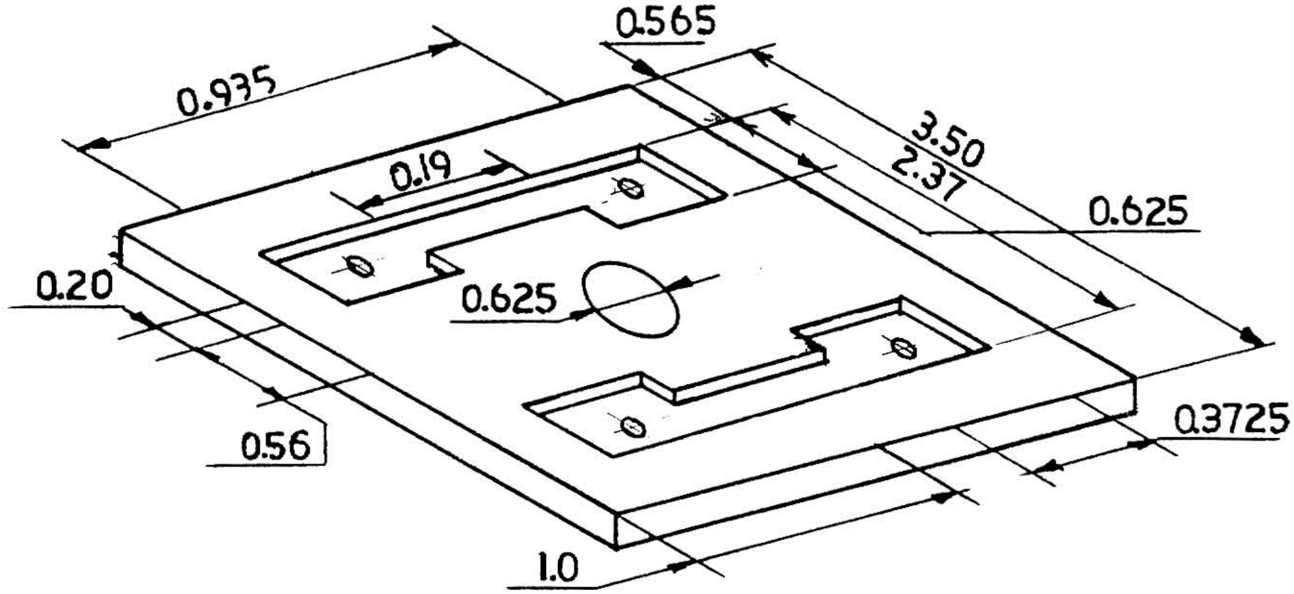
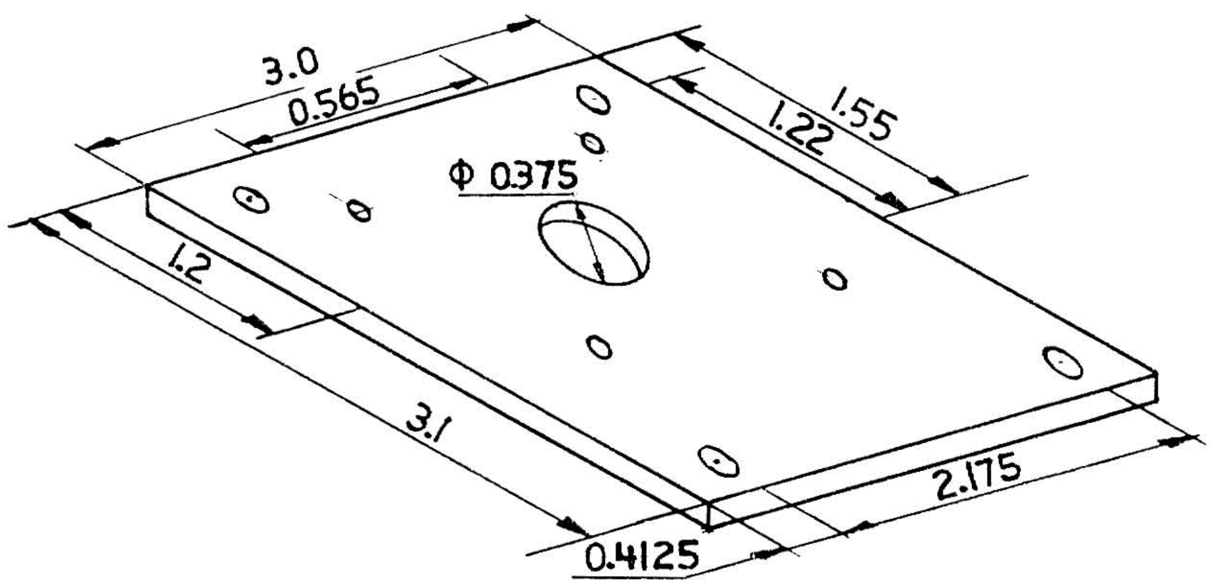


Figure 9. Z axis linear mechanism (Vertical guide and sliding block).



BASE PLATE

(All dimensions in inches)



TOP PLATE

Figure 10. Z axis linear mechanism (Base plate and top plate).

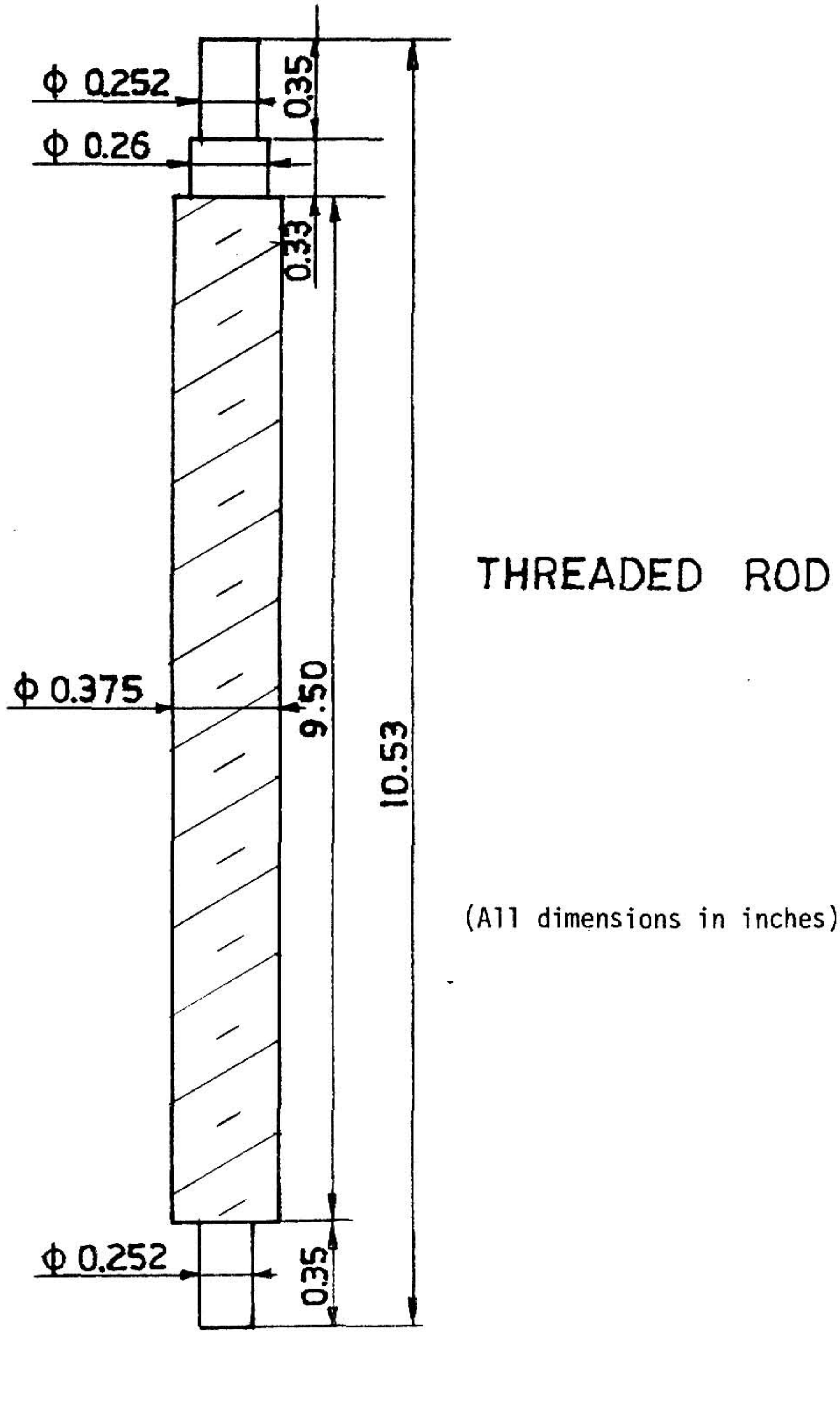


Figure 11. Z axis linear mechanism (Threaded rod).

The dimensions of the spur gears used to transfer rotational torque are as follows (Table 1).

Table 1. Gear sizes for coupling DC motor to the threaded rod.

DIMENSIONS (in inches)	Z AXIS VERTICAL MECHANISM	
	THREADED ROD	MOTOR
Outside Dia	0.563	0.378
Pitch Dia	0.495	0.313
Root Dia	0.425	0.233
Tooth Height	0.069	0.072
Number of Teeth	16	10

2.3.1 POWER REQUIREMENT FOR Z AXIS VERTICAL DRIVE.

A systematic approach for calculating the torque required from the DC motor driving the Z axis leadscrew is described below (6). Figure 12 is a simplified version of the force diagram that depicts the relationship between forces acting along Z axis.

F - Force exerted due to the weight of the load (mN)

P - Force required to push the load up the inclined plane (mN)

N - Normal force (mN)

F1 - Force of friction caused by the steel leadscrew turning through a brass nut (mN)

F2 - Force of friction caused by the aluminium block sliding against aluminium guides (mN)

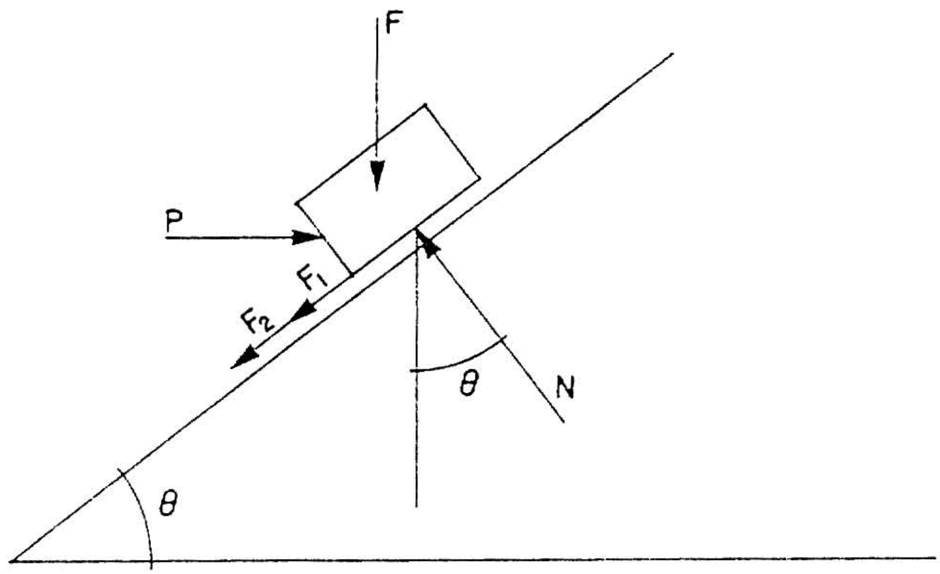


Figure 12. Simplified force diagram of Z Axis.

$\tan\theta = L/2R$.

L - Lead of the screw, m. R - Radius of the leadscrew, m.

F1 - $N\mu 1$.

F2 - $N\mu 2$.

$\mu 1$ - Coefficient of friction between steel and brass, assume $\mu 1$ = 0.5.

$\mu 2$ - Coefficient of friction between aluminium and aluminium, assume $\mu 2$ - 0.3.

θ - Lead angle , $3^{\circ} 24'$ or 3.404669° .

To find F, the following relationship was used. The mass of the total arm assembly is 453.6 grams.

F = MA.

F = 453.6 9.8.

F = 4445 mN.

Summing the forces in the Y direction gives the following.

F = N cos θ - F - 0.5 N sin θ - 0.3 N sin θ.

0 = 0.988 N - F - 0.02969 N - 0.0178 N.

1.0457 N = 4445

N = 4250 mN.

Summing the forces in the X direction gives the following.

F = P - N (sin θ + 0.5 cos θ + 0.3 cos θ) = 0.

P = 0.8579 N.

P = 3646 mN.

Since P acts at a distance R from the centerline of the threaded rod, the following relationship gives the estimate of the torque required to raise the load.

T = PR.

T - Torque, mNm.

R - Radius of the threaded rod , m.

T = 3646 × 0.00476.

T = 17.355 mNm.

Assume a 2% extra torque required due to the frictional losses at the gears.

T = 17.702 mNm.

2.4 ROTARY TABLE CONVERSION.

The task here is to equip the manually operated drive mechanisms of the rotary table with AC motors of appropriate size and capacity. Howsoever, prior to choosing a motor, the power requirement of each individual axis has to be determined. In this case, since the AC motors are already available, the following calculations (7) are carried out solely to ensure that the chosen motors are of sufficient capacity.

2.4.1 LEADSCREW POWER REQUIREMENT.

The AC motors on the three different axes - X linear, Y linear, and Z rotary, are all of 1/100 HP with a starting torque of 3502 mNm (milli newton meter) and a running torque of 2937 mNm.

Since the X axis mechanism is at the lowest level of all the three axis system, it is enough to calculate the power requirements at this axis. By lowest level here it is meant, that the X axis drive system is physically located underneath the other two axes. Thus it bears the loads of the other two axes and is consequently the axis with the heaviest load. This results in the X axis possessing the maximum frictional torque and therefore the largest torque needed to accelerate an inertial load.

The following equations were used in the calculations.

Equation 1 : $T = T_L + T_F$.

where ;

T - Total torque load, mNm (milli newton meter).

T_L - Torque required to accelerate an inertial load, mNm.

T_F - Frictional torque, mNm.

Equation 2 : $T_L = J_T \cdot a$

where ;

J_T - Total moment of inertia, $Kg.m^2$ (Kilogram meter^2).

a - acceleration, m/sec^2 (meter per second square).

Equation 3 : $J_T = J_R + J_S + J_{RL}$.

where ;

J_R - Moment of inertia of the rotor of the motor, $Kg.m^2$.

J_S - Moment of inertia of the steel leadscrew, $Kg.m^2$.

J_{RL} - Reflected moment of inertia of the load, $Kg.m^2$.

Equation 4 : $J_S = D^4 L \rho_S \times \pi/32$.

where ;

D - Diameter of the leadscrew in meters.

L - Lead of the screw in meters.

ρ_S - Density of steel in Kg/m^3, 7870 Kg/m^3.

Simplifying the above equation ;

$J_S = D^4 L \times 7.726 \times 10^2$.

The expression for the reflected inertia of a load when connected to a linear- screw thread is;

Equation 5 : $J_{RL} = M \times (L/2\pi)^2$.

where ;

M - Mass of the load, Kg.

L - Lead of the screw, m.

The mass on the X axis is 21 Kg including an end load of 1 Kg on the gripper. The lead of the screw is 0.00254 m.

Therefore ; $J_{RL} = 21 (0.00254/2\pi)^2$

$J_{RL} = 3.43 \times 10^{-6}$ $Kg.m^2$.

The rotor moment of inertia listed in the engineering specifications for the motor under consideration is 8.5×10^{-6} $Kg.m^2$.

The diameter of the leadscrew is 0.013335 m.

So, $J = (0.013335)^4 \times (0.00254) \times 772.6$

$J = 6.205 \times 10^{-6}$ Kg.m^2.

Using the equation for J_T, the total moment of inertia is;

$J_T = (8.5 + 6.2 + 3.43) \times 10^{-6}$ Kg.m^2.

$J_T = 18.13 \times 10^{-6}$ Kg.m^2.

The gearing ratio is 26 : 32 for motor to leadscrew and the motor is of 10 RPM. The total mass sliding over the X axis mechanism has a velocity of (26/32)x10 turns per minute or 0.135416 turns per second.

The lead of the screw is 0.00254 m, and so the velocity is 3.4396×10^{-4} m/sec. The acceleration will therefore be 1.72×10^{-4} m/sec^2.

Multiplying the acceleration by the total moment of inertia gives the torque required to accelerate the load from 0 m/sec to 3.4396×10^{-4} m/sec.

Then torque (T) is;

$T = 18.13 \times 10^{-6} \times 1.72 \times 10^{-4}$.

$T = 30.57 \times 10^{-6}$ mNm.

The calculations above do not take friction into account. The frictional torque has to be determined experimentally by a torque-wrench. Using this method the frictional torque was found to be 91.78 mNm.

Finally the total torque required to accelerate a load of 1 Kg is;

$T = 91.78 + 30.57 \times 10^{-6}$.

T = 92 mNm (approx.).

A 2% extra torque is required due to frictional losses at the gears. Therefore, T is approximately 93.62 mNm.

Since slow speed motors are used, the acceleration (measured in radian / sec^2) for the rotary movement about Z axis will be negligible. The frictional surface is also found to be about the same as in X axis. Since the Z axis mechanism carries much less weight than the X axis mechanism, the motor selected for X axis will suffice for the Z axis.

2.4.2 MOTOR ASSEMBLY FOR GEAR DRIVES.

All the three AC motors (for the two linear and one rotary axis mechanism) are coupled to the leadscrew (worm shaft for Z axis rotary mechanism) ends through a pair of spur gears. The motors are mounted horizontally by an arrangement comprising of a steel face plate, four aluminium tubings of equal length, and four long screws (figure 13). The threaded ends of the screws are tightened at the tapped holes of the motor gear-train

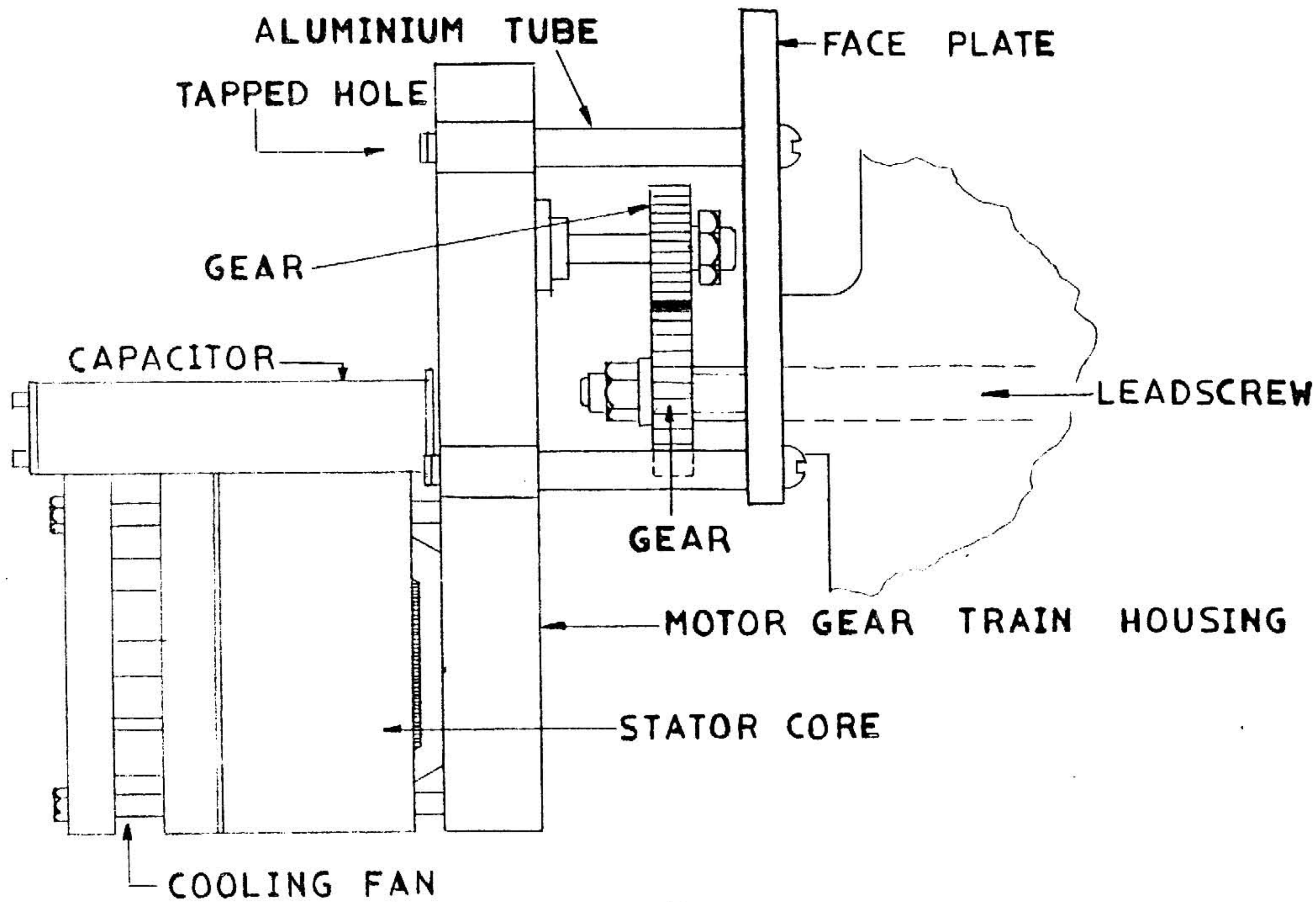


Figure 13. Motor - Leadscrew coupling for AC motors.

housing.

The dimensions of the gears for coupling the AC motors to the different drive mechanisms are presented in Table 2.

Table 2. Gear sizes for coupling AC motors to various axis mechanism.

DIMENSIONS	X AXIS LINEAR		Y AXIS LINEAR		Z AXIS ROTARY	
(inches)	MOTOR	LEADSCREW	MOTOR	LEADSCREW	MOTOR	WORMSHAFT
D_E	0.87	1.06	0.90	0.90	0.87	1.06
D_P	0.80	0.80	0.83	0.83	0.80	0.80
D_R	0.73	0.92	0.72	0.72	0.73	0.91
h_t	0.07	0.07	0.09	0.09	0.07	0.07
N	26	32	20	20	26	32

where;

D_E -- Outside diameter of the gear.

D_P -- Pitch diameter of the gear.

D_R -- Root diameter of the gear.

h_t -- Height of the tooth.

N -- Number of teeth in the gear.

Following (Table 3) are the dimensions of the worm and the gear that produce the rotary motion about the Z axis.

Table 3. Dimensions of the wormgear mechanism for angular displacement about Z axis (in inches).

DIMENSIONS	WORM	GEAR
Outside Dia	1.195	3.542
Pitch Dia	1.025	3.297
Root Dia	0.825	3.215
Tooth height	0.185	0.1625
	4 Threads per inch.	40 teeth

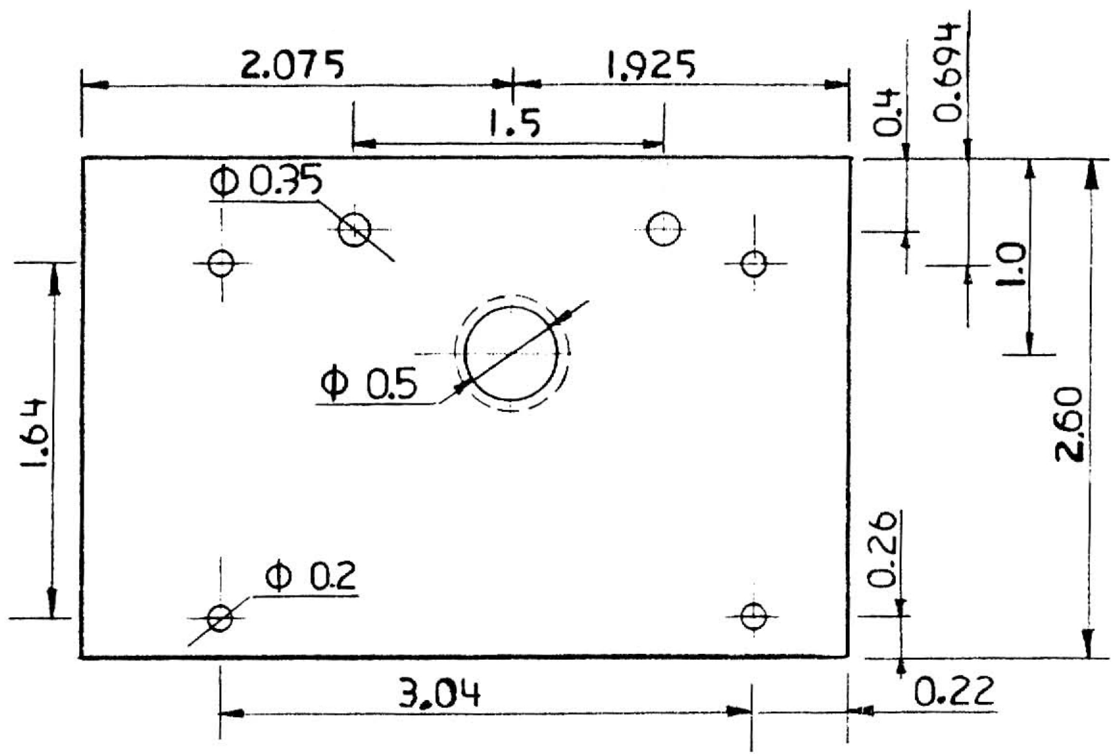
The dimensions of the steel face plates is shown in figure 14. Figure 15 shows the completed rotary table together with the AC motors assembled to it.

2.4.3 MACHINING OF MECHANICAL COMPONENTS.

The machining task of the mechanical components is accomplished in two successive phases.

1) Machining of the Z axis (vertical) drive and the gripper mechanism components.
2) Machining of components for coupling AC motors for the X axis linear, Y axis linear, and the Z axis rotary mechanism.

To ensure proper meshing of the gears, it is absolutely necessary to accurately machine the components and observe strict norms of tolerances.



(All dimensions in inches)

Figure 14. Dimensions of face plate.

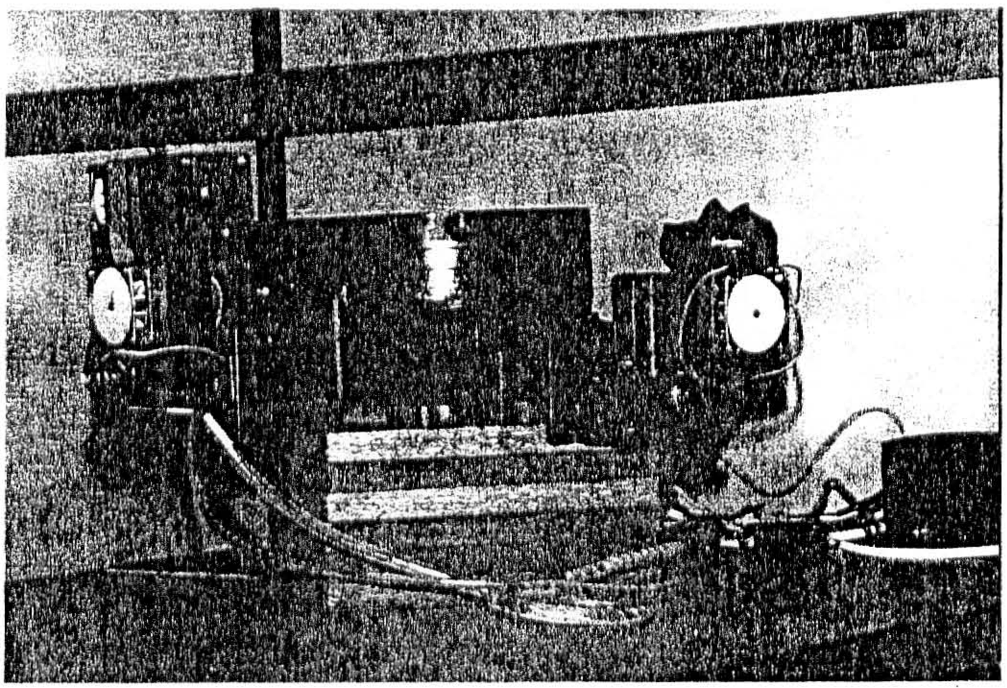


Figure 15. Completed rotary table with AC motors.

All the mechanical components have been machined in the Industrial engineering department workshop. A major portion of the machining has been done in the three axis Numerical Control (NC) vertical milling machine.

2.5 MOTOR CHARACTERISTICS.

Table 4 gives the characteristics of the AC motors (8), the DC motor, and the electromagnet chosen for the robot arm.

Table 4. Motor and Electromagnet characteristics.

AC MOTOR.

TYPE	MAKE	HP	START TORQUE	RUN TORQUE
AC	DAYTON	1/100	3502 mNm	2937 mNm

DC MOTOR.

TYPE	MAKE	STEP/REV	STEP ANGLE	PULL TORQUE	MAX STEPS/MIN.
DC	PHILLIPS	96	3 45	63.54 mNm	240

ELECTROMAGNET.

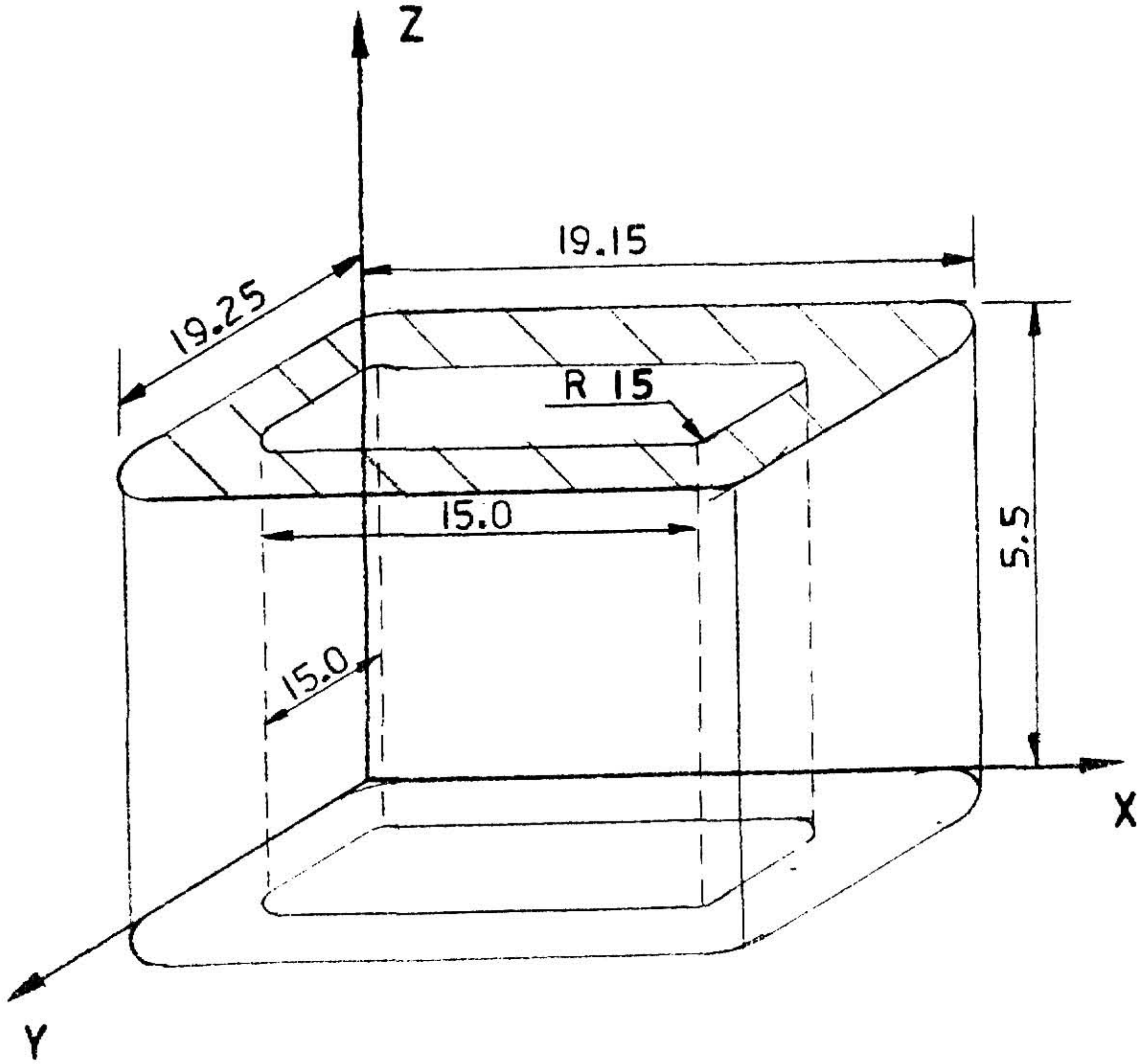
TYPE	MAKE	MIN. STROKE	MAX. STROKE
DC		24 oz. at 1/8 inch	3 oz. at 3/4 inch

The AC motors have a permanent split capacitor, 3 wire/ 2 pole/ 4 coil construction. The motors are reversible and operate at 115v, 60/50 Hz supply. These motors have self aligning, porous bronze sleeve-type bearings with oil reservoirs for all angle operation. A shrouded cooling permits continous operation. The DC motor operates on 5V DC, whereas the electromagnet operates on 6V

DC.

2.6 TRAVEL LIMITS AND WORK ENVELOPE

The rotary table allows a maximum displacement of 4.15 inches along X axis and 4.25 inches along Y axis. Together with the arm (15 inches) the maximum possible displacement of the table along the X axis and Y axis is 19.15 and 19.25 inches respectively. The robot arm is capable of vertical travel between a height of 5.5 inches and, a rotary displacement of 360 degrees as shown in figure 16.



(All dimensions in inches)

Figure 16. Working envelope of robot arm.

CHAPTER 3

HARDWARE DESIGN

3.1 AC MOTORS DRIVE CIRCUIT.

As mentioned earlier, three of the drive mechanisms are AC powered. They are the X axis linear, the Y axis linear, and the Z axis rotary mechanisms. The microcomputer and the drive circuits - 'the digital world' operates on +5V DC , whereas the AC motors or 'the real world' require a power supply of 110V, 60Hz, AC.

Fortunately, the OPTO-FILM model OFA-1202 solid state relays are there to establish the vital link between the digital and the real world. The relays consist of LED's (Light Emmitting Diodes) on the +5V DC side and photo-transistors on the AC side. The function of the photo-transistor is to respond to the DC signals. Of the two DC circuit connections- pin 1 and pin 16, the second is connected to +5V. The signal at the other pin (pin 1) is varied between +5V and 0V. When 5V prevails at both the pins of the DC circuit, there is no flow of current. A 0V at pin 1 allows flow of current and lights up a LED inside the relay. The photo transistor on the AC side reacts to the light and closes the AC circuit to enable flow of current.

The AC motors have three input leads; one is ground and the other two are for clockwise and anticlockwise rotation of the rotor. The various wire colors and the inference with respect to movement types is given below in Table 5.

Table 5. Wire connections for AC motors.

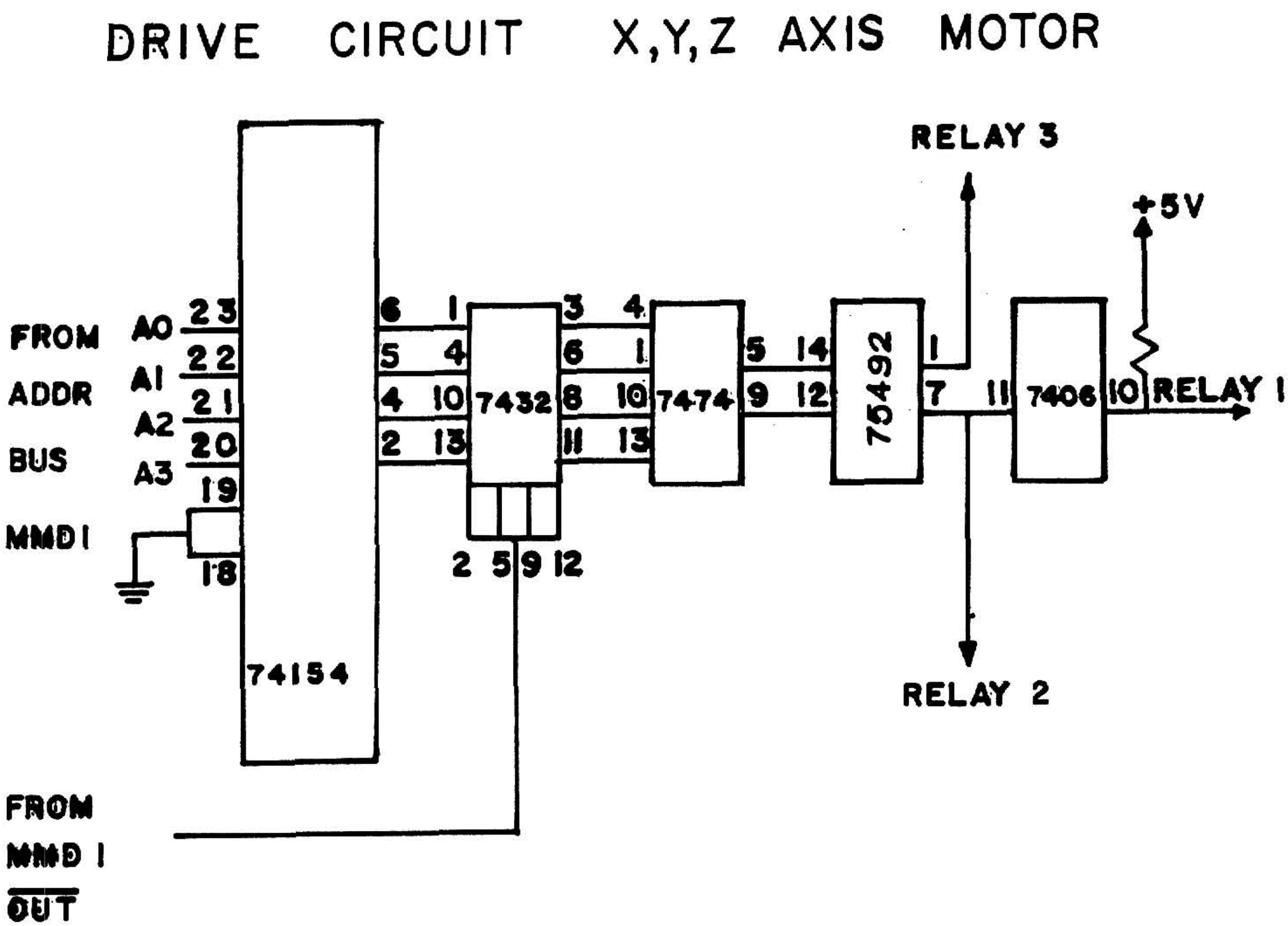
MOVEMENT TYPE	X AXIS MOTOR	Y AXIS MOTOR	Z AXIS MOTOR
FORWARD/CW	YELLOW	BLACK (thin)	YELLOW
REVERSE/CCW	GREEN	GREEN	GREEN
NEUTRAL	BLACK	BLACK	BLACK

Given the above, the task now is to ensure a 110V AC supply to either of the live wires and ground the neutral wire. The supply then has to be maintained for a particular length of time until the desired displacement along that axis has been achieved, and then turned off. This ON/OFF function of the motor is performed by the relays, which actually act as switch.

There are three relays for each AC motor. At least two of these must be energised simultaneously to make the motor run either in clockwise or anticlockwise direction. Figures 17 and 18 show a portion of the drive circuit for the X axis motor.

It can be seen from the figure that relays R3 and R2 must be turned on together to initiate a forward displacement of the rotary table. On the other hand, turning on relays R3 and R1 would enable displacement along opposite direction. In either case relay R3 must be energized to ensure closing of the AC circuit.

The drive circuit for the AC motors (9) starts at HD74154P 4 to 16 line decoder (figure 17). The logic table (10) for the decoder is presented in Table 6.



PIN HOOK UP

TYPE	GROUND	5V	MISC PIN CONNETIONS
74154	12	24	18 19 GROUND
7432	7	14	2,5,9, & 12- $\overline{\text{OUT}}$ -MMD I
7474	7	14	2,12 GROUND
75492	4	11	1-2,3-14,7-9-13,8-10-12
7406	7	14	2- +5V THRU RESISTOR 390Ω

Figure 17. IC circuit for X (Y and Z) axis AC motor.

DRIVE CIRCUIT X-AXIS
AC MOTOR RELAYS

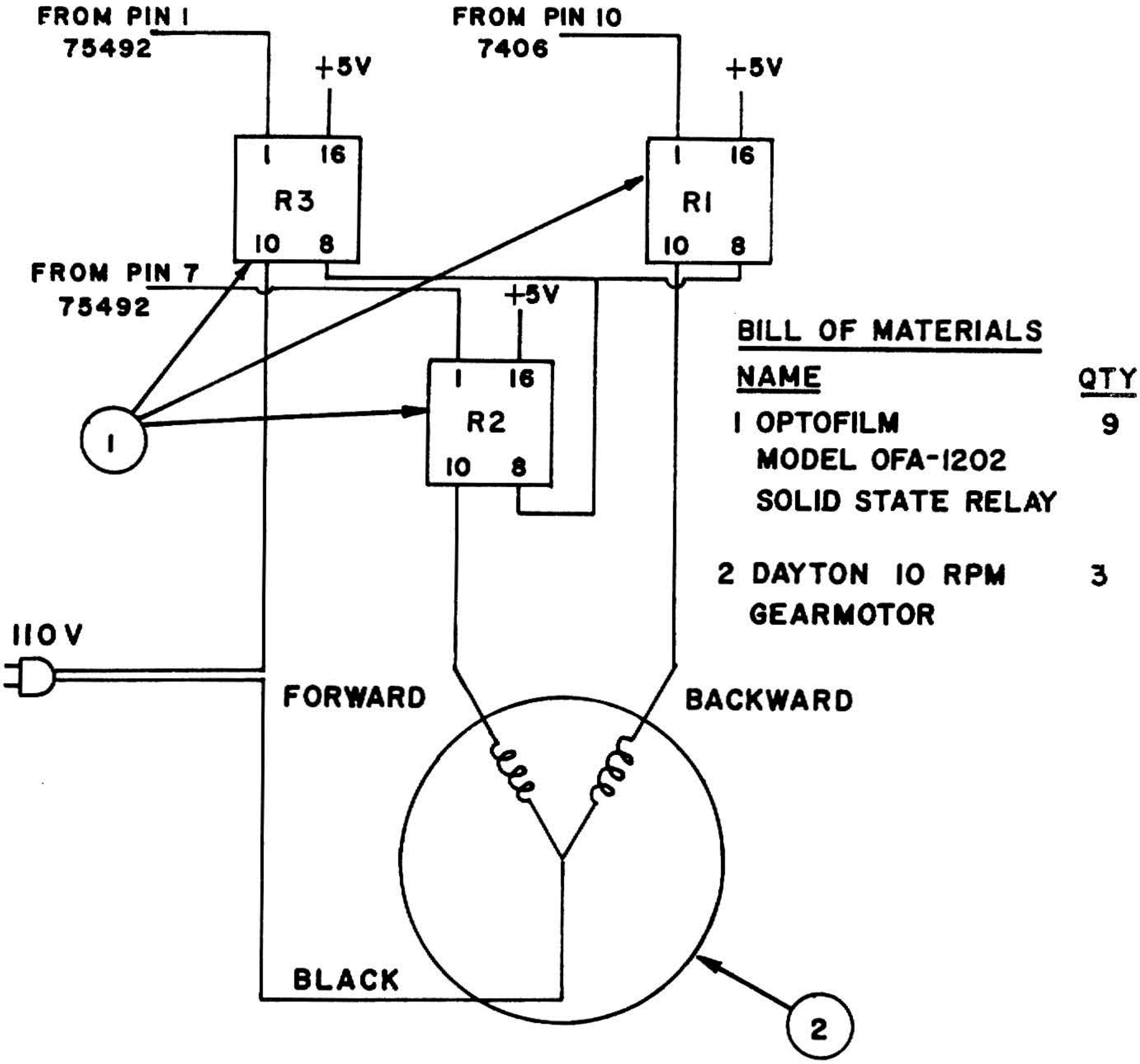


Figure 18. Relay and motor circuit for X (Y and Z) axis AC motor.

Table 6. logic table for 4 to 16 line decoder chip.

INPUT				OUTPUT (CHANNELS).															
D	C	B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

The pins 18 and 19 of the decoder must be connected to ground to enable the chip. Pins 20, 21, 22, 23 -- the input pins of decoder are connected to the outputs A3, A2, A1, and A0 of the address bus in the microcomputer socket.

The following discussions are restricted to the X axis motor. Identical parallel components are present for the Y axis and Z axis (rotary). The SN 7432 quad 2-input package follows the HD74154P chip in the drive circuit for the AC motors. The truth table (10) for the OR-gate is given in table 7.

Table 7. Logic table for OR-gate.

INPUT		OUTPUT
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

Perhaps the most important element of the circuit is the SN 7474 N D-type positive-edge-triggered flip-flop (latches) with PRESET and CLEAR. The truth table (10) for the flip-flop is given in the next table (Table 8).

Table 8. Logic table for positive- edge-triggered latch.

INPUTS				OUTPUTS	
PRESET	CLEAR	CLOCK	DATA	Q	$\overline{Q}$
0	1	x	x	1	0
1	0	x	x	0	1
0	0	x	x	*	*
1	1	$\vert$	1	1	0
1	1	$\vert$	0	0	1
1	1	0	x	Q0	Q0

In the drive circuit, the properties of the asynchronous inputs, namely PRESET and CLEAR, of the latch alone are utilised.

The SN75492AN chip is a Hex 'LED digit drivers' and serves the purpose of an inverter. The DM 7406N inverter chip has an identical function in the circuit. The 390 ohms resistors at the output pins of the 7406 chip serve as pull-up resistors. Thus the signal is amplified before it enters the solid state relays.

The following highlight of the chip behaviour (10) is provided to assist in understanding the technique of converting digital output to analog signals.

a) Data is presented in binary form at the input pins 20, 21, 22, and 23 of the 4 to 16 line decoder. The channel, whose number is decimal equivalent to the binary input at these pins, takes on logic 0. The internal channels are connected to the external pins of the decoder. Therefore an input of 0101 (decimal 5), i.e. logic 1 at pins 21 and 23 and logic 0 at pins 20 and 22 will result in a logic 0 at channel number 5 or pin 6 of the decoder.

b) Only the asynchronous inputs PRESET and CLEAR of the latch (7474 chip) inputs are utilised in the circuit. A logic 0 at PRESET overrides all other inputs and establishes logic 1 at the output Q. On the other hand, a logic 0 applied to CLEAR results in a logic 0 at the output pin.

c) The function of the 7406 chip is to present opposite signals to relays R1 and R2. This is quite essential to the circuit since turning on both the forward/clockwise and reverse/anticlockwise relays will cause the motor to remain stationary. The 7406 chip ensures that turning on one of the relays automatically switches off the other.

Consider the following sequence of tasks to help illustrate the chip workings of the circuit.

a) Turn on the forward relay and effect forward movement along X axis for 3 minutes.

b) Turn forward relay off and take a break for 15 seconds.

c) Turn on reverse relay and bring the rotary table back to its original position.

The following program can successfully perform the above sequence of tasks.

HI	LO	INSTR.BYTE	MNEM.CODE	COMMENTS
003	000	323	out *	
	001	005	<005>	ON/OFF relay (R3) is off.
	002	000	nop	5.33 microseconds.
	003	323	out *	
	004	003	<003>	forward relay (R1) is on.
	005	000	nop	5.33 microseconds.
	006	323	out *	
	007	004	<004>	ON/OFF relay (R3) is on.
	010	315	call	delay subroutine (3 min.).
	011	250	lo	
	012	003	hi	
	013	323	out *	
	014	005	<005>	ON/OFF relay (R3) is off.

015	000	nop	5.33 microseconds.
016	315	call	delay subroutine (15 sec.).
017	200	lo	
020	003	hi	
021	323	out *	
022	001	<001>	reverse relay (R1) is on.
023	000	nop	5.33 microseconds.
024	323	out *	
025	004	<004>	ON/OFF relay (R3) is on.
026	315	call	delay subroutine (3 min.).
027	250	lo	
030	003	hi	
031	166	halt	stop execution.

When the device code 005 appears at the address bus of the MMD-1, the same digital information is simultaneously applied to the input pins of the 4-to-16 line decoder. Therefore the fifth channel or pin 6 is at logic 0. At the next instant, a negative pulse (control signal OUT) is generated at the microcomputer bus socket for 1.33 microseconds (11). This results in a logic 0 at pin 1 (CLEAR 1) and logic 1 at pin 4 (PRESET 1), pin 10 and pin 13 of the latch (7474). The signal at pin 5 which is 0 (OFF) is then inverted at 75492 chip and +5V is presented at the pin 1 of the opto-relay (R3, figure 18) which ensures that the motor is off.

In the next step, logic 0 is applied to pin 1 of the forward X axis relay R2 by selecting the device code 003. Immediately following, the relay R3 is turned on with an instruction byte of 004 at the address 003 hi and 007 lo.

Once the forward movement has been initiated the main program shifts to a subroutine at 003 hi and 250 lo which is a delay program for 3 minutes. This means that the rotary table will continue moving in the forward direction for three minutes until the main program returns from the subroutine and changes the ON-OFF status of the relays R1, R2, and R3.

The rest of the program then does the following: a) stops the motor by executing a device code of 005, b) calls another subroutine delay program for 15 seconds located at 003 hi and 200 lo, c) inverses the ON-OFF status of R2 and R3, d) calls the same delay program for 3 minutes and then e) halts execution at the end.

The circuit for the Y axis and Z axis AC motors are identical to the one described above. All the three circuits use a common 74154 and 7406 chip.

The sequence of turning on and off the directional relays is totally at the discretion of the programmer. Therefore, it is left to the programmer to decide the sequence of movement of the robot arm. As can be seen in the next section, the operation of the DC motor or the electromagnet can be inserted into the program for AC motors and thereby have control software to execute any task within the capabilities of the robot.

3.2 DC MOTOR DRIVE CIRCUIT.

The DC stepper motor used is a PHILLIPS DC LOGIC MOTOR with non-salient poles and operating on +5V DC supply (12). Since the working voltage for the microcomputer is also +5V DC, no relay is required in this circuit. Apart from that, the socket equipped output port of the microcomputer which has been used as input to the DC motor drive circuit has inbuilt latches. Therefore no external latches have been used to maintain an input status at the port for a fixed interval of time.

The operating principle of a DC motor is shown in figure 19. The rotor of the DC motor is a permanent magnet with two poles-north and south. The stator has four seperate windings in it's armature. One end of each winding is connected to a +5V DC supply source, while the other end of the winding is connected to the collector of a TIP 29 NPN transistor (see figure 20).

The manufacturer's characteristics specifies a requirement of 96 steps to complete one rotation. Each time a set of windings are energized (or deenergized) the rotor makes a 3.75 turn called a 'step'. A brief explanation of the stepping mechanism is given below.

Let the initial input status be A - 1, B - 0, C - 0, and D - 0 (refer to figure 19). The section of the stator core through which winding A passes, produces a magnetic flux due to the

PRINCIPLES OF OPERATION
DC STEPPER MOTOR

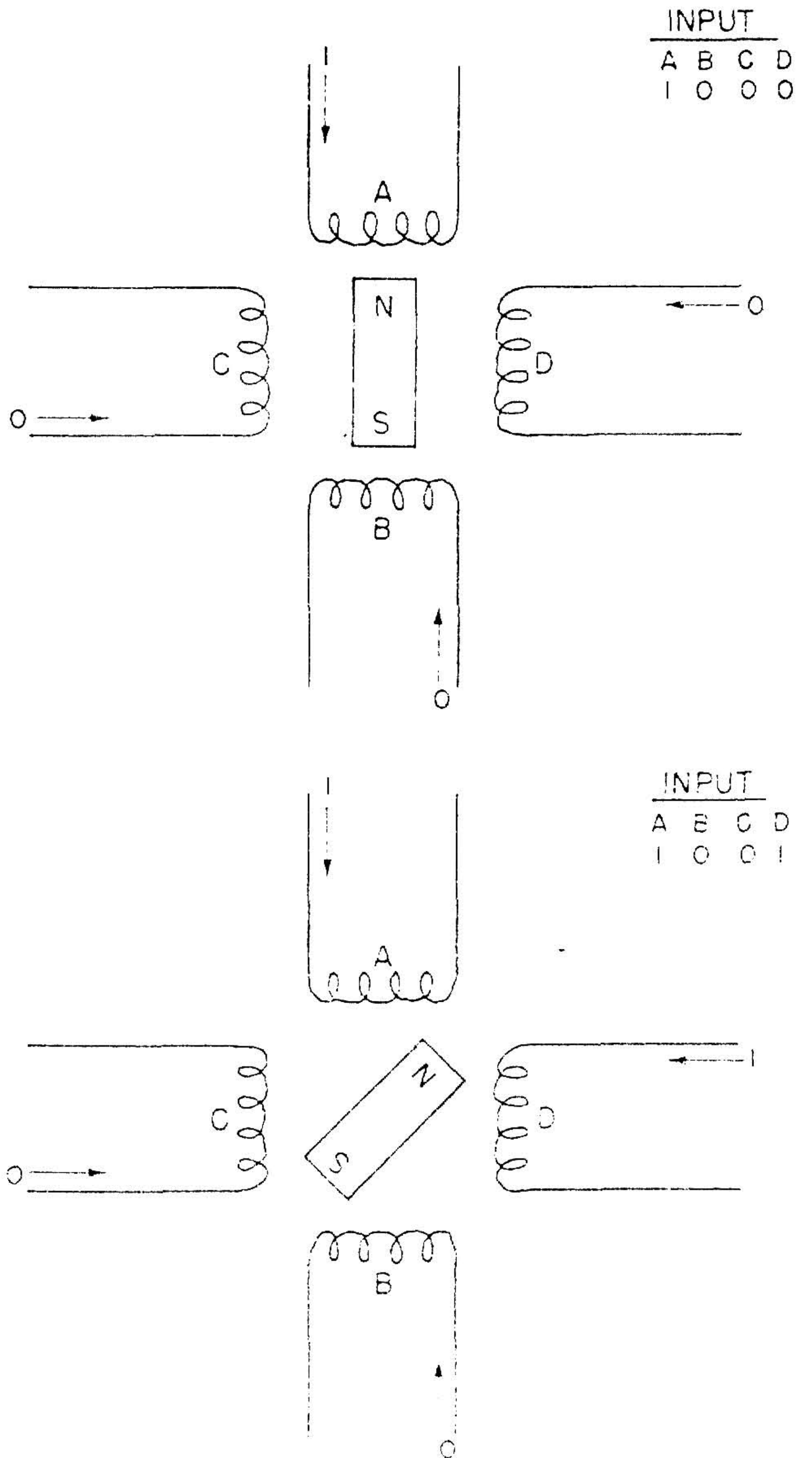
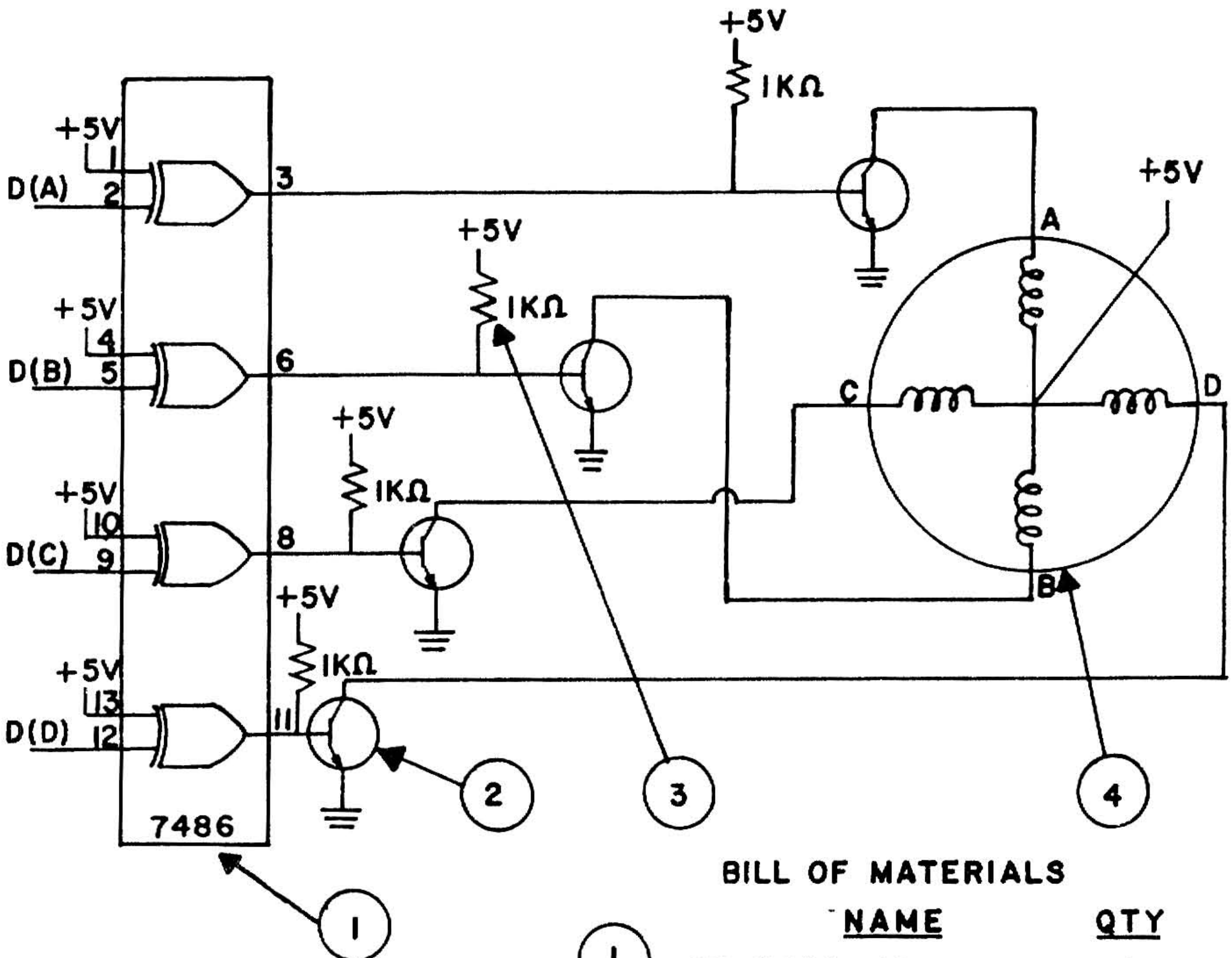


Figure 19. DC stepper motor principles of operation.

DRIVE CIRCUIT Z AXIS
DC STEPPER MOTOR



BILL OF MATERIALS

	NAME	QTY
1	SN 7486 IC	1
2	NPN TRANSISTOR TIP29	4
3	RESISTOR 1KΩ	4
4	PHILLIPS DC LOGIC MOTOR	1

PIN CONNECTIONS

1,4,10,13,14	+5V
3,6,8,11	TIP29
2,5,9,12	MMD-1 PORT 0
7	GROUND

Figure 20. DC motor circuit for Z axis linear mechanism.

current flowing through the winding A. The rotor which is a permanent magnet, interacts with the flux field to produce a force of attraction or repulsion depending on the initial location of the rotor. At the next moment when a different set of windings is energized (for example, A - 1, B - 0, C - 0, D - 1), the stator flux has moved in space compelling the rotor to move with it. The fact is, the faster the frequency of energizing (or deenergizing) process in the stator core, the higher the rotor (motor) RPM. Thus the speed of the DC stepper motor can be varied between a certain range.

At speeds lower than this range, the discrete steps of the motor will become visible. Attempts to run the motor at speeds higher than the range will produce vibrations.

The output of each individual exclusive-OR gate is connected in parallel to a 1 k.ohm pull up resistor before entering the base of the NPN transistor. The voltage at the output of each XOR gate is around 4.2V due to the resistances present within the gate. This voltage combines together with the voltage across the pull-up resistor to switch on the transistor and energise the necessary coil in the stator core. Refer to figure 19 for details.

As mentioned earlier, it is possible to vary the speed of the DC motor by varying the frequency at which the input status at the stator coil is changed. The motor speed thus becomes a direct function of the delay program run between two successive status inputs. Once an input is established at the stator coils, a substantial amount of time is allowed for the rotor to overcome the static load and make a turn of 3.75 in clockwise or anticlockwise direction. During this period of time the program is at the 'delay' section. Just before the turn or 'step' is completed the program returns to it's main section and initiates the next set of input at the stator coil - the rotation therefore seems continous.

If sufficient time elapses (at the delay program section) between two successive inputs, the stepping rate of the rotor is slowed down. At one point the stepping rate could be so low that the steps seem to be no more continous but intermittent.

The sequence of presentation of the binary digits at the input of the exclusive-OR gate will be in opposite cycle when the direction of the motor shaft rotation is reversed. Table 9 represents the sequence of codes for counterclockwise rotation of the motor.

Table 9. Counterclockwise codes for stepper motor.

BINARY CODE				OCTAL CODE
A	B	C	D	
1	0	0	0	010
1	0	1	0	012
0	0	1	0	002
0	1	1	0	006

0	1	0	0	004
0	1	0	1	005
0	0	0	1	001
1	0	0	1	011
1	0	0	0	010

The octal codes represented in binary digits appear at the port 0. Since the input of the XOR gate is the output at the port, pin2(A) will be at logic 1 and pins 5(B), 9(C), and 12(D) will be at logic 0. The truth table for an exclusive-OR gate is presented in table 10.

Table 10. EXCLUSIVE-OR logic table.

INPUT		OUTPUT
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

The output therefore will be logic 0 at pin 3 and logic 1 at the pins 6, 8, and 11. The logic 0 at pin 3 will allow current to flow across the 1 k.ohm resistor which in turn switches on the transistor and permits flow of +5V DC current from the center tap of the stator winding through coil A, collector and emitter of the transistor, down to the ground. The remaining three transistors, however, are not activated.

3.3 ELECTROMAGNET DRIVE CIRCUIT.

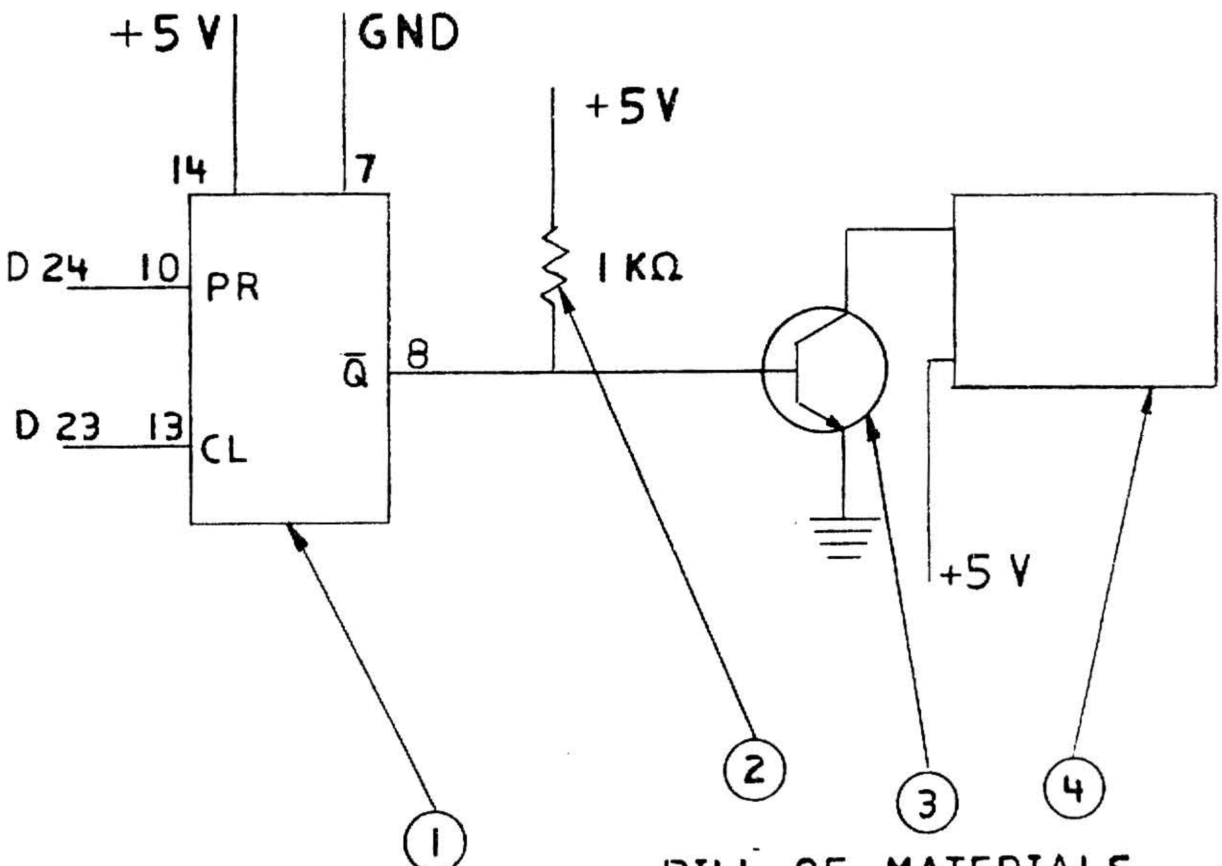
The electromagnet circuit is simple to operate due to the fact that each LED in three different ports are equipped with an in-built latch together with a plug-in socket. Once an output is thrown at a port, the status of the LED's do not change until the next output is initiated.

The output of the port acts as the input to a 7474 D-type positive edge triggered latch. The input being asynchronous at the PRESET and CLEAR pin, helps maintain either logic 0 or 1 at the output pin 7 (inverse of Q). The rest of the circuit resembles that of the DC motor circuit with a 1 k.ohm resistor and a TIP 29 NPN transistor.

Figure 21 shows the circuit for the gripper mechanism. Port 2 is used to generate the binary signals, where a logic 1 at the least significant socket deactivates the solenoid and a logic 1 at the next significant socket switches on the solenoid.

The electromagnet used for the purpose is a pull type continous duty solenoid operating on 6V DC (13).

ELECTROMAGNET DRIVE CIRCUIT



BILL OF MATERIALS

	NAME	QTY
①	DM 74LS74 N IC	1
②	RESISTOR 1K	1
③	NPN TRANSISTOR	1
④	6V DC ELECTROMAGN.	1

Figure 21. Gripper mechanism circuit.

CHAPTER 4

SOFTWARE DESIGN

4.1 TIMING CALCULATIONS REQUIRED FOR DELAY PROGRAMS

Displacements along different AC motor operated axes mechanisms are obtained by keeping the motors running for a finite period of time. The finite period of time is a function of the distance to be moved, the motor speed and the transmission ratios.

The software sends digital signals to the interfacing circuit which in turn generates the analog signal to turn on the motor. Once the motor is started, the software is at a 'delay' section where it runs amidst a network of finite loops. When the motor completes the finite run-time (at the end of which the desired displacement along that axis would have been achieved), the software returns from the delay section to generate the next analog signal to turn off the motor. The program at the delay section is called the 'delay program'.

4.1.1 DELAY PROGRAM FOR X AXIS LINEAR DISPLACEMENT

The X axis motor has a speed of 10 RPM and the number of threads per inch of the leadscrew is 10. The gear ratio for the motorshaft to the leadscrew is 26:32. Thus the time the motor needs to be operational for a half inch displacement of the table along the X axis can be calculated as follows.

1 revolution of the leadscrew requires 32/26 revolutions of the motor.

5 revolutions of the leadscrew requires 6.1538 revolutions of the motor.

10 revolutions of the motor is completed in 60 seconds.

6.1538 revolutions of the motor is completed in 36.9231 seconds.

Thus the delay program must run 36.9231 seconds for a half inch displacement of the rotary table along the X axis. The delay program for the X axis is shown in Table 11.

Table 11. Delay program for 1/2 inch linear displacement along X axis.

HI	LO	INSTR.BYTE	MNEM.CODE	EXECUTION TIME
030	210	036	mvi, E	9.33 microsec.
	211	001	<001>	
	212	006	mvi, B	
	213	034	<034>	
	214	016	mvi, C	
	215	000	nop	5.33 microsec.
	216	026	mvi, D	
	217	000	nop	
	220	025	dcr, D	6.66 microsec.

221	302	jnz	13.33 microsec.
222	220	lo	
223	030	hi	
224	015	dcr, C	
225	302	jnz	
226	216	lo	
227	030	hi	
230	005	dcr, B	
231	302	jnz	
232	214	lo	
233	030	hi	
234	035	dcr, E	
235	302	jnz	
236	212	lo	
237	030	hi	
240	311	ret	13.33 microsec

Delay program execution time;

REGISTER, B - 034 octal - 028 decrements.

REGISTER, C - 000 octal - 256 decrements.

REGISTER, D - 000 octal - 256 decrements.

Total time = 1(29.33+28(29.33+256(29.33+(20) 256)))+4(9.33)+13.33

= 36.911332 seconds.

Percentage error in the delay program is;

(36.9231-36.911332)/36.9231 % = 0.0318%

The displacement for the above program is 1/2 inch along the X axis. For higher or lower minimum displacement the data byte at the location 030 HI and 211 LO has to be increased or decreased accordingly.

4.1.2 DELAY PROGRAM FOR Y AXIS LINEAR DISPLACEMENT

The motor used for the Y axis is identical to that used for X axis. The leadscrew has a pitch of 10 TPI. The Gear ratio is 1:1 for the motor to the leadscrew. Then the time the motor must be operational for the table to move half an inch along the Y axis can be calculated as follows.

1 revolution of the leadscrew requires 1 revolution of the motor.

5 revolutions of the leadscrew requires 5 revolutions of the motor.

10 revolutions of the motor takes 60 seconds

5 revolutions of the motor takes 30 seconds

Thus the motor must be operational 30 seconds to effect half an inch movement of the table along the Y axis. This implies that the associated delay program must also take 30 seconds to execute. The delay program for the Y axis is shown in Table 12.

Table 12. Delay program for 1/2 inch linear displacement along Y axis.

HI	LO	INSTR. BYTE	MNEM. CODE
030	250	036	mvi, E
	251	001	<001> for 1/2 inch
	252	006	mvi, B
	253	027	<027>
	254	016	mvi, C
	255	000	nop
	256	026	mvi, D
	257	375	<375>
	260	025	dcr, D
	261	302	jnz
	262	260	lo
	263	030	hi
	264	015	dcr, C
	265	302	jnz
	266	256	lo
	267	030	hi
	270	005	dcr, B
	271	302	jnz
	272	254	lo
	273	030	hi
	274	035	dcr, E
	275	302	jnz
	276	252	lo

277	030	hi
300	311	ret

Delay program execution time;

REGISTER, B - 027 octal - 023 decrements.

REGISTER, C - 000 octal - 256 decrements.

REGISTER, D - 375 octal - 253 decrements.

Total time = 1(29.33+23(29.33+256(29.33+(20)253)))+4(9.33)+13.33

= 29.96675 seconds.

Percentage error in the delay program is;

(30-29.96675)/30 % = 0.110%.

The displacement for the above program is 1/2 inch along the Y axis. For higher or lower minimum displacement the data byte at the location 030 HI and 251 LO must be increased or decreased accordingly.

4.1.3 DELAY PROGRAM FOR Z AXIS ROTARY DISPLACEMENT.

The AC motor coupled to the Z axis rotary displacement is a 10 RPM motor. The gear ratio of the motor to the worm is 26:32. The other end of the worm has four threads spread over an inch. The gear which rotates the table has 40 teeth. Thus the time the motor must be operational for the table to move 15 degrees in circular motion about the Z axis can be calculated as follows.

360 degrees revolution is obtained by 40 rotation of the shaft.
15 degrees revolution is obtained by 1.667 rotation of the shaft.

1 revolution of the shaft requires 32/26 revolutions of the motor.

1.667 revolutions of the shaft requires 2.051282 revolutions of the motor.

10 rotations of the motor is accomplished in 60 seconds

2.051282 rotations of the motor is accomplished in 12.3077 seconds.

Thus the motor must be operational 12.3077 seconds to effect 15 degrees rotation of the table about the Z axis. This implies that the associated delay program must execute for 12.3077 seconds. The delay program for the Z axis rotary displacement is shown in table 13.

Table 13. Delay program for 15 degrees rotary displacement about Z axis.

HI	LO	INSTR.BYTE	MNEM.CODE
030	310	036	mvi, E
	311	001	<001> for 15 degrees
	312	006	mvi, B
	313	012	<012>

```
314     016     mvi, C
315     360     <360>
316     026     mvi, D
317     377     <377>
320     025     dcr, D
321     302     jnz
322     320     lo
323     030     hi
324     015     dcr, C
325     302     jnz
326     316     lo
327     030     hi
330     005     dcr, B
331     302     jnz
332     314     lo
333     030     hi
334     035     dcr, E
335     302     jnz
336     312     lo
337     030     hi
340     311     ret
```

Delay program execution time;

REGISTER, B - 012 octal - 010 decrements.

REGISTER, C - 360 octal - 240 decrements.

REGISTER, D - 377 octal - 255 decrements.

Total time = 1(29.33+10(29.33+240(29.33+(20)255)))+4(9.33)+13.33

= 12.310773 seconds.

Percentage error in the delay program is;

(12.310773-12.30769)/12.30769 % = 0.025%

Rotary displacement for the above delay program is 15 degrees about Z axis. For higher or lower minimum displacement, the data byte at the location 030 HI and 311 LO must be increased or decreased accordingly.

4.1.4 DELAY PROGRAM FOR Z AXIS LINEAR DISPLACEMENT

Whereas the other axes are driven by an AC motor, the Z axis is driven by a DC stepper motor. The linear movement along the Z axis enables the robot arm to travel up and down in a vertical motion.

Displacements along the Z axis are achieved by making the DC motor run through a definite number of steps. In other words, unlike AC motor driven axes systems where distance travelled is a direct function of the time period the motor is kept ON, displacement along this axis is a function of the total number of steps the DC motor takes. The software counts the number of steps actually taken and compares to the total required to achieve the desired displacement. When these totals are equal, the motor is turned off.

Unlike in previous sections, the delay program for this axis is to ensure a stepping rate between 200 to 250 steps/second. The delay program is stored beginning in location 035 HI and 200 LO. The entire delay program is shown in Table 14.

Table 14. Delay program for 0.3125 inch linear displacement along Z axis (fixed time).

HI	LO	INSTR.BYTE	MNEM.CODE
035	200	046	mvi, H
	201	013	<013>
	202	056	mvi, L
	203	023	<023>
	204	055	dcr, L
	205	302	jnz
	206	204	lo
	207	035	hi
	210	045	dcr, H
	211	302	jnz
	212	202	lo
	213	035	hi
	214	311	ret

The following calculations prove that the delay program will result in a stepping rate between 200 and 250 steps/sec.

Total delay program time = 11(29.33+19(20))+13.33 = 4505.96 microseconds.

To this delay program time, the time taken to throw the output to the rotor (13.33 microseconds) is added to give the time taken to move the DC motor one step. This one step rotates the DC motor through 3.75 degrees.

Thus for the DC motor to make one complete revolution it will approximately take;

(4505.96+13.33) 360/3.75 microseconds = 0.43385 seconds.

Thus in 1 second, the motor will rotate 360/0.43385 = 829.77 degrees. That is the motor will take 829.77/3.75 or 221 steps approximately.

The data byte at the location 034 HI and 005 LO (035 HI and 005 LO) of the Z axis movement section is for steps of 0.3125 inch. For higher or lower minimum displacement these data bytes have to be increased or decreased accordingly.

4.1.5 GENERAL PURPOSE DELAY PROGRAM

A 2 second delay program has been established at the location 037 HI and 200 LO. This delay program is called between changes in direction of movement or task. Table 15 shows the fixed delay program.

Table 15. Fixed delay program between tasks.

HI	LO	INSTR.BYTE	MNEM.CODE	COMMENTS
037	200	006	mvi, B	
	201	002	<002>	data for reg. B
	202	016	mvi, C	
	203	306	<306>	data for reg. C
	204	026	mvi, D	
	205	373	<373>	data for reg. D
	206	025	dcr, D	
	207	302	jnz	
	210	206	lo	
	211	037	hi	
	212	015	dcr, C	
	213	302	jnz	
	214	204	lo	
	215	037	hi	
	216	005	dcr, B	
	217	302	jnz	
	220	202	lo	
	221	037	hi	
	222	311	ret	

Total time = 1.999594 seconds.

The ultimate objective of microcomputer interfacing is to substitute software for hardware (5). The above principle relates to the theme of using microcomputer program in place of electronic and mechanical hardware devices.

In the current project, such a principle has been implemented in the 'instant brake apply' system. It has been discussed earlier that two of the important properties of an industrial robot is 'accuracy' and 'repeatability'. These properties are however difficult to maintain if the motors are unable to stop at the required instant of time. In particular, when the load is light and the AC motor rotors have a high inertia of motion, the motors may very well go for one or two more extra turns following disconnection of power supply. This results in extra displacement along the axis resulting in inaccuracy.

Such inaccuracies due to the electrical hardware can be compensated by software. This is accomplished by switching the motor in the reverse direction for one-tenth of a second. For AC motors with 10 RPM, a reverse motion switching for 0.1 second would counteract residual momentum in the direction of original movement. However the time elapsed between switching 'off' of motor and application of brakes should be kept minimal to reduce the margin of error. In this case this interval is equal to 37.3 microseconds.

The delay program for 0.1 second (approximately) is shown in Table 16.

Table 16. Delay program for reverse switching - instant brake apply (fixed time).

HI	LO	INSTR.BYTE	MNEM.CODE	COMMENTS
037	250	016	mvi, C	
	251	024	<024>	data for reg. C
	252	026	mvi, D	
	253	373	<373>	data for reg. D
	254	025	dcr, D	
	255	302	jnz	
	256	254	lo	
	257	037	hi	
	260	015	dcr, C	
	261	302	jnz	
	262	252	lo	
	263	037	hi	
	264	311	ret	

Total time = 0.100986 seconds.

4.2 GENERAL OVERVIEW OF SOFTWARE

The program to run the robot can be grouped into five distinct sections, each of which has it's own function. Apart from their individual functions, these five sections interact between themselves to produce the software control for the robot. These five sections are listed below.

a) Task definition section.

b) Task transfer section.

c) Task execution section.

d) Axis movement section.

e) Delay program section.

Figure 22 shows the general arrangement of the sections. The assembly language program contained within these sections are detailed in the appendix.

a) Task definition section : This section is the user defined section wherein the user is required to define the tasks in the desired sequence. Each task is completely defined by the direction or the action of movement and the distance. This requires three successive locations.

The first two locations are used to define either the action or direction of movement. Each action or direction of movement has it's own subprogram located within the software at various addresses. For example X+ is between 030 HI and 024 LO and 030 HI

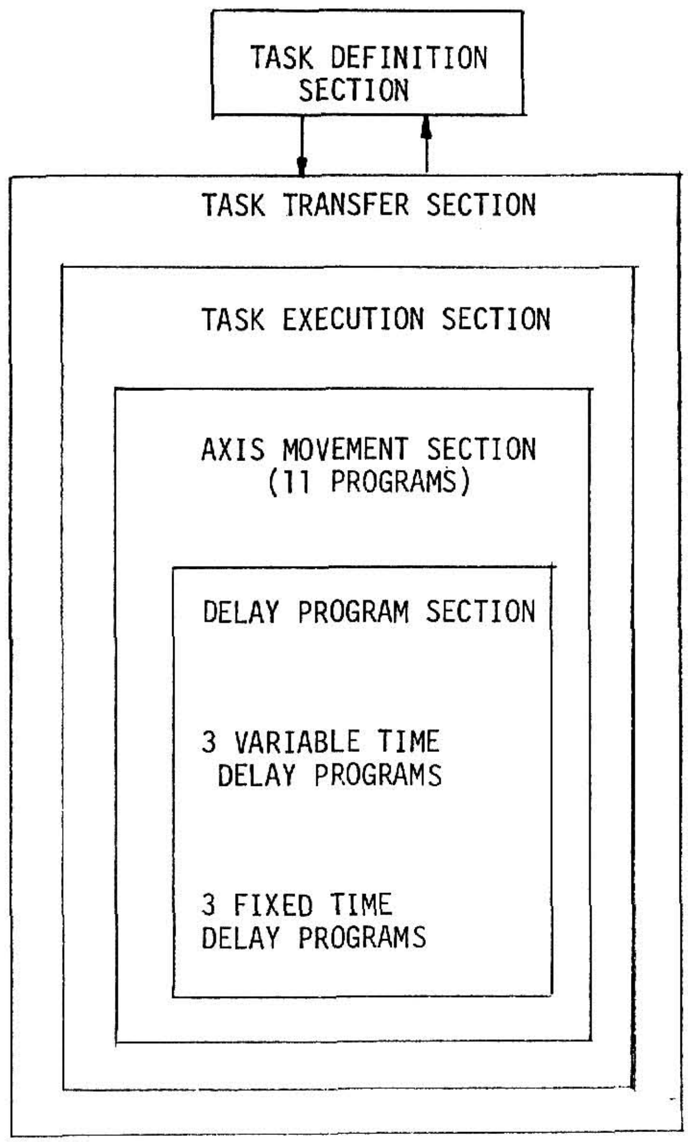


Figure 22. General Arrangement of the program sections.

and 063 LO, and X- is between 031 HI and 024 LO and 031 HI and 063 LO. Thus in the two locations, the user actually defines the starting address of the subprogram that will move the robot in the desired direction. Table 17 contains the addresses of the various subprograms which when executed will result in the robot moving in different directions.

The third location is used to define the magnitude in the direction of movement. In case of electromagnet activation and deactivation this byte can be 000 since it is not used.

In actuality, the task definition section is never 'run' but rather used as a 'memory'. The function of this section is to provide data to the executing sections of the program. The locations allotted to this section are between 037 HI and 000 LO and 037 HI and 177 LO (128 locations). This implies that a total of 42 different tasks can be sequenced at anyone time.

b) Task transfer section : The function of this section as the name implies, is to transfer data bytes from the task definition section to the task execution section. Also in addition to the previous task definition section, there is one location (036 HI and 001 LO) in this section wherein the user is required to input the number of tasks defined in the task definition section.

The program in this section is structured in a loop. Each pass through this loop transfers the set of three data required to perform one task. At the end of each pass, the number of tasks to be completed in sequence stored at location 036 HI and 001 LO

THIS BOOK IS OF POOR LEGIBILITY DUE TO LIGHT PRINTING THROUGH OUT IT'S ENTIRETY.

THIS IS AS RECEIVED FROM THE CUSTOMER.

Table 17. Locations of Task Subroutines (Axis Movement Section) and Their Associated Delay Programs

Axis Movement Sections.	Type of Power Drive	Located at Hi	Located at Lo	Delay Program at Hi	Delay Program at Lo	Stepping Rate Inches or Degrees	Location for Step Variation Hi	Location for Step Variation Lo	Maximum Travel Inches or Degrees
Initialization Switching Off of Motors and Electromagnet		030	000 020	-	-	-	-	-	-
X-Axis Linear Displacement	Positive	030	024 063	030	210	0.5 inch	030	211	4.15 inches
AC Motor	Negative	031	024 063		240				
Y-Axis Linear Displacement	Positive	030	064 122	030	250	0.5 inch	030	251	4.25 inches
AC Motor	Negative	031	064 122		300				
Z-Axis Rotary Displacement	Positive	030	124 163	030	310	15 Degrees	030	311	360°
AC Motor	Negative	031	124 163		340				
Z-Axis Linear Displacement	Positive	034	000 113	035	200	0.3125 inch	034	005	5.5 inches
DC Stepper Motor	Negative	035	000 113		214 Fixed Delay		035	005	
Gripping Mechanism	Grasp	034	150 161	-	-	-	-	-	-
Pull Type Solenoid	Release	035	150 161						

is decremented by 1. The remaining number of tasks is then displayed at port 1. Following completion of the specified number of tasks the program exits from the loop and the program execution is halted.

The program starts execution at the beginning location (036 HI and 000 LO) of this section and following completion of all the tasks halts execution at the location 036 HI and 045 LO. The initialization task (switching 'off' of the motors and deactivation of the solenoid) is also incorporated in this section. Figure 23 shows the logical flow diagram of this section and its interactions with the other sections.

c) Task execution section : There are two functions of the task execution section. The first is to save the existing register contents in the stacks for continued use of the two previous sections. The other is to call the appropriate axis movement subprogram, which when executed will result in desired motion. This task execution section is executed once for each distinct task.

The interaction of this section with the task transfer section and axis movement section is shown in figure 23.

d) Axis movement section : The purpose of this section is to generate analog signals to the electrical components (motors, solenoid) via interfacing circuit. This section contains the various subprograms which direct the movement along different axes. There are eleven axis movement subprograms (subroutines) under this section . They are as follows;

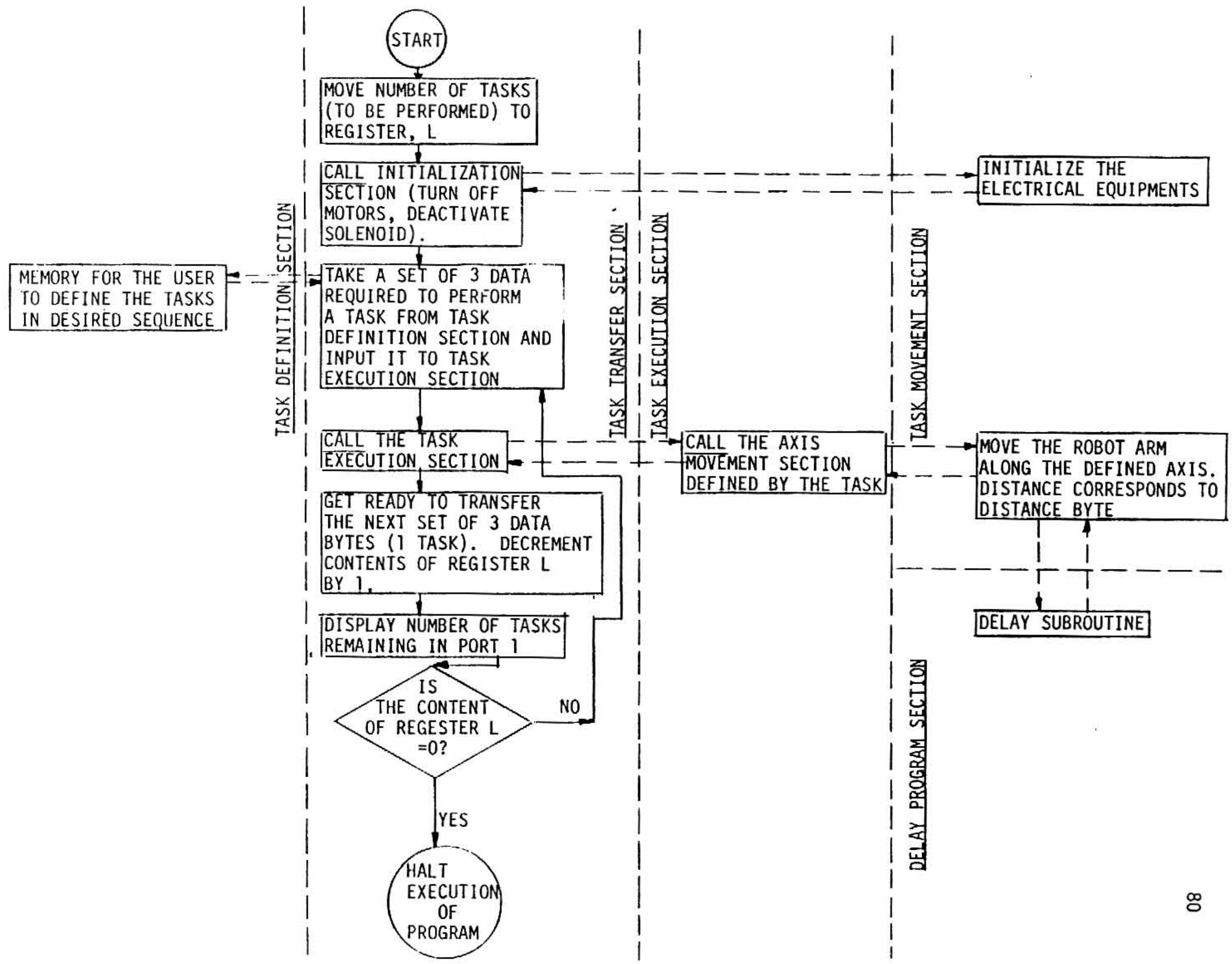


Figure 23. Logical flow diagram of the task transfer section and the task execution section together with associated interactions.

1) Initialization subroutine; used to turn off the motor and deactivate the solenoid. This subroutine is necessarily executed at the beginning of program execution.

2) X+ subroutine; for movement along X axis in positive direction.

3) X- subroutine; for movement along X axis in negative direction.

4) Y+ subroutine; for movement along Y axis in positive direction.

5) Y- subroutine; for movement along Y axis in negative direction.

6) Z + subroutine; for rotary movement about Z axis in counter clockwise direction.

7) Z - subroutine; for rotary movement about Z axis in clockwise direction.

8) Z+ subroutine; for movement along Z axis in positive direction.

9) Z- subroutine; for movement along Z axis in negative direction.

10) Electromagnet activation; grasps the end load.

11) Electromagnet deactivation; releases the end load.

The physical locations of these subroutines together with their associated delay subprograms are shown in Table 17. Figure 24 shows the logical flow diagram of the axis movement section along with the delay program section.

e) Delay program section : The function of this section is to

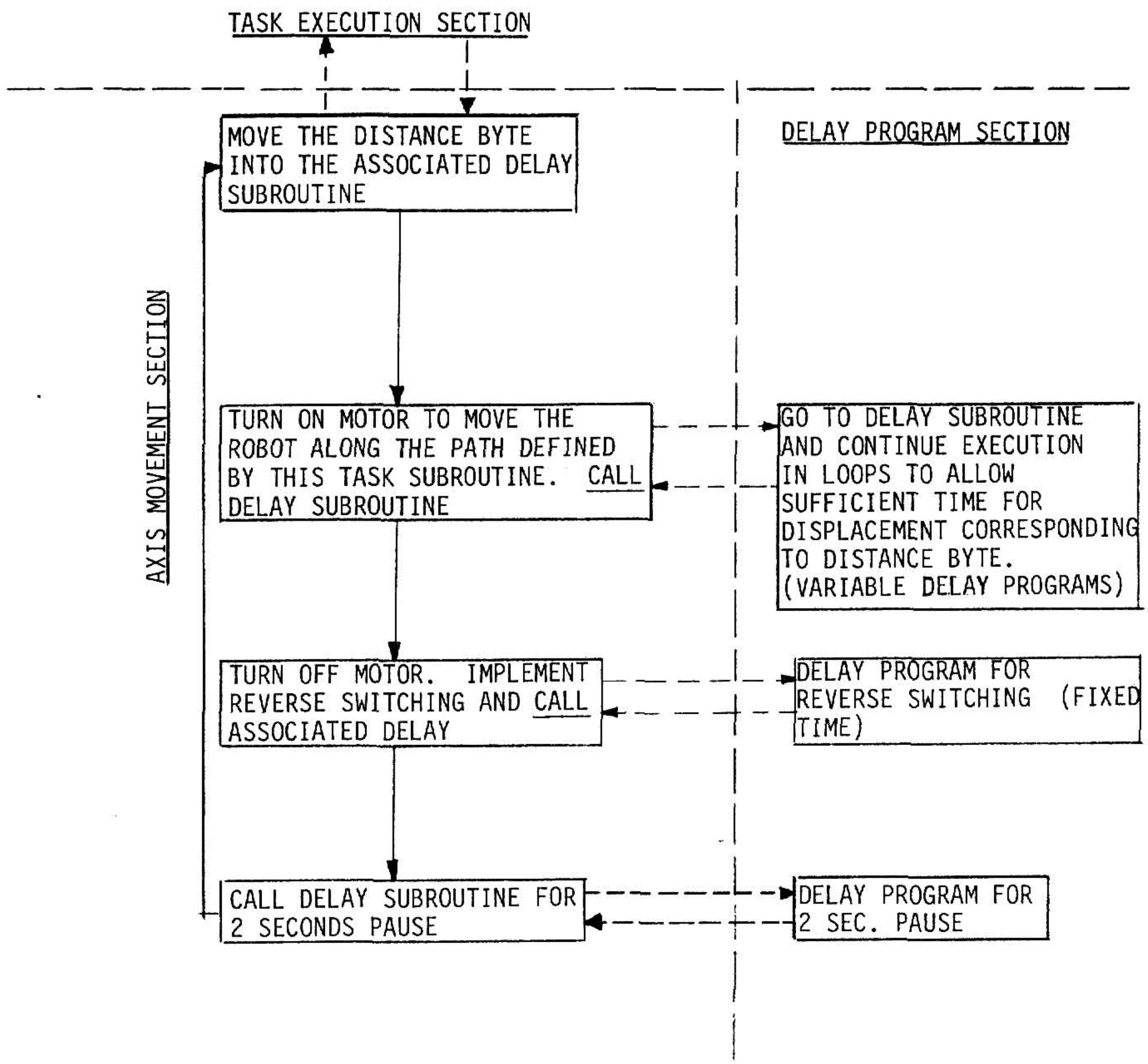


Figure 24. Logical flow diagram of the axis movement section together with associated interactions.

vary the duration of time during which the motors are kept operating. The duration of time is directly related to the distance to be moved along an axis. The delay program (fixed time) for the DC motor ensures a stepping rate within the range specified by the motor manufacturer. The solenoid does not require any delay program. Table 17 shows the location of the delay programs.

Other than the delay programs associated with the various axes movement programs (shown in Table 17), there are two fixed execution time 'General purpose delay programs'. The first program located at 037 HI and 250 LO has a fixed execution time of 0.1 seconds; it is called during reverse switching. The second delay program located at 037 HI and 200 LO has a fixed execution time of 2 seconds; this program is called between completion of one task and the start of the next task.

The details of the delay programs have been mentioned in the previous sections of this chapter.

4.3 STEPS FOR RUNNING THE ROBOT

Figure 25 shows the list of tasks that the user has to perform before running the robot.

The first task is to make the proper hardware connections for the AC motors, the DC motor and the solenoid.

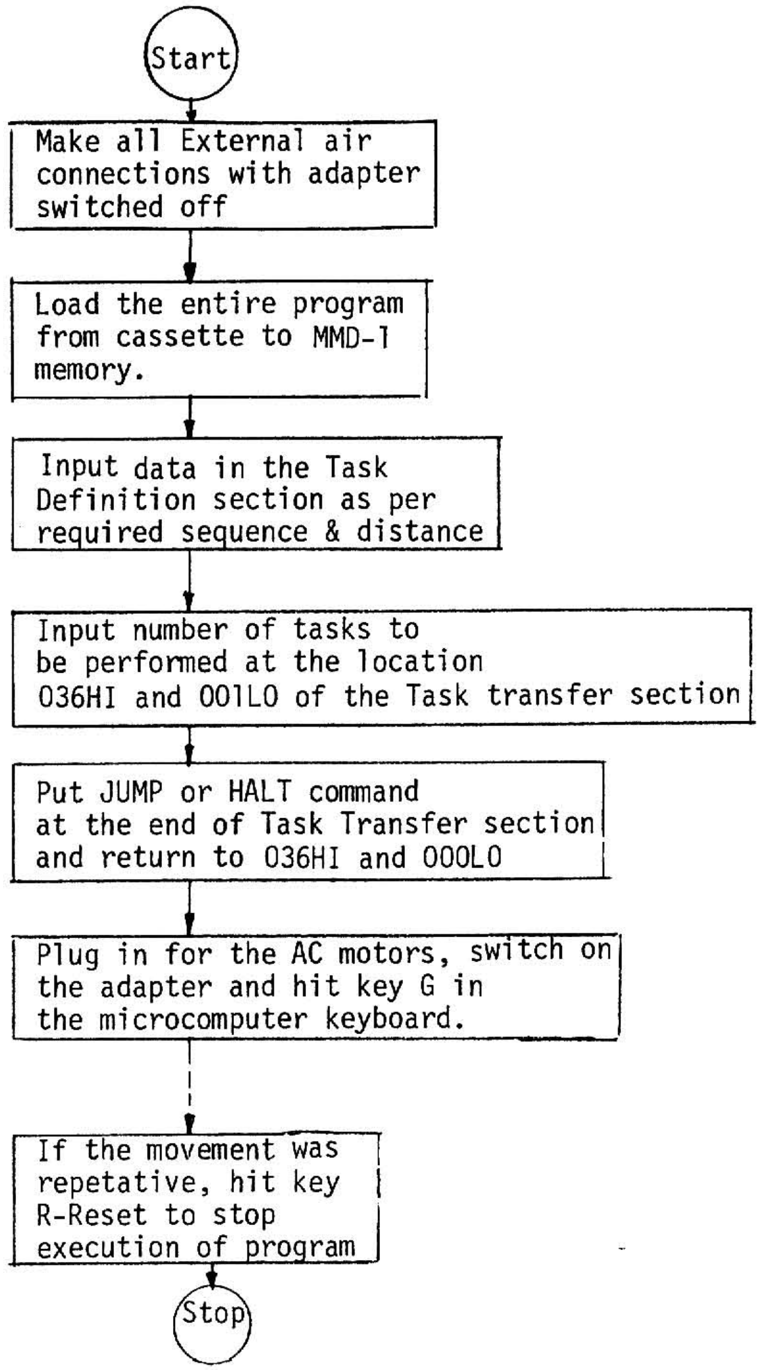


Figure 25. List of tasks in sequence prior to program execution.

1) AC motor connections;

a) connect OUT of MMD-1 to pin 5 of the second 7432.

b) for motor X, connect forward displacement wire to R2 and reverse displacement wire to R1.
for motor Y, connect forward displacement wire to R5 and reverse displacement wire to R4.
for motor Z, connect clockwise rotation wire to R8 and counterclockwise rotation wire to R7.

c) connect one wire of the AC plug to the farthest row of the 'Experimentor' breadboard (for R3, R6 and R9:pin 8).

d) connect the other wire of the plug to ground (neutral) of all the three motors.

e) connect pins A0, A1, A2,and A3 of the address bus to pins 23, 22, 21, and 20 of the 74154 decoder.

f) connect GND of MMD-1 to the ground of the breadboards.

2) DC motor connections;

a) connect pins 2, 5, 9, and 12 of 7486 to pins 0, 1, 2, and 3 of port 0 respectively.

b) connect wires labelled A, B, C, and D of the DC motor to the mid-pins (collector) of the transistors 1, 2, 3, and 4.

c) connect the fifth wire (labelled +5V) of the DC motor to +5V of the MMD-1 microcomputer.

3) Solenoid connections;

a) connect sockets 0 and 1 of port 2 to pins 13 and 10 of 7474 chip.

b) connect mid-pin (collector) of the transistor to one input point of the electromagnet.

c) connect another pin of the electromagnet to +5 of the MMD-1 computer.

4) General wire connections; connect +5V and ground of the adapter to +5V and ground of the breadboard.

The entire program to run the robot is recorded on a cassette tape. This program should be loaded into the microcomputer memory before running the robot.

The second task is for the user to define the sequence of tasks. Each task should be defined in it's entirety (direction or action and magnitude). Once the user has defined the sequence of tasks, this sequence should be input into the microcomputer as part of the task definition section.

Following this the user has to input the total number of tasks in location 036 HI and 001 LO.

If the sequence of task is to be executed once, a HALT instruction is to be input at the end of the task transfer section in location 036 HI and 045 LO. If the desired sequence of task is to be repeated continously, a JUMP (unconditional) instruction should be input at the same location. The program in this instance jumps back to either 036 HI and 000 LO or to 036 HI and 010 LO.

The last step is to turn on the power for the motors, solenoid and the breadboard and hit the G (Go) key on the keyboard. If all is well, the robot will execute the desired sequence of tasks.

It is important to ensure that a logic 0 prevails at the output pin 5 of all the three latches (7474 chip) of the AC motor circuit before the AC motors receive a power supply of 110V AC. This is to avoid the motors functioning before the G key is pressed.

The reason for above is that once the adapter is switched on, the breadboards receive DC power supply. The output conditions of the latches at this moment are established arbitrarily. Therefore if a logic 1 prevails at pin 5 of any positive-edge triggerred latch, the corresponding ON/OFF relay will complete the AC circuit on the 110V side. As soon as the motors are plugged into the 110V power source, the motor will start running even if the G key in the microcomputer has not been pressed.

A probable solution to the above problem is to switch on 110V supply and press G key at the same time. But, perhaps the best approach is to provide a 0V DC supply, for an instant to pin 1 of all the three latches before the AC motors are powered.

If the program is stopped by hitting the RESET key, it is necessary to bring the robot back to the original position. On the axes powered by the AC motors, it can be brought back by manually switching on the motors. For the DC motor powered axis, the leadscrew has to be manually turned till the home position is reached.

4.4 A SAMPLE PROGRAM

As discussed earlier in this chapter a task sequence can consist of 42 distinct tasks for the installed program. Further, as mentioned before, the data to be input in the task definition section must be in the same sequence in which the action (direction of movement) is desired.

To illustrate the data input in task definition section consider the following sequence of tasks that the robot is to execute.

TASK 1 : move 1 inch along X+ direction.

TASK 2 : rotate 60 degrees about Z axis in CCW direction.

TASK 3 : grasp the load.

TASK 4 : move 2.5 inch along Z+ axis.

TASK 5 : rotate 30 degrees about Z axis in CW direction.

TASK 6 : move 1 inch along Y- direction.

TASK 7 : release the load.

Table 18 details the actual code to be input in the task definition section for the above sequenced tasks. The addresses from the various subprograms came from Table 17. The total number of tasks is equal to 7, and therefore, the data byte 007 must be input in the location 036 HI and 001 LO. Assuming the tasks end with the release of the load, a HALT command is to be input at the location 036 HI and 045 LO.

Table 18. Codes for sample program

HI	LO	INSTR.BYTE	COMMENTS
037	000	024	lo of X+
	001	030	hi of X+
	002	002	for 1 inch travel
	003	124	lo of Z-ccw
	004	030	hi Z-ccw
	005	004	for 60 degrees rotation
	006	150	lo of grasp
	007	034	hi of grasp
	010	000	----
	011	000	lo of Z+
	012	034	hi of Z+
	013	010	for 2.5 inch travel
	014	124	lo of Z-cw
	015	031	hi of Z-cw
	016	002	for 30 degrees rotation
	017	064	lo of Y-
	020	031	hi of Y+
	021	002	for 1 inch travel
	022	150	lo of release
	023	035	hi of release
	024	000	----

The program is now ready to be executed beginning at the location 036 HI and 000 LO.

This is just one example of a sequence of tasks. Any sequence of such tasks can be executed by inputting the appropriate codes in the task definition section.

CHAPTER 5

CONCLUSIONS AND FUTURE DIRECTIONS

5.1 CONCLUSIONS

As stated in section 6 of chapter 1, the research was undertaken with two objectives in mind.

a) To design, develop and build a laboratory model of a small articulated robot with four degrees of freedom.

b) To demonstrate programming flexibility using a microcomputer.

With the reference to the first objective, a manually controlled rotary table has been successfully converted to a microrobot with four degrees of freedom. Additionally it has also been fitted with a gripper mechanism. This robot was designed and built within the resource (financial and material) constrained environment prevalent within the department. This prevented the purchase of new powerful high speed motors for increased speed in robot movement. Since it was designed for student for student use in the laboratory, the payload is restricted to 12 oz.s.

With respect to the second objective, a generalized software has been written. The software code permits execution of any defined sequence of tasks thus demonstrating the flexibility. The software has been so designed that data input by the user is kept to a minimum. Further, the software is upward compatible. Thus if a bigger microcomputer could converse with the MMD-1, the task definition need not be input by the user directly. The definition could be downloaded, from the bigger microcomputer, one task at a time. This capability permits the robot and the associated software to be integrated in a centrally controlled model of an automated factory.

The hardware and the software are so designed that, minimal changes have to be made if one or two more degrees of freedom are added to the robot. This further underscores the flexibility in the control of the robot. Finally, the hardware and software design can be easily adapted to other such motor driven systems.

5.2 FUTURE DIRECTIONS

One of the major limitations of the robot is the lack of feedback. Future work in the robot should incorporate feed back mechanisms.

The robot can be improved in other ways. Firstly, lighter materials can be used for the base and the supports. Secondly, high speed motors with braking can be installed for the various drives. Such high speed motors will not only improve the speed of the robot but also it's accuracy.

A major research effort in the future is to integrate this robot in a centrally controlled model. This would involve establishing communication protocols between the central computer and the MMD-1. The tasks will be defined in a user oriented language rather than in machine code. The central computer will convert the user defined tasks to machine codes and transfer them to the MMD-1 which will then act upon them. This effort without doubt will be a major undertaking at the masters level.

REFERENCES

REFERENCES

1) Kramer, Brad E., "Robots", Unpublished report, Dept. of Industrial Engg., Kansas State University, Fall, 1981.

2) SME, "Robotics Today", Fall, 1981.

3) SME, "Robotics Today", Spring, 1980.

4) SME, "Robotics Today", Fall, 1979.

5) Tanner, William, "Industrial Robots", Robotics International of SME, Dearborn, Michigan, Vol 1, 1981.

6) Higdon, Archie and Stiles, William, "Engineering Mechanics", Prentice-Hall, Inc., Englewood Cliffs, N.J., 1955.

7) Wilson, Frank and Harvey, Phillips, "Tool Engineers Handbook", McGraw-Hill Book Co., Inc., New york, 1959.

8) Graingers, "Motorbook Catalog", W. W. Grainger, Inc., Topeka, KS., No. 352, Winter 1978-79.

9) Mignano, Bruce, "Microcomputer Controlled Two axes Milling Machine", Unpublished MS Thesis, Dept. of Industrial Engg., Kansas State University, 1981.

10) Larsen, David, Rony, Peter and Titus, Jonathan, "The Bug Book V", E & L Instruments, Inc., Danbury, CT, May 1977.

11) Larsen, David, Rony, Peter and Titus, Jonathan, "The Bug Book VI", E & L Instruments, Inc., Danbury, CT, May 1977.

12) Airpax, "Stepper Motor Handbook", North American Philips Corp., Chesere, Conn, 1981.

13) Herbach and Rademan, Inc., "Motorbook Catalog", Philadelphia, PA., Vol 46#5, Summer 1980.

THIS BOOK CONTAINS NUMEROUS PAGES THAT WERE BOUND WITHOUT PAGE NUMBERS.

THIS IS AS RECEIVED FROM CUSTOMER.

APPENDIX

A. TASK DEFINITION SECTION.

HI	LO	INSTR.BYTE	COMMENT
037	000	---	lo address of subprogram for 1st task
	001	---	hi address of subprogram for 1st task
	002	---	distance byte for 1st task
	003	---	lo address of subprogram for 2nd task
	004	---	hi address of subprogram for 2nd task
	005	---	distance byte for 2nd task
	---	---	
	---	---	
	---	---	
	---	---	
	---	---	
	---	---	
	---	---	
	---	---	
	---	---	
037	177	---	last possible address in this section

B. TASK TRANSFER SECTION

HI	LO	INSTR. BYTE	MNEM. CODE	COMMENTS
036	000	056	mvi, L	move to reg. L
	001	xxx	<xxx>	number of tasks
	002	000	nop	
	003	315	call	initialization
	004	000	lo	
	005	030	hi	
	006	000	nop	
	007	000	nop	
	010	026	mvi, D	move to reg. D
	011	036	<036>	
	012	036	mvi, E	move to reg. E
	013	211	<211>	
	014	006	mvi, B	move to reg. B
	015	037	<037>	
	016	016	mvi, C	move to reg. C
	017	000	<000>	
	020	012	ldax, B	load into accum.
	021	022	stax, D	store accum. contents
	022	034	inr, E	increment E by 1
	023	014	inr, C	increment C by 1
	024	012	ldax, B	load into accum.
	025	022	stax, D	store accum. contents
	026	014	inr, C	increment C by 1

027	012	ldax, B	load for distance
030	315	call	task exection section
031	200	lo	
032	036	hi	
033	000	nop	
034	014	inr, C	increment C by 1
035	035	dcr, E	decrement E by 1
036	055	dcr, L	decrement L by 1
037	175	mov A, L	move accum. cont. to L
040	323	out	output accum. contents
041	001	port 1	
042	302	jnz	conditional jump
043	020	lo	
044	036	hi	
045	166	halt	execution

C. TASK EXECUTION SECTION

HI	LO	INSTR.BYTE	MNEM.CODE	COMMENTS
036	200	305	push B	into stack
	201	325	push D	into stack
	202	345	push H	into stack
	203	000	nop	
	204	000	nop	
	205	000	nop	
	206	000	nop	
	207	000	nop	
	210	315	call	axis movement section
	211	xxx	lo	
	212	xxx	hi	
	213	000	nop	
	214	000	nop	
	215	341	pop H	from the stack
	216	321	pop D	from the stack
	217	301	pop B	from the stack
	220	311	ret	to task transf. section

D. AXIS MOVEMENT SECTION.

a) SWITCHING OFF OF MOTORS AND DEACTIVATING SOLENOID.

HI	LO	INSTR. BYTE	MNEM.CODE	COMMENTS
030	000	000	nop	
	001	000	nop	
	002	000	nop	
	003	323	out	
	004	005	<005>	motor x (R3) off.
	005	000	nop	
	006	323	out	
	007	012	<012>	motor y (R6) off.
	010	000	nop	
	011	323	out	
	012	017	<017>	motor z (R9) off.
	013	000	nop	
	014	076	mvi,A	
	015	001	<001>	deactivate solenoid.
	016	323	out	
	017	002	port 2	
	020	311	ret	to section c).

b) FORWARD DISPLACEMENT ALONG X-AXIS(POSITIVE).

HI	LO	INSTR.BYTE	MNEM.CODE	COMMENTS
003	024	062	sta	store accum.
	025	211	<211>	lo
	026	030	<030>	hi
	027	000	nop	
	030	323	out	
	031	003	<003>	R2 (forward) relay on.
	032	000	nop	
	033	323	out	
	034	004	<004>	R3 (ON/OFF) relay on.
	035	000	nop	
	036	315	call	subroutine for X.
	037	210	lo	
	040	030	hi	
	041	323	out	
	042	005	<005>	R3 (ON/OFF) relay off.
	043	000	nop	
	044	323	out	
	045	001	<001>	R1 (reverse relay) on.
	046	000	nop	
	047	323	out	
	050	004	<004>	R3 (ON/OFF) relay on.
	051	000	nop	
	052	315	call	subroutine (reverse).

053	250	lo	
054	037	hi	
055	323	out	
056	005	<005>	R3 (ON/OFF) relay off.
057	000	nop	
060	315	call	subroutine (break).
061	200	lo	
062	037	hi	
063	311	ret	to section c).

c) FORWARD DISPLACEMENT ALONG Y-AXIS (POSITIVE).

HI	LO	INSTR.BYTE	MNEM.CODE	COMMENTS
030	064	062	sta	store accum.
	065	251	<251>	lo
	066	030	<030>	hi
	067	000	nop	
	070	323	out	
	071	010	<010>	R5 (forward) relay on.
	072	000	nop	
	073	323	out	
	074	011	<011>	R6 (ON/OFF) relay on.
	075	000	nop	
	076	315	call	Subroutine for Y.
	077	250	lo	
	100	030	hi	
	101	323	out	
	102	012	<012>	R6 (ON/OFF) relay off.
	103	000	nop	
	104	323	out	
	105	007	<007>	R4 (reverse) relay on.
	106	000	nop	
	107	323	out	
	110	011	<011>	
	111	000	nop	
	112	315	call	subroutine (reverse).

```
113     250     lo
114     037     hi
115     323     out
116     012     <012>     R6 (ON/OFF) relay off.
117     000     000

120     315     call      subroutine (break).
121     200     lo
122     037     hi
123     311     ret       to section c).
```

d) ROTARY DISPLACEMENT (COUNTER-CLOCKWISE) ABOUT Z-AXIS.

HI	LO	INSTR.CODE	MNEM.CODE	COMMENTS
030	124	062	sta	store accum.
	125	311	<311>	lo
	126	030	<030>	hi
	127	000	nop	
	130	323	out	
	131	014	<014>	R7 (co-cl) relay on.
	132	000	nop	
	133	323	out	
	134	016	<016>	R9 (ON/OFF) relay on.
	135	000	nop	
	136	315	call	subroutine for Z.
	137	310	lo	
	140	030	hi	
	141	323	out	
	142	017	<017>	R9 (ON/OFF) relay off.
	143	000	nop	
	144	323	out	
	145	015	<015>	R8(reverse) relay on.
	146	000	nop	
	147	323	out	
	150	016	<016>	R9 (ON/OFF) relay on.
	151	000	nop	
	152	315	call	subroutine (reverse).

153	250	lo	
154	037	hi	
155	323	out	
156	017	<017>	R9 (ON/OFF) relay off.
157			
160	315	call	
161	200	lo	
162	037	hi	
163	311	ret	to section c).

e) REVERSE DISPLACEMENT ALONG X-AXIS (NEGATIVE).

HI	LO	INSTR.BYTE	MNEM.CODE	COMMENT
031	024	062	sta	store accum.
	025	211	<211>	lo
	026	030	<030>	hi
	027	000	nop	
	030	323	out	
	031	001	<001>	R1 (back.) relay on.
	032	000	nop	
	033	323	out	
	034	004	<004>	R3 (ON/OFF) relay on.
	035	000	nop	
	036	315	call	subroutine for X.
	037	210	lo	
	040	030	hi	
	041	323	out	
	042	005	<005>	R3 (ON/OFF) relay off.
	043	000	nop	
	044	323	out	
	045	003	<003>	R2 (reverse) relay on.
	046	000	nop	
	047	323	out	
	050	004	<004>	R3 (ON/OFF) relay on.
	051	000	nop	
	052	315	call	subroutine (reverse).

053	250	lo	
054	037	hi	
055	323	out	
056	005	<005>	R3 (ON/OFF) relay off.
057	000	nop	
060	315	call	subroutine (break).
061	200	lo	
062	037	hi	
063	311	ret	to section c).

f) REVERSE DISPLACEMENT ALONG Y-AXIS (NEGATIVE).

HI	LO	INSTR.BYTE	MNEM.CODE	COMMENTS
031	064	062	sta	store accum.
	065	251	<251>	lo
	066	030	<030>	hi
	067	000	nop	
	070	323	out	
	071	007	<007>	R4 (back.) relay on.
	072	000	nop	
	073	323	out	
	074	011	<011>	R6 (ON/OFF) relay on.
	075	000	nop	
	076	315	call	subroutine for Y.
	077	250	lo	
	100	030	hi	
	101	323	out	
	102	012	<012>	R6 (ON/OFF) relay off.
	103	000	nop	
	104	323	out	
	105	010	<010>	R5 (reverse) relay on.
	106	000	nop	
	107	323	out	
	110	011	<011>	R6 (ON/OFF) relay on.
	111	000	nop	
	112	315	call	subroutine (reverse).

113	250	lo	
114	037	hi	
115	323	out	
116	012	<012>	R6 (ON/OFF) relay off.
117	000	nop	
120	315	call	subroutine (break).
121	200	lo	
122	037	hi	
123	311	ret	to section c).

g) ROTARY DISPLACEMENT (CLOCKWISE) ABOUT Z-AXIS.

HI	LO	INSTR.BYTE	MNEM.CODE	COMMENTS
031	124	062	sta	store accum.
	125	311	<311>	lo
	126	030	<030>	hi
	127	000	nop	
	130	323	out	
	131	015	<015>	R8 (clock.) relay on.
	132	000	nop	
	133	323	out	
	134	016	<016>	R9 (ON/OFF) relay on.
	135	000	nop	
	136	315	call	subroutine for Z.
	137	310	lo	
	140	030	hi	
	141	323	lo	
	142	017	<017>	R9 (ON/OFF) relay off.
	143	000	nop	
	144	323	out	
	145	014	<014>	R7 (reverse) relay on.
	146	000	nop	
	147	323	out	
	150	016	<016>	R9 (ON/OFF) relay on.
	151	000	nop	
	152	315	call	subroutine (reverse).

153	250	lo	
154	037	hi	
155	323	out	
156	017	<017>	R9 (ON/OFF) relay off.
157	000	nop	
160	315	call	subroutine (break).
161	200	lo	
162	037	hi	
163	311	ret	to section c).

h) VERTICAL DISPLACEMENT ALONG Z AXIS (UP). CLOCKWISE ROTATION OF THE DC MOTOR.

HI	LO	INSTR.BYTE	MNEM.CODE	COMMENTS
034	000	062	sta	store accum.
	001	005	<005>	lo
	002	034	<034>	hi
	003	000	nop	
	004	006	mvi, B	
	005	001	<001>	for 0.3125 inch.
	006	016	mvi, C	
	007	140	<140>	decimal 96.
034	010	076	mvi, A	
	011	010	<010>	
	012	323	out	
	013	000	port 0	
	014	315	call	fixed delay.
	015	200	lo	
	016	035	hi	
034	017	076	mvi, A	
	020	011	<011>	
	021	323	out	
	022	000	port 0	
	023	315	call	fixed delay.

```
        024     200         lo
        025     035         hi

034     026     076         mvi, A
        027     001         <001>
        030     323         out
        031     000         port 0
        032     315         call          fixed delay.
        033     200         lo
        034     035         hi

034     035     076         mvi, A
        036     005         <005>
        037     323         out
        040     000         port 0
        041     315         call          fixed delay.
        042     200         lo
        043     035         hi

034     044     076         mvi, A
        045     004         <004>
        046     323         out
        047     000         port 0
        050     315         call          fixed delay.
        051     200         lo
        052     035         hi

034     053     076         mvi, A
        054     006         <006>
```

```
       055    323         out
       056    000         port 0
       057    315         call             fixed delay.
       060    200         lo
       061    035         hi

034    062    076         mvi, A
       063    002         <002>
       064    323         out
       065    000         port 0
       066    315         call             fixed delay.
       067    200         lo
       070    035         hi

034    071    076         mvi, A
       072    012         <012>
       073    323         out
       074    000         port 0
       075    315         call             fixed delay.
       076    200         lo
       077    035         hi

034    100    015         dcr, C
       101    302         jnz
       102    010         lo
       103    034         hi

034    104    005         dcr, B
       105    302         jnz
```

	106	006	lo	
	107	034	hi	
034	110	315	call	subroutine (break).
	111	200	lo	
	112	037	hi	
	113	311	ret	to section c).

i) VERTICAL DISPLACEMENT ALONG Z AXIS (DOWN). COUNTERCLOCKWISE ROTATION OF THE DC MOTOR.

HI	LO	INSTR.BYTE	MNEM.CODE	COMMENTS
035	000	062	sta	store accum.
	001	005	<005>	lo
	002	035	<035>	hi
	003	000	nop	
	004	006	mvi, A	
	005	001	<001>	for 0.3125 inch.
	006	016	mvi, B	
	007	140	<140>	decimal 96.
035	010	076	mvi, A	
	011	010	<010>	
	012	323	out	
	013	000	port 0	
	014	315	call	fixed delay.
	015	200	lo	
	016	035	hi	
035	017	076	mvi, A	
	020	012	<012>	
	021	323	out	
	022	000	port 0	
	023	315	call	fixed delay.
	024	200	lo	

```
       025    035       hi

035    026    076       mvi, A
       027    002       <002>
       030    323       out
       031    000       port 0
       032    315       call          fixed delay.
       033    200       lo
       034    035       hi

035    035    076       mvi, A
       036    006       <006>
       037    323       out
       040    000       port 0
       041    315       call          fixed delay.
       042    200       lo
       043    035       hi

035    044    076       mvi, A
       045    004       <004>
       046    323       out
       047    000       port 0
       050    315       call          fixed delay.
       051    200       lo
       052    035       hi

035    053    076       mvi, A
       054    005       <005>
       055    323       out
```

	056	000	port 0	
	057	315	call	fixed delay.
	060	200	lo	
	061	035	hi	
035	062	076	mvi, A	
	063	001	<001>	
	064	323	out	
	065	000	port 0	
	066	315	call	fixed delay.
	067	200	lo	
	070	035	hi	
035	071	076	mvi, A	
	072	011	<011>	
	073	323	out	
	074	000	port 0	
	075	315	call	fixed delay.
	076	200	lo	
	077	035	hi	
035	100	015	dcr, C	
	101	302	jnz	
	102	010	lo	
	103	035	hi	
035	104	005	dcr, B	
	105	302	jnz	
	106	006	lo	

	107	035	hi	
035	110	315	call	subroutine (break).
	111	200	lo	
	112	037	hi	
	113	311	ret	to section c).,.

j) ELECTROMAGNET ACTIVATION (GRASP).

HI	LO	INSTR.BYTE	MNEM.CODE	COMMENTS
034	150	000	nop	
	151	076	mvi, A	
	152	002	<002>	data.
	153	323	out	
	154	002	port 2	
	155	315	call	subroutine (break).
	156	200	lo	
	157	037	hi	
034	160	000	nop	
	161	311	ret	to section c).

k) ELECTROMAGNET DEACTIVATION (RELEASE).

HI	LO	INSTR.BYTE	MNEM.CODE	COMMENTS.
035	150	000	nop	
	151	076	mvi, A	
	152	001	<001>	data.
	153	323	out	
	154	002	port 2	
	155	315	call	subroutine (break).
	156	200	lo	

	157	037	hi	
035	160	000	nop	
	161	311	ret	to section c).

DESIGN AND DEVELOPMENT
OF A MICROCOMPUTER CONTROLLED
SMALL ARTICULATED ROBOT

by

MAHMOOD HASAN

M. S., Leningrad Polytechnic Institute, 1979

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1983

ABSTRACT

The purpose of this research is to convert a rotary table with three degrees of freedom to a microrobot with four degrees of freedom and a grasp-release mechanism. The conversion process was in three phases. The first was to design and fabricate the mechanical components, followed by design and fabrication of hardware. The final phase involved design of sofware to demonstrate micro-control of the robot. The system and the program are tested by a sample program that enables the robot arm to handle end loads under the control of the microcomputer.