DESIGN OF A DEXTEROUS MECHANICAL HAND

By

TOMMY J. METZNER

Bachelor of Science in Mechanical Engineering

Oklahoma State University

Stillwater, Oklahoma

1984

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December 1987

DESIGN OF A DEXTEROUS

MECHANICAL HAND

Thesis Approved:

Thesis Adviser

Dean of the Graduate College

PREFACE

An articulated hand model that can mimick human hand motion was designed, fabricated, and controlled. The hand consisted of three fingers. Two fingers were similar to the index finger of the human hand and the other finger is similar to the human thumb in terms of its relation to the palm. The hand has the capability to grasp objects with different shapes and physical properties. These objects include the following: egg, styrofoam cup, ball, and an aluminum can. The model is the first of its kind in that each finger is designed as a three-link, 3R robot with a voidless workspace.

The model's fingers are moved by nine linear actuators and controlled with the the aid of an A/D board connected to a IBM-AT computer. The hand is portable which allows for it to be attached to the Unimate PUMA 700 robot.

I wish to express my sincere gratitude to all the people who assisted me in this work and during my stay Oklahoma State University. In particular, I am indebted to my major adviser, Dr. A. H. Soni for his intelligent guidance, invaluable help, and encouragement.

I am also thankful to committee member Dr. Gary Young for his cooperation during the couse of this work. I am thankful for the ideas, suggestions, and assistance of

Ken Fowler. I would also like to thank my brother, Ron Metzner, for his suggestions and advice on the electronics used in this project.

I am grateful for the cooperation given by Steve Harris for the use of the MAE Lab.

Special thanks is due to the Fluid Power Research Center for the use of their facilities and the School of Mechanical and Aerospace Engineering for the financial support of this work.

Finally, I am thankful to my mother and father for their continued support and encouragement during my stay at Oklahoma State University and also I am grateful to my father for his assistance on the mechanical drawings used throughout this thesis.

TABLE OF CONTENTS

APPENDIXES Page

LIST OF FIGURES

Figure | Page

CHAPTER I

INTRODUCTION

The grasping capabilities of the end-effector on an industrial robot limits its application in the industrial environment. Since the human hand is capable of numerous grasps, an end-effector modeled after a human hand would be very desireable.

Numerous studies have been made on the kinematics and forces associated with articulated mechanical hands (1). The University of Utah and Massachusetts Institute of Technology are developing, probably the most proclaimed, a tendon operated, multiple degrees of freedom mechanical hand (2). This hand closely resembles the human hand in that it has an anthropomorphic geometry and its dexterity has reached the capacity of a human. Hitachi has developed a robot hand which employs the use of shape memory alloy (SMA) to control the actuation of the fingers and wrist. This method of actuation produced a highly efficient robot hand with a very lightweight mode of actuation (3).

The main goal of this research was to develop, construct, and control a dexterous mechanical hand that would exhibit prehension of an object. The complete system is shown in Figure 1.

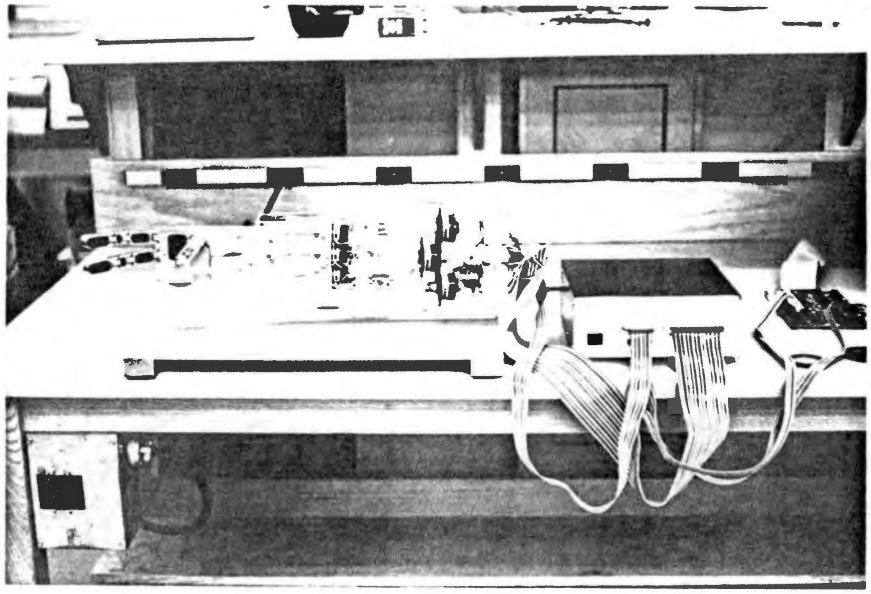


Figure 1. Designed Mechanical Hand

There are only two basic categories of grip used in manufacturing work: the "three-fingered" grip and the "wrap-around" grip (4). Since only three fingers are required to facilitate these grips, the designed hand has only three fingers. Each finger is composed of three binary links modeled from the previous results using the concept of a 3R robot with a voidless workspace (5). The fingers are situated on the palm very similar to the fingers on the human hand. The fingers are actuated by a system of pulleys and cables and nine linear actuators as shown in Figure 2. Shown in Figure 3 is the entire mechanical hand mounted on a base which can be oriented in numerous positions. The motor control circuits (A), sensor circuit (B), and power supply (C) necessary for the control and actuation of the hand are shown in Figure 3. Ribbon cable is used to connnect the hand and motors and sensors to the computer and control circuits.

The control algorithm is menu driven which allows for easy user interaction. The algorithm is coded in C since its execution speed is sufficient for the finger actuation.

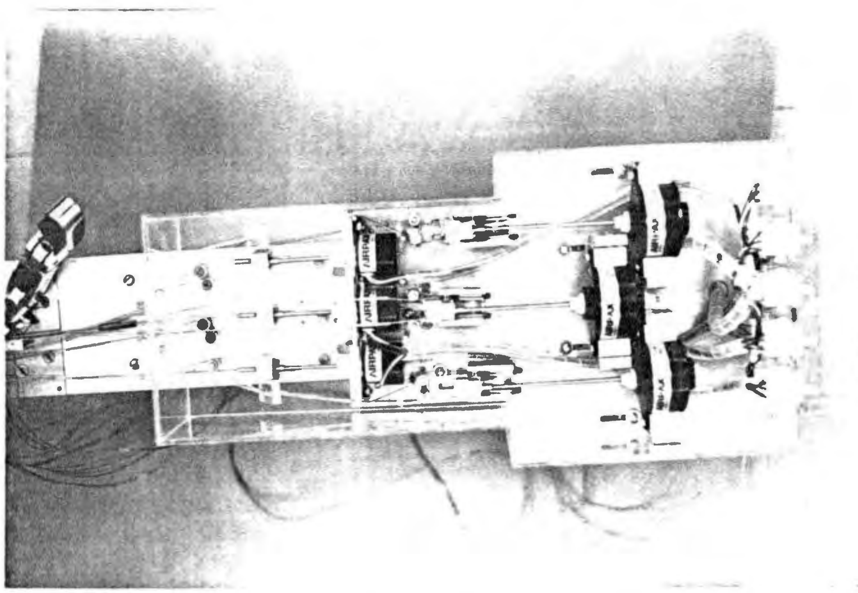


Figure 2. Pulley and Cable System

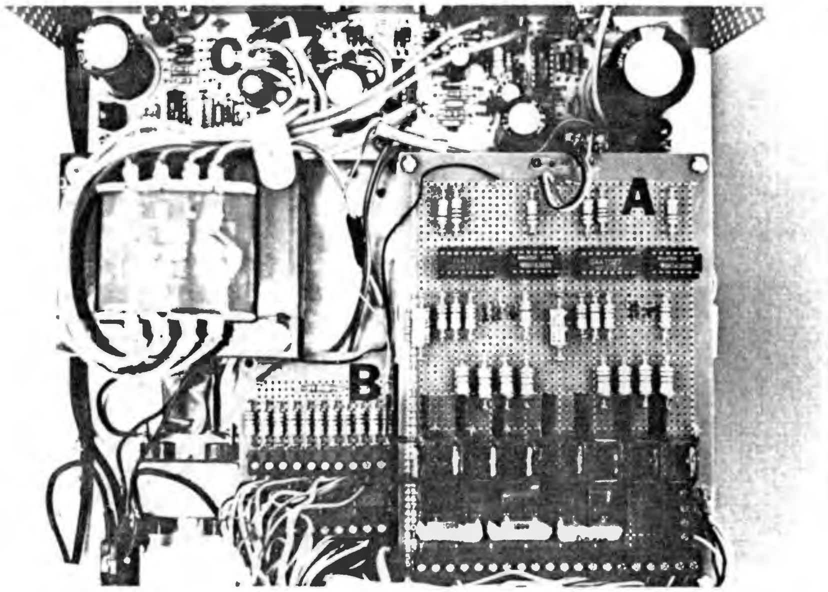


Figure 3. Electronic Hardware

CHAPTER II

DESIGN AND CONSTRUCTION

Nine linear actuators are used to control the movements of the finger links; in addition, force feedback is used to indicate stability of grasps on objects. Refer to the appropriate sections for detailed description of the control electronics and software.

Finger Links

The fingers are designed so that for the least joint motion a voidless workspace will exist. Previous research indicates that the link lengths a, b, and c ratios which produce this voidless workspace are 0.4, 0.2, and 0.4 respectively (5). The graph of the workspace envelop in a two dimensional plane is shown in Figure 5. The actual dimensional characteristics for each finger link are given in Appendixes A and B.

The finger links are constructed from 7075-T6 aluminum because of its high strength-to-weight ratio and machinability (6). Other aluminums such as 2024 and 6061 were considered but due to the physical properties of these aluminums, they were not used except where strength was not an important factor. The finger links form a revolute pair by the use of a ball bearing and a pin. The decision to use ball bearings was based on the fact that low radial forces will be experienced and a reduction in frictional components is highly desireable. The finger links are machined thin for weight reduction. Because of the slow movement and low weight of the finger links, inertial effects were not considered during this study.

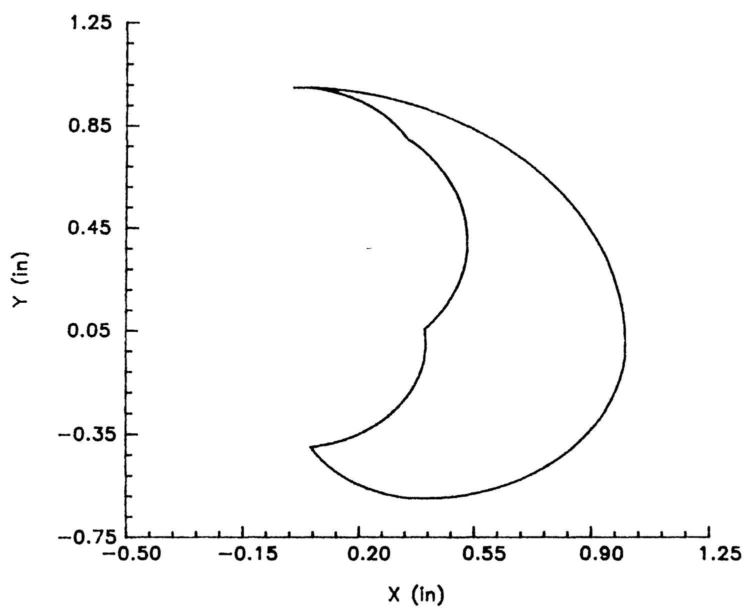


Figure 4. Workspace for a 3R Robot with Link Ratios of 0.4, 0.2, and 0.4.

Palm Configuration

The palm is constructed of the same material as the finger links. The main function of the palm is to provide a stable surface for grasping to take place and to anchor the fingers. The configuration in which the fingers are anchored is important in such that the positions will limit the dexterity of the hand.

The configuration is based upon a geometrical intersection of three workspaces and in part by the anthropomorphic geometry of the human hand. The size of the hand is determined once again by the anthropomorphic geometry in addition to the fact that larger hands are more efficient in the industrial environment (7). The final configuration is shown in Figure 5.

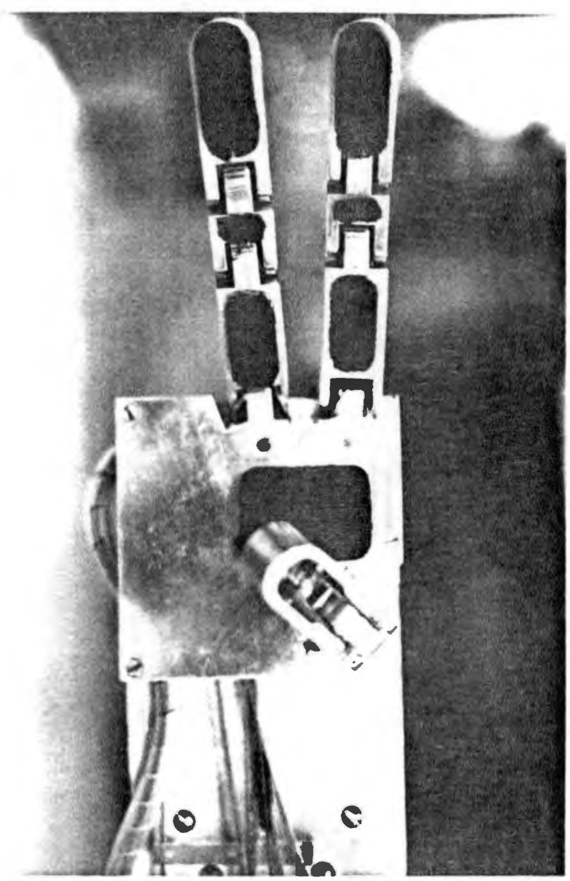


Figure 5. Palm Configuration

Actuation System

The actuation system is the most important element which determines the mechanical performance of the mechanical hand. The actuation of the fingers is achieved by the use of a pulley-cable system. Miniature pulleys are installed within the finger joints and links. A special cable made of an aramid fiber is used because of its unique flexiblity characteristics and strength-to-weight ratio. The pulley inside each link is offset from the center of the link joint as shown in Figure 6. A moment is produce about the joint when tension is applied to the cable. The direction of tension corresponds to the direction of finger motion. With this simple system, it is quite easy to control the movement of the fingers.

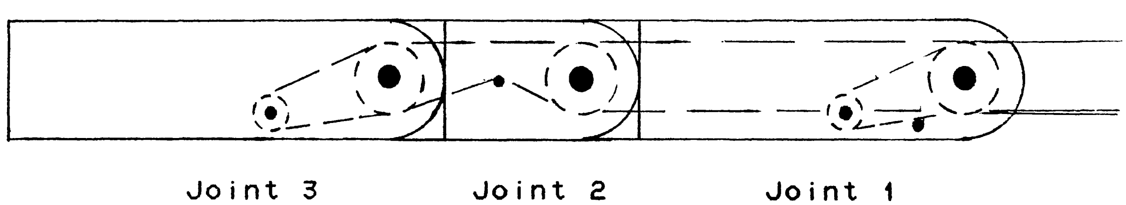


Figure 6. Pulley Arrangement

The fingers' cables are linked to the actuator by a system of cable routings on the palm and on the base plate of the mechanical hand. This routing is shown in Figures 7 and 8.

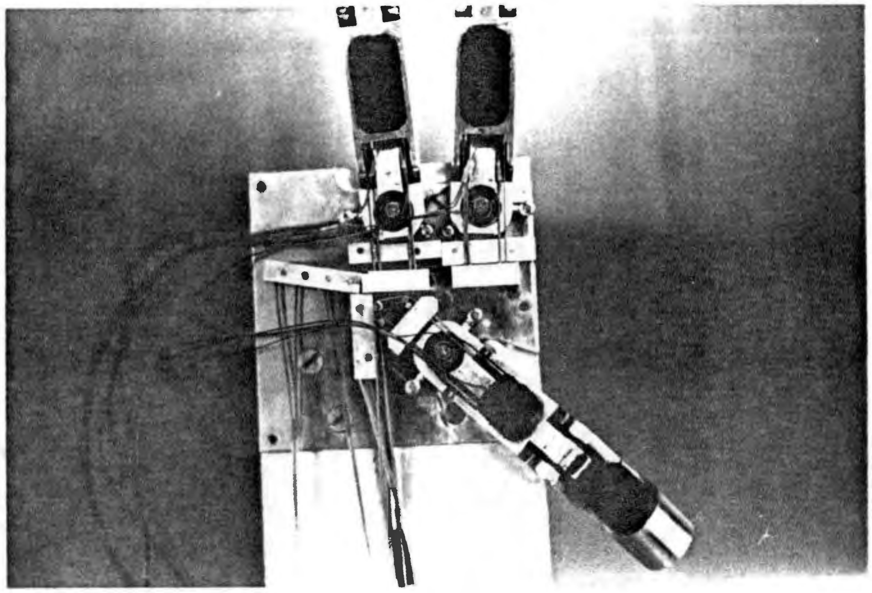


Figure 7. Cable Routing-Palm

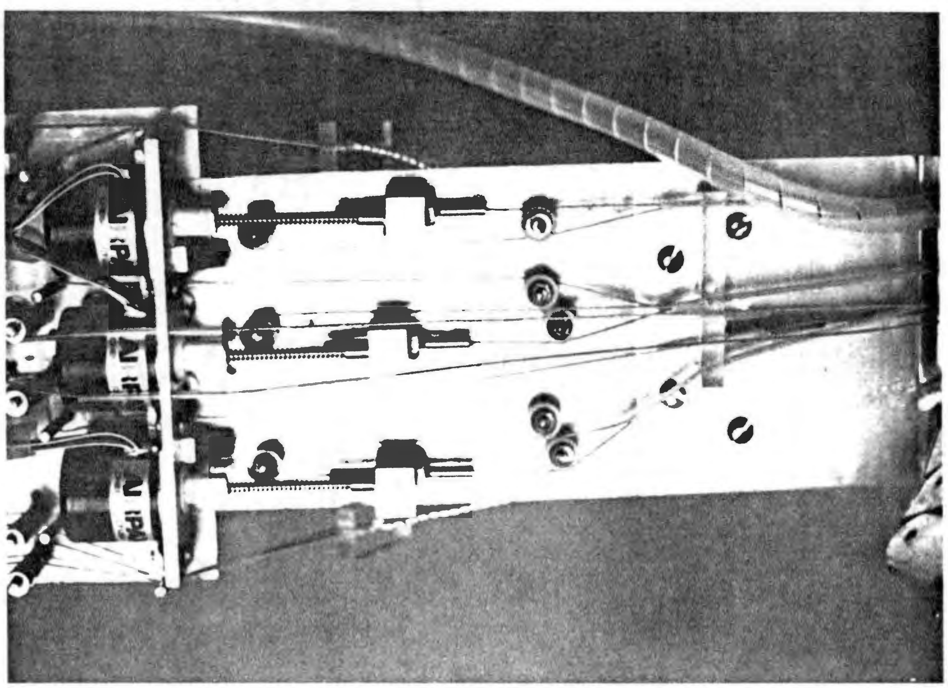


Figure 8. Cable Routing-Base Plate

The method of providing adequate tension to the cables is accomplished through the use of linear actuators. The linear actuator is basically a stepping motor with a lag screw attached to it thus changing rotational motion to linear motion.

Before the linear actuators were selected, the maximum weight that the hand could successfully grasp was decided to be 3 lbf. After this weight is known it is necessary to calculate the required actuating forces which would provide acceptable grasp forces.

A cubical object weighing 3 lbf would require approximately 1.11 lbf of grasping force normal to the object. This is based on the following calculations (Refer to Figure 9):

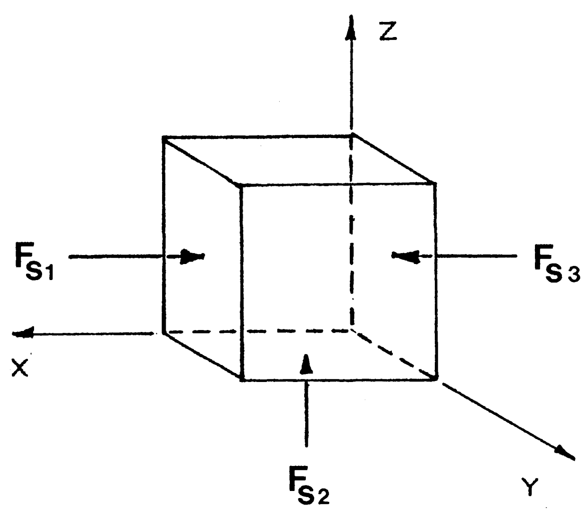


Figure 9. Forces Acting on a Simple Cube

For a stable grasp $\Sigma Fx = \Sigma Fy = \Sigma Fz = ma = 0$

If equilibrium exists $Fs1 = Fs2 = Fs3$

Since $\Sigma F = 0$ $3.0 * Fs - mg = 0$

$3 * \mu N = 3$ lbf

For Rubber on Steel $\mu s = 0.9$

Therefore $2.7 * N = 3.0$

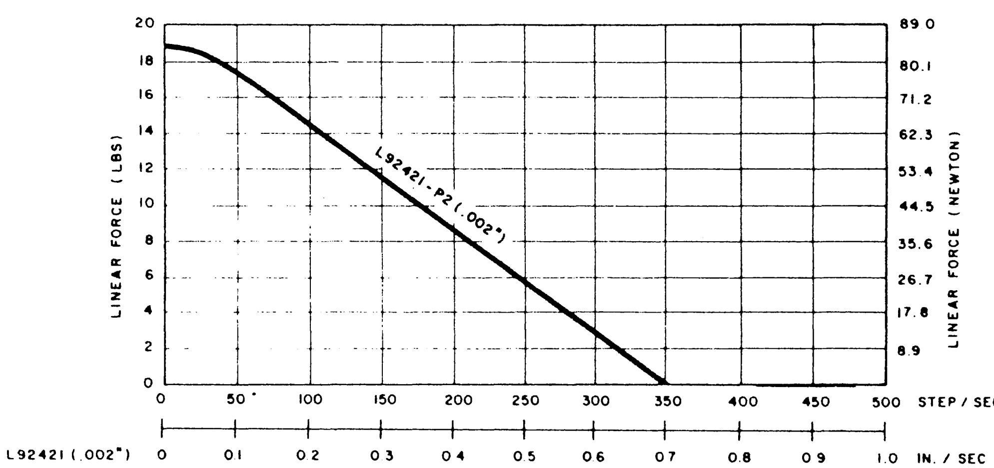
$N = 1.11$ lbf

Where N is the normal force necessary to maintain a stable grasp.

Next, with the given pulley offset, it is necessary to determine the required tension produced in the cable to provide this calculated force. With an offset of 0.125 inches, simple calculations are made for a point 0.625 inches along the length of the finger from the joint. A moment of 0.694 in-lbf is required to produce a normal force of 1.11 lbf at this point. This would require an actuation force of 5.55 lbf.

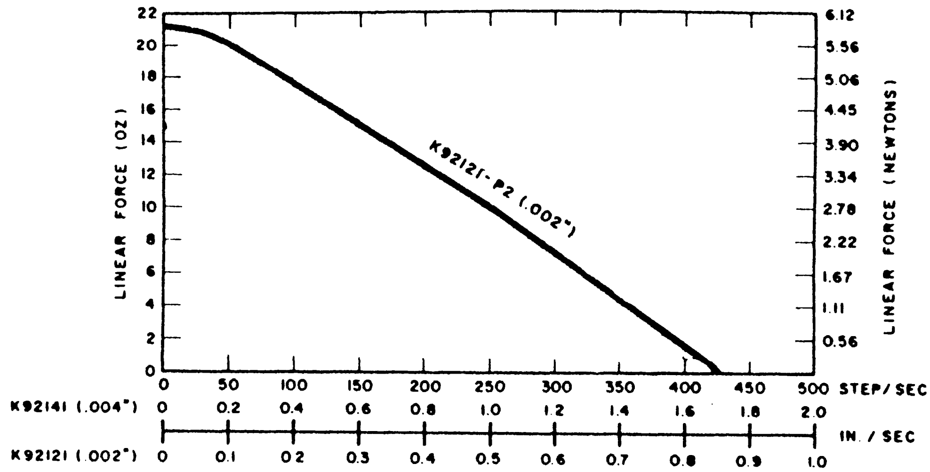
With this force in mind, a selection of the mode of actuation is narrowed down substantially. A mode of actuation that is small and lightweight is highly desirable. After an extensive product search, a linear actuator was found to be the most appropriate mode of actuation because of its cost and force output. However, one concern is the size and weight of the linear actuator. Two different actuators were found that would produce the desired linear force and were small and lightweight. One actuator, Airpax K-92121, will produce an approximate force of 2.5 lbf and will be used for the rotation of the fingers. The other actuator, Airpax K-92421, will produce an approximate force of 4.7 lbf and will be used for the finger link movements.

Obviously, an additional mechanism is required to produce an actuation force of 5.55 lbf. Using a simple lever mechanism, actuation forces up to 15.3 lbf can be obtained and at the same time serve as the link-up between the cable and actuator. Since the tension in the cable is a constant force as the linear actuator moves, the force will remain constant once the object makes point contact with the finger's surface. However, the force of the finger will vary according to the linear rate of the actuator shaft. The force output and linear rate of each actuator are shown in Figures 10 and 11. The actuators and mechanism system are shown in Appendix G.



Source: AMSI Corporation, 1986 Catalog.

Figure 10. Typical Linear Force vs. Linear Rate at 20 C (L92421)



Source: AMSI Corporation, 1986 Catalog.

Figure 11. Typical Linear Force vs. Linear Rate at 20 C (L92121)

Sensors

For stable grasping, defined forces must be transferred from the actuator by the fingers to the object being grasped. This can only be accomplished by some form of sensor and in this case a tactile sensor is used. Using conductive foam, a two fold purpose is achieved. One purpose, the foam acts as an artificial skin by conforming to an object's surface when grasped and secondly it gives the mechanical hand a sense of touch when the foam is properly utilized by acting as a force sensor.

The conductive foam has an approximate resistance value of 120000 ohms per unit area for a thickness of one-eighth of an inch. When the foam is compressed the resistance changes, therefore by constructing a simple circuit that allows for the application of the voltage divider rule and using the foam as a resistor, the voltage output will change according to the applied force or pressure to the foam. By applying known loads to a specific sensor, the sensor is calibrated for a range of loads. The conductive foam is not flawless but it exhibits properties which allow it to be a very useful tactile sensor. It has some non-linearites and hystersis when large loads are applied to the foam or a load is applied for long periods of time. A typical curve illustrating the force versus resistance is shown in Figure 12 (9). Appendix I shows a schematic drawing of the sensor circuit and gives details on the construction of the foam sensor.

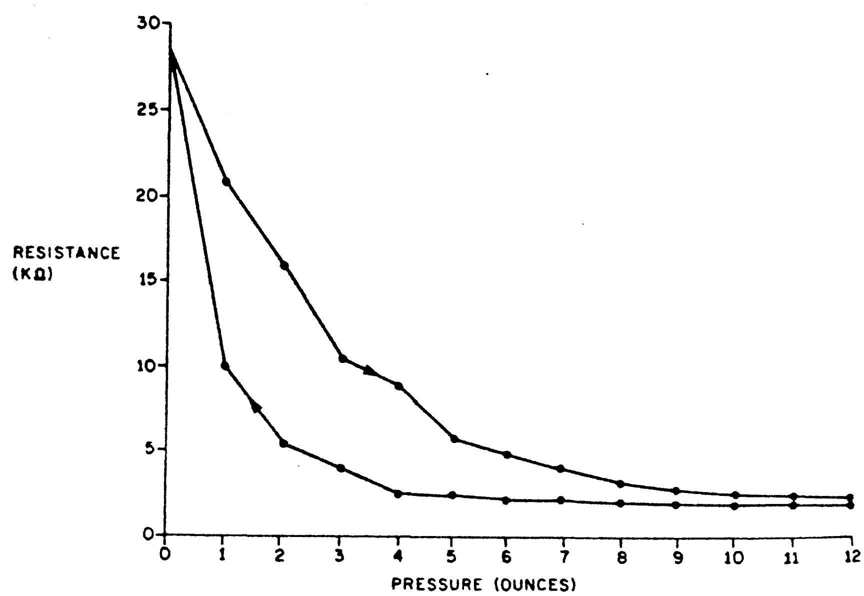


Figure 12. Typical Resistance Change in Antistatic Foam

Twelve sensors are strategically placed on the finger links and palm as shown in Figure 13. This placement of sensors is based on the fact that for a stable grasp by a finger, specific contact points must be known.

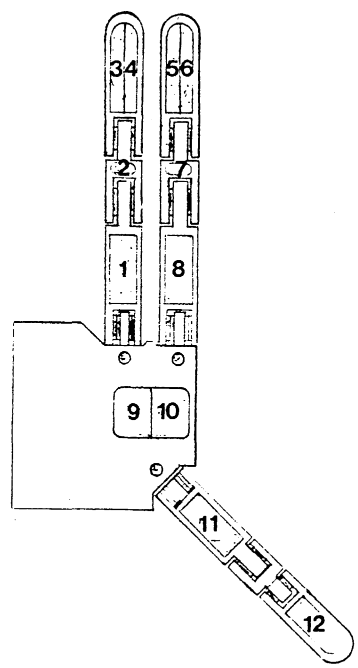


Figure 13. Sensor Locations

Demonstration Platform

A platform is made that is used only for demonstrating the dexterous mechanical hand. The platform allows for the palm to be rotated by 90 degrees from side to side and tilted 45 degrees forwards or backwards. The platform's dimensions are given in Appendix G and the platform is shown in Figure 14.

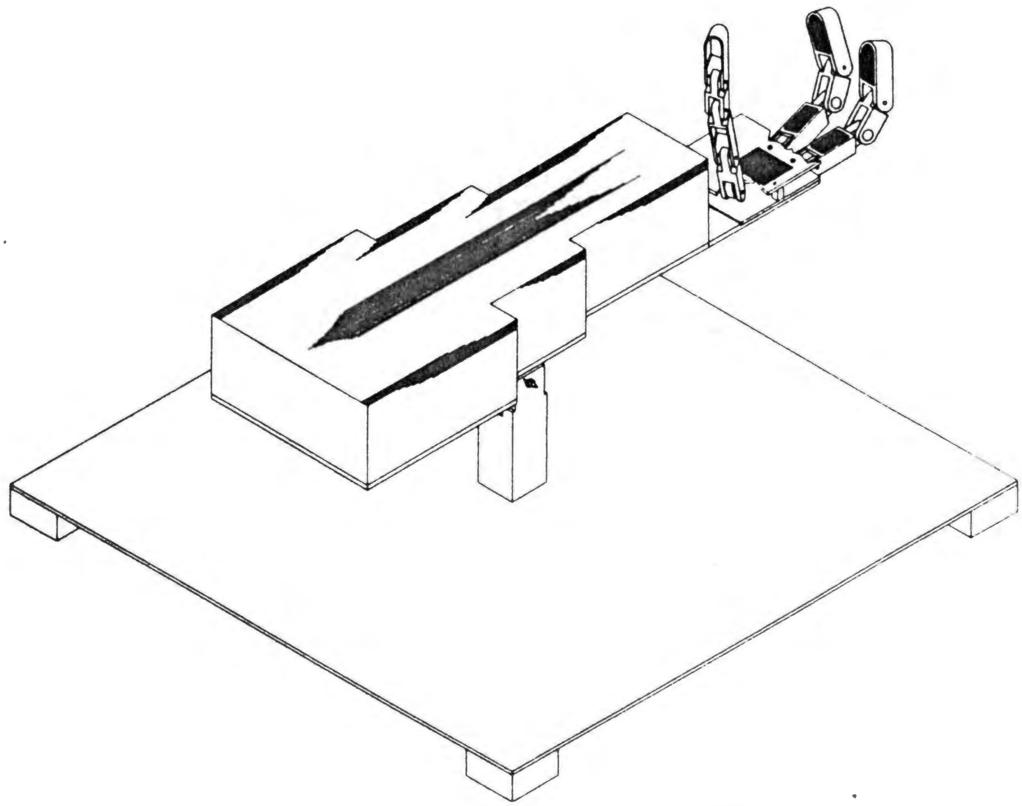


Figure 14. Demonstration Platform

CHAPTER III

LINEAR ACTUATOR CONTROL SYSTEM

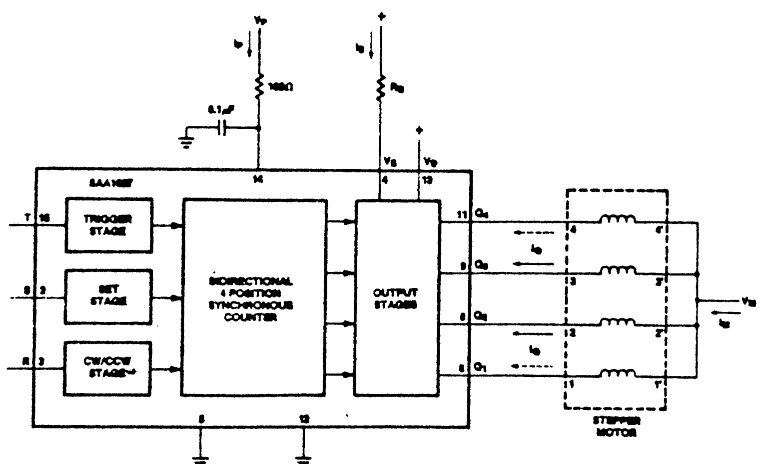
Achieving similar dexterity of that of the human hand in terms of moving the fingers with the speed and the force necessary to successfully grasp an object is a very difficult task. The actuator control circuit and the sensor circuit play an important role in the control of the mechanical hand.

Control Circuit

The nine linear actuators are controlled by the use of the two Signetics SAA 1027 Stepper Motor Controller IC and a set of switching relays (Refer to Appendix J).

The nine linear actuators are four phase motors. By controlling the state of two phase pairs in terms of "on" or "off", the actuator is driven by the IC driver. Referring to Figure 15, it readily can be seen that controlling the actuators is an easy task when utilizing a computer in conjunction with an analog-to-digital converter. Using the A/D screw terminal, communication between the controller and computer is established (10). By setting pin (2) high, 1, and setting pin (3) low, 0, the direction of the linear motion is set (11). By causing pin (15) to go high, the actuator will step each time it is sent a pulse via computer, hence, the speed of the screw rate is easily controlled in a computer program. In the case of the K-92421 the maximum screw rate is 0.50 in/sec and the K-92121 has a maximum screw rate of 0.85 in/sec.

Since the K-92121 is a 12 volt motor and the K-92421 is a 5 volt motor, two independent stepper motor controllers are constructed. Bits 0 through 3 on digital I/O port 1 are used in controlling the speed and direction of the motors (Refer to Figure 15).



Source: Linear Data and Applications Manual Volume I Signetics Corporation, (1985).

Figure 15. Controller Circuit Layout

The selection of the motors is accomplished through the digital I/O port 0. Bits 1 through 6 are used in this case to activate the relays. By placing relays in the motors' windings the circuit is broken; hence, each motor can be turned "on" or "off" at will. With an additional power supply, it is possible to control two or more motors simultaneously with this type of motor selection technique. The screw terminal connections are shown in Appendix K.

Control System Modeling

Two types of control systems are used in controlling the mechanical hand. An software open-loop control system (Figure 16) is incorporated during the use of the teach pendent mode and a software closed-loop system (Figure 17) is used during the use of the computer mode (12). The teach pendent and computer modes will be discussed at a later time.

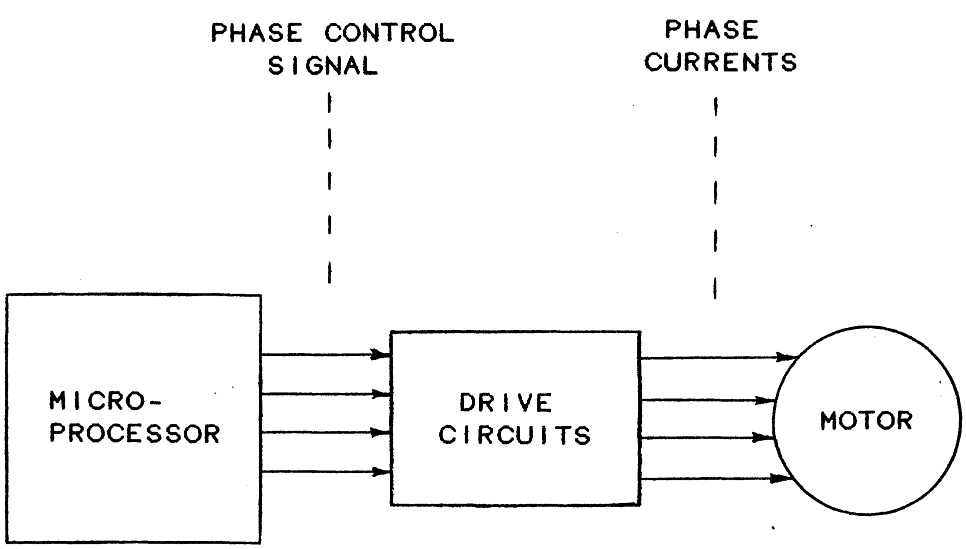


Figure 16. Open-Loop Control System

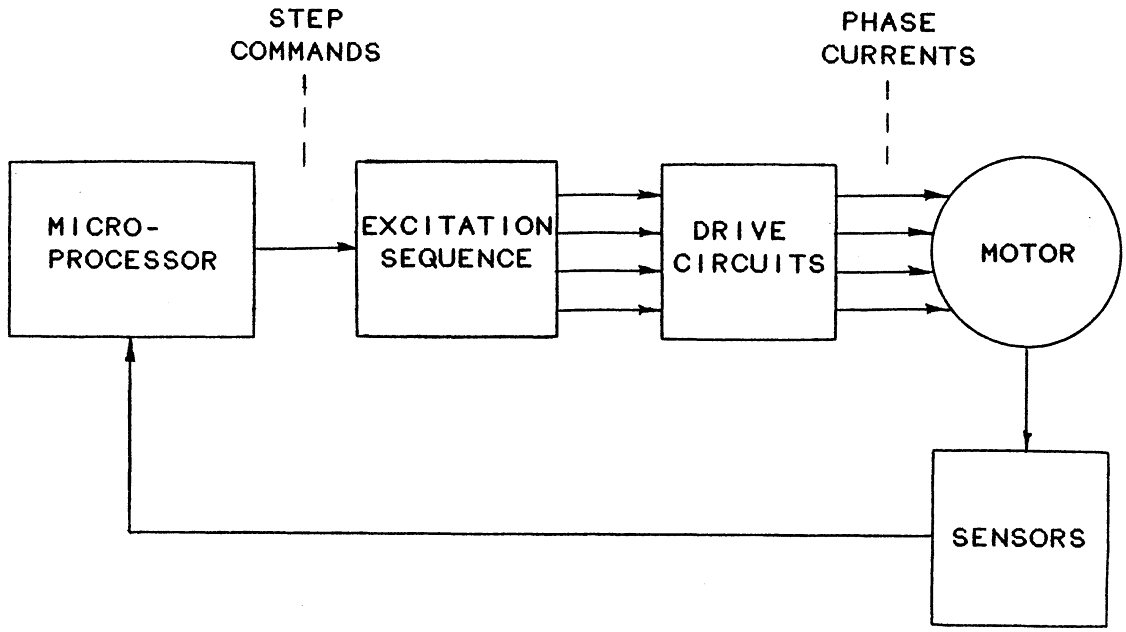


Figure 17. Closed-Loop Control System

The software for the open-loop system is designed such that the motors are controlled by a single pulse each time an arrow key is pressed. The user has total control of the movement of each finger. The sensors are not used here except for referencing the force being exerted on the object.

The closed-loop system's feedback element is primarily the force sensor. Ideally, a grasped object is stable when symmetrically placed contact points have equal forces exerted on them. Using this theory, the software is developed so that the computer will search until the forces exerted on the object are equal according to the relative position of the finger joint contact point.

CHAPTER IV

CONTROL ALGORITHM

The control algorithm is written in "C" and is included in Appendix L. The algorithm is written in a modular format with the main program being subdivided into two segments, demonstration and teach pendent modes. The remainder of the program is composed of subroutines. This type of modular programming allowed for easy debugging.

Demonstration Mode

This mode is designed specifically for demonstration purposes. Four files were created using the teach pendent mode. The four objects grasped are an egg, styrofoam cup, aluminum can, and a ball. These objects were chosen because of their different physical properties such as weight, size, and fragility. The range of grasping capabilities are exhibited through the selection of these objects.

Another feature was installed in this mode that allows the user to replay the grasping motion of any object that has been previously grasped and its sequence of motions stored on file.

Teach Pendent Mode

The teach pendent mode allows the user to place an object on the palm and control the grasping manually or by computer. The resulting movements are saved on file under any given name so that the motions can be replayed if desired.

For manual control, the fingers are actuated one at a time by using the arrow keys on the keyboard. The direction of finger movement is indicated by the direction of the arrow. After a desired position has been obtained, the user is required to enter that position if it is to be used for later reference or playback. Feedback from the sensors is not utilized in this particular teach pendent mode.

The computer teach pendent mode is controlled by a closed-loop system. The user specifies a desired force applied to an object and the computer, using the feedback from the sensors, initiates a sequence of finger movements that will grasp the object. The force entered must remain between 0.0 and 1.0 lbf.

The algorithm is set-up so that the sensors are continually scanned in a predetermined sequence. This sequence was determined by experimentally using different sets of sequences. The sequence that gave the appropriate information so that a stable grasped could be obtained quickly was used.

The user also can control the speed of actuation. These speeds begin at 0%, 0.0 in/sec, to 100%, 0.85 in/sec, for moving the fingers side to side and 0.0 in/sec to 0.5 in/sec for opening and closing the fingers. Medium speeds are suggested for optimum performance.

CHAPTER V

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

There is much to learn from the prototype mechanical hand. The configuration of the finger locations with respect to the palm is a major consideration in the design of the hand. Different locations and the geometric design of the finger links change the grasping capabilities and dexterity of the mechanical hand. All configurations considered were graphically determined by hand. A computer simulation of a three-fingered mechanical hand is highly recommended in order to obtain the optimum finger-palm configuration for a given range of manipulative tasks.

Another important feature which will limit the dexterity of the mechanical hand is the accessible region of the individual fingers. In this case, the finger lengths were designed such that for specific joint rotations there existed a voidless workspace. With this characteristic the designed mechanical fingers could actually reach areas on the palm not accessible to a by fingers on a human hand of comparable size.

There are many modes of actuation available but systems that can deliver the appropriate forces, actuate the fingers at substantial speeds, and provide a small and lightweight package are difficult to obtain and are often costly. The tendon operated actuation system provides a lightweight and efficient mode for finger movement. All control systems and drive components are located away from the actual mechanical hand which allows for various types of drive components and arrangement. The linear actuators selected provide an efficient means of moving the finger links and also deliver the required forces; however, due to unavoidable frictional components the speed is slower that desired. Servo motors could serve as an alternate choice for actuation but the control system would be much more complex and not as cost effective. Another mode of actuation considered was the use of shape memory alloy (SMA) (13). SMA was experimented with but found that an elaborate actuating and cooling system would be required. However, SMA does have excellent potential for this type of application assuming the appropriate research is formulated.

Another area which affects the overall performance of the mechanical hand is the type of sensor used on the fingers and palm. Although crude, the conductive foam has proven to be an excellent sensor for touch.

A much more sophisticated means of using a sensor, would be using piezo crystals to sense touch and detect objects being grasped. Arranging the piezo crystals into an array throughout the hand, the location of the signal from the crystal can be used to fill an array of 1's and 0's; hence a two dimensional picture can be generated similar to that of a computer vision system. This type of information would be excellent for artificial intelligence applications.

Optical sensors could be implemented as another source for touch and placement information. Potentiometers added to the hand would allow for a low cost and very accurate means of positioning the finger at selected points on an object. Addition of these potentiometers would require a mathematical model in terms of a kinematic analysis of the mechanical hand. Furthermore, for a much more sophisticated model, inertial effects and system damping should be fully developed into the mathematical model.

The task of designing an excellent dexterous mechanical hand requires a team effort. This research area comprises many areas of complexity and disciplines. Major areas which require research and development are configuration design, actuation system, control system, and sensors. Obviously, each of these areas are comprised of sub-groups and only can been dealt with by qualified persons.

The list for improvements and sophistication of the dexterous mechanical hand is long but the hand developed will manipulate a number of different objects successfully (Refer to Figures 18 and 19). The main scope of this project was to develop a working dexterous mechanical hand that can be used as a test bed for future research in this exciting area that has much potiential in the manufacturing environment and the biomedical field.

Figure 18. Grasp for a Can

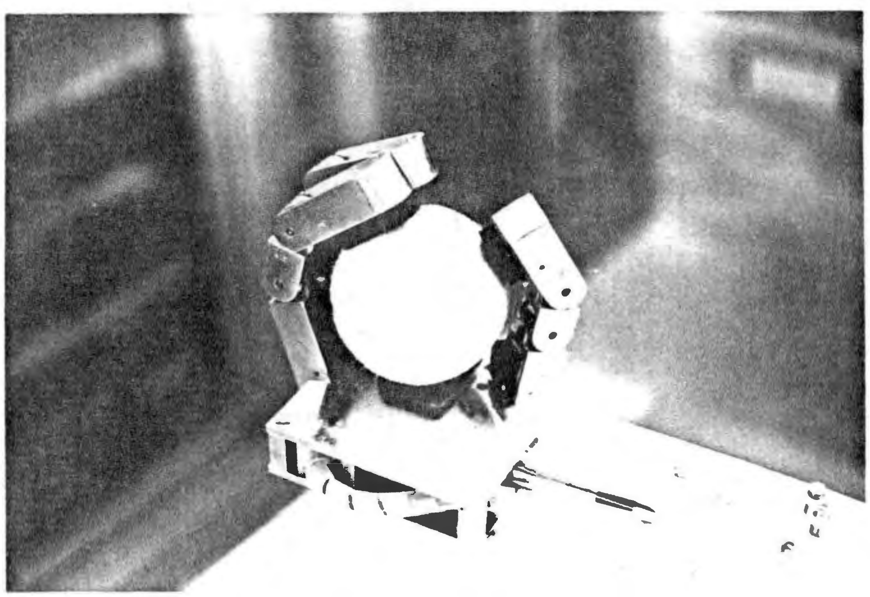


Figure 19. Grasp for a Ball

SELECTED BIBLIOGRAPHY

1. Salisbury, J. K. "Kinematic and Force Analysis of Articulated Mechanical Hands." ASME Transactions, Journal of Mechanisms, Transmissions, and Automation in Design, 105(1983), pp. 35-42.

2. Jacobsen, S. C., Wood, J. E., Knutti, D. F., Biggers, K. B. "The Utah/MIT Dexterous Hand: Work in Progress", SDF/MIT 1st International Symposium of Robotics Research, Bretton Woods, NH 1983.

3. "Hitachi's Robot Hand." Robotics Age, July, 1984, pp. 18-20.

4. Wright, Paul K. and Mark R. Cutkosky. "Design of Grippers." Mechanical Design of Robots, Ed. Eugene I. Rivin. New York: McGraw-Hill, 1988 1988, p. 98.

5. Laughlin, Gary Lynn. "Position Robot with One Degree of Redundancy." (Unpub. M.S. thesis, Oklahoma State University, 1986).

6. "Materials." Society of Automotive Engineers Handbook, Vol. 2 (1986), pp. 10.01-10.39.

7. Salvendy, Guariel. "Hand Size and Assembly Performance." American Institute of Industrial Engineers Transactions, 3, 1(March 1971), pp. 32-36.

8. Holzmann, W. and J. M. McCarthy. "Computing the Friction Forces Associated with a Three Fingered Grasp." Proceedings of the 1985 IEEE Conference on Robotics and Automation. St. Louis, pp. 594-600

9. Mallin, Jay. "A Simple Sense of Touch for Robotic Fingers." Robotics Age, November/December, 1983, pp. 24-27.

10. User's Manual for DT2801 Series Single Board Analog and Digital I/O Systems. Malborough, Massachusetts: Data Translation, Inc., 1985.

11. Linear Data and Applications Manual Volume I. Sunnyvale, California: Signetics Corporation, 1985.

12. Acarnley, P. P. Stepping Motors: A Guide to Modern Theory and Practice. London and New York: The Institute of Electrical Engineers, 1982.

13. "Robot Hand with Shape Memory Musculature." IBM Technical Disclosure Bulletin, 28, 1(June 1985), pp. 302-303.

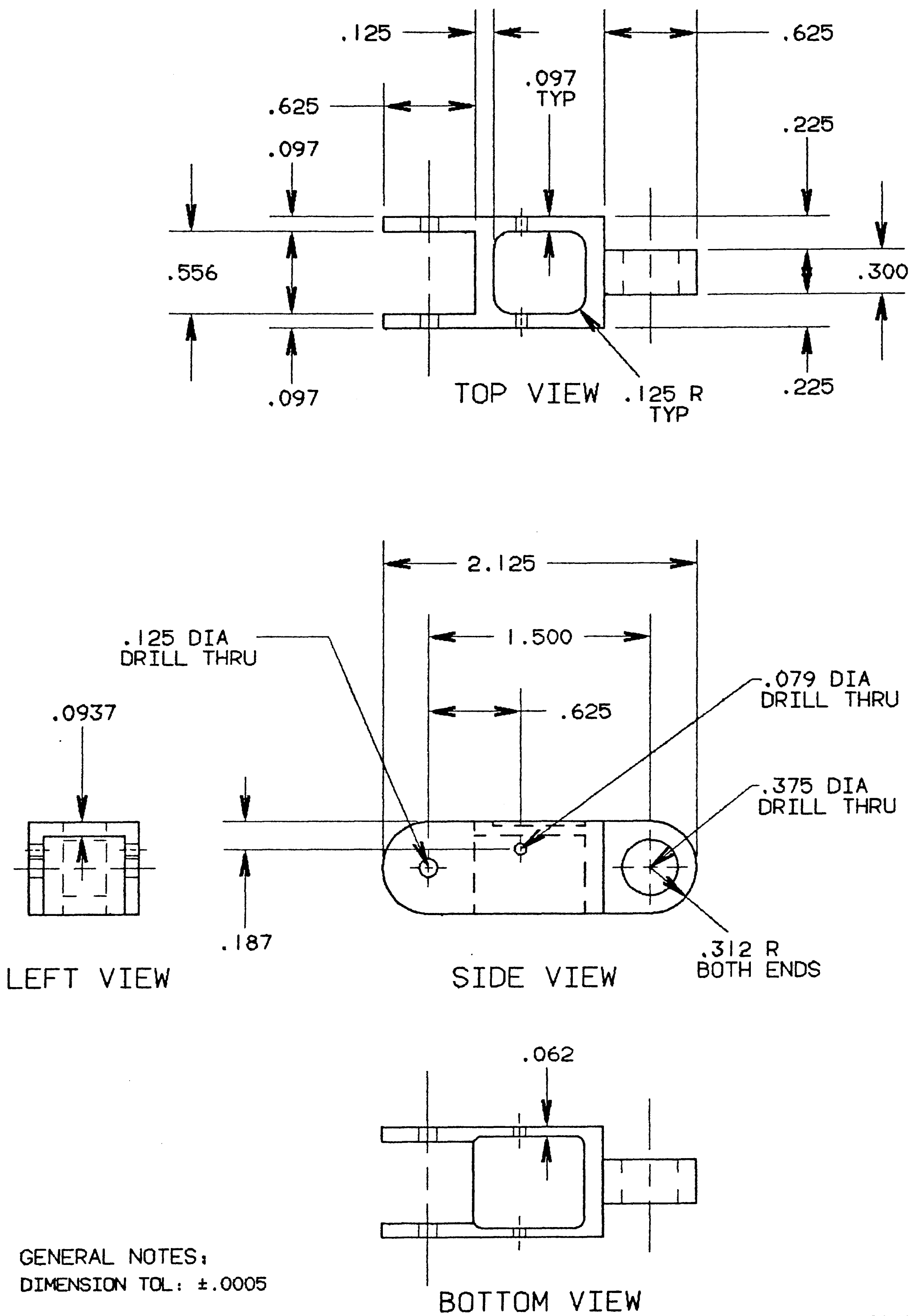
APPENDIXES

APPENDIX A

DETAILED THUMB LINK DRAWINGS

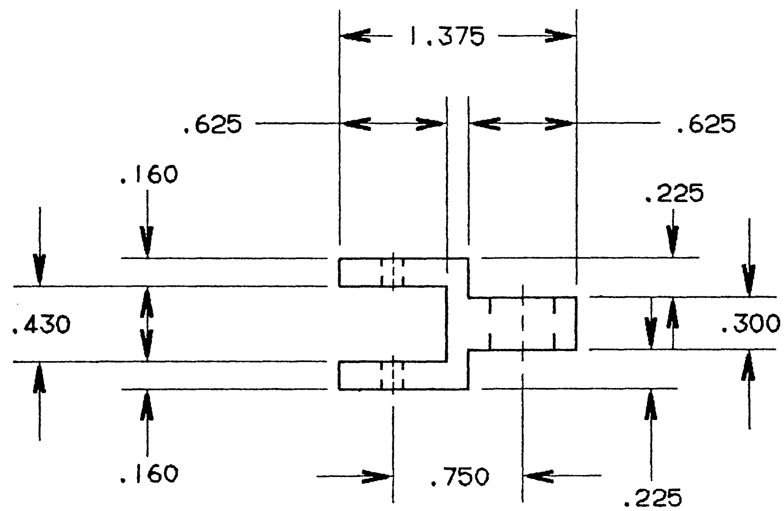
AND ASSEMBLY PHOTOGRAPHS

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	LINK, THUMB, 1ST	METZNER	DATE: 11-86

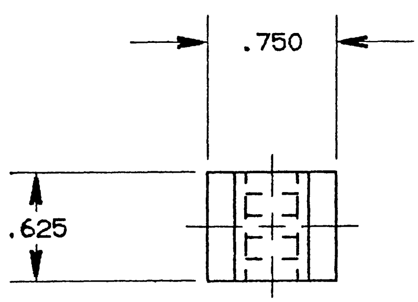


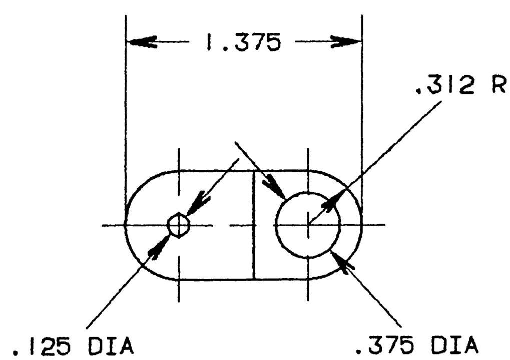
GENERAL NOTES:
DIMENSION TOL: ±.0005

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	LINK, THUMB, 2ND	METZNER	DATE: 11-86



TOP VIEW



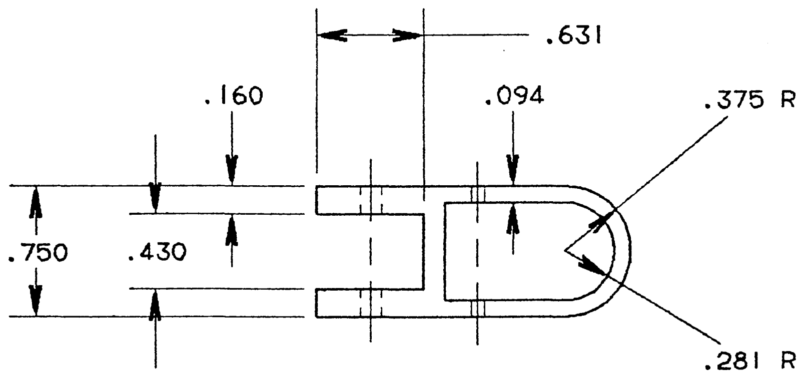


LEFT VIEW

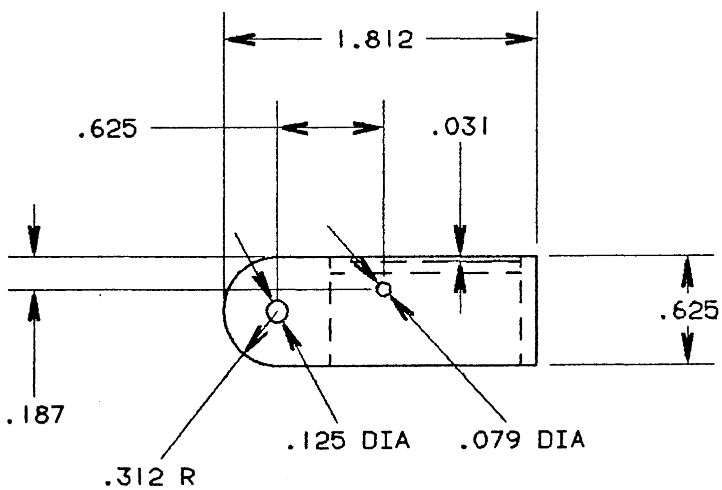
SIDE VIEW

GENERAL NOTES:
DIMENSION TOL: ±.0005

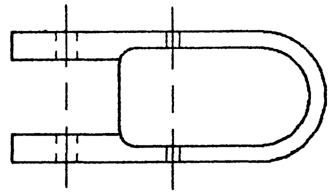
MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	LINK, THUMB, 3RD	METZNER	DATE: 11-86



TOP VIEW

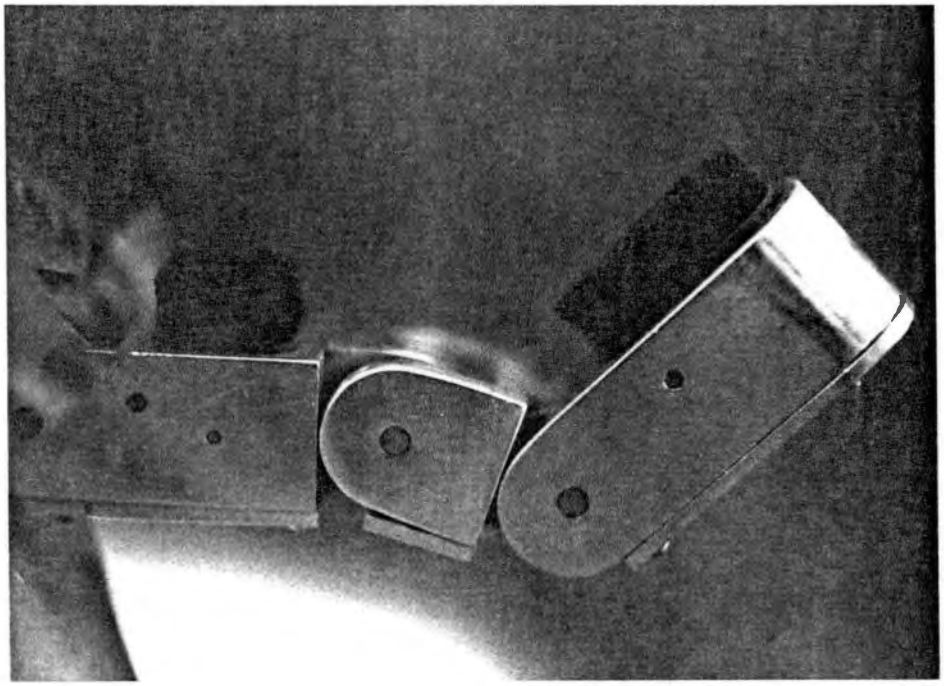


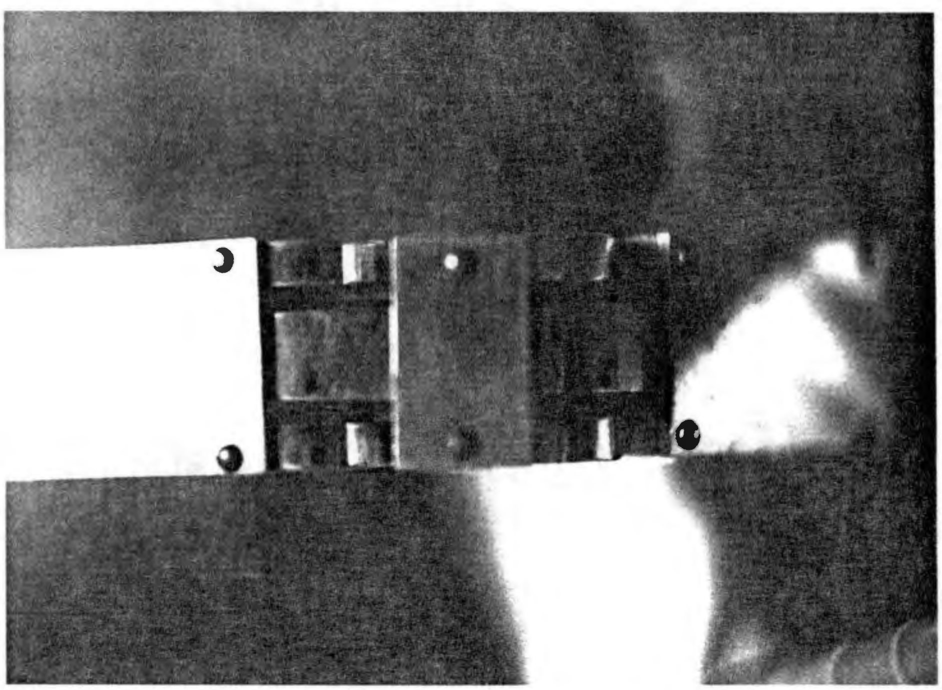
SIDE VIEW



GENERAL NOTES:
DIMENSION TOL: ±.0005

BOTTOM VIEW

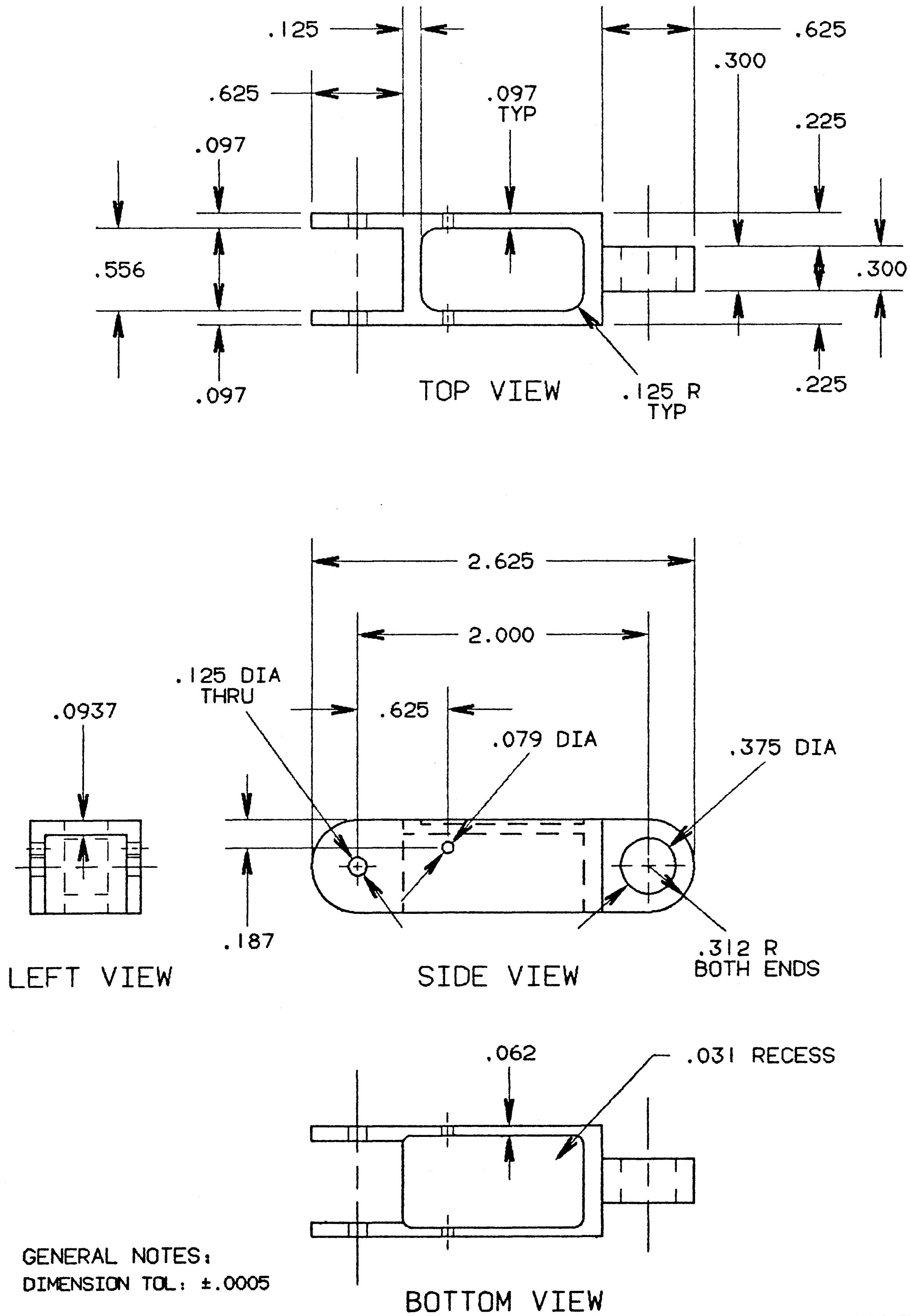




APPENDIX B

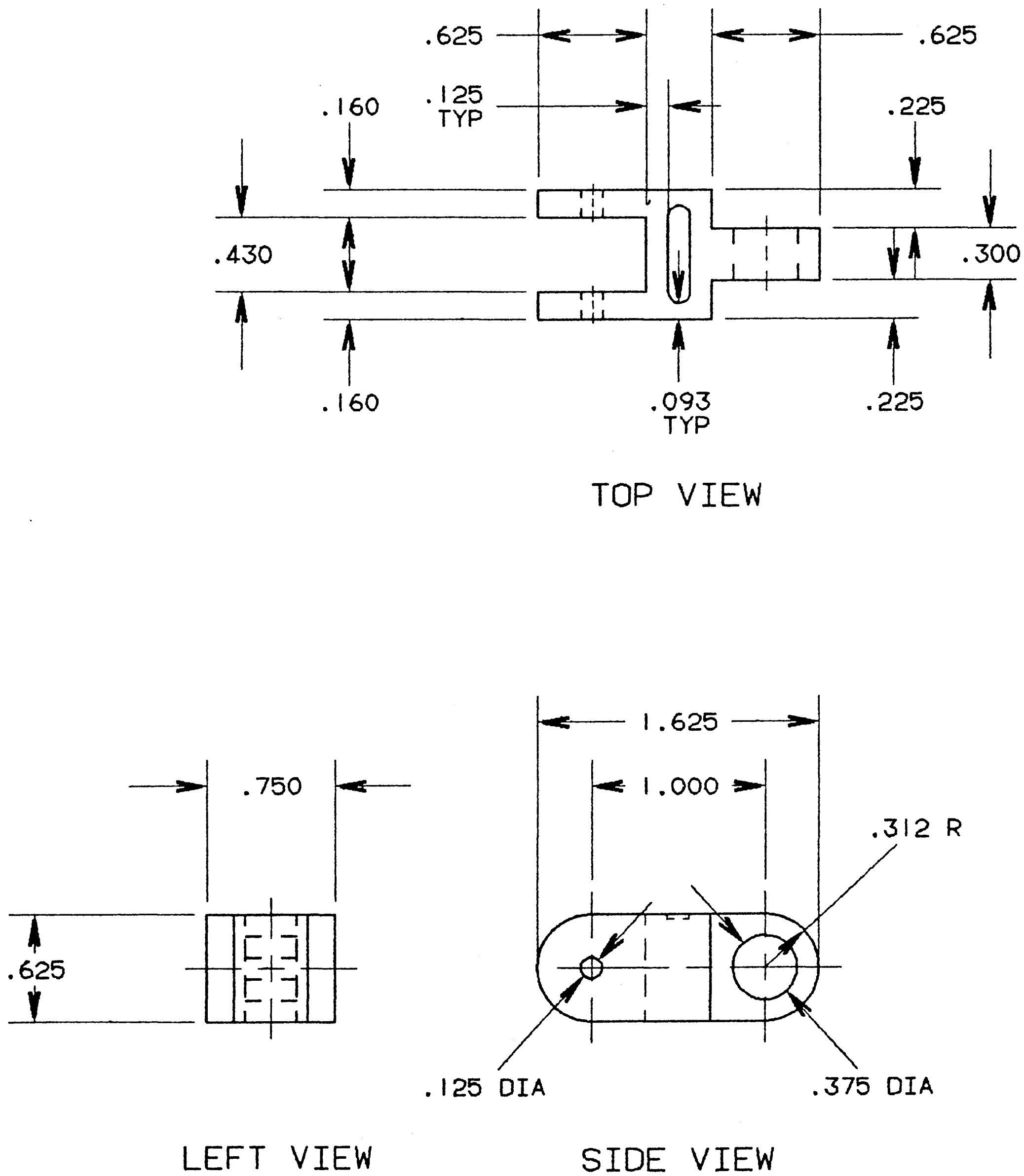
DETAILED FINGER LINK DRAWINGS AND ASSEMBLY PHOTOGRAPHS

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	LINK, FINGER, 1ST	METZNER	DATE: 11-86



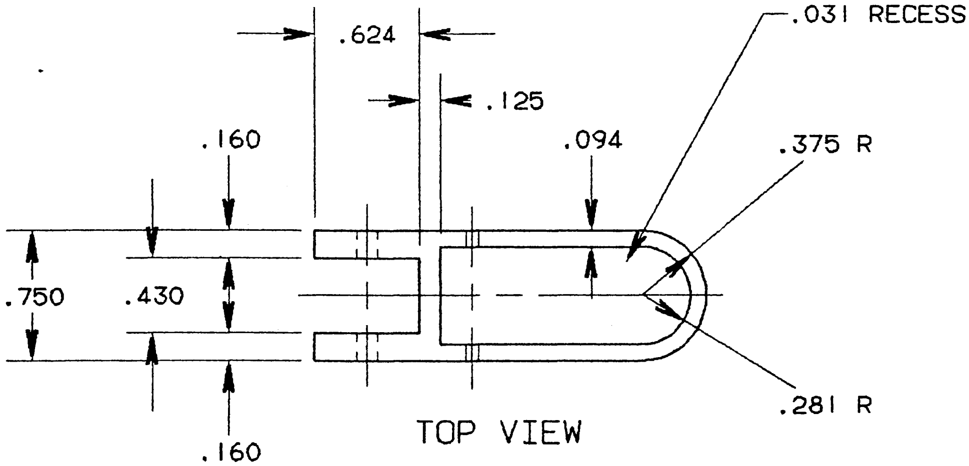
GENERAL NOTES:
DIMENSION TOL: ±.0005

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	LINK,FINGER,2ND	GETZNER	DATE: 11-86

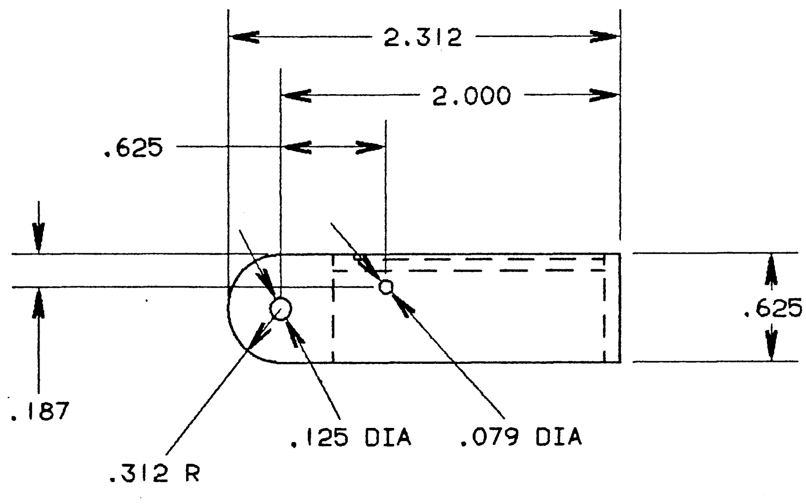


GENERAL NOTES:
DIMENSION TOL: ±.0005

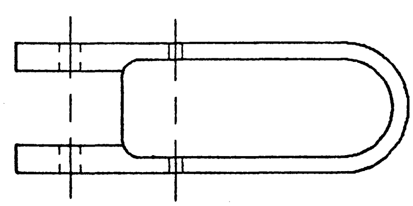
MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	LINK, FINGER, 3RD	METZNER	DATE: 11-86



TOP VIEW

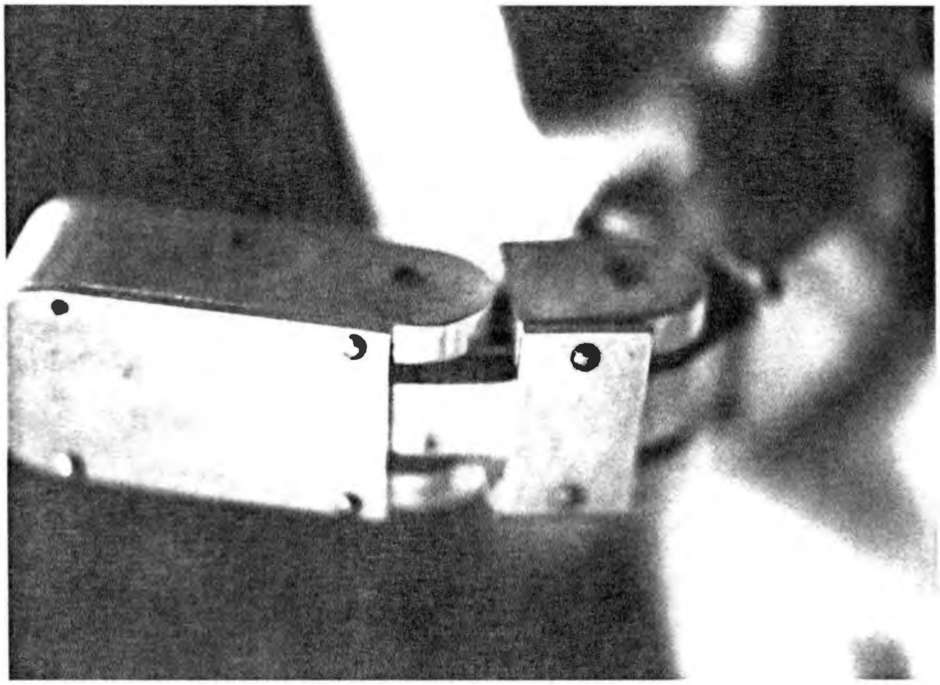


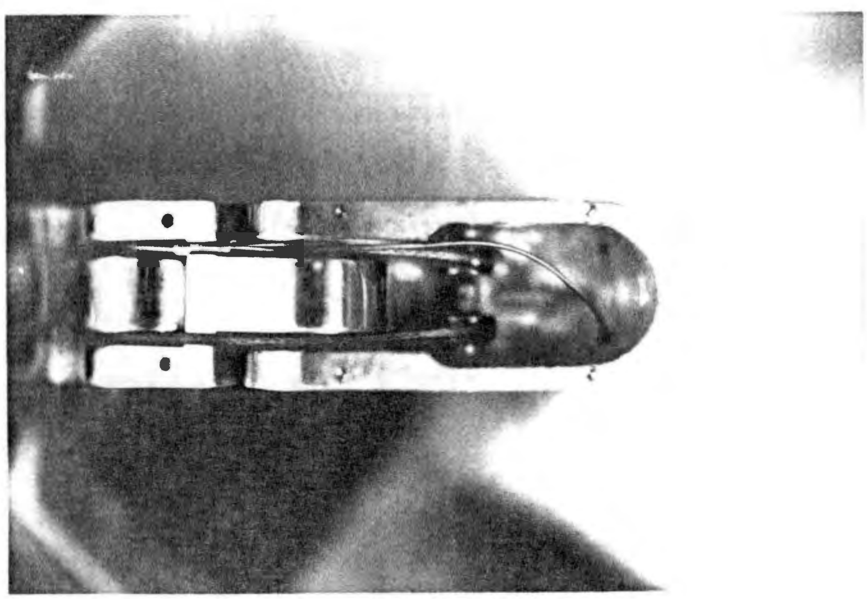
SIDE VIEW



BOTTOM VIEW

GENERAL NOTES:
DIMENSION TOL: ±.0005



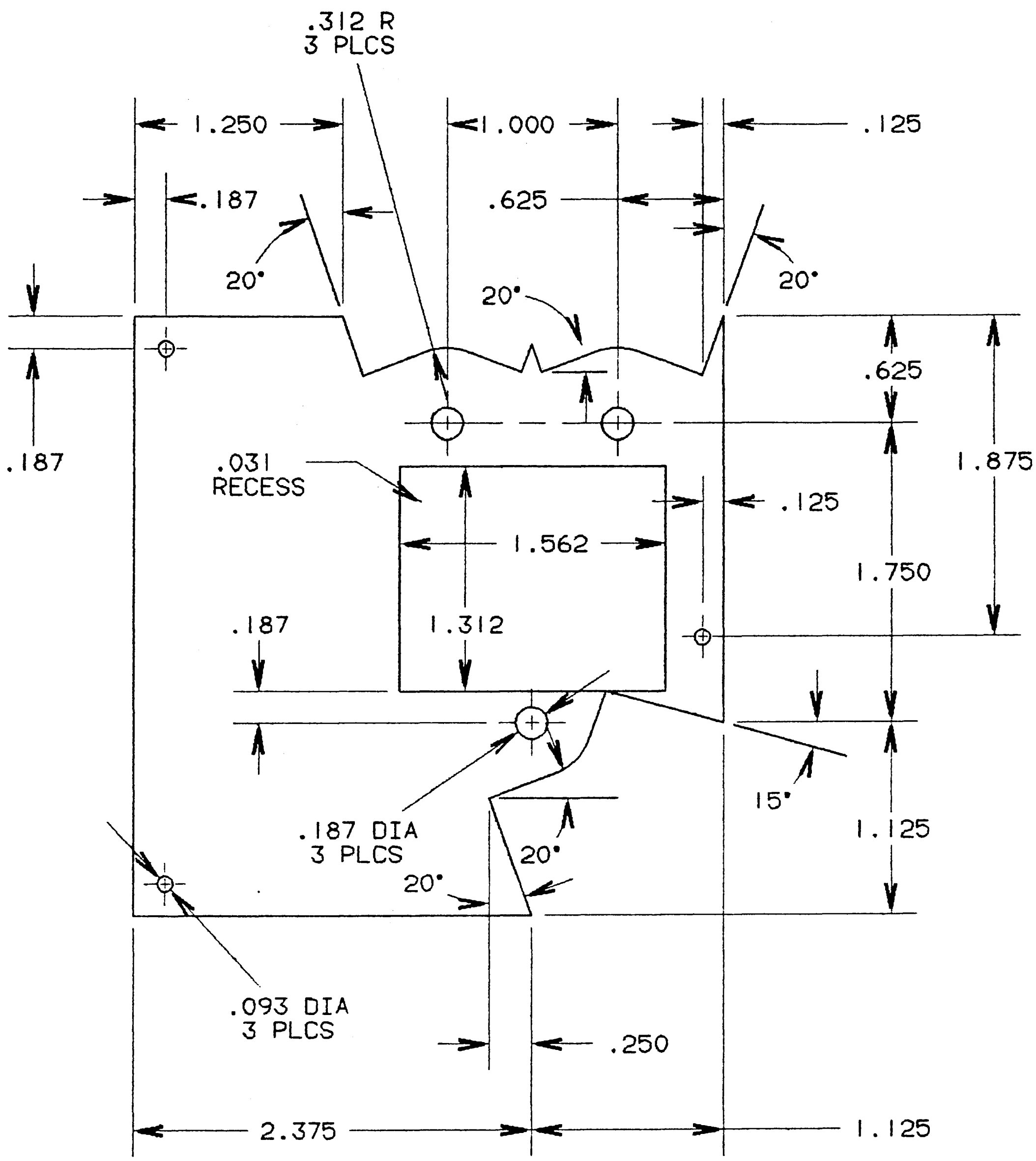


APPENDIX C

DETAILED PALM PLATE DRAWINGS

AND ASSEMBLY PHOTOGRAPHS

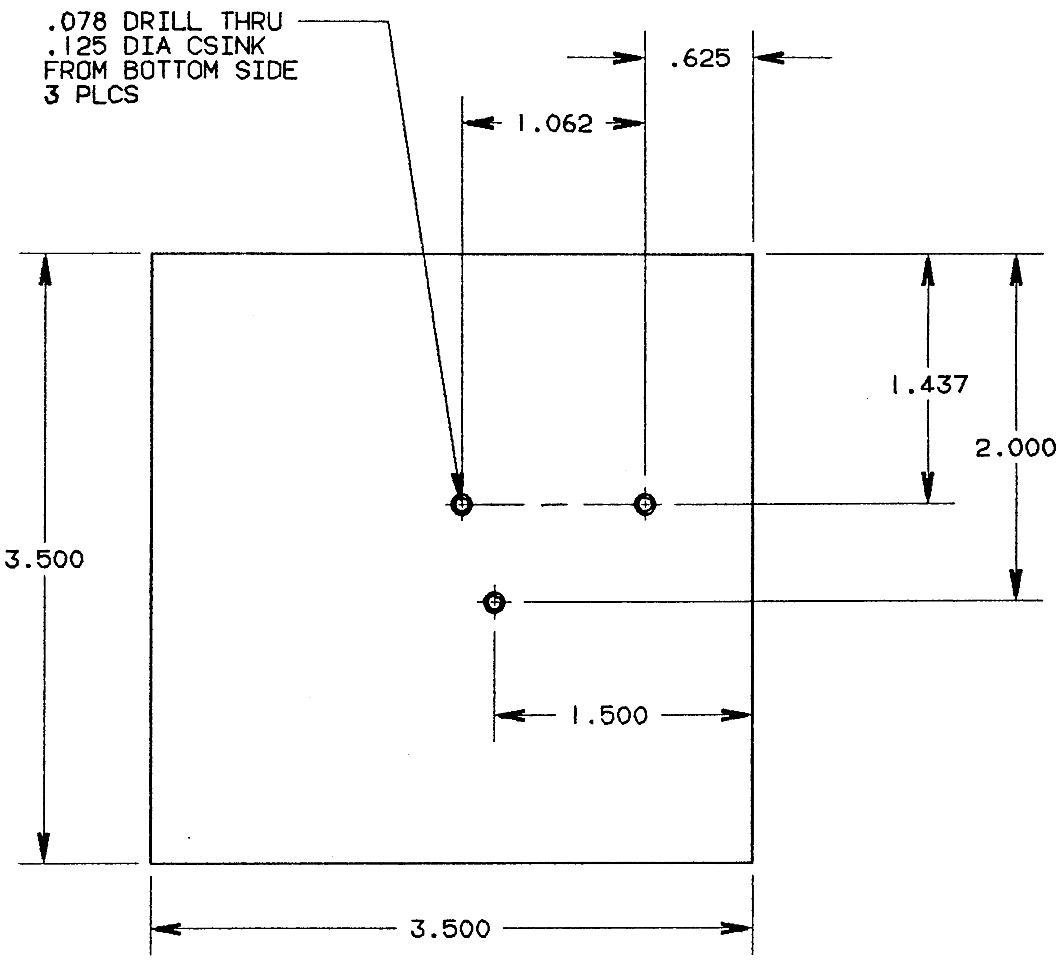
MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM PLATE	PLATE, PALM, TOP	METZNER	DATE: 11-86



TOP VIEW

GENERAL NOTES:
DIMENSION TOL: ±.0005
.125 THK TOOL PLATE

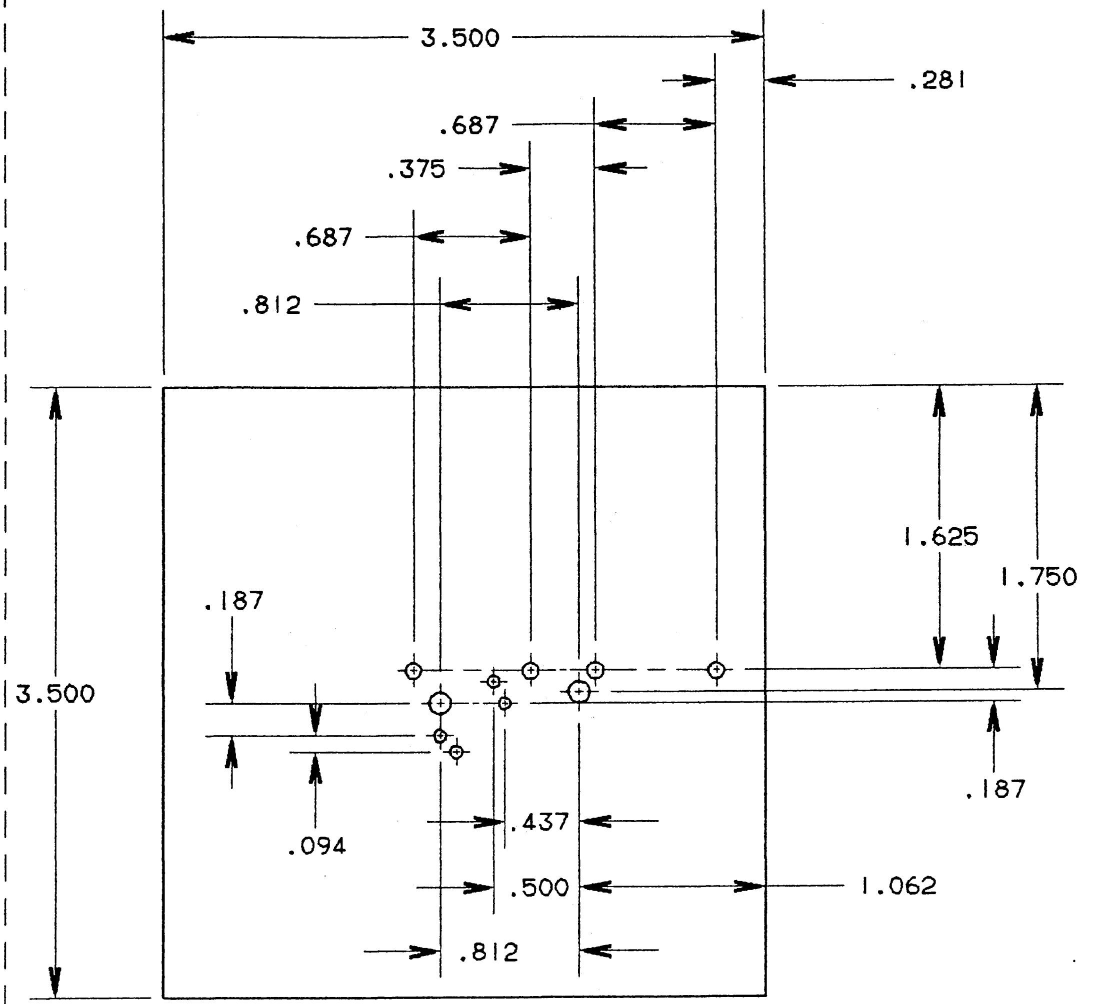
MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM PLATE	PLATE, BOTTOM	METZNER	DATE: 11-86



TOP VIEW

GENERAL NOTES:
DIMENSION TOL: ±.0005
.125 THK TOOL PLATE

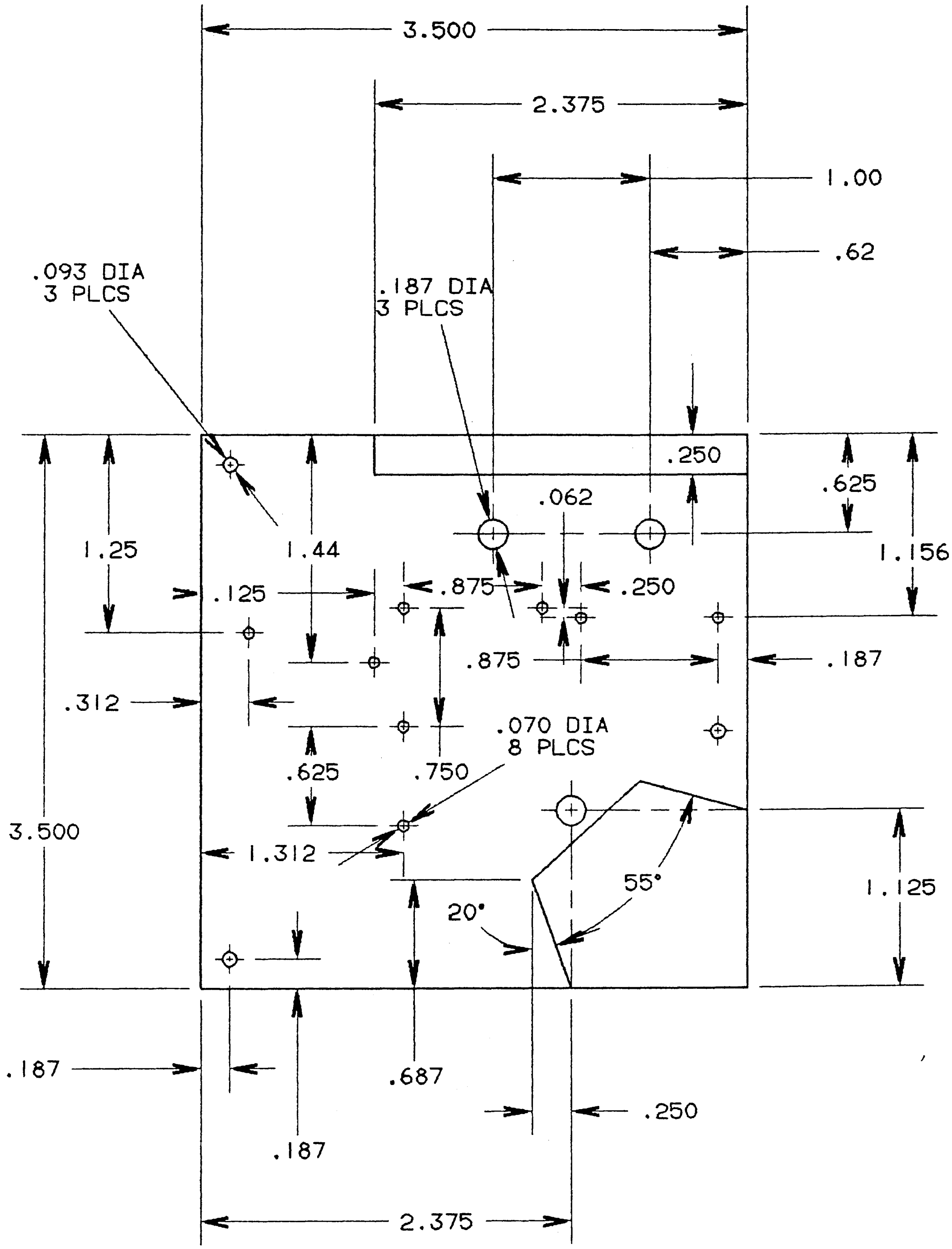
MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM PLATE	PLATE, BOTTOM	METZNER	DATE: 11-86



TOP VIEW

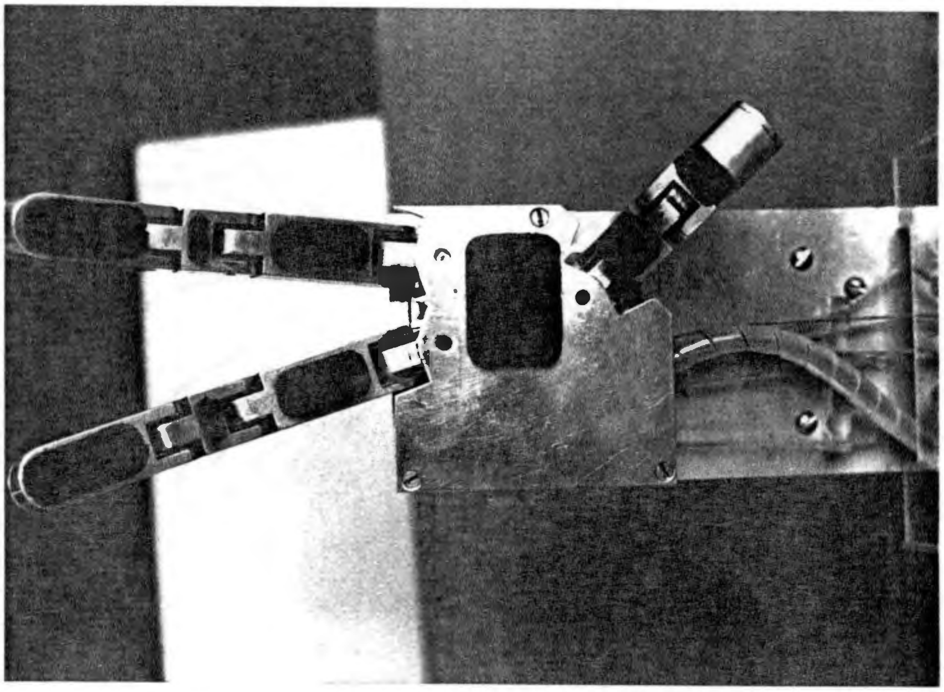
GENERAL NOTES:
DIMENSION TOL: ±.0005
.125 THK TOOL PLATE
HOLES ARE THE SAME SIZE AS IN BOTTOM PLATE.

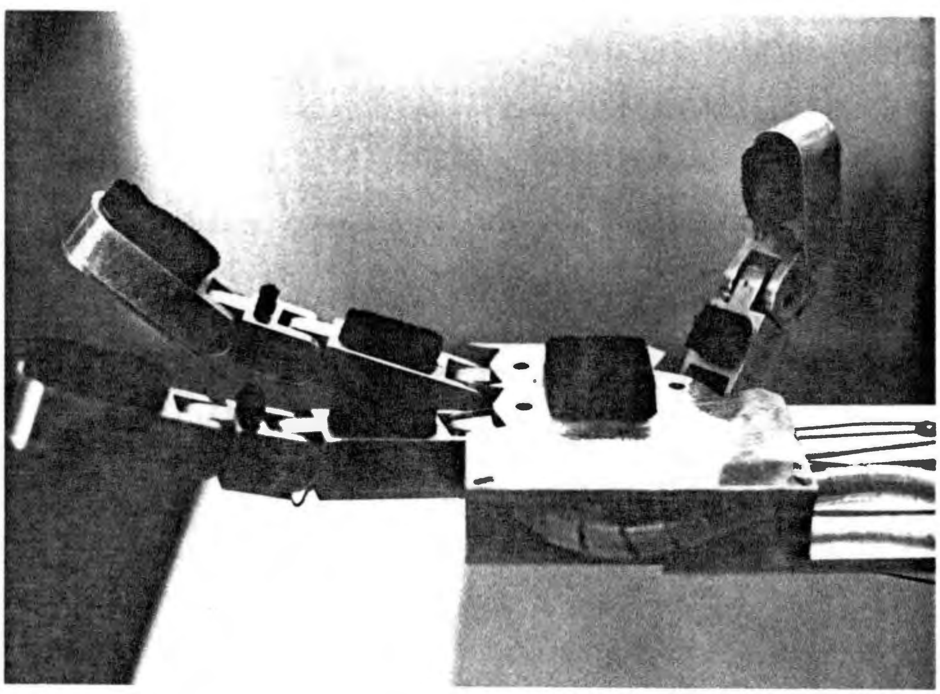
MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM PLATE	PLATE, PALM, BOTTOM	METZNER	DATE: 11-86



TOP VIEW

GENERAL NOTES:
DIMENSION TOL: ±.0005

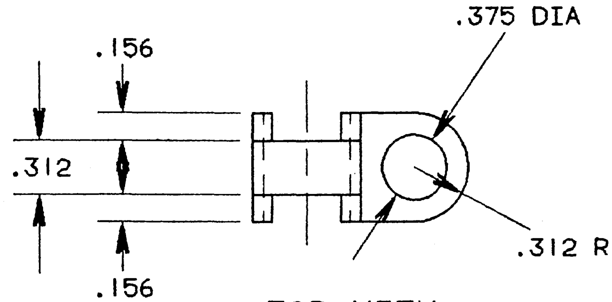




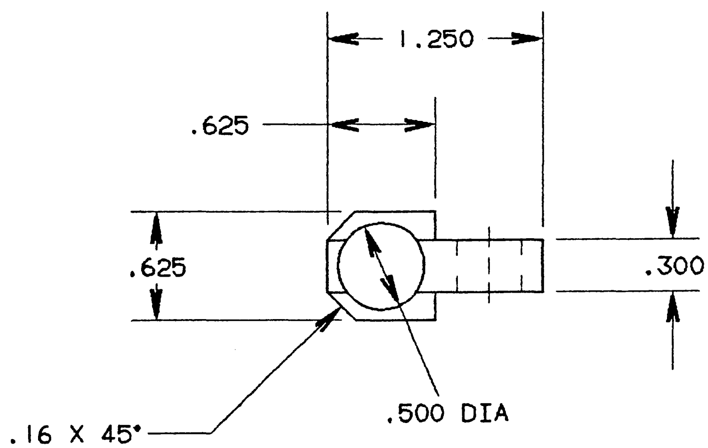
APPENDIX D

PALM-TO-FINGER LINK
AND ASSEMBLY PHOTOGRAPH

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	LINK, FINGER	METZNER	DATE: 11-86

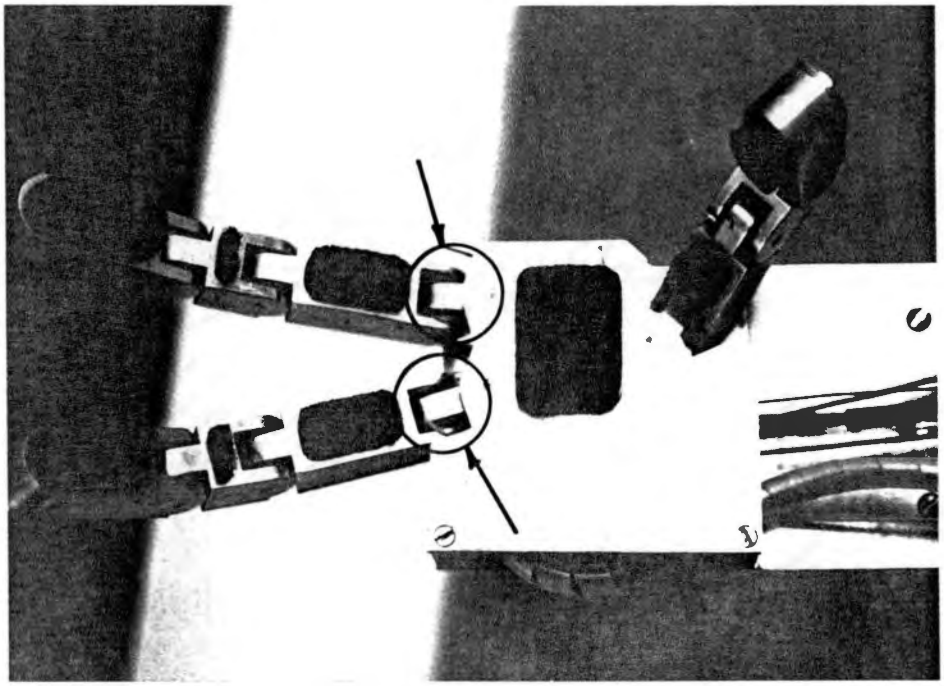


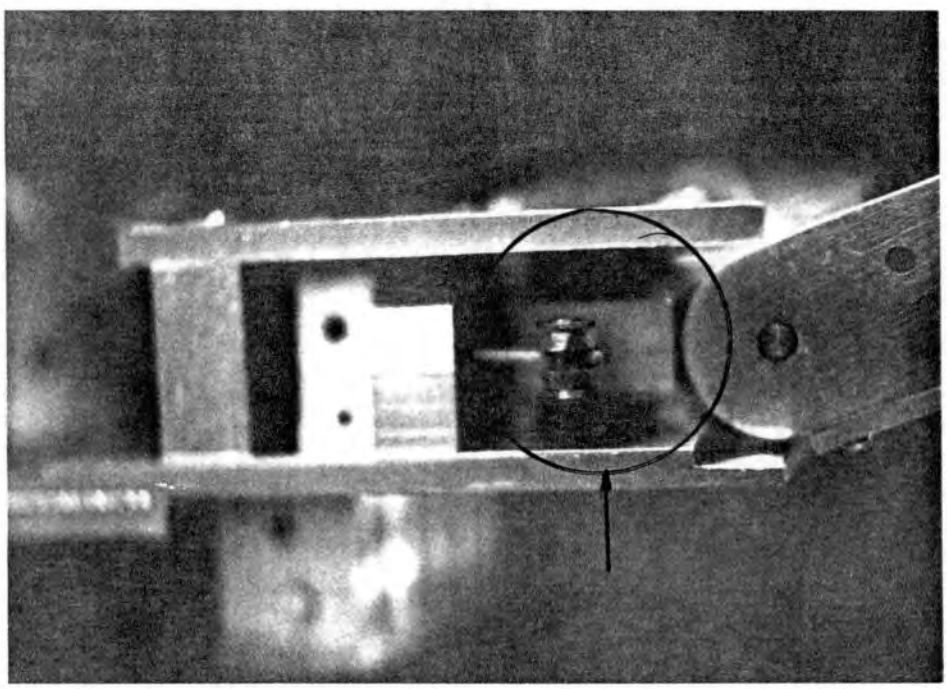
TOP VIEW



SIDE VIEW

GENERAL NOTES:
DIMENSION TOL: ±.0005



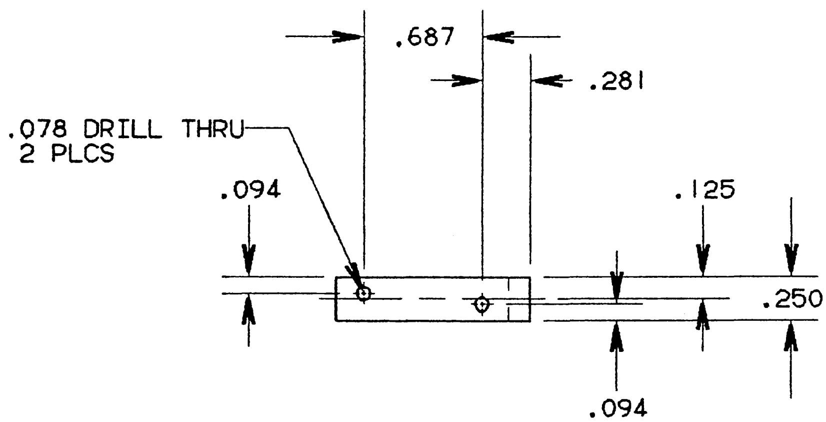


APPENDIX E

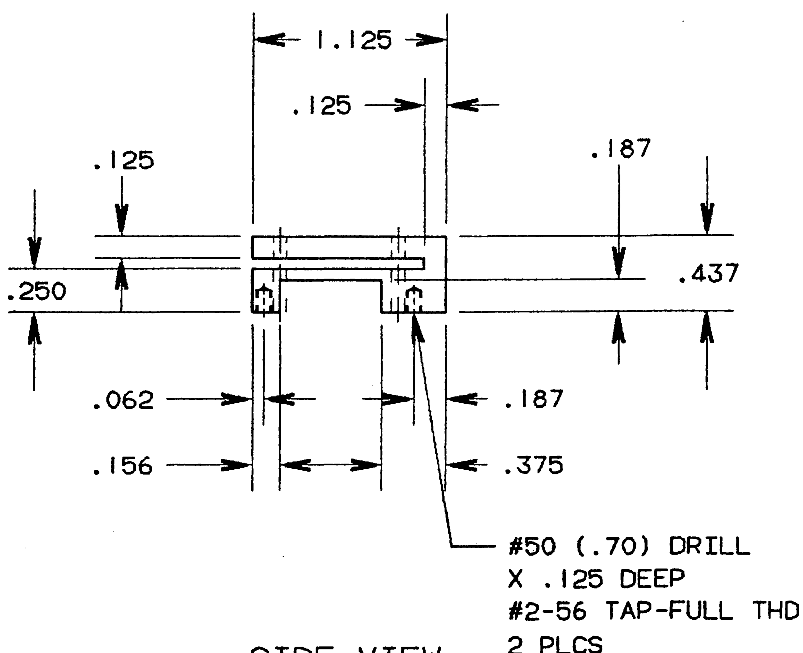
DETAILED PULLEY ASSEMBLY DRAWINGS

AND PHOTOGRAPH

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	BLOCK, PULLEY	METZNER	DATE: 11-86



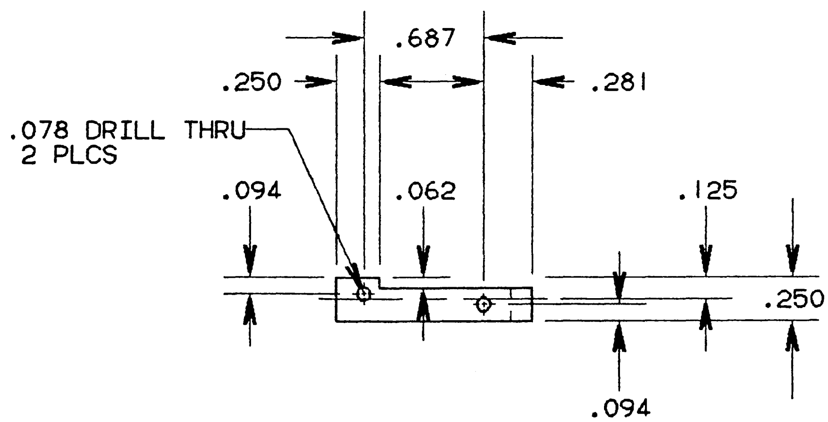
TOP VIEW



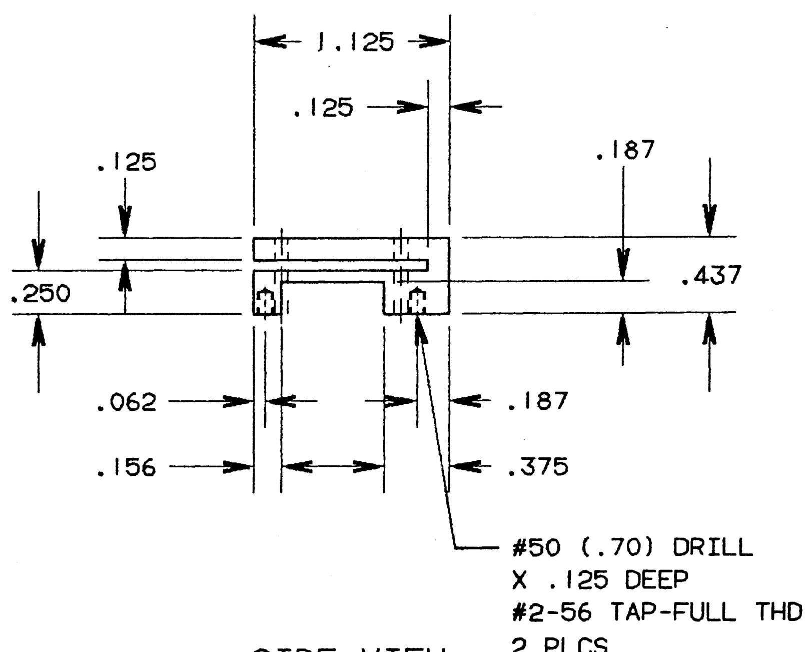
SIDE VIEW

GENERAL NOTES:
DIMENSION TOL: ±.0005

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE: 1/1
7075 ALUMINUM	BLOCK, PULLEY	METZNER	DATE: 11-86



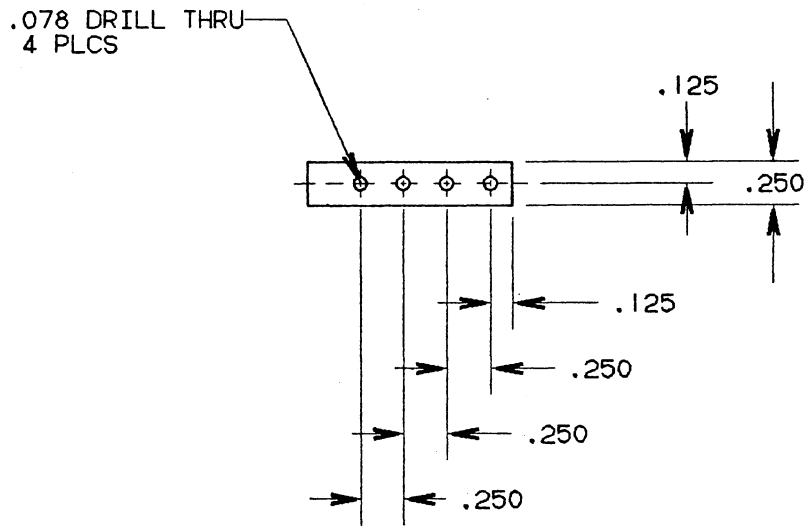
TOP VIEW



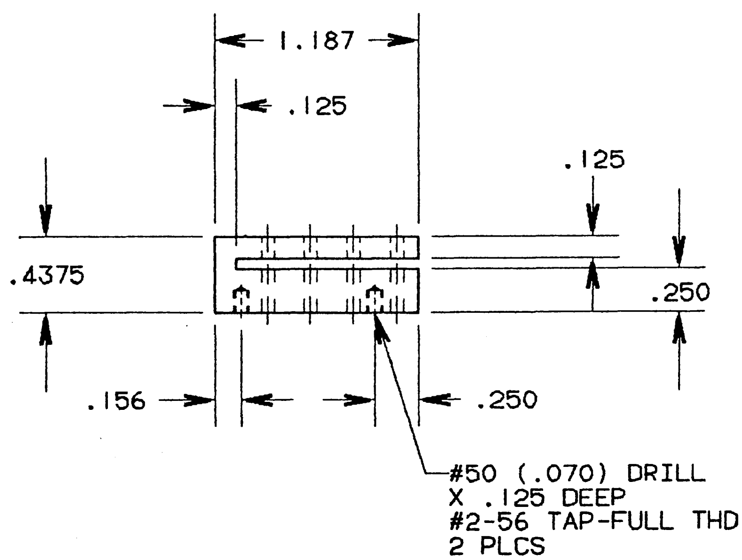
SIDE VIEW

GENERAL NOTES:
DIMENSION TOL: ±.0005

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	BLOCK, PULLEY	METZNER	DATE: 11-86



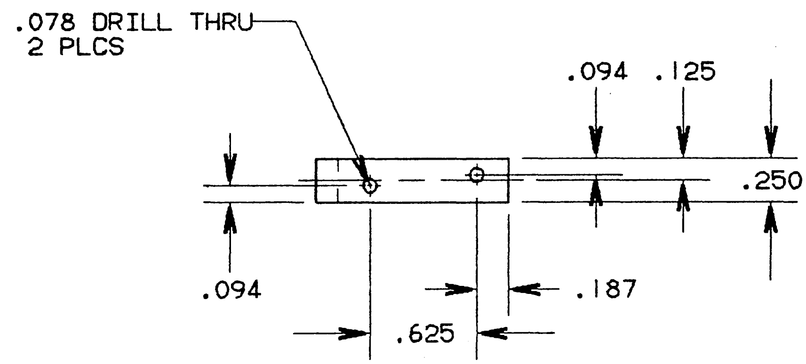
TOP VIEW



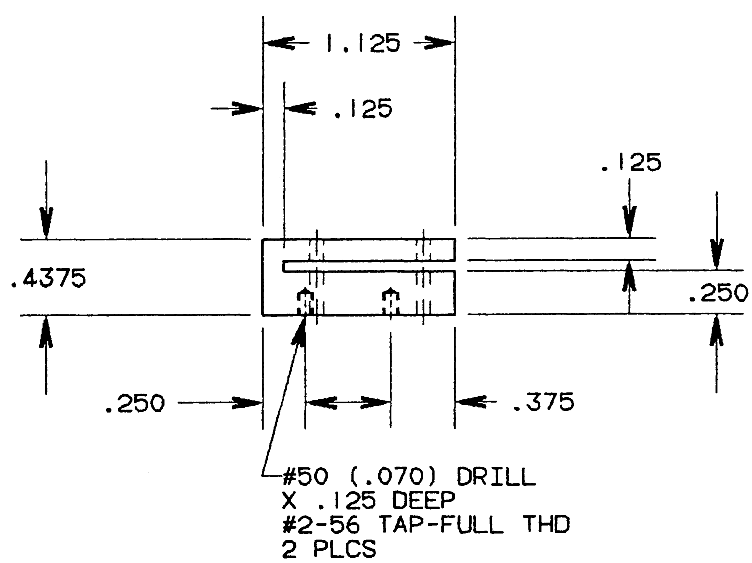
SIDE VIEW

GENERAL NOTES:
DIMENSION TOL: ±.0005

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	BLOCK, PULLEY	METZNER	DATE: 11-86



TOP VIEW

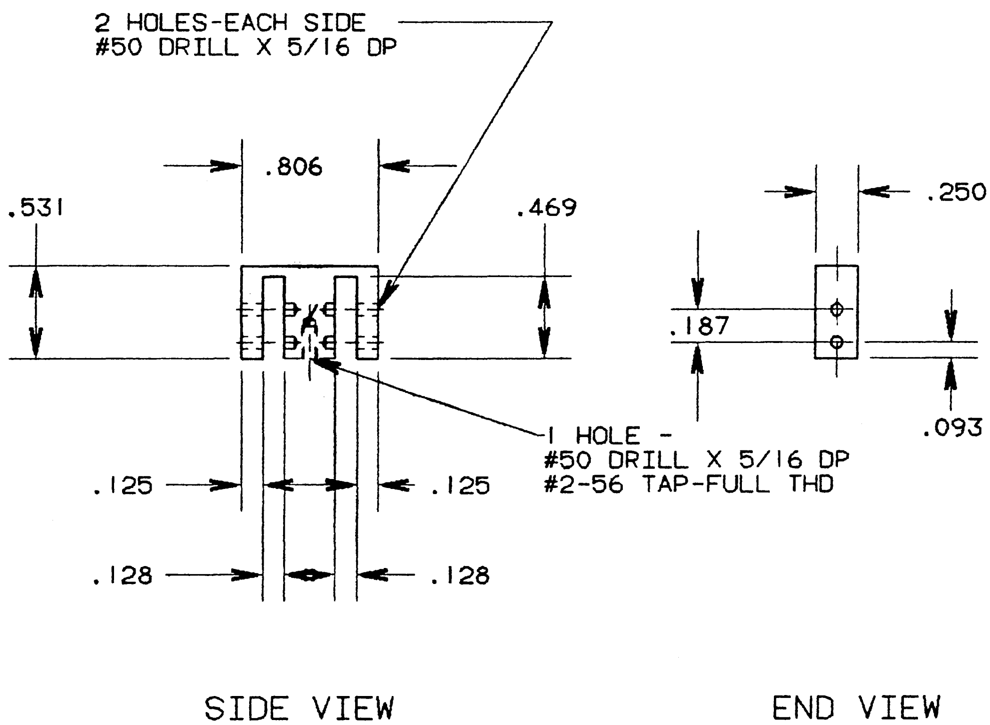


SIDE VIEW

GENERAL NOTES:

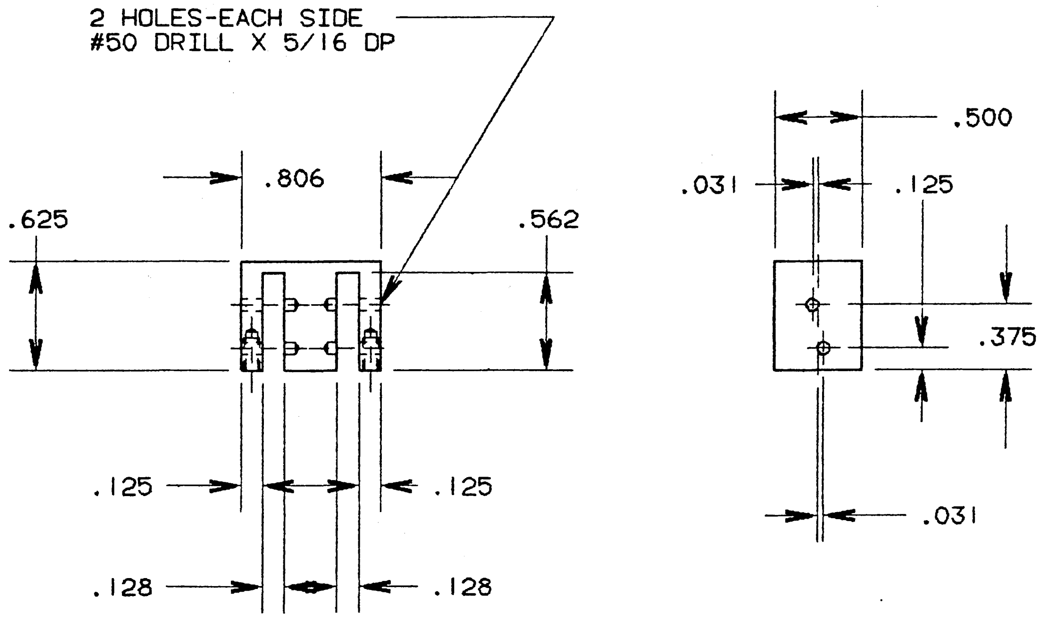
DIMENSION TOL: ±.0005

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	PULLEY, BOTTOM	METZNER	DATE: 11-86



GENERAL NOTES:
DIMENSION TOL: ±.0005

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	BLOCK,PULLEY,BOTTOM	METZNER	DATE: 11-86

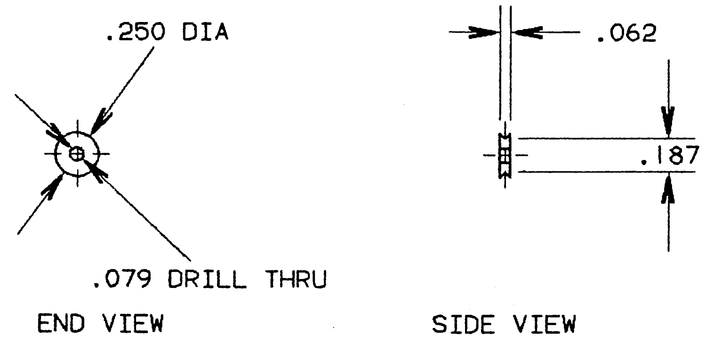


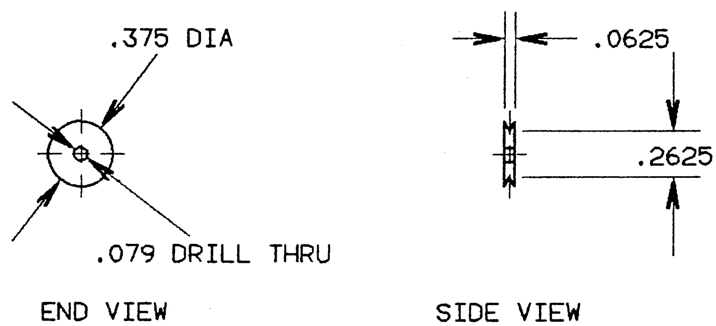
SIDE VIEW

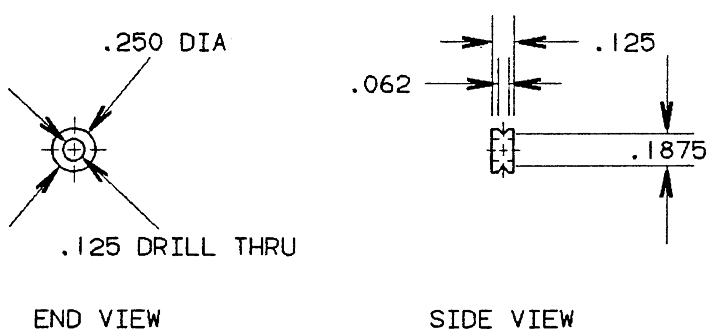
END VIEW

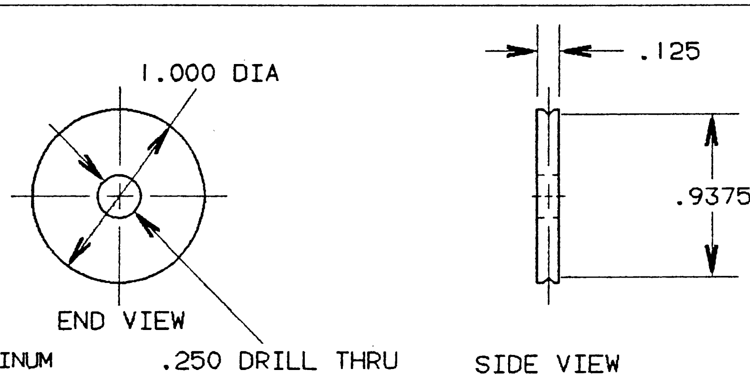
GENERAL NOTES:
DIMENSION TOL: ±.0005

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
360 BRASS*	PULLEYS	METZNER	DATE: 11-86



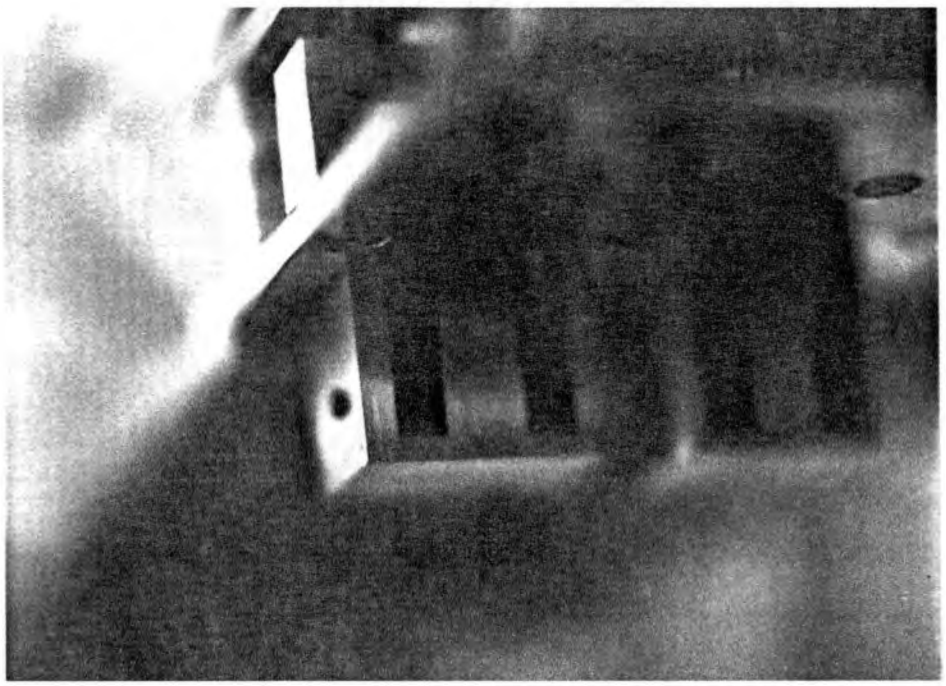






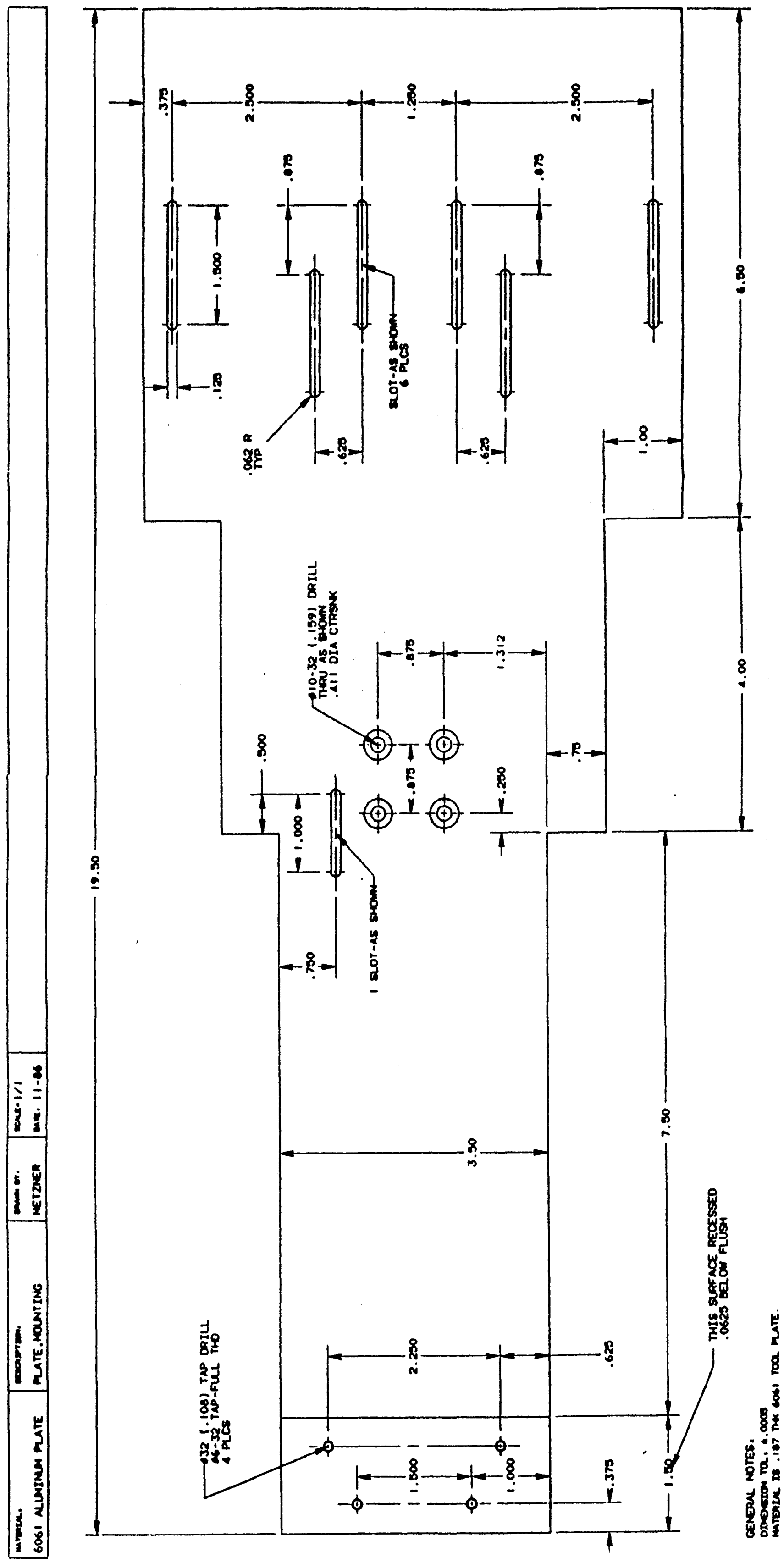
*MATL=6061 ALUMINUM

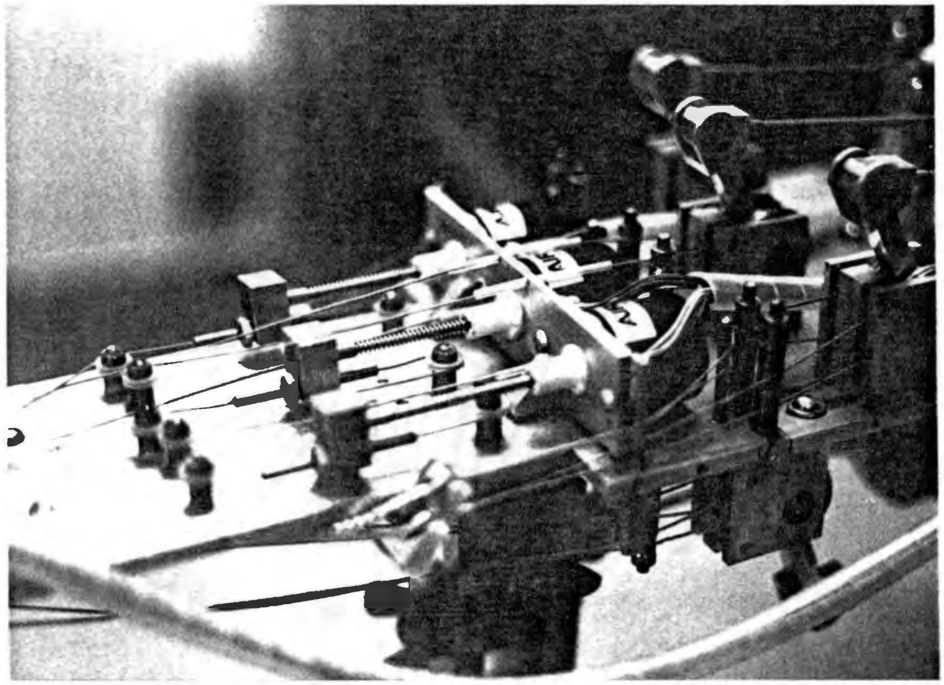
GENERAL NOTES:
DIMENSION TOL: ±.0005

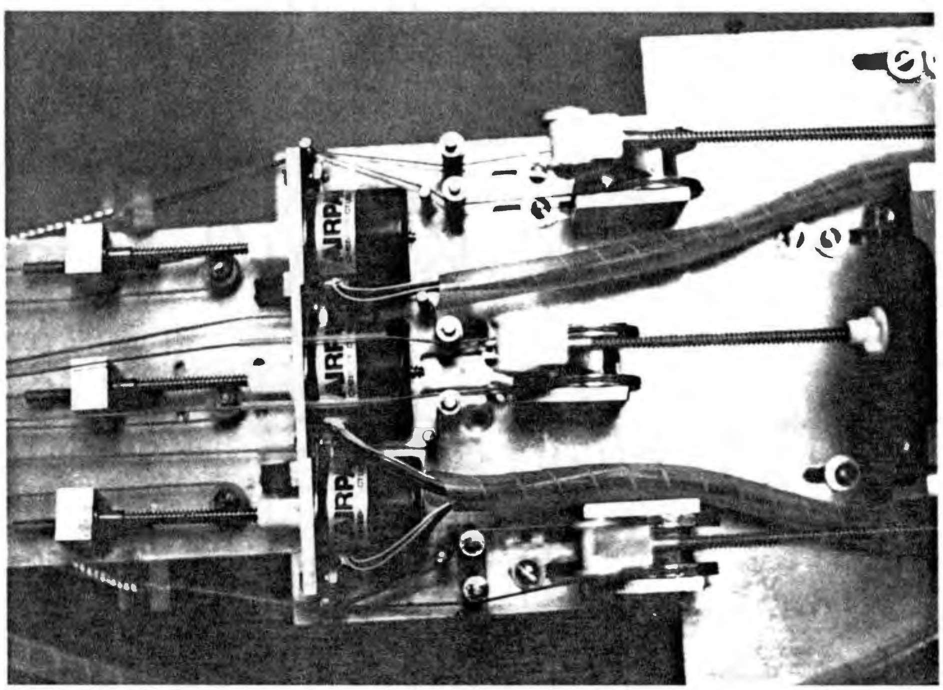


APPENDIX F

DETAILED BASE-PLATE DRAWING AND ASSEMBLY PHOTOGRAPHS


MATERIAL:
6061 ALUMINUM PLATE
DESCRIPTION:
PLATE, MOUNTING
DRAWN BY:
METZNER
SCALE: 1/1
DATE: 11-86
19.50
.375
2.500
1.250
2.500
.875
.875
1.500
.125
.062 R
TYP
.625
.625
SLOT-AS SHOWN
6 PLCS
1.00
6.50
#10-32 (.159) DRILL
THRU AS SHOWN
.411 DIA CTRSNK
.875
1.312
.500
1.000
.875
.250
.75
4.00
1 SLOT-AS SHOWN
.750
3.50
7.50
THIS SURFACE RECESSED
.0625 BELOW FLUSH
#32 (.108) TAP DRILL
#4-32 TAP-FULL THD
4 PLCS
2.250
.625
1.500
1.000
.375
1.50
GENERAL NOTES:
DIMENSION TOL. ±.0005
MATERIAL IS .187 THK 6061 TOOL PLATE.

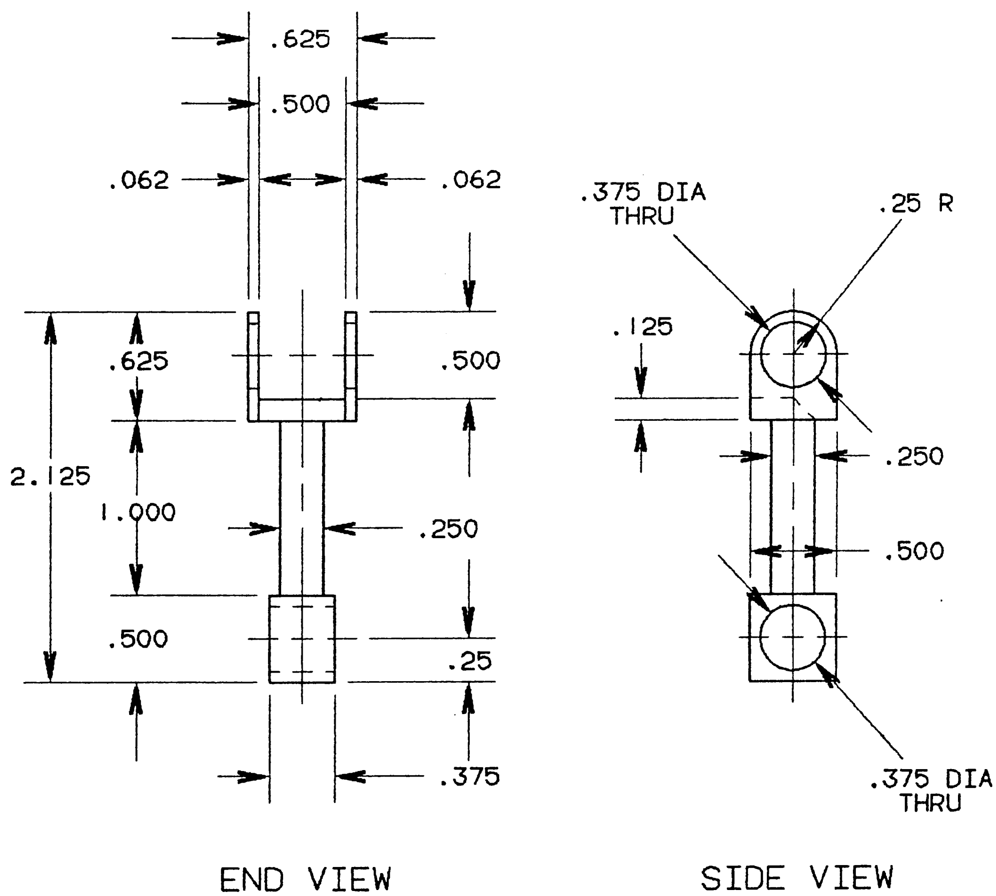




APPENDIX G

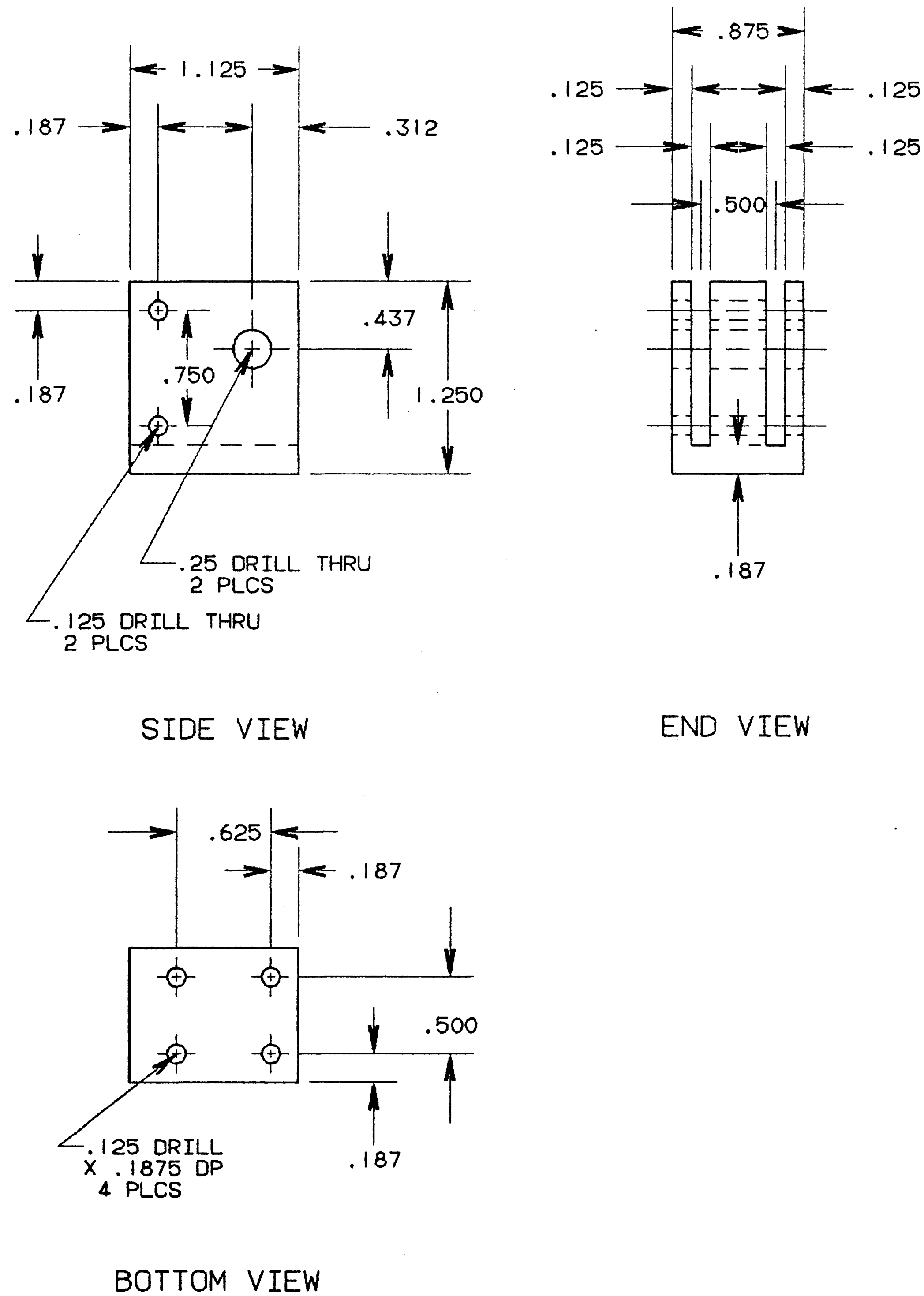
ACTUATION SYSTEM AND SUB-ASSEMBLY DRAWINGS AND ASSEMBLY PHOTOGRAPHS

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	LEVER, ARM	METZNER	DATE: 11-86



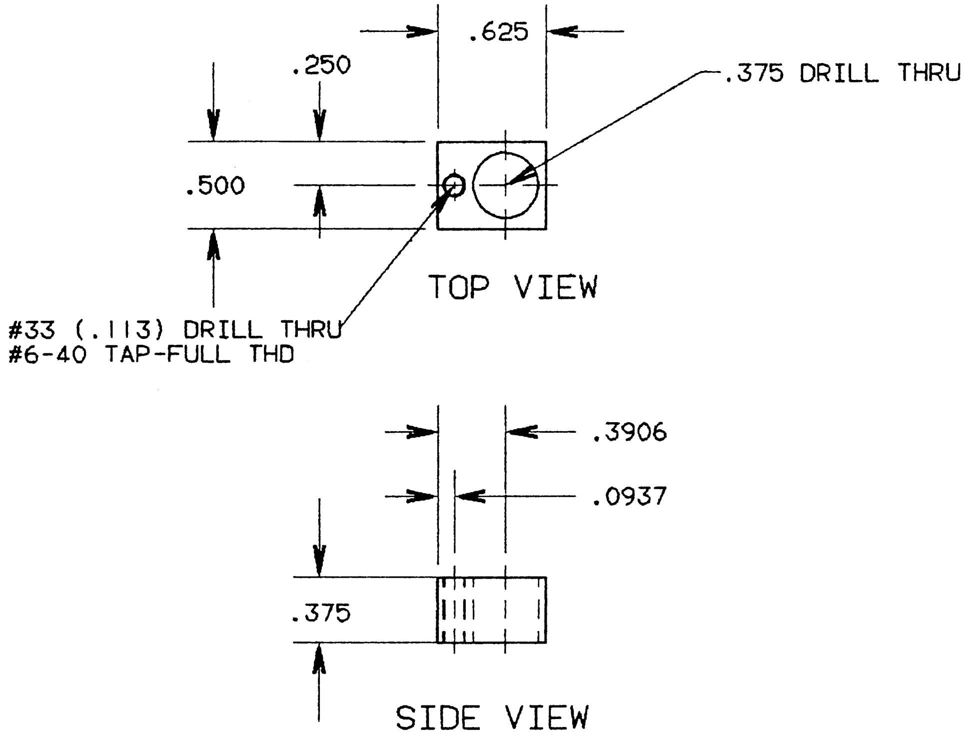
GENERAL NOTES:
DIMENSION TOL: ±.0005

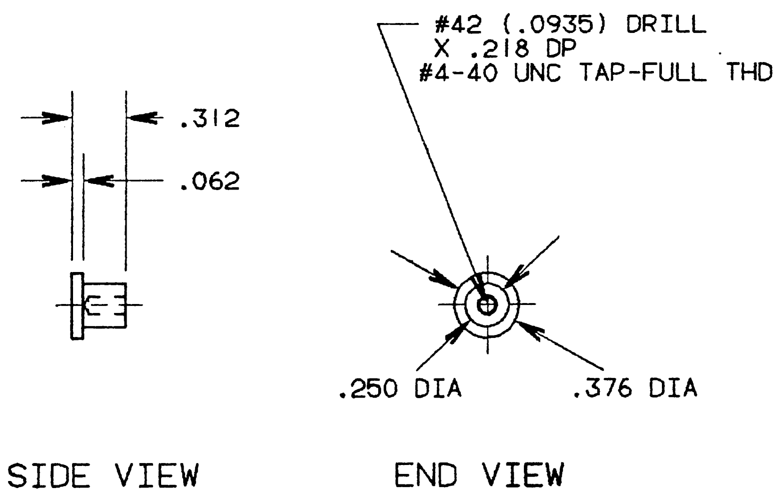
MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	PULLEY ASSEMBLY, LEVER	METZNER	DATE: 11-86



GENERAL NOTES:
DIMENSION TOL: ±.0005

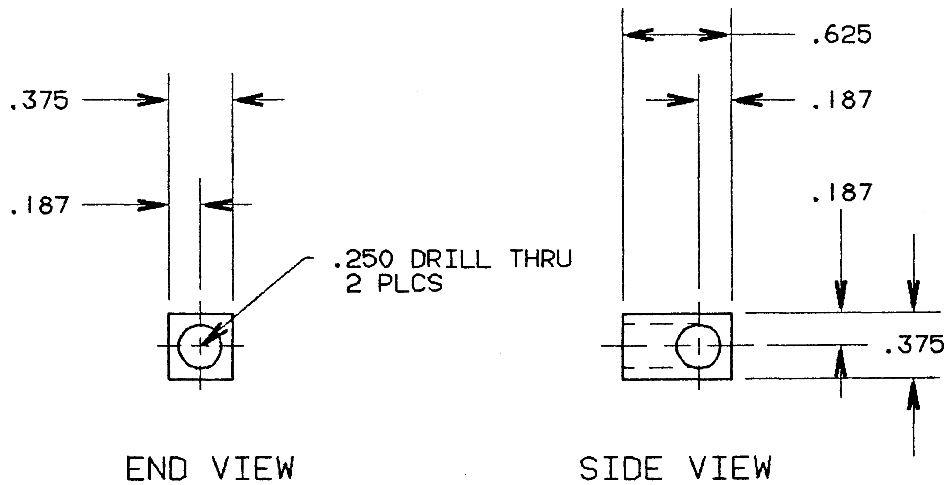
MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	SWIVEL,CABLE	METZNER	DATE: 11-86

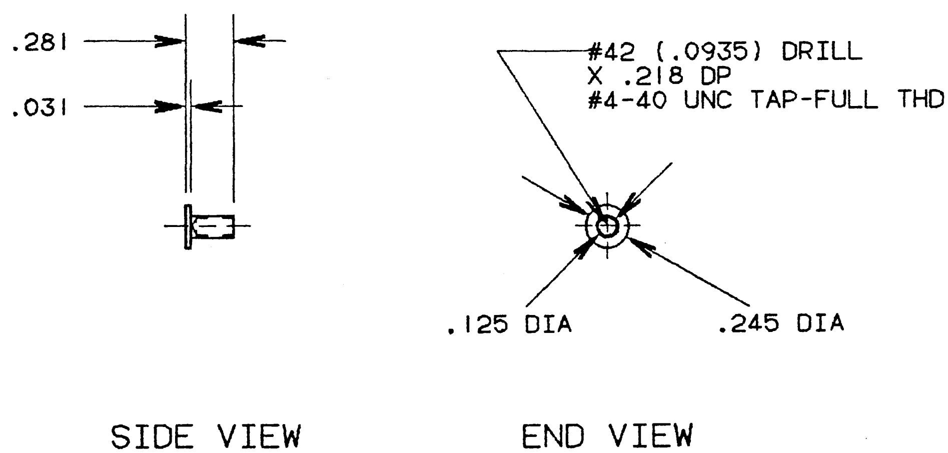




GENERAL NOTES:
DIMENSION TOL: ±.0005

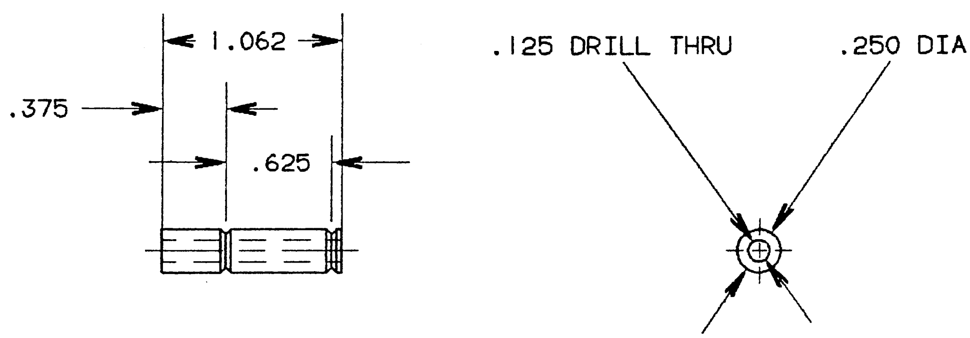
MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	SWIVEL, LEVER	METZNER	DATE: 11-86





GENERAL NOTES:
DIMENSION TOL: ±.0005

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
360 BRASS	SPINDLE, CABLE	METZNER	DATE: 11-86

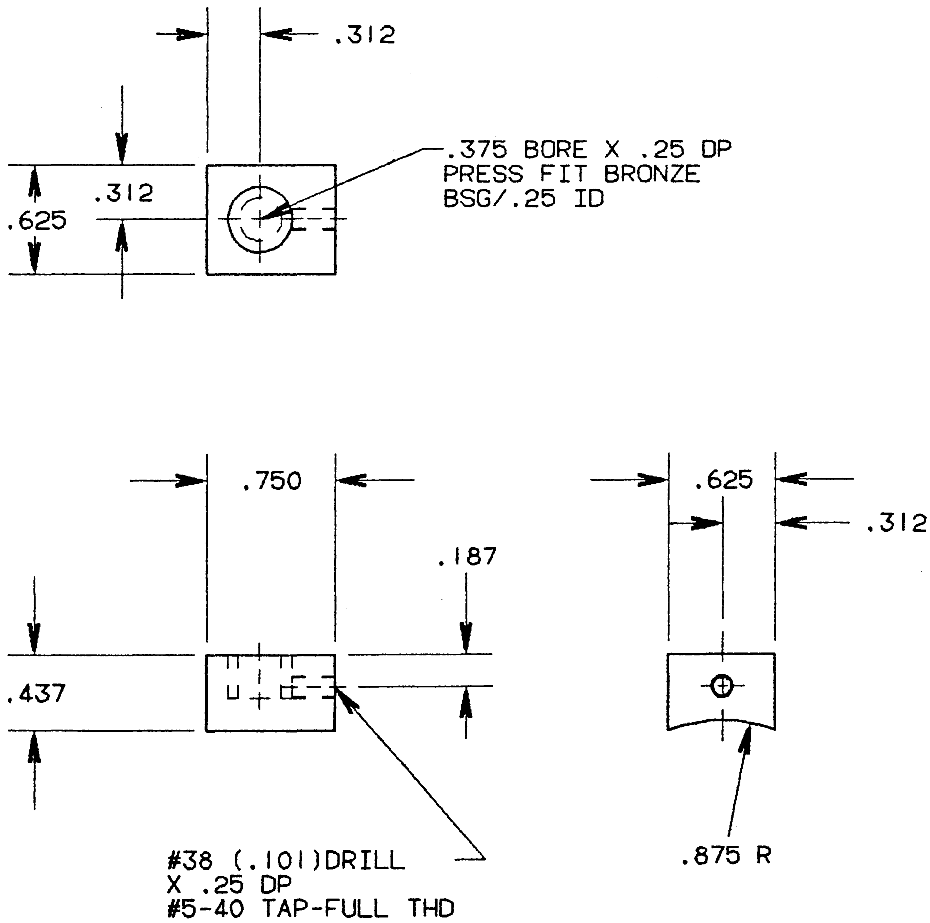


SIDE VIEW

END VIEW

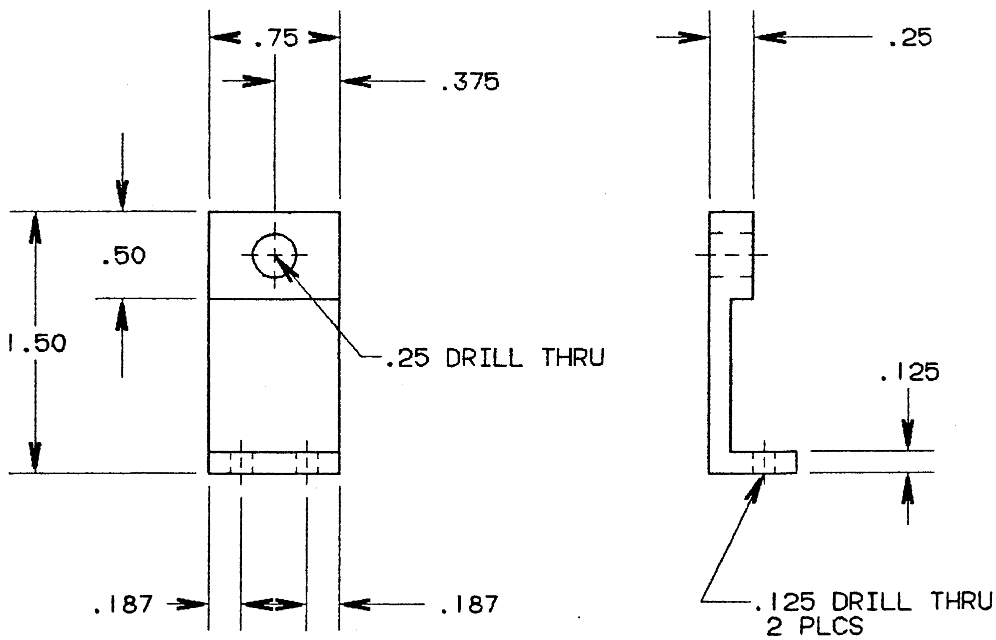
GENERAL NOTES:
DIMENSION TOL: ±.0005

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	ACTUATOR, GIMBAL	METZNER	DATE: 11-86



GENERAL NOTES:
DIMENSION TOL: ±.0005

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	ACTUATOR, GIMBAL	METZNER	DATE: 11-86

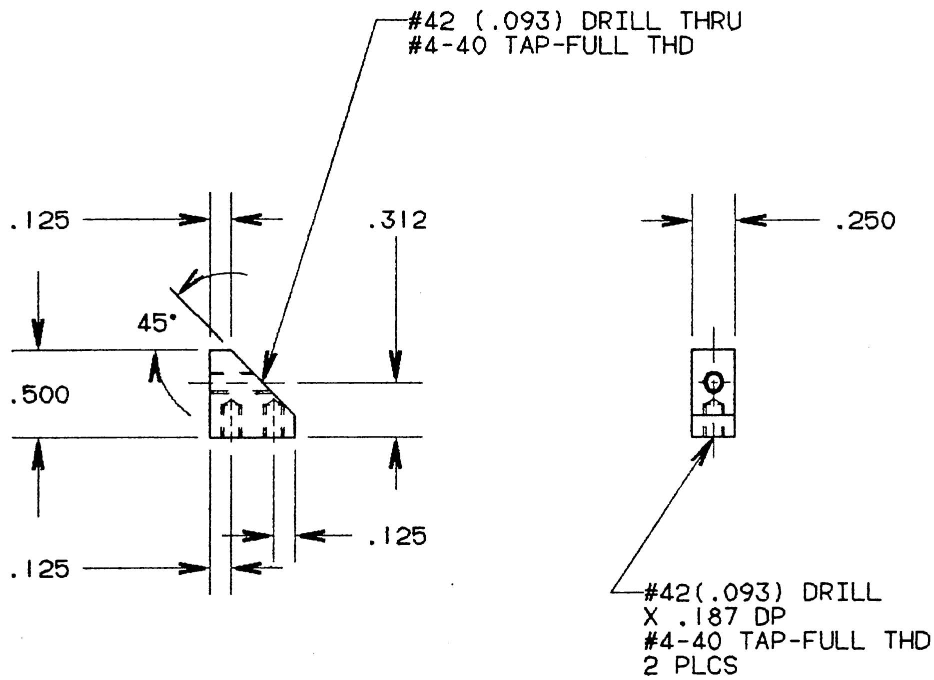


SIDE VIEW

END VIEW

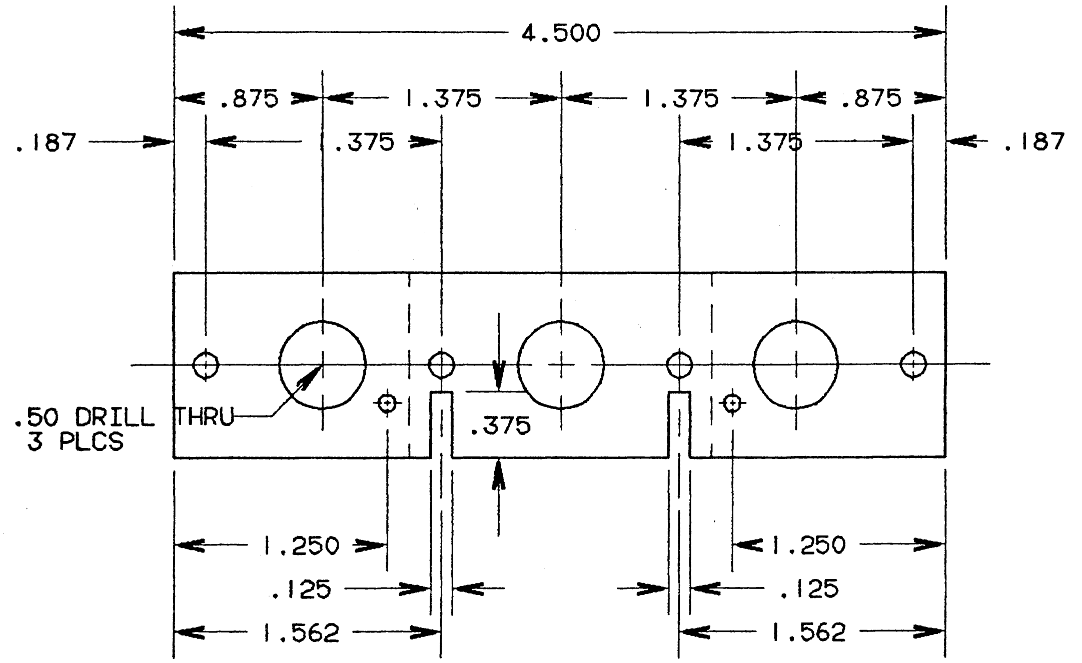
GENERAL NOTES:
DIMENSION TOL: ±.0005

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	BLOCK, ACTUATOR PLATE	METZNER	DATE: 11-86

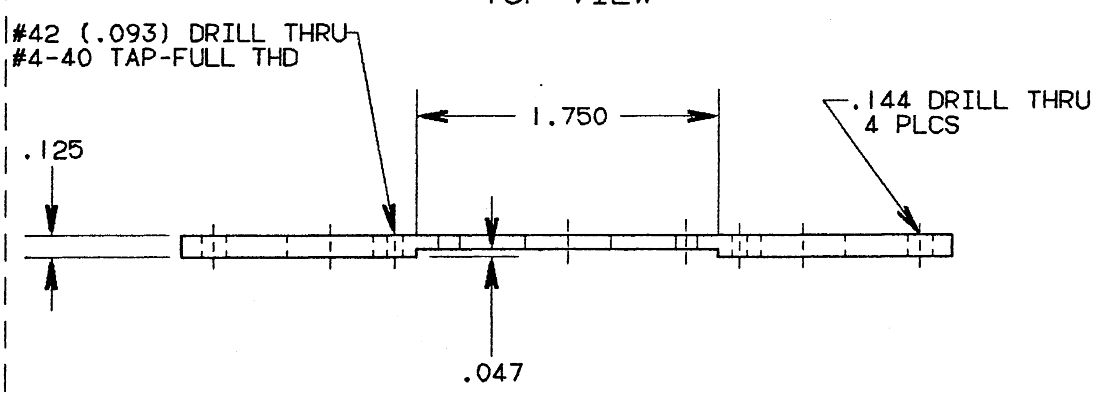


GENERAL NOTES:
DIMENSION TOL: ±.0005

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
6061 ALUMINUM PLATE	PLATE, ACTUATOR ASSY	METZNER	DATE: 11-86

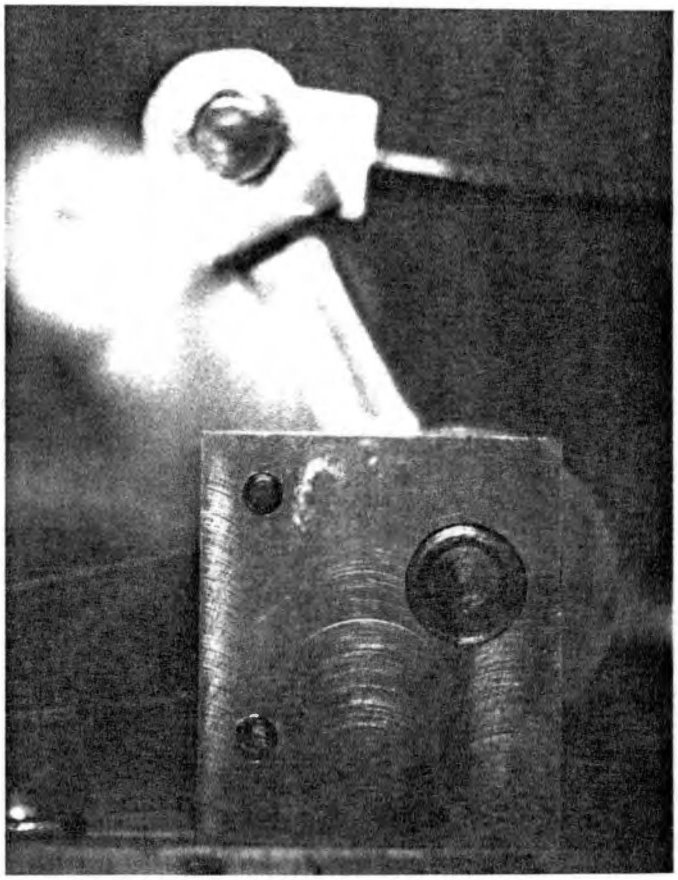


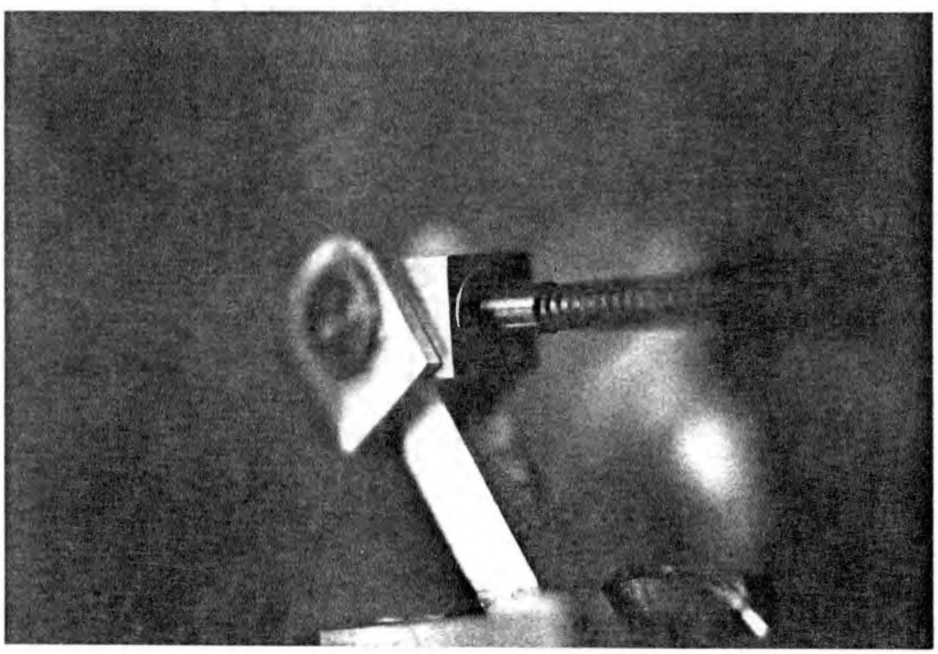
TOP VIEW

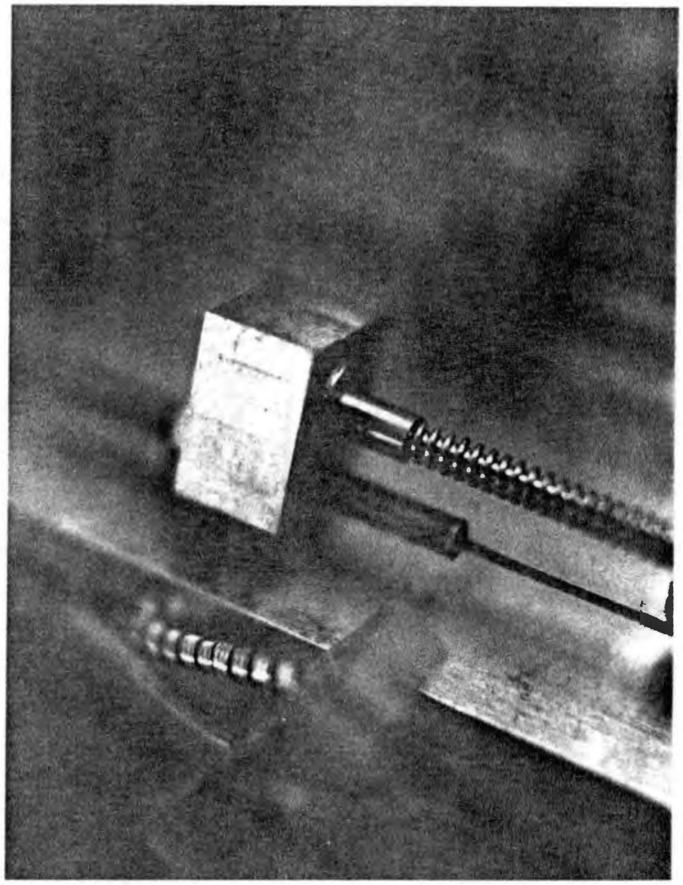


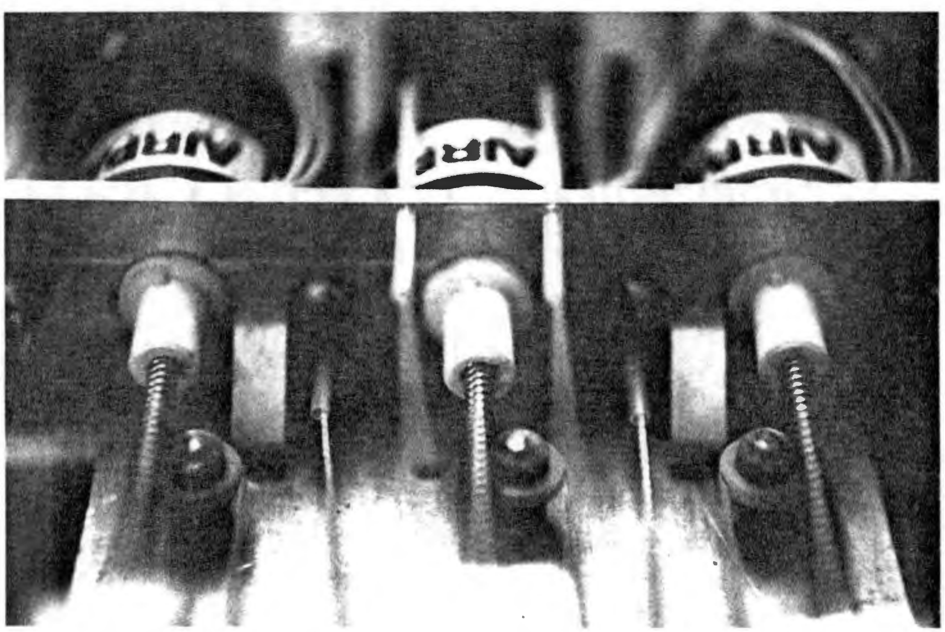
SIDE VIEW

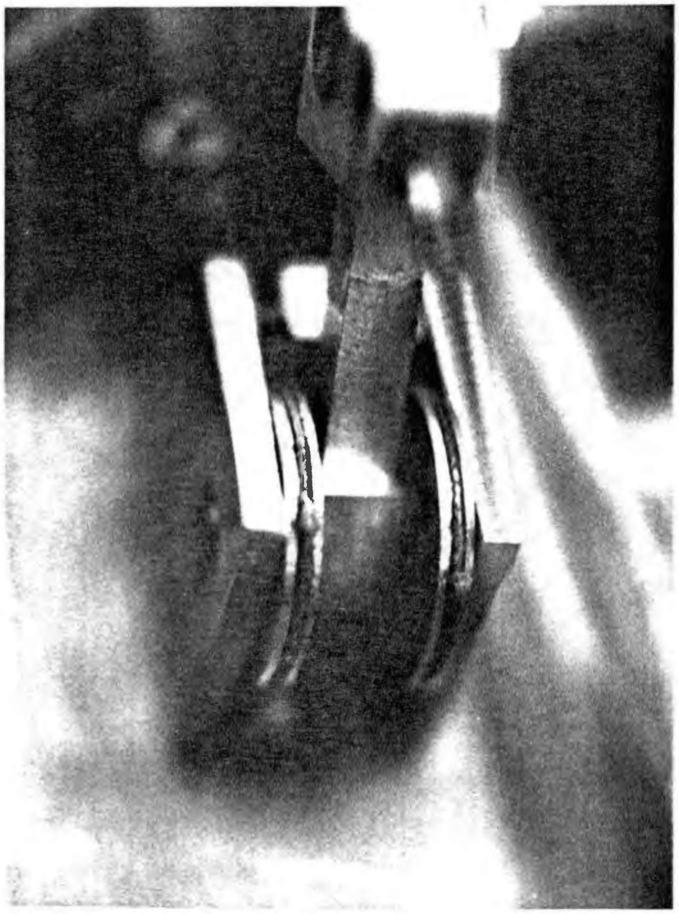
GENERAL NOTES:
DIMENSION TOL: ±.0005

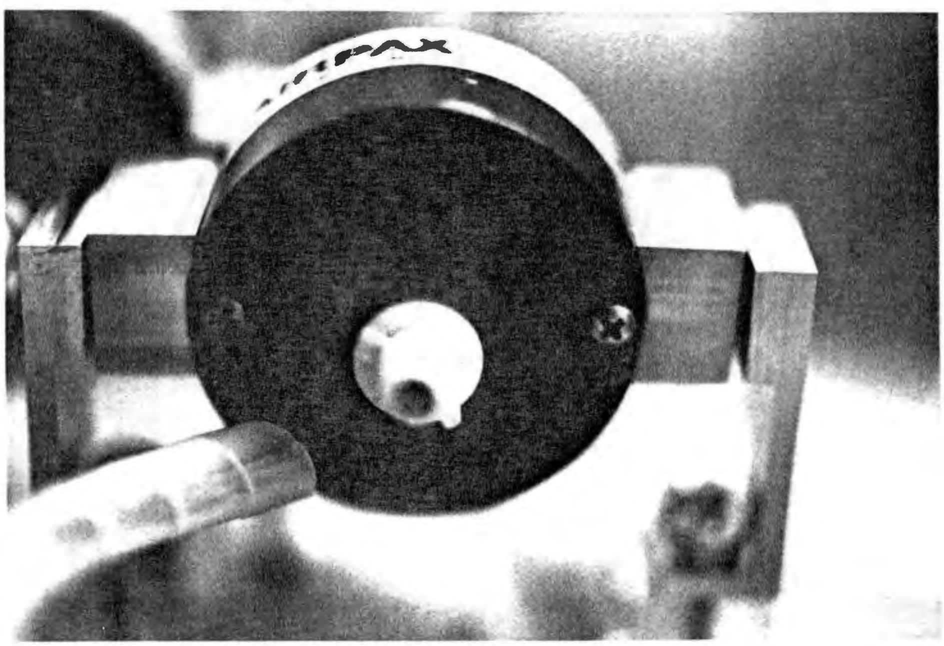








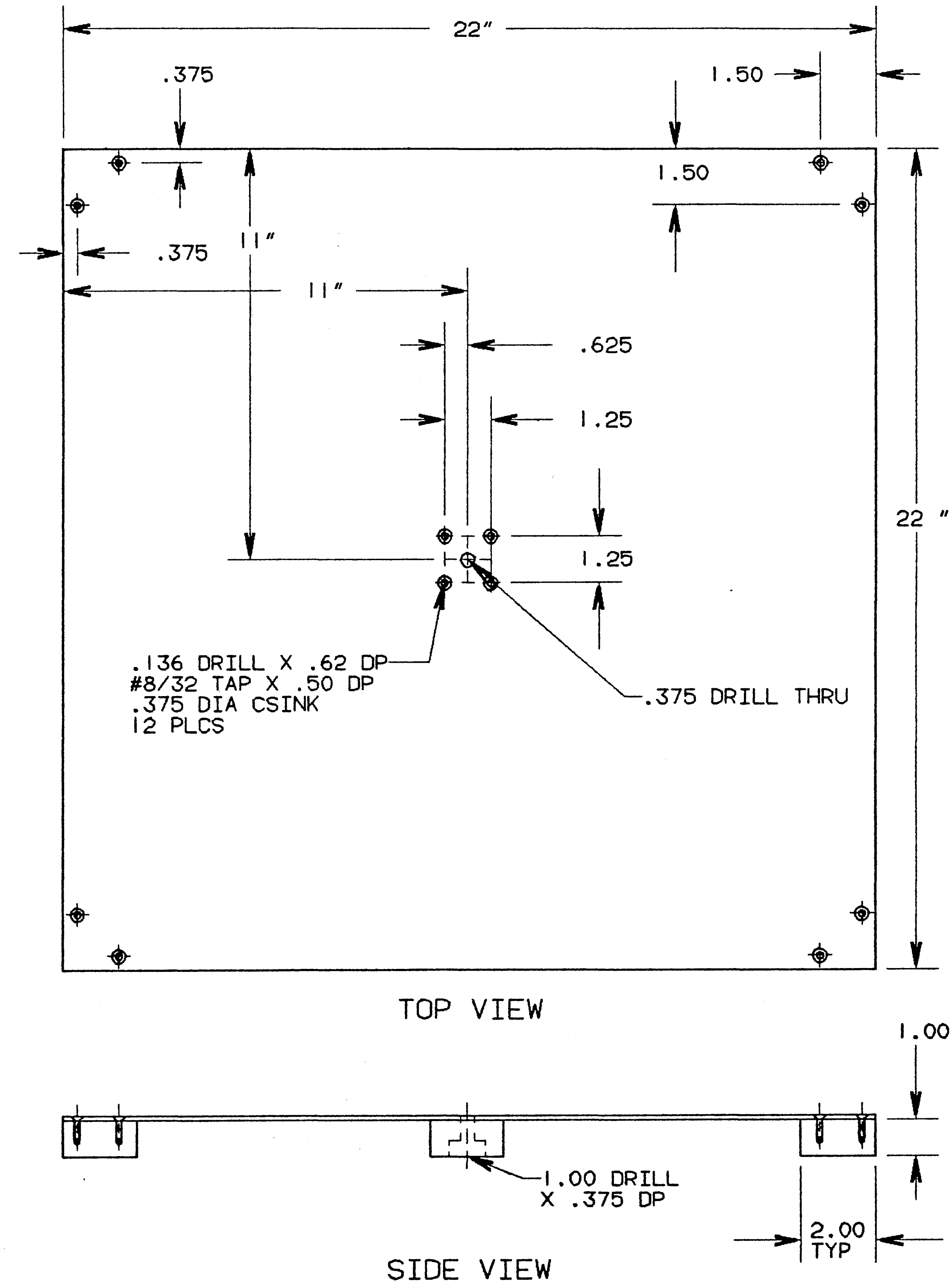




APPENDIX H

DEMONSTRATION PLATFORM DRAWINGS
AND ASSEMBLY PHOTOGRAPH

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/4
7075 ALUMINUM	BASE	METZNER	DATE: 11-86

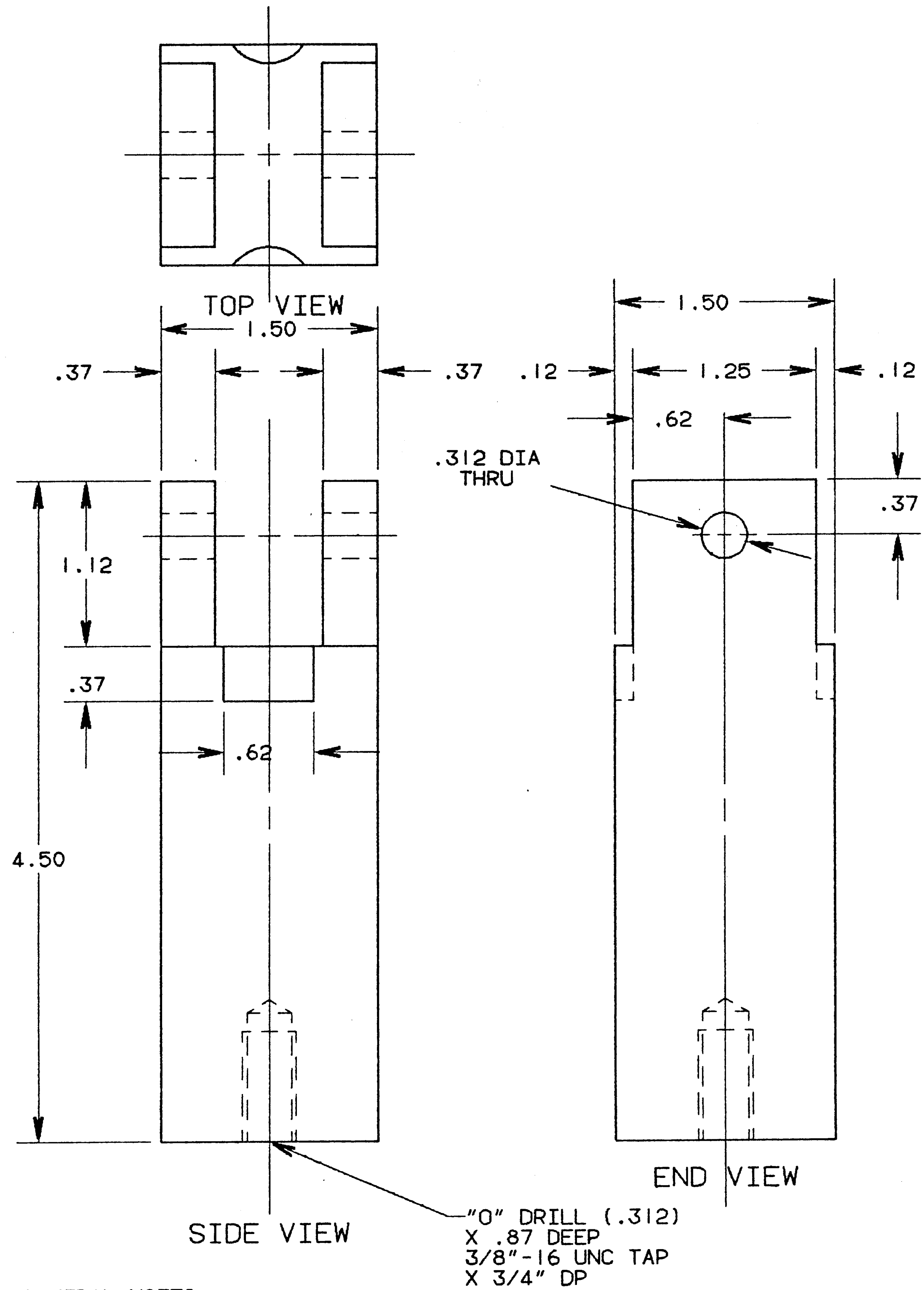


GENERAL NOTES:

DIMENSION TOL: ±.0005

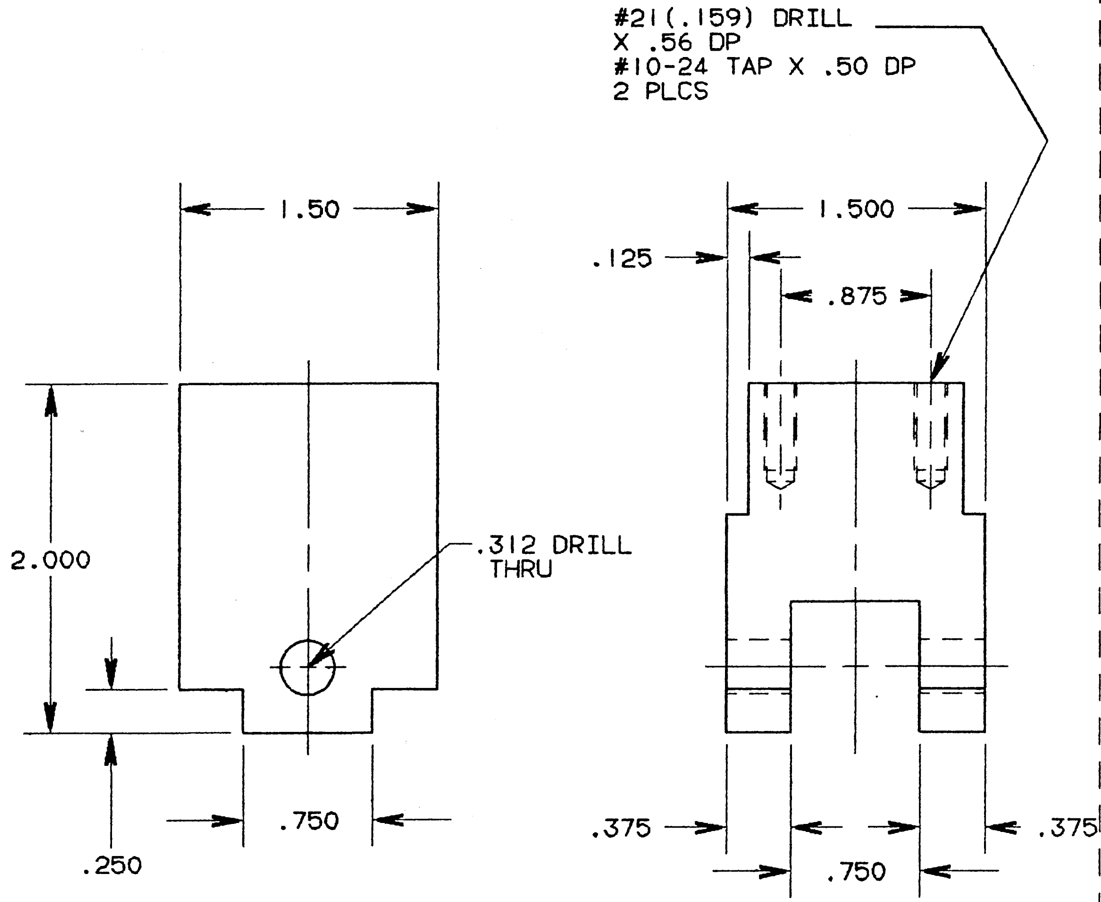
MATL=.125 THK ALUMINUM TOOL PLATE

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	ATTACHMENT, BASE	METZNER	DATE: 11-86



GENERAL NOTES:
DIMENSION TOL: ±.0005

MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	BASE, ASSEMBLY	METZNER	DATE: 11-86

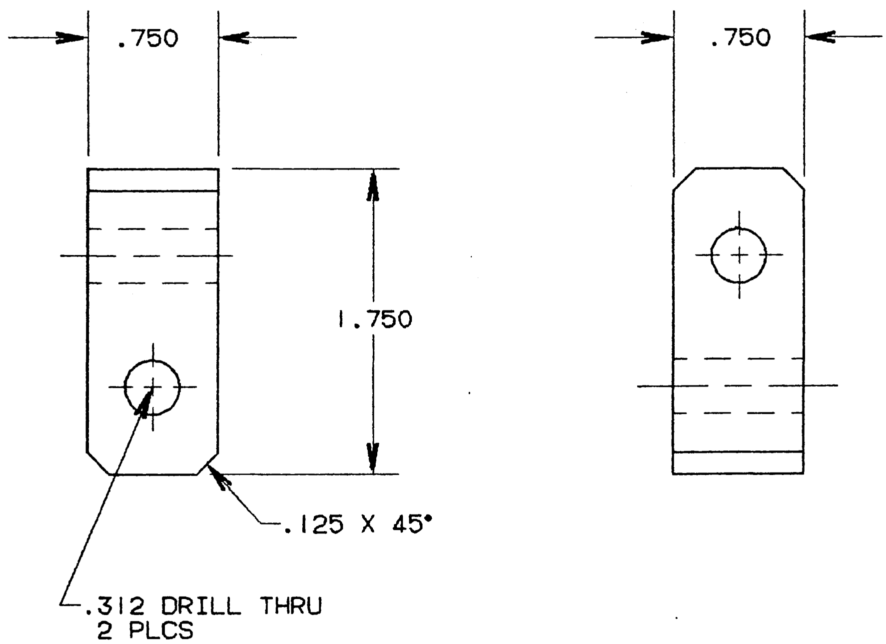


SIDE VIEW

END VIEW

GENERAL NOTES:
DIMENSION TOL: ±.0005

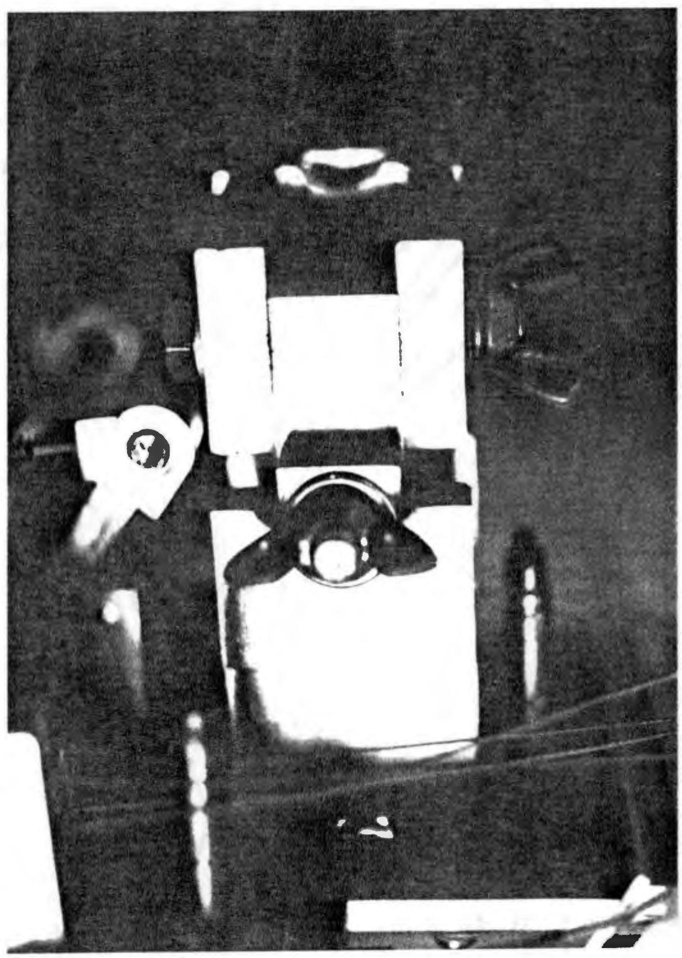
MATERIAL:	DESCRIPTION:	DRAWN BY:	SCALE= 1/1
7075 ALUMINUM	BASE, ASSEMBLY	METZNER	DATE: 11-86



SIDE VIEW

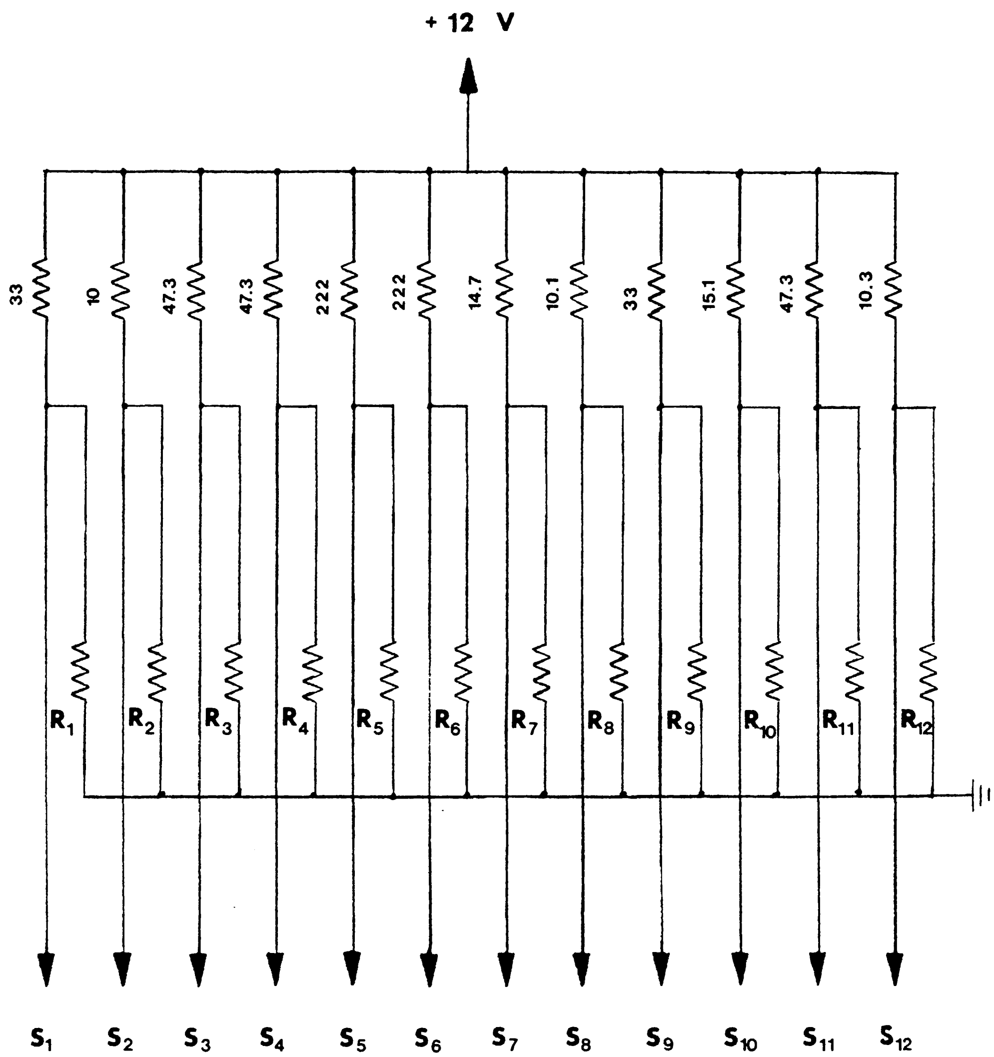
END VIEW

GENERAL NOTES:
DIMENSION TOL: ±.0005



APPENDIX I

SENSOR CIRCUIT DIAGRAM

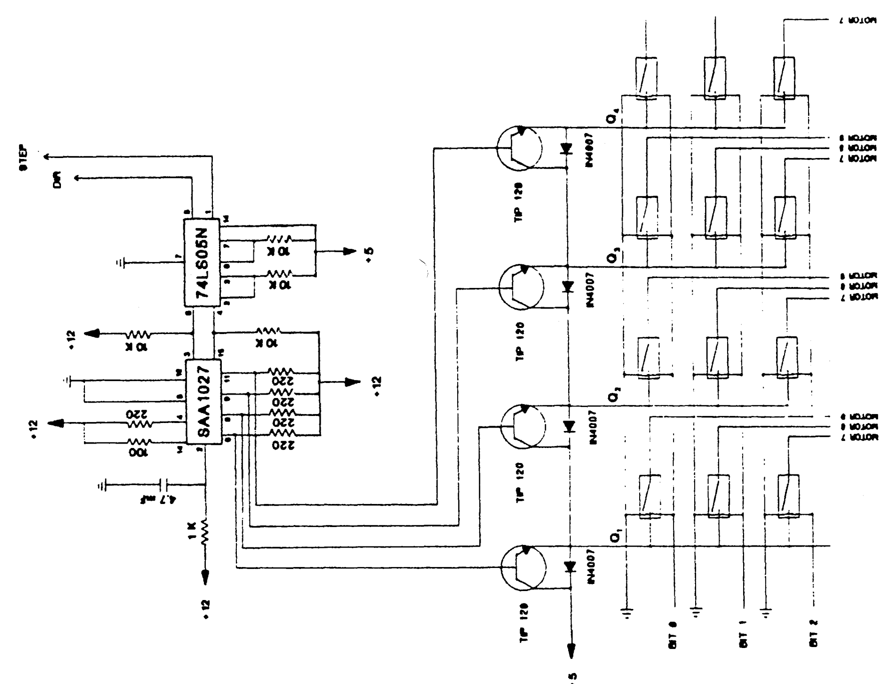


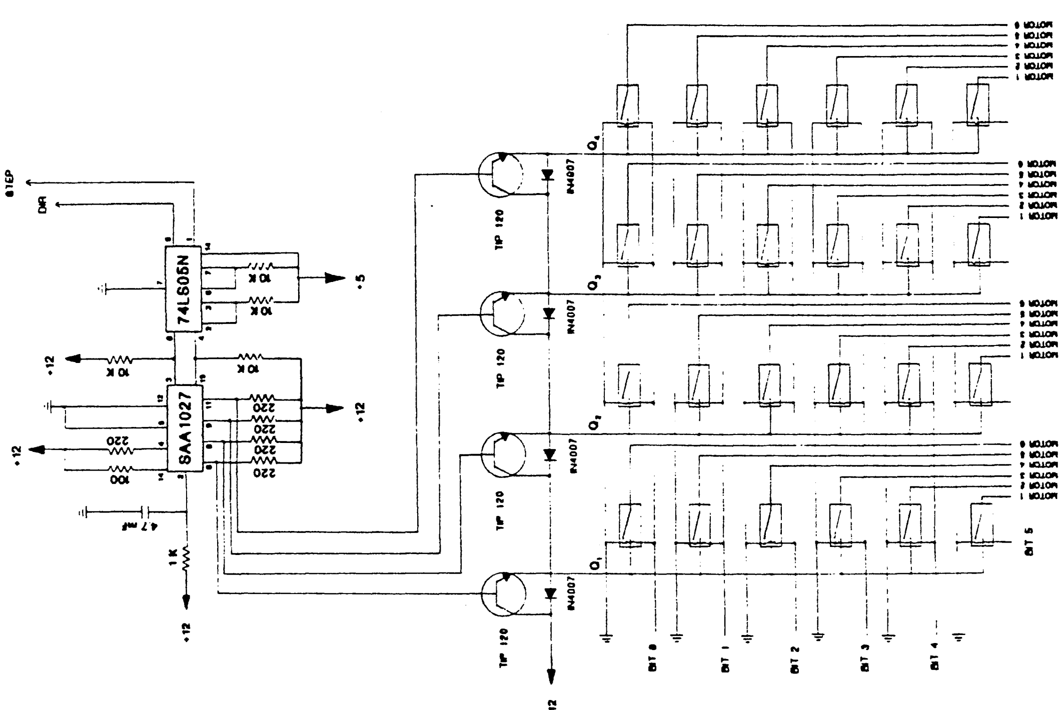
APPROXIMATE SENSOR RESISTANCE VALUES:

Rs1 -- 42.6K	Rs7 -- 32.0K
Rs2 -- 37.6K	Rs8 -- 44.8K
Rs3 -- 58.0K	Rs9 -- 48.8K
Rs4 -- 65.0K	Rs10 -- 50.8K
Rs5 -- 550K	Rs11 -- 30.8K
Rs6 -- 860K	Rs12 -- 52.0K

APPENDIX J

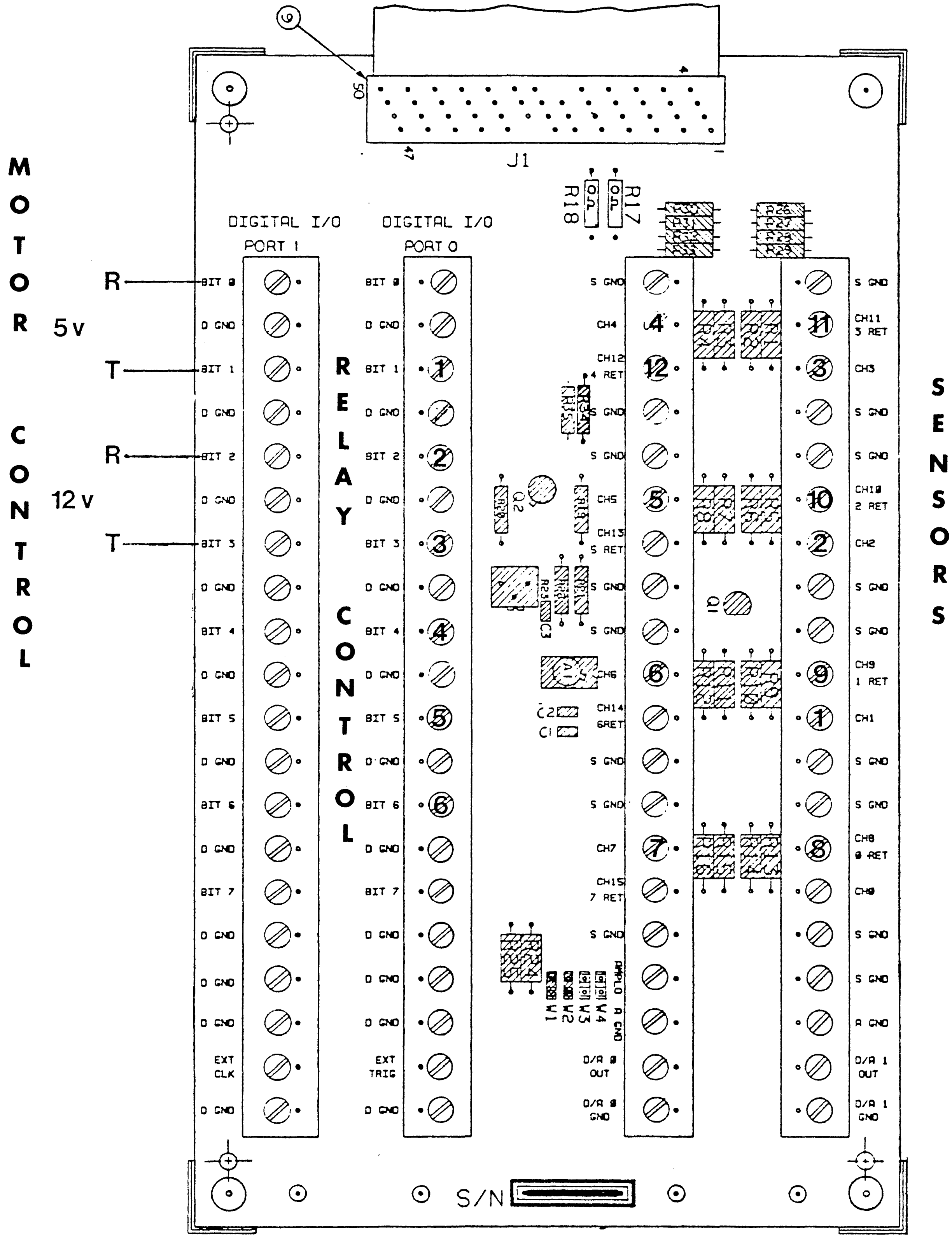
CONTROL CIRCUIT DIAGRAM


STEP
DIR
74LS05N
SAA1027
10 K
220
100
1 K
+12
+5
TIP 120
IN4007
BIT 0
BIT 1
BIT 2


STEP
DIR
74LS05N
SAA1027
10 K
220
100
1 K
+12
+5
TIP 120
IN4007
BIT 0
BIT 1
BIT 2
BIT 3
BIT 4
BIT 5

APPENDIX K

SCREW TERMINAL CONNECTIONS


MOTOR CONTROL
5 v
12 v
R
T
R
T
RELAY CONTROL
SENSORS
DIGITAL I/O PORT 1
DIGITAL I/O PORT 0
J1
S/N
EXT CLK
EXT TRIG
D/A 0 OUT
D/A 1 OUT

APPENDIX L

LIST OF CONTROL ALGORITHM

```
/*-----------------------------------------------------------------------*/
/*                                                                       */
/*  Thesis title:  DESIGN OF A DEXTEROUS MECHANICAL HAND                 */
/*  Approved by:  DR. A. H. SONI                                         */
/*  Written by: TOMMY J. METZNER                                         */
/*  Date completed:  DECEMBER 4, 1987                                    */
/*                                                                       */
/*                                                                       */
/*  This program is the control algorithm for the OSU Dexterous Hand.  The */
/*  hand is controlled by an IBM-AT in conjunction with a DATA TRANSLATION */
/*  A/D converter, series 2805.  Nine linear actuators are activated     */
/*  according to the feedback from twelve independent, pressure sensitive */
/*  sensors located on the palm and finger links.  The sensors provide   */
/*  different voltages depending on the applied pressure.  The voltages are */
/*  sent to the screw terminal and hence to the A/D converter.  The sensors */
/*  are read in a sequential manner which provides accurate information that */
/*  allows for proper control responses to the read voltages.            */
/*                                                                       */
/*  The algorithm is divided into two sections.  One section allows for a */
/*  menu driven selection of pre-determined objects which are grasped.  The */
/*  other section is for teach pendent application.  The user specifies the */
/*  the desired force applied to an object and the subsequent control    */
/*  movements are stored for later reference.                            */
/*                                                                       */
/*       ------------------------------------------------------------    */
/*                                                                       */
/*  VARIABLE LIST--                                                      */
/*                                                                       */
/*    Sensors:  PL ---- Left palm sensor.                                */
/*              PR ---- Right palm sensor.                               */
/*              S11 --- First finger and first link sensor.              */
/*              SL13 -- First finger and left side of the third link.    */
/*              SR13 -- First finger and right side of the third link.   */
/*              S21 --- Second finger and first link.                    */
/*              S22 --- Second finger and second link.                   */
/*              SL23 -- Second finger and left side of third link.       */
/*              SR23 -- Second finger and right side of third link.      */
/*              S31 --- Third finger and first link.                     */
/*              S32 --- Third finger and second link.                    */
/*              SL33 -- Third finger and left side of third link.        */
/*              SR33 -- Third finger and right side of third link.       */
/*                                                                       */
/*    Actuators:  A11 --- Actuates first link of the first finger.       */
/*                A123 -- Actuates second & third link of the first finger. */
/*                A21 --- Actuates first link of the second finger.      */
/*                A223 -- Actuates second & third link of the second finger. */
/*                A31 --- Actuates first link of the third finger.       */
/*                A323 -- Actuates second & third link of the third finger. */
/*                ROT1 -- Controls rotation of first finger.             */
/*                ROT2 -- Controls rotation of second finger.            */
/*                ROT3 -- Controls rotation of third finger.             */
/*                                                                       */
/*   OTHER USEFUL VARIABLES--                                            */
/*                                                                       */
/*     dir -- Direction of motor rotation.                               */
/*     fgr -- Finger selection.                                          */
/*     fgr_lnk -- Finger link selection.                                 */
/*     file1 -- Motion code storage array.                               */
/*     force1 -- Reference force exerted on object.                      */
/*     force2 -- Actual force exerted on object.                         */
/*     port -- Digital I/O port select.                                  */
/*     spd -- Speed of motor rotation.                                   */
/*     time -- Time variable.                                            */
/*                                                                       */
/*-----------------------------------------------------------------------*/
```

```
#include <dos.h>
#include <stdio.h>
#include <math.h>

#define BLS  printf("3[2J");
#define DELAY  for(i=0; i<=30000; ++i);

/*  ASSIGN ADDRESSES FOR DATA TRANSLATION  */

#define CMDREG 0x2ed          /* TI-101 */
#define DATREG 0x2ec          /* TI-100 */
#define STREG  0x2ed          /* TI-101 */

/*  DEFINE CONSTANTS FOR D/A BOARD  */

#define CMDMSK 0x4
#define REDMSK 0x5
#define WRTMSK 0x2
#define GAIN    0x0
#define REDNOW  0xc

#define CCLEAR  0x1
#define CSTOP   0xf
#define CSOUT   0x5
#define CDIOOUT 0x7
#define DIOPORT0 0x0
#define DIOPORT1 0x1

/*  DEFINE MOTOR CONTROL CONSTANTS FOR 5 & 12 VOLT MOTORS  */

#define BWRD5   0x4
#define FWRD5   0x5
#define ZEROB5  0x0
#define ZEROF5  0x1
#define BWRD12  0x10
#define FWRD12  0x18
#define ZEROB12 0x0
#define ZEROF12 0x8

/*  DEFINE SENSOR CHANNEL CONSTANTS  */

#define CHANL1  0x1             /* S11 */
#define CHANL2  0x2             /* S12 */
#define CHANL3  0x3             /* SL13 */
#define CHANL4  0x4             /* SR13 */
#define CHANL5  0x5             /* S21 */
#define CHANL6  0x6             /* S22 */
#define CHANL7  0x7             /* SL23 */
#define CHANL8  0x8             /* SR23 */
#define CHANL9  0x9             /* PL */
#define CHANL10 0xa             /* PR */
#define CHANL11 0xb             /* S31 */
#define CHANL12 0xc             /* S33 */

/*  DEFINE MOTOR SELECTION CONSTANTS  */

#define A11  0x2
#define A123 0x4
#define A21  0x8
#define A223 0x10
#define A31  0x20
#define A323 0x40
#define ROT1 0x2
#define ROT2 0x4
#define ROT3 0x8
```

```
extern int getch();
int time;
int ctr1,ctr2,ctr3;

main()
{
 char data,file_name[15],motion,obj,resp,select;
 int control,dir,flag,fgr,fgr_lnk,file1[500],i,key,spd;
 float force1,force2;
 short int port;

 FILE *fp;
 resp = 'y';
 BLS;
 SET();
 printf("%s%s%s%s%s%s%s",
        "This is the control program for the OSU DEXTEROUS HAND.",
        "The program is divided into two sections. One is for",
        "demonstration purposes which has several predetermined",
        "objects that are grasped when selected.  The other section",
        "allows for the user to grasp a desired object.  At this",
        "point, the user can specify a teach pendent or computer",
        "mode.");
 printf("Press 'RETURN' to continue......");
 key = getch();
 while (resp == 'y' || resp == 'Y')
 {
  BLS;
  printf("3[37m%s%s%s3[7;37m%s",
         "Select one of the following:",
         "1 --> DEMONSTRATION",
         "2 --> TEACH PENDENT",
         "Enter --> ");
  scanf("%s",&select);
  if (select == '1')
  {
   BLS;
   printf("3[37m%s%s%s",
          "Experimentation with five objects that have different physical",
          "properties has shown that obviously many grasps are possible.",
          "The most appropriate grasps have been selected for demonstration.");
  }

  /*  DEMONSTRATION MODE  */

  while (select == '1')
  {
   SET_ZERO();
   selection:  printf("3[37m%s%s%s%s%s%s3[7;37m%s",
                      "Select one of the following objects: ",
                      "1 --> STYROFOAM CUP",
                      "2 --> EGG",
                      "3 --> ALUMINUM CAN",
                      "4 --> BALL",
                      "Enter --> ");
   scanf("%s",&obj);

   /*  PLAYBACK MOTION SEQUENCE ON FILE  */

   switch (obj)
   {
    case '1':
             fp = fopen("styro_cup,dat", "r");
             READ_FILE(file1,"styro_cup,dat");
             fclose(fp);
```

```
            REPLAY(file1);
            break;

  case '2':
            fp = fopen("egg.dat", "r");
            READ_FILE(file1,"egg.dat");
            fclose(fp);
            REPLAY(file1);
            break;

  case '3':
            fp = fopen("alum_can.dat", "r");
            READ_FILE(file1,"alum_can.dat");
            fclose(fp);
            REPLAY(file1);
            break;

  case '4':
            fp = fopen("ball.dat", "r");
            READ_FILE(file1,"ball.dat");
            fclose(fp);
            REPLAY(file1);
            break;

  default:
           BLS;
           printf("3[37m%s%s3[7;37m%s",
                  "You did not enter one of the listed numbers.",
                  "Do you want to re-enter an object? (y-yes, n-no)",
                  "Enter --> ");
           scanf("%s",&resp);
           if (resp == 'y' || resp == 'Y')
           {
            BLS;
            select = '1';
           }
           else select = '0';

 }
 if (obj == '1' || obj == '2' || obj == '3' || obj == '4' || obj == '5')
 {
  printf("3[37m%s3[7;37m%s",
         "Do you want to try something else? (y-yes,n-no)",
         "Enter (y or n) --> ");
  scanf("%s",&resp);
  if (resp == 'y' || resp == 'Y')
  {
   select = '1';
   BLS;
  }
  else
  select = '0';
 }
}
BLS;
if (select == '0')
{
 printf("3[37m%s%s%s3[7;37m%s",
        "Do you want to enter ",
        "TEACH PENDENT ",
        "mode? (y-yes, n-no)",
        "Enter ( y or n ) --> ");
 scanf("%s",&resp);
 if (resp == 'y' || resp == 'Y')
   select = '2';
 else
```

```
   select = '0';
}

/*  TEACH PENDENT MODE  */

if (select == '2')
{
 BLS;

 printf("3[37m%s%s%s",
        "The hand is now in the teach pendent mode.  The user has the",
        "option to control the actuation of the fingers or let the",
        "computer control the actuation.");
}
while (select == '2')
{
 SET_ZERO();
 select2:  printf("3[37m%s%s%s3[7;37m%s",
            "Please select a control mode: ",
            "1 --> USER",
            "2 --> COMPUTER",
            "Enter (1 or 2) --> ");
            scanf("%s",&select);
   BLS;
   printf("3[37m%s%s%s%s",
          "Enter the name of the file that you want to store",
          "the data in.",
          "Example:  Filename.dat",
          "Enter --> ");
   scanf("%s",&file_name);
   BLS;
   printf("Control Begins.....");
   DELAY;
   BLS;
   printf("Place object on palm.....");
   printf("Press 'ENTER'");
   key = getch();
   if ( key != 13 )
   {
    BLS;
    printf("Press 'ENTER' when ready.....");
    key = getch();
 }

 /*   MANUAL CONTROL   */

 switch (select)
 {
  case '1':
            PICTURE();
            control = 0;
            ctr1 = 0;
            ctr2 = 0;
            ctr3 = 0;
            flag = 0;

            while(control == 0)
            {
             SET_ZERO();
             if(ctr1 != 0) BLS;
             printf("%s%s%s%s",
                    "Enter type of motion desired:",
                    "1 --> SIDE-TO-SIDE",
                    "2 --> OPEN-AND-CLOSE",
                    "Enter (1 or 2) --> ");
             scanf("%s",&motion);
```

```
if(motion == '1')
{
 printf("%s%s",
        "Enter number of finger in which control is desired:",
        "--> ");
 scanf("%d",&fgr);
 fgr = fgr+40;
 file1[++ctr1] = fgr;
 MOTOR(fgr);
 printf("%d",file1[ctr1]);
}
else
{
 printf("%s%s",
        "Enter the number of the finger and link:",
        "Enter --> ");
 scanf("%d",&fgr_lnk);
 file1[++ctr1] = fgr_lnk;
 MOTOR(fgr_lnk);
}
BLS;
printf("%s%s%s%s",
       "Enter rate of accuation:",
       "100% ......fastest",
       "  1% ......slowest",
       "Enter --> ");
scanf("%d",&spd);

time = 12000/spd;
/* speed conversion */

if (ctr1 == 1)
{
 BLS;
 printf("%s%s%s%s%s",
        "The arrow keys are used to control actuation.",
        "up arrow   OPEN",
        "dwn arrow  CLOSE",
        "--> ROTATE RIGHT",
        "<-- ROTATE LEFT");
 printf("Depress 's' to save a desired position.");
 printf("Press 'RETURN' to continue.....");
 key = getch();
}
BLS;
printf("Begin Actuation.....(Press 's' to save)");
printf("Working......");
data = 's';
while(data == 's' || data == 'S')
{
 key = getch();
 if (Key == 72)
 {
  dir = FWRD5;
MOVE(dir,time);
  file1[++ctr1] = 1;
  flag = 1;
 }
 if (Key == 80)
 {
  dir = BWRD5;
MOVE(dir,time);
  file1[++ctr1] = 0;
  flag = 1;
 }
 if (Key == 75)
```

```
              {
               dir = BWRD12;
               MOVE(dir,time);
               file1[++ctr1] = 0;
              }
              if (key == 77)
              {
               dir = FWRD12;
               MOVE(dir,time);
               file1[++ctr1] = 1;
              }
              if (key == 83 || key == 115) data = 'a';
             }
            printf("%s%s",
                   "Do you want to control another finger? (y-yes, n-no)",
                   "Enter (y or n) --> ");
            scanf("%s",&resp);
            if (resp == 'y' || resp == 'Y')
               control = 0;
            else
               control = 1;
           }
           WRITE_FILE(ctr1,file1,file_name);
           break;

case '2':
           BLS;
           printf("Enter a gripping force between 0.0 and 1.1 lbf:");
           printf("Enter --> ");
           scanf("%f",&force1);
           BLS;
           printf("Actuation begins.....");
           SENSOR_READ(CHANL9,force2);

           /*   CHECK ORIENTATION OF OBJECT   */

           while (force2 < force1)
           {
            CNTRL(CHANL9,force2,fgr_lnk,dir,port,time);
            CNTRL(CHANL1,force2,fgr_lnk,dir,port,time);
            CNTRL(CHANL2,force2,fgr_lnk,dir,port,time);
            CNTRL(CHANL3,force2,fgr_lnk,dir,port,time);
            CNTRL(CHANL4,force2,fgr_lnk,dir,port,time);
            CNTRL(CHANL5,force2,fgr_lnk,dir,port,time);
            CNTRL(CHANL6,force2,fgr_lnk,dir,port,time);
            CNTRL(CHANL7,force2,fgr_lnk,dir,port,time);
            CNTRL(CHANL8,force2,fgr_lnk,dir,port,time);
            CNTRL(CHANL10,force2,fgr_lnk,dir,port,time);
            CNTRL(CHANL11,force2,fgr_lnk,dir,port,time);
            CNTRL(CHANL12,force2,fgr_lnk,dir,port,time);
           }
           break;

default:
           printf("Re-enter a control mode: ");
           goto select2;
}
BLS;
printf("3[37m%s3[7;37m%s3[37m",
       "Do you want to RE-PLAY the sequence?",
       "Enter (y or n) --> ");
scanf("%s",&resp);
if (resp == 'y' || resp == 'Y')
{
 READ_FILE(file1,file_name);
 REPLAY(file1);
```

```
    BLS;
    printf("3[37m%s3[7;37m%s",
            "Do you want to change modes?",
            "Enter (y or n) --> ");
    scanf("%s",&resp);
    if (resp == 'n' || resp == 'N')
     select = '0';
    else
    {
     BLS;
     select = '2';
    }
   }
   BLS;
   printf("3[37m%s3[7;37m%s",
           "Do you want to rerun this program?",
           "Enter (y or n) --> ");
   scanf("%s",&resp);
   if (resp == 'n' || resp == 'N')
   {
    BLS;
    printf("3[37m");
   }
 }
}

/*-------------------------- F U N C T I O N S ----------------------------*/


/*  CHECK FORCE TO DETERMINE WHICH FINGER TO MOVE   */

ARGU(channel,force2,fgr)

short int channel;
int fgr;
float force2;
{

 float force3;
 short int channel2;

 force3 = force2;

 if (channel == 3 || channel == 4)
 {
  if (channel == 3) channel2 = CHANL4;
  else channel2 = CHANL3;
  SENSOR_READ(channel2,force2);
  while (force2 > (force3 + 1.0) || force2 < (force3 - 1.0))
  {
   MOTOR(fgr);
   MOVE(fgr,time);
  }
 }

 if (channel == 7 || channel == 8)
 {
  if (channel == 7) channel2 = CHANL8;
  else channel2 = CHANL7;
  SENSOR_READ(channel2,force2);
  while (force2 > (force3 + 1.0) || force2 < (force3 - 1.0))
  {
   MOTOR(fgr);
   MOVE(fgr,time);
  }
```

```
 }

 if (channel == 11 || channel == 12)
 {
  if (channel == 11) channel2 = CHANL12;
  else channel2 = CHANL11;
  SENSOR_READ(channel2,force2);
  while (force2 > (force3 + 1.0) || force2 < (force3 - 1.0))
  {
   MOTOR(fgr);
   MOVE(fgr,time);
  }
 }
}
                    /*--------------------------------------*/

/*  CONTROL FINGER BY CLOSED-LOOP FEEDBACK  */

CNTRL(channel,force1,fgr,dir,port,time)

float force1;
int dir,fgr,time;
short int channel,port;
{

 float force2;

 SENSOR_READ(channel,force2);
 if (force2 < force1)
 {
  dir = BWRD5;
  while (force2 < force1)
  {
   MOTOR(fgr);
   MOVE(dir,time);
   SENSOR_READ(channel,force2);
   ARGU(channel,force2,fgr);
  }
 }
 else
 {
  dir = FWRD5;
  while (force2 > force1)
  {
   MOTOR(fgr);
   MOVE(dir,time);
   SENSOR_READ(channel,force2);
   ARGU(channel,force2,fgr);
  }
 }
}
                    /*----------------------------------------*/

/*  SELECT APPROPRIATE MOTOR  */

MOTOR(fgr)

int fgr;
{
short int mtr;
int dump;

 if( fgr == 11 ) mtr = A11;
 else
```

```
 {
  if( fgr == 21 ) mtr = A21;
  else
  {
   if( fgr == 31 ) mtr = A31;
   else
   {
    if ( fgr == 13 ) mtr = A123;
    else
    {
     if ( fgr == 23 ) mtr = A223;
     else
     {
      if ( fgr == 33 ) mtr = A323;
      else
      {
       if ( fgr == 41 ) mtr = ROT1;
       else
       {
        if ( fgr == 42 ) mtr = ROT2;
        else mtr = ROT3;
       }
      }
     }
    }
   }
  }
 }
 outp(CMDREG,CSTOP);
 dump = inp(DATREG);
 while(!(CMDMSK & inp(STREG)));
 outp(CMDREG,CCLEAR);
 while(!(CMDMSK & inp(STREG)));
 outp(CMDREG,CDIOOUT);
 while(WRTMSK & inp(STREG));
 outp(DATREG,DIOPORT0);
 while(WRTMSK & inp(STREG));
 outp(DATREG,mtr);
}

                /*---------------------------------------*/

/*  MOVE FINGER  */

MOVE(dir,time)

int dir,time;
{
 int dump,i,zero;

 outp(CMDREG,CSTOP);
 dump = inp(DATREG);
 while(!(CMDMSK & inp(STREG)));
 outp(CMDREG,CCLEAR);
 while(!(CMDMSK & inp(STREG)));
 outp(CMDREG,CDIOOUT);
 while(WRTMSK & inp(STREG));
 outp(DATREG,DIOPORT1);
 while(WRTMSK & inp(STREG));
 outp(DATREG,dir);
 for(i=0; i<=time; ++i);
 outp(CMDREG,CDIOOUT);
 while(WRTMSK &inp(STREG));
 outp(DATREG,DIOPORT1);
 while(WRTMSK & inp(STREG));
 if(dir < 6)
```

```
 {
  if (dir == FWRD5)
    zero = ZEROF5;
  else
    zero = ZEROB5;
 }
 else
 {
  if (dir == FWRD12)
    zero = ZEROF12;
  else
    zero = ZEROB12;
 }
 outp(DATREG,zero);
 for (i=0; i<=time; i++);
}

                   /*---------------------------------------*/

/*  DISPLAY A PICTURE OF THE HAND DURING TEACH PENDENT MODE   */

PICTURE()
{
 BLS;
 printf("%s3[32m\%s%s",
        "                            OSU DEXTEROUS HAND",
        "                                  __    __ ",
        "                                 :  :  :  :");
 printf("3[37m%s3[32m%s3[37m%s3[32m%s%s%s%s",
        " <-- Finger 2 ",
        "                                 :  :  :  :",
        "                   Finger 1 -->",
        " :__:  :__:",
        "                                 :__:  :__:",
        "                                 :  :  :  :",
        "                                 :  :  :  :");

 printf("%s%s%s%s",
        "                                 :__:  :__:",
        "                           ________________ ",
        "                           :               :",
        "                           :     ______    :",
        "                           :     :     :   :");

 printf("3[37m%s3[32m%s%s%s3[37m%s3[32m%s",
        "             Palm -->","    :     :______:   :",
        "                           :               /___________",
        "                           :             /_____:_:_____:",
        "  <-- Finger 3",
        "                           :__________ / ");

 printf("3[37m");
}

                   /*---------------------------------------*/

/*  READ A SAVED FILE FOR REPEAT OF MOTIONS   */

READ_FILE(file1,name)

int file1[200];
char name[15];
{
 int k,n;
 FILE *fp;

 k = 1;
```

```
 fp = fopen(name, "r");
 fscanf(fp,"%d",&n);

 file1[K] = n;

 while (K != (n+1))
 {
  ++K;
  fscanf(fp,"%d",&file1[K]);
 }
 fclose(fp);
}

                    /*------------------------------------*/

/*  REPLAY A SET OF STORED MOTION CODES   */

REPLAY(file1)

int file1[200];
{
 int fgr,fgr_lnk,j,key,no_elm,time;

 no_elm = file1[1];
 j = 2;
 time = 500;
 while(j < no_elm)
 {
 key = getch();
  if (file1[j] == 11 || file1[j] == 13 || file1[j] == 21)
  {
   fgr_lnk = file1[j];
   while(file1[j+1] == 0 || file1[j+1] == 1)
   {
    if (file1[j+1] == 0)
    {
     MOTOR(fgr_lnk);
     MOVE(BWRD5,time);
     ++j;
    }
    else
    {
     MOTOR(fgr_lnk);
     MOVE(FWRD5,time);
     ++j;
    }
   }
  }
  if (file1[j] == 23 || file1[j] == 31 || file1[j] == 32)
  {
   fgr_lnk = file1[j];
   while(file1[j+1] == 0 || file1[j+1] == 1)
   {
    if (file1[j+1] == 0)
    {
     MOTOR(fgr_lnk);
     MOVE(BWRD5,time);
     ++j;
    }
    else
    {
     MOTOR(fgr_lnk);
     MOVE(FWRD5,time);
     ++j;
    }
   }
```

```
    }
    if (file1[j] == 41 || file1[j] == 42 || file1[j] == 43)
    {
     fgr = file1[j];
     while(file1[j+1] == 0 || file1[j+1] == 1)
     {
      if (file1[j+1] == 0)
      {
       MOTOR(fgr);
       MOVE(BWRD12,time);
       ++j;
      }
      else
      {
       MOTOR(fgr);
       MOVE(FWRD12,time);
       ++j;
      }
     }
    }
   ++j;
  }
 }

                    /*------------------------------------*/

/*  READ VOLTAGE OUTPUT FROM SENSOR  */

SENSOR_READ(channel,force2)

short int channel;
float force2;
{
   int dump,x,y;
   float dataval,volt;

   outp(CMDREG,CSTOP);
   dump = inp(DATREG);
   while(!(CMDMSK & inp(STREG)));
   outp(CMDREG,REDNOW);
   while(WRTMSK & inp(STREG));
   outp(DATREG,GAIN);
   while((WRTMSK & inp(STREG)));
   outp(DATREG,channel);
   while(!(REDMSK & inp(STREG)));
   x=inp(DATREG);
   while(!(REDMSK & inp(STREG)));
   y=inp(DATREG);
   dataval=y*256 + x;
   volt=10.07*(2.0*dataval/4096.0-1.0);
   if (channel == 1) force2 = (volt-1.11)/(-.0071);  /*  EQUATIONS ARE BASED   */
   if (channel == 2) force2 = (volt-1.51)/(-.008);   /*  ON THE ASSUMPTION     */
   if (channel == 3) force2 = (volt-2.4)/(-.013);    /*  THAT THE VOLTAGE      */
   if (channel == 4) force2 = (volt-2.4)/(-.013);    /*  VARIES LINEARLY WITH */
   if (channel == 5) force2 = (volt-2.4)/(-.013);    /*  RESPECT TO THE FORCE.*/
   if (channel == 6) force2 = (volt-2.4)/(-.013);    /*  THE EQUATIONS ARE     */
   if (channel == 7) force2 = (volt-1.51)/(-.008);   /*  DEVELOPED FROM CALI- */
   if (channel == 8) force2 = (volt-1.11)/(-.0071);  /*  BRATION CONSTANTS OB-*/
   if (channel == 9) force2 = (volt-2.6)/(-.018);    /*  TAINED EXPERIMENTALLY*/
   if (channel == 10) force2 = (volt-1.51)/(-.0071);
   if (channel == 11) force2 = (volt-.8)/(-.004);
   if (channel == 12) force2 = (volt-1.2)/(-.0081);
}

                    /*----------------------------------------*/
```

```
/* RESET A/D BOARD  */

SET()
{
 int v;

 outp(CMDREG,CSTOP);
 v = inp(DATREG);
 while(!(CMDMSK & inp(STREG)));
 outp(CMDREG,CCLEAR);
 while(!(CMDMSK & inp(STREG)));
 outp(CMDREG,CSOUT);
 while(WRTMSK & inp(STREG));
 outp(DATREG,2);
}

                    /*----------------------------------------*/

/*  SET ALL DIGITAL I/O PORTS TO ZERO  */

SET_ZERO()
{
int v;

 outp(CMDREG,CSTOP);
 v = inp(DATREG);
 while(!(CMDMSK & inp(STREG)));
 outp(CMDREG,CCLEAR);
 while(!(CMDMSK & inp(STREG)));
 outp(CMDREG,CDIOOUT);
 while(WRTMSK & inp(STREG));
 outp(DATREG,DIOPORT0);
 while(WRTMSK & inp(STREG));
 outp(DATREG,0);
 while(!(CMDMSK & inp(STREG)));
 outp(CMDREG,CDIOOUT);
 while(WRTMSK & inp(STREG));
 outp(DATREG,DIOPORT1);
 while(WRTMSK & inp(STREG));
 outp(DATREG,0);
}

                    /*----------------------------------------*/

/*  WRITE ALL FINGER MOVE CODES TO FILE  */

WRITE_FILE(ctr,file1,filename)

int ctr,file1[200];
char filename[15];

{
 int j;
 FILE *fp;

 fp = fopen(filename, "w");

 fprintf(fp,"%d",ctr);

 for(j=1;j<ctr;j++)
 {
  fprintf(fp,"%d",file1[j]);
 }
 fclose(fp);
}
```

VITA

Tommy Joe Metzner

Candidate for the Degree of

Master of Science

Thesis: DESIGN OF A DEXTEROUS MECHANICAL HAND

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Bartlesville, Oklahoma, June 29, 1962, the son of Ronald D. and Patsy J. Metzner. Married to Renee Dysinger on March 28, 1987.

Education: Graduated from Sooner High School, Bartlesville, Oklahoma, in May, 1980. Received Bachelor of Science Degree in Mechanical Engineering from Oklahoma State University in December, 1984. Completed requirements for the Master of Science Degree in Mechanical Engineering at Oklahoma State University in December, 1987.