California AHMCT Program
University of California at Davis
California Department of Transportation

A DESCRIPTON OF THE PHOTOGRAMMETRY TARGET PROJECT PREMARK PAINTING SYSTEM* (DRAFT)

Kenneth Sprott
Phillip Wong
Walter Nederbragt
Robert Olshausen
Bahram Ravani

AHMCT Research Report
UCD-ARR-94-09-09-01

Interim Report

July 14, 1994

* This work was supported by the California Department of Transportation (Caltrans) Advanced Highways and Maintenance and Construction Technology Program at UC-Davis and by the Federal Highway Administration (FHWA).

Technical Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle A DESCRIPTION OF THE PHOTOGRAMMETRY TARGET PROJECT PREMARK PAINTING SYSTEM	5. Report Date July 14, 1994
	6. Performing Organization Code
7. Author(s) Kenneth Sprott, Phillip Wong, Walter Nederbragt, Robert Olshausen, Bahram Ravani	8. Performing Organization Report No.
9. Performing Organization Name and Address Automated Highway Maintenance and Construction Technology Center Department of Mechanical and Aeronautical Engineering University of California, Davis, CA 95616	10. Work Unit No. (TRAIS)
	11. Contract or Grant 32J804
12. Sponsoring Agency Name and Address California Department of Transportation Sacramento, CA 94273-0001	13. Type of Report and Period Covered Interim Report 1992 - 1994
	14. Sponsoring Agency Code

15. Supplementary Notes

This study was conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration.

16. Abstract

This report provides a description of a premark painting system developed at UC, Davis. The system is designed to paint photogrammetry premarks used in the aerial surveys of roadways. Overall system descriptions are given for the premark painting system. A detailed description is given for each of the major subsystems and the support equipment. The proposed system operation is described. Recommendations for future uses of the system are made along with recommendations for development into a commercial product.

17. Key Words Highway, maintenance, construction, pavement Equipment, new technology, robotics, painting	18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

20. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 126	22. Price

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Abstract

This report provides a description of a premark painting system developed at UC Davis. The system is designed to paint photogrammetry premarks used in the aerial surveys of roadways. Overall system descriptions are given for the premark painting system. A detailed description is given for each of the major subsystems and the support equipment. The proposed system operation is described. Recommendations for future uses of the system are made along with recommendations for development into a commercial product.

Table of Contents

List of Figures

Disclaimer / Disclosure

The contents of this report reflect the views of the authors. The contents do not necessarily reflect the official views or policies of the STATE OF CALIFORNIA or the FEDERAL HIGHWAY ADMINISTRATION and the UNIVERSITY OF CALIFORNIA. This report does not constitute a standard, specification, or regulation.

This work was supported by Federal Highway Administration Contract No. ??????????????? and by the California Department of Transportation (Caltrans) Advanced Highway Maintenance and Construction Technology Program at UC-Davis.

1. Introduction

Photogrammetry is a method which uses aerial ground photos to survey roadways. This method requires ground target points, or premarks, as ground control calibration points. These ground target points are manually painted on the roadway by a survey crew. This operation is both time consuming for the survey crew and exposes them to traffic hazard. This report describes an automated system for painting the ground target points that reduces the cycle time and protects the crew from traffic hazards.

Caltrans uses 50 scale photogrammetric mapping to make maps of their roadways. These maps have a scale of 1"=50', and are made from aerial photographs which are taken at approximately 1"=250' scale. Ground located target points are used as ground control calibration points. These ground target points, or "premarks", are often placed at previously surveyed reference points. These premarks are generally painted every 450 feet and resemble a large "X" (see figure 1). Recently, Caltrans has been experimenting with a smaller ground target point, referred to as the abbreviated premark. This abbreviated premark is also a painted mark, resembling a small set of black and white bars. They are placed at 50 foot intervals between the larger premarks and allow the photogrammetrist to better estimate the road elevation between the large premarks. The large premarks take 15 -20 minutes to manually paint alongside the roadway. During this time, the survey crew is exposed to traffic hazard.

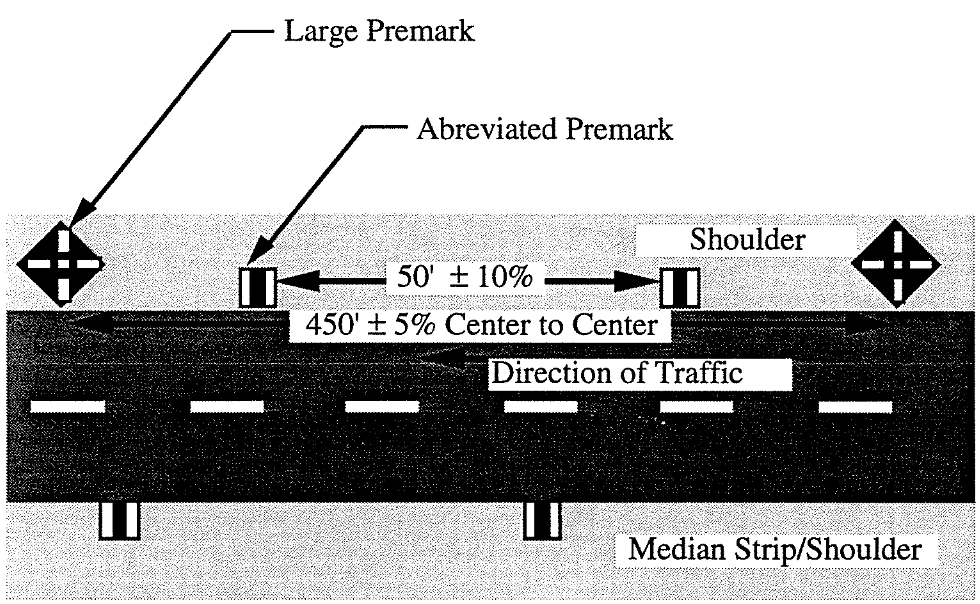


Figure 1: Premark and Abbreviated Premark Placement

This report introduces a prototype system for automating the placement of both the premark and the abbreviated premark. This system consists of two major subsystems; the Free Nozzle Gantry System (FNGS) that paints the large premark and the Linear Free Nozzle System (LFNS) that paints the abbreviated premark. These subsystems use common support equipment and are mounted in a fully self contained trailer for field operation. This report describes the two major subsystems, the system architecture, and the system operation.

2. System Description

The premark painting system consists of the Free Nozzle Gantry System (FNGS), the Linear Free Nozzle System (LFNS), and support and control equipment that are shared by both systems. These systems are mounted in an enclosed trailer that can be towed to the field site. System operation is carried out through the use of a hand held controller that can be placed in the cab of the towing vehicle. The system is designed to minimize the exposure of the survey crew to traffic hazards while meeting the premark painting requirements.

2.1 Free Nozzle Gantry System

The FNGS is designed to paint the large premarks on the highway shoulder at intervals of 450 feet. Its primary component is a gantry robot which provides positioning of a painting end effector in two independent directions. The end effector for the robot supports three independently operated paint guns on a platform that can rotate between two positions. The gantry robot is supported on a ball screw/air motor system that raises and lowers the gantry between a transportation and a painting position.

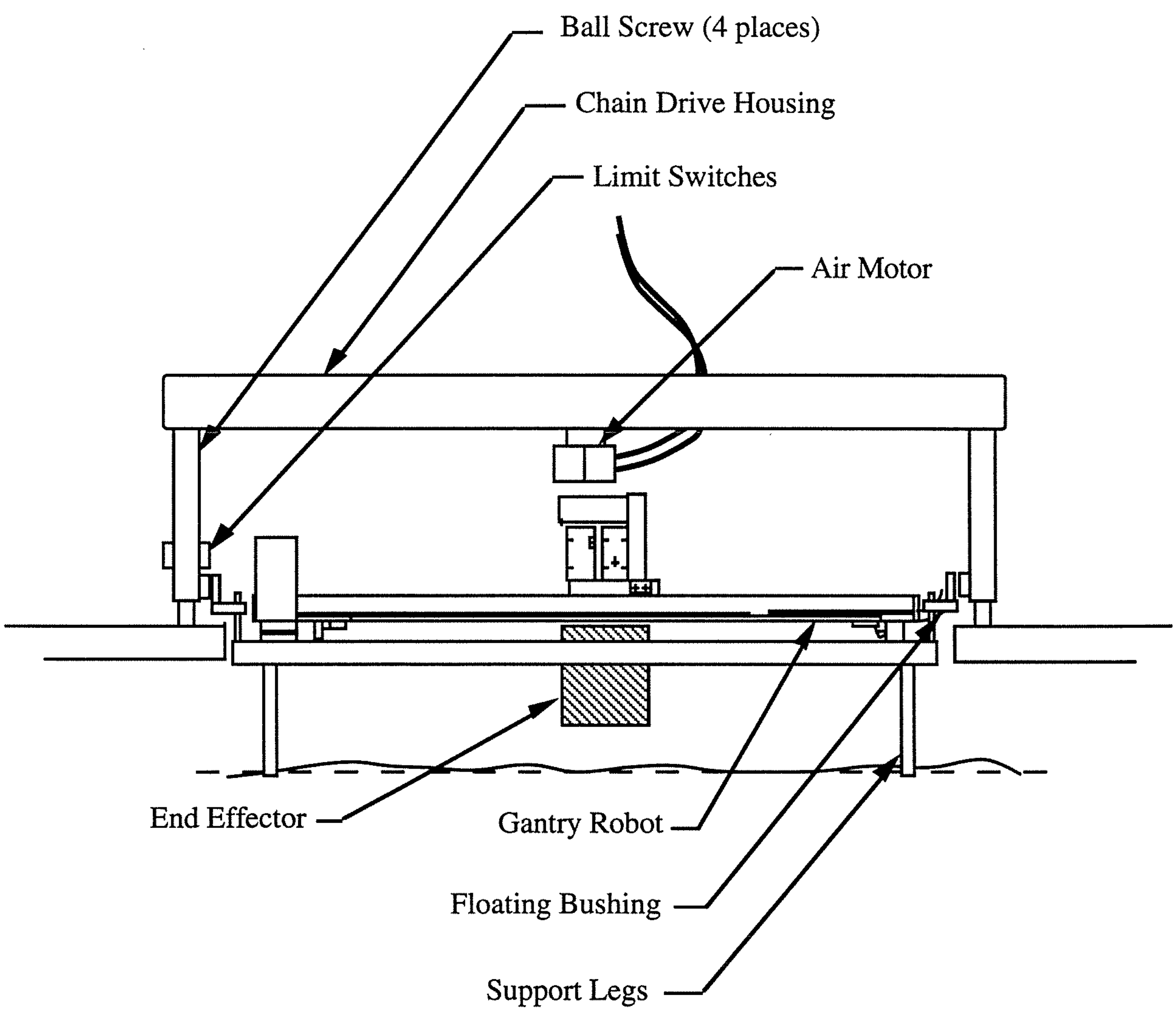


Figure 2: Free Nozzle Gantry

2.2 Linear Free Nozzle System

The LFNS is designed to paint the abbreviated premarks on the highway shoulder at intervals of 50 feet. It consists of three cam activated paint guns mounted to a pneumatic band cylinder. The pneumatic band cylinder provides gun translation in the painting direction. The rate of translation is controlled by pneumatic flow control valves on the band cylinder return ports. The band cylinder is mounted to a carriage that is raised and lowered between a travel position and a paint position by pneumatic cylinders.

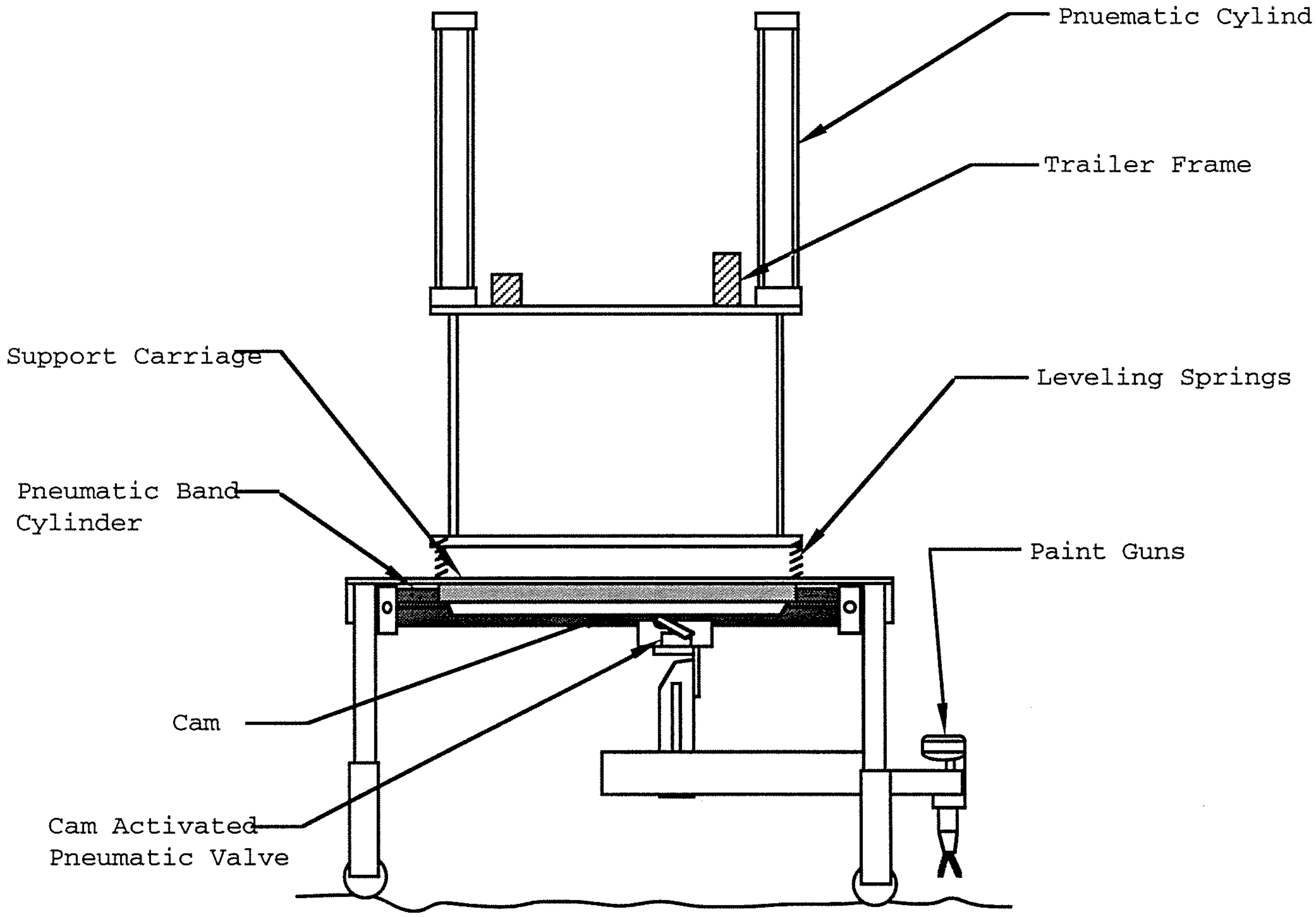


Figure 3: Linear Free Nozzle System

2.3 Trailer Mount

The FNG and LFN along with common support equipment are mounted in a fully enclosed trailer. The trailer is a modified 16 foot cargo trailer. The modifications consist of a hole cut into the trailer floor forward of the tandem axles for the FNGS; two small holes cut into the side and floor of the trailer at the back end; and two exhaust ducts through the roof provided for the support equipment. The trailer mounting of the premark system is designed to be completely self contained for field operations. No additional power or equipment are require to operate the system in the field. Figure 4 shows the internal layout of the trailer.

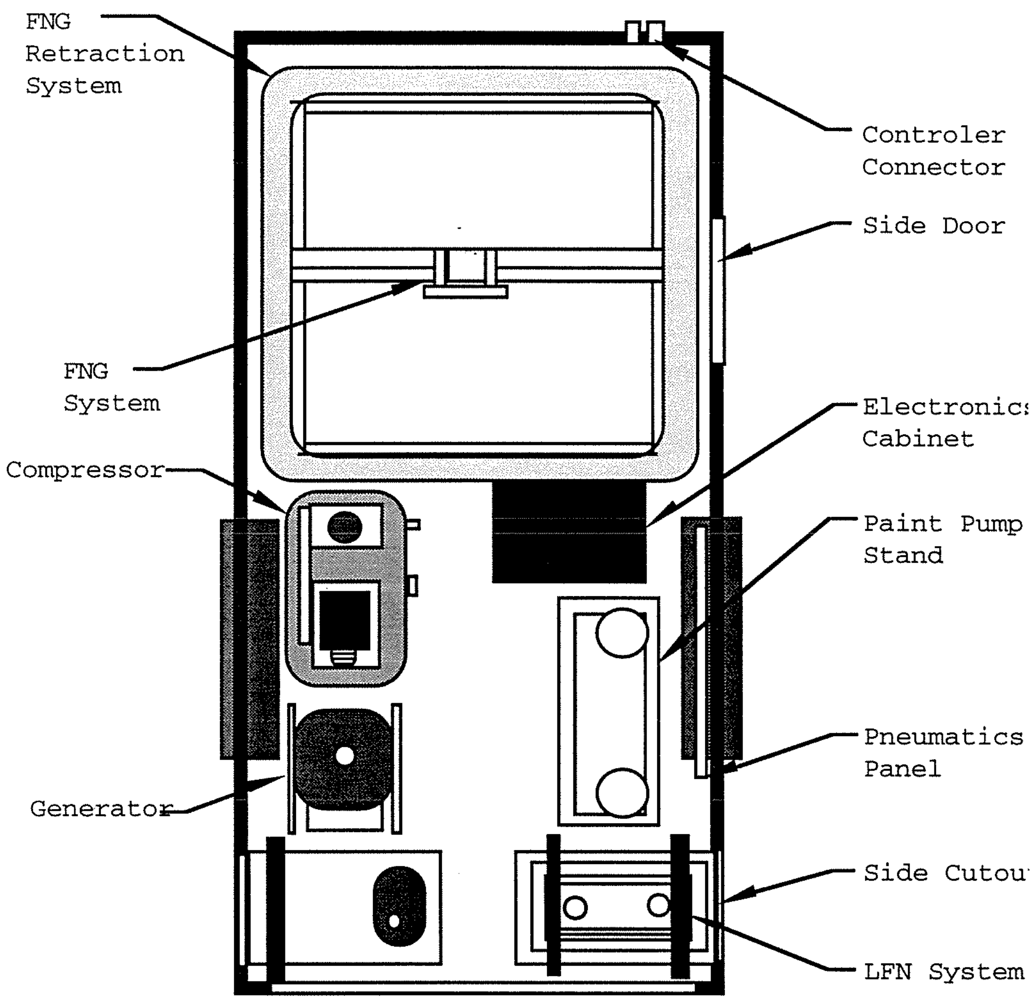


Figure 4: Trailer Layout

3. System Architecture

The premark painting system has a simple modular architecture. The system consists of the LFN, the FNG, the control/electronics system, the pneumatic system, and the paint system. These systems are powered by a gasoline powered generator and compressor. The system connectivity can be seen in figure 5.

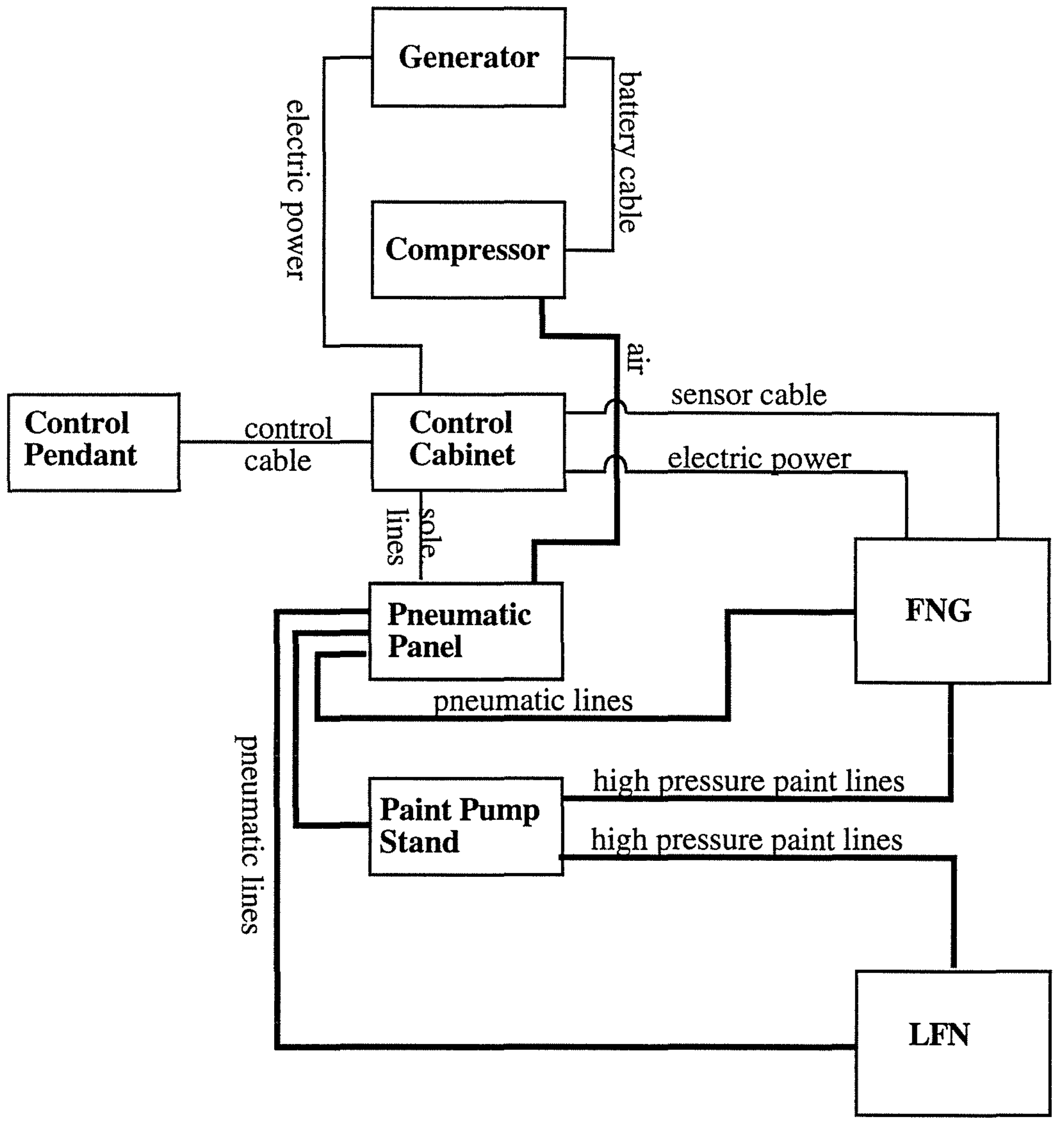


Figure 5: System Connections

3.1 Control System

The control system architecture is a simple hierarchical system. At the top of the hierarchy is a small hand held unit consisting of a touch pad, LCD screen display, and a small computer board. The unit's touch pad controls all the functions of the FNGS and the LFNS including the activation of the paint pumps and mixers. In addition, the hand held unit provides a manual operation mode for the gantry robot. Figure 6 shows the control system architecture.

Communications to the robot controller are accomplished through the RS-232 serial communications standard. Simple digital lines control a signal conditioning board that interfaces directly to the pneumatic solenoids located on the pneumatic panel. Manual overrides for all solenoid functions are also provided.

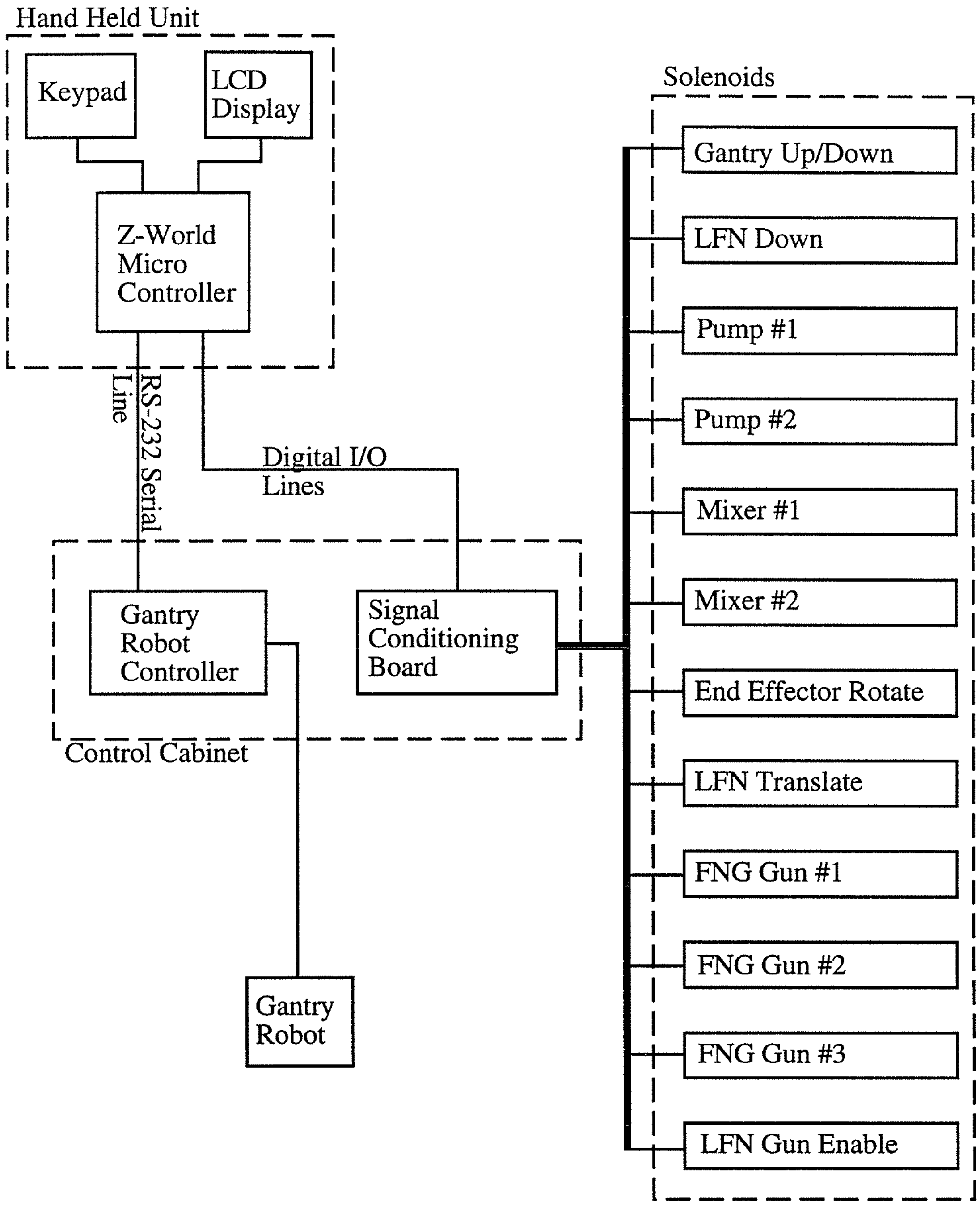


Figure 6: Control System Architecture

3.1.1 The Hand Held Unit

The hand held unit is a standard Z-World Corporation Little Giant mini controller. This unit is programmed directly in the C programming language. The Little Giant has 16 digital I/O channels, 8 high voltage relay drivers, 6 Analog-to-Digital channels, 1 Digital-communication port. In addition to the Little Giant board, the system uses a Z-

World Keyboard Display Module (KDM) to provide the keypad and LCD interface. Both the Little Giant and the KDM are housed in a standard Z-World enclosure. Figure 7 shows the hand held unit.

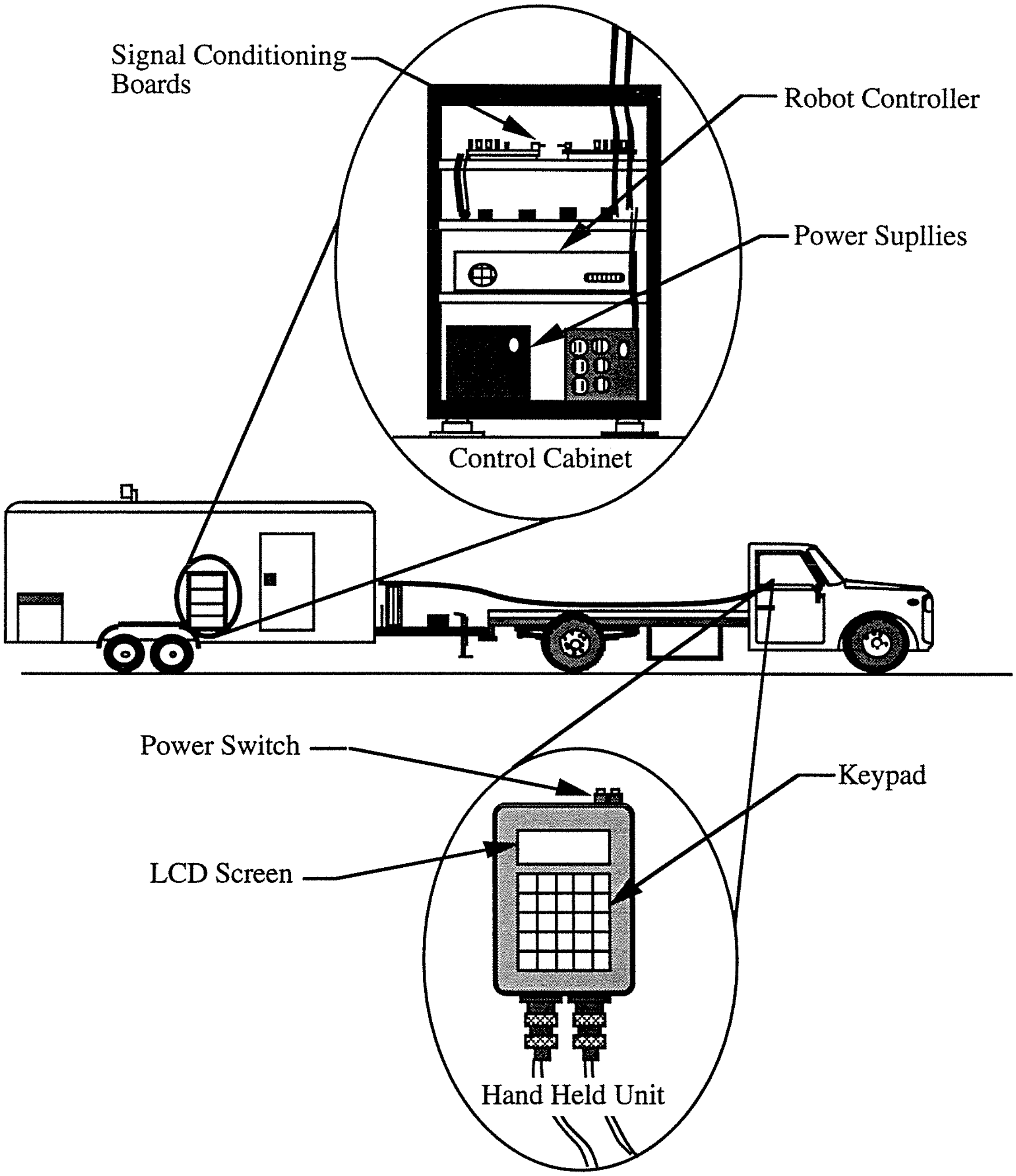


Figure 7: Control System Components

Currently, the prototype system is using approximately 60% of the computing capacity of the system. It is anticipated that production systems will not use much more than present. Thus, future growth in the system can be accommodated easily. Hardware utilization of the system is about 30%. There currently are 8 high voltage relay drivers, 6 Analog-to Digital channels, 1 Digital-to-Analog channel, and 1 channel of digital I/O available for use.

3.1.2 The Robot Controller

The robot controller is a standard Intelligent Actuator Inc. controller. This unit consists of a power transformer and an intelligent controller. The controller accepts high level commands from the RS-232 serial port and commands the robot directly. All book-keeping functions such as position and velocity measurements are also accomplished by the robot controller. In addition, the controller is outfitted with additional logic to allow path planning and interpolation. These functions simplify robot positioning and increases the efficiency of the robot.

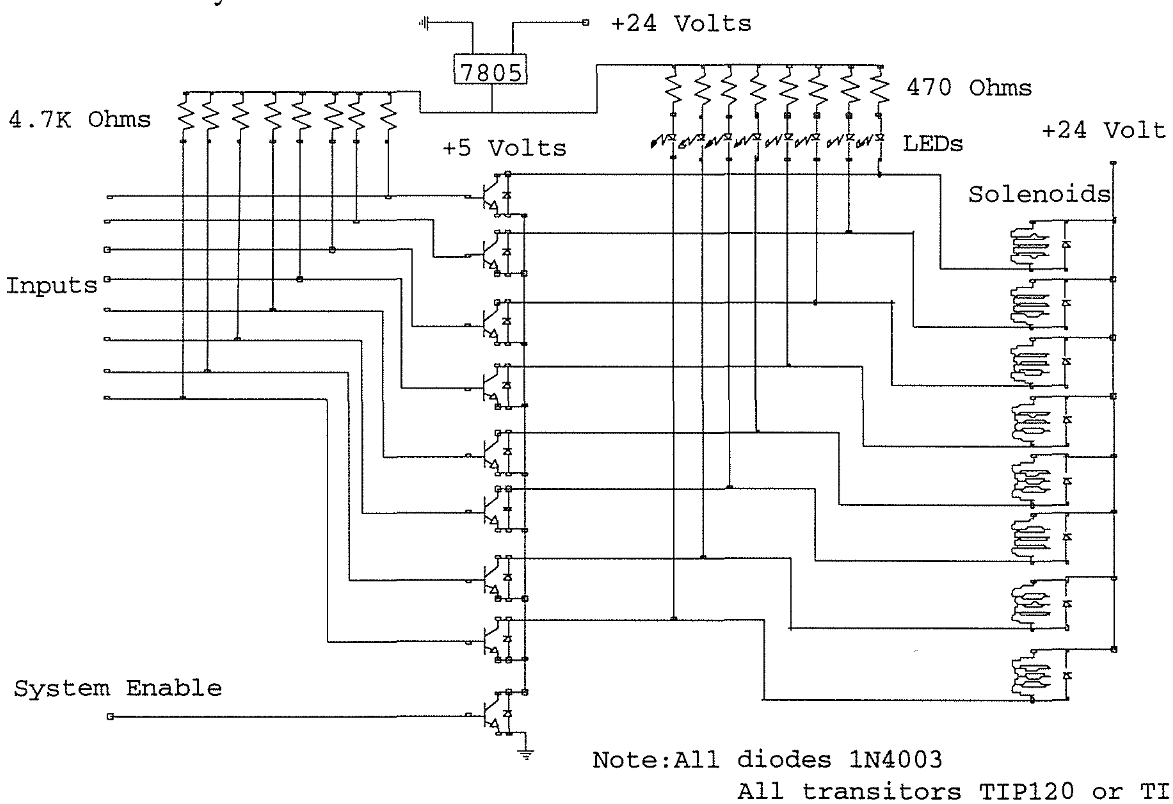


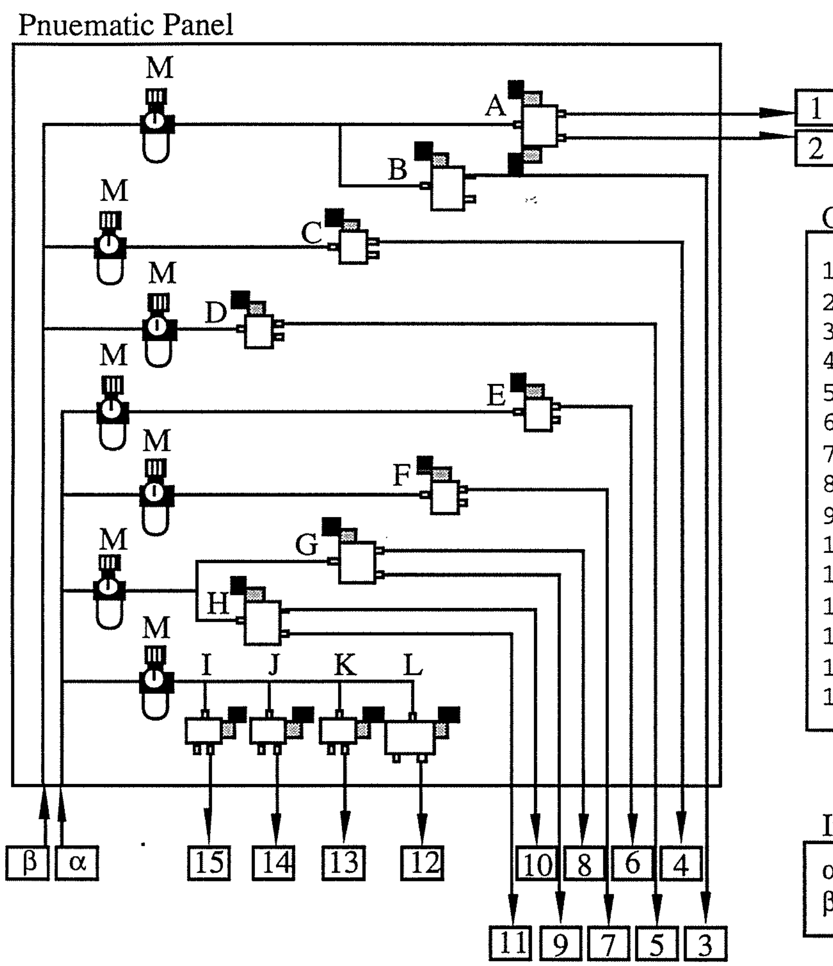
Figure 8: Signal Conditioning Board Schematic

3.1.3 The Signal Conditioning Board

The signal conditioning board is a simple solid state board that accepts TTL level input and outputs +24 Volt, 5 Amp signals to drive the pneumatic solenoids. This board is custom made and two were used in the painting system. Figure 8 shows a schematic of the board.

3.2 Support Systems

The LFNS and the FNGS share the use of the pneumatic system and the paint system. Both systems are custom installations mounted inside the trailer. In addition to these systems, the trailer has an electric generator and an air compressor both of which are required to provide internal power.



Output Lines

1: Gantry Up
2: Gantry Down
3: LFN Down
4: Pump #1
5: Pump #2
6: Mixer #1
7: Mixer #2
8: End Effector Rotat
9: End Effector Rotat
10: LFN Translate #1
11: LFN Translate #2
12: LFN Gun Enable
13: Gantry Gun #1
14: Gantry Gun #2
15: Gantry Gun #3

Input Lines

α: Supply #1
β: Supply #2

Components

A: 5-port, 3-way solenoid
B: 5-port, 2-way solenoid
C: 3-port, 2-way solenoid
D: 3-port, 2-way solenoid
E: 3-port, 2-way solenoid
F: 3-port, 2-way solenoid
G: 5-port, 2-way solenoid
H: 5-port, 2-way solenoid

I: 3-port, 2-way solenoid
J: 3-port, 2-way solenoid
K: 3-port, 2-way solenoid
L: 5-port, 2-way solenoid
M: Filter/Regulator

Note: All panel lines 3/8" plastic.

Figure 9: Pneumatic Schematic

3.2.1 The Pneumatic System

The pneumatic system is a custom designed system that provides control air to the LFN, the FNG, and the paint system. The system consists of pressure regulators, air dryers, and solenoids attached to a panel which in turn is mounted to the wall of the trailer. The control solenoids are Norgren Nugget 200's running on +24 Volts, 0.25 Amps. Figure 9 shows a schematic representation of the pneumatic system.

3.2.2 The Paint System

Both the FNGS and the LFNS use the paint system which consists of 2 Binks Inc pneumatic airless pumps to pressurize the paint , 2 pneumatic mixers, and assorted paint manifolds. The pumps are 30:1 ratio capable of producing a 3000 psi paint pressure at an air inlet pressure of 100 psi. Each pump has a flow rate of 0.75 GPM. The paint used throughout development of this system was Pervo paint 4773A white and 4775A black.

3.2.3 Auxiliary Systems

The paint system requires the use of an electric generator and an air compressor to provide the power necessary to operate. Both of these items are mounted inside the trailer with the exhaust vented through a port through the roof of the trailer. The generator is a 4 kW Dayton portable generator providing power to run the control system and the gantry robot. The power from the generator is filtered through an in-line power conditioner to minimize noise in the electric system. The air compressor is a two-stage, 175 psi Speedaire compressor providing air to run the pneumatic system. The compressor is capable of providing 17 CFM air to the system.

3.3 FNGS Components

The FNGS consists of three primary systems, the robot, the end effector, and the retraction mechanism. Each of the three systems operates independently of the others but all three are controlled through the hand held control unit.

3.3.1 Gantry Robot

The gantry robot is an Intelligent Actuator Inc. robot . The robot's maximum work space is 53"x53". The maximum acceleration is 1 g, with a payload capacity of 60 lbs. The maximum speed is 49.2 in/sec. The servo motor system has a rated repeatability of +/-0.08mm. Figure 10 shows the overall dimensions of the robot.

3.3.2 The End Effector

The end effector is a custom built component that provides the interface between the gantry robot and the paint guns. It supports three Binks model 550 pneumatically activated paint guns on individually adjustable mounts. The adjustment is in both the lateral direction, the spacing between the guns, and the vertical direction. Lateral

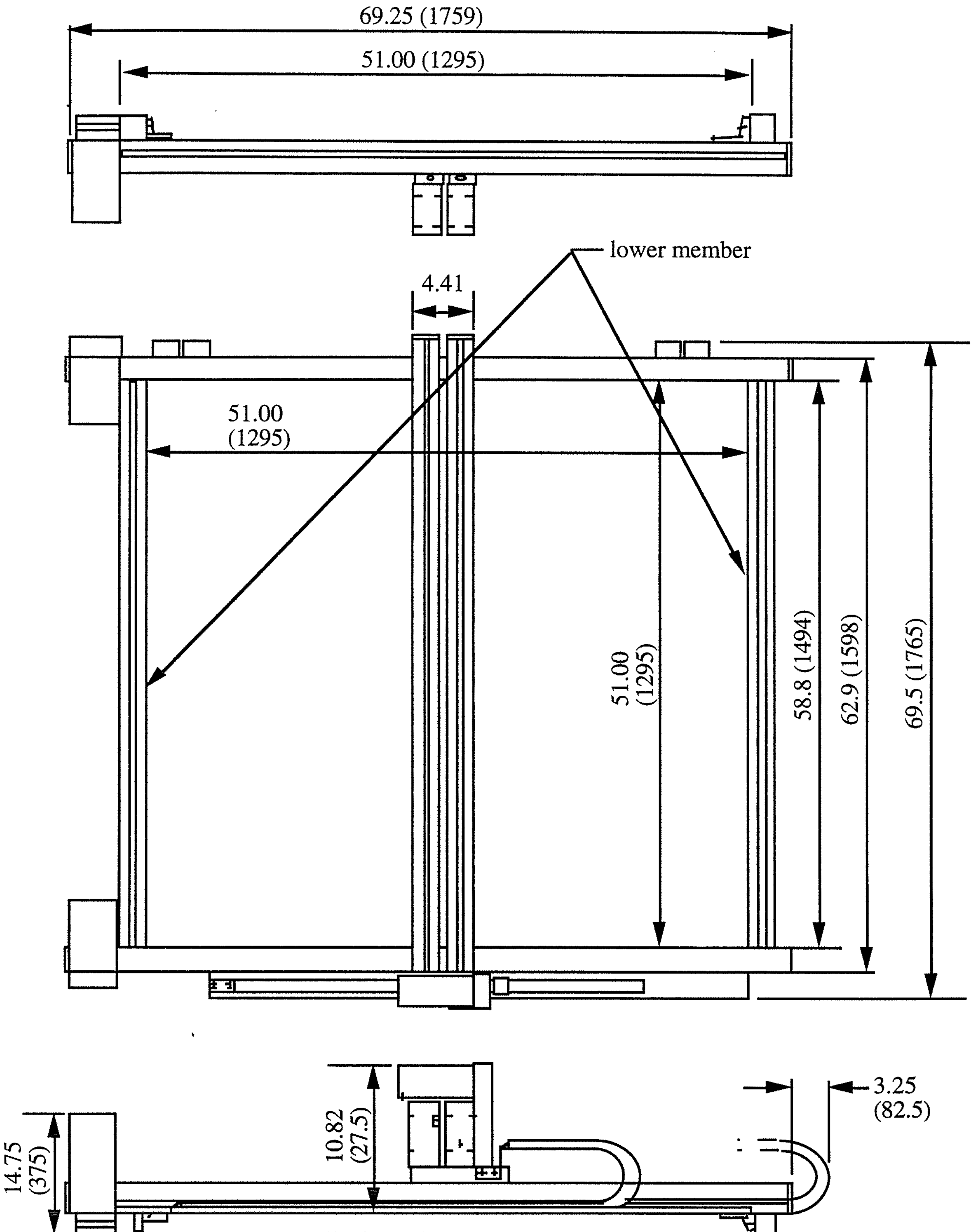


Figure 10: Gantry Robot Dimensions

adjustment is accomplished by sliding the guns along their mounting bars. A total of two inches of adjustment is available for each gun in this direction. The vertical adjustment is accomplished by turning the adjustment nut associated with each mount. A total of four inches of adjustment is available in the vertical direction.

The end effector is capable of rotating the paint guns about a vertical axis between two positions located 90 ° apart. The motion is accomplished through the use of a pneumatic actuator located on the non-rotating section of the end-effector. The actuator is a Phd Inc. model R11A actuator capable of producing 77 in-lbs of torque at 100 psi.

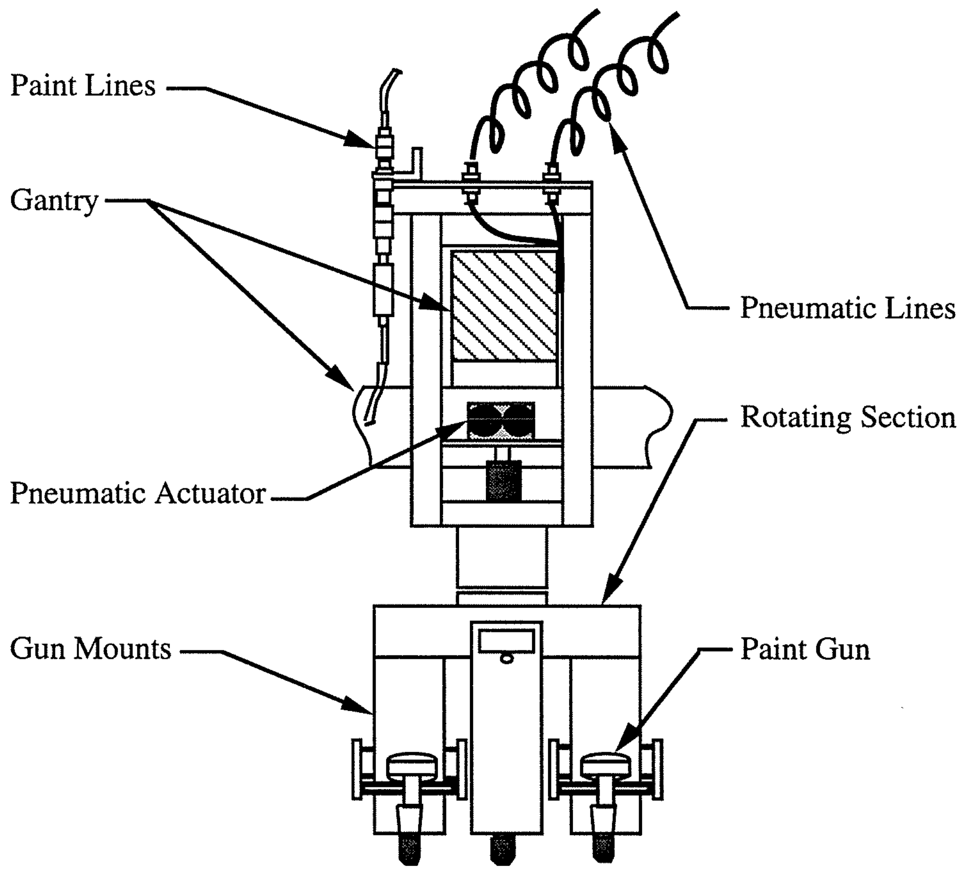


Figure 11: End Effector

3.3.3 The Retraction Mechanism

The retraction mechanism is a custom built system that raises and lowers the gantry robot between a transportation position and a painting position on the ground. The system consists of a geared air motor driving a roller chain assembly which in turn drives four ball screw jacks located at the four corners of the mechanism. The gantry robot is attached to the mechanism through four floating bushings. The bushings allow the gantry to settle onto three legs so that the gantry is level with respect to the roadway in the painting position (see figure 2).

3.4 LFNS Components

The primary components of the LFNS are the retraction system and the translation system. The retraction system is a pneumatically activated system that raises and lowers the LFN between a painting position and a transportation position. The translation system serves the same purpose as the gantry robot in the FNGS; providing paint gun translation in the painting direction.

3.4.1 The Retraction System

The retraction system consists of two pneumatic actuators that raise/lower the LFN carriage between two positions. The system is normally raised in a transportation position. This is accomplished by porting one side of the pneumatic actuators to an air tank maintained at a constant pressure. The other ports of the actuators are attached to the output of a pneumatic valve on the pneumatic panel. The system lowers to a painting position when the pneumatic valve is activated. The descent rate is controlled through the use of flow control valves on each port. This system minimizes the amount of control hardware required for operation and provides a fail safe system in the sense that a power failure causes the system to raise into a stowed position (see figure 12).

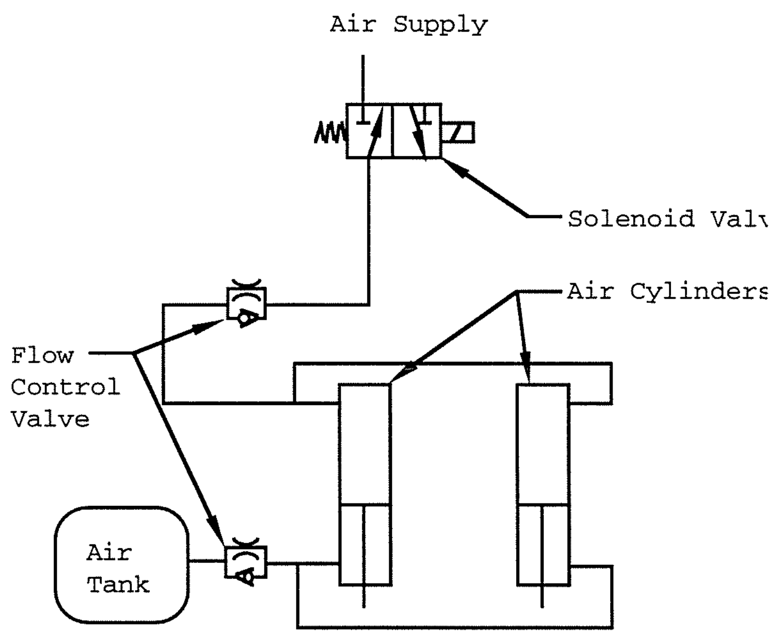


Figure 12: Retraction Control System

3.4.2 The Translation System

The translation system consists of a pneumatic band cylinder supporting three Binks model 550 paint guns mounted in a manner similar to those on the FNG. The band cylinder is a Tol-O-Matic cylinder that is controlled by a solenoid valve located on the pneumatics panel. The rate of travel is controlled by flow control valves on the ports of the band cylinder. The paint guns are mounted to adjustable mounts attached to the band cylinder. As with the FNG, the spacing between the guns and the gun height can be independently adjusted. Gun activation is accomplished through the use of three cam operated air valves. The air to the gun activation system is controlled by a gun enable solenoid located on the pneumatic panel. As the band cylinder travels through its stroke, it pulls the cam valves across a cam on the LFN carriage. The shape of the cam determines the paint pattern deposited by the LFN (see figure 3).

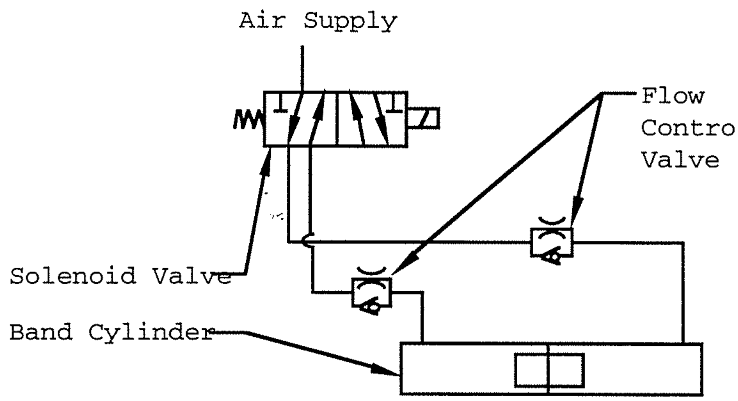


Figure 13: LFN Translation Control

4. System Operation

The painting system is designed to be a completely self-contained system requiring no external power for operation. Both the LFNS and the FNGS can be controlled remotely using the small hand held controller. The system is designed so that the operator is located in the cab of the towing vehicle during all painting operations. This method minimizes the exposure of the survey crew to road hazards.

Normal operating procedures will require that the painting systems and the auxiliary equipment be started and calibrated from within the trailer. This procedure would be accomplished in the maintenance yard prior to field work. The system should be able to operate for long periods without requiring personnel to enter the trailer.

4.1 FNGS Operation

After the system has been calibrated in the maintenance yard or some other safe location, the trailer is towed to a desired site and stopped. The operator will use the auto-paint system to select a large premark from the hand held controller menu. The system will then lower the gantry into the painting position; paint a large premark; then raise the gantry back into a transport position. The cycle time for this operation is approximately two minutes from stopping to ready to travel position.

4.2 LFNS Operation

After the system has been calibrated in the maintenance yard or some other safe location, the trailer is towed to a desired site and stopped. The operator will use the auto-paint system to select an abbreviated premark from the hand held controller menu. The system will then lower the LFN carriage into the painting position; paint an abbreviated premark; then raise the LFN carriage back into a transport position. The cycle time for this operation is approximately 30 seconds from stopping to ready to travel position.

5. Conclusions

The premark painting system is designed to paint photogrammetric ground control targets without exposing the survey crew to a traffic hazard. The system will paint both a standard premark and the experimental abbreviated premark. The premark system is completely self contained and requires no external power to operate. The system operator is located within the cab of the towing vehicle during all painting operations. The operator controls the system through a small hand held control box with a simple keypad interface. This method of painting premarks will significantly improve the safety of roadway surveying operations. In addition, the painting system will help the survey crews to operate more efficiently by allowing them to concentrate on their surveying duties rather than the painting of premarks.

6. Recommendations

The painting system has been completed and its operation demonstrated both in the workshop and on the roadway. However, additional field testing is required to develop detailed operating procedures that will maximize the system's efficiency. This field testing is necessary to determine the best method to integrate the system into the survey operation. This testing will also generate data on the overall reliability of the subsystems during continuous operation. This data will be invaluable during the transition from prototype system to production system.

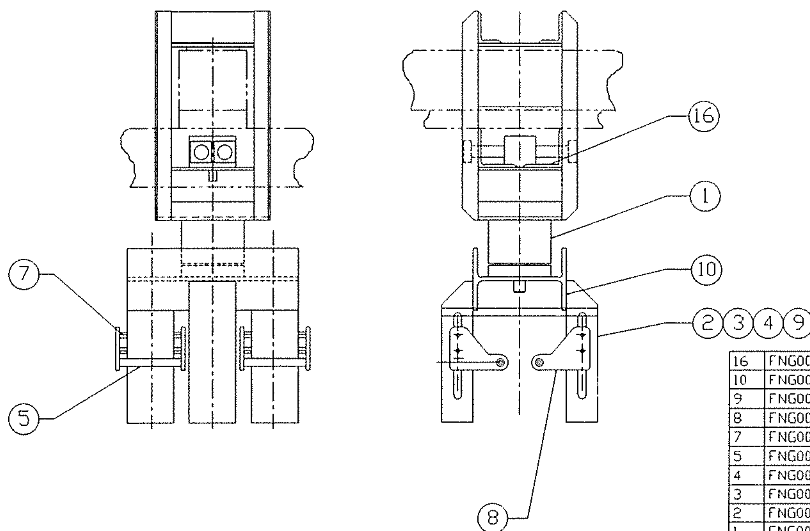
Future enhancements to the design must also be evaluated. Some of the possible improvements include the addition of a GPS receiver to the FNG; a closed circuit television system inside the trailer; and sensor systems for fault detection and premark placement. The GPS receiver would allow real time surveying of the premark. This would further improve the efficiency of the surveying operation. A closed circuit television system would allow the operator to verify the quality of the premark as it is painted. The sensor systems include pressure transducers for the paint and pneumatic system; volt meters for the electric system; and position sensors for the trailer to determine premark spacing.

APPENDIX A:
FNG DRAWINGS

FNG DRAWING LIST

DWG NO.	DESCRIPTION
FNGASSY	END EFFECTOR ASSY
FNG0001	BEARING BLOCK ASSEMBLY
FNG0002	CARRIER TUBE
FNG0003	CARRIER ENDS
FNG0004	CARRIER PISTON
FNG0005	GUN HOLDING ROD
FNG0006	BEARING SUPPORT
FNG0007	NOZZLE MOUNT BOLT
FNG0008	NOZZLE MOUNT PLATE
FNG0009	THREADED ROD
FNG0010	CARRIER MOUNT
FNG0011	ROTARY MOUNT
FNG0012	BEARING SHAFT
FNG0014	BEARING BLOCK
FNG0015	BEARING BLOCK SUPPORT
FNG0016	ACTUATOR MOUNT

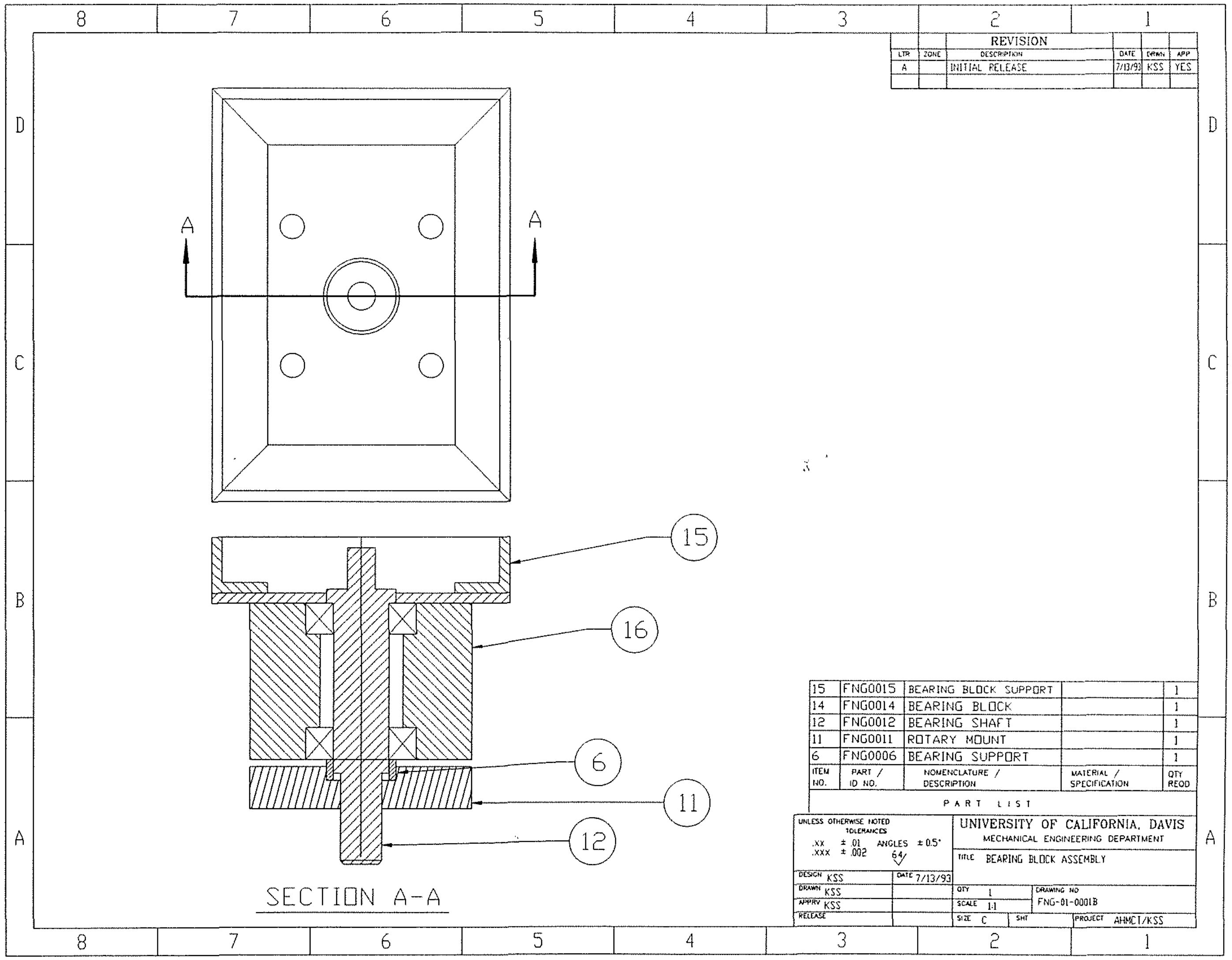
REVISION					
LTR	ZONE	DESCRIPTION	DATE	DRWN	APP
C		INITIAL RELEASE	7/13/93	KSS	YES



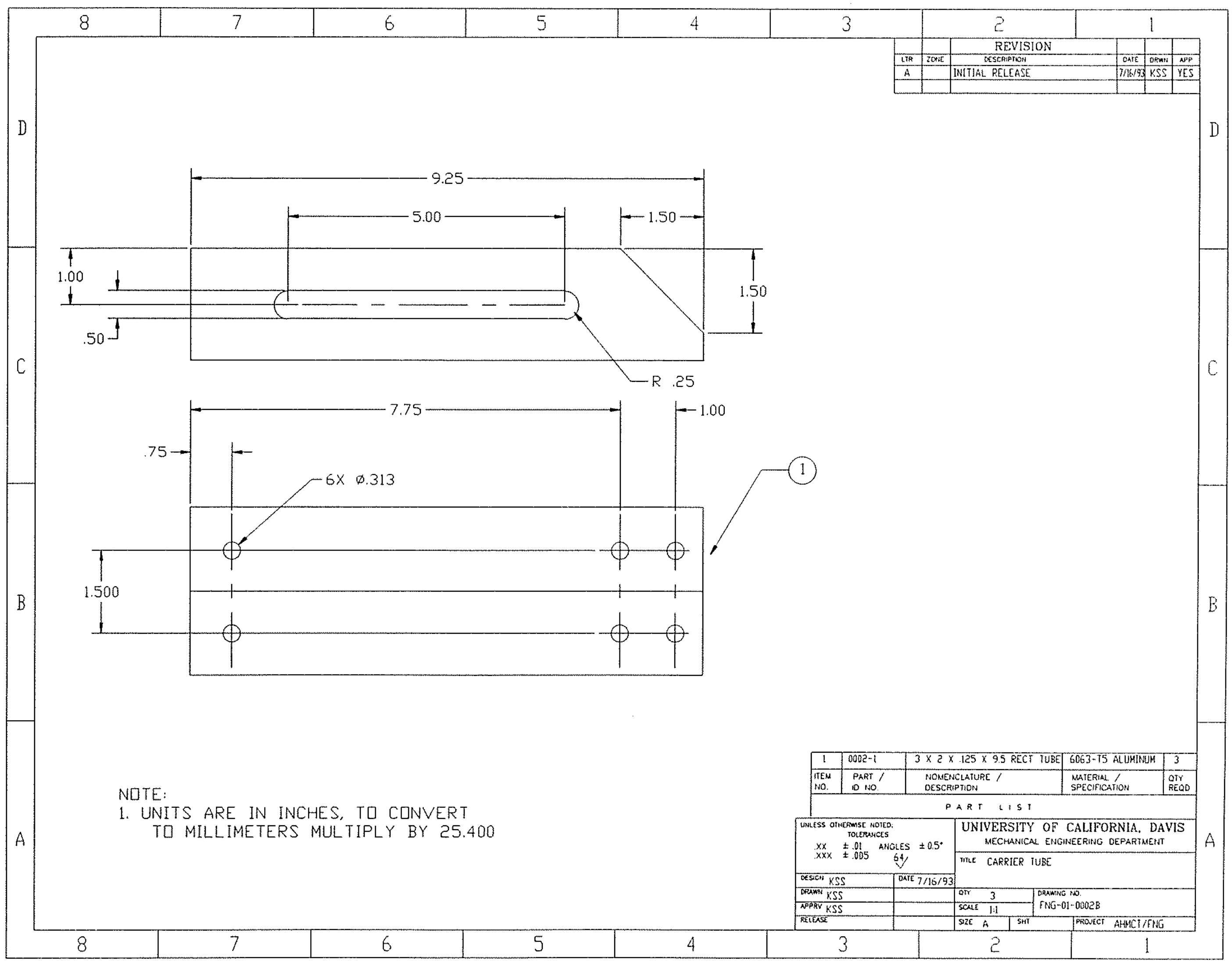
ITEM NO.	PART / ID NO.	NOMENCLATURE / DESCRIPTION	MATERIAL / SPECIFICATION	QTY REQD
16	FNG0016	ACTUATOR MOUNT		1
10	FNG0010	CARRIER MOUNT		1
9	FNG0009	THREADED ROD		3
8	FNG0008	NOZZLE MOUNT PLATE		6
7	FNG0007	NOZZLE MOUNT BOLT		12
5	FNG0005	GUN HOLDING ROD		3
4	FNG0004	CARRIER PISTONS		3
3	FNG0003	CARRIER ENDS		3
2	FNG0002	CARRIER TUBE		3
1	FNG0001	BEARING BLOCK ASSEMBLY		1

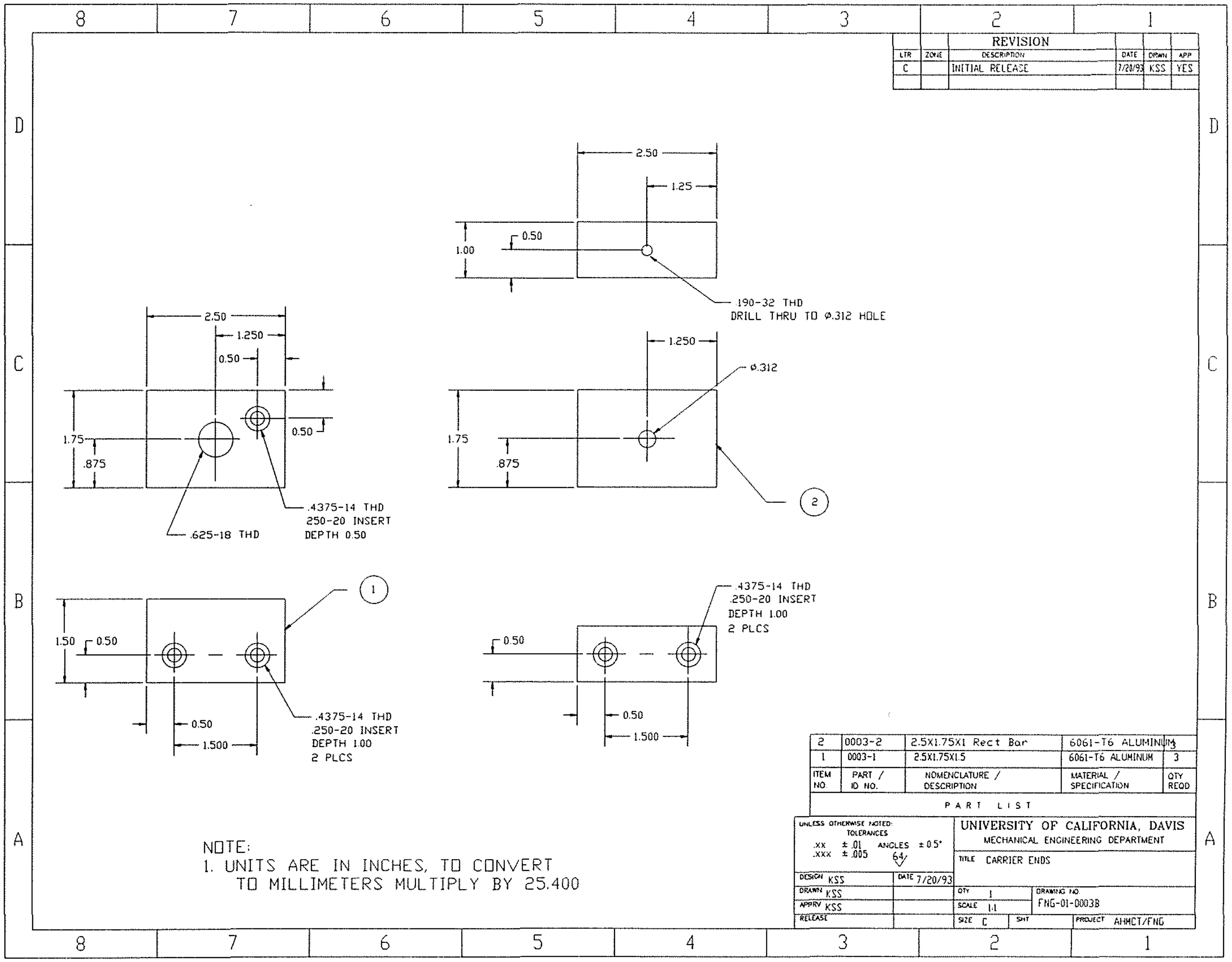
PART LIST

UNLESS OTHERWISE NOTED: TOLERANCES .XX ± .XXX ± ANGLES ± ✓	UNIVERSITY OF CALIFORNIA, DAVIS MECHANICAL ENGINEERING DEPARTMENT
DESIGN KSS / DATE 7/13/93	TITLE END EFFECTOR ASSEMBLY
DRAWN KSS	QTY 1 / DRAWING NO FNG - ASSY
APPRV KSS	SCALE 1:4
RELEASE	SIZE C / SHT / PROJECT AHMCT/FNG

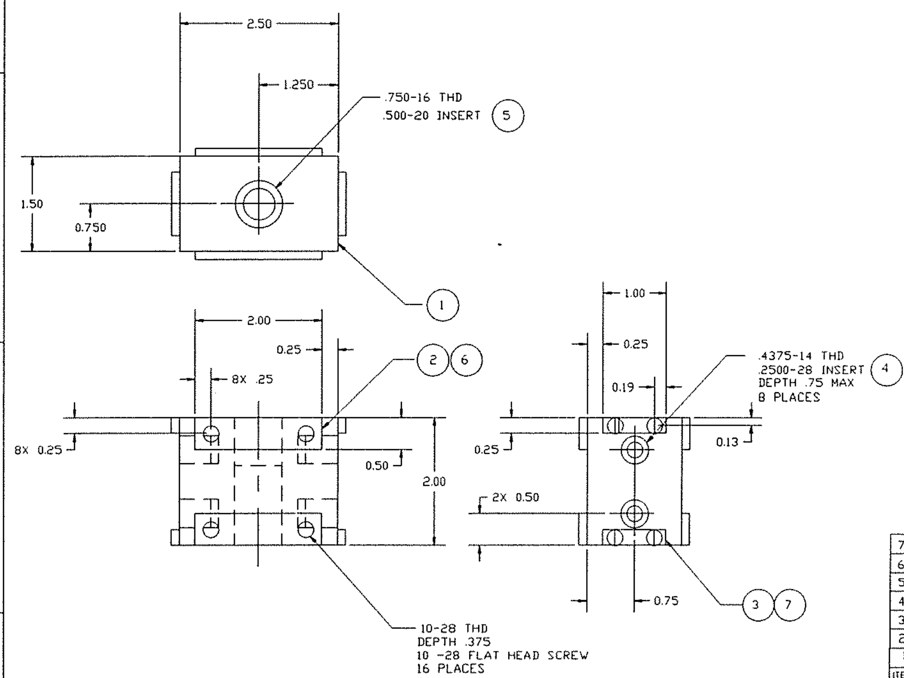


REVISION
LTR ZONE DESCRIPTION DATE DRWN APP
A INITIAL RELEASE 7/13/93 KSS YES
A
A
SECTION A-A
15
16
6
11
12
15 FNG0015 BEARING BLOCK SUPPORT 1
14 FNG0014 BEARING BLOCK 1
12 FNG0012 BEARING SHAFT 1
11 FNG0011 ROTARY MOUNT 1
6 FNG0006 BEARING SUPPORT 1
ITEM NO. PART / ID NO. NOMENCLATURE / DESCRIPTION MATERIAL / SPECIFICATION QTY REQD
PART LIST
UNLESS OTHERWISE NOTED TOLERANCES
.XX ± .01 ANGLES ± 0.5°
.XXX ± .002
64
DESIGN KSS DATE 7/13/93
DRAWN KSS
APPRV KSS
RELEASE
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE BEARING BLOCK ASSEMBLY
QTY 1
SCALE 1:1
DRAWING NO FNG-01-0001B
SIZE C
SHT
PROJECT AHMCT/KSS


REVISION
LTR | ZONE | DESCRIPTION | DATE | DRWN | APP
A | | INITIAL RELEASE | 7/16/93 | KSS | YES
9.25
5.00
1.50
1.00
.50
1.50
R .25
7.75
1.00
.75
6X Ø.313
1
1.500
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT
TO MILLIMETERS MULTIPLY BY 25.400
1 | 0002-1 | 3 X 2 X .125 X 9.5 RECT TUBE | 6063-T5 ALUMINUM | 3
ITEM NO. | PART / ID NO. | NOMENCLATURE / DESCRIPTION | MATERIAL / SPECIFICATION | QTY REQD
PART LIST
UNLESS OTHERWISE NOTED: TOLERANCES
.XX ± .01 ANGLES ± 0.5°
.XXX ± .005
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE CARRIER TUBE
DESIGN KSS
DATE 7/16/93
DRAWN KSS
APPRV KSS
RELEASE
QTY 3
SCALE 1:1
DRAWING NO. FNG-01-0002B
SIZE A
SHT
PROJECT AHMCT/FNG


REVISION
LTR ZONE DESCRIPTION DATE DRWN APP
C INITIAL RELEASE 7/20/93 KSS YES
.190-32 THD
DRILL THRU TO φ.312 HOLE
φ.312
.625-18 THD
.4375-14 THD
.250-20 INSERT
DEPTH 0.50
.4375-14 THD
.250-20 INSERT
DEPTH 1.00
2 PLCS
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400
2 0003-2 2.5X1.75X1 Rect Bar 6061-T6 ALUMINUM 3
1 0003-1 2.5X1.75X1.5 6061-T6 ALUMINUM 3
ITEM NO PART / ID NO. NOMENCLATURE / DESCRIPTION MATERIAL / SPECIFICATION QTY REQD
PART LIST
UNLESS OTHERWISE NOTED: TOLERANCES .XX ± .01 .XXX ± .005 ANGLES ± 0.5°
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE CARRIER ENDS
DESIGN KSS DATE 7/20/93
DRAWN KSS
APPRV KSS
RELEASE
QTY 1
SCALE 1:1
DRAWING NO FNG-01-0003B
SIZE C
PROJECT AHMCT/FNG

REVISION					
LTR	ZONE	DESCRIPTION	DATE	DRWN	APP
A		INITIAL RELEASE	7/21/93	KSS	YES



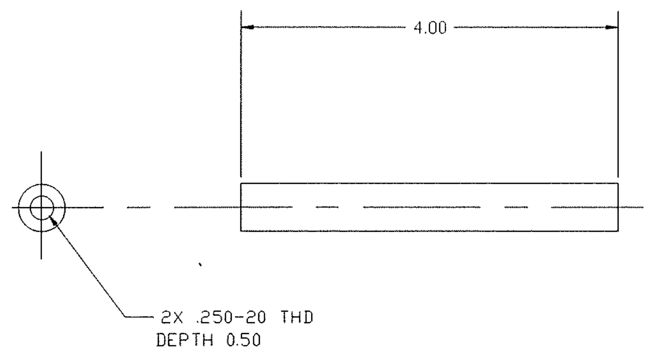
ITEM NO.	PART / ID NO.	NOMENCLATURE / DESCRIPTION	MATERIAL / SPECIFICATION	QTY REQD
7	0004-7	.032X.25X1 SHIM STOCK	AL LAM SHIM	4
6	0004-6	.032X.5X1 SHIM STOCK	AL LAM SHIM	4
5		.500-20 INSERT		1
4		.250-20 INSERT		4
3	0004-3	.125X.25X1 RECT BAR	TEFLON	4
2	0004-2	.125X.5X2 RECT BAR	TEFLON	4
1	0004-1	1.5X2.5X2 RECT BAR	6061-T6 ALUMINUM	1

PART LIST

NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400

UNLESS OTHERWISE NOTED: TOLERANCES .XX ± .01 .XXX ± .005 ANGLES ± 0.5° 64		UNIVERSITY OF CALIFORNIA, DAVIS MECHANICAL ENGINEERING DEPARTMENT	
		TITLE CARRIER PISTON	
DESIGN KSS	DATE 7/21/93		
DRAWN KSS		QTY 3	DRAWING NO FNG-01-0004B
APPRV KSS		SCALE 1:1	
RELEASE		SIZE A / SHT	PROJECT AHMCT/FNG

LTR	ZONE	REVISION DESCRIPTION	DATE	DRWN	APP
A		INITIAL RELEASE	7/13/93	KSS	YES



NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400

ITEM NO.	PART / ID NO.	NOMENCLATURE / DESCRIPTION	MATERIAL / SPECIFICATION	QTY REQD
1		Ø1/2" x 4"	6061-T6 ALUMINUM	3

PART LIST

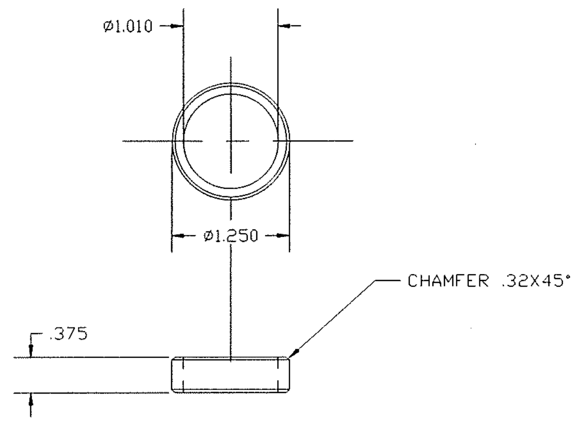
UNLESS OTHERWISE NOTED:
TOLERANCES
.XX ±.01 ANGLES ±0.5°
.XXX ±.002
64√

UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT

TITLE GUN HOLDING ROD

DESIGN KSS	DATE 7/13/93				
DRAWN KSS		QTY 3		DRAWING NO. FNG-01-0005B	
APPRV KSS		SCALE 1:1			
RELEASE		SIZE B	SHT	PROJECT AHMCT/FNG	

REVISION					
LTR	ZONE	DESCRIPTION	DATE	DRWN	APP
A		INITIAL RELEASE	7/13/93	KSS	YES



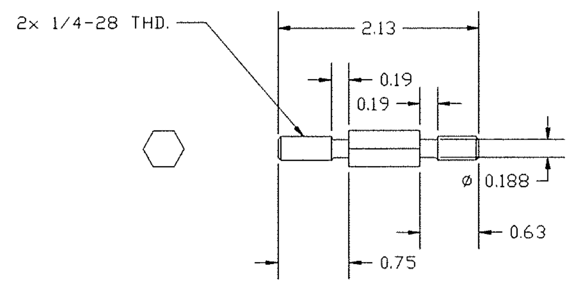
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400

1	?	?		1
ITEM NO.	PART / ID NO.	NOMENCLATURE / DESCRIPTION	MATERIAL / SPECIFICATION	QTY REQD

PART LIST

UNLESS OTHERWISE NOTED: TOLERANCES .XX ±.01 .XXX ±.002 ANGLES ±0.5° 64√		UNIVERSITY OF CALIFORNIA, DAVIS MECHANICAL ENGINEERING DEPARTMENT	
		TITLE BEARING SUPPORT	
DESIGN KSS	DATE 7/13/93		
DRAWN KSS		QTY 1	DRAWING NO. FNG-01-0006B
APPRV KSS		SCALE 1:1	
RELEASE		SIZE B SHT	PROJECT AHMCT/FNG

REVISION					
LTR	ZONE	DESCRIPTION	DATE	DRWN	APP
A		INITIAL RELEASE	7/13/93	KSS	YES



ITEM NO.	PART / ID NO.	NOMENCLATURE / DESCRIPTION	MATERIAL / SPECIFICATION	QTY REQD
1		2.5" x 3/8" HEX STOCK	6061-T6 ALUMINUM	12

PART LIST

NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400

UNLESS OTHERWISE NOTED:
TOLERANCES
.XX ± .01
.XXX ± .002
ANGLES ±0.5°
64 √

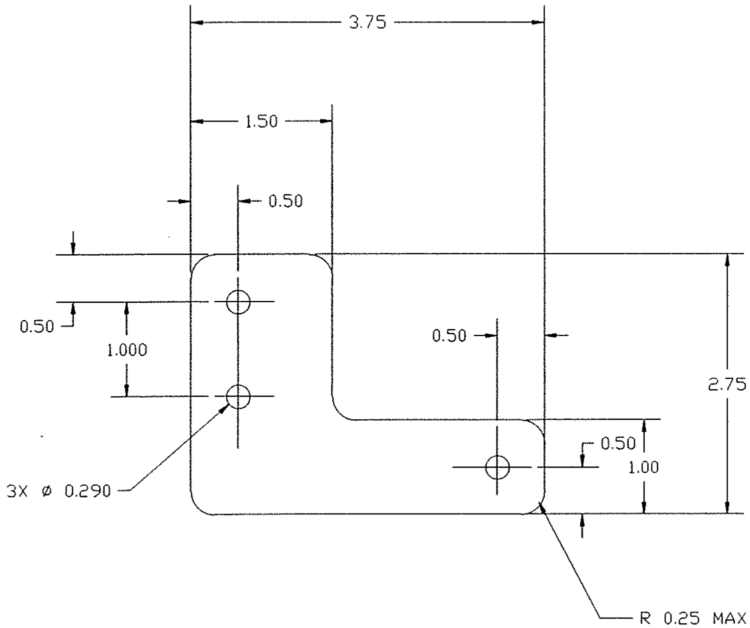
DESIGN KSS	DATE 7/13/93
DRAWN KSS	
APPRV KSS	
RELEASE	

UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT

TITLE NOZZLE MOUNT BOLT

QTY 12	DRAWING NO. FNG-01-0007B	
SCALE 1:1		
SIZE B	SHT	PROJECT AHMCT/FNG

REVISION					
LTR	ZONE	DESCRIPTION	DATE	DRWN	APP
A		INITIAL RELEASE	7/13/93	KSS	YES



NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400

ITEM NO.	PART / ID NO.	NOMENCLATURE / DESCRIPTION	MATERIAL / SPECIFICATION	QTY REQD
1		3" x 4" x 0.25" PLATE	6061-T6 ALUMINUM	6

PART LIST

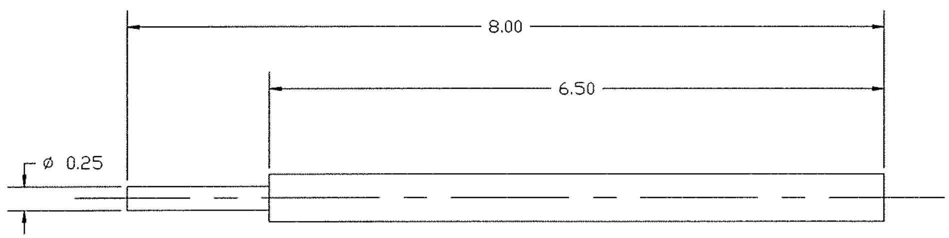
UNLESS OTHERWISE NOTED:
TOLERANCES
.XX ± .01
.XXX ± .002
ANGLES ±0.5°
64 √

UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT

TITLE NOZZLE MOUNT PLATE

DESIGN KSS	DATE 7/13/93		
DRAWN KSS		QTY 6	DRAWING NO. FNG-01-0008B
APPRV KSS		SCALE 1:1	
RELEASE		SIZE B / SHT	PROJECT AHMCT/FNG

REVISION					
LTR	ZONE	DESCRIPTION	DATE	DRWN	APP
A		INITIAL RELEASE	7/13/93	WN/DD	YES

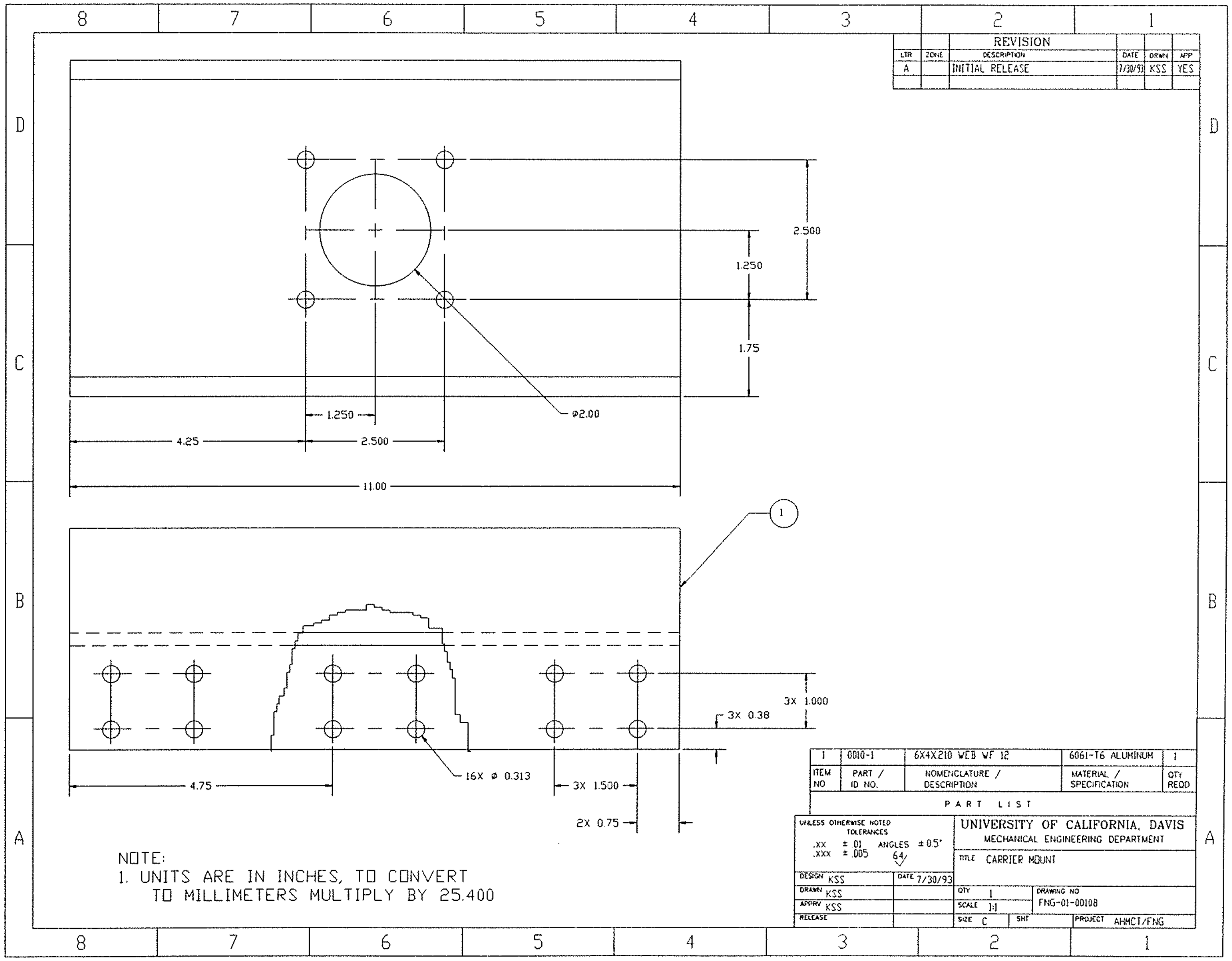


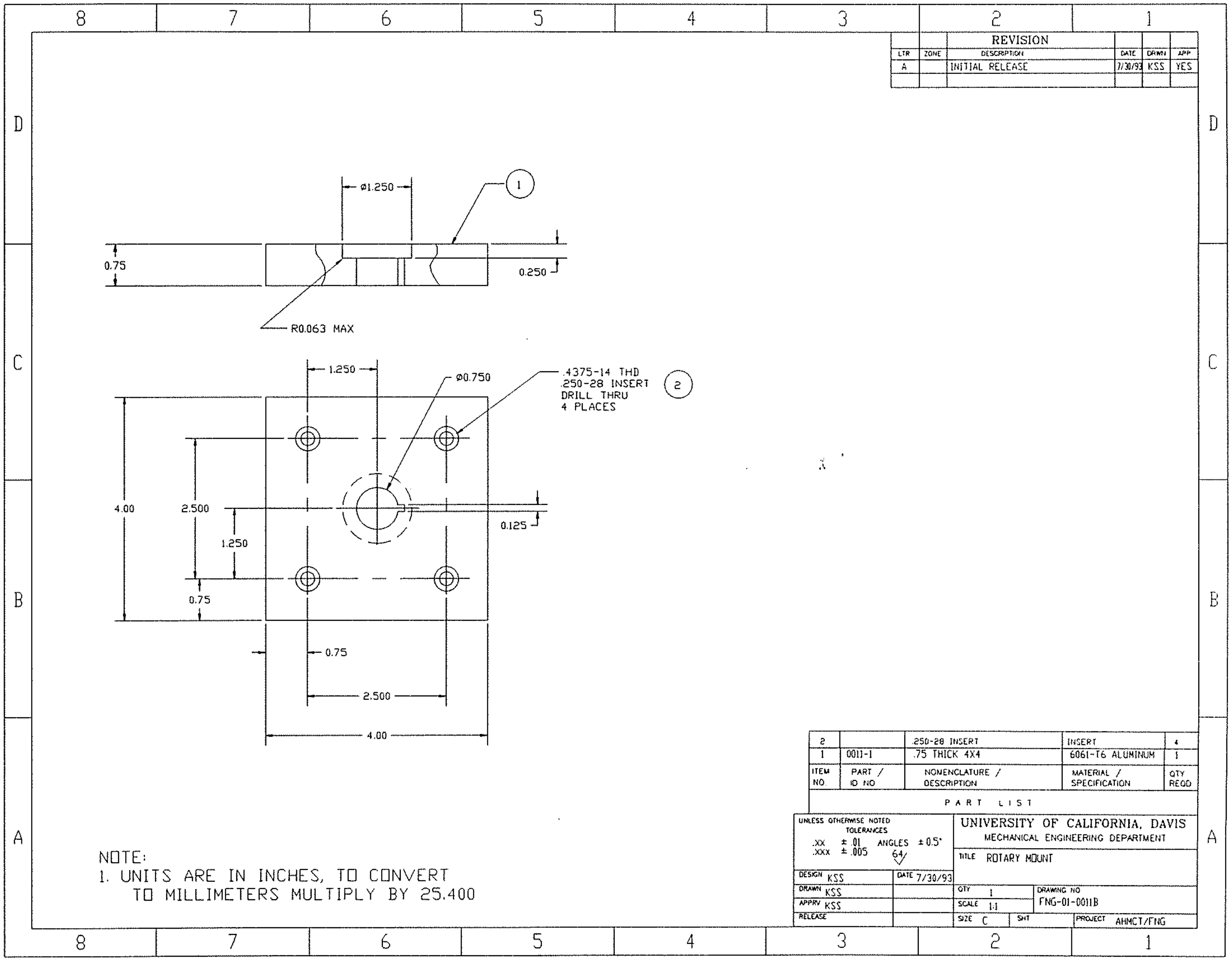
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400

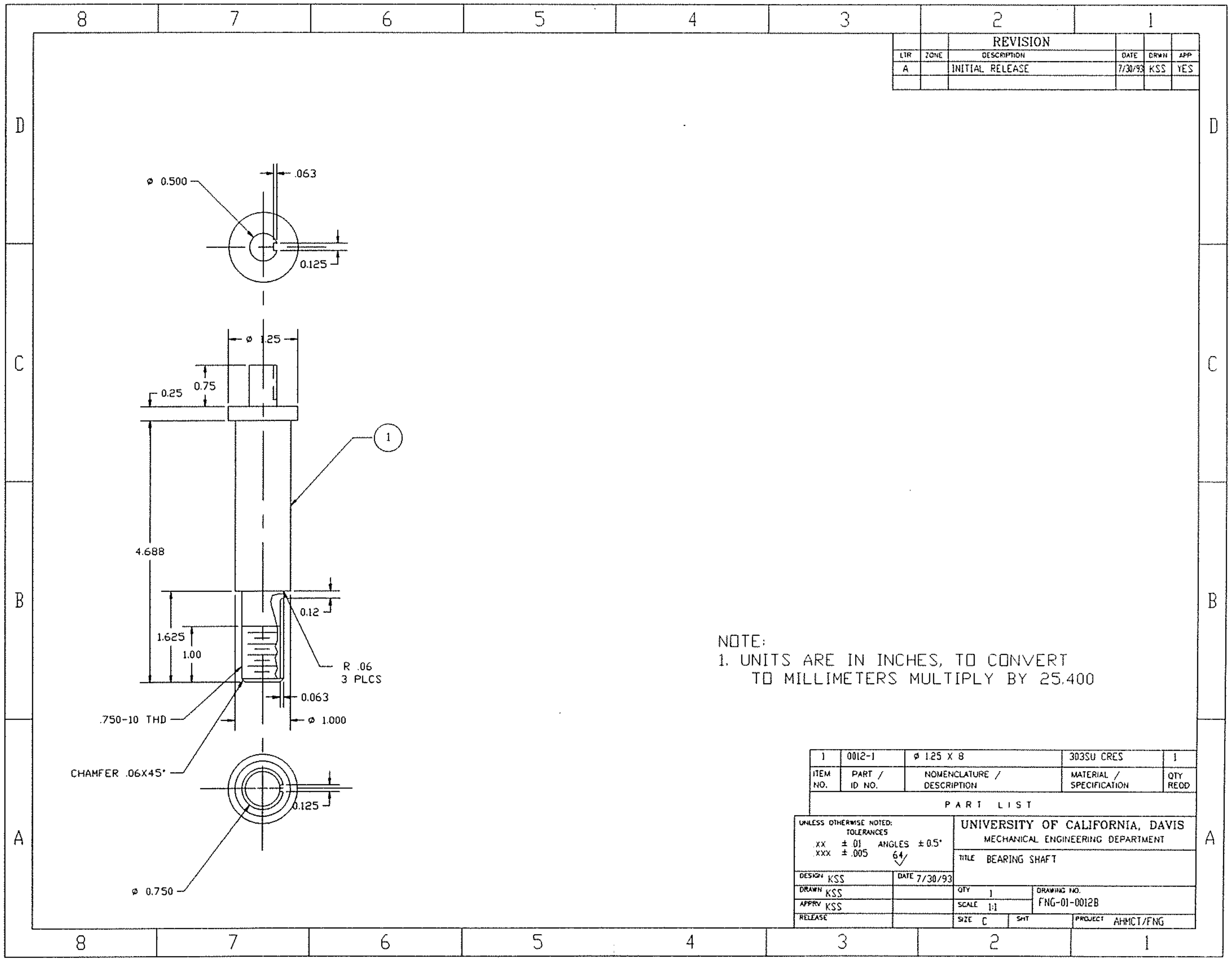
ITEM NO.	PART / ID NO.	NOMENCLATURE / DESCRIPTION	MATERIAL / SPECIFICATION	QTY REQD
1		1/2-20 THD x 8"	BRASS THREADED ROD	3

PART LIST

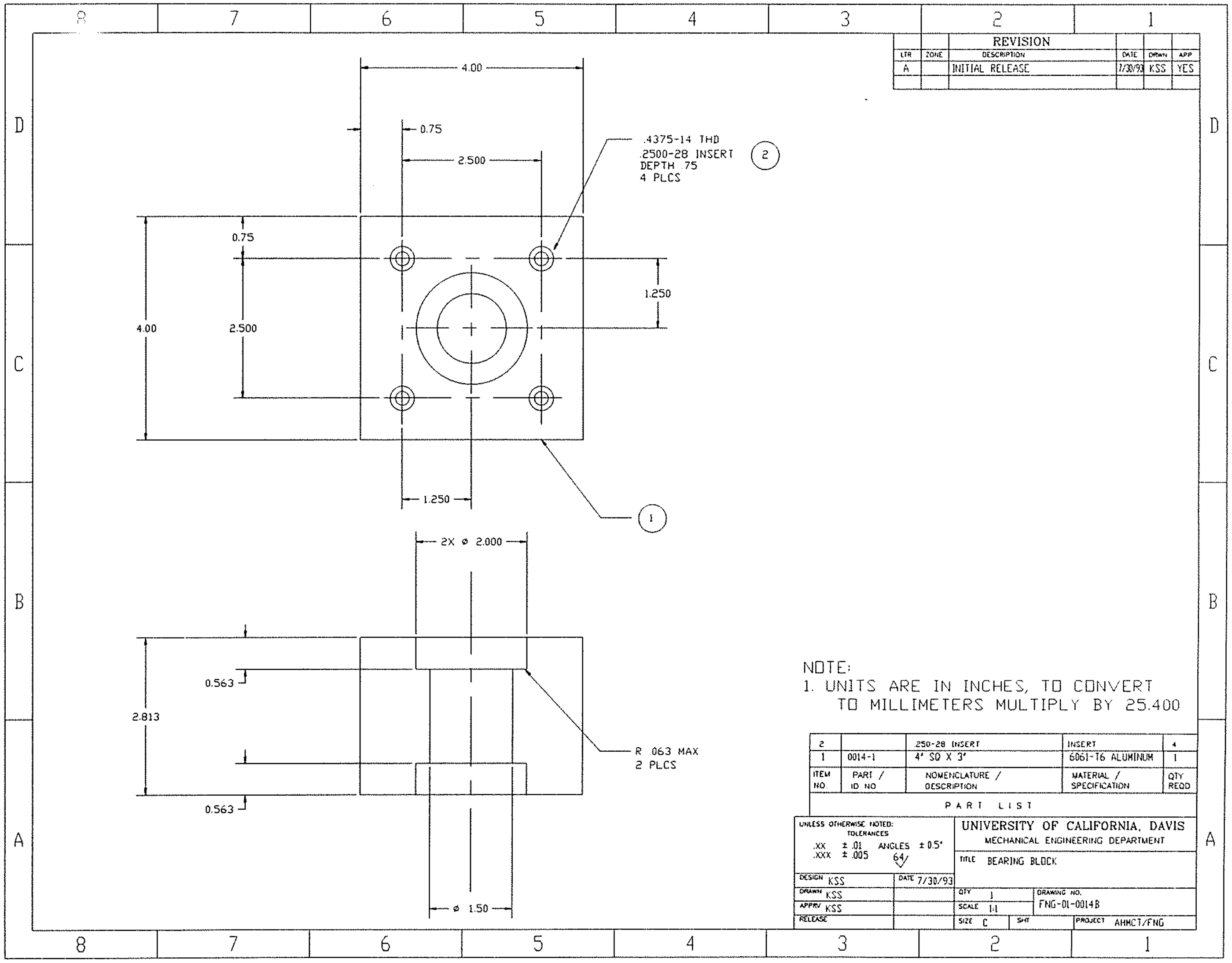
UNLESS OTHERWISE NOTED: TOLERANCES .XX ± .01 .XXX ± .002 ANGLES ±0.5° 64√		UNIVERSITY OF CALIFORNIA, DAVIS MECHANICAL ENGINEERING DEPARTMENT	
		TITLE THREADED ROD	
DESIGN KSS	DATE 7/13/93		
DRAWN KSS		QTY 3	DRAWING NO. FNG-01-0009B
APPRV KSS		SCALE 1:1	
RELEASE		SIZE B SHT	PROJECT AHMCT/FNG


REVISION
LTR ZONE DESCRIPTION DATE DRWN APP
A INITIAL RELEASE 7/30/93 KSS YES
2.500
1.250
1.75
1.250
4.25
2.500
φ2.00
11.00
1
3X 1.000
3X 0.38
16X ⌀ 0.313
4.75
3X 1.500
2X 0.75
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400
1 0010-1 6X4X.210 WEB WF 12 6061-T6 ALUMINUM 1
ITEM NO PART / ID NO. NOMENCLATURE / DESCRIPTION MATERIAL / SPECIFICATION QTY REQD
PART LIST
UNLESS OTHERWISE NOTED TOLERANCES
.XX ± .01 ANGLES ± 0.5°
.XXX ± .005 64
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE CARRIER MOUNT
DESIGN KSS DATE 7/30/93
DRAWN KSS
APPRV KSS
RELEASE
QTY 1
SCALE 1:1
DRAWING NO FNG-01-0010B
SIZE C SHT PROJECT AHMCT/FNG

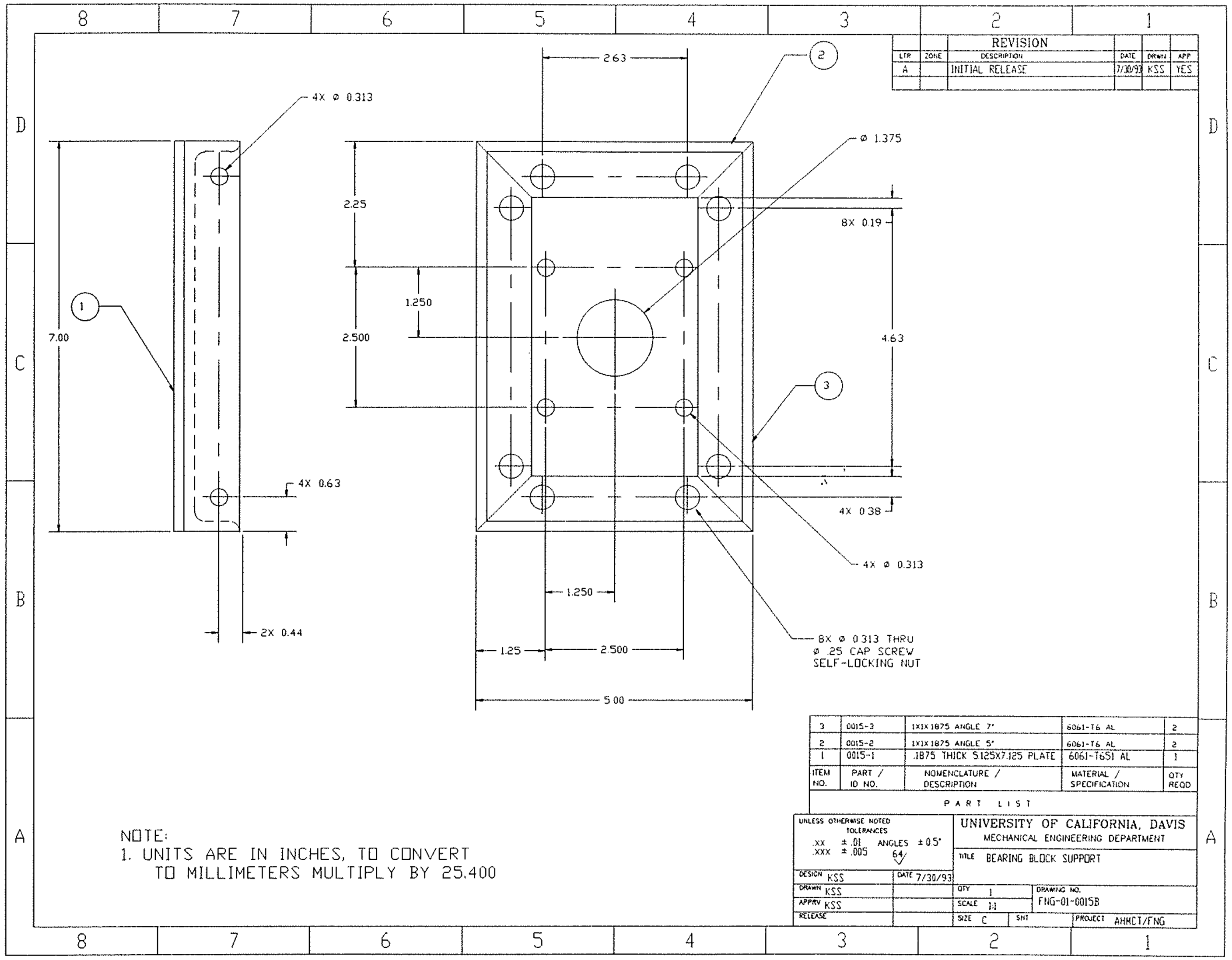

REVISION
LTR | ZONE | DESCRIPTION | DATE | DRAWN | APP
A | | INITIAL RELEASE | 7/30/93 | KSS | YES
ø1.250
0.75
0.250
R0.063 MAX
1.250
ø0.750
.4375-14 THD
.250-28 INSERT
DRILL THRU
4 PLACES
4.00
2.500
1.250
0.125
0.75
0.75
2.500
4.00
2 | | .250-28 INSERT | INSERT | 4
1 | 0011-1 | .75 THICK 4X4 | 6061-T6 ALUMINUM | 1
ITEM NO. | PART / ID NO | NOMENCLATURE / DESCRIPTION | MATERIAL / SPECIFICATION | QTY REQD
PART LIST
UNLESS OTHERWISE NOTED TOLERANCES
.XX ± .01 ANGLES ± 0.5°
.XXX ± .005
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE ROTARY MOUNT
DESIGN KSS
DATE 7/30/93
DRAWN KSS
APPRV KSS
RELEASE
QTY 1
SCALE 1:1
DRAWING NO FNG-01-0011B
SIZE C
SHT
PROJECT AHMCT/FNG
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT
TO MILLIMETERS MULTIPLY BY 25.400

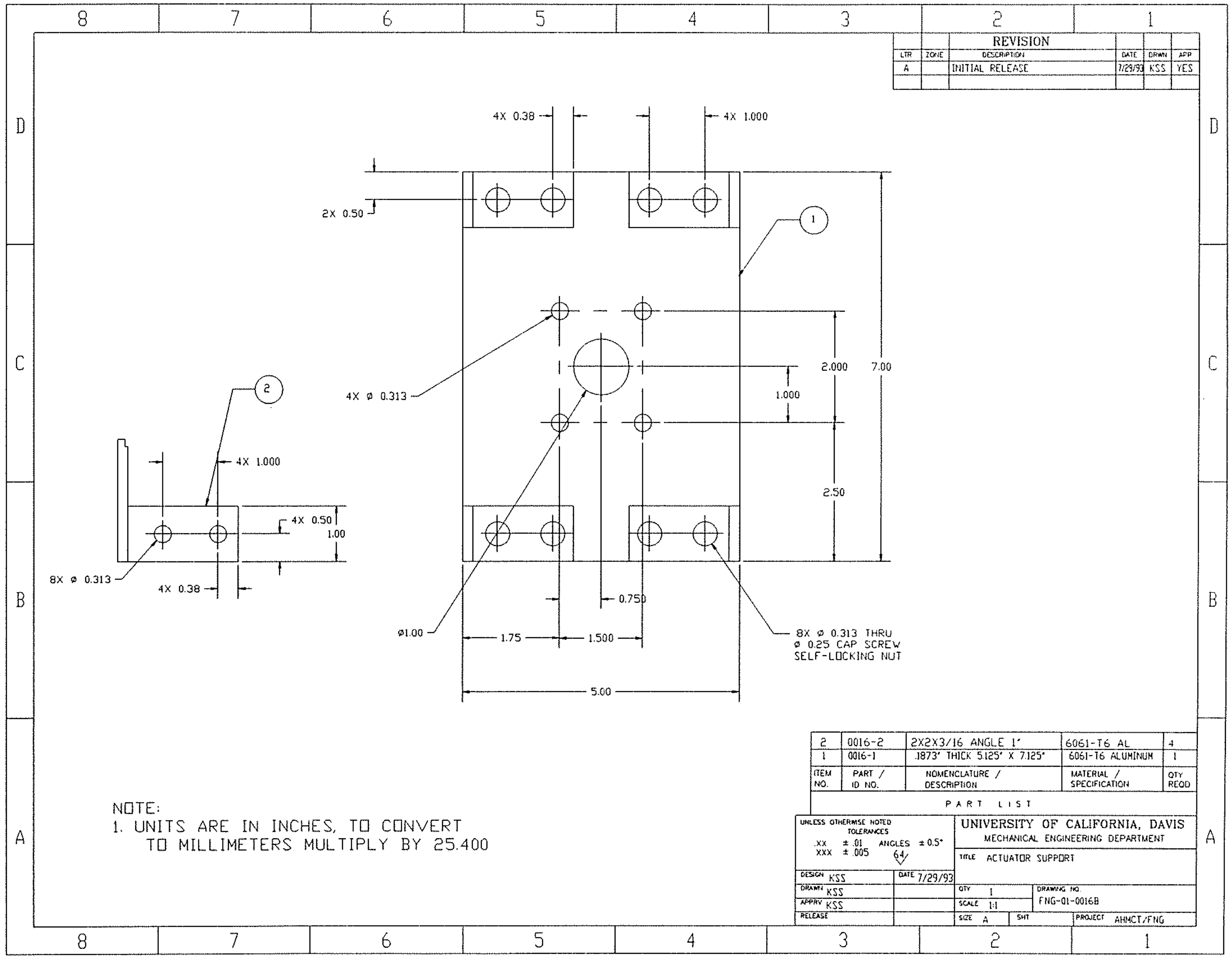


REVISION
LTR | ZONE | DESCRIPTION | DATE | DRWN | APP
A | | INITIAL RELEASE | 7/30/93 | KSS | YES
ø 0.500
.063
0.125
ø 1.25
0.75
0.25
4.688
1.625
1.00
0.12
R .06
3 PLCS
0.063
ø 1.000
.750-10 THD
CHAMFER .06X45°
0.125
ø 0.750
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT
TO MILLIMETERS MULTIPLY BY 25.400
1 | 0012-1 | ø 125 X 8 | 303SU CRES | 1
ITEM NO. | PART / ID NO. | NOMENCLATURE / DESCRIPTION | MATERIAL / SPECIFICATION | QTY REQD
PART LIST
UNLESS OTHERWISE NOTED: TOLERANCES
XX ± .01 ANGLES ± 0.5°
XXX ± .005
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE BEARING SHAFT
DESIGN KSS
DATE 7/30/93
DRAWN KSS
APPRV KSS
RELEASE
QTY 1
SCALE 1:1
SIZE C
SHT
DRAWING NO. FNG-01-0012B
PROJECT AHMCT/FNG



REVISION
LTR ZONE DESCRIPTION DATE DRWN APP
A INITIAL RELEASE 7/30/93 KSS YES
4.00
0.75
2.500
.4375-14 THD
.2500-28 INSERT
DEPTH .75
4 PLCS
2
0.75
1.250
4.00
2.500
1.250
1
2X ⌀ 2.000
0.563
2.813
0.563
R .063 MAX
2 PLCS
⌀ 1.50
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT
TO MILLIMETERS MULTIPLY BY 25.400
2 | .250-28 INSERT | INSERT | 4
1 | 0014-1 | 4" SQ X 3" | 6061-T6 ALUMINUM | 1
ITEM NO. | PART / ID NO | NOMENCLATURE / DESCRIPTION | MATERIAL / SPECIFICATION | QTY REQD
PART LIST
UNLESS OTHERWISE NOTED:
TOLERANCES
.XX ± .01 ANGLES ± 0.5°
.XXX ± .005
64
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE BEARING BLOCK
DESIGN KSS DATE 7/30/93
DRAWN KSS
APPRV KSS
RELEASE
QTY 1
SCALE 1:1
DRAWING NO. FNG-01-0014B
SIZE C SHT
PROJECT AHMCT/FNG


REVISION
LTR | ZONE | DESCRIPTION | DATE | DRWN | APP
A | | INITIAL RELEASE | 7/30/93 | KSS | YES
4X ø 0.313
2.63
ø 1.375
8X 0.19
2.25
1.250
7.00
2.500
4.63
4X 0.63
4X 0.38
4X ø 0.313
1.250
2X 0.44
8X ø 0.313 THRU
ø .25 CAP SCREW
SELF-LOCKING NUT
1.25
2.500
5.00
3 | 0015-3 | 1X1X.1875 ANGLE 7" | 6061-T6 AL | 2
2 | 0015-2 | 1X1X.1875 ANGLE 5" | 6061-T6 AL | 2
1 | 0015-1 | .1875 THICK 5.125X7.125 PLATE | 6061-T651 AL | 1
ITEM NO. | PART / ID NO. | NOMENCLATURE / DESCRIPTION | MATERIAL / SPECIFICATION | QTY REQD
PART LIST
UNLESS OTHERWISE NOTED TOLERANCES
.XX ± .01 ANGLES ± 0.5°
.XXX ± .005
64
DESIGN KSS DATE 7/30/93
DRAWN KSS
APPRV KSS
RELEASE
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE BEARING BLOCK SUPPORT
QTY 1
SCALE 1:1
DRAWING NO. FNG-01-0015B
SIZE C
SHT
PROJECT AHMCT/FNG
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT
TO MILLIMETERS MULTIPLY BY 25.400

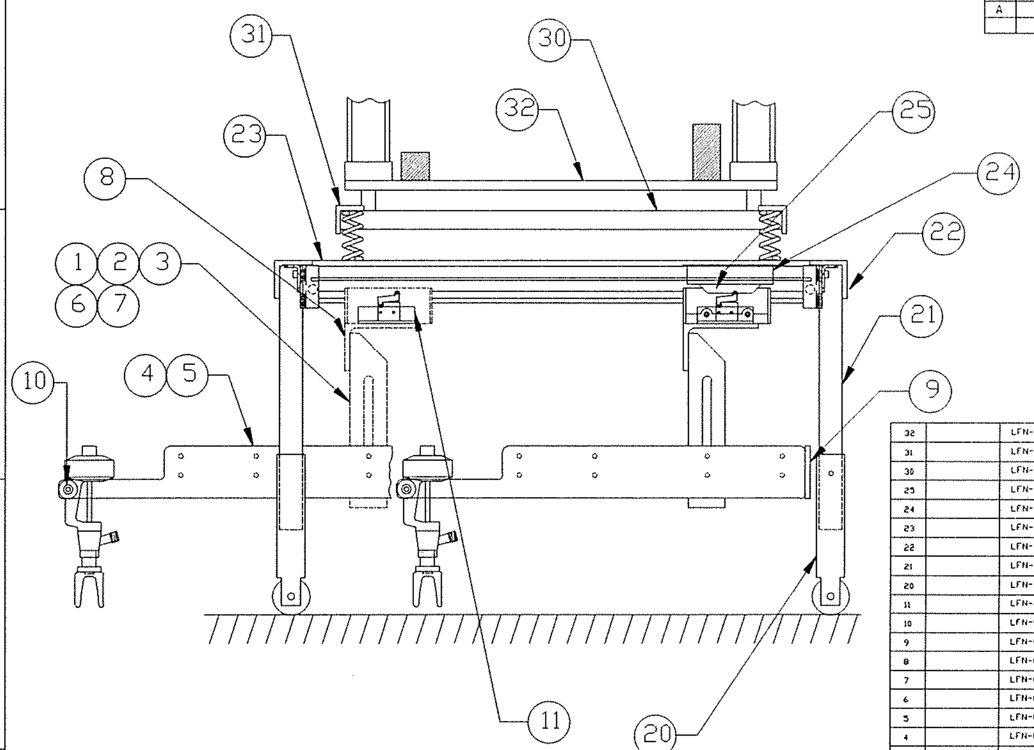

REVISION
LTR | ZONE | DESCRIPTION | DATE | DRWN | APP
A | INITIAL RELEASE | 7/29/93 | KSS | YES
4X 0.38
4X 1.000
2X 0.50
4X ⌀ 0.313
2.000
7.00
1.000
2.50
4X 0.50
1.00
8X ⌀ 0.313
4X 0.38
0.750
⌀1.00
1.75
1.500
5.00
8X ⌀ 0.313 THRU
⌀ 0.25 CAP SCREW
SELF-LOCKING NUT
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400
2 | 0016-2 | 2X2X3/16 ANGLE 1" | 6061-T6 AL | 4
1 | 0016-1 | .1873" THICK 5.125" X 7.125" | 6061-T6 ALUMINUM | 1
ITEM NO. | PART / ID NO. | NOMENCLATURE / DESCRIPTION | MATERIAL / SPECIFICATION | QTY REQD
PART LIST
UNLESS OTHERWISE NOTED TOLERANCES .XX ± .01 XXX ± .005 ANGLES ± 0.5°
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE ACTUATOR SUPPORT
DESIGN KSS
DATE 7/29/93
DRAWN KSS
APPRV KSS
RELEASE
QTY 1
SCALE 1:1
DRAWING NO. FNG-01-0016B
SIZE A
SHT
PROJECT AHMCT/FNG

APPENDIX B:
LFN DRAWINGS

LFN DRAWING LIST

DWG. NO.	DESCRIPTION
LFN0001	TOP CARRIER END
LFN0002	BOTTOM CARRIER END
LFN0003	CARRIER PISTON
LFN0004	PLATE - CARRIER TO GUN
LFN0005	SPACER FOR BOLT
LFN0006	THREADED ROD MOD.
LFN0007	CARRIER TUBE
LFN0008	CARRIER TUBE MOUNT
LFN0009	PLATE END
LFN0010	BAR TO HOLD GUN/SPACER
LFN0011	VALVE HOLDER
LFN0020	FRAME - LEG BOTTOM
LFN0021	FRAME - LEG TOP
LFN0022	FRAME - SHORT ANGLE
LFN0023	FRAME - LONG ANGLE
LFN0024	CAM HOLDER
LFN0025	PAINT CONTROL CAM
LFN0030	RETRACT. SUPPORT BEAM
LFN0031	SPRING SUPPORT
LFN0032	CYLINDER SUPPORT

REVISION						
LTR	ZONE	DESCRIPTION	DATE	DRAWN	APP
A		INITIAL RELEASE	3/9/94	WWN	YES



ITEM NO.	PART / ID NO.	NOMENCLATURE / DESCRIPTION	MATERIAL / SPECIFICATION	QTY REQD
32		LFN-0032	N/A	1
31		LFN-0031	N/A	2
30		LFN-0030	N/A	1
25		LFN-0025	N/A	3
24		LFN-0024	N/A	3
23		LFN-0023	N/A	2
22		LFN-0022	N/A	2
21		LFN-0021	N/A	4
20		LFN-0020	N/A	4
11		LFN-0011	N/A	3
10		LFN-0010	N/A	9
9		LFN-0009	N/A	3
8		LFN-0008	N/A	1
7		LFN-0007	N/A	3
6		LFN-0006	N/A	3
5		LFN-0005	N/A	12
4		LFN-0004	N/A	6
3		LFN-0003	N/A	3
2		LFN-0002	N/A	3
1		LFN-0001	N/A	3

PART LIST

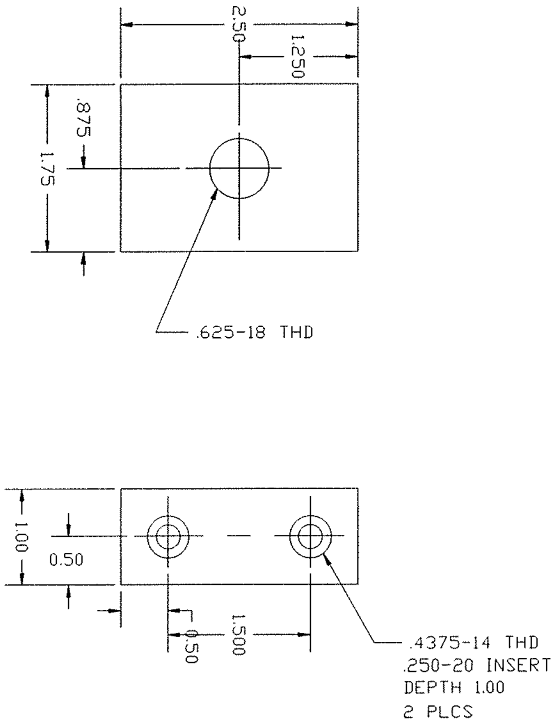
UNLESS OTHERWISE NOTED:
TOLERANCES
.XX ± N/A ANGLES ± N/A
.XXX ± N/A N/A

DESIGN WWN DATE 3/9/94
DRAWN WWN
APPRV WWN
RELEASE

UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT

TITLE ASSEMBLY DWG FOR LFN

QTY 1 DRAWING NO LFN ASSY
SCALE N/A
SIZE C SHT PROJECT AHMCT/LFN



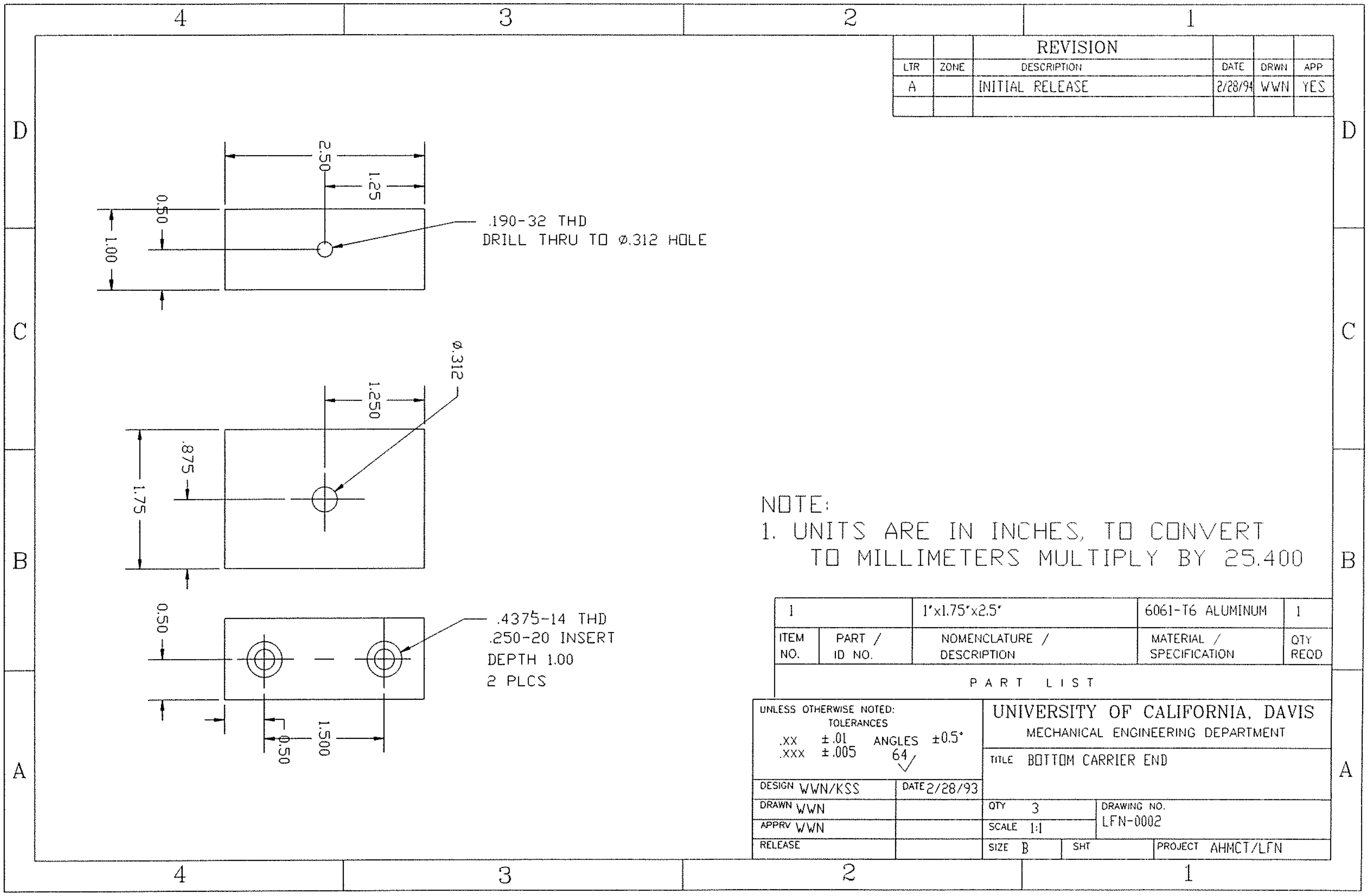
REVISION					
LTR	ZONE	DESCRIPTION	DATE	DRWN	APP
A		INITIAL RELEASE	2/28/94	WWN	YES

NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400

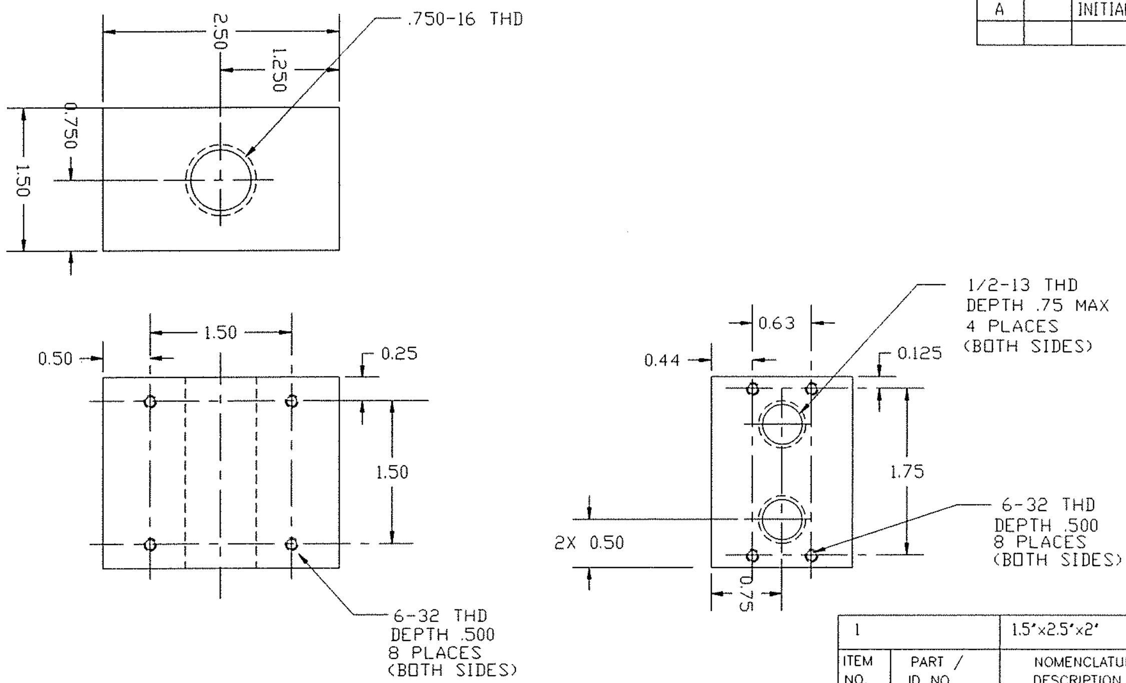
ITEM NO.	PART / ID NO.	NOMENCLATURE / DESCRIPTION	MATERIAL / SPECIFICATION	QTY REQD
1		1"x1.75"x2.5"	6061-T6 ALUMINUM	1

PART LIST

UNLESS OTHERWISE NOTED: TOLERANCES .XX ±.01 .XXX ±.005 ANGLES ±0.5° 64√		UNIVERSITY OF CALIFORNIA, DAVIS MECHANICAL ENGINEERING DEPARTMENT	
		TITLE TOP CARRIER END	
DESIGN WWN/KSS	DATE 2/28/93		
DRAWN WWN		QTY 3	DRAWING NO. LFN-0001
APPRV WWN		SCALE 1:1	
RELEASE		SIZE B SHT	PROJECT AHMCT/LFN


REVISION
LTR ZONE DESCRIPTION DATE DRWN APP
A INITIAL RELEASE 2/28/94 WWN YES
2.50
1.25
0.50
1.00
.190-32 THD
DRILL THRU TO ø.312 HOLE
ø.312
1.250
.875
1.75
0.50
.4375-14 THD
.250-20 INSERT
DEPTH 1.00
2 PLCS
0.50
1.500
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT
TO MILLIMETERS MULTIPLY BY 25.400
1 1"x1.75"x2.5" 6061-T6 ALUMINUM 1
ITEM NO. PART / ID NO. NOMENCLATURE / DESCRIPTION MATERIAL / SPECIFICATION QTY REQD
PART LIST
UNLESS OTHERWISE NOTED:
TOLERANCES
.XX ±.01
.XXX ±.005
ANGLES ±0.5°
64
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE BOTTOM CARRIER END
DESIGN WWN/KSS
DATE 2/28/93
DRAWN WWN
APPRV WWN
RELEASE
QTY 3
SCALE 1:1
SIZE B
SHT
DRAWING NO. LFN-0002
PROJECT AHMCT/LFN

REVISION					
LTR	ZONE	DESCRIPTION	DATE	DRWN	APP
A		INITIAL RELEASE	2/28/94	WWN	YES

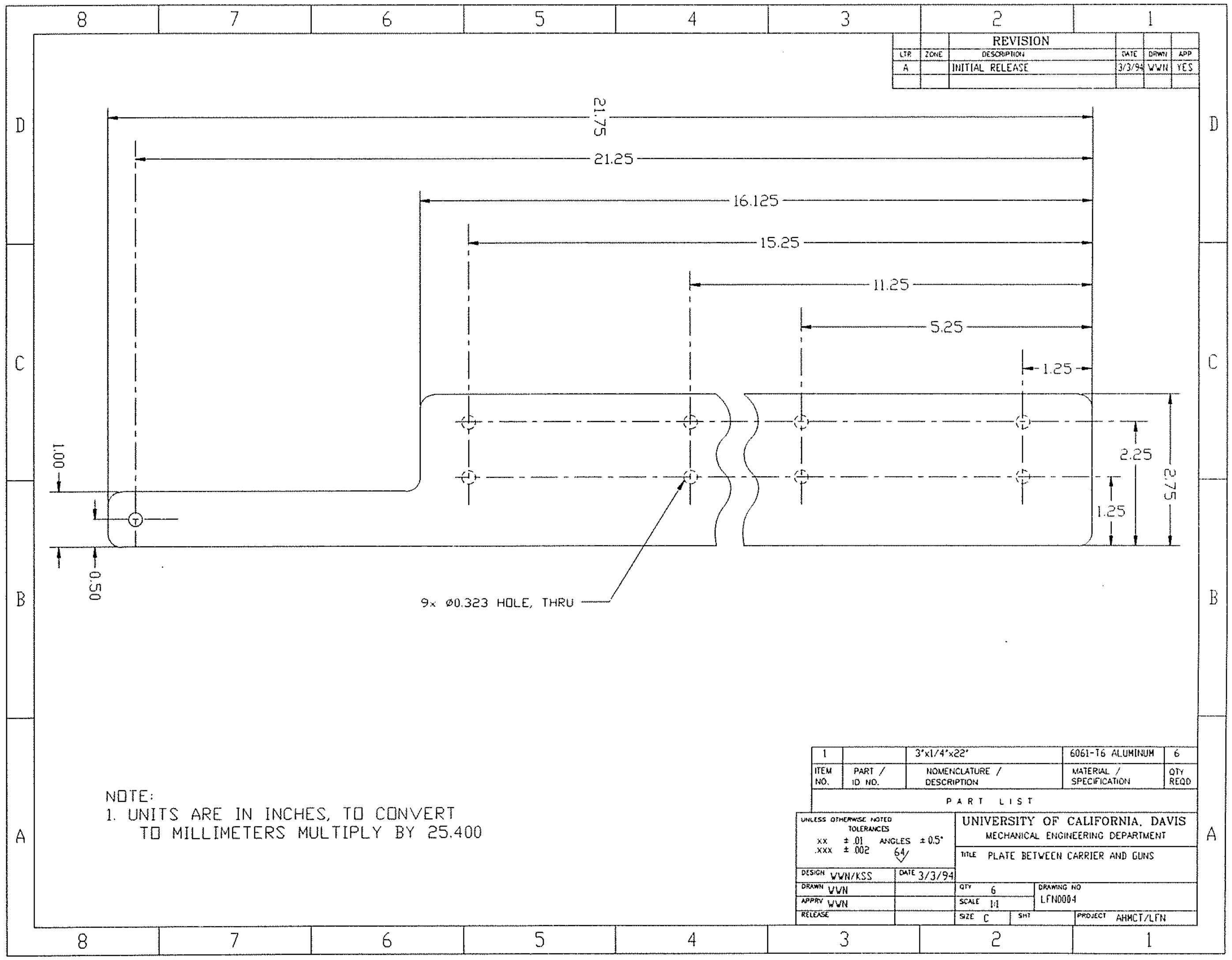


ITEM NO.	PART / ID NO.	NOMENCLATURE / DESCRIPTION	MATERIAL / SPECIFICATION	QTY REQD
1		1.5"x2.5"x2"	6061-T6 ALUMINUM	3

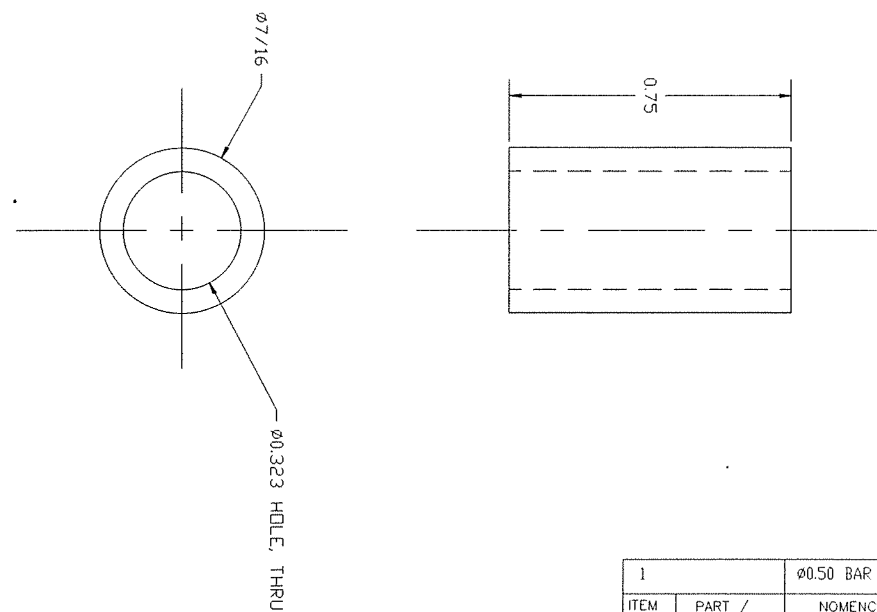
PART LIST

NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400

UNLESS OTHERWISE NOTED: TOLERANCES .XX ±.01 .XXX ±.005 ANGLES ±0.5° 64√	UNIVERSITY OF CALIFORNIA, DAVIS MECHANICAL ENGINEERING DEPARTMENT
DESIGN WWN/KSS · DATE 2/28/94	TITLE CARRIER PISTON
DRAWN WWN	QTY 3 · DRAWING NO LFN-0003
APPRV WWN	SCALE 1:1
RELEASE	SIZE B · SHT · PROJECT AHMCT/LFN


REVISION
LTR | ZONE | DESCRIPTION | DATE | DRWN | APP
A | INITIAL RELEASE | 3/3/94 | WWN | YES
21.75
21.25
16.125
15.25
11.25
5.25
1.25
1.00
0.50
2.25
2.75
1.25
9x ⌀0.323 HOLE, THRU
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT
TO MILLIMETERS MULTIPLY BY 25.400
1 | 3"x1/4"x22" | 6061-T6 ALUMINUM | 6
ITEM NO. | PART / ID NO. | NOMENCLATURE / DESCRIPTION | MATERIAL / SPECIFICATION | QTY REQD
PART LIST
UNLESS OTHERWISE NOTED TOLERANCES
.XX ± .01 ANGLES ± 0.5°
.XXX ± .002
64
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE PLATE BETWEEN CARRIER AND GUNS
DESIGN WWN/KSS
DATE 3/3/94
DRAWN WWN
APPRV WWN
RELEASE
QTY 6
SCALE 1:1
DRAWING NO LFN0004
SIZE C
SHT
PROJECT AHMCT/LFN

REVISION					
LTR	ZONE	DESCRIPTION	DATE	DRWN	APP
A		INITIAL RELEASE	2/28/94	WWN	YES



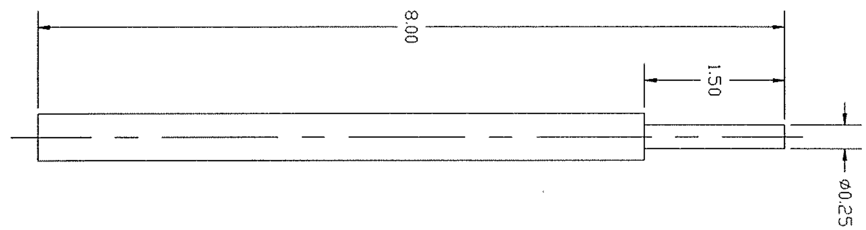
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400

1		Ø0.50 BAR 0.75" LONG	6061-T6 ALUMINUM	12
ITEM NO.	PART / ID NO.	NOMENCLATURE / DESCRIPTION	MATERIAL / SPECIFICATION	QTY REQD

PART LIST

UNLESS OTHERWISE NOTED: TOLERANCES .XX ± .01 .XXX ± .002 ANGLES ±0.5° 64√		UNIVERSITY OF CALIFORNIA, DAVIS MECHANICAL ENGINEERING DEPARTMENT		
		TITLE SPACER BETWEEN CARRIER PISTON AND PLATE		
DESIGN WWN	DATE 2/28/94			
DRAWN WWN		QTY 12	DRAWING NO LFN-0005	
APPRV WWN		SCALE 4:1		
RELEASE		SIZE B	SHT	PROJECT AHMCT/LFN

REVISION					
LTR	ZONE	DESCRIPTION	DATE	DRWN	APP
A		INITIAL RELEASE	3/2/94	WWN	YES



ITEM NO.	PART / ID NO.	NOMENCLATURE / DESCRIPTION	MATERIAL / SPECIFICATION	QTY REQD
1		1/2-20 THD. ROD 8" LONG	BRASS THD. ROD	3

PART LIST

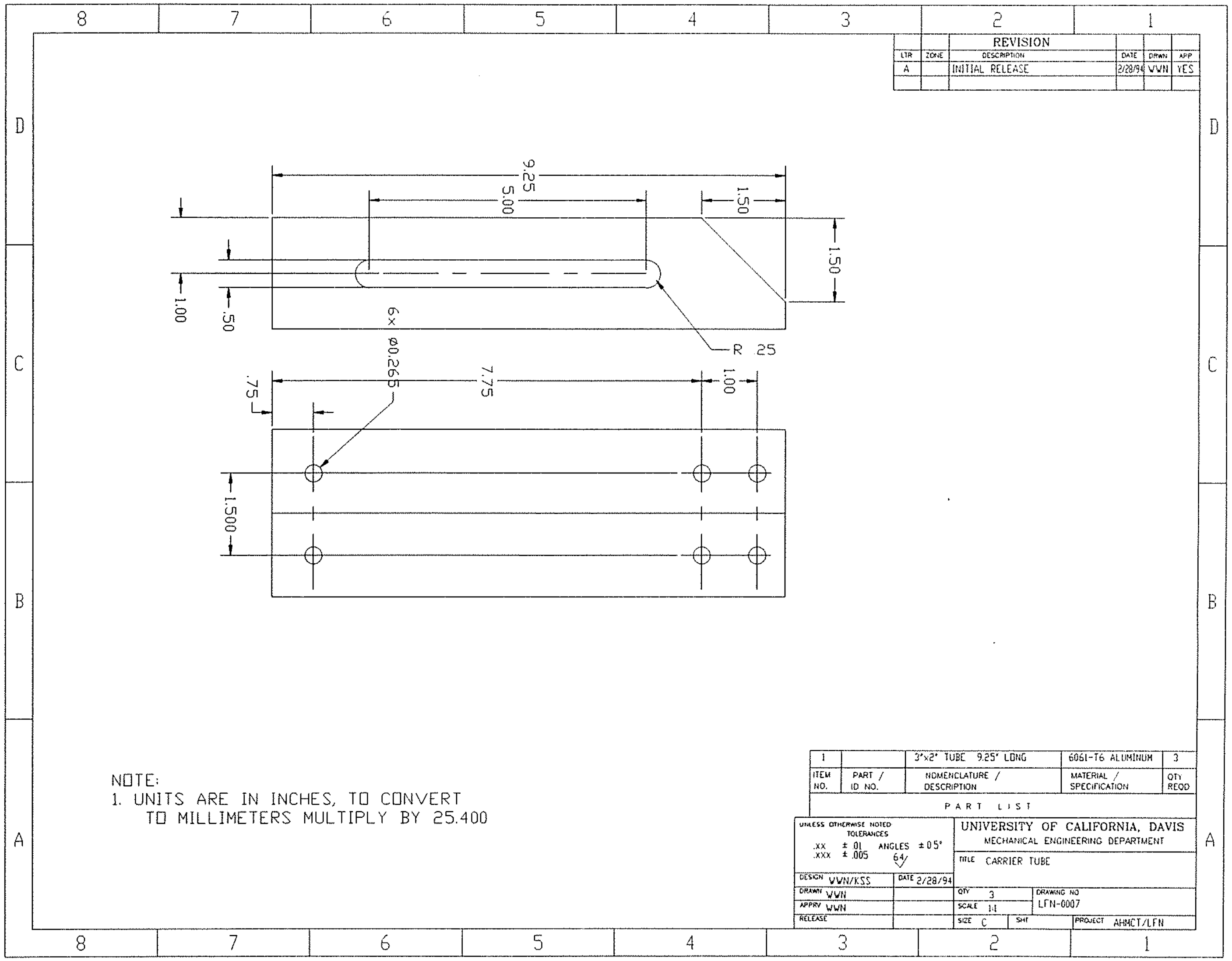
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400

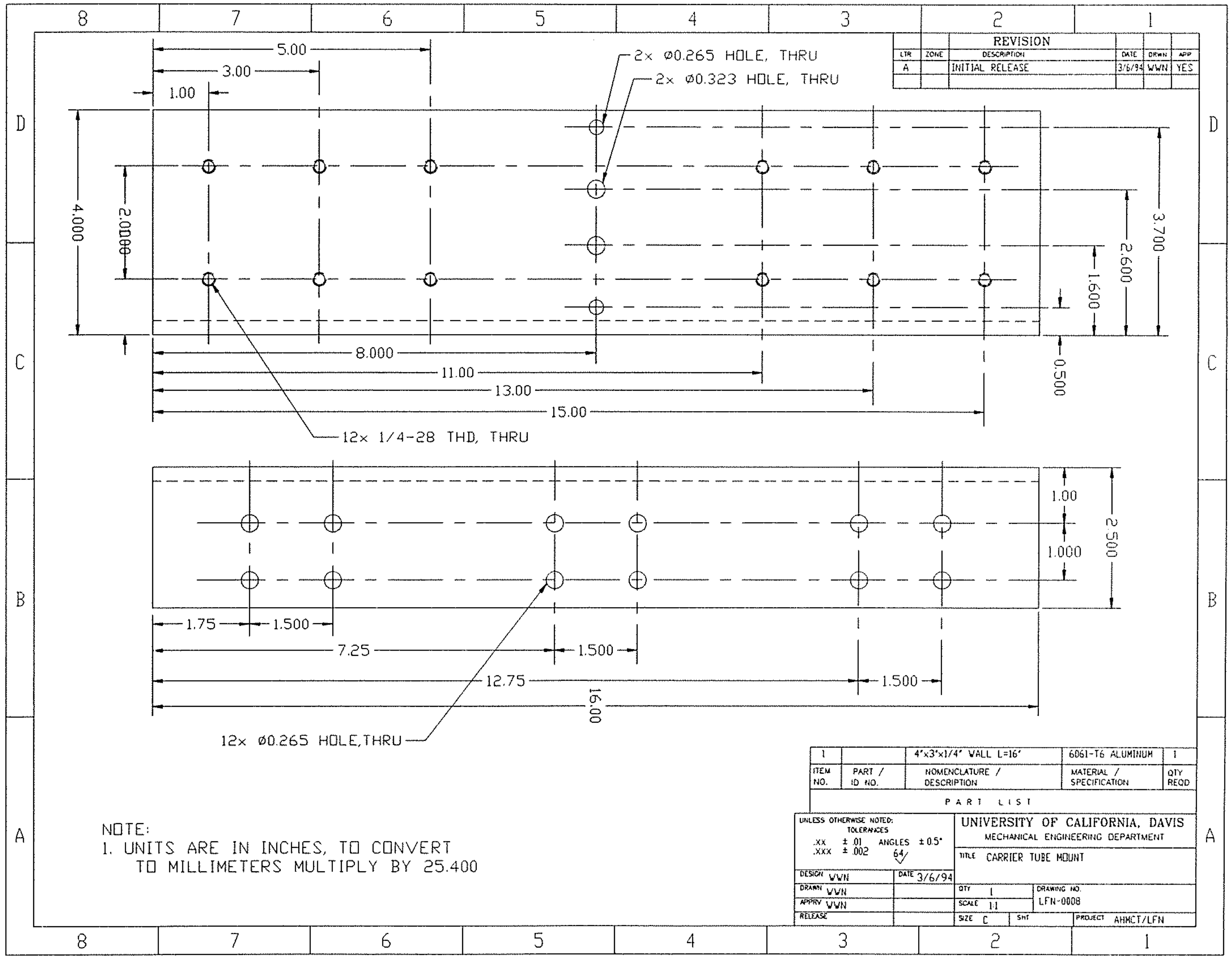
UNLESS OTHERWISE NOTED:
TOLERANCES
.XX ±.01
.XXX ±.002
ANGLES ±0.5°
64√

UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT

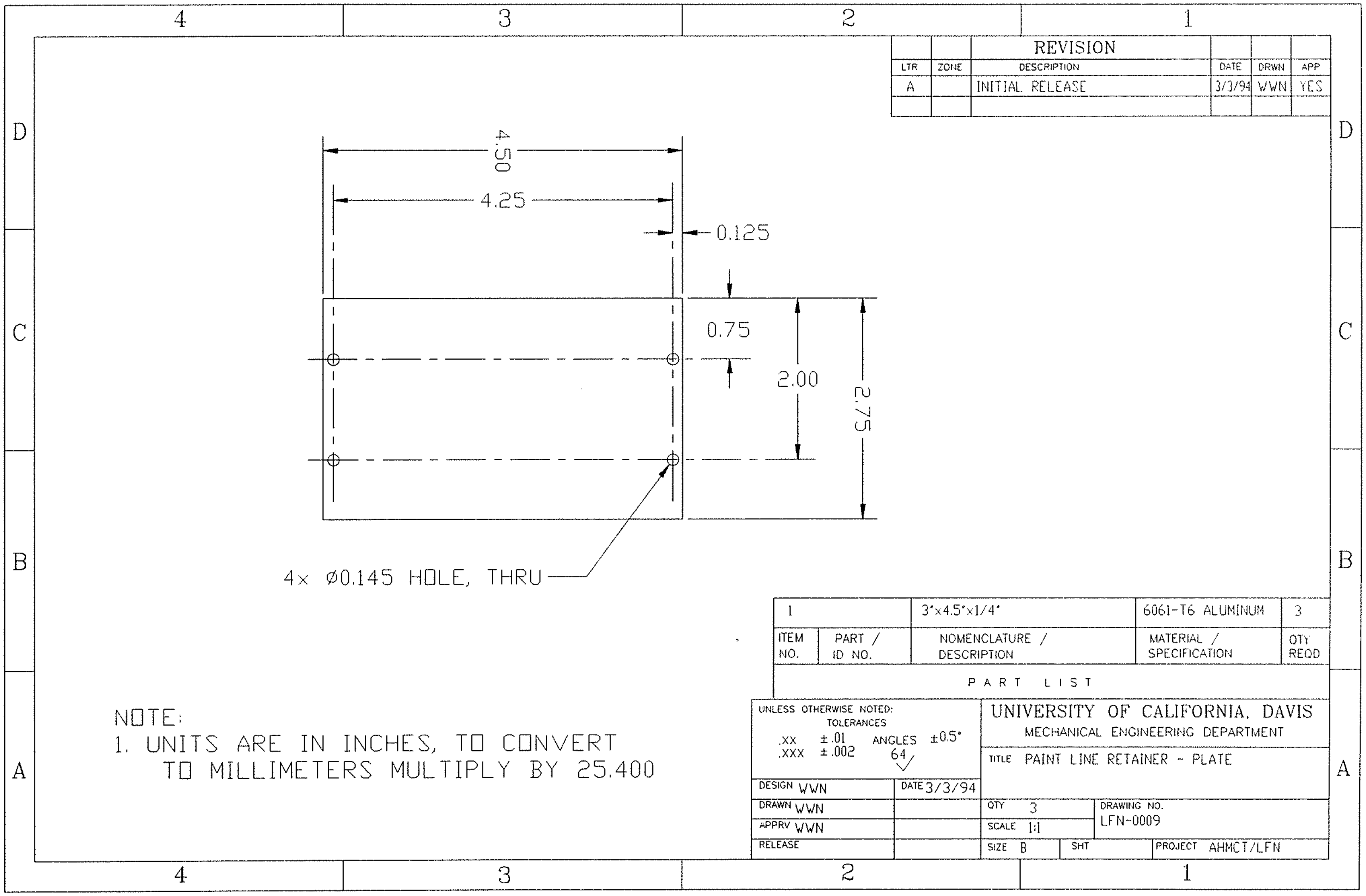
TITLE THREADED ROD MODIFICATIONS

DESIGN WWN/KSS	DATE 3/2/94		
DRAWN WWN		QTY 3	DRAWING NO. LFN-0006
APPRV WWN		SCALE 1:1	
RELEASE		SIZE B	SHT	PROJECT AHMCT/LFN

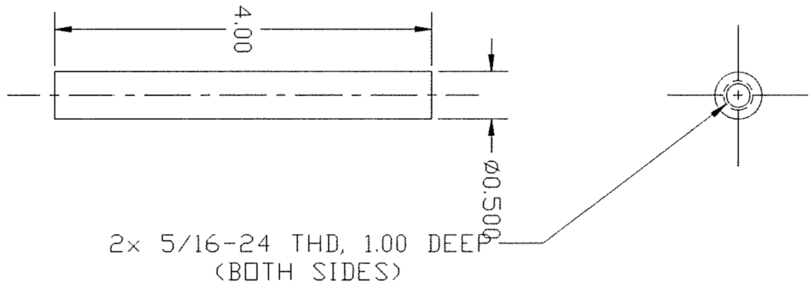

REVISION
LTR | ZONE | DESCRIPTION | DATE | DRWN | APP
A | INITIAL RELEASE | 2/28/94 | VWN | YES
9.25
5.00
1.50
1.50
1.00
.50
R .25
6x ø0.265
.75
7.75
1.00
1.500
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT
TO MILLIMETERS MULTIPLY BY 25.400
1 | 3"x2" TUBE 9.25" LONG | 6061-T6 ALUMINUM | 3
ITEM NO. | PART / ID NO. | NOMENCLATURE / DESCRIPTION | MATERIAL / SPECIFICATION | QTY REQD
PART LIST
UNLESS OTHERWISE NOTED
TOLERANCES
.XX ± .01 ANGLES ± 0.5°
.XXX ± .005
64
DESIGN VWN/KSS
DATE 2/28/94
DRAWN VWN
APPRV VWN
RELEASE
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE CARRIER TUBE
QTY 3
DRAWING NO LFN-0007
SCALE 1:1
SIZE C
SHT
PROJECT AHMCT/LFN



REVISION
LTR | ZONE | DESCRIPTION | DATE | DRWN | APP
A | | INITIAL RELEASE | 3/6/94 | WWN | YES
2× Ø0.265 HOLE, THRU
2× Ø0.323 HOLE, THRU
12× 1/4-28 THD, THRU
12× Ø0.265 HOLE,THRU
5.00
3.00
1.00
4.000
2.000
8.000
11.00
13.00
15.00
3.700
2.600
1.600
0.500
1.00
1.000
2.500
1.75
1.500
7.25
12.75
16.00
1 | | 4"x3"x1/4" WALL L=16" | 6061-T6 ALUMINUM | 1
ITEM NO. | PART / ID NO. | NOMENCLATURE / DESCRIPTION | MATERIAL / SPECIFICATION | QTY REQD
PART LIST
UNLESS OTHERWISE NOTED: TOLERANCES .XX ± .01 .XXX ± .002 ANGLES ± 0.5° 64
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE CARRIER TUBE MOUNT
DESIGN VWN DATE 3/6/94
DRAWN VWN
APPRV VWN
RELEASE
QTY 1 DRAWING NO. LFN-0008
SCALE 1:1
SIZE C SHT PROJECT AHMCT/LFN
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400


REVISION
LTR
ZONE
DESCRIPTION
DATE
DRWN
APP
A
INITIAL RELEASE
3/3/94
WWN
YES
4.50
4.25
0.125
0.75
2.00
2.75
4× Ø0.145 HOLE, THRU
1
3"×4.5"×1/4"
6061-T6 ALUMINUM
3
ITEM NO.
PART / ID NO.
NOMENCLATURE / DESCRIPTION
MATERIAL / SPECIFICATION
QTY REQD
PART LIST
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400
UNLESS OTHERWISE NOTED:
TOLERANCES
.XX ± .01
.XXX ± .002
ANGLES ±0.5°
64
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE PAINT LINE RETAINER - PLATE
DESIGN WWN
DATE 3/3/94
DRAWN WWN
APPRV WWN
RELEASE
QTY 3
SCALE 1:1
SIZE B
SHT
DRAWING NO. LFN-0009
PROJECT AHMCT/LFN

REVISION					
LTR	ZONE	DESCRIPTION	DATE	DRWN	APP
A		INITIAL RELEASE	3/7/94	WWN	YES



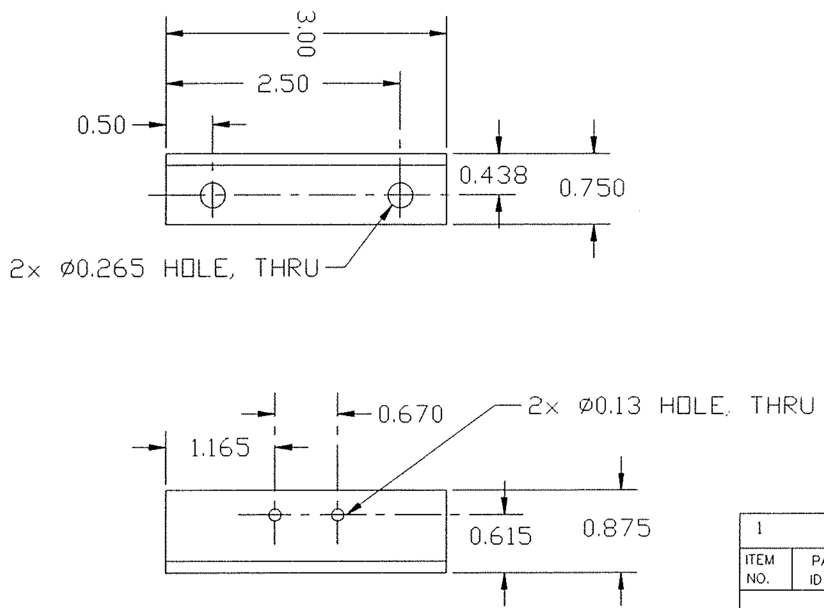
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400

ITEM NO.	PART / ID NO.	NOMENCLATURE / DESCRIPTION	MATERIAL / SPECIFICATION	QTY REQD
1		Ø0.500 BAR x L=4"	6061-T6 ALUMINUM	9

PART LIST

UNLESS OTHERWISE NOTED: TOLERANCES .XX ±.01 .XXX ±.002 ANGLES ±0.5° 64√		UNIVERSITY OF CALIFORNIA, DAVIS MECHANICAL ENGINEERING DEPARTMENT	
		TITLE BAR TO HOLD GUN & SPACER	
DESIGN WWN/KSS	DATE 3/7/94		
DRAWN WWN		QTY 9	DRAWING NO. LFN-0010
APPRV WWN		SCALE 1:1	
RELEASE		SIZE B / SHT	PROJECT AHMCT/LFN

REVISION					
LTR	ZONE	DESCRIPTION	DATE	DRWN	APP
A		INITIAL RELEASE	3/7/94	WWN	YES

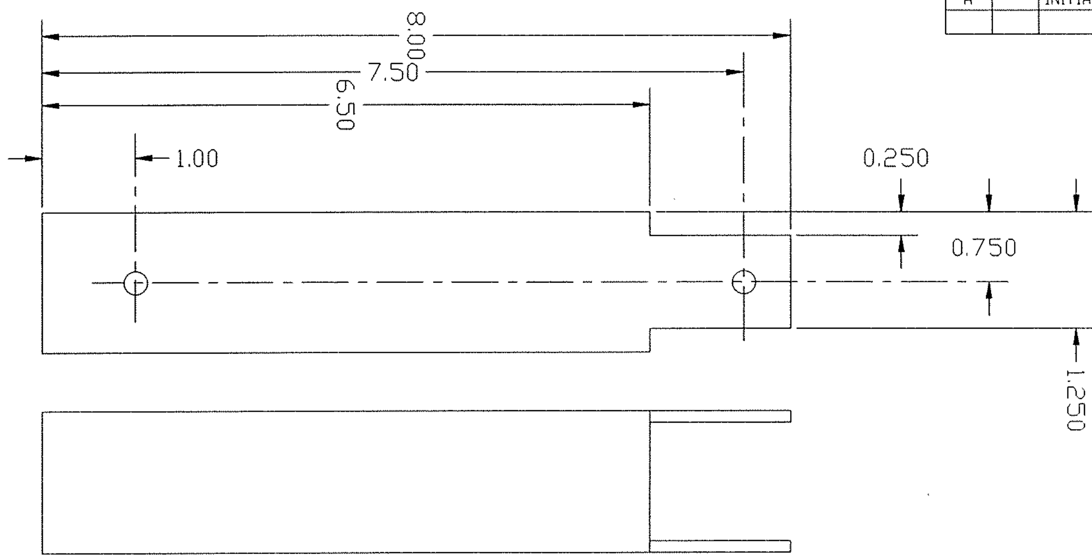


1		1"x1"x1/8"W ANGLE L=3"	6061-T6 ALUMINUM	3
ITEM NO.	PART / ID NO.	NOMENCLATURE / DESCRIPTION	MATERIAL / SPECIFICATION	QTY REQD

PART LIST

NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400

UNLESS OTHERWISE NOTED: TOLERANCES .XX ±.01 .XXX ±.002 ANGLES ±0.5° 64√		UNIVERSITY OF CALIFORNIA, DAVIS MECHANICAL ENGINEERING DEPARTMENT	
		TITLE VALVE HOLDER	
DESIGN WWN	DATE 3/7/94		
DRAWN WWN		QTY 3	DRAWING NO. LFN-0011
APPRV WWN		SCALE 1:1	
RELEASE		SIZE B SHT	PROJECT AHMCT/LFN



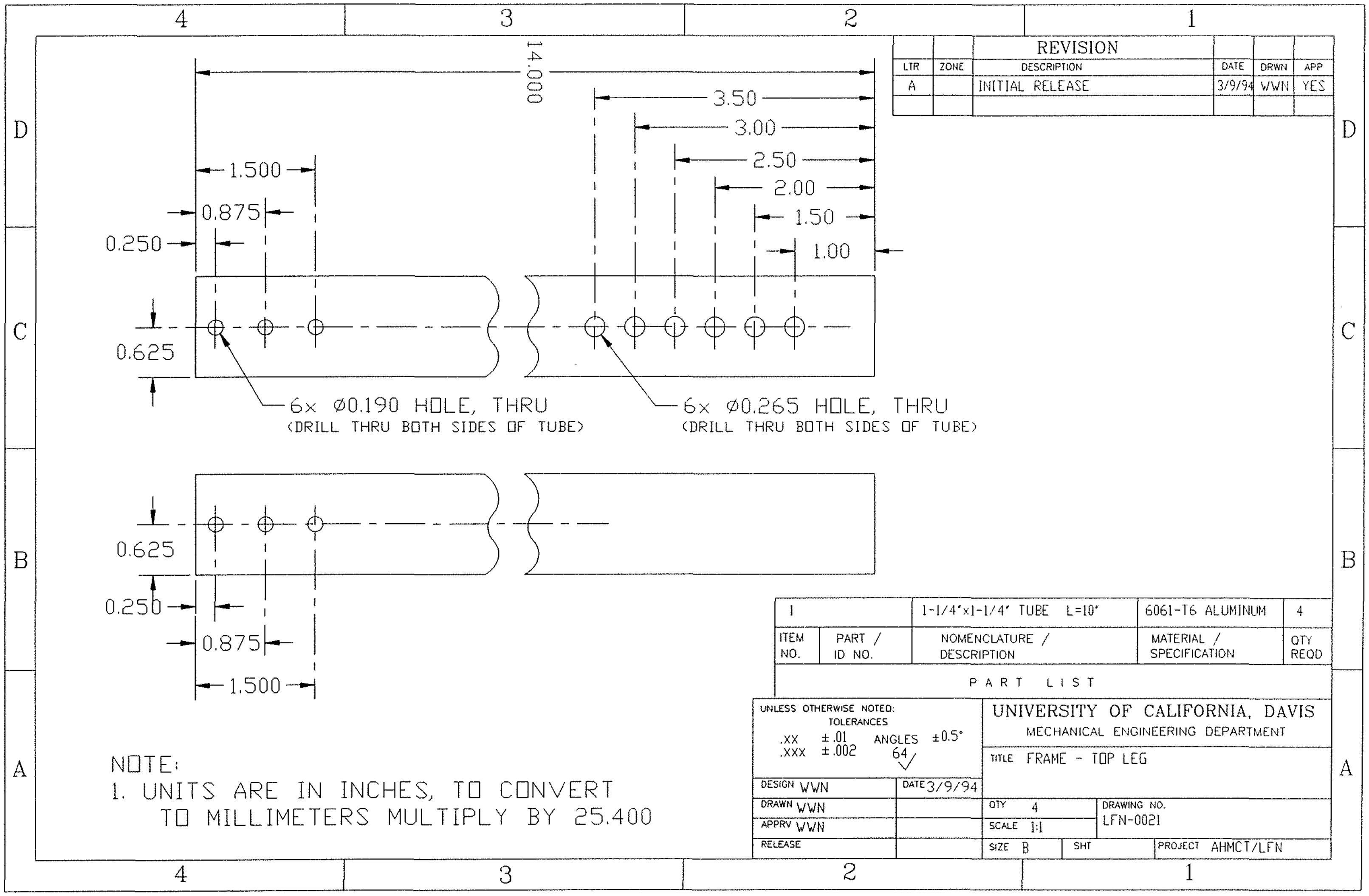
REVISION					
LTR	ZONE	DESCRIPTION	DATE	DRWN	APP
A		INITIAL RELEASE	3/9/94	WWN	YES

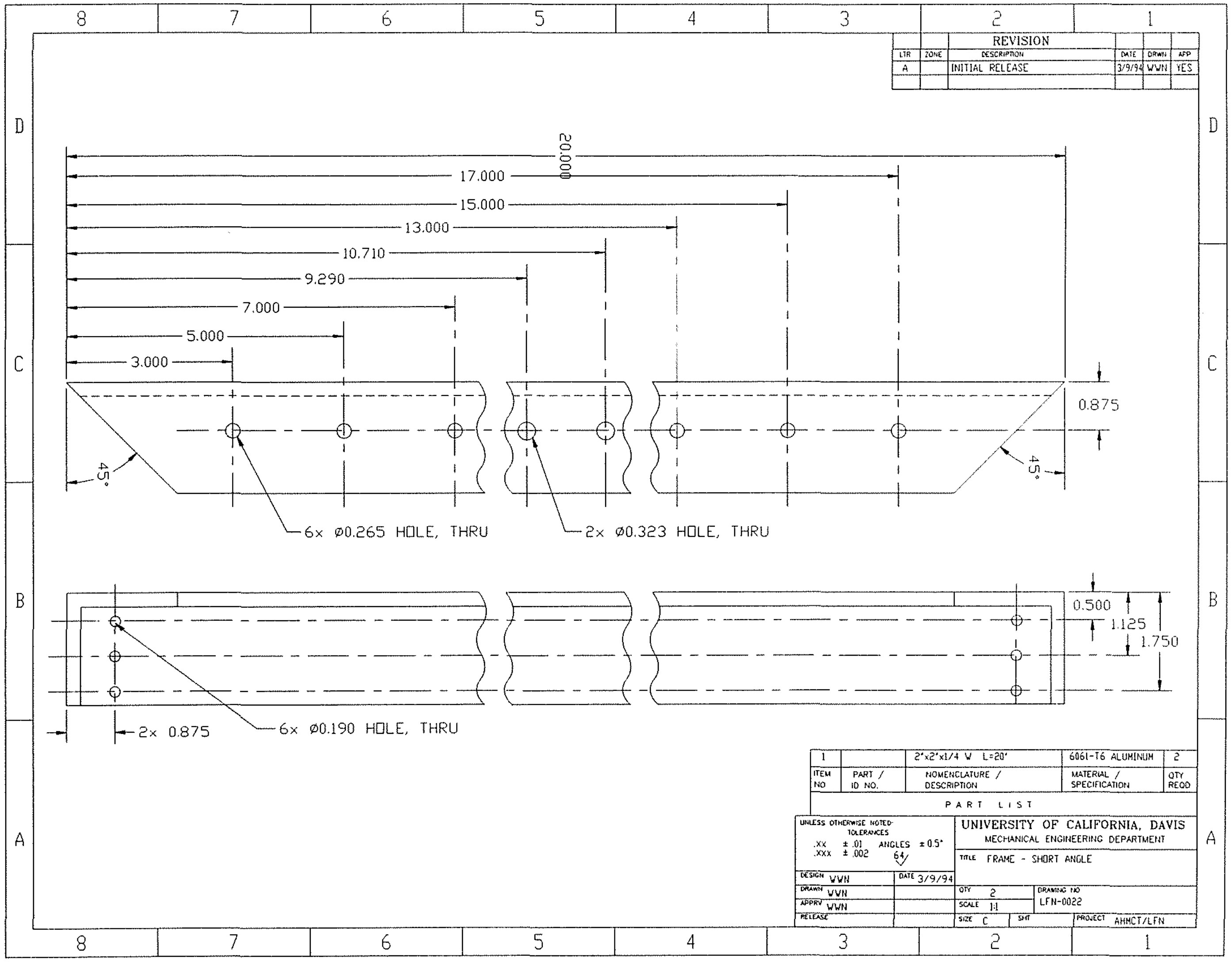
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400

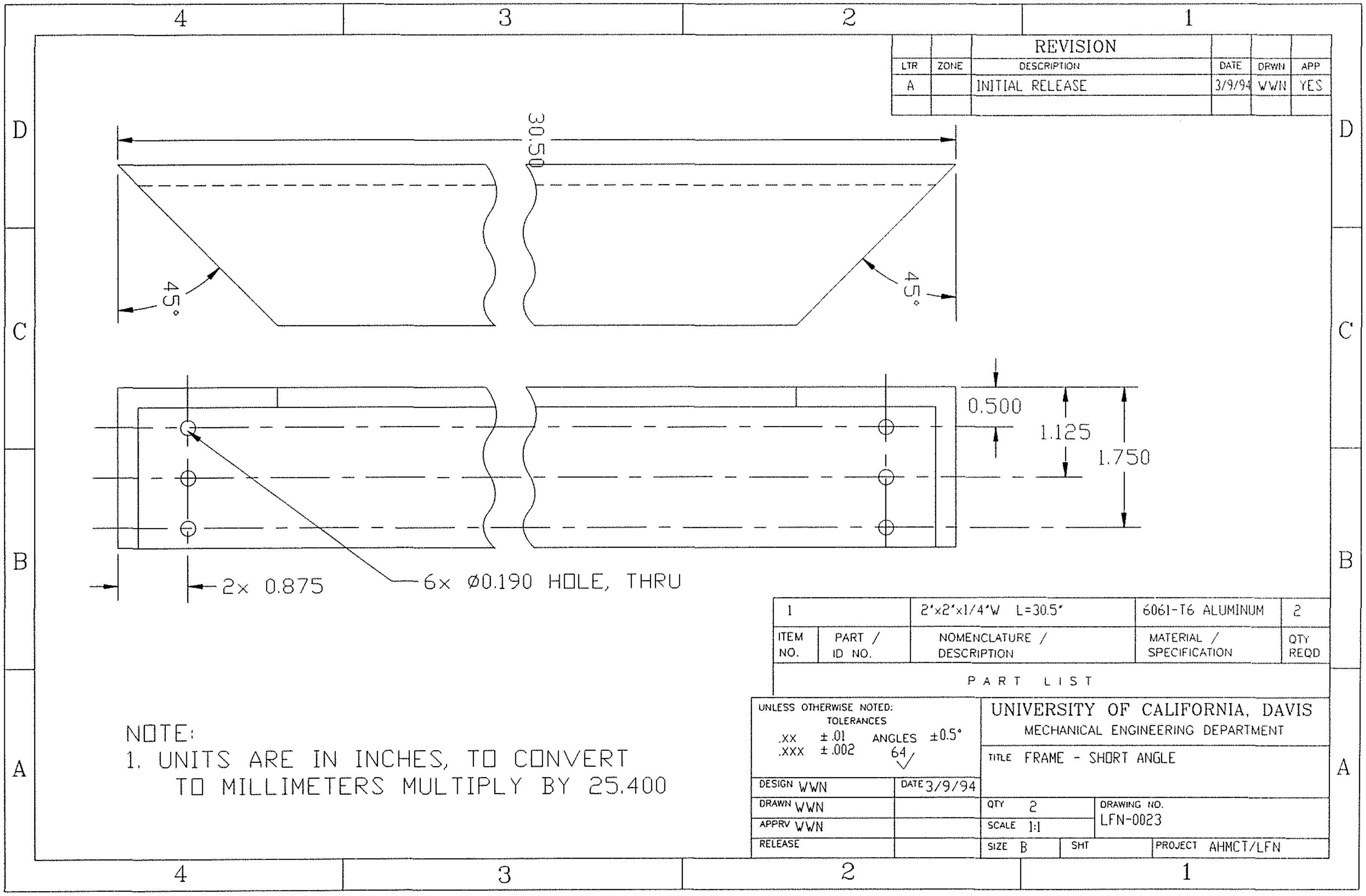
ITEM NO.	PART / ID NO.	NOMENCLATURE / DESCRIPTION	MATERIAL / SPECIFICATION	QTY REQD
1		1-1/2"x1-1/2" TUBE L=8"	6061-T6 ALUMINUM	4

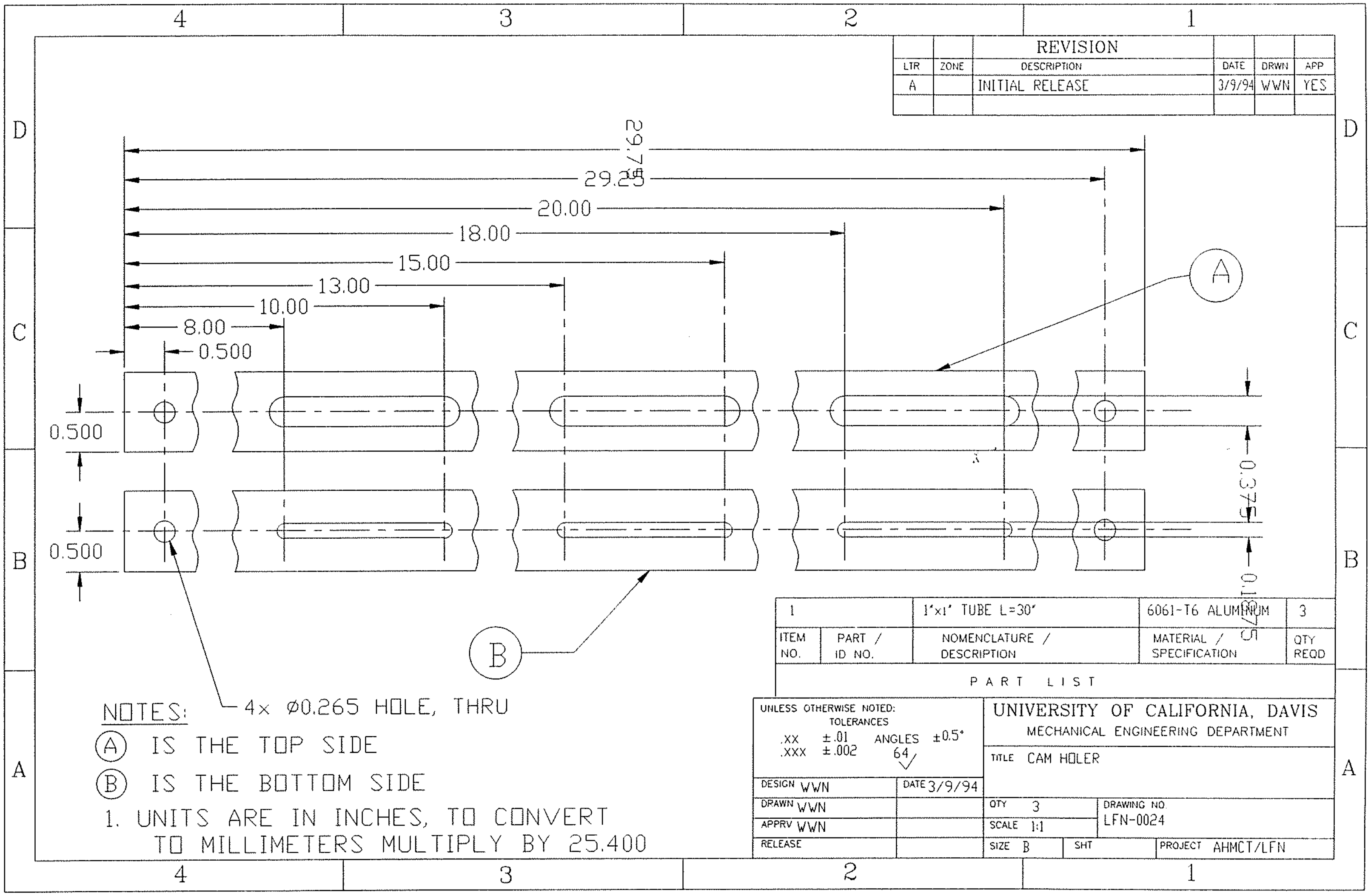
PART LIST

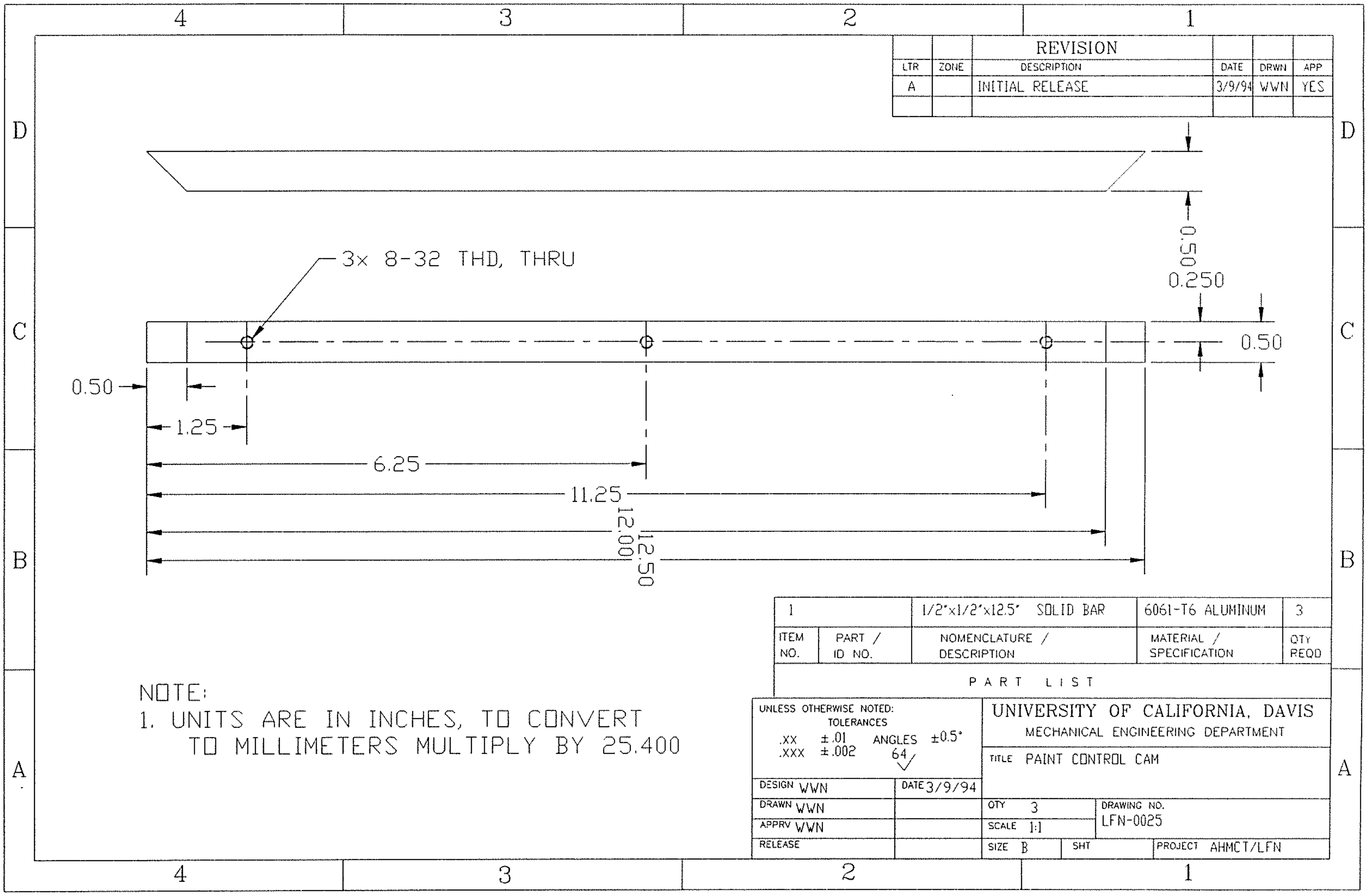
UNLESS OTHERWISE NOTED: TOLERANCES .XX ±.01 .XXX ±.002 ANGLES ±0.5° 64√	UNIVERSITY OF CALIFORNIA, DAVIS MECHANICAL ENGINEERING DEPARTMENT
DESIGN WWN DATE 3/9/94	TITLE FRAME - BOTTOM LEG
DRAWN WWN	QTY 4 DRAWING NO. LFN-0020
APPRV WWN	SCALE 1:1
RELEASE	SIZE B SHT PROJECT AHMCT/LFN


14.000
3.50
3.00
2.50
2.00
1.50
1.00
1.500
0.875
0.250
0.625
6x Ø0.190 HOLE, THRU
(DRILL THRU BOTH SIDES OF TUBE)
6x Ø0.265 HOLE, THRU
(DRILL THRU BOTH SIDES OF TUBE)
0.625
0.250
0.875
1.500
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT
TO MILLIMETERS MULTIPLY BY 25.400
REVISION
LTR | ZONE | DESCRIPTION | DATE | DRWN | APP
A | | INITIAL RELEASE | 3/9/94 | WWN | YES
1 | | 1-1/4"x1-1/4" TUBE L=10" | 6061-T6 ALUMINUM | 4
ITEM NO. | PART / ID NO. | NOMENCLATURE / DESCRIPTION | MATERIAL / SPECIFICATION | QTY REQD
PART LIST
UNLESS OTHERWISE NOTED:
TOLERANCES
.XX ± .01
.XXX ± .002
ANGLES ±0.5°
64
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE FRAME - TOP LEG
DESIGN WWN
DATE 3/9/94
DRAWN WWN
APPRV WWN
RELEASE
QTY 4
SCALE 1:1
SIZE B
SHT
DRAWING NO. LFN-0021
PROJECT AHMCT/LFN
A
B
C
D
1
2
3
4

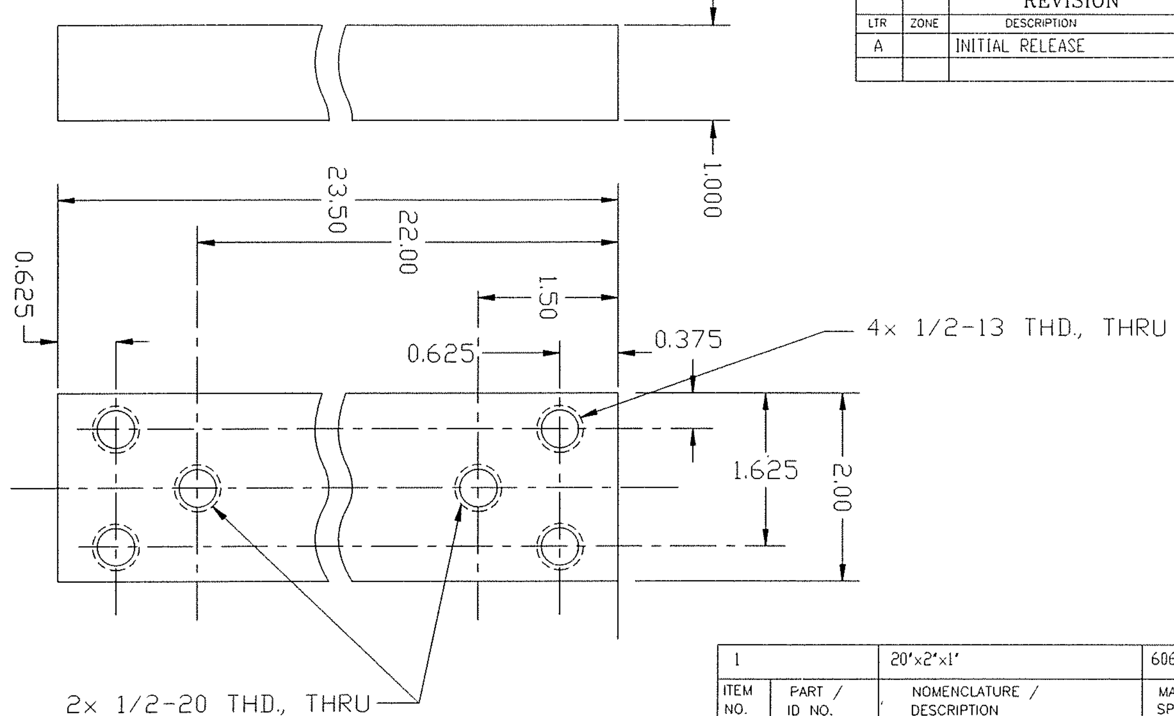

REVISION
LTR ZONE DESCRIPTION DATE DRWN APP
A INITIAL RELEASE 3/9/94 WWN YES
20.000
17.000
15.000
13.000
10.710
9.290
7.000
5.000
3.000
0.875
45°
45°
6x Ø0.265 HOLE, THRU
2x Ø0.323 HOLE, THRU
0.500
1.125
1.750
2x 0.875
6x Ø0.190 HOLE, THRU
1 2"x2"x1/4 W L=20' 6061-T6 ALUMINUM 2
ITEM NO PART / ID NO. NOMENCLATURE / DESCRIPTION MATERIAL / SPECIFICATION QTY REQD
PART LIST
UNLESS OTHERWISE NOTED: TOLERANCES .XX ± .01 .XXX ± .002 ANGLES ± 0.5° 64
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE FRAME - SHORT ANGLE
DESIGN WWN DATE 3/9/94
DRAWN WWN
APPRV WWN
RELEASE
QTY 2
SCALE 1:1
DRAWING NO LFN-0022
SIZE C
SHT
PROJECT AHMCT/LFN


REVISION
LTR ZONE DESCRIPTION DATE DRWN APP
A INITIAL RELEASE 3/9/94 WWN YES
30.50
45°
45°
0.500
1.125
1.750
2x 0.875
6x Ø0.190 HOLE, THRU
1 2"x2"x1/4"W L=30.5" 6061-T6 ALUMINUM 2
ITEM NO. PART / ID NO. NOMENCLATURE / DESCRIPTION MATERIAL / SPECIFICATION QTY REQD
PART LIST
UNLESS OTHERWISE NOTED: TOLERANCES
.XX ± .01 .XXX ± .002 ANGLES ±0.5° 64
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE FRAME - SHORT ANGLE
DESIGN WWN DATE 3/9/94
DRAWN WWN
APPRV WWN
RELEASE
QTY 2
SCALE 1:1
DRAWING NO. LFN-0023
SIZE B SHT PROJECT AHMCT/LFN
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400


REVISION
LTR ZONE DESCRIPTION DATE DRWN APP
A INITIAL RELEASE 3/9/94 WWN YES
29.75
29.25
20.00
18.00
15.00
13.00
10.00
8.00
0.500
0.500
0.500
0.375
0.1875
A
B
4× Ø0.265 HOLE, THRU
NOTES:
(A) IS THE TOP SIDE
(B) IS THE BOTTOM SIDE
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400
1 1"x1" TUBE L=30" 6061-T6 ALUMINUM 3
ITEM NO. PART / ID NO. NOMENCLATURE / DESCRIPTION MATERIAL / SPECIFICATION QTY REQD
PART LIST
UNLESS OTHERWISE NOTED: TOLERANCES
.XX ±.01 .XXX ±.002 ANGLES ±0.5° 64
DESIGN WWN DATE 3/9/94
DRAWN WWN
APPRV WWN
RELEASE
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE CAM HOLER
QTY 3
SCALE 1:1
DRAWING NO. LFN-0024
SIZE B
SHT
PROJECT AHMCT/LFN


REVISION
LTR | ZONE | DESCRIPTION | DATE | DRWN | APP
A | | INITIAL RELEASE | 3/9/94 | WWN | YES
3x 8-32 THD, THRU
0.50
0.250
0.50
0.50
1.25
6.25
11.25
12.00
12.50
1 | 1/2"x1/2"x12.5" SOLID BAR | 6061-T6 ALUMINUM | 3
ITEM NO. | PART / ID NO. | NOMENCLATURE / DESCRIPTION | MATERIAL / SPECIFICATION | QTY REQD
PART LIST
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400
UNLESS OTHERWISE NOTED: TOLERANCES
.XX ±.01
.XXX ±.002
ANGLES ±0.5°
64
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE PAINT CONTROL CAM
DESIGN WWN
DATE 3/9/94
DRAWN WWN
APPRV WWN
RELEASE
QTY 3
SCALE 1:1
DRAWING NO. LFN-0025
SIZE B
SHT
PROJECT AHMCT/LFN

REVISION						
LTR	ZONE	DESCRIPTION	DATE	DRWN	APP
A		INITIAL RELEASE	5/2/94	WWN	YES

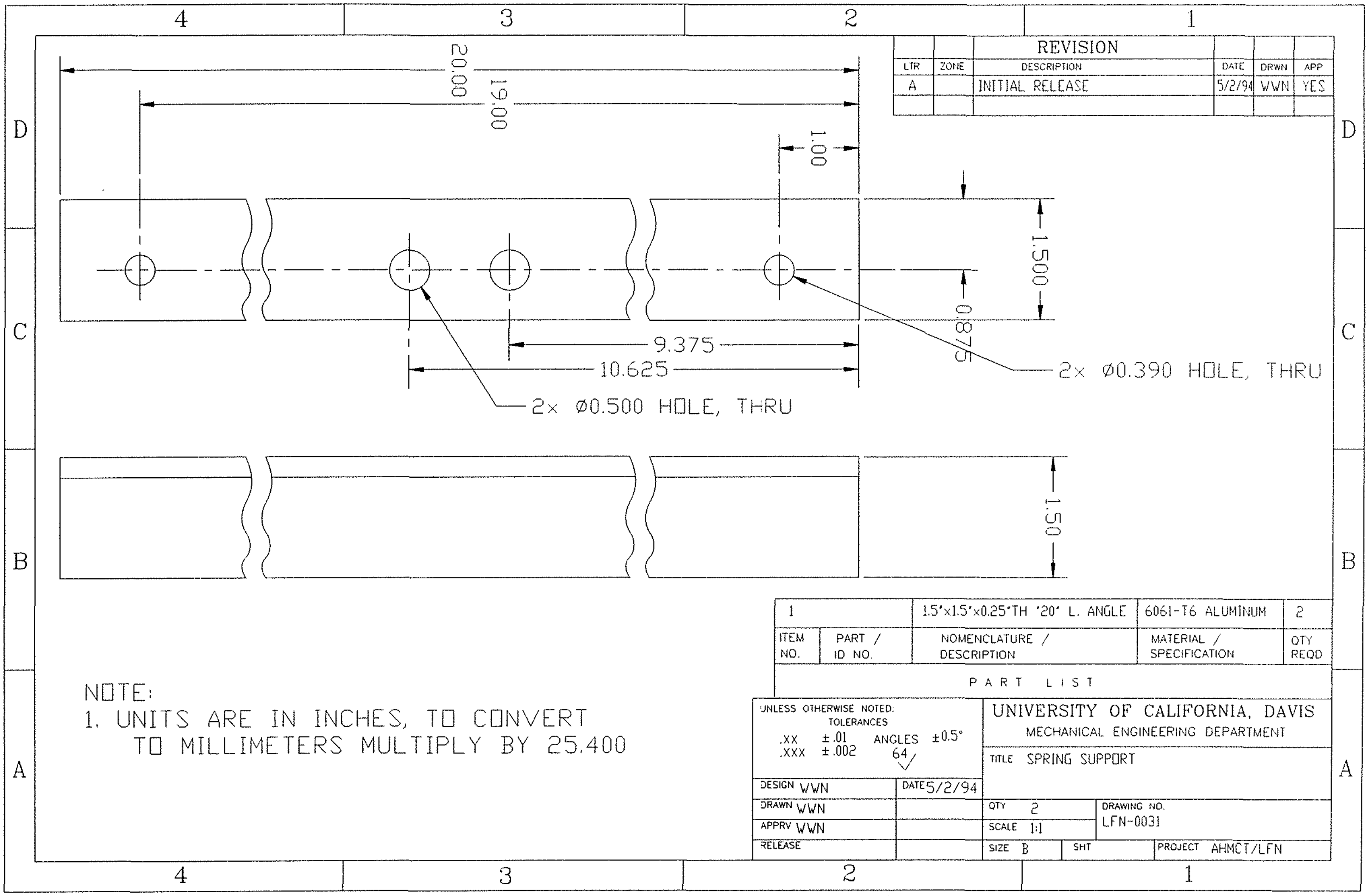


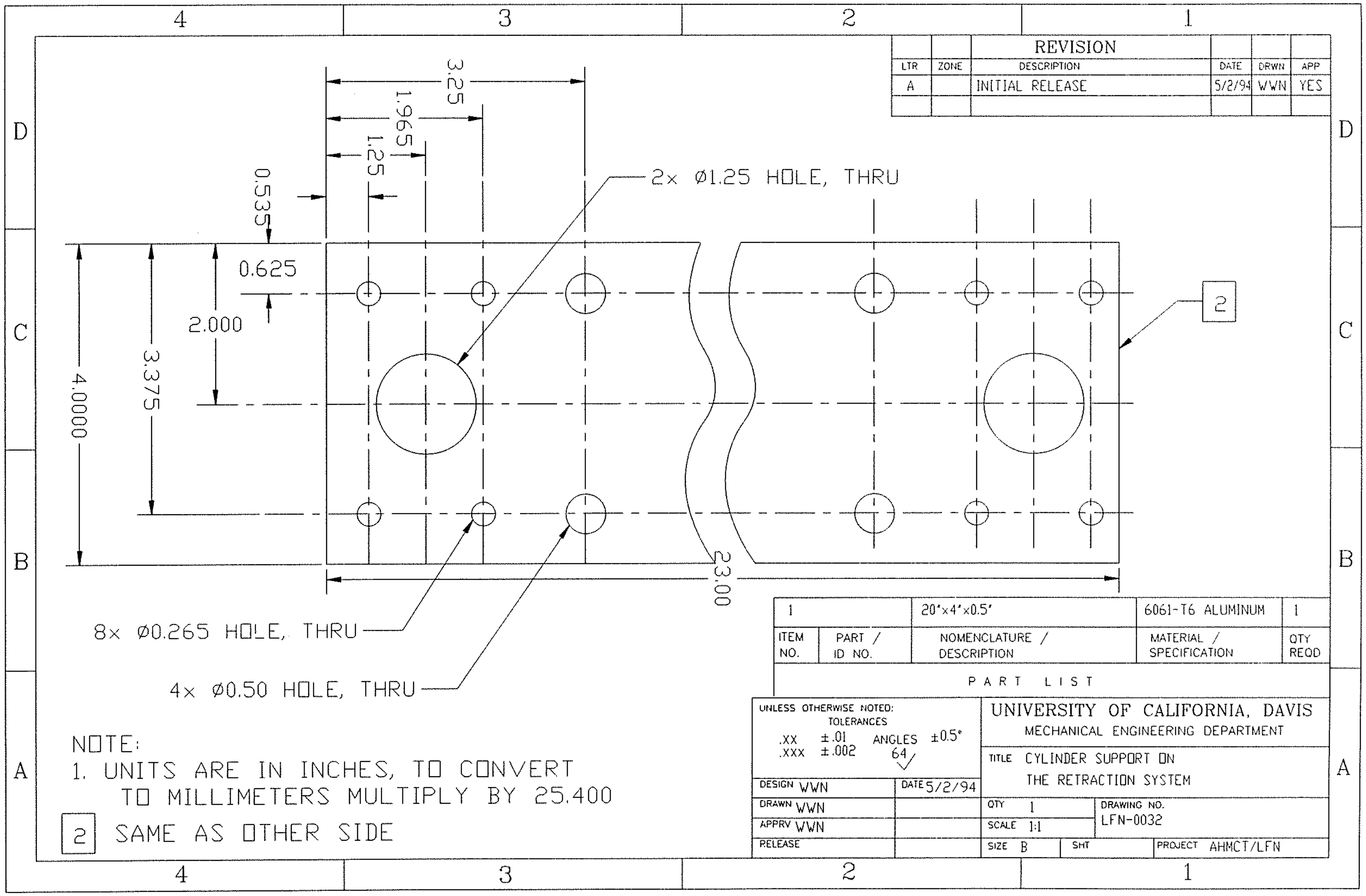
ITEM NO.	PART / ID NO.	NOMENCLATURE / DESCRIPTION	MATERIAL / SPECIFICATION	QTY REQD
1		20'x2'x1'	6061-T6 ALUMINUM	1

PART LIST

NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400

UNLESS OTHERWISE NOTED: TOLERANCES .XX ±.01 .XXX ±.002 ANGLES ±0.5° 64 √		UNIVERSITY OF CALIFORNIA, DAVIS MECHANICAL ENGINEERING DEPARTMENT	
		TITLE RETRACTION SUPPORT BEAM	
DESIGN WWN	DATE 5/2/94		
DRAWN WWN		QTY 1	DRAWING NO. LFN-0030
APPRV WWN		SCALE 1:1	
RELEASE		SIZE B / SHT	PROJECT AHMCT/LFN


REVISION
LTR | ZONE | DESCRIPTION | DATE | DRWN | APP
A | INITIAL RELEASE | 5/2/94 | WWN | YES
20.00
19.00
1.00
1.500
0.875
9.375
10.625
2x Ø0.500 HOLE, THRU
2x Ø0.390 HOLE, THRU
1.50
1 | 1.5"x1.5"x0.25"TH '20' L. ANGLE | 6061-T6 ALUMINUM | 2
ITEM NO. | PART / ID NO. | NOMENCLATURE / DESCRIPTION | MATERIAL / SPECIFICATION | QTY REQD
PART LIST
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400
UNLESS OTHERWISE NOTED: TOLERANCES
.XX ±.01
.XXX ±.002
ANGLES ±0.5°
64
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE SPRING SUPPORT
DESIGN WWN
DATE 5/2/94
DRAWN WWN
APPRV WWN
RELEASE
QTY 2
SCALE 1:1
DRAWING NO. LFN-0031
SIZE B
SHT
PROJECT AHMCT/LFN



REVISION
LTR ZONE DESCRIPTION DATE DRWN APP
A INITIAL RELEASE 5/2/94 WWN YES
3.25
1.965
1.25
0.535
0.625
2.000
3.375
4.0000
23.00
2x Ø1.25 HOLE, THRU
8x Ø0.265 HOLE, THRU
4x Ø0.50 HOLE, THRU
2
1 20"x4"x0.5" 6061-T6 ALUMINUM 1
ITEM NO. PART / ID NO. NOMENCLATURE / DESCRIPTION MATERIAL / SPECIFICATION QTY REQD
PART LIST
UNLESS OTHERWISE NOTED: TOLERANCES
.XX ±.01
.XXX ±.002
ANGLES ±0.5°
64
UNIVERSITY OF CALIFORNIA, DAVIS
MECHANICAL ENGINEERING DEPARTMENT
TITLE CYLINDER SUPPORT ON THE RETRACTION SYSTEM
DESIGN WWN DATE 5/2/94
DRAWN WWN
APPRV WWN
RELEASE
QTY 1
SCALE 1:1
DRAWING NO. LFN-0032
SIZE B
SHT
PROJECT AHMCT/LFN
NOTE:
1. UNITS ARE IN INCHES, TO CONVERT TO MILLIMETERS MULTIPLY BY 25.400
2 SAME AS OTHER SIDE

APPENDIX C:
SOFTWARE LISTING

```
/*==========================================================================
PROGRAM
  premark.c

SYNPOSIS
  this program basically causes a z-world controller to act like
  a teach pendant, and a controller to an iai gantry robot.
  the teach pendant part cannot be taught. not yet at least.
  the program will do both a large premark and an abbreviated one.
  this program can also control the pumps and the mixers.

USES
  this program uses all global variables to minimize the stack usage.
  the important ones are commented below.

  - StatusWords: this contains the status word returned by the iai
               controller. this is all text. see manual for meaning
               of the characters.
  - ActPos: the actuator position in 0.01 mm units. this is a number.
  - WantedActPos: the requested actuator position in 0.01 mm units. this
                is a number.
  - manual, jog: if in these modes they are defined as ON.
  - TotalPoints: total number of points uploaded to the controller.
  - acceleration, velocity: desired acceleration (0.01G units) and
                        desired velocity (mm/sec)
  - Input, Output: input and output string for serial communications
  - InCount, OutCount: the length of the input and output string.
  - increment: increment to move in increment mode (0.01 mm units)
  - AxesPattern, Direction: axes pattern to move and direction. see manual
                        for the exact character for the desired axis
                        pattern.
  - SolndPattern: expand to binary and negate and you'll get an idea of
                which solenoids are open.


AUTHOR
  P.W. Wong

VERSION
  0.10 09/09/93  proved serial communications possible to iai controller
  0.20 09/10/93
  0.30 09/13/93  implemented keypad decoding
  0.40 09/14/93  implemented home command
  0.50 09/15/93  implemented control over acceleration and velocity
  0.60 09/16/93  implemented increment mode
  0.70 09/17/93  implemented jog mode
  0.80 09/20/93  cleaned up jog mode commands.
  0.90 09/22/93  implemented the automatic modes
  0.91 09/30/93  fixed automatic modes to use robot path planning
  0.99 12/15/93  final prototype code

THINGS TO DO
  - fully implement downloadable paths to the iai memory card
  - get better coordinates for the premarks
```

```
 - tighten up the timing of the guns

=======================================================================*/

#include <stdio.h>

#define KEYPAD_SIZE 24
#define LK_LINES 4
#define LK_COLS 20
#define LK_BLINK 1
#define ON 1
#define OFF 0
#define FWD 1
#define REV 0
#define VERSION 0.99
#define MONTH 12
#define DAY 14
#define YEAR 93

struct CmmndFormat
 {
  char *Cmmnd;     /* command string - pad with 0 for fill-in-the-blanks */
  char ComLength;  /* transmission length in bytes */
  char RespLength; /* command response length */
  char VarCommand; /* reserved for future use. leave 0 */
 } Commands[] = {
   {"?99IS@@\r\n\0", 9, 24, 0},        /* 0: status, page 8 */
   {"?99RP@@\r\n\0", 9, 42, 0},        /* 1: actuator position, page 9 */
   {"?99/7  @@\r\n\0", 11, 9, 0},      /* 2: servo on/off, page 11 */
   {"?99/8 @@\r\n\0", 10, 9, 0},       /* 3: home, page 12 */
   {"?99/A3000000000000000@@\r\n\0", 25, 9, 0},
                              /* 4: Linear motion, page 13 */
   {"?99/B@@\r\n\0", 10, 9, 0},        /* 5: stop actuator, page 17 */
   {"?99/900000@@\r\n\0", 14, 9, 0},   /* 6: jog actuator, page 18 */
   {"?99/D0000@@\r\n\0", 13, 9, 0},    /* 7: set acceleration, page 22 */
   {"?99/P@@\r\n\0", 0, 10, 1},        /* 8: set parameter, page 33 */
   {"?99PR@@\r\n\0", 10, 0, 1},        /* 9: get parameters, page 34 */
   {"?99!!@@\r\n\0", 9, 0, 0},         /* 10: reset controller, page 38 */
   {"/99  @@\r\n\0", 9, 0, 0},         /* 11: exec. command */
   {"?99/B @@\r\n\0", 10, 9, 0},       /* 12: stop motion */
   {"?99/H000000010000@@\r\n\0", 21, 9, 0},
                              /* 13: path motion */
   {"?99/000001000000100000000000000@@\r\n\0", 34, 9, 0}};
                              /* 14: load point data */

struct PlayList
 {
  int AccelWanted;       /* acceleration desired - 0.01G units */
  int VeloWanted;        /* velocity desired - mm/sec */
  long int Actuator1Pos; /* actuator 1 position - 0.01 mm units */
  long int Actuator2Pos; /* actuator 2 position - 0.01 mm units */
  char EEPosition;       /* end effector position - 0x4 or 0x0 */
  char gun1;             /* 0x1 to open - black gun */
  char gun2;             /* 0x8 to open - white gun */
```

```
  char gun3;            /* 0x2 to open - black gun */
 };

struct PlayList Xmark[] = {
  {50, 275, 45577, 70123, 0x1, 0x0, 0x0, 0x0},
  {50, 275, 61756, 54943, 0x1, 0x0, 0x0, 0x0}, /* this is the bottom */
  {50, 100, 70824, 46075, 0x1, 0x0, 0x0, 0x0},
  {50, 100, 108220, 8480, 0x1, 0x1, 0x1, 0x1},
  {50, 100, 109220, 7480, 0x1, 0x0, 0x1, 0x1},
  {50, 100, 112220, 4480, 0x1, 0x0, 0x0, 0x0},
  {50, 100, 61756, 62156, 0x0, 0x0, 0x0, 0x0},
  {50, 100, 71824, 72224, 0x0, 0x0, 0x0, 0x0},
  {50, 100, 108220, 108620, 0x0, 0x1, 0x1, 0x1},
  {50, 100, 109220, 109620, 0x0, 0x0, 0x1, 0x1},
  {50, 100, 112220, 112620, 0x0, 0x0, 0x0, 0x0},
  {50, 100, 67224, 49475, 0x1, 0x0, 0x0, 0x0},
  {50, 275, 42575, 74524, 0x1, 0x0, 0x0, 0x0}, /* top half */
/*  {50, 275, 46975, 70124,  0x1, 0x1, 0x0, 0x1}, */
  {50, 100, 9479, 107620, 0x1, 0x1, 0x1, 0x1},
  {50, 100, 8479, 108620, 0x1, 0x0, 0x1, 0x1},
  {50, 100, 5479, 111620, 0x1, 0x0, 0x0, 0x0},
  {50, 100, 45577, 71123, 0x0, 0x0, 0x0, 0x0},
  {50, 100, 54568, 54968, 0x0, 0x0, 0x0, 0x0},
  {50, 100, 46375, 45875, 0x0, 0x0, 0x0, 0x0},
  {50, 100, 9479,   9480, 0x0, 0x1, 0x1, 0x1},
  {50, 100, 8479,   8480, 0x0, 0x0, 0x1, 0x1},
  {50, 100, 5479,   5480, 0x0, 0x0, 0x0, 0x0},
  {50, 275, 71000, 40000, 0x1, 0x0, 0x0, 0x0}, /* this is the center */
  {50, 275, 67224, 49475, 0x1, 0x0, 0x0, 0x0},
  {50, 275, 65756, 50943, 0x1, 0x0, 0x0, 0x0},
  {50, 300, 63756, 52943, 0x1, 0x1, 0x0, 0x1},
  {50, 300, 59868, 57831, 0x1, 0x1, 0x1, 0x1},
  {50, 300, 49577, 67123, 0x1, 0x0, 0x0, 0x0},
  {50, 200, 45577, 71123, 0x1, 0x0, 0x0, 0x0},
  {50, 200, 36000, 80000, 0x0, 0x0, 0x0, 0x0},
  {50, 200, 73224, 73624, 0x0, 0x0, 0x0, 0x0},
  {50, 200, 50577, 50977, 0x0, 0x1, 0x0, 0x1},
  {50, 200, 41577, 41977, 0x0, 0x0, 0x0, 0x0},
  {50, 200, 36177, 47377, 0x0, 0x0, 0x0, 0x0}, /* sides of the square */
  {50, 200, 62824, 74024, 0x0, 0x1, 0x0, 0x0},
  {50, 200, 63024, 75824, 0x0, 0x0, 0x0, 0x0},
  {50, 200, 73224, 73624, 0x0, 0x0, 0x0, 0x0},
  {50, 200, 73624, 63224, 0x0, 0x0, 0x0, 0x0},
  {50, 200, 46977, 36577, 0x0, 0x0, 0x0, 0x1},
  {50, 200, 45177, 34777, 0x0, 0x0, 0x0, 0x0},
  {50, 200, 80000, 44000, 0x1, 0x0, 0x0, 0x0},
  {50, 200, 71000, 40000, 0x1, 0x0, 0x0, 0x0},
  {50, 200, 55356, 50543, 0x1, 0x0, 0x0, 0x1},
  {50, 200, 44177, 61723, 0x1, 0x0, 0x0, 0x0},
  {50, 200, 40177, 65723, 0x1, 0x0, 0x0, 0x0},
  {50, 200, 50977, 76523, 0x1, 0x0, 0x0, 0x0},
/*  {50, 200, 72156, 55343, 0x1, 0x1, 0x0, 0x0}, */
  {50, 200, 73956, 53543, 0x1, 0x1, 0x0, 0x0},
  {50, 200, 76156, 51343, 0x1, 0x0, 0x0, 0x0},
```

```
   {50, 100, 60672, 59315, 0x0, 0x0, 0x0, 0x0},
   {00,  00,    00,    00, 0x0, 0x0, 0x0, 0x0}};

struct PlayList premark[] = {
  {50, 250, 0,     0,     0x0, 0x0, 0x0, 0x0},
  {50, 300, 37361, 33920, 0x0, 0x0, 0x0, 0x0},
  {50, 300, 56361, 52920, 0x0, 0x1, 0x1, 0x1},
  {50, 300, 57361, 53920, 0x0, 0x0, 0x1, 0x1},
  {50, 300, 60061, 56620, 0x0, 0x0, 0x0, 0x0},
  {50, 300, 65061, 61620, 0x0, 0x0, 0x0, 0x0},
  {50, 300, 47361, 43920, 0x0, 0x0, 0x0, 0x0},   /* this one */
  {50, 300, 12926, 9420,  0x0, 0x0, 0x0, 0x0},
  {00, 00,  00,    00,    0x0, 0x0, 0x0, 0x0}};

char StatusWords[49];
long int ActPos[4], WantedActPos[4], increment;
int manual, jog, TotalPoints;
int acceleration, velocity;
char Input[101], Output[101], InCount, OutCount;
char AxesPattern, Direction;
char Solnd0Pattern, Solnd1Pattern;

main()

{
  char previous;
  int index;
  int lk_kxget();
  int F1key, F2key, F3key, F4key;
  int servos;

  Initialize();
  lk_printf("\x1b1");
  lk_printf("\x1be");

  F1key = F2key = F3key = F4key = OFF;
  jog = servos = OFF;
  WantedActPos[0] = WantedActPos[1] = increment = (long)0;
  acceleration = 475;
  velocity = 55;
  manual = ON;
  Solnd0Pattern = (char) 0;
  Solnd1Pattern = (char) 0;

  index = 0;
  while (index != 4)
    {
      index = lk_kxget(0);
      if (index != -1) lk_setbeep(300);
                                   /* the switch pattern is as below:
                                      +---+---+---+---+---+---+
                                      + 4 + 8 + 12+ 16+ 20+ 24+
                                      +---+---+---+---+---+---+
                                      + 3 + 7 + 11+ 15+ 19+ 23+
```

```
                          +---+---+---+---+---+---+
                          + 2 + 6 + 10+ 14+ 18+ 22+
                          +---+---+---+---+---+---+
                          + 1 + 5 + 9 + 13+ 17+ 21+
                          +---+---+---+---+---+---+
                         */
    switch (index) {
      case 0:
       break;
      case 1:  /* pumps on */
       Solnd1Pattern = Solnd1Pattern ^ (char)(1 << 1); /* pump 1 */
       Solnd1Pattern = Solnd1Pattern ^ (char)(1 << 2); /* pump 2 */
       outport (0x42, (char) Solnd1Pattern);
       break;
      case 2: /* end effector rotate */
       if (servos == ON)
         {
          Solnd1Pattern = Solnd1Pattern ^ (char)(1 << 0);
          outport (0x42, (char) Solnd1Pattern);
         }
       else  /* premark extend */
         {
          Solnd0Pattern = Solnd0Pattern ^ (char)(1 <<7);
          outport (0x40, (char) Solnd0Pattern);
         }
       break;
      case 3:  /* home the robot */
       Home();
       break;
      case 4:
       servo(OFF); /* emergency stop */
       outport (0x41, (char)0xf); /* turn off the drivers */
       outport (0x40, (char)0x1); /* disable the driver ports */
       outport (0x43, (char)0xf);
       outport (0x42, (char)(1<<7));
       lk_printf("\x1b1");
       lk_printf("\x1be");
       lk_printf("Emergency Stop\n");
       lk_printf("Emergency Stop\n");
       lk_printf("Correct error Cycle\n");
       lk_printf("power to restart");
       exit(1);
       break;
      case 5: /* mixers on */
       Solnd0Pattern = Solnd0Pattern ^ (char)(1 << 5);
       Solnd0Pattern = Solnd0Pattern ^ (char)(1 << 6);
       outport (0x40, (char) Solnd0Pattern);
       break;
      case 6:
/*        Status(); */
       if (servos == ON)
         { /* gantry down */
          Solnd1Pattern = Solnd1Pattern ^ (char) (1 << 4);
          outport (0x42, (char)Solnd1Pattern);
```

```
    }
   else
    { /* premark down */
     Solnd1Pattern = Solnd1Pattern ^ (char)(1 << 3);
     outport (0x42, (char)Solnd1Pattern);
    }
   break;
  case 7:
   if (manual == 1)
    {
     lk_printf("\x1bp000");
     lk_printf("\x1be");
     lk_printf("Automatic Mode \n");
     lk_printf("  - F1 X-mark\n");
     lk_printf("  - F2 gantry test\n");
     lk_printf("  - F3 premark");
     manual = OFF;
   }
   else
   {
     lk_printf("\x1bp000");
     lk_printf("\x1bc");
     lk_printf("Manual Mode \n");
     manual = ON;
   }
   break;
  case 8:
   if (manual == ON)
    {
     if (jog == OFF)
      {
       jog = ON;
       lk_printf("\x1b1\x1be");
       lk_printf("Jog mode");
      }
     else
      {
       jog = OFF;
       lk_printf("\x1b1\x1be");
      }
    }
   break;
  case 9:
   if (servos == 1)
    {
     servo(OFF);
     lk_printf("\x1b1");
     lk_printf("\x1be");
     lk_printf("Servos disabled\n");
     servos = OFF;
    }
   else
    {
     servo(ON);
```

```
          lk_printf("\x1b1");
          lk_printf("\x1be");
          lk_printf("Servos enabled\n");
          servos = ON;
        }
      break;
    case 10:
/*        lk_printf("\x1b1");   */ /* clear display */
/*        lk_printf("\x1be"); */
      if (servos == ON)
        {
          Solnd1Pattern = Solnd1Pattern ^ (char) (1<<5);
          outport (0x42, (char)Solnd1Pattern);
        }
      break;
    case 11:
      break;
    case 12:
      if (manual)
        {
          if (F1key == OFF)
            {
              F1key = ON;
              F2key = OFF;
              lk_printf("\x1bp000\x1bc");
              lk_printf("Set accl on");
            }
          else
            {
              F1key = OFF;
              lk_printf("\x1bp000\x1bc");
              SetAccel();
            }
        }
      else
        {
          LoadMagicMode(Xmark);
          AutoMagicMode(Xmark);
        }
      break;
    case 13:
      if (manual)
        {
          if (jog)
            {
              Direction = '1';
              AxesPattern = '2';
              JogGant();
              Direction = '0';
              AxesPattern = '1';
              JogGant();
            }
          else
            {
```

```
                WantedActPos[1] = ActPos[1] + increment;
                WantedActPos[0] = ActPos[0] - increment;
                if (WantedActPos[0] < (long)0)  WantedActPos[0] = (long)0;
                MoveGant();
              }
          }
        break;
      case 14:
        if (manual)
          {
            if (jog)
              {
                Direction = '1';
                AxesPattern = '2';
                JogGant();
              }
            else
              {
                WantedActPos[1] = ActPos[1] + increment;
                WantedActPos[0] = ActPos[0];
                MoveGant();
              }
          }
        break;
      case 15:
        if (manual)
          {
            if (jog)
              {
                Direction = '1';
                AxesPattern = '2';
                JogGant();
                Direction = '1';
                AxesPattern = '1';
                JogGant();
              }
            else
              {
                WantedActPos[1] = ActPos[1] + increment;
                WantedActPos[0] = ActPos[0] + increment;
                MoveGant();
              }
          }
        break;
      case 16:
        if (manual)
          {
            if (F2key == OFF)
              {
                F2key = ON;
                if (F1key)
                  SetAccel();
                F1key = OFF;
                lk_printf("\x1bp000\x1bc");
```

```
        lk_printf("Set velo on");
      }
     else
      {
        F2key = OFF;
        lk_printf("\x1bp000\x1bc");
      }
    }
   else
    {
     LoadMagicMode(premark);
     AutoMagicMode(premark);
    }
   break;
  case 17:
   if (manual)
    {
     if (jog)
      {
        Direction = '0';
        AxesPattern = '1';
        JogGant();
      }
     else
      {
        WantedActPos[0] = ActPos[0] - increment;
        if (WantedActPos[0] < (long)0)  WantedActPos[0] = (long)0;
        WantedActPos[1] = ActPos[1];
        MoveGant();
      }
    }
   break;
  case 18:
   StopJogGant();
   break;
  case 19:
   if (manual)
    {
     if (jog)
      {
        Direction = '1';
        AxesPattern = '1';
        JogGant();
      }
     else
      {
        WantedActPos[0] = ActPos[0] + increment;
        WantedActPos[1] = ActPos[1];
        MoveGant();
      }
    }
   break;
  case 20:
   if (manual)
```

```
        {
         if (F1key)
           acceleration += 10;
         else
           if (F2key)
             velocity += 5;
           else
             increment += (long)1000;
        }
      else
        {
         MakePremark();
        }
      break;
    case 21:
      if (manual)
        {
         if (jog)
           {
             Direction = '0';
             AxesPattern = '2';
             JogGant();
             Direction = '0';
             AxesPattern = '1';
             JogGant();
           }
         else
           {
             WantedActPos[1] = ActPos[1] - increment;
             if (WantedActPos[1] < (long)0)  WantedActPos[1] = (long)0;
             WantedActPos[0] = ActPos[0] - increment;
             if (WantedActPos[0] < (long)0)  WantedActPos[0] = (long)0;
             MoveGant();
           }
        }
      break;
    case 22:
      if (manual)
        {
         if (jog)
           {
             Direction = '0';
             AxesPattern = '2';
             JogGant();
           }
         else
           {
             WantedActPos[1] = ActPos[1] - increment;
             WantedActPos[0] = ActPos[0];
             if (WantedActPos[1] < (long)0)  WantedActPos[1] = (long)0;
             MoveGant();
           }
        }
      break;
```

```
            case 23:
             if (manual)
               {
                if (jog)
                  {
                    Direction = '0';
                    AxesPattern = '2';
                    JogGant();
                    Direction = '1';
                    AxesPattern = '1';
                    JogGant();
                  }
                else
                  {
                    WantedActPos[1] = ActPos[1] - increment;
                    if (WantedActPos[1] < (long)0)  WantedActPos[1] = (long)0;
                    WantedActPos[0] = ActPos[0] + increment;
                    MoveGant();
                  }
               }
             break;
            case 24:
             if (manual)
               {
                if (F1key)
                  {
                    acceleration -= 10;
                    if (acceleration < 0) acceleration = 0;
                  }
                else
                  if (F2key)
                    {
                     velocity -= 5;
                     if (velocity < 0) velocity = 0;
                    }
                  else
                    {
                     increment -= (long) 1000;
                     if (increment < 0) increment = 0;
                    }
               }
          }
        if (manual == ON)
          {
            ActPosition();
            DispPosition();
          }
        index = -1;
      }
    }

/*=================================================================
PROCEDURE
MakePremark
```

```
=================================================================*/

MakePremark()

{
 lk_printf("\x1be");
 lk_printf("Premark in progress \n");
 Solnd1Pattern = Solnd1Pattern | (1<<3);
 Solnd0Pattern = Solnd0Pattern | (1<<4);
 outport (0x40, (char)Solnd0Pattern);
 outport (0x42, (char)Solnd1Pattern);
 lk_printf("Guns on, frame down\n");
 lk_tdelay(10000);
 Solnd0Pattern = Solnd0Pattern | (1<<7);
 outport (0x40, (char)Solnd0Pattern);
 lk_printf("Guns in motion");
 lk_tdelay(10000);
 Solnd1Pattern = Solnd1Pattern ^ (1<<3);
 Solnd0Pattern = Solnd0Pattern ^ (1<<4);
 Solnd0Pattern = Solnd0Pattern ^ (1<<7);
 outport(0x40, (char)Solnd0Pattern);
 outport(0x42, (char)Solnd1Pattern);
}

/*=================================================================
PROCEDURE
 LoadMagicMode

SYNPOSIS
 this procedure loads the point data into the iai robot controller.
 basically, you need to pass a data structure of the points you want
 to move to. end the list with 0 acceleration

USES
 WhichOne - pointer to the data structure
=================================================================*/
LoadMagicMode(WhichOne)

struct PlayList WhichOne[];

{
 int index, index1, offset;
 char value[10];

 index = 0;

 while (WhichOne[index].AccelWanted != 0)
   {
    ClearBuf(Output, 40);
    ClearBuf(Input, 40);
    strcpy(Output, Commands[14].Cmmnd);   /* load point command formation */

    ClearBuf(value, 6);
```

```
        itoa((index+1), value);             /* fill in point offset */
        index1 = itoan((index+1))-1;
        for (offset = 0; index1 >= 0; index1--, offset++)
          Output[8-offset] = value[index1];

        ClearBuf(value, 6);
        ltoa(WhichOne[index].Actuator1Pos, value); /* fill in actuator 1 part */
        index1 = ltoan(WhichOne[index].Actuator1Pos)-1;
        for (offset = 0; index1 >= 0; index1--, offset++)
          Output[15-offset] = value[index1];

        ClearBuf(value, 6);
        ltoa(WhichOne[index].Actuator2Pos, value); /* fill in actuator 2 part */
        index1 = ltoan(WhichOne[index].Actuator2Pos)-1;
        for (offset = 0; index1 >= 0; index1--, offset++)
          Output[22-offset] = value[index1];

        InCount = (char) Commands[14].RespLength;
        ser_rec_s0(Input, &InCount);
        OutCount = (char) Commands[14].ComLength;
        ser_send_s0(Output, &OutCount);
/*        printf("load mode: %s", Output); */
        while (OutCount != (char) 0); /* wait for transmission to finish */
        while (InCount != (char) 0);  /* wait for reception to finish */

        if(Input[0] == '%')
          {
            lk_printf("\x1be");
            lk_printf("Set Points:\n");
            lk_printf(" Transmission error\n");
            lk_printf(" %s", Input);
            printf("Set Points: Transmission error %s", Input);
            servo(OFF);
            outport (0x41, (char)0xf); /* turn off the drivers */
            outport (0x40, (char)0x1); /* disable the driver ports */
            outport (0x43, (char)0xf);
            outport (0x42, (char)(1<<7));
            exit(1);
          }

        Output[0] = '/';           /* Convert to command sequence */
        Output[5] = '@';             /* and execute the command */
        Output[6] = '@';
        Output[7] = '\r';
        Output[8] = '\n';
        OutCount = (char) 9;

        ser_send_s0(Output, &OutCount);
        while (OutCount != (char) 0); /* wait for transmission to finish */
        index++;
      }
    TotalPoints = index;
  }
```

```
/*============================================================================
PROCEDURE
 AutoMagicMode

SYNPOSIS
 this procedure executes the point data load into the iai robot
 controller previously by the LoadMagicMode procedure. you must,
 however,  pass a data structure of the points you want
 to move to in order to properly position the opening/closing of the
 guns and the rotation of the end effector. as always you must
 end the list with 0 acceleration

USES
 WhichOne - pointer to the data structure
============================================================================*/

AutoMagicMode(WhichOne)

struct PlayList WhichOne[];

{
 int index, index1, offset, lk_kxget(), itoan();
 long int labs();
 char Old1SolndPattern, value[8];

 index = 0;

 ClearBuf(Output, 40);
 ClearBuf(Input, 40);
 strcpy(Output, Commands[13].Cmmnd);  /* execute point command formation */

 ClearBuf(value, 6);
 itoa(WhichOne[0].VeloWanted, value); /* fill in velocity part */
 index1 = itoan(WhichOne[0].VeloWanted)-1;
 for (offset = 0; index1 >= 0; index1--, offset++)
  Output[8-offset] = value[index1];
 ClearBuf(value, 6);
 itoa(TotalPoints, value);          /* fill in point limits */
 index1 = itoan(TotalPoints)-1;
 for (offset = 0; index1 >= 0; index1--, offset++)
  Output[16-offset] = value[index1];

 InCount = (char) Commands[13].RespLength;
 ser_rec_s0(Input, &InCount);
 OutCount = (char) Commands[13].ComLength;
 ser_send_s0(Output, &OutCount);
 while (OutCount != (char) 0); /* wait for transmission to finish */
 while (InCount != (char) 0);  /* wait for reception to finish */

 if(Input[0] == '%')
  {
   lk_printf("\x1be");
   lk_printf("Run Points:\n");
   lk_printf(" Transmission error\n");
```

```
    lk_printf(" %s", Input);
    printf("Run Points: Transmission error %s", Input);
    servo(OFF);
    outport (0x41, (char)0xf); /* turn off the drivers */
    outport (0x40, (char)0x1); /* disable the driver ports */
    outport (0x43, (char)0xf);
    outport (0x42, (char)(1<<7));
    exit(1);
   }

  Output[0] = '/';          /* Convert to command sequence */
  Output[5] = '@';            /* and execute the command */
  Output[6] = '@';
  Output[7] = '\r';
  Output[8] = '\n';
  OutCount = (char) 9;

  ser_send_s0(Output, &OutCount);
  while (OutCount != (char) 0);  /* wait for transmission to finish */

  Old1SolndPattern = Solnd1Pattern;

  while (WhichOne[index].AccelWanted != 0)
   {
    WantedActPos[0] = WhichOne[index].Actuator1Pos;
    WantedActPos[1] = WhichOne[index].Actuator2Pos;
    lk_printf("\x1bp000\x1bc");
    lk_printf("Move # %d....", index);

    Solnd1Pattern = (Old1SolndPattern | WhichOne[index].EEPosition);
    Solnd0Pattern = ((WhichOne[index].gun1 << 1) |
                (WhichOne[index].gun2 << 2) |
                (WhichOne[index].gun3 << 3));

    outport (0x40, (char)Solnd0Pattern);
    outport (0x42, (char)Solnd1Pattern);

                        /*
                          now loop and look for the desired
                          coordinate to fall within a square
                          defined by +/- 5 mm on all sides of
                          the desired setpoint. better look
                          intently, otherwise you might miss
                          the set point
                        */
/*
    while (!((WantedActPos[0] < (ActPos[0] + (long)900)) &&
         (WantedActPos[0] > (ActPos[0] - (long)900))) ||
         !( (WantedActPos[1] < (ActPos[1] + (long)900)) &&
         (WantedActPos[1] > (ActPos[1] - (long)900))))
*/
    while ( ((labs(WantedActPos[0] - ActPos[0])) >= (long)1000) &&
          ((labs(WantedActPos[1] - ActPos[1])) >= (long)1000))
     {
```

```
        ActPosition();
/*        DispPosition();  */

        if (lk_kxget(0) == 4)
          {
           servo(OFF);

           OutCount = (char) Commands[10].ComLength;
           ser_send_s0(Output, &OutCount);
           while (OutCount != (char) 0);  /* wait for transmission to finish */

           outport (0x41, (char)0xf); /* turn off the drivers */
           outport (0x40, (char)0x1); /* disable the driver ports */
           outport (0x43, (char)0xf);
           outport (0x42, (char)(1<<7));

           lk_printf("\x1b1");
           lk_printf("\x1be");
           lk_printf("Emergency Stop\n");
           lk_printf("Emergency Stop\n");
           lk_printf("Correct error Cycle\n");
           lk_printf("power to restart");
           exit(1);
          }
      }                     /* turn off the guns during the pauses
                               between the points */

    lk_printf("\x1bp000\x1bc");
    index++;
   }
  manual = ON;
  outport (0x40, (char)Solnd0Pattern);
  outport (0x42, (char)Old1SolndPattern);
  Solnd1Pattern = Old1SolndPattern;
  lk_printf("\x1bp000\x1bc");
}

/*=====================================================================
PROCEDURE
  StopJogGant

USES
  nothing as far as I can tell

SYNPOSIS
  this procedure stops all axis of the gantry during the jog command
=====================================================================*/
StopJogGant()

{
  ClearBuf(Input, 50);
  ClearBuf(Output, 50);

  strcpy(Output, Commands[12].Cmmnd);    /* servo command formation */
```

```
  Output[5] = '3';

  InCount = (char) Commands[12].RespLength;
  ser_rec_s0(Input, &InCount);
  OutCount = (char) Commands[12].ComLength;
  ser_send_s0(Output, &OutCount);
  while (OutCount != (char) 0);  /* wait for transmission to finish */
  while (InCount != (char) 0);   /* wait for reception to finish */

  if(Input[0] == '%')
   {
    lk_printf("\x1be");
    lk_printf("Stop Motion:\n");
    lk_printf("Transmission error\n");
    lk_printf(" %s", Input);
    printf("Stop Motion:Transmission error %s", Input);
    servo(OFF);
    outport (0x41, (char)0xf); /* turn off the drivers */
    outport (0x40, (char)0x1); /* disable the driver ports */
    outport (0x43, (char)0xf);
    outport (0x42, (char)(1<<7));
    exit(1);
   }

  Output[0] = '/';           /* Convert to command sequence */
  Output[5] = '@';             /* and execute the command */
  Output[6] = '@';
  Output[7] = '\r';
  Output[8] = '\n';
  OutCount = (char) 9;

  ser_send_s0(Output, &OutCount);
  while (OutCount != (char) 0);  /* wait for transmission to finish */
}

/*=========================================================================
PROCEDURE
  JogGant

USES
  nothing as far as I can tell

SYNPOSIS
  this procedure jogs the specified axises of the gantry
=========================================================================*/
JogGant()

{
  char value[8];
  int index, offset;
  int itoan();

  ClearBuf(Input, 50);
  ClearBuf(Output, 50);
```

```
 offset = 0;

 strcpy(Output, Commands[6].Cmmnd);    /* get linear motion command */

 Output[5] = AxesPattern;
 Output[6] = Direction;

 itoa(velocity, value);              /* fill in velocity part */
 index = itoan(velocity)-1;
 for (offset = 0; index >= 0; index--, offset++)
   Output[9-offset] = value[index];

 InCount = (char) Commands[6].RespLength;
 ser_rec_s0(Input, &InCount);
 OutCount = (char) Commands[6].ComLength;
 ser_send_s0(Output, &OutCount);
 while (OutCount != (char) 0);  /* wait for transmission to finish */
 while (InCount != (char) 0);   /* wait for reception to finish */

 if(Input[0] == '%')
   {
     lk_printf("\x1be");
     lk_printf("Jog:\n");
     lk_printf(" Transmission error\n");
     lk_printf(" %s", Input);
     printf("Jog: Transmission error %s", Input);
     servo(OFF);
     outport (0x41, (char)0xf); /* turn off the drivers */
     outport (0x40, (char)0x1); /* disable the driver ports */
     outport (0x43, (char)0xf);
     outport (0x42, (char)(1<<7));
     exit(1);
   }

 Output[0] = '/';          /* Convert to command sequence */
 Output[5] = '@';           /* and execute the command */
 Output[6] = '@';
 Output[7] = '\r';
 Output[8] = '\n';
 OutCount = (char) 9;

 ser_send_s0(Output, &OutCount);
 while (OutCount != (char) 0);  /* wait for transmission to finish */
}

/*=========================================================================
PROCEDURE
 SetAccel

USES
 this procedure uses the global variable acceleration to get the
  user's desired acceleration

SYNPOSIS
```

```
this procedure sets the acceleration of the gantry according to the
user's wishes.
===========================================================================*/
SetAccel()

{
 char value[5];
 int index, offset;
 int itoan();

 offset = 0;
 if (acceleration == 0) return(0);
 itoa(acceleration, value);
 index = itoan(acceleration) - 1;

 ClearBuf(Input, 50);
 ClearBuf(Output, 50);

 strcpy(Output, Commands[7].Cmmnd);     /* fill in accleration command */
 for (offset = 0; index >= 0; index--, offset++)
  {
   Output[8-offset] = value[index];
  }

 InCount = (char) Commands[7].RespLength;
 ser_rec_s0(Input, &InCount);
 OutCount = (char) Commands[7].ComLength;
 ser_send_s0(Output, &OutCount);
 while (OutCount != (char) 0);  /* wait for transmission to finish */
 while (InCount != (char) 0);   /* wait for reception to finish */

 if(Input[0] == '%')
  {
   lk_printf("\x1be");
   lk_printf("Acceleration:\n");
   lk_printf(" Transmission error\n");
   lk_printf(" %s", Input);
   printf("Acceleration: Transmission error %s", Input);
   servo(OFF);
   outport (0x41, (char)0xf); /* turn off the drivers */
   outport (0x40, (char)0x1); /* disable the driver ports */
   outport (0x43, (char)0xf);
   outport (0x42, (char)(1<<7));
   exit(1);
  }

 Output[0] = '/';           /* Convert to command sequence */
 Output[5] = '@';            /* and execute the command */
 Output[6] = '@';
 Output[7] = '\r';
 Output[8] = '\n';
 OutCount = (char) 9;

 ser_send_s0(Output, &OutCount);
```

```
 while (OutCount != (char) 0);  /* wait for transmission to finish */
}

/*=====================================================================
PROCEDURE
 MoveGant

USES
 velocity - the desired velocity of the movement
 WantedActPos - array of wanted actuator positions

SYNPOSIS
 this procedure moves the gantry to the specified position.
 sets the acceleration of the gantry according to the
 user's wishes.
=====================================================================*/
MoveGant()

{
 char value[8];
 int index, offset;
 int itoan();

 if (velocity == 0) return(0);

 ClearBuf(Input, 50);
 ClearBuf(Output, 50);
 offset = 0;

 strcpy(Output, Commands[4].Cmmnd);     /* get linear motion command */

 itoa(velocity, value);              /* fill in velocity part */
 index = itoan(velocity)-1;
 for (offset = 0; index >= 0; index--, offset++)
  Output[8-offset] = value[index];

 ltoa(WantedActPos[0], value);          /* fill in 1st act. position */
 index = ltoan(WantedActPos[0]) - 1;

 for (offset = 0; index >= 0; index--, offset++)
  Output[14-offset] = value[index];

 ltoa(WantedActPos[1], value);          /* fill in 2nd act. position */
 index = ltoan(WantedActPos[1]) - 1;
 for (offset = 0; index >= 0; index--, offset++)
  Output[20-offset] = value[index];

 InCount = (char) Commands[4].RespLength;
 ser_rec_s0(Input, &InCount);
 OutCount = (char) Commands[4].ComLength;
 ser_send_s0(Output, &OutCount);
 while (OutCount != (char) 0);  /* wait for transmission to finish */
 while (InCount != (char) 0);  /* wait for reception to finish */
```

```
  if(Input[0] == '%')
   {
     lk_printf("\x1be");
     lk_printf("Linear Motion:\n");
     lk_printf(" Transmission error\n");
     lk_printf(" %s", Input);
     printf("Linear motion: Transmission error %s", Input);
     servo(OFF);
     outport (0x41, (char)0xf); /* turn off the drivers */
     outport (0x40, (char)0x1); /* disable the driver ports */
     outport (0x43, (char)0xf);
     outport (0x42, (char)(1<<7));
     exit(1);
   }

  Output[0] = '/';          /* Convert to command sequence */
  Output[5] = '@';           /* and execute the command */
  Output[6] = '@';
  Output[7] = '\r';
  Output[8] = '\n';
  OutCount = (char) 9;

  ser_send_s0(Output, &OutCount);
  while (OutCount != (char) 0);  /* wait for transmission to finish */
}

/*==========================================================================
PROCEDURE
  servo

USES
  OnOff - the value whether on or off.

SYNPOSIS
  this procedure turns the servos on or off.
==========================================================================*/
servo(OnOff)

int OnOff;

{
  ClearBuf(Input, 100);
  ClearBuf(Output, 100);

  strcpy(Output, Commands[2].Cmmnd);     /* servo command formation */
  Output[5] = '7';

  if (OnOff == 1)
    Output[6] = '1';              /* servo on command */
  else
    Output[6] = '0';              /* servo off command */

  InCount = (char) Commands[2].RespLength;
  ser_rec_s0(Input, &InCount);
```

```
  OutCount = (char) Commands[2].ComLength;
  ser_send_s0(Output, &OutCount);
  while (OutCount != (char) 0);  /* wait for transmission to finish */
  while (InCount != (char) 0);   /* wait for reception to finish */

  if(Input[0] == '%')
   {
    lk_printf("\x1be");
    lk_printf("Servo:\n");
    lk_printf(" Transmission error\n");
    lk_printf(" %s", Input);
    printf("Servo: Transmission error %s", Input);
    outport (0x41, (char)0xf); /* turn off the drivers */
    outport (0x40, (char)0x1); /* disable the driver ports */
    outport (0x43, (char)0xf);
    outport (0x42, (char)(1<<7));
    exit(1);
   }

  Output[0] = '/';          /* Convert to command sequence */
  Output[5] = '@';           /* and execute the command */
  Output[6] = '@';
  Output[7] = '\r';
  Output[8] = '\n';
  OutCount = (char) 9;

  ser_send_s0(Output, &OutCount);
  while (OutCount != (char) 0);  /* wait for transmission to finish */
}

/*==========================================================================
PROCEDURE
  DispPosition

USES
  StatusWords - the status word received from the gantry robot

SYNPOSIS
  this procedure gets the status word and interprets it. after it
  finishes interpreting it, it then displays the results on the LCD
  display.
==========================================================================*/

DispPosition()

{
  char number[7];
  int index, index1;
  int atoi();

/*
  printf("Status Word: %s", StatusWords);
  printf("Sup: %c Mde: %c Prg: %c Crd: %c Wrt: %c Axs: %c \n",
        StatusWords[5], StatusWords[6], StatusWords[7], StatusWords[8],
```

```
        StatusWords[9], StatusWords[10]);
    printf("Status: %c %c %c %c \n", StatusWords[14], StatusWords[15],
        StatusWords[16], StatusWords[17]);


    for (index = 0; index < 2; index++)
     {
      for (index1 = 0; index1 < 6; index1++)
        number[index1] = StatusWords[20+index1+(index*6)];
      number[index1] = '\0';
      ActPos[index] = atol(number);
     }

    printf("1:%d 2:%d 3:%d 4:%d", ActPos[0], ActPos[1],
        ActPos[2], ActPos[3]);
    printf("Mode: %d acc:%d vel:%d\n", manual, acceleration, velocity);
   */

    lk_printf("\x1b0");
    lk_printf("\x1bp100");
    lk_printf("\1bc");
    lk_printf("\x1bp100");
    lk_printf("Stat: %c %c ", StatusWords[14], StatusWords[15]);

    if (Solnd1Pattern & ((1<<1) || (1<<2)))
     lk_printf(":PP");
    else
     lk_printf(":pp");

    if (Solnd0Pattern & ((1<<5) || (1<<6)))
     lk_printf("MM");
    else
     lk_printf("mm");

    if (Solnd1Pattern & 1)
     lk_printf("R");
    else
     lk_printf("r");

    if (Solnd0Pattern & (1<<1))
     lk_printf("B");
    else
     lk_printf("b");

    if (Solnd0Pattern & (1<<2))
     lk_printf("W");
    else
     lk_printf("w");

    if (Solnd0Pattern & (1<<3))
     lk_printf("B");
    else
     lk_printf("b");
    lk_printf("\x1bp200");
```

```
 lk_printf("\1bc");
 lk_printf("\x1bp200");
 lk_printf("M:");
 if (manual)
   {
     if (jog)
       lk_printf("J ");
     else
       lk_printf("M ");
   }
 else
   lk_printf("A ");
 lk_printf("A:%3.0dV:%3.0dI:%4.0ld\n", acceleration, velocity, increment);
 lk_printf("\x1bp300");
 lk_printf("\x1bc");
 lk_printf("\x1bp300");
 lk_printf("1:%6.0ld 2:%6.0ld ", ActPos[0], ActPos[1]);
 lk_printf("\x1b1");
}

/*=============================================================================
PROCEDURE
  ActPostion

USES
  StatusWords - the status word received from the gantry robot

SYNPOSIS
  this procedure gets the status word from the gantry controller.
=============================================================================*/
ActPosition()

{
 int index, index1;
 char number[9];
 long atol();

 ClearBuf(Input, 100);
 ClearBuf(Output, 100);

 InCount = (char) Commands[1].RespLength;  /* get status */
 ser_rec_s0(StatusWords, &InCount);
 OutCount = (char) Commands[1].ComLength;
 ser_send_s0(Commands[1].Cmmnd, &OutCount);
 while (OutCount != (char) 0);  /* wait for transmission to finish */
 while (InCount != (char) 0);   /* wait for reception to finish */

 for (index = 0; index < 2; index++)
   {
     for (index1 = 0; index1 < 6; index1++)
       number[index1] = StatusWords[20+index1+(index*6)];
     number[index1] = '\0';
     ActPos[index] = atol(number);
   }
```

```
}

/*==========================================================================
PROCEDURE
  Home

USES
  nothing as far as I can tell

SYNPOSIS
  this procedure homes the robot
==========================================================================*/
Home()

{
  lk_printf("\x1bp000\x1bc\x1bp000");
  lk_printf("homing... ");
  printf("Homing gantry... beginning\n");

  ClearBuf(Input, 100);
  ClearBuf(Output, 100);

  strcpy(Output, Commands[3].Cmmnd);     /* form home command */
  Output[5] = '7';
  InCount = (char) Commands[3].RespLength;
  ser_rec_s0(Input, &InCount);
  OutCount = (char) Commands[3].ComLength;
  ser_send_s0(Output, &OutCount);
  while (OutCount != (char) 0);  /* wait for transmission to finish */
  while (InCount != (char) 0);   /* wait for reception to finish */

  if(Input[0] == '%')
   {
    lk_printf("\x1be");
    lk_printf("Home:\n");
    lk_printf(" Transmission error\n");
    lk_printf(" %s", Input);
    printf("Home: Transmission error %s", Input);
    servo(OFF);
    outport (0x41, (char)0xf); /* turn off the drivers */
    outport (0x40, (char)0x1); /* disable the driver ports */
    outport (0x43, (char)0xf);
    outport (0x42, (char)(1<<7));
    exit(1);
   }

  Output[0] = '/';          /* Convert to command sequence */
  Output[5] = '@';            /* and execute the command */
  Output[6] = '@';
  Output[7] = '\r';
  Output[8] = '\n';
  OutCount = (char) 9;
  ser_send_s0(Output, &OutCount);
  while (OutCount != (char) 0);  /* wait for transmission to finish */
```

```
 ActPosition();
 while (!((StatusWords[14] == '7') && (StatusWords[15] == '7')
   && (StatusWords[16] == '7')))
   {
    DispPosition();
    ActPosition();
   }
 lk_printf("\x1bp000\x1bc\x1bp000");
 lk_printf("homing... done ");

 printf("Homing gantry... done\n");
}

/*==========================================================================
PROCEDURE
 ClearBuf

USES
 in - input string
 count - number of nulls to use

SYNPOSIS
 this procedure clears out the input string and makes it null.
==========================================================================*/
ClearBuf(in, count)

char in[];
int count;

{
 int index;

 for (index = 0; index < count; index++)
   in[index] = '\0';
}

Status()

{
 int index;
 int atoi();

 ClearBuf(Input, 50);
 ClearBuf(Output, 50);
 InCount = (char) Commands[0].RespLength;
 ser_rec_s0(Input, &InCount);
 OutCount = (char) Commands[0].ComLength;
 ser_send_s0(Commands[0].Cmmnd, &OutCount);
 while (OutCount != (char) 0);  /* wait for transmission to finish */
 while (InCount != (char) 0);   /* wait for reception to finish */

 lk_printf("\x1bp000\x1be");
 if (Input[5] == '0')
```

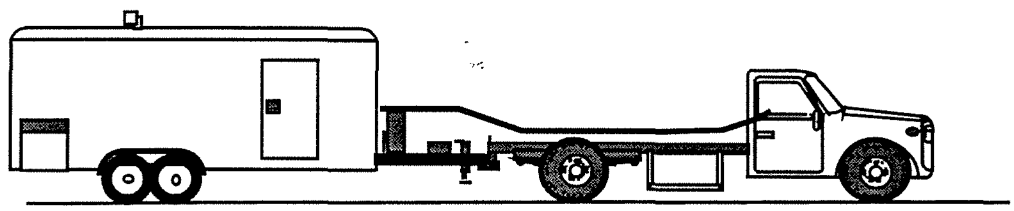
```
  lk_printf("With memory card \n");
else
  lk_printf("ROM - no memory card \n");

lk_printf("Startup: ");

switch (Input[6]) {
  case '0':
    lk_printf("Idle\n");
    break;
  case '1':
    lk_printf("Ext Strt\n");
    break;
  case '2':
    lk_printf("Teaching\n");
}

lk_printf("Md: ");
if (Input[7] == '1')
  lk_printf("R P");

switch (Input[8]) {
  case '0':
    lk_printf("No cd wrt on\n");
    break;
  case '1':
    lk_printf("Cd in wrt on\n");
    break;
  case '2':
    lk_printf("No cd wrt off\n");
    break;
  case '3':
    lk_printf("Cd in wrt off\n");
    break;
}

if (Input[9] == '1')
  printf("Writing to card\n");

lk_printf("Axes: %c \n", Input[10]);
for (index = 0; index <= atoi(Input+10); index++)
  {
    printf("Axis %d: ", index);
    switch (Input[14+index]) {
      case '0':
        printf("Servo off\n");
        break;
      case '1':
        printf("Servo on\n");
        break;
      case '3':
        printf("Initiating home\n");
        break;
      case '4':
```

```
        printf("Servo off, stop\n");
        break;
      case '5':
        printf("Servo on, stop\n");
        break;
      case '7':
        printf("Servo on, home, stop");
        break;
    }
  }
}

Initialize()

{
 int index;
 int lk_kxget();

 ser_init_s0((char)4, (char) 8);  /* initialize serial port according
                                     to parameters on page 53         */
 lk_kxinit();
 lk_init();
 lk_init_keypad();

 ClearBuf(Input, 98);
 ClearBuf(Output, 98);

 outport (0x41, (char)0xf);   /* pioa command */
 outport (0x40, (char)0x0);   /* pioa data */
 outport (0x43, (char)0xf);   /* piob command */
 outport (0x42, (char)0x0);   /* piob data */
/*
 InCount = (char) Commands[10].RespLength;
 ser_rec_s0(Input, &InCount);

 OutCount = (char) Commands[10].ComLength;
 ser_send_s0(Commands[10].Cmmnd, &OutCount);
*/
/*  while (OutCount != (char) 0);*/  /* wait for transmission to finish */

 servo(OFF);

/*  while (InCount != (char) 0);*/   /* wait for reception to finish */
/*  printf("Resetting Controller \n"); */

 lk_printf("\x1bp000\x1be");
 lk_printf("The PreMark Maker\n");
 lk_printf("  Version %0.2f\n", VERSION);
 lk_printf("  %2d/%2d/%2d \n", MONTH, DAY, YEAR);
 lk_printf("F1 to continue! ");
 while (lk_kxget(0) != 12) hitwd();
}
```

APPENDIX D: OPERATION MANUAL

The Premark Maker

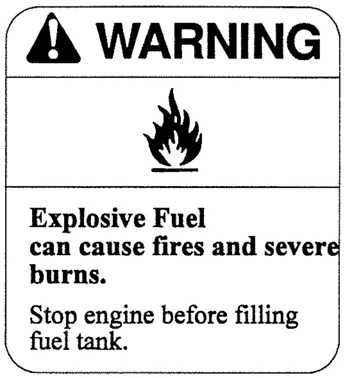
User's Manual
Preliminary Version
July 19, 1994



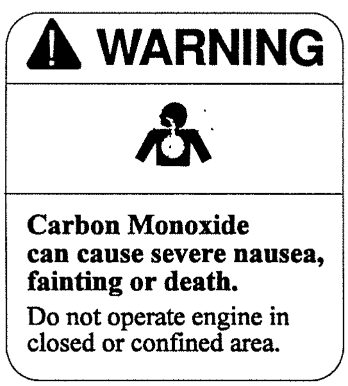
Advanced Highway Maintenance and Construction Technologies Center
Department of Mechanical Engineering
University of California, Davis
Davis, CA 95616
Technical Support: (916) 752-4180

READ CAREFULLY BEFORE ATTEMPTING TO ASSEMBLE, INSTALL, OPERATE OR MAINTAIN THE PRODUCT DESCRIBED. PROTECT YOURSELF AND OTHERS BY OBSERVING ALL SAFETY INFORMATION. FAILURE TO COMPLY WITH INSTRUCTIONS COULD RESULT IN PERSONAL INJURY AND/OR PROPERTY DAMAGE! RETAIN INSTRUCTIONS FOR FUTURE REFERENCE.

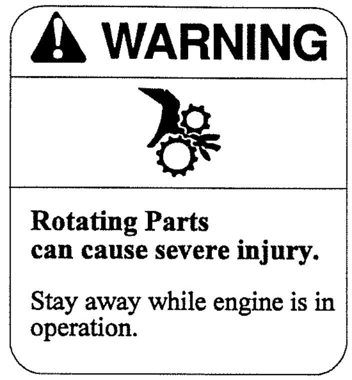
0. SAFETY INFORMATION



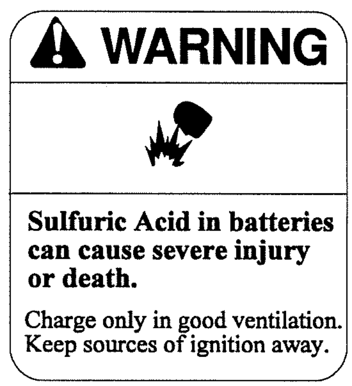
Explosive Fuel!
Gasoline is extremely flammable and its vapors can explode if ignited. Store gasoline only in approved containers, in well-ventilated, unoccupied buildings, away from sparks or flames. Do not fill the fuel tank while the engine is hot or running, since spilled fuel could ignite if it comes in contact with hot parts or sparks from ignition. Do not start the engine near spilled fuel; wipe up spills immediately. Never use gasoline as a cleaning agent.



Lethal Exhaust Gases!
Engine exhaust gases contain poisonous carbon monoxide. Carbon monoxide is odorless, colorless, and can cause death if inhaled. Avoid inhaling exhaust fumes, and never run the engine in a closed building or confined area.

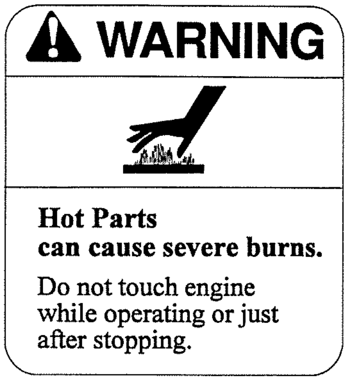


Rotating Parts!
Keep hands, feet, hair, and clothing away from all moving parts to prevent injury. Never operate the engine with covers, shrouds, or guards removed.



Dangerous Acid, Explosive Gases!
Batteries contain sulfuric acid. To prevent acid burns, avoid contact with skin, eyes, and clothing. Batteries produce explosive hydrogen gas while being charged. To prevent a fire or explosion, charge batteries only in well ventilated areas. Keep sparks, open flames, and other sources of ignition away from the battery at all times. Keep batteries out of the reach of children. Remove all jewelry when servicing batteries.

Before disconnecting the negative ground cable, make sure all switches are OFF. If ON, a spark will occur at the ground cable terminal which *could cause an explosion if hydrogen gas or gasoline vapors are present.*



Hot Parts!
The crankcase, cylinder head, exhaust system, and other components can get extremely hot from operation. To prevent severe burns, do not touch these areas while the engine is running or immediately after it is turned off. Never operate the engines with heat shields or guards removed.

Accidental Starts!
Before servicing the engine or equipment, always disconnect the spark plug lead to prevent the engine from starting accidentally. Ground the lead to prevent sparks that could cause fires.

On engines equipped with a 12-volt battery and/or electric start, disconnect the battery cables from the battery. Always disconnect the negative (-) cable first.

Before disconnecting the negative ground cable, make sure all switches are OFF. If ON, a spark will occur at the ground cable terminal which could cause an explosion if hydrogen gas or gasoline vapors are present.

High Voltage!
Never touch electrical wires or components while the engine is running. They can be sources of electrical shock which could cause severe injury or burns.

Overspeed is Hazardous!
Never tamper with the governor components or settings to increase the maximum speed. Severe personal injury and damage to the engine or equipment can result if operated at speed above maximum.

Engine Maintenance!
Do not attempt to crank or start engine until it has been properly serviced with recommended motor oil. Any attempt to crank or start the engine before it has been properly servicing with the recommended oil will result in engine failure.

Release Air pressure before Maintenance!
Before servicing the compressor package, make sure power source has been turned off and system air pressure has been released.

Do not Adjust Pressure Control Components!
Do not attempt to change the settings on control components. Pressure switch and pilot valve settings are preset at the factory for normal operating conditions. Altering the settings will result in compressor and motor damage.

1.0 System Overview and Description

1.1 Introduction

The UC Davis/CalTrans Premark Painting Trailer is an integrated system that can paint 2 different types of aerial surveying marks. The Trailer is self-contained with onboard power and painting systems. Figure 1.1 shows the basic layout of the Painting Trailer. All functions, except engine start and shutdown are accessible from the hand-held control unit.

1.2 Power Systems (Figure 1.2)

Electrical power is provided to the Trailer by a 4 kilowatt, 8HP Dayton generator. This generator provides electric service for the gantry robot and its controller. A 24 volt DC power transformer provides power for the air solenoids. Pneumatic power is provided by a 12HP 30 gallon Speedaire compressor. Pneumatic power is used for the paint pumps, as well as for all the actuators in the Trailer.

1.3 Control Systems

The entire system is controlled from the Hand-held Control Unit (Figure 1.3). This unit controls the gantry robot motion and activates all the actuators and solenoids necessary for operation. Interface boards are located on the top shelf of the Control cabinet (Figure 1.5-a). The gantry robot interface unit is also in the control cabinet on the third shelf (Figure 1.5-c). Manual controls to actuate all systems are located on a control panel on the second shelf of the control cabinet (Figure 1.5-b). Power supplies for all interface units are located on the bottom of the control cabinet (Figure 1.5-d).

1.3.1 Controller Software

The Hand Control Unit controls the Gantry in 3 different modes: 1) Automatic, 2) Incremental, and 3) Continuous.

In Automatic mode, the Gantry will paint a premark or a test mark. This mode is accessible by pressing the 'Auto Paint Sys' button to bring up the menu, and pressing 'F1' to activate the automatic premark painting mode or pressing 'F2' to activate the automatic test mark painting mode.

In Incremental mode, the gantry will move a preset distance in the direction chosen by the operator. To use this mode, choose gantry functions by pressing the 'Gant Funct/LFN Funct' button. Press the 'Jog' button to toggle between Continuous and Incremental mode. When the mode indicator (Figure 1.4) shows 'M:M', the unit is in Incremental mode. When the mode indicator shows "M:J", the unit is in Continuous mode. Pressing '+' or '-' increases or decreases the movement increment. Next, selection of the direction initiates the movement of the gantry. The orientation of the keypad is such that the keypad should be placed parallel to the ground, with arrow on the '2' key pointing to the back of the trailer.

In Continuous mode, the gantry will move in the direction chosen, until the 'Stop' key is pressed. The keypad orientation is as described above.

Pressing 'All Stop' will result in a software controlled emergency stop. If a more drastic emergency stop is required, the left red switch on the top of the hand unit should be pressed.

1.4 Paint Systems

Paint is pumped using 2 Binks pneumatic paint pumps. One each is provided to pump black and white paint. In addition, located on each paint can cover, are paint mixers to keep the paint well mixed. The mixers operation is controlled from the Hand Controller.

1.5 Premark Systems

A gantry robot is used to paint the large X-style premark (Figure 1.6). A custom designed Linear Free Nozzle system (Figure 1.7) is used to paint abbreviated premarks. All operations are fully automatic and are controlled from the Hand Controller.

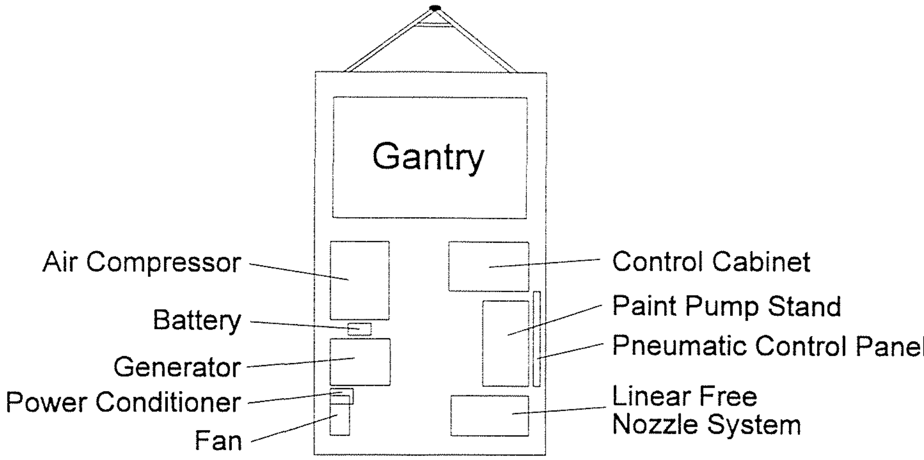


Figure 1.1: Trailer Layout

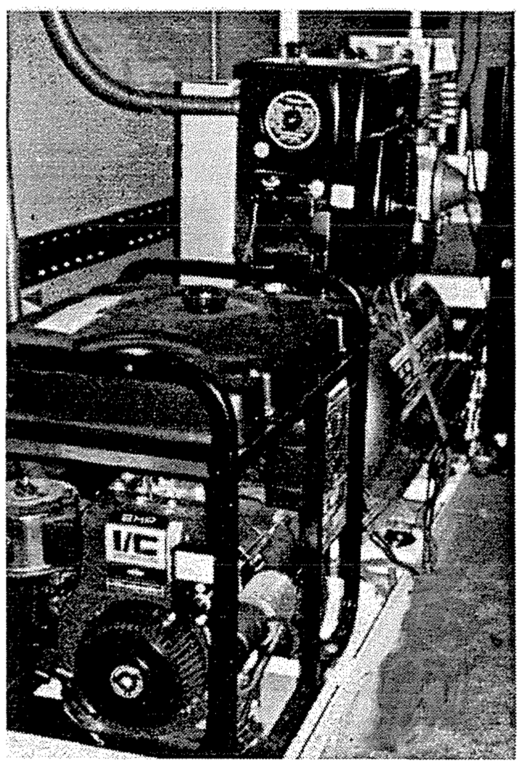


Figure 1.2: Power Systems

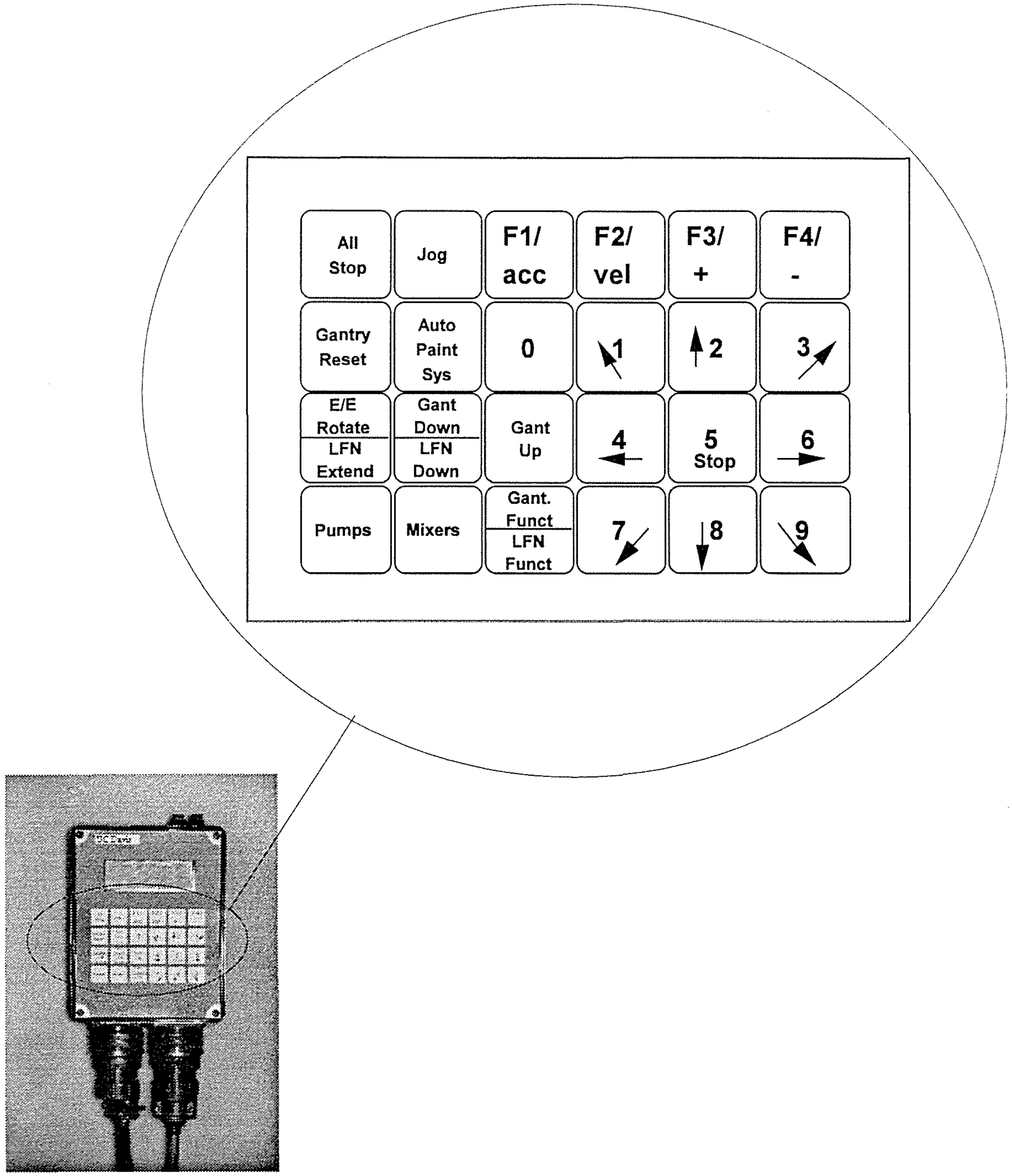
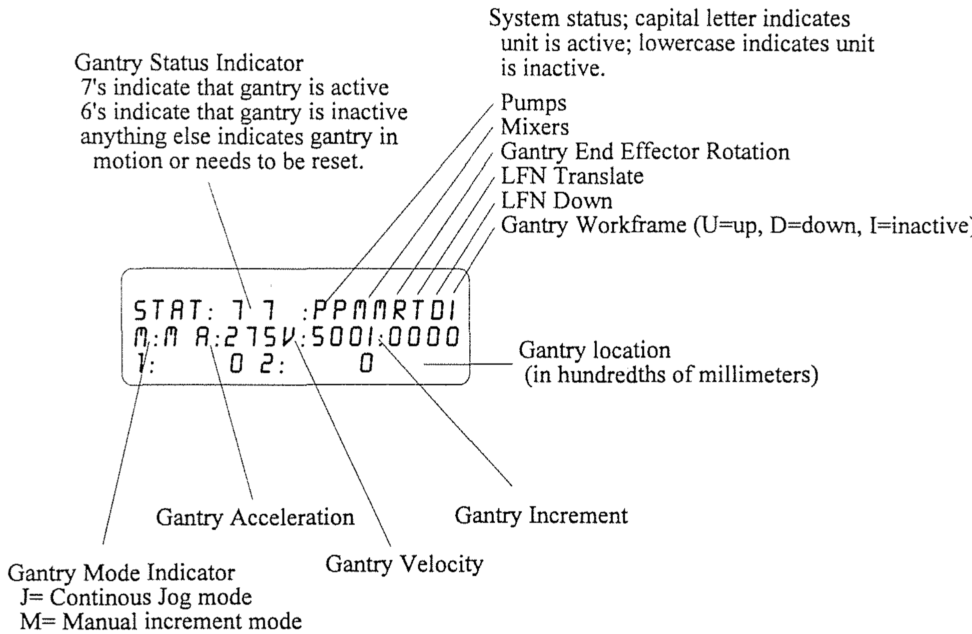
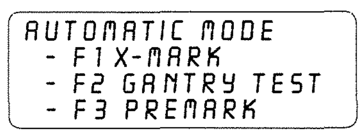


Figure 1.3: Hand Controller



Status Display



Automatic Menu

Figure 1.4: LCD Status Display

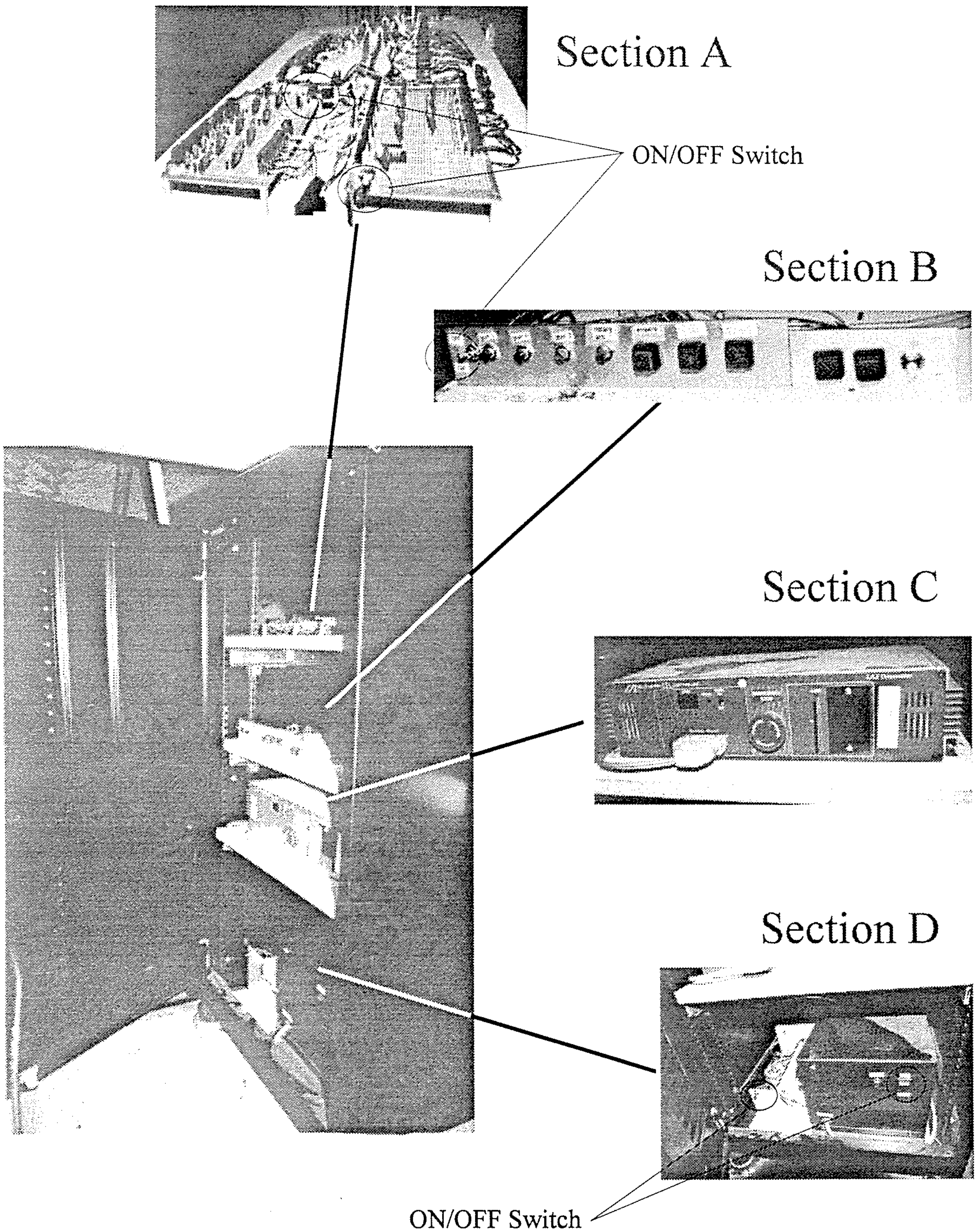


Figure 1.5: Controller Cabinet

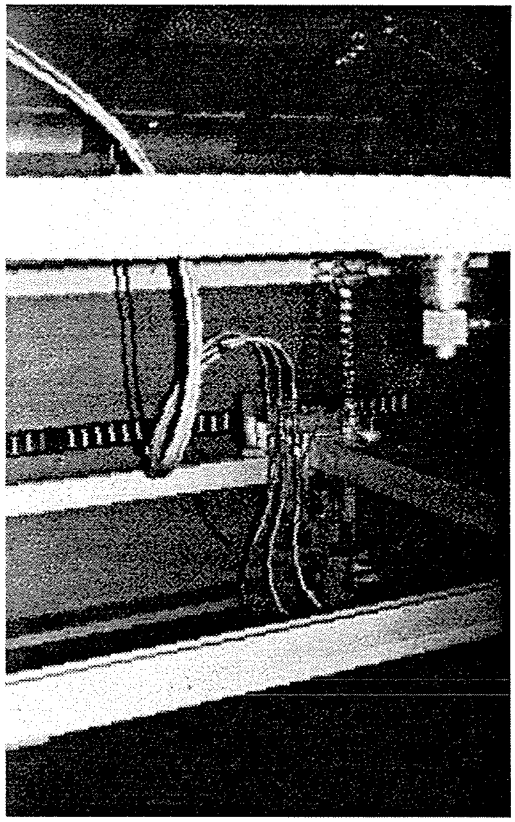


Figure 1.6: Gantry Robot

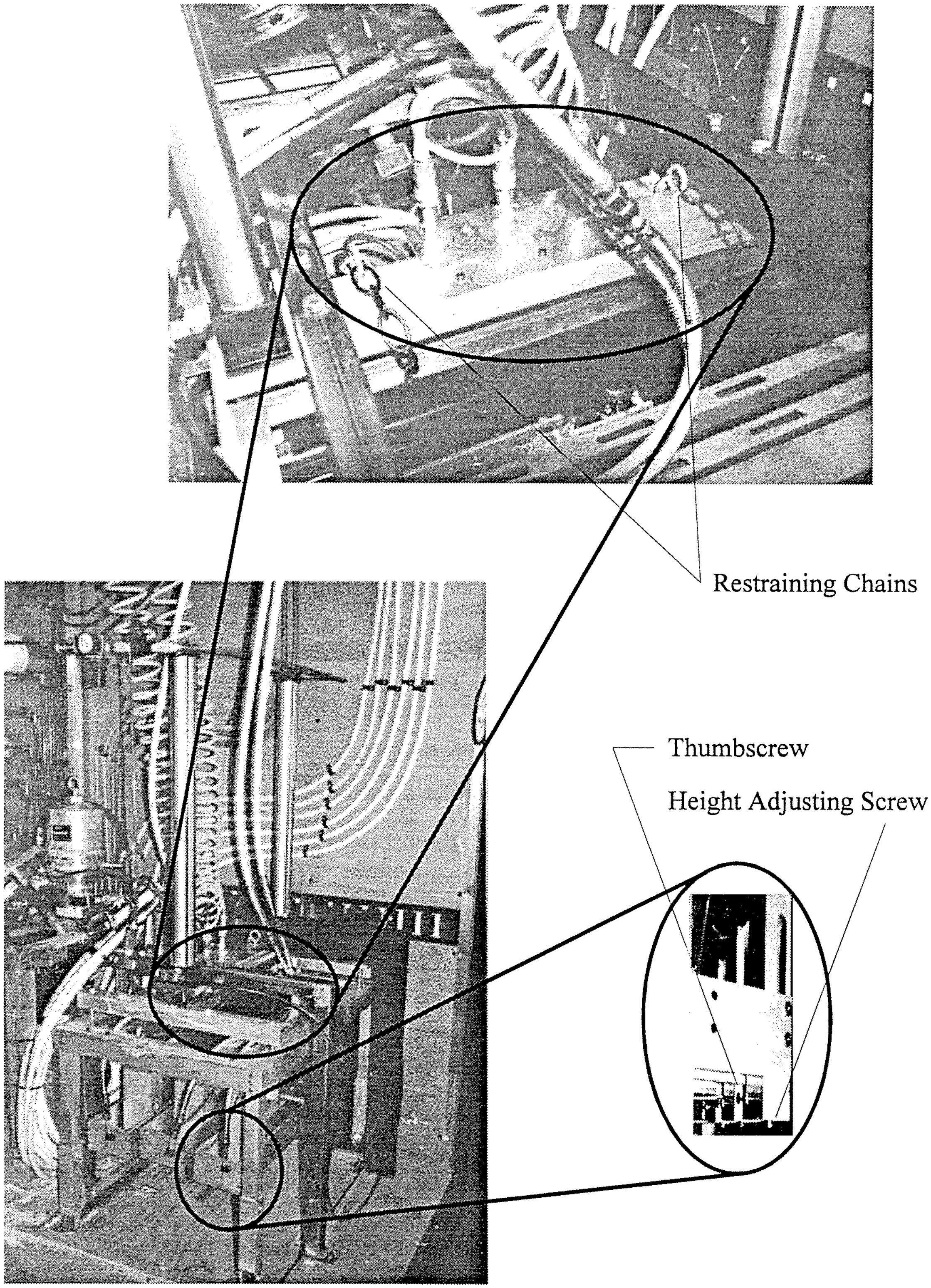


Figure 1.7: Linear Free Nozzle System

2.0 System Startup

2.1 Introduction

This section describes the startup procedures common to the operation of the Gantry Free Nozzle system and the Linear Free Nozzle system. To reduce the startup time, the operations described here can be accomplished in parallel with multiple workers. Please refer to Figure 1.1 for the location of equipment. Before proceeding, check to make sure that all access covers and floor covers have been removed, and all openings exposed. The gantry workframe legs may be removed to facilitate the removal of the floor covers.

2.2 Mixing the Paint

! Warning !

For best performance, pay particular attention to the addition of residual water to the paint. Small amounts of water will seriously degrade the quality of the premarks painted on the road.

2.2.1 Using a screwdriver, carefully remove the lid from the 5 gallon paint can. One can of black and one can of white traffic paint will be necessary to prime the system and begin initial calibration of the systems. Use Pervo Paint Fast-drying Water Based Road Paint, #4773A (white) and #4775A (black).

2.2.2 Using an approved mixing device, mix the paint. Pay particular attention to the sediment at the bottom of the cans, since the paint may have settled during shipment from the factory.

! Warning !

Failure to properly mix the paint to a smooth consistency will result in clogged pump siphon screens and a loss of paint pressure.

2.2.3 When all paint has been mixed, carefully place the paint containers in the pump stand. The black paint should be placed in the stand on the side nearest the front of the trailer. The white paint should be placed in the stand on the side nearest the back of the trailer. Carefully insert the siphon and agitator into the paint. Place the cover on top of the paint can.

2.3 Starting the Air Compressor

General instructions are provided here for starting the Air Compressor. If more details are required, please see the Air Compressor's Manufacturer manual.

2.3.1 Close the main system outlet valve, located near the air compressor tank (Figure 2.3.1).

2.3.2 Actuate the compressor unloader valve. Pull up on end of the pin, until it is in an upright position (Figure 2.3.2).

2.3.3 Open the gasoline valve, located near the gasoline tank.

2.3.4 Apply full choke. The knob is located to the left of the ignition key.

2.3.5 Crank engine by turning the key switch to the 'crank' position. Crank engine a maximum of 10 seconds to prevent overheating of starter motor.

2.3.6 Once engine has started, allow 2 minutes for the engine to reach operating temperature. Once operating temperature has been reached, reduce choke, and release pin on the unloader valve.

! Warning !

The crankcase, cylinder head, exhaust system, and other components can get extremely hot from operation. To prevent severe burns, do not touch these areas while the engine is running or immediately after it is turned off. Never operate the engines with heat shields or guards removed.

! Warning !

Keep hands, feet, hair, and clothing away from all moving parts to prevent injury. Never operate the engine with covers, shrouds, or guards removed.

2.3.7 Allow 5 minutes for the air pressure to reach operating pressure.

2.3.8 Open the main system outlet valve.

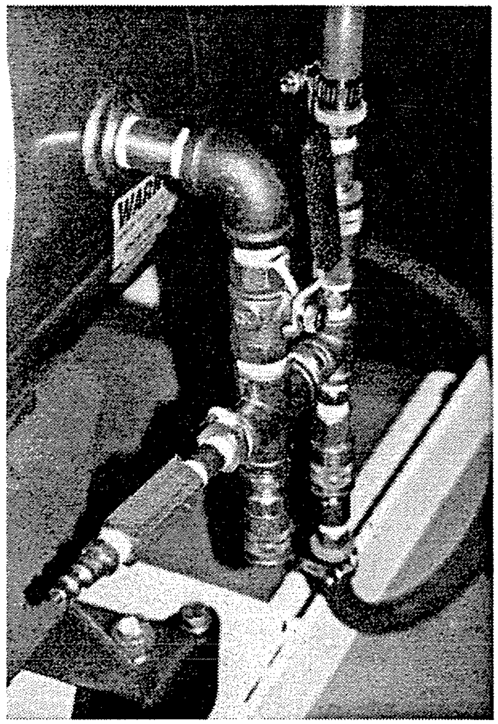


Figure 2.3.1: Main Air Valve

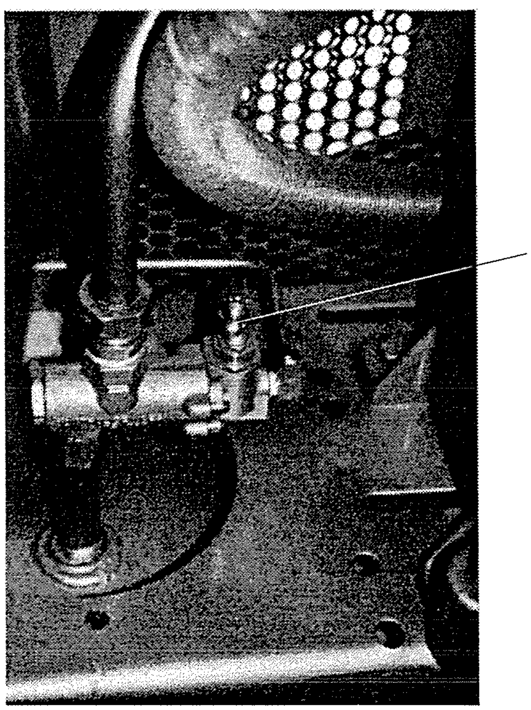


Figure 2.3.2: Unloader Valve

2.4 Starting the Generator

General instructions are provided here for starting the Generator. If more details are required, please see the Generator's Manufacturer manual.

2.4.1 Unplug all loads connected to generator by removing plugs at generator outlet panel.

! Warning !

Failure to unplug all loads connected to generator may result in damage to electrical loads and/or damage to generator unit.

2.4.2 Open gasoline shutoff valve. Turn on ignition switch, located near cylinder head, to the 'ON' position.

2.4.3 Apply full choke. The choke actuation lever is located at the base of the air cleaner.

2.4.4 Press the starter button. Apply the starter a maximum of 10 seconds. Allow 1 minute between start attempts to allow starter to cool.

! Warning !

Failure to allow starter to cool may cause overheating and premature failure of the starter motor.

2.4.5 Once engine has started, allow it to warm up 2 minutes.

2.4.6 Place choke in run position.

2.4.7 Verify that all electrical equipment is off (Figure 1.5). Also check the Control Cabinet to ensure that the Gantry power supply and Power transform is off. (Figure 1.5-d and Figure 2.4.2)

2.4.8 Plug the fan and the power conditioner (Figure 2.4.1) into the outlet panel of the generator.

2.4.9 Turn on the fan to the 'HI' setting and turn on the power conditioner.

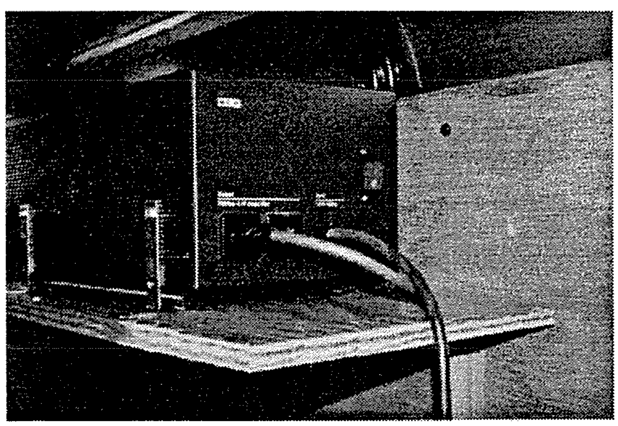


Figure 2.4.1: Power Conditioner

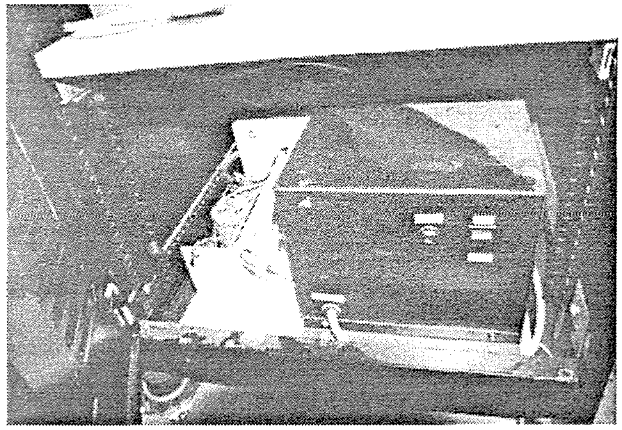


Figure 2.4.2: Gantry Power Supply and Power Transformer

2.5 Hand Controller Setup

2.5.1 Retrieve the Hand Controller (Figure 2.5.1) and attach the cables. The larger cable attaches to the left plug.

2.5.2 Remove the covers from the pipe fittings at the front of the trailer. Place the covers in a safe storage place.

2.5.2 Insert the other cable end into the pipe fittings. Attach the cable end to the mating plug inside the trailer. (Figure 2.5.2)

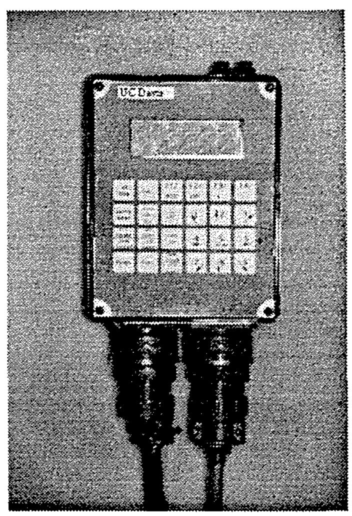


Figure 2.5.1: Hand Controller

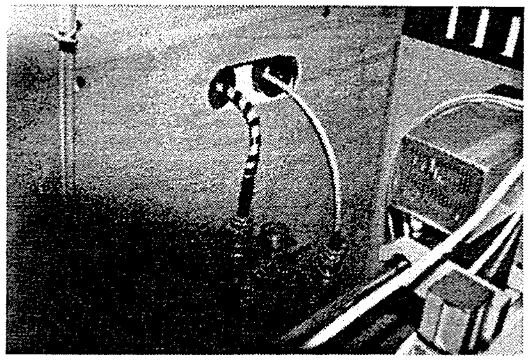


Figure 2.5.2: Cable Plugs Inside Trailer

2.6 System Power On

! Warning !

Before turning on the Gantry, confirm that no personnel are located within the workspace.

2.6.1 Turn on the power transformer, located on the bottom of the Control Cabinet. (Figure 2.6.1)

2.6.2 Turn on the Gantry power supply, located on the bottom of the Control Cabinet. (Figure 2.6.1) The Gantry LED display (Figure 1.5-c) should show 'EP'. If not, re-cycle power by turning off the Gantry power supply, waiting 30 seconds, and turning it on again.

2.6.3 Turn on the Solenoid Interface boards, located on the top shelf of the Control Cabinet. (Figure 2.6.2)

2.6.4 Turn on the Hand Controller, using the Red switch on the left. (Figure 2.6.3)

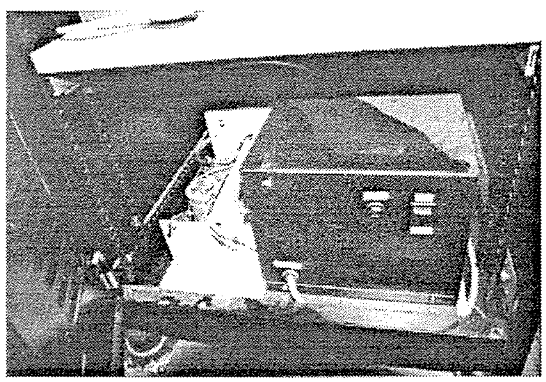


Figure 2.6.1: Gantry Power Supply and Power Transformer

Figure 2.6.2: Solenoid Interface Boards

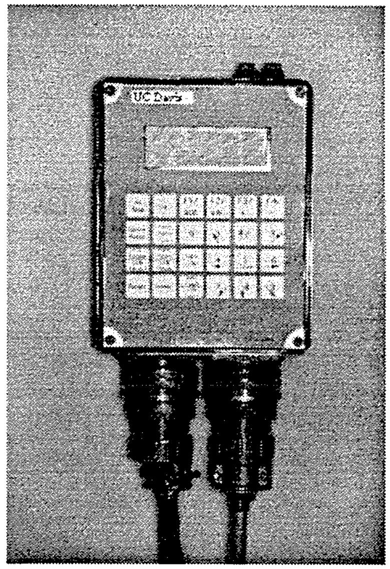


Figure 2.6.3: Hand Controller

2.7 Priming the Paint Pumps

2.7.1 Close the main air inlet to the paint pumps. The inlet is closed when the handle is in the horizontal position. (Figure 2.7.1)

2.7.2 Close the paint pump outlet valves. Also, close the paint pump bleed valve. (Figure 2.7.2 and Figure 2.7.3)

2.7.3 Place the manual control switch in the 'ON' position and depress the 'PUMPS' switch (Figure 2.7.4). If desired at this time, the paint mixers can be activated to keep the paint mixed. To activate the mixers, depress the 'MIXERS' switch. (Figure 2.7.3)

2.7.4 If the mixers are to be used, adjust the mixing speed by turning the control valve to increase or decrease operating speed. (Figure 2.7.5)

2.7.5 Adjust the black and white pump air regulators, located on the Regulator Panel, to approximately 60 PSI (Figure 2.7.5)

2.7.6 Place a bucket at the end of the bleed valve hose.

2.7.7 Open the pump bleed valve. (Figure 2.7.2)

! Warning !

Exercise caution - pump may be under extremely high pressure. Secure hose to prevent 'hose whip' when pressure is relieved.

2.7.8 Open pump air inlet valve. (Figure 2.7.1)

2.7.9 Pump is fully primed when paint is flowing from bleed valve hose.

2.7.10 Close inlet air valve. (Figure 2.7.1)

2.7.11 Close bleed valve. (Figure 2.7.2)

2.7.12 Repeat steps 2.7.8 through 2.7.11 for the other paint pump.

2.7.13 Depress 'PUMPS' switch to deactivate the pumps.

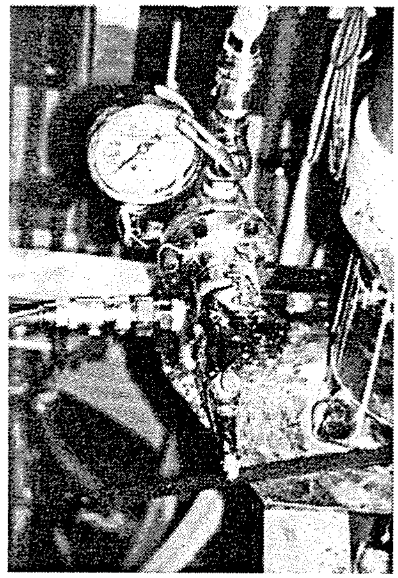


Figure 2.7.1: Pump Air Inlet Valve

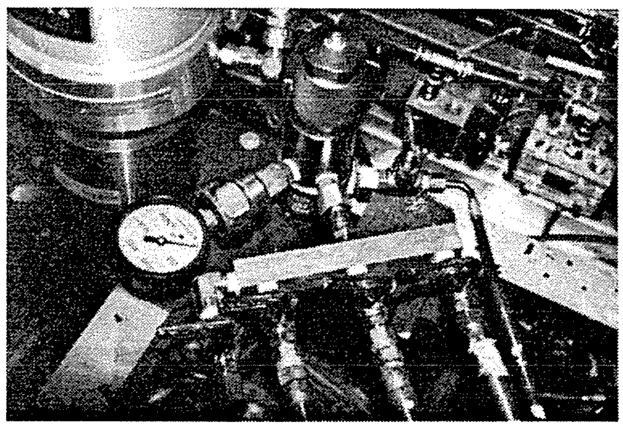


Figure 2.7.2: Pump Bleed Valve and Pump Outlet Valves

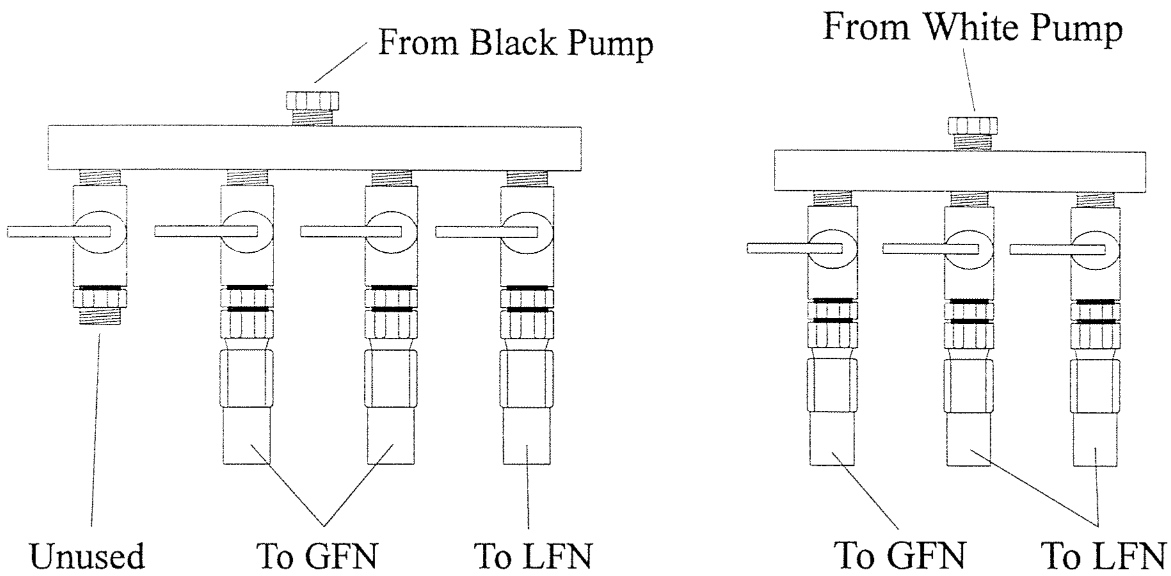


Figure 2.7.3: Pump Outlet Locations

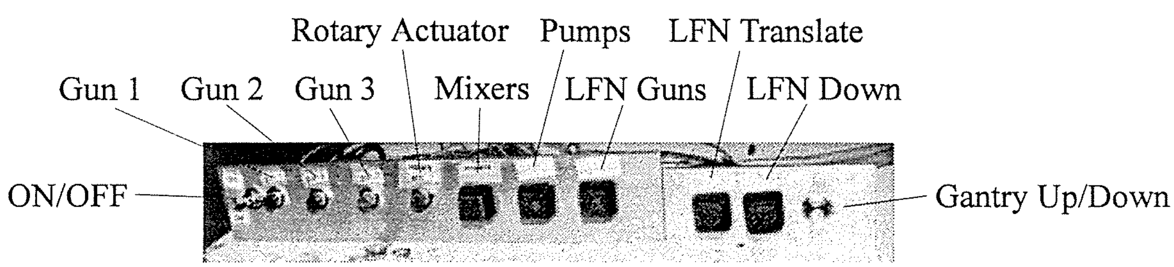


Figure 2.7.4: Manual Control Panel

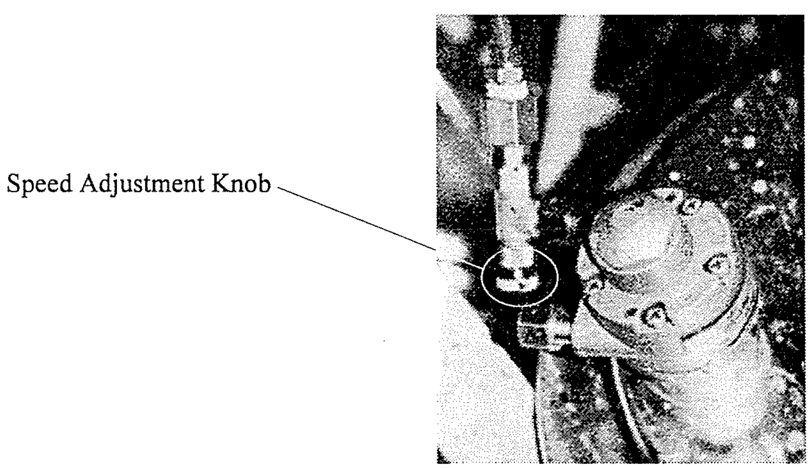


Figure 2.7.5: Mixer Speed Control

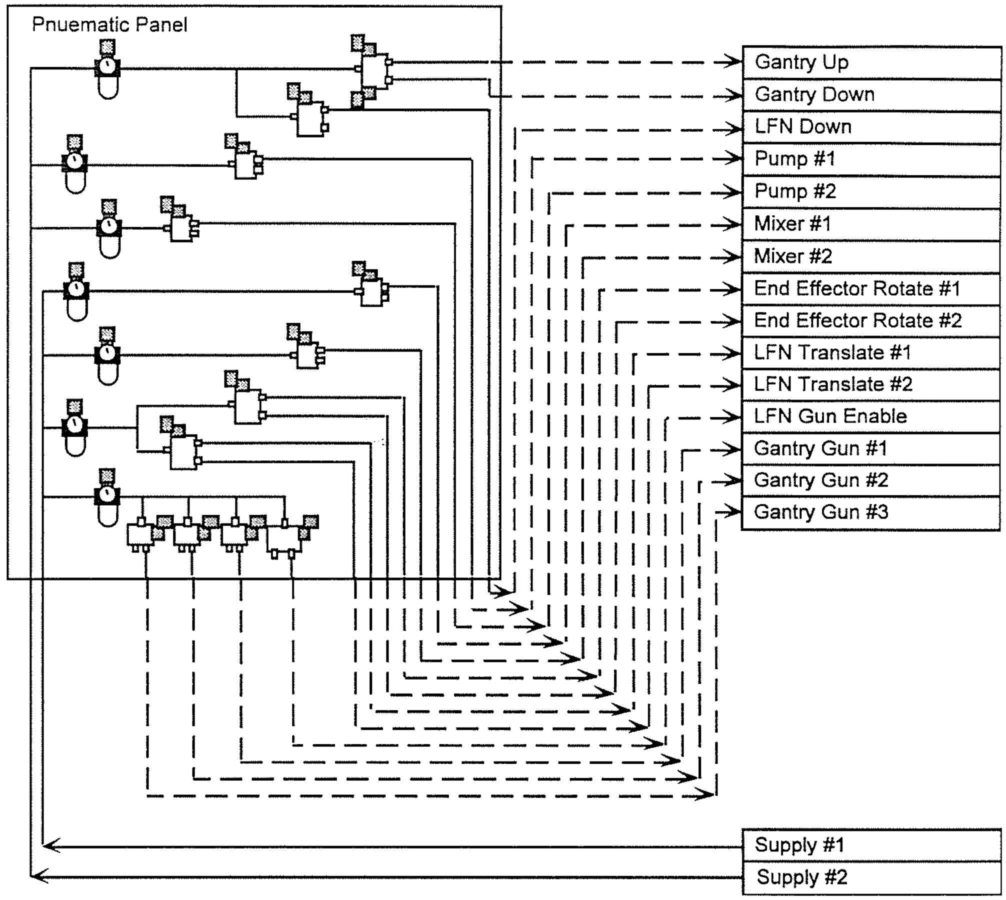


Figure 2.7.6: Regulator Panel

2.8 Priming the Paint Lines

! Warning !

The paint will flow from the gun outlet with considerable pressure. Skin damage may result if contact is made with any body part.

! Warning !

Before operating the gantry workframe, insure that there are no obstructions.

2.8.1 Lower the Gantry frame to a comfortable working height by actuating the 'UP/DOWN' switch on the Manual Control Panel.

2.8.2 Place a bucket underneath the gun outlets of the Gantry End Effector (Figure 2.8.1)

2.8.3 Open the black pump air inlet valve. (Figure 2.7.1)

2.8.4 Open the black paint outlet valves connected to the Gantry. There should be 2 black paint valves to open. Reference Figure 2.7.3 for the location.

2.8.5 Depress the 'GUN 1' switch to open the first black paint gun. With the 'GUN 1' switch depress, press the 'PUMPS' switch to activate the paint pump. (Figure 2.7.4)

2.8.6 Allow pump to run, until paint runs freely from the gun outlet.

2.8.7 Depress the 'PUMPS' switch to deactivate the pump. Release the 'GUN 1' switch. (Figure 2.7.4)

2.8.8 Depress the 'GUN 3' switch. With the 'GUN 3' switch depress, press the 'PUMPS' switch. (Figure 2.7.4)

2.8.9 Allow pump to run, until paint runs freely from the gun outlet.

2.8.10 Depress the 'PUMPS' switch to deactivate the pump. Release the 'GUN 3' switch. (Figure 2.7.4)

2.8.11 Close the black pump air inlet valve. (Figure 2.7.1)

2.8.12 Open the white pump air inlet valve. (Figure 2.7.1)

2.8.13 Open the white paint outlet valves connected to the Gantry. There should be 1 white paint valve to open (Figure 2.7.3).

2.8.14 Depress the 'GUN 2' switch to open the first white paint gun. With the 'GUN 2' switch depress, press the 'PUMPS' switch to activate the paint pump. (Figure 2.7.4)

2.8.15 Allow pump to run, until paint runs freely from the gun outlet.

2.8.16 Depress the 'PUMPS' switch to deactivate the pump. Release the 'GUN 2' switch. (Figure 2.7.4)

2.8.17 Close the white pump air inlet.

2.8.18 The Gantry paint lines are now primed.

2.8.19 Position a bucket underneath the paint gun outlets of the Linear Free Nozzle (LFN) system.

2.8.20 Depress the 'Pre. Guns' button on the manual control panel (Figure 2.7.4)

2.8.21 Depress the 'PUMPS' button. (Figure 2.7.4)

2.8.22 Open the pump air inlet valves for both paint pumps (Figure 2.7.1).

2.8.23 Open the black and white pump outlet valves. There should be 2 white outlet valves and 1 black outlet valve to open (Figure 2.7.3).

2.8.24 Locate the roller cam valve for each of the guns (Figure 2.8.2). Depressing the cam valve will open the paint gun. Depress each one until paint flows freely from the gun outlet.

2.8.25 Close the air inlet valve to both pumps.

2.8.26 Depress the 'PUMPS' switch to deactivate the pumps. Depress the 'Pre. Guns' to deactivate the LFN guns.

2.8.27 Remove the paint buckets from underneath the Gantry and LFN systems.

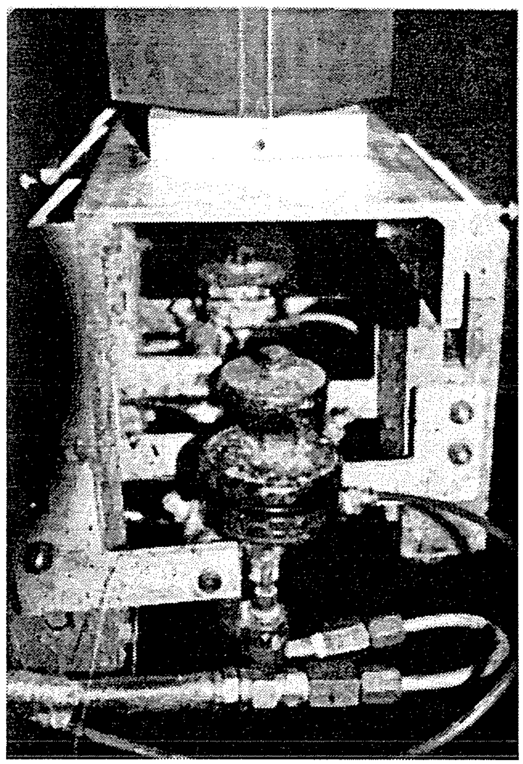


Figure 2.8.1: Gantry Guns

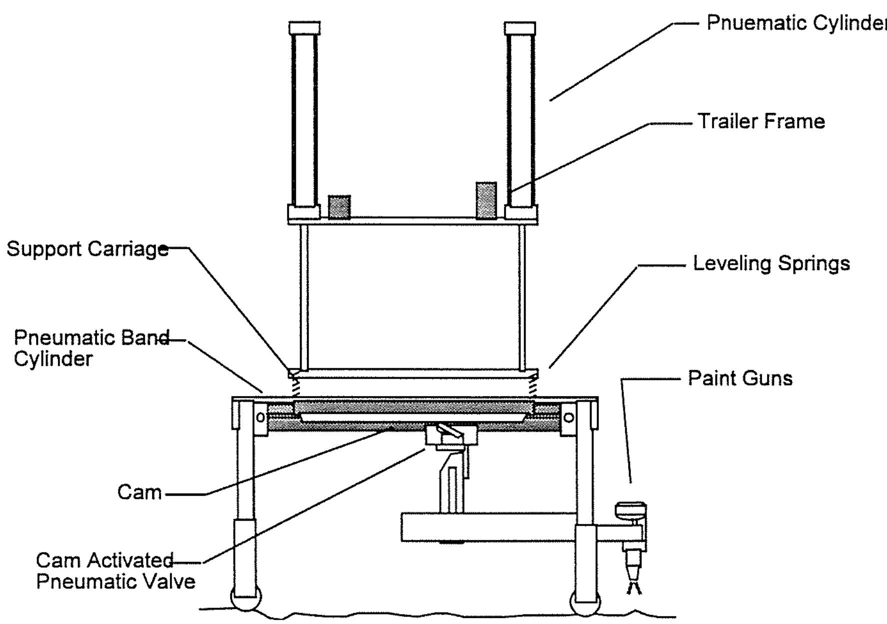


Figure 2.8.2: Roller Cam Location on LFN

2.9 Nozzle Installation

! Warning !

The paint will flow from the nozzle tip with considerable pressure. Skin damage may result if contact is made with any body part.

! Note !

Should any system malfunction during automatic control, depress the LEFT red switch on the hand controller to activate the EMERGENCY STOP. Note that gantry motion may not cease immediately.

2.9.1 Prepare 3 (three) size 1380 Binks paint nozzles. Also prepare 3 (three) nozzle filters. Pay particular attention to stray dirt and paint specks.

! Warning !

Lack of cleanliness will result in abnormal spray patterns and lack of performance from the spray nozzle.

2.9.2 Assemble the nozzle unit, as shown in Figure 2.9.1.

! Note !

The paint nozzle is keyed to fit into the orange spray tip.

2.9.3 Screw the nozzle unit onto the end of the Gantry spray gun. Using a 7/8" wrench, tighten the mount nut.

2.9.3 With the orange spray tip hand tight, turn the tip 90 degrees from the final desired position, perpendicular to the line of travel of the spray head. See Figure 2.9.2 for the orientation of the spray pattern.

2.9.4 Tighten the Nozzle nut by turning the nut 90 degrees. The orientation should be as close to perpendicular to the line of travel as possible.

2.9.5 Repeat Steps 2.9.2 through 2.9.4 for the rest of the guns.

2.9.6 Prepare 2 (two) size 1180 and 1 (one) size 1380 Binks spray tip. Also prepare 3 (three) nozzle filters. Pay particular attention to stray dirt and paint specks. Unchain the 2 chains restraining the LFN unit. (Figure 2.9.3)

! Warning !

Lack of cleanliness will result in abnormal spray patterns and lack of performance from the spray nozzle.

2.9.7 Repeat Steps 2.9.2 through 2.9.4. Note that the 2 (two) size 1180 spray tips are to be located on the outside guns of the LFN and the size 1380 tip on the center gun.

2.9.8 Place a piece of Simulated Road Material (i.e., Mineral Surface Rolled Roofing), approximately 1 foot wide by 3 foot long, under the Gantry end effector. (Figure 2.9.4) Place a similar piece under the guns of the LFN.

2.9.9 Position the Gantry by actuating the 'UP/DOWN' switch on the manual control panel (Figure 2.7.2) so that the gantry gun tips are approximately 12 inches above the Simulated Road Material.

2.9.10 Depress the 'PUMPS' switch to activate the paint pumps.

2.9.11 Open the main air inlet for both pumps. (Figure 2.7.1) The valve handle should be in the vertical position.

2.9.12 Open the outlet valves for the Gantry system. There should be 2 black valves and 1 white valve to open. Reference 2.7.3 for valve location.

2.9.13 On the regulator panel, adjust the air regulator to 130 PSI for the black pump, and 115 PSI for the white pump. (Figure 2.7.6)

2.9.14 Depress the 'GUN 1' switch for approximately 2 seconds. Depress the 'GUN 3' switch for approximately 2 seconds. Depress the 'GUN 2' switch for approximately 2 seconds. The resulting spray pattern should look like Figure 2.9.5. Use a wrench to adjust the orientation of the orange spray tip to bring the spray pattern into alignment. Manually pivot the entire gun to adjust the start point of the spray pattern.

2.9.15 Repeat Step 2.9.14 with a new piece of Simulated Road Material until alignment is achieved.

2.9.16 Depress the 'Pre. Guns' button on the manual control panel to activate the LFN guns.

2.9.17 Depress the 'Pre. Down' button on the manual control panel to lower the LFN.

2.9.18 Depress each of the cam valves for approximately 2 seconds each. The spray pattern should look similar to Figure 2.9.5. Use a wrench to adjust the orientation of the tip to bring the spray pattern into alignment. Manually pivot the gun to adjust the start point of the spray pattern.

2.9.19 Depress the 'Pre. Down' button on the manual control panel to raise the LFN. Depress the 'Pre. Guns' button to deactivate the LFN guns.

2.9.20 Place the manual control panel master switch to 'OFF'.

2.9.21 Cut a 3' wide by 4' long section of Simulated Road Material. Place it into the Gantry workspace as shown in Figure 2.9.6.

2.9.22 Using the hand controller, press the 'mixers' button to activate the mixers under computer control. The LCD display should respond as shown in Figure 2.9.7.

2.9.23 Manually move the gantry end effector to the center of the workspace. Press the 'Gantry Reset' button. The gantry workframe should rise and the end effector should move to the corner. Reference Figure 2.9.6 for correct orientation.

2.9.24 Press the 'GFN Funct/LFN Funct' button to select Gantry functions. Press the 'Gantry Down' button. The gantry workframe should begin lowering. When it reaches the ground and shuts off, press the 'Gantry Down' button again.

2.9.25 Press the 'Pumps' button to activate the paint pumps. Press the 'Mixers' button to deactivate the paint mixers. Reference Figure 2.9.7 for correct LCD display indications.

2.9.26 Press the 'Auto Paint System' button. The menu (Figure 2.9.8) should be shown. Press 'F2' to activate the gantry test.

2.9.27 When the gantry motion is complete, deactivate the paint pumps by pressing the 'Pumps' button and re-activate the mixers by pressing the 'Mixers' button.

2.9.28 Observe the painted test mark. The black-white-black stripes should meet, without overlapping edges. If not, continue to Step 2.9.29, otherwise continue to Step 2.9.30.

! Note !

The quality of first few runs after initial system startup will be poor since the nozzle tips will contain water and entrained air.

It is recommended that the test mark be performed a few times before adjustment to the system is made.

2.9.29 Loosen the thumbscrew of the gun assembly that needs adjustment. (Figure 2.9.9) Using a 9/16" socket on the height adjustment screw (Figure 2.9.9) adjust the height of the gun assembly. Raising the gun assembly increases the width of the spray pattern; lowering the gun assembly decreases the width of the spray pattern. Adjust gun assembly accordingly. Retighten the thumbscrew when finished. Adjust all guns. Repeat Steps 2.9.25 through 2.9.29 with a new piece of Simulated Road Material until all stripe edges meet.

2.9.30 Press the 'Gant Up' button to raise the gantry workframe. Raise workframe approximately halfway. Press the 'Gant Up' again button to stop the workframe at the desired position.

2.9.31 Place a piece of Simulated Road Material underneath the LFN.

2.9.32 Press the 'Pumps' button to activate the pumps. Press the 'Mixers' button to deactivate the mixers.

2.9.33 Press the 'Auto Paint Sys' button. The menu should be displayed. Select 'F3' to paint an abbreviated premark. The LFN should cycle. Observe the mark painted. The edges of all stripes should meet, with no overlap. If not, proceed to Step 2.9.33, otherwise continue to Step 2.9.35.

! Note !

The quality of first few runs after initial system startup will be poor since the nozzle tips will contain water and entrained air.

It is recommended that the test mark be performed a few times before adjustment to the system is made.

2.9.34 Loosen the thumbscrew of the gun assembly that needs adjustment. (Figure 2.9.10) Using a 9/16" socket on the height adjustment screw (Figure 2.9.10) adjust the height of the gun assembly. Raising the gun assembly increases the width of the spray pattern; lowering the gun assembly decreases the width of the spray pattern. Adjust gun assembly accordingly. Retighten the thumbscrew when finished. Adjust all guns. Repeat Steps 2.9.33 through 2.9.34 with a new piece of Simulated Road Material until all stripe edges meet.

2.9.35 Press the 'Pumps' button to deactivate the pumps. Press the 'Mixers' button to activate the mixers.

2.9.36 All calibration procedures are now complete.

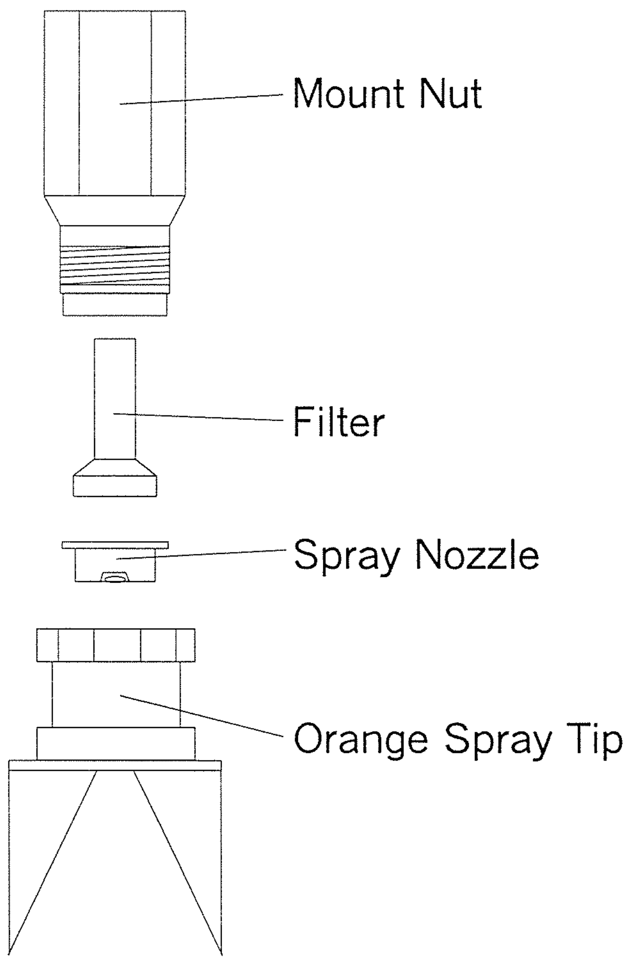


Figure 2.9.1: Nozzle Unit

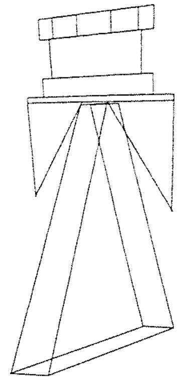


Figure 2.9.2: Spray Pattern

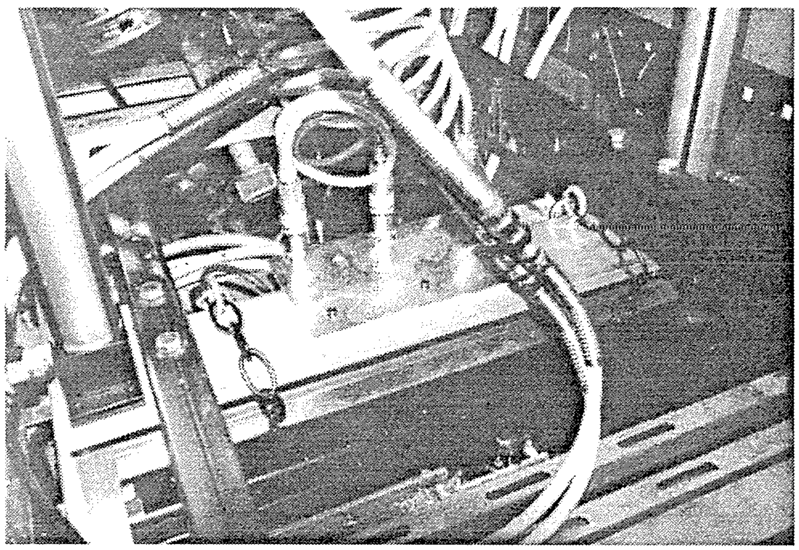


Figure 2.9.3: LFN Chains

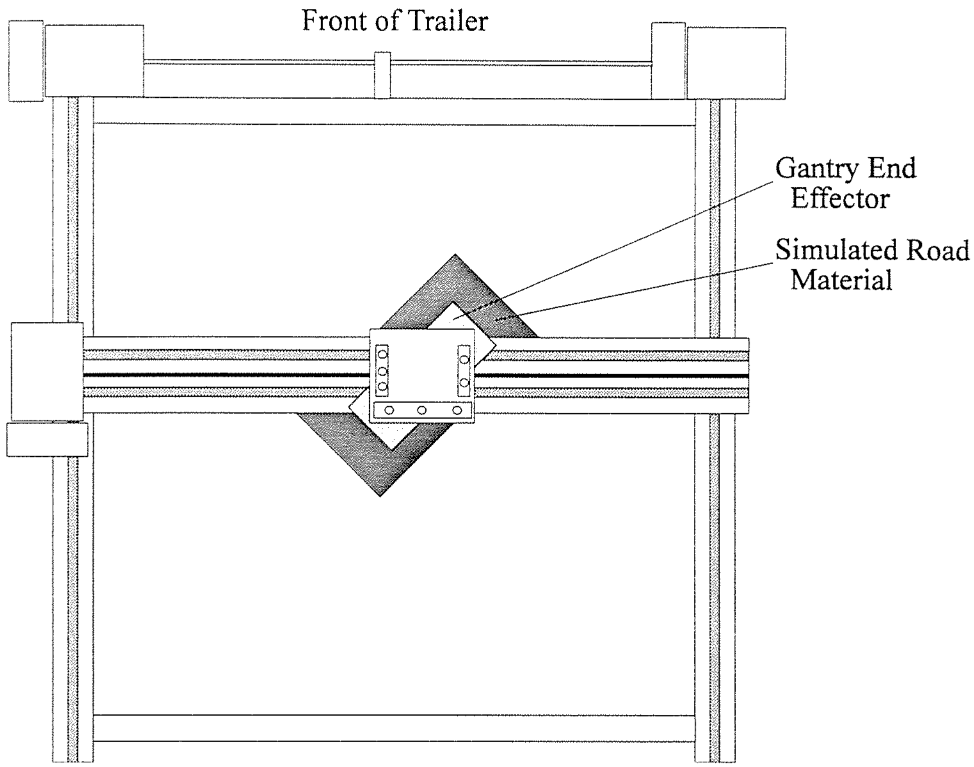


Figure 2.9.4: Material Installation

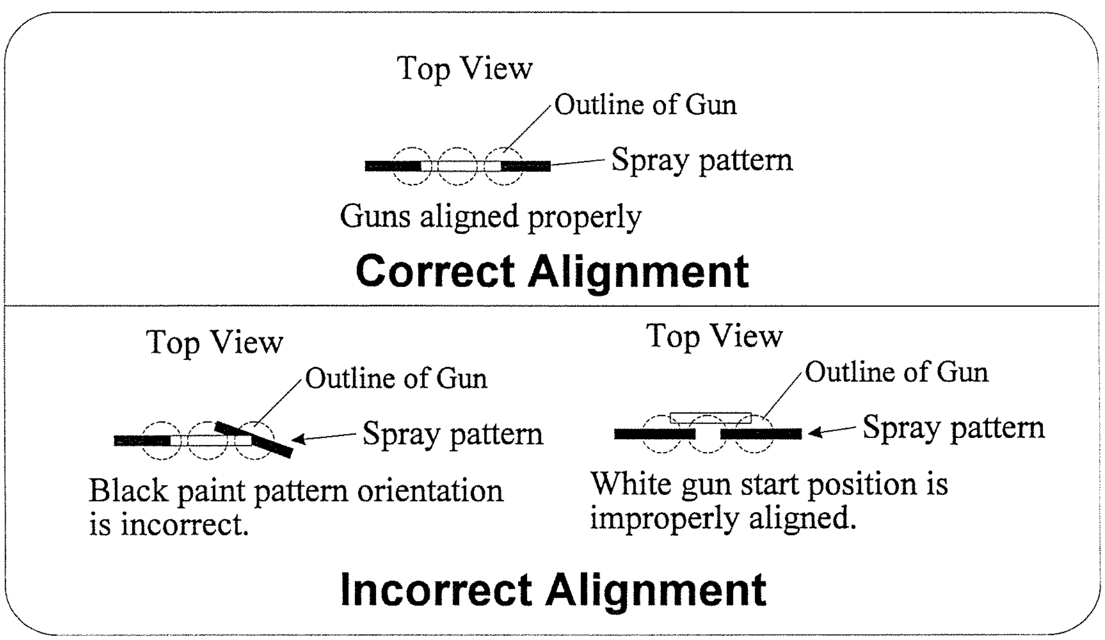


Figure 2.9.5: Spray Orientation

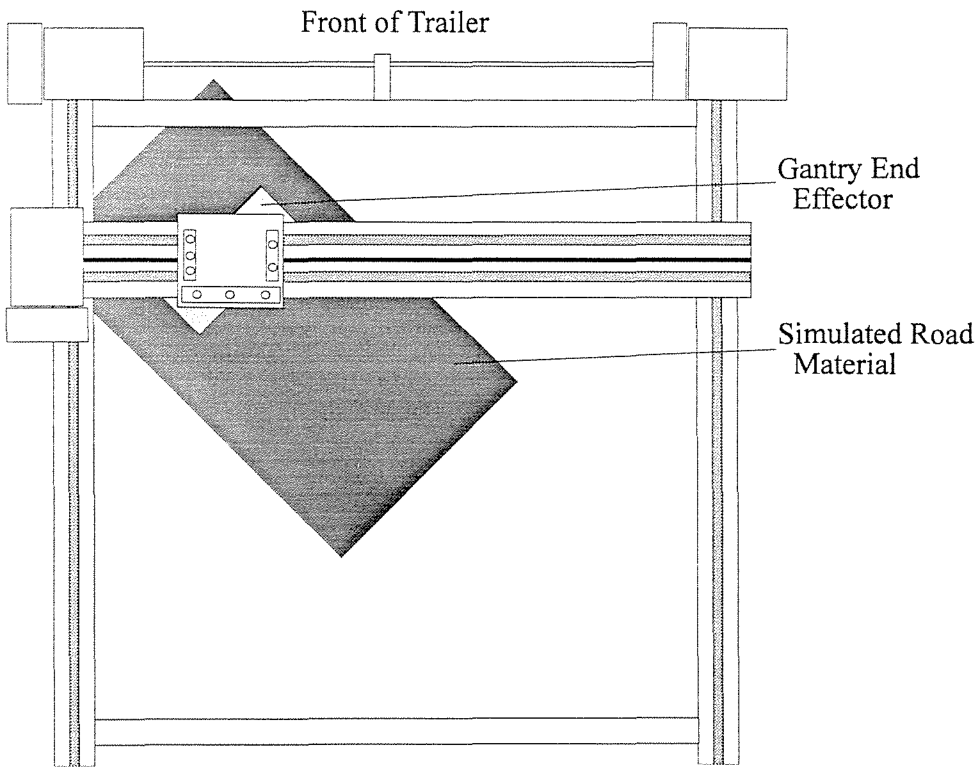


Figure 2.9.6: Material Orientation for Gantry Test

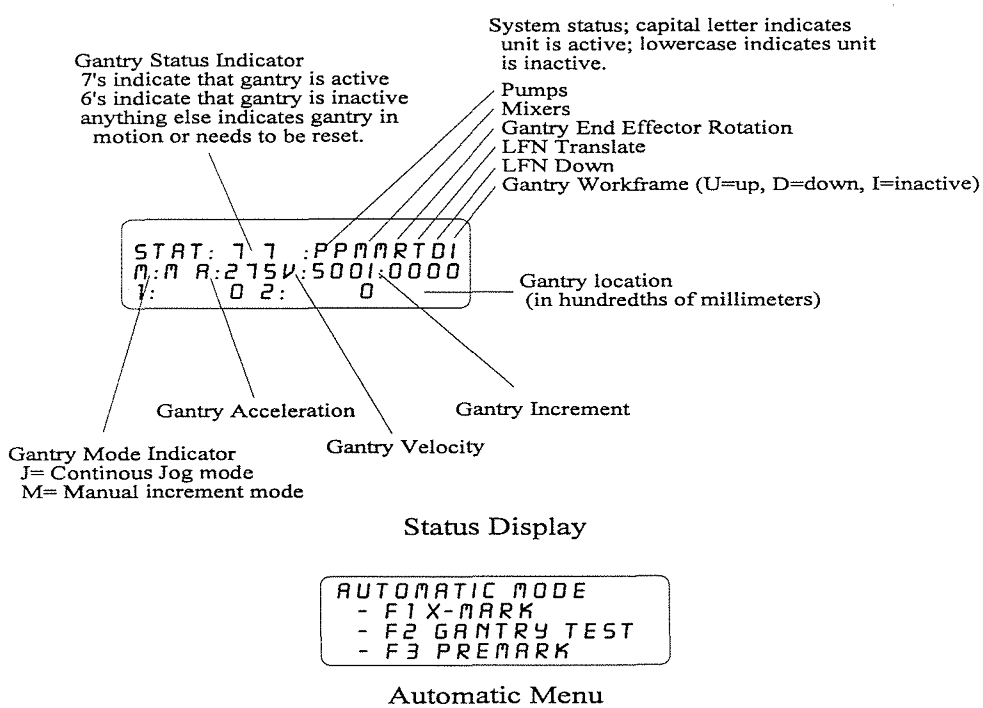


Figure 2.9.7: LCD Status Display

```
AUTOMATIC MODE
 - F1 X-MARK
 - F2 GANTRY TEST
 - F3 PREMARK
```

Automatic Menu

Figure 2.9.8: LCD Menu Display

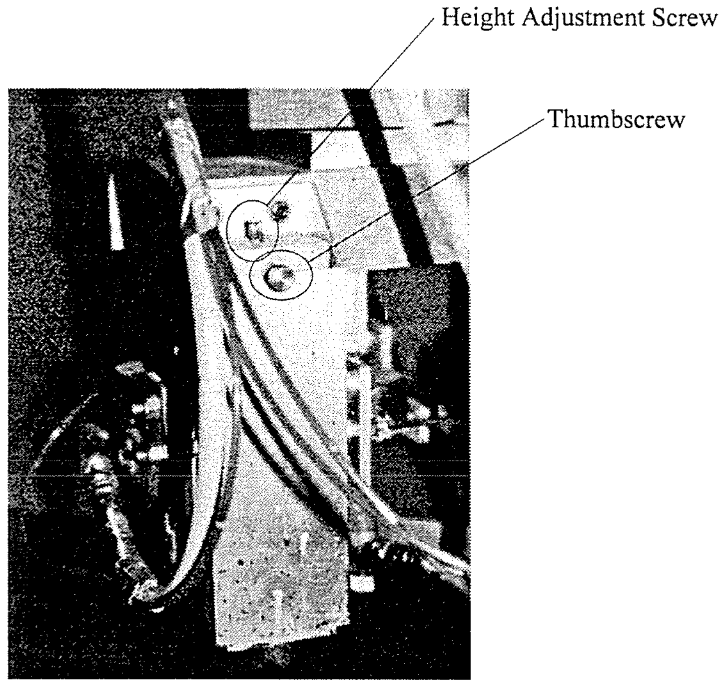


Figure 2.9.9: Gantry Gun Adjustment Locations

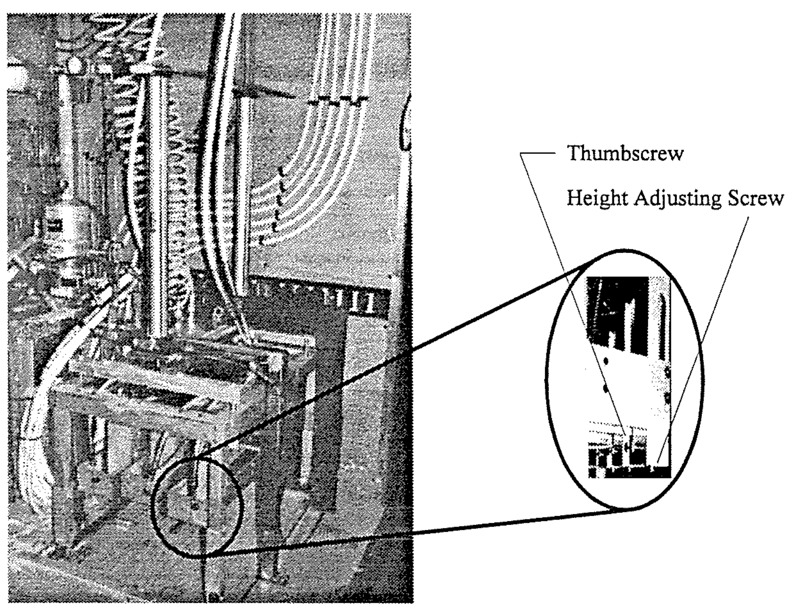


Figure 2.9.10: LFN Gun Adjustment Locations

3.0 Normal Operations

3.1 Introduction

This sections describes the normal operating procedure of the Premark trailer.

3.2 Placing a premark

! Note !

The surveying marks must be placed according to standard CalTrans operating procedures. Please reference the appropriate CalTrans standard for exact details.

3.2.1 When the appropriate location has been established, look at the status panel on the LCD of the Hand controller (Figure 3.2.1). If the paint mixers are active, press the 'Mixers' button on the Hand controller to deactivate the paint mixers.

3.2.2 Press the 'Pumps' button to activate the paint pumps.

3.2.3 Press the 'Auto Paint Sys' button to bring up the selection menu.

3.2.4 Press 'F1' to begin the process of painting the premark.

3.2.5 Wait approximately 2.5 minutes. When the LCD display indicates that the process has completed, disable the pumps by press the 'Pumps' button. Re-activate the paint mixers by pressing the 'Mixers' button.

3.2.6 Proceed to the next location.

3.3 Placing an abbreviated premark

! Note !

The surveying marks must be placed according to standard CalTrans operating procedures. Please reference the appropriate CalTrans standard for exact details.

3.3.1 When the appropriate location has been established, look at the status panel on the LCD of the Hand controller (Figure 3.2.1). If the paint mixers are active, press the 'Mixers' button on the Hand controller to deactivate the paint mixers.

3.3.2 Press the 'Pumps' button to activate the paint pumps.

3.3.3 Press the 'Auto Paint Sys' button to bring up the selection menu.

3.3.4 Press 'F3' to begin the process of painting the abbreviated premark.

3.3.5 Wait approximately 2.5 minutes. When the LCD display indicates that the process has completed, disable the pumps by press the 'Pumps' button. Re-activate the paint mixers by pressing the 'Mixers' button.

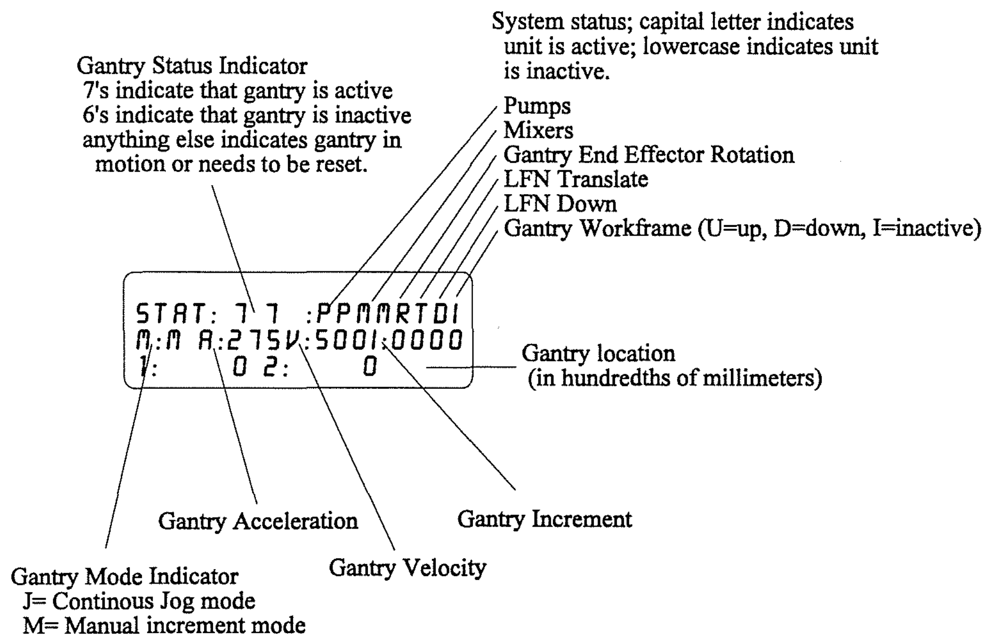
3.3.6 Proceed to the next location.

3.4 On Road Maintenance

3.4.1 Deactivate the gantry by pressing the 'Gant Funct/LFN Funct' button. Check the hand controller LCD panel to confirm that the gantry is deactivated (Figure 3.2.1).

3.4.2 Manually position the gantry end effector to allow for easy reach of the spray nozzles.

3.4.3 For optimum results, the spray nozzles should be lightly scrubbed with a wet brush every 2 hours. Also, the nozzles should be lightly scrubbed if the machine is not going to be used for a period of more than 30 minutes.



Status Display

Figure 3.2.1: Status Panel of the Hand Controller

4.0 Cleanup Procedures

! Warning !

Waste paint is toxic. Disposed of waste paint according to established toxic waste procedures. Consult the appropriate CalTrans standard for details.

! Warning !

The paint will flow from the nozzle tip with considerable pressure. Skin damage may result if contact is made with any body part.

4.1 Introduction

This section describes the cleanup procedures. Some sections maybe performed in parallel, if necessary, to speed up clean up time.

4.2 Nozzle Cleanup

4.2.1 Using a 9/16" wrench remove the complete nozzle unit from the end of the spray guns. Disassemble the unit.

4.2.2 Flush each component thoroughly under running water. Use a small scrub brush to remove deposits from the components. Pay particular attention to the spray tip orifice.

! Warning !

Failure to clean the spray tip thoroughly will result in a clogged tip. Decreased painting performance will result and tip replacement maybe necessary.

4.2.3 Allow all components to air dry.

4.3 Paint Pump Cleanup

4.3.1 Remove the paint cans from the pump stand. Replace the paint lids.

! Warning !

Improper replacement of paint lids will cause premature drying and thickening of paint.

4.3.2 Place the end of the siphon hoses in a clean bucket of water. Gently scrub the siphon strainer and siphon pipe to remove paint and paint deposits. Replace water with clean water.

4.3.3 Close the main air inlet to the paint pumps. The inlet is closed when the handle is in the horizontal position. (Figure 2.7.1)

4.3.4 Close the paint pump outlet valves. Also, close the paint pump bleed valve. (Figure 2.7.2 and Figure 2.7.3)

4.3.5 Place the manual control switch in the 'ON' position and depress the 'PUMPS' switch.

4.3.6 Adjust the black and white pump air regulators, located on the Regulator Panel, to approximately 60 PSI (Figure 2.7.5)

4.3.7 Place a bucket at the end of the bleed valve hose.

4.3.8 Open the pump bleed valve. (Figure 2.7.2)

! Warning !

Exercise caution - pump may be under extremely high pressure. Secure hose to prevent 'hose whip' when pressure is relieved.

4.3.9 Open pump air inlet valve. (Figure 2.7.1)

4.3.10 Pump is fully clean when clear water flows from bleed valve hose.

4.3.11 Close inlet air valve. (Figure 2.7.1)

4.3.12 Close bleed valve. (Figure 2.7.2)

4.3.13 Repeat steps 4.3.7 through 4.3.12 for the other paint pump.

4.3.14 Depress 'PUMPS' switch to deactivate the pumps.

4.4 Cleaning the Paint Hoses

! Warning !

The water will flow from the gun outlet with considerable pressure. Skin damage may *result if contact is made with any body part.*

4.4.1 Lower the Gantry frame to a comfortable working height by actuating the 'UP/DOWN' switch on the Manual Control Panel.

4.4.2 Place a bucket underneath the gun outlets of the Gantry End Effector (Figure 2.8.1)

4.4.3 Open the black pump air inlet valve. (Figure 2.7.1)

4.4.4 Open the black paint outlet valves connected to the Gantry. There should be 2 black paint valves to open. Reference Figure 2.7.3 for the location.

4.4.5 Depress the 'GUN 1' switch to open the first black paint gun. With the 'GUN 1' switch depress, press the 'PUMPS' switch to activate the paint pump. (Figure 2.7.4)

4.4.6 Allow pump to run, until clean water runs freely from the gun outlet.

4.4.7 Depress the 'PUMPS' switch to deactivate the pump. Release the 'GUN 1' switch. (Figure 2.7.4)

4.4.8 Depress the 'GUN 3' switch. With the 'GUN 3' switch depress, press the 'PUMPS' switch. (Figure 2.7.4)

4.4.9 Allow pump to run, until clean water runs freely from the gun outlet.

4.4.10 Depress the 'PUMPS' switch to deactivate the pump. Release the 'GUN 3' switch. (Figure 2.7.4)

4.4.11 Close the black pump air inlet valve. (Figure 2.7.1)

4.4.12 Open the white pump air inlet valve. (Figure 2.7.1)

4.4.13 Open the white paint outlet valves connected to the Gantry. There should be 1 white paint valve to open. Reference Figure 2.7.3 for the location.

4.4.14 Depress the 'GUN 2' switch to open the first white paint gun. With the 'GUN 2' switch depress, press the 'PUMPS' switch to activate the paint pump. (Figure 2.7.4)

4.4.15 Allow pump to run, until clean water runs freely from the gun outlet.

4.4.16 Depress the 'PUMPS' switch to deactivate the pump. Release the 'GUN 2' switch. (Figure 2.7.4)

4.4.17 Close the white pump air inlet.

4.4.18 The Gantry paint lines are now clean.

4.4.19 Position a bucket underneath the paint gun outlets of the Linear Free Nozzle (LFN) system.

4.4.20 Depress the 'Pre. Guns' button on the manual control panel (Figure 2.7.4)

4.4.21 Depress the 'PUMPS' button. (Figure 2.7.4)

4.4.22 Open the pump air inlet valves for both paint pumps.

4.4.23 Open the black and white pump outlet valves. There should be 2 white outlet valves and 1 black outlet valve to open. Reference 2.7.3 for the location.

4.4.24 Locate the roller cam valve for each of the guns. (Figure 2.8.2) Depressing the cam valve will open the paint gun. Depress each one until clean water flows freely from the gun outlet.

4.4.25 Close the air inlet valve to both pumps.

4.4.26 Depress the 'PUMPS' switch to deactivate the pumps. Depress the 'Pre. Guns' to deactivate the LFN guns.

4.4.27 Open all paint line outlet valves. Relieve residual pressure in pump and lines by opening the bleed valves.

4.4.28 Close all paint line outlet valves. Close the bleed valves.

4.4.29 Remove the paint buckets from underneath the Gantry and LFN systems.

4.4.30 Using a wet brush, gently clean the ends of the paint guns to remove paint and other deposits.

4.5 System Power Down

4.5.1 Turn off the Gantry power supply, located on the bottom of the Control Cabinet. (Figure 2.6.1)

4.5.2 Turn off the Solenoid Interface boards, located on the top shelf of the Control Cabinet. (Figure 2.6.2)

4.5.3 Turn off the power transformer, located on the bottom of the Control Cabinet. (Figure 2.6.1)

4.5.4 Turn off the hand controller by pressing the left switch on the top of the unit.

4.5.5 Turn the power conditioner off.

4.5.6 Unplug all cords from the outlet of the generator. Turn the switch located near the spark plug to the OFF position. Close the fuel shutoff valve.

! Warning !

Failure to unplug all loads connected to generator may result in damage to electrical loads and/or damage to generator unit.

4.5.7 Close the main system outlet valve, located near the air compressor tank. (Figure 2.3.1) Stop the compressor by placing the key switch in the 'OFF' position. Close the fuel shutoff valve.

4.5.8 Disconnect all cables from the Hand Controller. Stow the cables and Hand Controller.

4.5.9 Replace all floor and wall panels. Close all doors.

4.5.10 System shutdown is complete.

5.0 Troubleshooting

5.1 Introduction

This section describes the troubleshooting procedures. For troubleshooting information pertaining to the electric generator and air compressor, please see the manufacturer's manual.

Symptom	Cause	Solution
The paint pump is making a 'chugging' sound.	The intake siphon screen is clogged.	Re-mix paint to remove solids and clean the intake siphon screen.
	Air is in the pump.	Re-prime the pump. See section 2.7.
	Air is in the paint lines.	Re-prime the paint lines. See section 2.8
	Paint level is low.	Replenish paint.
Paint pressure is inadequate.	The intake siphon screen is clogged.	Re-mix paint to remove solids and clean the intake siphon screen.
	Air is in the pump.	Re-prime the pump. See section 2.7.
	Air is in the paint lines.	Re-prime the paint lines. See section 2.8
	Paint level is low.	Replenish paint.
Spray pattern is too wide and paint looks watery.	Water is in the paint.	Run the test pattern a few times to flush the water from the lines. If pattern does not improve, the paint should be replaced. Re-prime the pump (section 2.7) and re-prime the paint lines (section 2.8).
Improper spray pattern.	Paint pressure is too low.	Adjust paint pressure.
	Nozzle is clogged.	Clean the nozzle.
Gantry does not respond. 'EP' is not displayed on gantry controller LED panel.	Gantry controller detected an operating fault.	Reset the gantry controller by pressing the 'RESET' button on the front panel.
		Power cycle the gantry controller and power cycle the hand controller.
Automatic modes unavailable.	Electronics/controller fault.	Contact Technical Support at (916) 752-4180.