

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

**TELEOPERATED AND AUTOMATED
MAINTENANCE EQUIPMENT ROBOTICS
(Phase I)**

Sonja Sun, Jim Mehlschau, Nelson Smith, Keith Kruetzfeldt
Paul Chen, Andrew Frank, Tsusuan Chang

AHMCT Research Report

UCD-SEP-94-09-30-01

UCD-ARR-94-09-30-01

Final Report of Contract

RTA-32J804

September 30, 1994

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

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Teleoperated and Automated Maintenance Equipment Robotics		5. Report Date September 1994	
		6. Performing Organization Code	
7. Author(s) Sonja Sun, Jim Mehlschau, Nelson Smith, Keith Krutzeldt Paul Chen, Andrew Frank, Tsusuan Chang		8. Performing Organization Report No.	
9. Performing Organization Name and Address University of California Davis Davis, California 95616		10. Work Unit No. (TRAIS)	
		11. Contract or Grant 32J804	
12. Sponsoring Agency Name and Address California Department of Transportation Sacramento, CA 94273		13. Type of Report and Period Covered Final, 1992 - 1994	
		14. Sponsoring Agency Code	
15. Supplementary Notes This project was performed in cooperation with the U. S. Department of Transportation, Federal Highway Administration			
16. Abstract In an attempt to eliminate the operator's exposure to the hazards associated with highway maintenance, University of California at Davis in conjunction with FMC under a grant from California Department of Transportation (CALTRANS) has developed a teleoperation system (remote control package) for a front-end loader. The radio remote control system developed enables the front-end loader to be operated remotely by current Caltrans workers performing a full range of functions at a distance up to 300 ft (91 meters). Operating modes can be easily shifted from remote mode to on-board mode or vice versa. The system has proven to be reliable, durable, safe, and sufficiently easy to operate. The performance of the remotely operated loader is equivalent or better than that of the conventional on-board manually operated machine. The feasibility of teleoperating an unmanned highway maintenance equipment has been well demonstrated. It is concluded that the remote control package is ready for commercialization and can be easily adapted to general purpose heavy equipment and other mobile machinery. Recommendations for further study include the integration of telepresence technology and other feedback systems into the remote control package along with capability for teachable automatic functions to enhance productivity of the front-end loader for both remote and on-board operation. Study of human factors related to teleoperation of highway maintenance vehicle equipped with feedback systems is also desired.			
17. Key Words Teleoperated, automated, maintenance, construction, operations		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 124	22. Price R/D

Executive Summary

In an attempt to eliminate the operator's exposure to the hazards associated with highway maintenance, University of California at Davis in conjunction with FMC under a grant from California Department of Transportation (CALTRANS) has developed a teleoperation system (remote control package) for a Case 621 front-end loader. The teleoperation of the loader is accomplished through an on-vehicle computer-based control system and an operator control unit via a secure RF communication link. Vehicle mobility and loader functions are actuated through the main interface and associated actuators. Two remote operator control units have been developed, a stationary on-truck workstation and a backpack/tummy pack style portable unit. The on-truck workstation is essentially a mock up of the CASE 621 loader cabin interior providing the operator a familiar operating environment. The portable control unit is supported via chest straps and tummy pack with all communication equipment carried in a backpack. This unique control unit was developed with emphasis on human factors and is very easy to operate.

The remote control package has built-in safety features including reliable communication, emergency stop and automatic failure stop. Several semi-automatic functions, such as return to dig, travel height, dump height, float and bucket shaking, were also implemented to speed up operations.

The radio remote control system developed enables the front-end loader to be operated remotely by current Caltrans workers performing a full range of functions at a distance up to 300 ft (91 meters). Operating modes can be easily shifted from remote mode to on-board mode or vice versa. The system has proven to be reliable, durable, safe, and sufficiently easy to operate. The overall performance of the remotely operated loader is equivalent or better than that of the conventional on-board manually operated machine.

The feasibility of teleoperating an unmanned high-way maintenance equipment has been well demonstrated. It is concluded that the remote control package is ready for commercialization and can be easily adapted to general purpose heavy equipment and other

mobile machinery.

Recommendations for further development include the integration of telepresence technology and other feedback systems into the remote control package along with the capability for teachable automatic functions to enhance productivity of the front-end loader for both remote and on-board operation. Study of human factors related to teleoperation of highway maintenance vehicle equipped with feedback systems is also needed.

The steering control interface could be improved by using proportional electrohydraulic controls in conjunction with an appropriate algorithm to eliminate "jerking" resulting from on/off solenoid valves. The stationary remote control unit could be further perfected by refining the foot pedals.

DISCLAIMER / DISCLOSURE

"The research reported herein was performed as part of the Advanced Highway Maintenance and Construction Technology (AHMCT) Center, within the Department of Mechanical and Aeronautical Engineering at the University of California, Davis and the Division of New Technology and Materials Research at the California Department of Transportation. It is evolutionary and voluntary. It is a cooperative venture of local, state and federal governments and universities."

"The contents of this report reflect the views of the author(s) who is (are) responsible for the facts and the accuracy of the data presented herein. 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."

TABLE OF CONTENTS

TECHNICAL REPORT STANDARD TITLE PAGE	i
EXECUTIVE SUMMARY	ii
DISCLAIMER / DISCLOSURE	iv
LIST OF FIGURES	viii
CHAPTER 1 - INTRODUCTION	
1.1 - Background	1
1.2 - Objectives	1
1.3 - Significance	2
1.4 - Hypothesis	4
1.5 - Research Approaches.....	4
CHAPTER 2 - SYSTEM REQUIREMENTS AND DESCRIPTION	
2.1 - System Requirements	5
2.2 - System Description	6
CHAPTER 3 - LCC/MACHINE INTERFACE AND REMOTE OPERATOR UNITS.....	
3.1 - LCC/Machine Interface	8
3.1.1 - Bucket Function	8
3.1.2 - Power Steering	9
3.1.3 - Brake Control.....	9
3.1.4 - Throttle Control	9
3.1.5 - Transmission	13
3.2 - Remote Operator Units	13
3.2.1 - Stationary Remote Control Workstation	14
3.2.1.1 - Concept	14
3.2.1.2 - Unit Description.....	15
3.2.1.3 - Design	16
3.2.2 - Portable Remote Operator Unit	20
3.2.2.1 - Design Requirements	20
3.2.2.2 - Development of The Unit	20
3.2.2.3 - Unit Description.....	21

CHAPTER 4 - REMOTE LOADER CONTROL SYSTEM	25
4.1 - Introduction	25
4.2 - Remote Control System	25
4.2.1 - Operator Interface	25
4.2.2 - Operator Control Computer (OCC)	28
4.3 - On-Board Control System	28
4.3.1 - System Description	28
4.3.2 - Loader Control Computer	31
4.3.3 - Solenoid Drivers	32
4.3.4 - Digital Input And Output Board	32
4.3.5 - Transmission Multiplexer	34
4.4 - RF Communication Link	35
4.4.1 - Theory of Spread Spectrum Communication Operation	35
4.4.2 - Unit Description	37
4.4.3 - Communication Protocol	38
4.5 - Software Design	38
CHAPTER 5 - TESTING AND RESULTS	46
5.1 - Introduction	46
5.2 - Subsystem Testing	46
5.3 - Overall Performance	50
CHAPTER 6 - CONCLUSION AND RECOMMENDATIONS	53
4.1 - Conclusion	53
5.1 - Recommendations for Further Study	53
REFERENCES	54
APPENDIX A1 OCC Connection Interface	55
APPENDIX A2 Pin Diagram of OCC	56
APPENDIX A3 Diagram of LCC Main Interface Board	57
APPENDIX A4 Diagram of LCC Front Panel Interface	60
APPENDIX A5 LCC Connection Interface	63
APPENDIX A6 Circuit Of Transmission Multiplexer	64
APPENDIX A7 Communication Protocol	66
APPENDIX A8 I/O Address and Bit Definitions for LCC	68

APPENDIX B1	Source Code for OCC	69
APPENDIX B2	Source Code for LCC	82
APPENDIX C1	Mechanical Drawings For Portable Unit	103
APPENDIX C2	Mechanical Drawings For Stationary Unit	114
APPENDIX D	Operating Instruction	120
APPENDIX E	Summary Of Survey On Remote Operation By Caltrans Workers	123

LIST OF FIGURES

Figure 2.1	Block diagram of the remote control loader system	7
Figure 3.1	Hydraulic circuit of LCC/machine interface	10
Figure 3.2	Pneumatic circuit of brake system interface	11
Figure 3.3	Hydraulic interface module	12
Figure 3.4	Brake system pneumatic interface module	12
Figure 3.5	Layout of stationary remote operator unit.....	15
Figure 3.6	Photo of the stationary remote operator unit	19
Figure 3.7	Photo of the portable remote operator unit	19
Figure 3.8	Schematic diagram of the portable remote operator unit	22
Figure 4.1	Block diagram of remote control system	27
Figure 4.2	Block diagram of the Operator Control Computer (OCC)	28
Figure 4.3	Block diagram of the on-board loader control system	30
Figure 4.4	Block diagram of the Loader Control Computer (LCC)	31
Figure 4.5	Photo of the LCC	32
Figure 4.6	SN7404-N hex buffer integrated circuit.....	33
Figure 4.7	Sample circuit of attenuators.....	33
Figure 4.8	Basic circuit of transmission multiplexer	34
Figure 4.9	An overview of the modem's topology	37
Figure 4.10	Software flowchart for OCC	40
Figure 4.11	Software flowchart for LCC	41
Figure 5.1	The front-end loader operated with the portable remote control unit	51
Figure 5.2	The front-end loader operated with the stationary remote control unit	51

Chapter 1

Introduction

1.1 BACKGROUND

The use of heavy equipment, such as crawler tractors, dozers, and loaders, is crucial to highway maintenance. These machines are often used in hazardous situations. For example, they are used to repair highway damage after an avalanche, to clear land slides, to clean up hazardous materials, to clear snow on winter storm-closed highways, etc. In such operations where the soil and snow banks are unstable, road boundaries are often not recognizable, and the toxicity of spills is not known, the operators are working in a high risk condition.

Significant hazards exist for which the enclosed cab devices and other preventive measures may offer little operator protection. Casualties and injuries occur every year across the nation, despite preventive efforts. The need to remove the operator from the machine has been made evident by such hazards inherent in highway maintenance.

In an attempt to eliminate the operator's exposure to the hazards associated with highway maintenance and to improve productivity, University of California at Davis in conjunction with FMC under a grant from California Department of Transportation has developed a teleoperation system (remote control package) for a front-end loader. The remote control package enables the operator to leave the loader promptly with the portable control unit to a safe site or to the on-truck workstation to operate the loader remotely.

1.2 OBJECTIVES

The ultimate goal of this project is to eliminate the need for Caltrans machine operators to be exposed to hazardous operating conditions and to increase operation speed and productivity in these situations. The concept allows the equipment to be operated normally by the human operator in the cab, but when the vehicle is in a dangerous

situation such as in a landslide in mountainous terrain, the operator can leave the cab and perform his duty safely away from the machine by teleoperation.

The objectives are:

1. To design and build a remote control system for an existing Caltrans front end loader.
2. To test the machine to evaluate its capabilities and limitations in teleoperation.
3. To determine ways to improve teleoperation performance of the machine.
4. To develop techniques for increasing its capabilities beyond existing limitations.

1.3 SIGNIFICANCE

- Safety

The primary motivation for implementing this remote control loader is to improve the safety of loader operation. The normal on-board operation of the loader exposes the operator to serious level of hazards. The main hazards are from three sources, namely operation in unstable soil or snow banks, handling of hazardous material spills and collision with other vehicles. Remote control of the loader permits the operator to be at a safe location with respect to these hazards.

- Reduced Operator Fatigue

A benefit which can be expected from remote control of the loader is reduced operator fatigue. In normal operation the operator is subjected to a significant noise, shock and vibration environment. The remote operator will be able to perform his/her duties from a position which can be maintained in a much more comfortable condition. This reduced operator fatigue will also contribute to the overall safety of operation.

- Higher Production Rates

The production rate achieved with the remote controlled loader can be expected to be higher than with the operator on the machine. The fact that the operator will not

experience the shock and vibration environment of the machine will permit operation of the machine at the limits of achievable performance without regard to personal comfort. Since the operator is removed from the machine and located in a spot where he can see the entire operating field, he no longer has to turn around when backing up, reducing operational time further. The implementation of automatic functions and teachable programs can also enhance the production rate.

- Cost Savings

Cost savings can be expected from two different aspects of remote control, the elimination of some operator related hardware costs and the increased production rates anticipated. Increased production rates will result in cost savings because the time to complete a given job will be reduced. The main candidate for hardware cost reduction is the requirement for a roll cage and an air conditioned cab. The cage may not be necessary if manned operation of the machine is limited to transportation.

- Technical Improvement

Development of a remote control system for the loader will permit evaluation of loader performance without the operator in the machine. The operator may be compensating for performance limitations such as slowing travel in response to limited suspension. This has the potential for increasing understanding of the actual performance of the loader. It is conceivable that this experiment will disclose avenues for development which can be used to improve the performance of loaders in general.

- Improved Quality of Life

The net result of all these benefits will be an improved quality of life. The operator will enjoy a safer and more pleasant work environment and the public will receive a faster and more economical response to highway maintenance requirements.

- General Application

The teleoperated front-end loader is expected to be used in highway maintenance immediately. The radio remote control technology developed could be applied to various

equipment for other operations.

1.4 HYPOTHESIS

The technical hypothesis prior to this project is that it is feasible to modify existing Caltrans equipment for teleoperation. We developed a remote control package for a CASE 621 front-end loader. We anticipate that such a system can also be adapted to other existing machines, such as crawler tractors, dozers, shadow trucks, mowers, etc., for highway maintenance, particularly when situations are hazardous to workers.

It is also assumed that it is desired to construct a system by using as many commercially available components as possible. New components will be developed as needed.

1.5 RESEARCH APPROACHES

This project was carried out with a close collaboration with FMC. FMC has been involved in remote control and teleoperation of earth vehicles since mid 1960s and has been most active in robotics vehicle technology development over the last 10 years. FMC has over 30 years of unmanned vehicle experience. The militarily derived radio remote control kits and operator control units that FMC has developed for the US Army are compatible to our system and can be readily integrated to the front-end loader.

The FMC and UC Davis engineering responsibilities were defined to eliminate overlap of effort. The two groups cooperated in establishing requirements for the project and reviews of the project progress. After overall requirements are established, FMC was responsible for development of the communication and control system. UC Davis was responsible for design and development of the necessary actuators and operating units, and evaluation of the performance of the various subsystems as a whole. The remote control system was modified at the last stage by UC Davis to improve reliability, expandability and overall performance.

Chapter 2

System Requirements and Description

2.1 SYSTEM REQUIREMENTS

The purpose of the project is to apply radio remote control technology to heavy equipment to reduce operator risks. The task is to design and construct a remote control package and implement it to a full scale heavy duty mobile machine. The machine can then be teleoperated with safeguards and operational capabilities. The requirements that were established by the project team include the following:

- The remote control system shall be able to control a full scale front-end loader with all operations, including:

1. Vehicle mobile controls:

- a. Engine speed
- b. Vehicle steering control
- c. Transmission control -- direction (forward, neutral, and reverse) and gears (1,2 and 3)
- d. Brake control

2. Operation of the front-end loader bucket:

- a. Boom (arm) motion -- up and down
- b. Bucket motion -- tilt and roll back

- The control system shall be easily installed or removed from the vehicle.
- The control system shall be simple to operate and can be operated by Caltrans workers.
- It shall have built-in safety features, including reliable communication, emergency stop and automatic stop, when communication is severed.
- The teleoperated front-end loader can be remotely operated by an operator at a distance of up to 300 feet (91 meters).

2.2 SYSTEM DESCRIPTION

Figure 2.1 is a block diagram of the Remote Control Loader System which consists of five major subsystems: 1) Remote Operator Unit (ROU), 2) Operator Control Computer (OCC), 3) Radio Frequency Data Link (RFDL), 4) Loader Control Computer (LCC) and 5) Actuator and sensor interface. The system provides control of the Case 621 Loader mobility and bucket functions from the remote (ROU). The mobility and loader control is performed through electrohydraulic, electropneumatic, electrical and digital logic interface controlled by an on-board (LCC). The LCC communicates over a serial RS-232C/RF link with the OCC to receive operator commands from the ROU.

A Case 621 articulated front-end loader with a 2.5 yard (2.3 meter) bucket was used as the remote controlled equipment. The loader is a diesel engine-powered, hydraulically-operated, four-wheel drive articulated front-end loader that has built-in options for automatic travel and dump height control and automatic return to dig.

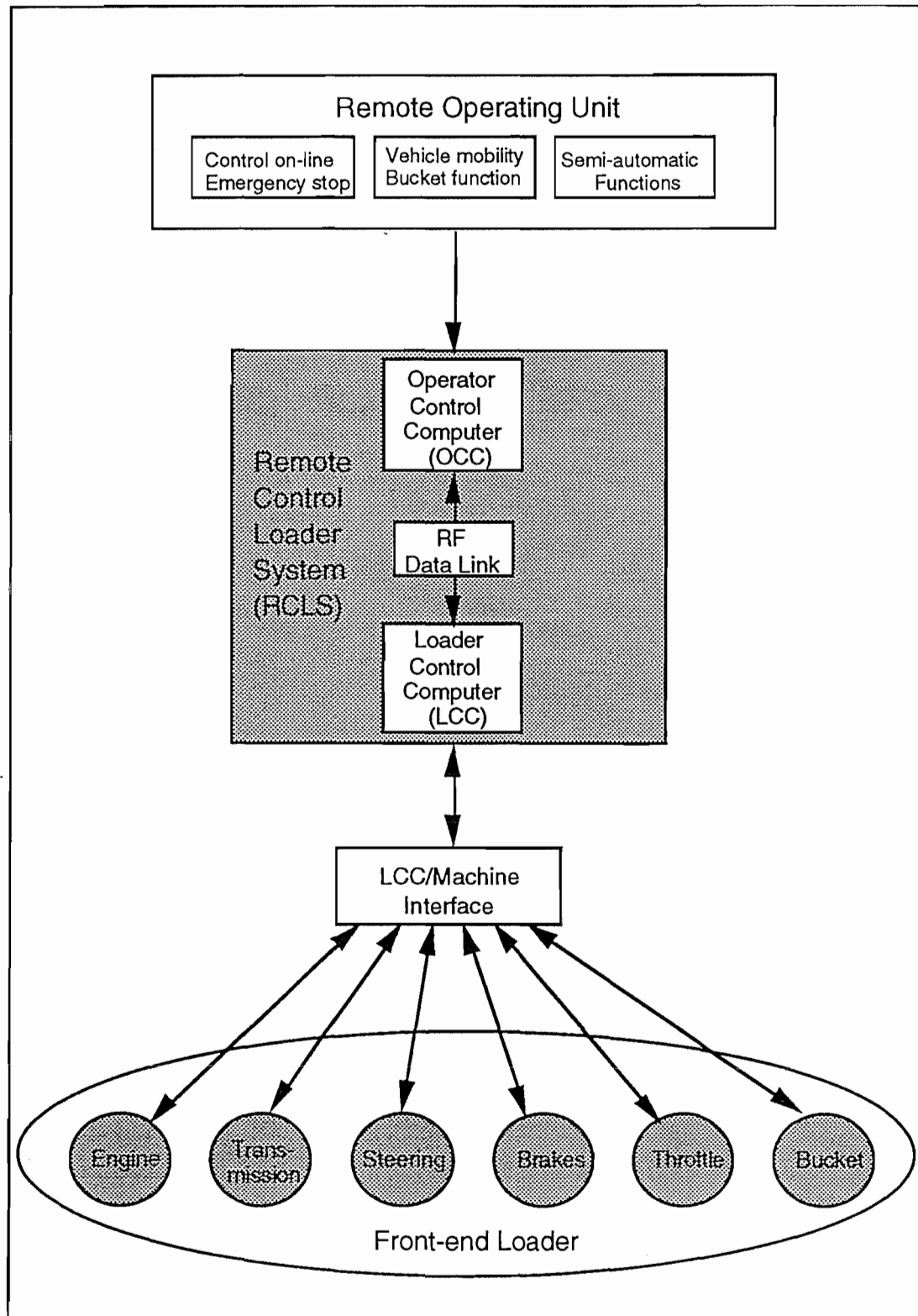


Figure 2.1 Block diagram of the Remote Control Loader System

Chapter 3

LCC/Machine Interface and Remote Operator Units

3.1 LCC/MACHINE INTERFACE

Teleoperation of the front-end loader is implemented through hydraulic, pneumatic, electrical and electronic interfaces controlled by the on-board Loader Control Computer (LCC). When in on-board manual mode, the vehicle mobility and bucket actions are controlled by the on-board stock/existing manipulators, such as steering wheel, brake pedal, loader joysticks, gear shift control, etc., which direct the existing hydraulic, pneumatic, electrical and electronic systems. Interfacing with the pilot operated hydraulic bucket controls, steering and brakes was done with pressure regulators in tandem with solenoid operated control valves. These control valves are activated by relays managed by the LCC. The hydraulic and pneumatic circuits with the main interface components are shown in Figure 3.1 and 3.2, respectively. The added components for remote control interfacing are indicated in gray lines with the black lines representing the existing circuits on the CASE 621 front-end loader. These components were integrated into two modules: a hydraulic module and a pneumatic module as shown in Figure 3.3 and 3.4, respectively. The two modules were mounted at the right side of the loader outside the cabin.

The throttle activation was by an off-the-shelf electrical proportional linear actuator that receives a 0 - 5 volts input from the LCC. The transmission used logic level electronic controls and was directed at this level by the LCC. Status of loader position sensing switches were sensed directly by the LCC.

3.1.1 Bucket function.

Pressure reducing valves were used to control hydraulic pilot pressures managing actuator directional control valves and thus provide speed adjustment for each bucket function. Three-way solenoid valves were used in series with the existing controls to

activate individual bucket motions by selecting a preset pilot pressure (ON mode -- for remote) or the existing manual control (OFF mode). Bucket lowering was arranged with two speeds to accommodate lowering an empty or loaded bucket.

3.1.2 Power steering.

Remote steering was controlled by a four-way solenoid operated hydraulic valve with an adjustable flow control, both were installed in series with the existing power steering unit oil supply. Actuation of the steering cylinders was connected in parallel with the existing system. Testing revealed that this approach could operate with variable flows and did not need the flow control. A rotary potentiometer was used to sense steering angle and provide the LCC with a corresponding variable voltage.

3.1.3 Brake control.

The air over hydraulic brake system was controlled much the same as the bucket control. Pressure regulators were used to provide air pressures representing three levels of braking and three-way solenoid-operated valves were used to select the braking level. One solenoid valve was installed in series with the existing brake control valve to select between the existing system (power off) and the first level of preset braking. Two additional solenoid valves directed the second and third pressure levels to the same valve port receiving the first level. The LCC activated relays based on instruction from the ROU to power up appropriate valves.

3.1.4 Throttle control.

Remote throttle control was first accomplished using two air cylinders of differing and adjustable strokes with spring return, connected together in series and operated by a pair of three-way solenoid-controlled air valves. The air supply was from a pressure regulator connected to the brake air supply system. This approach provided three throttle settings above idle and was equipped with a mechanical quick release on the actuator anchor. Satisfactory performance was observed from the three position, two air cylinder-

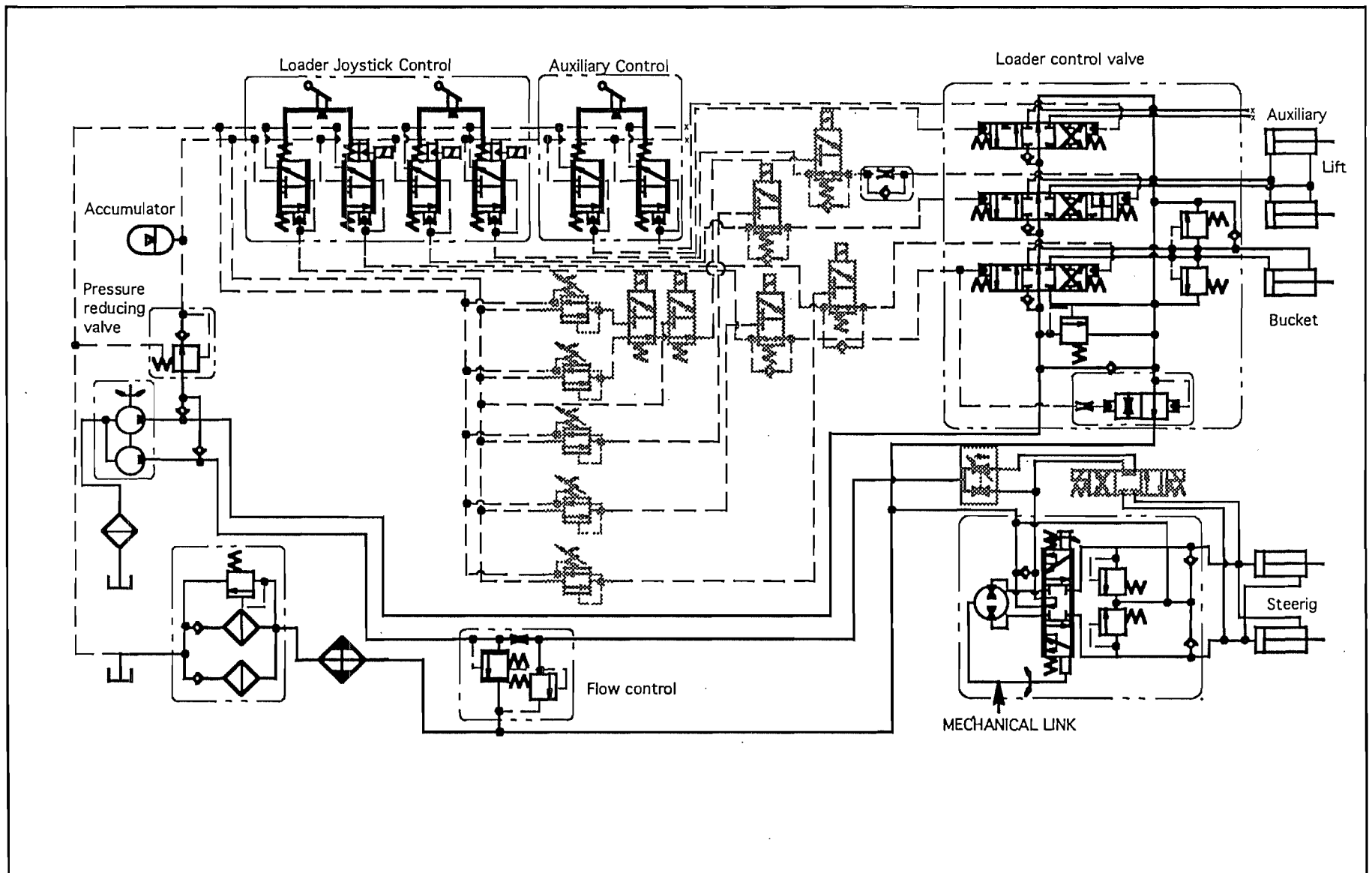


Figure 3.1. Hydraulic circuit of LCC/machine interface

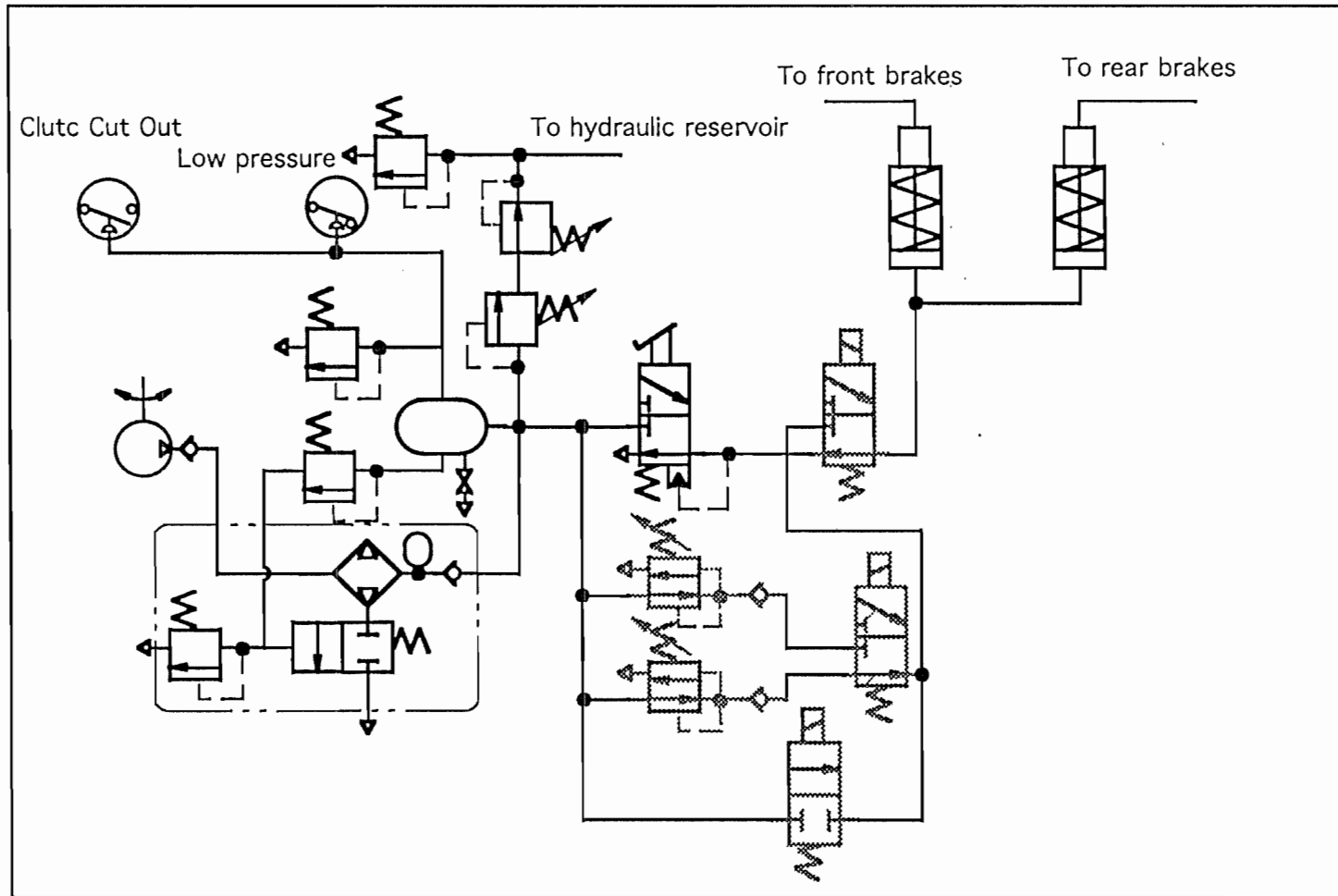


Figure 3.2 Pneumatic circuit of brake system interface

activated three speed control system. Settings of the step approach were actually quite well positioned for loader operation. However, the step change in engine RPM was unnerving, suggesting proportional throttle control would be desirable.

In the final design, an ADCO ERC (electronic remote control) system consisting of an actuator and control module was selected. The ERC system is a fully proportional linear actuator with a closed loop servo control system. The electronic controller receives a command input and sends it to the actuator whose position change closes the loop. Any change in command signal will result in a corresponding change in actuator position. Thus ERC provides fast, smooth and precise control of this remote function. The unit is free wheeling when power is off allowing rapid return to neutral as well as manual over-ride. Even though the actuator offered free wheeling capability, a linkage was incorporated in the installation to allow manual over-ride without needing to move the actuator. Testing following installation of the ADCO unit demonstrated the benefit of proportional control for improved productivity, as well as an operator confidence builder.

3.1.5 Transmission.

Transmission control, including direction and gear selection, was done electrically by providing the existing electronic transmission control module with a multiplexer.

3.2 REMOTE OPERATOR UNITS

The primary motivation for implementing the remote control system to the front-end loader is to improve the safety of the operation. However, when operating the loader remotely it is expected that the same loader performance and productivity, if not better and higher, to be achieved as from in-vehicle operation. Therefore, it is essential to provide the operator a 'friendly' remote operating environment by considering ergonomics and presenting a comfortable operating station that can be easily and quickly adapted by skilled loader operators.

Two remote operator units have been designed and developed. One is a portable

backpack/tummy pack unit supported via shoulder and chest strips with all communications equipment carried in the backpack. The other is an on-truck stationary control station, which is essentially a mock up of the cabin of the Case 621 front-end loader. All controls are located and operated in a manner such that an in-vehicle operator would find all remote controls very familiar. The main advantage of the portable unit is that it is lightweight, compact and portable. The unit can be stored on the loader, and when a hazardous situation is encountered, the operator can easily carry the portable unit leaving the vehicle and resuming the operations from a safe site. This portable unit also allows the operator to walk around to adjust his position for a better view of the loader during the remote operation. The on-truck work station enables the operator to adapt the new working unit slightly faster by being situated in a familiar environment and allows the operator a comfortable and environmental controlled space for long term operation. The station provides a better view during the remote operation due the elevation of the station base which is mounted on a pick-up bed. However, it requires the use of a pick-up truck and an additional driver to drive it to the site.

Both units share the same Operator Control Computer (OCC), which is housed in a backpack. Potentiometers are used on both units to sense positions of steering, braking and throttle, and were calibrated in such a way that the stationary remote control workstation and the portable operating unit are exchangeable as far as the OCC and the corresponding software are concerned.

3.2.1 Stationary remote control workstation

3.2.1.1 Concept

The primary reason for the remote operation of the end loader is to place the operator in a safe environment when dangerous operating conditions occur. However, it is necessary for the operator to be confident in his actions when operating remotely. Thus, the stationary remote workstation has been designed to present the operator with familiar operating environment. The stationary workstation is also configured to allow

for an easy transition between in-vehicle and remote operations while simultaneously acting as a test bed for systems to make teleoperation as feasible as possible.

3.2.1.2 Unit Description

The stationary workstation is essentially a duplication of the interior of a CASE 621 front-end loader vehicle cabin. All controls are located and operated in a manner that a in-vehicle operator would find very familiar (see Figure 3.5 for its layout). There is a standard steering wheel, brake and throttle pedals, a column mounted gear and direction selector, and a two-axis joystick for arm and bucket control. In addition to these controls there is also a control panel for repetitive arm and bucket functions. The panel features include; return to dig, travel height, dump height, float and bucket jog functions. The return to dig function allows the operator to return the bucket to an optimum position

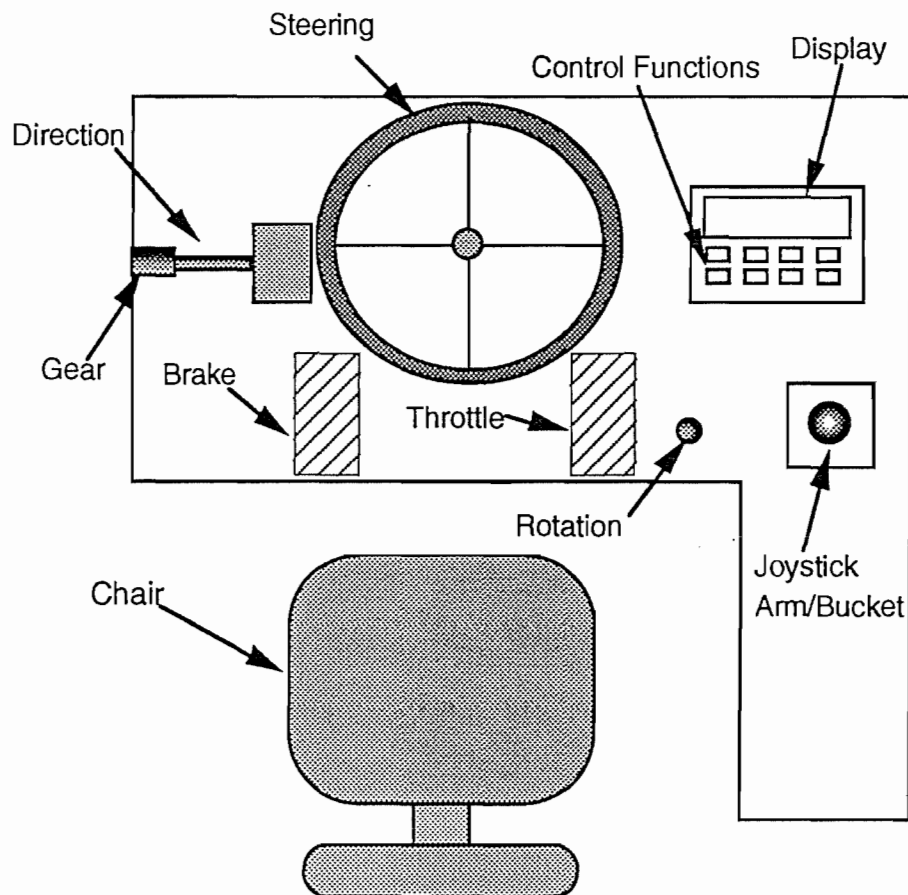


Figure 3.5 Layout of Stationary Remote Operator Unit.

parallel to and just clearing the ground, ready to drop and load when the pile is approached. The bucket jog feature allows the operator to shake the bucket via push-button to level bucket loads preventing spill during load transport. The travel and dump height functions allow the bucket to quickly reach the heights determined by the existing built-in limit switches. The float function puts the bucket in neutral position for leveling or scraping.

The stationary workstation is mounted to a rotating base with a DC gear motor for position control. This configuration allows the station to be rotated via a toggle switch located on the control panel, which enables the operator to maintain a direct line of sight with the vehicle (rotational travel is full 360 degrees). The operator's seat is also fully adjustable to accommodate a full range of users.

The stationary unit was designed to fit in the bed of a mid-sized pick-up truck for ease of transportation. The overall dimension of the station is 48" x 48" x 45" (122 cm x 122 cm x 114 cm). The station's base is similar in design to a loading pallet so the workstation can easily be lifted onto a truck bed with a small forklift (see Figure 3.6). The truck battery system can be used to power the rotating base and additional upcoming functions. Use of the truck's battery allows for a fully independent remote operating package.

3.2.1.3 Design

1. **Steering:** The steering system uses a common automotive steering wheel bolted to a bearing supported shaft that drives a rotary potentiometer. Steering displacement is approximately 3/4 turn lock to lock as found in the loader. Steering wheel position is converted to a voltage signal by the rotary potentiometer and is then converted to a digital signal, transmitted over RF modem, and then converted back to analog as are all input voltage signals from the workstation. Since the stationary workstation uses the same Operator Control Computer as the portable unit, trim pots are placed on the rotary pot to match voltage signal ranges between the two units (this is also

done for all the other input voltage signals which must be matched).

The added remote steering system on the front-end loader is position controlled and not rate proportional. For simplicity, the steering tracks at a constant angular velocity until the position indicator on the front-end loader matches the position indicated by the rotary potentiometer on the operator units (both stationary and portable). This may seem awkward at first to an operator but is easily compensated for with little practice.

2. Brake: The brake consists of a simple spring return lever type pedal coupled directly to a rotary potentiometer. Although encoding at the station is full analog the brake actuators on the end loader are stepped. The voltage signal is compared to three levels, i.e., light, medium and full brake. This method is simple and low cost while proving to be very adequate for operators.

3. Throttle: The throttle acts exactly the same as the brake except that the output signal is transmitted in full analog and the control is proportional. This gives the operator complete modulation of engine power to control wheel slippage and hydraulic power.

4. Gear/Direction Select: The transmission control utilizes a column mounted forward/neutral/reverse lever and an end mounted rotary 1-2-3 gear position selector, exactly as found on the front-end loader. As the positions are all on/off, control is digital and sent to the end loader as such.

5. Arm/Bucket Control: A two axis analog joystick is used. Forward/backward controls the arm and left/right controls the bucket as found in the front-end loader. Although the joystick is analog, behavior is digital as there is no rate control on the actuators, only on/off. Rate can be modulated somewhat by using engine throttle to increase hydraulic flow and to increase bucket and arm speeds.

6. Control Box On-line: The control box on-line is a toggle switch which is actuated when the operator decides to take control of the vehicle remotely. In order to activate the control box, however, the transmission must be in neutral, brake fully applied

and communications between OCC and LCC must be fully established. Control box on-line indicator is on when the operator presses this switch and communications between OCC and LCC is established.

7. Emergency Stop: This red-colored push button is used to stop the vehicle upon entering a hazardous condition. When this control is actuated (the button is pushed), the Vehicle Control Computer will perform the following functions: fully apply the brake, set the throttle to zero, shift transmission to neutral and stop any arm or bucket movements.

8. Automatic functions: All functions are push-buttons. A digital signal is sent to the control computer, which implements desired arm/bucket control commands by interfacing with arm/bucket control at the front-end loader. The functions implemented in this phase of the project include return to dig, travel height, dump height, bucket rattle and bucket float.

9. Station: The workstation is constructed of aluminum, 1/16" (0.16 cm) thick, to keep weight down and to make modifications easy. Two constraints determined panel sizing: similarity to front-end loader and the need to be able to rotate within a truck bed. Thus, the station was made as large as could possibly fit in a mid-sized truck (48"/122 cm clearance between wheel wells). In order to maximize sizing, it was necessary to keep the station's center of gravity offset to the center of rotation. The increased loads on the system due to the offset was compensated for by using an oversized gear motor, oversized tapered bearings for support, and 0.125" (0.32 cm) thick, 1" by 1" (2.54 cm x 2.54 cm) square steel tubing for the rotational assembly. The oversizing also allows for the addition of more equipment in the future. The entire assembly is mounted to a rugged 4'x 4' (1.22 m x 1.22 m) square pallet-like base for stability and durability and also allows for transport via a forklift.

10. Station Rotation: Rotation of the remote control station is necessary to align the operator with the direction of the front of the vehicle. This gives the operator a sense of which way he is going. The rotational position of the workstation is controlled by a

simple on/off dual position toggle switch to control rotation direction. Rotational velocity is fixed. A high torque geared 12V Brushless DC motor provides power. The bearing support is over designed to allow for heavy loads and the addition of more equipment.

3.2.2 Portable Remote Operator Unit

3.2.2.1 Design Requirements

When an operator operates the front-end loader remotely with the portable operating unit, his/her working environment changes substantially. Instead of sitting on the set operating all the manipulators in the vehicle cab, he/she has to carry and support the operating unit and operating control computer during the entire remote operation. With the portable unit, he/she can no longer use his/her feet to control the brake and throttle as he/she used to during on-board operation. In order to achieve good performance and efficiency of the teleoperated loader, the following design requirements have to be met:

1. The unit shall be compact and lightweight.
2. All functions shall be performed by two hands and all operating items shall be easily handled with easy reach.
3. All actions taken shall be clear without confusion, conflict or interaction with one another.

3.2.2.2 Development of the unit

Development of a "friendly" remote operator station involved fabricating, evaluating and modifying several variations. The initial unit used standard toggle switches for gear selection, forward/reverse, an adjacent pair of lever operated rotary switches for throttle and brakes, a lighted push button to activate the remote, a three-position rotary switch for steering and a two-axis joystick for bucket controls. These input devices were mounted on the cover of a 6" x 7-7/8" x 4" (15 cm x 20 cm x 10 cm) aluminum electrical enclosure (control box), which was carried on a pair of curved light-weight metal tubes that hooked over the operator's shoulders and held the box such that

the operator's lower arm was nearly horizontal when accessing the controls. This initial control box, while very crude, permitted building confidence in the bucket, brake and steering interfaces. For the second generation, the box was arranged to rotate and result in steering switch activation. The resulting improvement in steering the loader helped with the decision to concentrate efforts on further refinement of the operator interface. The next generations involved mounting the box for tilting fore and aft along with the rotation. Both the throttle and forward/reverse functions were associated with the tilting mode at different times. An addition to the tilting and turning concept was motorcycle like handle bars with a grip lever on the left for brakes. For the next variation, the left handle bar was modified to offer a twist operated throttle in addition to the brake grip. Forward/reverse remained with the tilt action. While each variation offered improvement to the ease of operator interaction, a common weakness crept in when one activity needed to be momentarily maintained while varying another. A common result was the accidental operation of multiple functions often undoing a previous objective. Frequently the steering valve would remain activated while addressing a throttle need then require reversing the steering to get back on course. Since mobility of the control box seemed to surface as the weak link, including the feel of operating the bucket joystick, the next concept removed the tilt and turning of the control box rather than attempt to employ detents or spring rates to isolate functions.

3.2.2.3 Unit Description and Design Features

The final version of the portable operating unit, as shown in Figure 3.7, consists of three components: a support frame, the left sub-unit and the right sub-unit. Figure 3.8 is a schematic diagram of the control panel of this unit.

The left sub-unit is mainly a small quarter section "steering wheel" with two matching curved grip bars. The grip bar below the steering wheel is used to activate the brakes and the one above and in-board of the steering wheel rim is for the throttle control, which can be actuated by the thumb. Both grip bars have spring return and are mounted on the

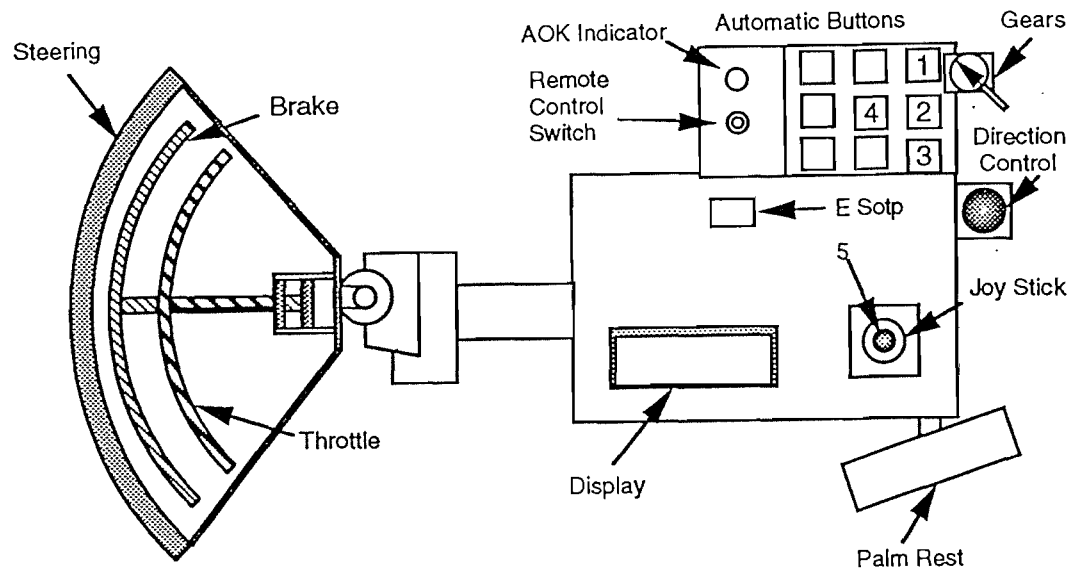


Figure 3.8 Schematic Diagram of the Portable Remote Operator Unit

steering wheel to follow the rotation of the steering wheel. The steering displacement is approximately 90° locked by two stops on the steering journal. Rotary potentiometers are attached at all hinge points to detect position changes.

The right sub-unit is a control box based unit that hosts the rest of the remote operating items. The control box is a 7.5" x 4" x 2.4" (19 cm x 10 cm x 6 cm) aluminum electrical enclosure on which there are a dual-axis joystick, a forward/reverse lever, an emergency stop button and a display window. A 4.75" x 2.5" x 1" (12 cm x 6.3 cm x 2.5 cm) aluminum electrical enclosure was attached to the control box to hold a gear selection switch, an auto mode toggle switch, a control box on-line indicator and an array of push buttons for automatic functions. This box was inclined at 45° for better viewing and easy controlling. A smooth curved tube-block provides the operator a steady arm rest for operating the bucket joystick and the forward/reverse lever. Since the joystick and the forward/reverse are, respectively, the first and the second most frequent used

forward/reverse are, respectively, the first and the second most frequent used manipulators, they were considered the primary items on the right panel and arranged closest to the palm rest. The gear switch, the function buttons, the mode switch and the indicator were considered the secondary items and were set farther. The emergency stop is placed away from the frequently used items, thereby preventing accidental activation. All the items can be easily handled and within easy reach (the farthest item is 7" (17.8 cm) from the palm rest).

Both the left and right sub-units are mounted on the support frame which is hooked to the shoulder strips from the backpack hosting the OCC. The operating unit can be adjusted to a desired position such that the operator's lower arm is nearly horizontal during the operation. A curved bar wrapped with sponge which is against the operator's stomach to further secure the unit can also be adjusted to make a comfortable fit according to the operator's physical size.

This version of the remote control unit has several design features. All control tasks are divided reasonably in to two groups for the left and right hands according to the nature of the tasks and human operator's capability, so they can be carried out without confusion or interference. Three-curved-bar design enables the operator to control steering, throttle and brake simultaneously using only the left hand. The operator can, while resting his left hand on the steering wheel, steer the vehicle with his palm, grab the throttle bar with his thumb and apply the brake with the rest of his fingers. An additional friction on steering function helps stabilize and, consequently, isolates the three functions. This control arrangement resulted in smooth interaction with the operator. The field test has shown that the operators are comfortable to operate the loader bucket while applying the throttle and brake at same time as needed. The unit weighs 9.5 lb (4.3 kg) with the overall dimension of 19.75" x 12.5" x 12" (50 cm x 31.8 cm x 30.5 cm). It is compact and can be easily stored in the cab of the front-end loader, and easily removed by the operator when remote operation is desired.

The electronic controls are the same as on the stationary workstation described in Section 3.2.1.3. Five automatic functions have been implemented. Except for the rattle function, which is activated through the push button at the center of the joystick handle, the other four functions, i.e., Return to Dig, Travel Height, Dump Height and float, are activated via the push buttons on the button array.

Chapter 4

Remote Control System

4.1 INTRODUCTION

The Remote Control Loader System (RCLS) for providing remote control of a Case 621 front-end loader includes the Operator Control Computer (OCC), the RF Data Link (RFDL) and the Loader Control Computer (LCC) (refer to Figure 2.1). The RCLS takes input from the Remote Operator Unit (ROU) and controls the front-end loader via a set of actuators. The system is based on the package designed and built by FMC Corporation. The hardware and software were a modified version of a system which was developed for remote control of military vehicles. The FMC unit worked reasonably well at providing basic control functions. It was determined, however, that in order to provide the flexibility and variety of control functions required by this project, an extensive redesign of the hardware and software for LCC would be necessary. The loader control unit described in this report is the result of this effort. The CPU and I/O boards from the FMC system were retained, but additional analog and digital interface circuitry was added as well as solid state relays for driving the various solenoids and actuators. The unit was then packaged in a smaller case so that it would fit behind the seat of the Case 621 front-end loader. Then all new software was written using standard C language. This provided transportable code which could be easily modified, as opposed to the code written for the non-standard computer language used by FMC.

4.2 THE REMOTE CONTROL SYSTEM

4.2.1 Remote Operator Interface

The Remote Loader Control System allows an operator to remotely control the front-end loader through a Remote Operator Unit. The controls include (refer to the system block diagram in Figure 4.1): Control Box On-line, Emergency Stop, Steering,

Throttle, Brake, Transmission-direction (Forward, Reverse) and Gear (1, 2 and 3), Arm, Bucket, and Semi-automatic functions that include return to dig, travel height, dump height, float and rattle.

The Remote Operator Unit also provides Control Box On-line Indicator Light.

1. Steering: Steering input is an analog signal. The voltage of this signal corresponds to the desired steering angle of the loader. The voltage can vary from 0 volt (full left) to +2.5 volts (full right).

2. Throttle: This control is used to set the throttle control of the bucket loader. Throttle control uses a proportional control device, with both its input and output are analog.

3. Brake: The brake input to OCC is an analog signal. This analog signal is sent to the LCC where it is converted to a four-level digital output that drives the braking solenoids.

4. Transmission: Transmission is a digital control and made up of two separate controls:

- Travel: This control is a set of two signals, forward and reverse. It specifies the direction of travel of the front-end loader. Neutral is obtained when both forward and reverse are set to zero.

- Gear: This control sets the gear selection of transmission. The range for this control is 1, 2 and 3, two signals used to implement gear selection.

5. Arm: Arm input is an analog signal. This signal normally has a value that corresponds to no arm motion. Anything below this value will lower the arm and anything above it will lift.

6. Bucket: Bucket control is used to roll the front-end loader bucket forward or backward. Bucket input is an analog signal. This control normally has a value which corresponds to no bucket motion. Anything below this value, will roll back the bucket and anything above it will roll it forward.

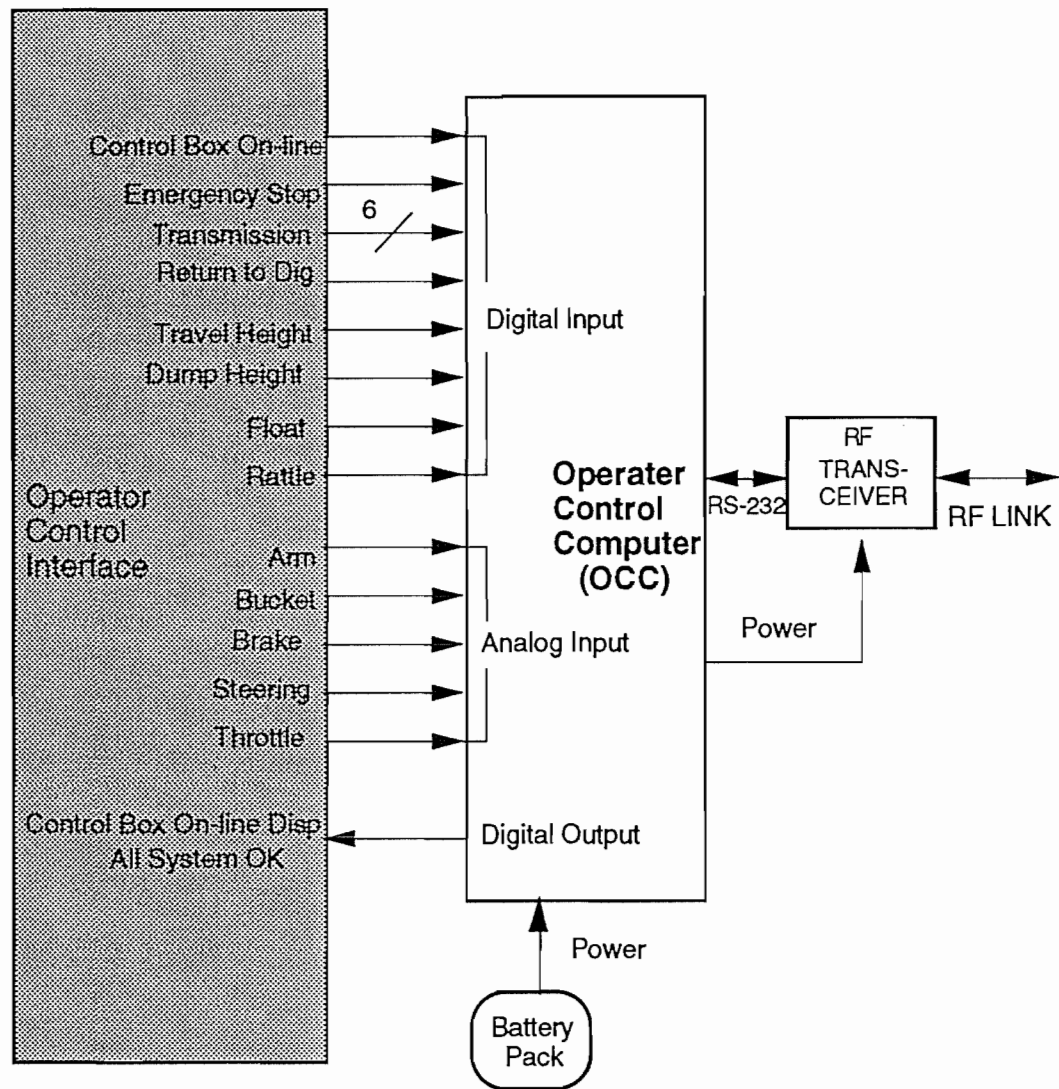


Figure 4.1 Block diagram of remote control system

7. **Control Box On-line:** Control box on-line has a digital input and a digital output. When the Control On-line display is lit, all the controls on the operator control unit are active. In this state, the controls could be used to remotely operate the bucket. This display also reflects the state of RF communications between OCC and LCC. When

communications is re-established, this display would be turned on again.

4.2.2 Operator Control Computer (OCC)

The Operator Control Computer is essentially a single board C-Programmable miniature controller (Tiny Giant from Zworld) with an RS-232C serial port and a power supply. Figure 4.1 is a block diagram of the Operator Control Computer. The unit is mounted in a 18" x 13.75" x 2" (46 cm x 32 cm x 5 cm) box-frame made of 1/16" (0.16 cm) thick metal sheet and packaged into a backpack. The box-frame also holds an RF transmitter used to communicate the operator commands from the OCC to the LCC. The OCC takes input from the remote operator via the Remote Operator Unit through a 25 pin cable. The OCC connector interface and the pin assignment are included in Appendese A1 and A2.

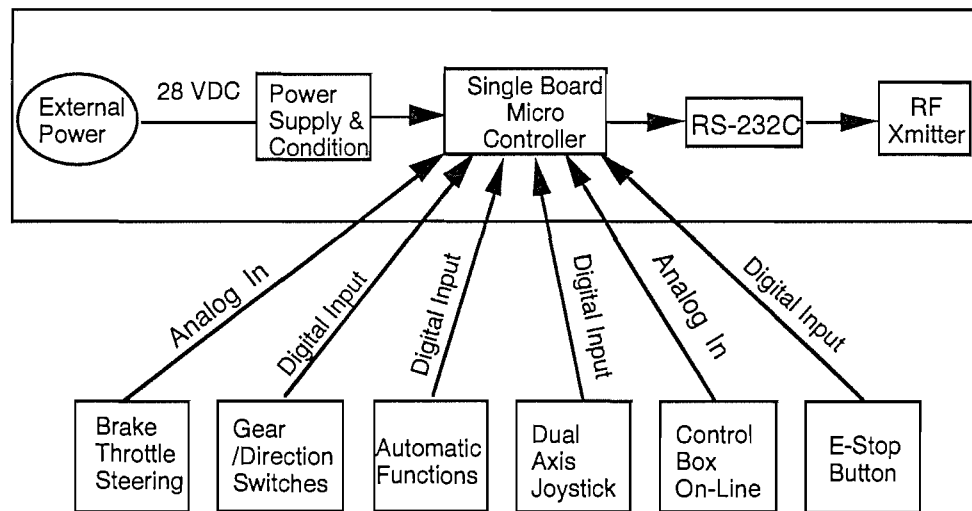


Figure 4.2 Block diagram of the Operator Control Computer (OCC)

4.3 ON-BOARD CONTROL SYSTEM

4.3.1 System Description

Figure 4.3 shows a block diagram of the on-board Vehicle Control System. All of the signals with the exception of the transmission controls operate in parallel with the

the signals with the exception of the transmission controls operate in parallel with the manual controls on board the front-end loader.

The direction and gear selection functions need to be controlled by either the manual levers on the loader or by the remote unit, so a transmission multiplexer is installed between the loader console and the transmission control module. A single signal, Remote/Local, determines which inputs will be passed on to the transmission module.

The arm and bucket control functions as well as the steering and brakes are implemented with on/off solenoid valves which are activated by solid state relays in the vehicle control unit.

Tracking steering is achieved through the use of an analog feedback signal of steering angle. The computer compares the machine angle to the steering angle requested by the remote console and activates the appropriate steering solenoid until the angles are equal.

Throttle control is achieved through the use of a fully proportional linear actuator. Digital input of the return to dig, travel height, and dump height sensors facilitate the implementation of automated bucket functions. Input of the neutral position of the manual direction lever, the brake pedal, and the AOK (all system OK) output of the loader are provided to the Vehicle Control System to implement safety functions.

The Loader Control Computer communicates with the remote Operator Control Computer via a pair of RF transceivers operating at the frequency range 902-928 MHz.

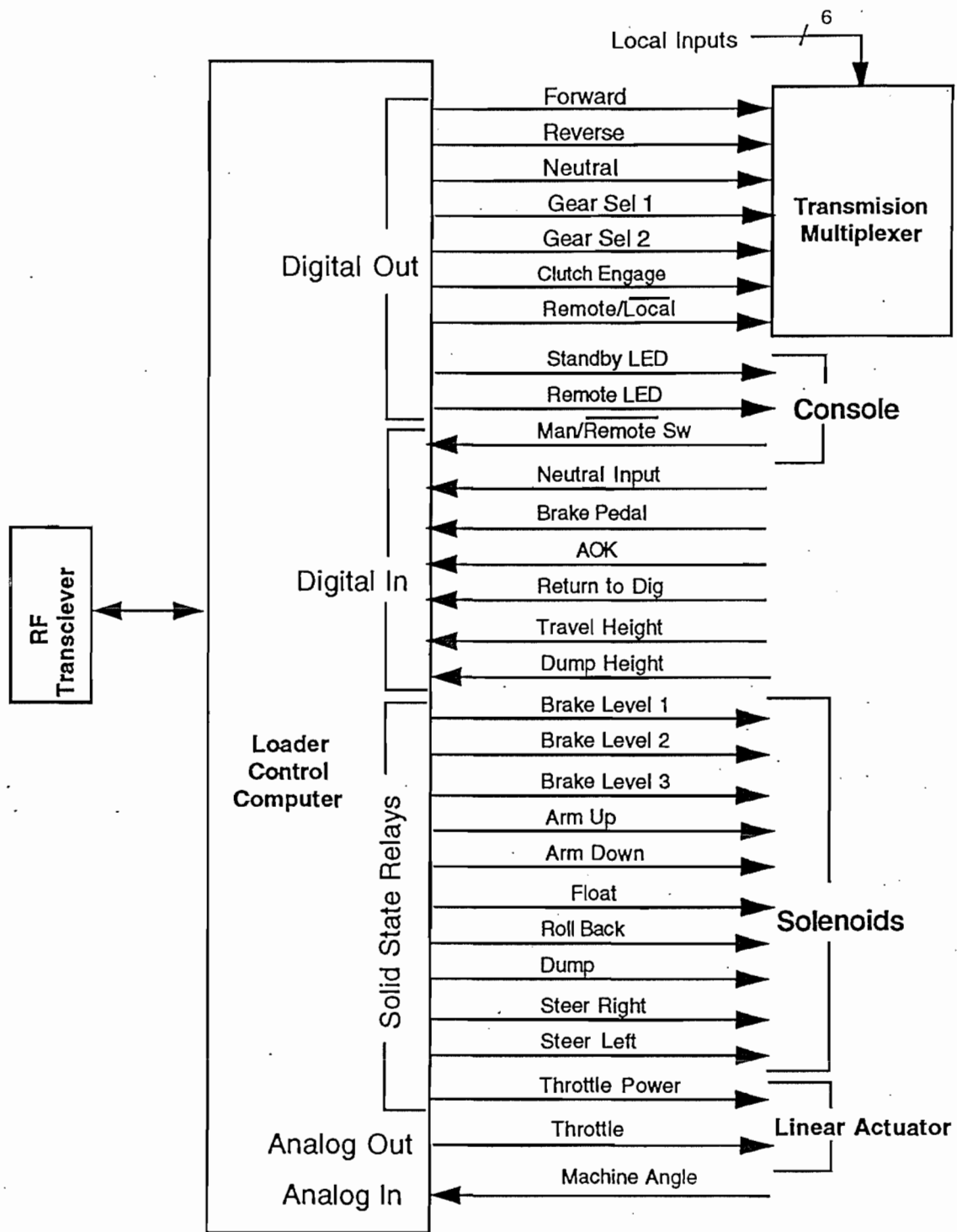


Figure 4.3 Block diagram of the on-board loader control system

4.3.2 Loader Control Computer (LCC)

The Loader Control Computer (LCC) is composed of four STD bus cards (Figure 4.4). The heart of the system is a Ziotech ZT8809 single board computer. With its 8088

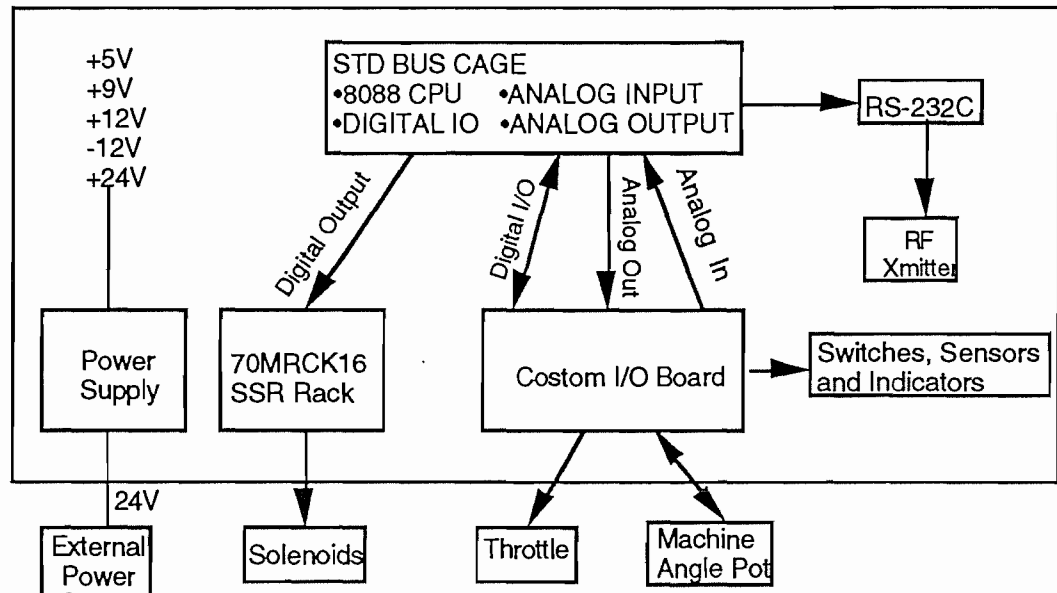


Figure 4.4 Block diagram of the Loader Control Computer (LCC)

CPU and 8 MHz clock rate, the ZT8809 is a fairly low performance device; however it is quite sufficient for a control application such as this. The ZT8809 card uses a standard DOS operating system and has two serial ports and 544 Kilobytes of battery backed-up static RAM. Part of the RAM can be configured as RAM disks so that application programs and data can be stored using regular DOS commands. Stored applications can be set to automatically execute upon power up by adding their names and paths in the AUTOEXEC.BAT file on disk R.

The ZT8809 is supported by three I/O cards from Win Systems, Inc.: an LPM-750 card provides up to 48 lines of digital I/O, a LPM-A/D12 provides 16 channels of 12-bit A/D converter, and an LPM-D/A8 supplies 8 channels of 8-bit A/D converter.

Figure 4.5 is a photo the Loader Control Computer. The diagrams of the main interface board and the front board of the LCC are included in Appendices A3 and A4, respectively.

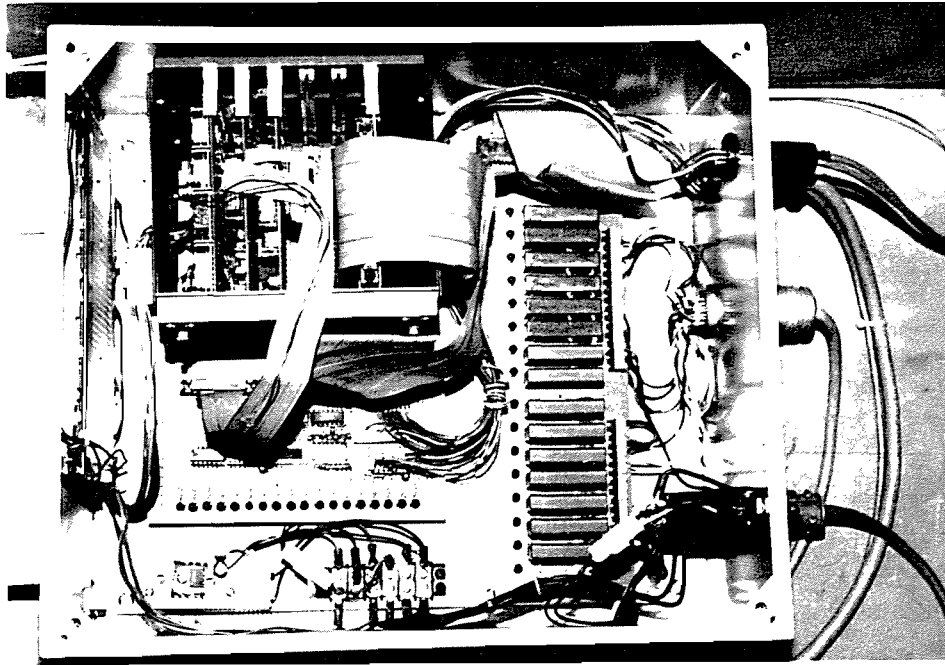


Figure 4.5 Photo of the LCC

4.3.3 Solenoid Drivers

Connected to the LPM-7508 digital I/O card is a sixteen-slot I/O module board. This board can hold up to 16 standard I/O modules that can be either input modules or solid state relays. The Loader Control System uses this board to hold solid state relays for driving the hydraulic and pneumatic solenoids, which operate the various loader functions. The modules can provide up to 5 amps of current, which is sufficient for this application. The outputs have 5 amp fuses to protect against short circuits.

4.3.4 Digital Input And Output Board

The digital inputs and outputs described in the system description above are interfaced to the LPM-7508 card via a custom board designed specifically for this purpose. The transmission multiplexer described in the next section requires 24-volt open-collector drivers for its inputs. This is provided by a SN7407-N hex buffer integrated circuit as shown below.

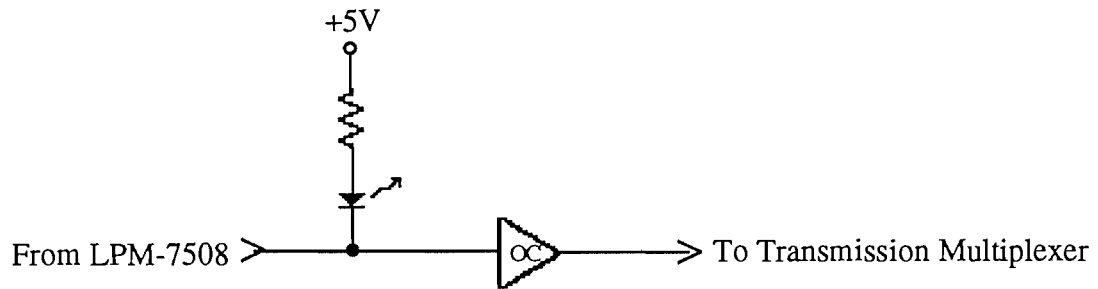


Figure 4. 6 SN7404-N hex buffer integrated circuit

Each output channel also drives an LED so that maintenance personnel can tell which outputs are active by looking at the board. The digital inputs from the loader are 24 volt signals, so attenuators are necessary to reduce the level to 5 volts. Since some of these inputs contain inductive components, capacitors are placed after the attenuators to guard against spikes. The processed inputs are then applied to schmidt trigger gates to provide hysteresis for noise immunity. Finally, open-collector inverters are used to invert the signals and pass them along to the LPM-7508. The pull-up resistors for these inverters have LED's in series to provide a visual indication of the state of each input. A sample of these circuits is shown below.

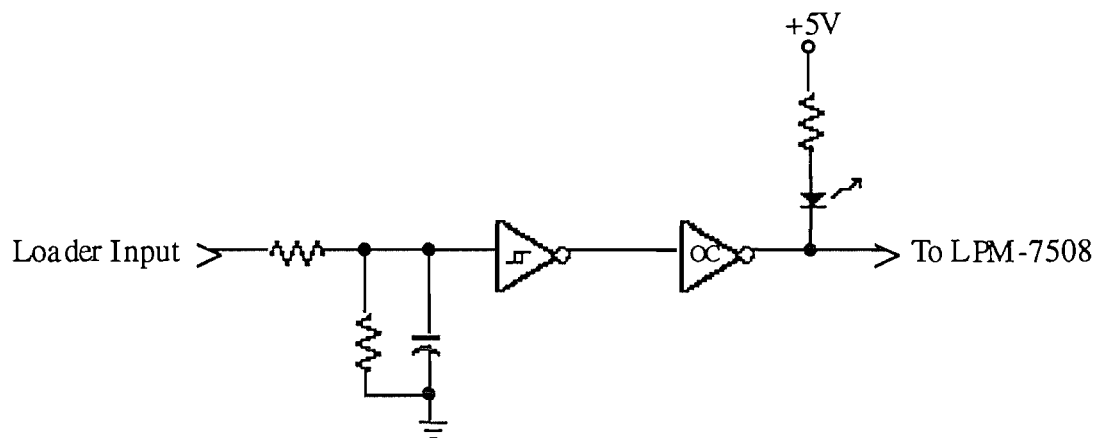


Figure 4.7 Sample circuit of attenuators

It should be noted that I/O modules could have been used for these input and output functions in place of this custom board. Since noisy signals from the front-end loader have been a problem with past systems, it was decided that the higher triggering level and noise immunity of the schmidt trigger approach was a valuable asset.

4.3.5 Transmission Multiplexer

As was noted in the system description above, most of the outputs to the front-end loader operate in parallel with the existing on-board systems. The transmission, however, must only receive its commands from one source at a time, so a transmission multiplexer circuit was constructed. The Case 621 uses an electronic module to control its transmission. The inputs to this module normally come from the transmission lever on the drivers console. These wires were cut just before the module and circuit board containing the multiplexer was inserted between the console and the module. Inputs from the LCC also come into the multiplexer and a single control line from the LCC determines which set of inputs the transmission will respond to. The multiplexer is implemented with optoisolators as shown below.

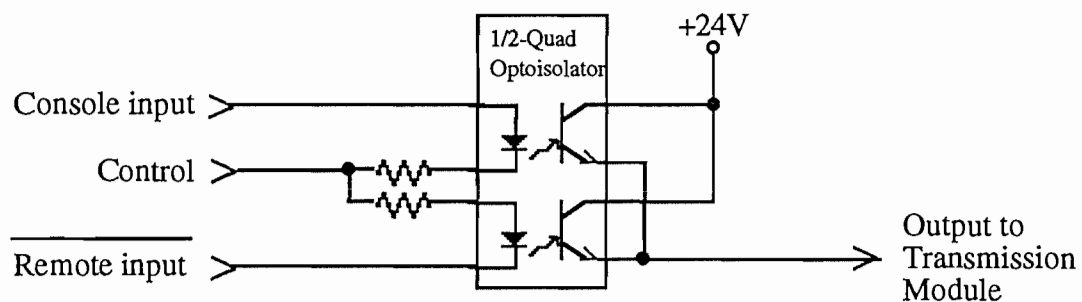


Figure 4.8 Basic circuit of transmission multiplexer

Since the inputs to the optoisolators is polarity sensitive, it is a simple matter to multiplex the inputs by connecting two isolators in parallel as shown. When the control

line is grounded, the upper input diode conducts in response to a 24-volt input from the console control while the lower diode is either at 0 volt or reverse biased. When the control line is raised to 24 volts, the lower diode will conduct in response to a low level on its cathode while the upper diode is disabled. The circuit of multiplexer is included in Appendix A5.

4.4 RF COMMUNICATION LINK

4.4.1 Theory of Spread Spectrum Communication Operation

A pair of spread spectrum technology RF modems (STI SpectraData model 5500) was used for data communication for the remote control of the front-end loader. In performing spread spectrum transmission, the SpectraData distributes random data across a range of frequencies in a seemingly random but actually a defined manner. The corresponding SpectraData receiver de-spreads the signal to restore the data. Unlike conventional AM (single frequency) or FM (narrow band) radio, spread spectrum transmits redundant data over several frequencies across a broad band. This is done by a pre-defined methods, so narrow frequency interference will not obscure communication. The corresponding receiver despreads the received signal and restores it to digital form. additionally, a built-in error detecting algorithm ensures data integrity even when a SpectraData receiver is on the borderline of the transmitter range; either the received data is accurate or it is rejected.

The pair of modems in the RF link system is configured to communicate with each other and only each other. When more than one pair of modems are in the same area and may all hear each other's messages, the software contained in each modem would filter out messages which are not intended for it. For the front-end loader application, the system was configured in a point-to-point topology; a modem that gets data from its serial port will turn the data and the calculated error-detecting checksum into a packet. To this packet the modem adds its address and the destination modem's address as well as

other critical network information. After sending the packet, the modem waits for an acknowledgment from the destination modem that the packet was received without error. If it does not receive such an acknowledgment, the modem will re-transmit the packet and continue doing so until it either receives an acknowledgment or exceeds its maximum number of re-tries. At this point, the transmitting modem will send the next data packet. The modem that receives packet over its radio will filter it based on the destination address. Only the modem with its own serial number as the destination address will save the packet. Upon retention of a packet, the modem also extracts the original message from the packet and subjects the data to the error-detection algorithm. If the checksum matches the message checksum, the modem sends the data out its serial port and radios acknowledgment to the sender. If the two checksums disagree, the data is not sent out the RS-232C port nor will the radio send an acknowledgment. This guarantees the user's device only receives error-free transmissions and that the transmitter does not continue to send corrupted data.

The communication topology of the spread spectrum modem is illustrated in Figure 4.9. For each pair of the communication unit, one of the seven radio frequency bands and one of the 65,536 sub-channels will be chosen. The serial number of the corresponding receiver modem is specified when setting up the transmitting modem. These settings with other configuration information are stored in non-volatile electrically erasable, programmable, read-only memory (EEPROM). This spread spectrum transmission and internal error checking provide superior data reliability and security when compared to single-channel narrow band transmission systems.

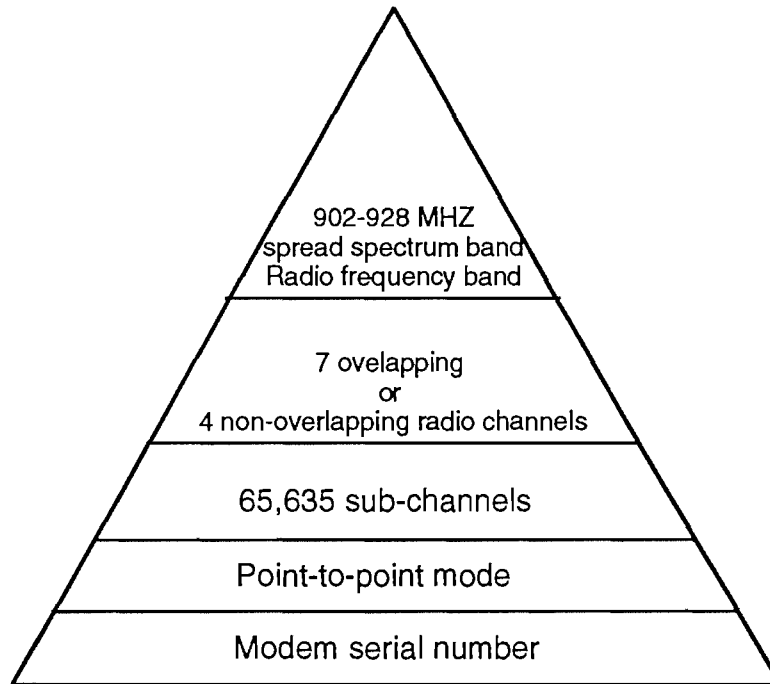


Figure 4.9 An overview of the modem's topology

4.4.2 Unit Description

The front-end loader control system uses two identical RF units for transmitting the data between the OCC and LCC. Each unit is a transceiver, that is, it is capable of both transmitting and receiving data. Each RF unit has a data modem that exchanges digital information between a host computer and controller where the computer and controller use RS-232C communications. The transceivers use spread spectrum transmissions that can transmit over several frequencies in the 902-928 MHz range. The RF system consists of a modem/transceiver module, a build-in RS-232C serial port, power supply and antenna for each end of the data link. The vehicle antenna with its modem was mounted to the top of the vehicle and its location is high enough that the bucket can fully extend without disturbing the RF path. The antenna currently used is 1/4 wavelength, 0 dB gain omni-directional whip type with a nominal range of 800 feet (244 meters). However, with the optional 1/2 wavelength, 2.5 dB gain omni-directional whip antenna, the

operation range can be doubled to 1600 feet (488 m). The physical size of each transceiver is 3.9 in x 6.5 in x 8.0 in (10 cm x 16.5 cm x 2 cm) and the power requirement is 9 VDC at 0.4 amps.

The data interface is a standard RS-232C and the data format is eight data bits, one stop and start bit, and no parity bits. The data rate is selectable from 1.2 to 19.2 K Baud and for this application 9.6 K Baud rate was used. The internal error checking routing uses CRC-16 algorithm.

4.4.3 Communications Protocol

LCC and OCC communicates via a simple communications protocol. OCC can generate two types of messages: keep alive message or line mode is active. Keep alive message is sent when Box On-line is not true. At least one of the two commands are sent at all times every 200 milliseconds. LCC uses these messages to find out whether OCC is alive or there is a problem with communications.

LCC, however, only generates one type of message, namely, status message. This message is generated every second. OCC uses it to establish connection with LCC. The detailed description of OCC and LCC messages are included in Appendix A7.

4.5 SOFTWARE DESIGN

The software for the Operator Control Computer (OCC) and Loader Control Computer (LCC) was developed on a PC/DOS platform. However, due to different target hardware used for the two computers, different development environment was used to implement the software for each computer. Dynamic C v2.0 supplied with the Z-world single-board computer was used as the development tool for OCC. The LCC program was developed in Microsoft C v7.00 using an IBM PC compatible computer. Its executable codes were then downloaded to the single board computer and running in the STD DOS environment.

Figure 4.10 and 4.11 are the software flow charts for OCC and LCC respectively.

As mentioned in the previous section, once the communication is established, the OCC will send "Keep-Alive" message or Command message every 100 milliseconds, and LCC will send out the system status message to OCC every second. On LCC, the message receiving procedure employs hardware interrupt for receiving data on time. The system status message is sent out character by character until the transmitter register is empty. When LCC receives a complete message, it will check if there is any error during the transmission. If the message is a correct command, it will be decoded and explained according to the protocol (refer to Appendix A7). The program checks the system status all the time and reports it to the OCC. It also monitors the communication status. When the communication fails or ceases, the program enables the loader system to go to Standby state, which is essentially the same state as the Emergency stop.

Six digital I/O ports and three analog I/O ports are used in the system currently. The addresses for these ports and the bit definitions for each port are listed in Appendix A8.

The source codes are listed in Appendices B1 and B2.

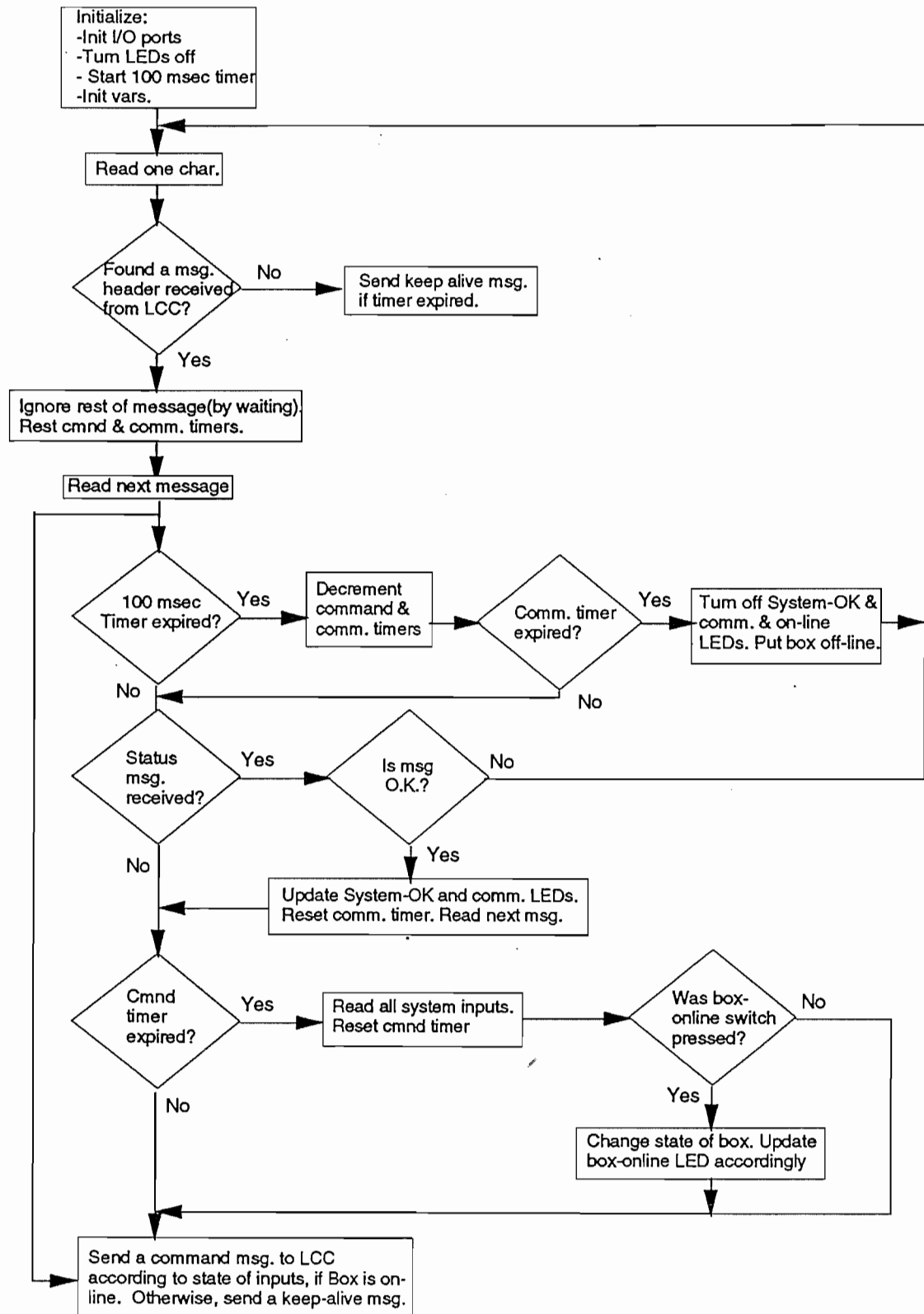


Figure 4.10 Software Flow Chart for Operator Control Computer

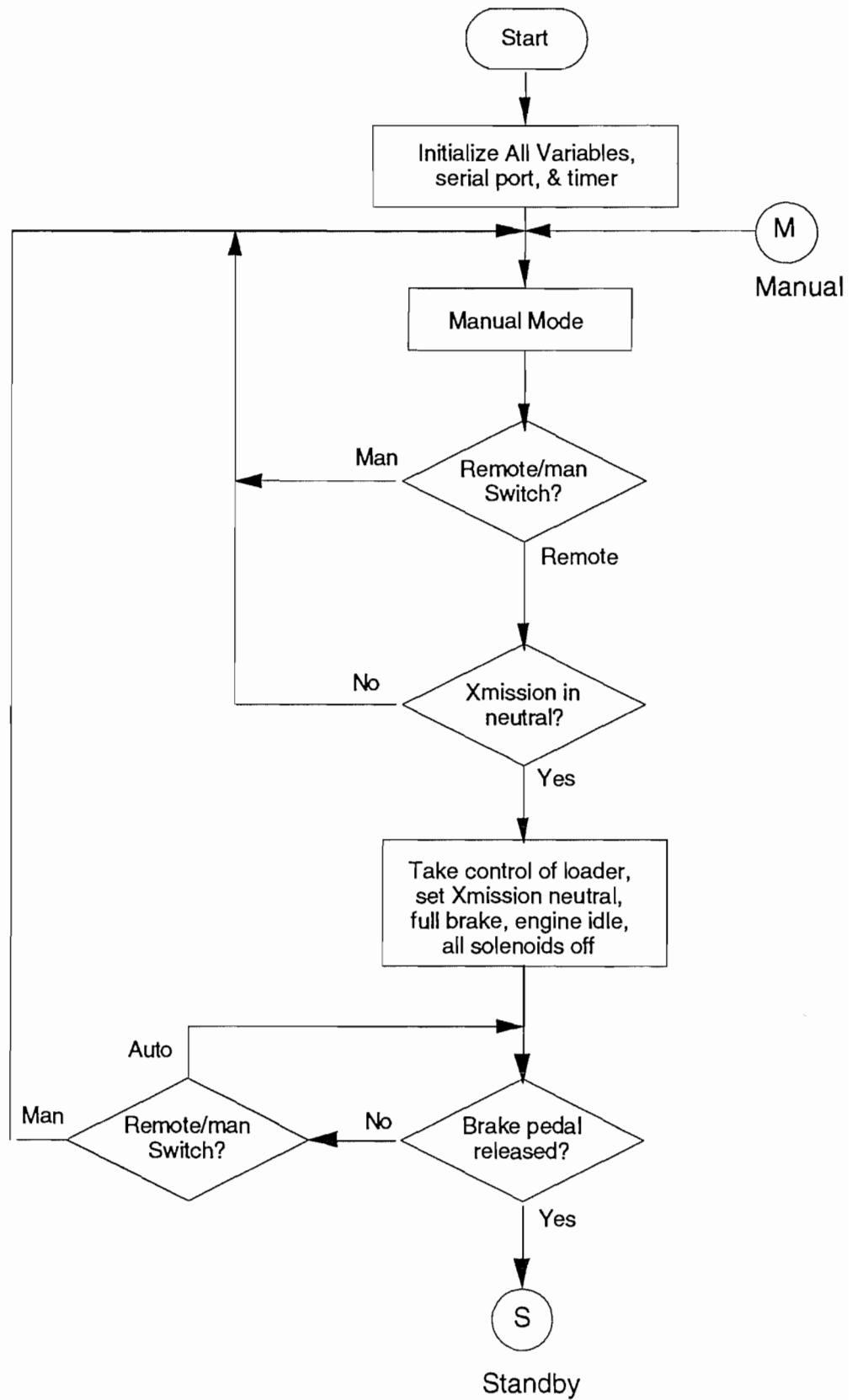


Figure 4.11 Software flowchart of OCC (part 1)

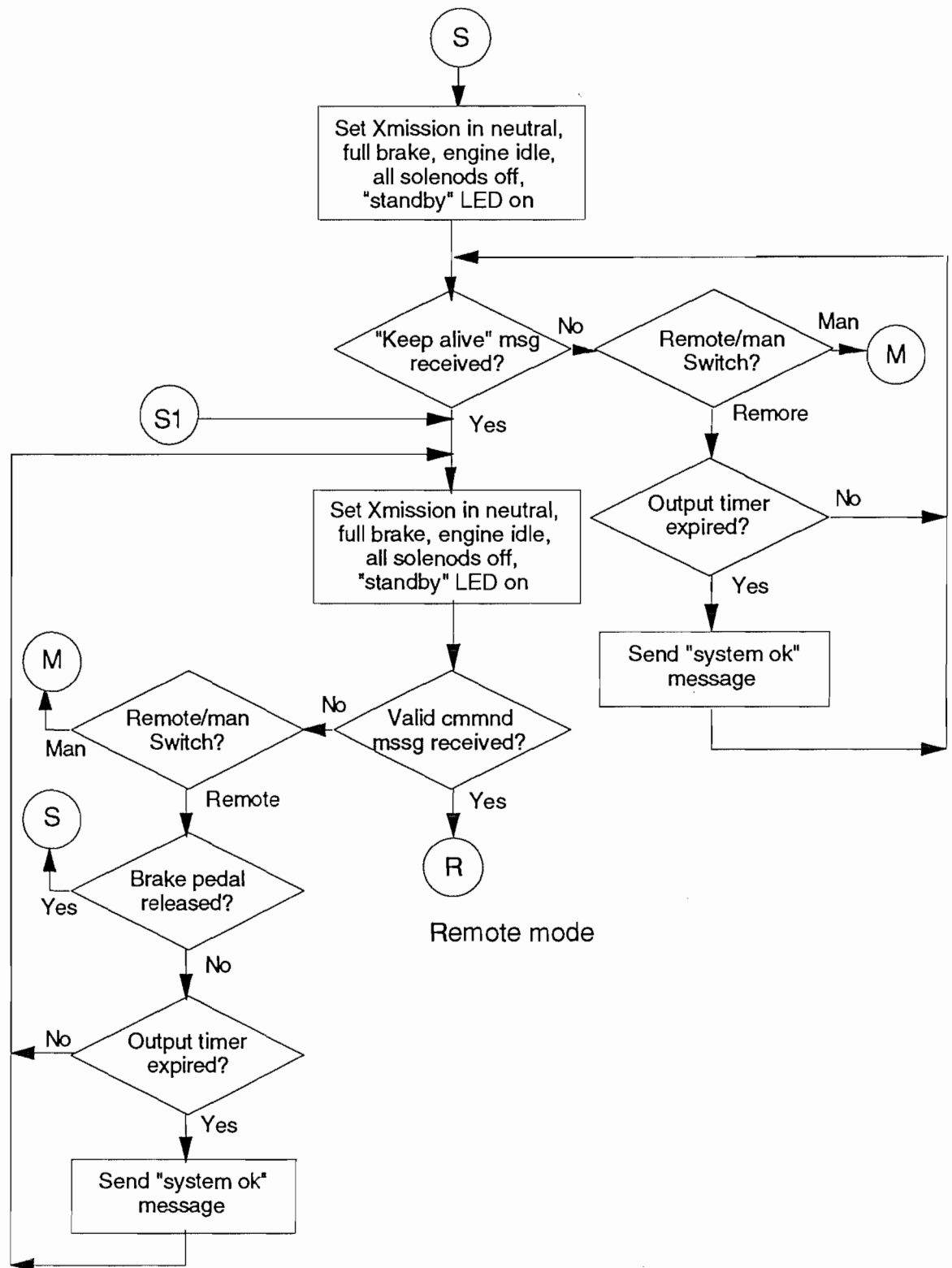


Figure 4.11 Software flowchart of OCC (part 2)

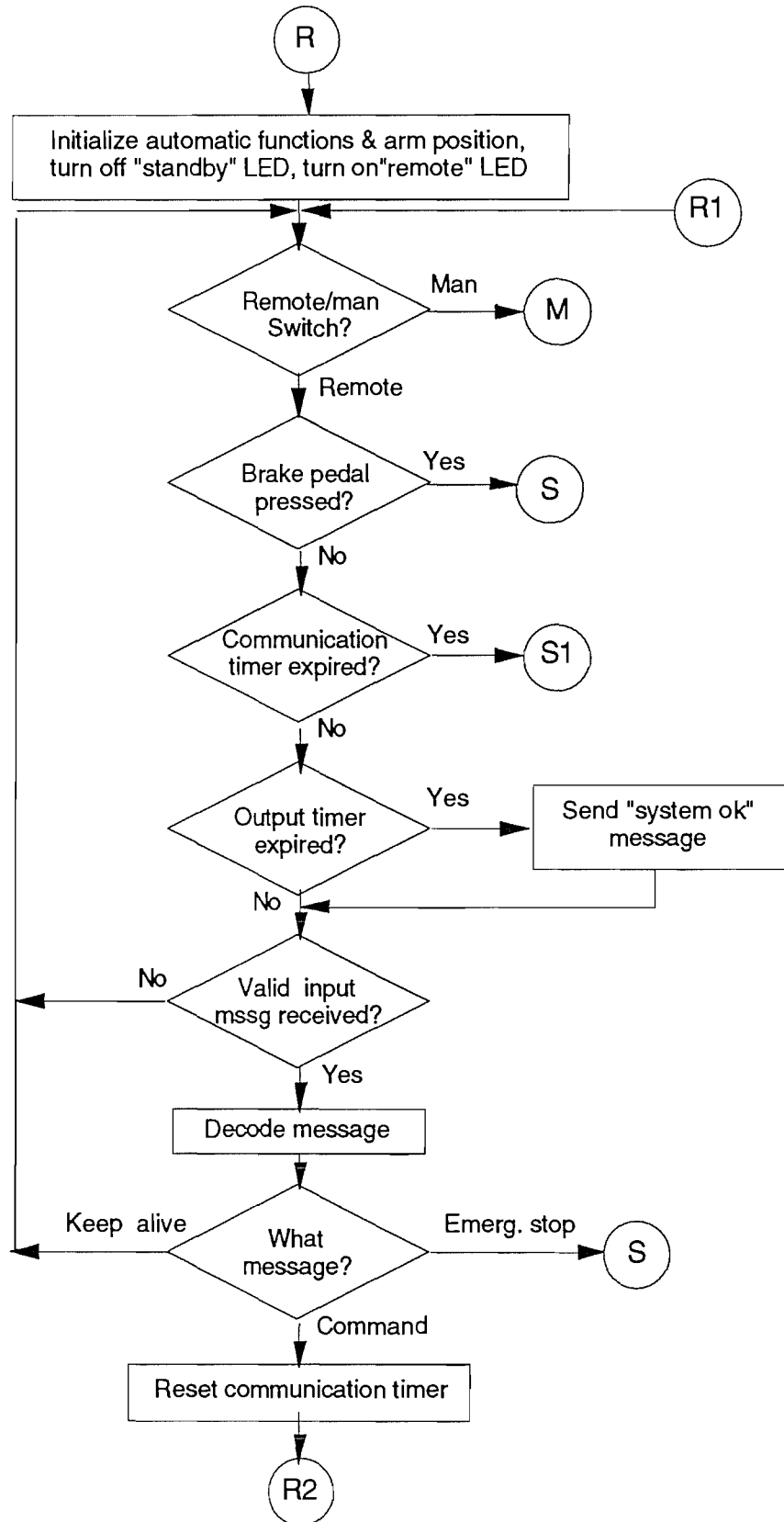


Figure 4.11 Software flowchart of OCC (part 3)

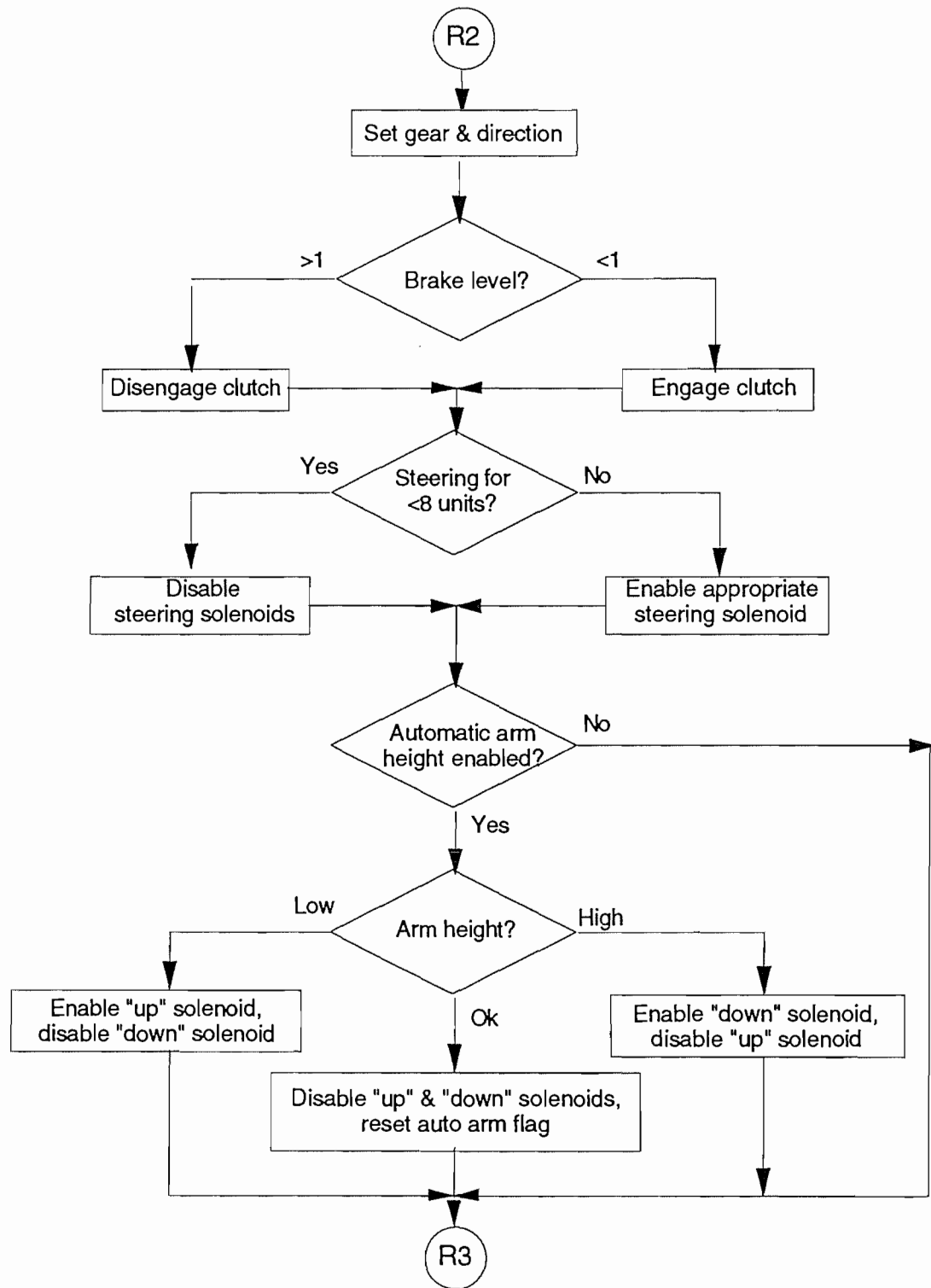


Figure 4.11 Software flowchart of OCC (part 4)

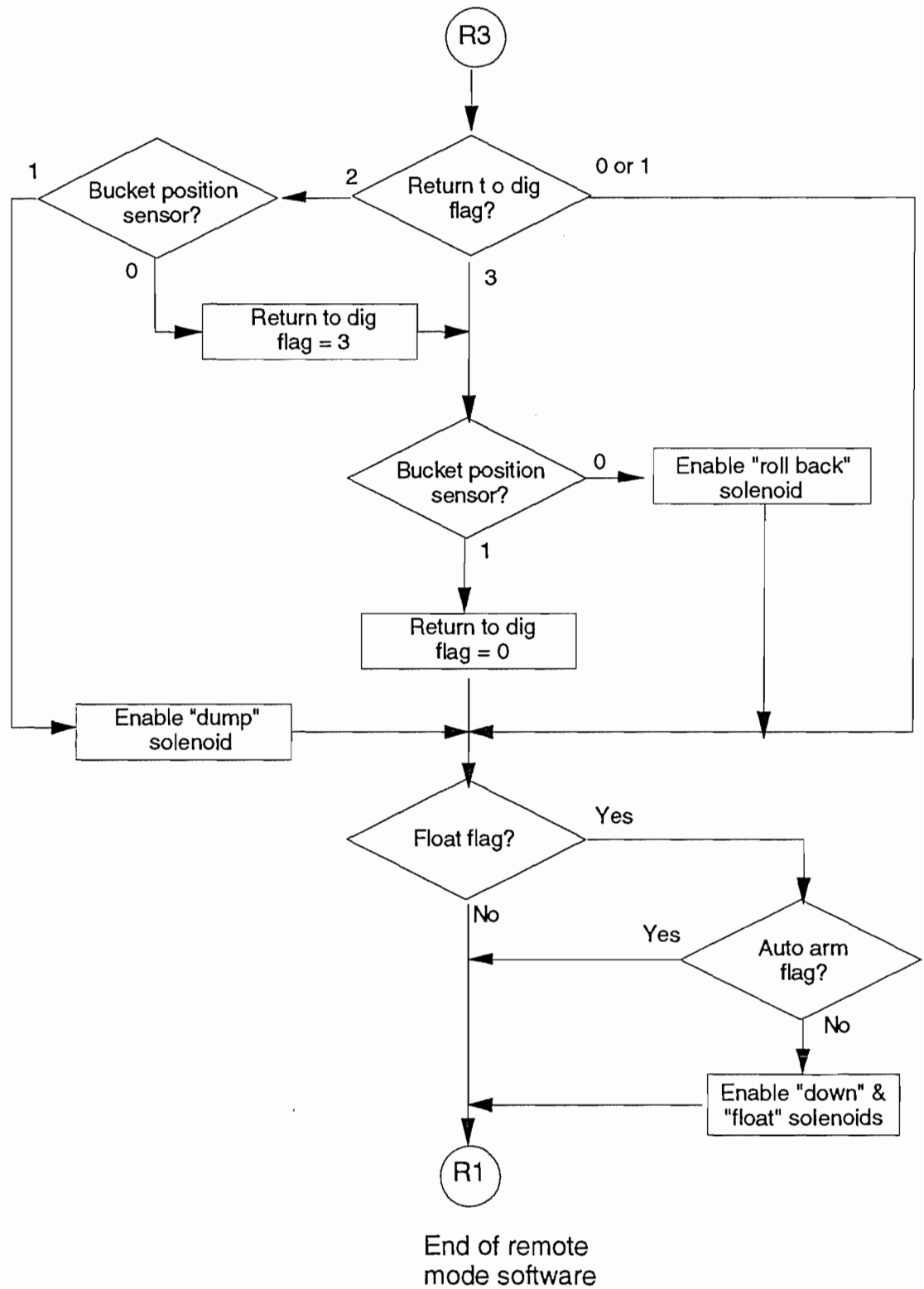


Figure 4.11 Software flowchart of OCC (part 5)

Chapter 5

Testing and Results

5.1 INTRODUCTION

The remotely operated front-end loader has been through a variety of field tests. The early stage tests were conducted to verify the design concept and the performance of various subsystems. The final field test was focused on the overall performance of the remotely controlled front-end loader. In addition to the final test in an open field at University of California at Davis, the operation performance was also evaluated at a number of day-long continuous demonstrations where the machine was operated by Caltrans maintenance workers at the locations of the demonstrations including Caltrans District 11 (San Diego), District 4 (Oakland) and District 3 (Sacramento). The testing included operations up to 300 feet (91.4 m) away, and it covered a wide range of front-end loader functions from scraping, digging, and dumping to obstacle navigation.

5.2 SUBSYSTEM TESTING

The subsystems were first tested with a hardwired (tethered) operator control unit prior to the implementation of the radio remote control package. It was then modified and further tested with the radio remote control.

- Brakes:

The three-level brake seems to be sufficient for smooth vehicle operation. No down grades or overhauling load conditions occurred. The transition between stages is smooth enough to prevent load loss due to abrupt transitions.

- Throttle:

The throttle was initially a three-stage device like the brakes. Satisfactory performance was observed from the two air cylinder activated control systems. However, this turned out to be a little too abrupt for clean maneuvers, especially for powering up for

bucket operations when increased hydraulic flow was needed. Thus, the throttle has been updated to be fully proportional to provide the smooth speed change. This works exceptionally well and allows the operator the same degree of control as found in the vehicle.

- Steering:

Steering angle tracking was achieved by using a solenoid operated hydraulic valve and constant flow hydraulic source along with matching of variable voltage position indicators. This approach seemed quite adequate for operators who readily adapt to the concept of turning to master small increments for small steering angle, waiting for vehicle to finish responding then adding additional turn angle in the same fashion. Once an operator becomes accustomed to the articulating nature of the front-end loader steering is easily accomplished. However, for those who visualize steering wheel movement and vehicle response as they do on-board, the tendency is to hunt for the desired steering angle. Since that goal is also changing, the result is a continuously jerking vehicle. Operators also initially have difficulty with the lack of rate proportional steering but seem to easily adjust to the position steering without problems. Experience, as little as a half-hour, allows operators to compensate for the articulating steering to the positional steering, and the time lag in response (about 200 milliseconds). Further research and tuning on this function will improve the response.

- Arm/Bucket:

The arm and bucket functioned satisfactorily. However, since they are only on/off controls, unlike the on-board joystick, there is a relatively larger deadload range around the neutral position of the joystick. When operated by an operator who was not aware of this, a time lag between the control and the bucket action was observed. Even though the 200 ms lag is small, it is perceptible. The time delay is seemingly further worsened due to the lack of proportional control. This is offset though by throttle modulation, which increases

hydraulic flow, and, consequently, speeding up arm/bucket maneuvers. Resetting the pressure regulators that control speeds can also help this situation.

- Portable remote operator Unit:

The functioning of the portable unit was easily picked up by an unfamiliar user. The unit has been used extensively in the field by a wide variety of experienced and inexperienced vehicle operators. However, the steering takes some time for the operators to get familiar with. This is because the remote front-end loader steers by articulation so it becomes difficult to determine which way to steer especially if the vehicle is coming head-on to the operator.

The portable unit allows the remote operator to move around to gain better visibility. This became more obvious when the loader was remotely operated to load the gravel from a storage pile sectioned by the concrete walls. However, the weight of the back pack , 17.2 lb (7.8 kg) on the operator becomes a nuisance during prolonged operations. The portable unit's interface 32-pin plug interface appeared to fatigue after extensive use. This problem though can be easily remedied by upgrading the plug to a more robust type; there has been no communication loss yet due to the loose plug although it may soon occur.

- Stationary Unit:

The stationary unit has also undergone extensive testing. The learning curve for the stationary unit appeared to be shorter than that for the portable unit, simply because the stationary unit is a mock up of the vehicle cab and is therefore familiar to vehicle operators. The articulating steering problem is also apparent with the stationary workstation, although it did not appear to be as significant since the steering is more familiar and thus easily accounted for. One major problem with the stationary unit is the drive mechanism for the rotating base. Previously the drive used a belt, which slips easily. When the stationary unit was not perfectly flat and at some incline, the belt would slip, especially with a large operator. As result, the stationary unit would move and remain in a position where gravity

pulls on the offset. This problem was fixed and updated to a gear drive to prevent slipping. The modified drive train allows the base to rotate as desired for good line-of-sight vision.

- Radio Frequency Communication

Initial testing through vehicle field operations revealed a tendency for communications to terminate momentarily and cause a vehicle shut down. Since the communications program was designed to stop when the vehicle did not receive valid data communication for approximately 300 milliseconds this was expected. However it became frustrating to need to constantly re-initiate communication by repeating start up procedures from the front-end loader. Thus, the communications program was re-written such that if communication was lost, the vehicle paused while the system automatically re-initialized. This caused for approximate half second pauses in vehicle operations when ever communications “dropped out.” This solution allowed for continual testing.

To further remedy the communication problem, it was theorized that higher gain antennas for the RF modem would alleviate the drop out by boosting signal strength. However before this solution was implemented it was soon discovered that the test site at which the majority of the vehicle testing had been completed was in close proximity to a cellular phone relay station. The frequency of the cellular phone tower was overlapped to the operating frequency of the communication system (850 ~ 950 MHz range vs. 902 ~ 928 MHz range) and the prevailing power of the tower caused loader communication failure. By relocating the loader to a new test site away from the high power cellular station, the communication drop out no longer occurred. The communications have been reliable since then. Communication interference was observed while vehicle operations were performed during a live news broadcast as the news van utilized transmitters to broadcast the live signal, which was also near the communications frequency. However, it was insignificant and the system was automatically reset and the communication was quickly re-built without the audience noticing.

It should be emphasized that, although there have been communications drop outs, they have not been dangerous. The design of the communications system causes the vehicle to terminate operations so as not to create a dangerous situation with a runaway vehicle. The automatic re-initializing of the communications also occurs only when communications have been re-established further preventing dangerous situations.

5.3 OVERALL PERFORMANCE

After the Loader Control Computer was modified and the program rewritten, the overall performance of the remotely controlled front-end loader was evaluated. The final field test was conducted in an empty vineyard field at University of California Davis. Figures 5.1 and 5.2 are photos of the front-end loader remotely operated using portable unit and stationary work station, respectively. The evaluation was extended to day-long continuous demonstrations, where the machine was operated by Caltrans maintenance workers at various sites including the courtyard of Caltrans District 11 facilities (San Diego), Jack London Square at Oakland during District 4 Equipment Rodeo) and an open field at Ranch Marietta, Sacramento (District 3). The tasks performed include digging, loading, free dumping, truck dumping, land leveling and obstacle navigation.

The field tests have shown that operators can maneuver the front-end loader successfully with both control units and perform the full range of loader functions from a site up to 300 feet (91 meters) away from the vehicle. Remote operations were performed successfully and nearly as quickly and well as if the operator were in the machine. During the six day-long continuous operations, the control system functioned very well with only one failure on the throttle control due to a loose wire connection inside of the proportional actuator purchased from ADCO. No other breakdown or safety problem had occurred. Twelve Caltrans maintenance workers had operated the loader with the remote control units, both portable and stationary. It took them less than half an hour (some of them



Figure 5.1 The front-end loader operated with the portable remote control unit



Figure 5.2 The front-end loader operated with the stationary remote control unit

claimed 10 minutes) to get used to the control and feel comfortable to operate the loader remotely. The maximum distance tested was around 300 feet (91 meters). When operating the loader from a distance greater than 200 feet (61 meters) the operator lost the sound that the vehicle was making. It became very difficult for the operator to know engine loading and thus the operator lost a major sense as to the status of the vehicle. Long distances also make it difficult to determine precise positioning of the vehicle relative to the point of operations.

Chapter 6

Conclusion And Recommendations

6.1 CONCLUSION

The radio remote control system developed enables the front-end loader to be operated remotely by current Caltrans workers to perform full range of functions at a distance up to 300 ft (91 meters). The operation modes can be easily shifted from remote mode to on-board mode or vice versa. The system is reliable, durable, safe, and easy to operate. The overall performance of the remotely-operated front-end loader has been very satisfactory.

The feasibility of teleoperating an unmanned high-way maintenance equipment has been well demonstrated. It is concluded that the remote control package is ready for commercialization and can be easily adapted to general purpose heavy equipment and other mobile machine.

6.2 RECOMMENDATIONS FOR FURTHER STUDY

The recommendations for further study include the integration of telepresence technology and other feedback systems into the remote control package to make the teleoperated loader more efficient and controllable, and the development of teachable automatic functions to enhance productivity of the front-end loader for both remote and on-board operation. Study of human factors related to teleoperation of highway maintenance vehicle equipped with feedback systems is also desired.

The steering control could be altered and improved by using proportional electrohydraulic solenoids to eliminate "jerking" when operated by the operators who have not adapted to the existing steering control system. The stationary unit could be further perfected by refining the foot pedals.

REFERENCES

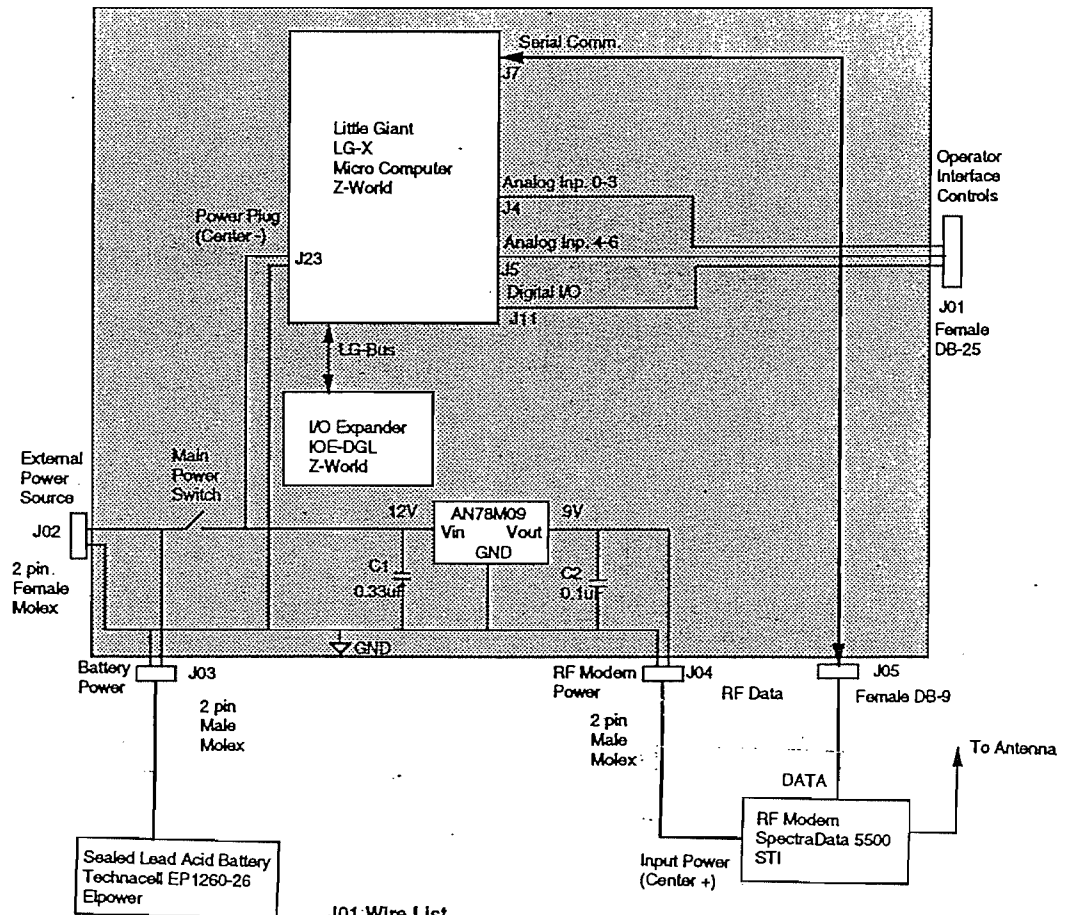
1. Koskinen, K., A. Halme and A. Tuunanen, 1994. "Perception and Navigation System for Autonomous Mobile Applications (the PANORMA Project) -- Experiences and Results." Proceedings of International Conference On Machine Automation. Tampere, Finland.
2. Nieminen, Turo and Mikko Sampo, 1993. "Unmanned Vehicle for Agricultural and Off-Highway Applications." SAE Paper 932475.
3. Fariba, Kamiar, 1993. "Remote Control Loader (RCL) System Documentation." FMC documentation.
4. Zworld engineering, 1992. "Little Giant Miniature Controller Technical Manual, Dynamic C Version 2." pp. 125.
5. Scientific Technologies Inc., 1992. "Spectradata 5500 Series Installation and Operating Manual."
6. ZiaTeck, 1992. "ZT 8808/8809 single Board V20 Computer."

APPENDIX A1

OCC Control Connector Interface

Signal Name	Direction	Type	Specifications
Control Box On-line	Input	Binary	Momt., NO, Closes to Gnd
Emergency Stop	Input	Binary	Main., NO, Closes to Gnd
Steering	Input	Analog	0-2.5 V
Throttle	Input	Analog	0-2.5 V
Brake	Input	Analog	0-2.5 V
Transmission (drive)	Input	Binary	Main., NO, Closes to Gnd
Transmission (reverse)	Input	Binary	Main., NO, Closes to Gnd
Transmission (Gear, Low Bit)	Input	Binary	Main., NO, Closes to Gnd
Transmission (Gear, High Bit)	Input	Binary	Main., NO, Closes to Gnd
Low	Input	Binary	Main., NO, Closes to Gnd
De clutch	Input	Binary	Main., NO, Closes to Gnd
Arm	Input	Analog	0-2.5 V
Bucket	Input	Analog	0-2.5 V
Return to Dig	Input	Binary	Momt., NO, Closes to Gnd
Travel Height	Input	Binary	Momt., NO, Closes to Gnd
Dump Height	Input	Binary	Momt., NO, Closes to Gnd
Radle	Input	Binary	Momt., NO, Closes to Gnd
Float	Input	Binary	Momt., NO, Closes to Gnd
Control Box On-line	Output	Binary	Active-High
All System OK	Output	Binary	Active-High
Comm. Link OK	Output	Binary	Active-High
Power (Several)	Output		5V
Ground (Several)	Output		0V

Pin Diagram for OCC



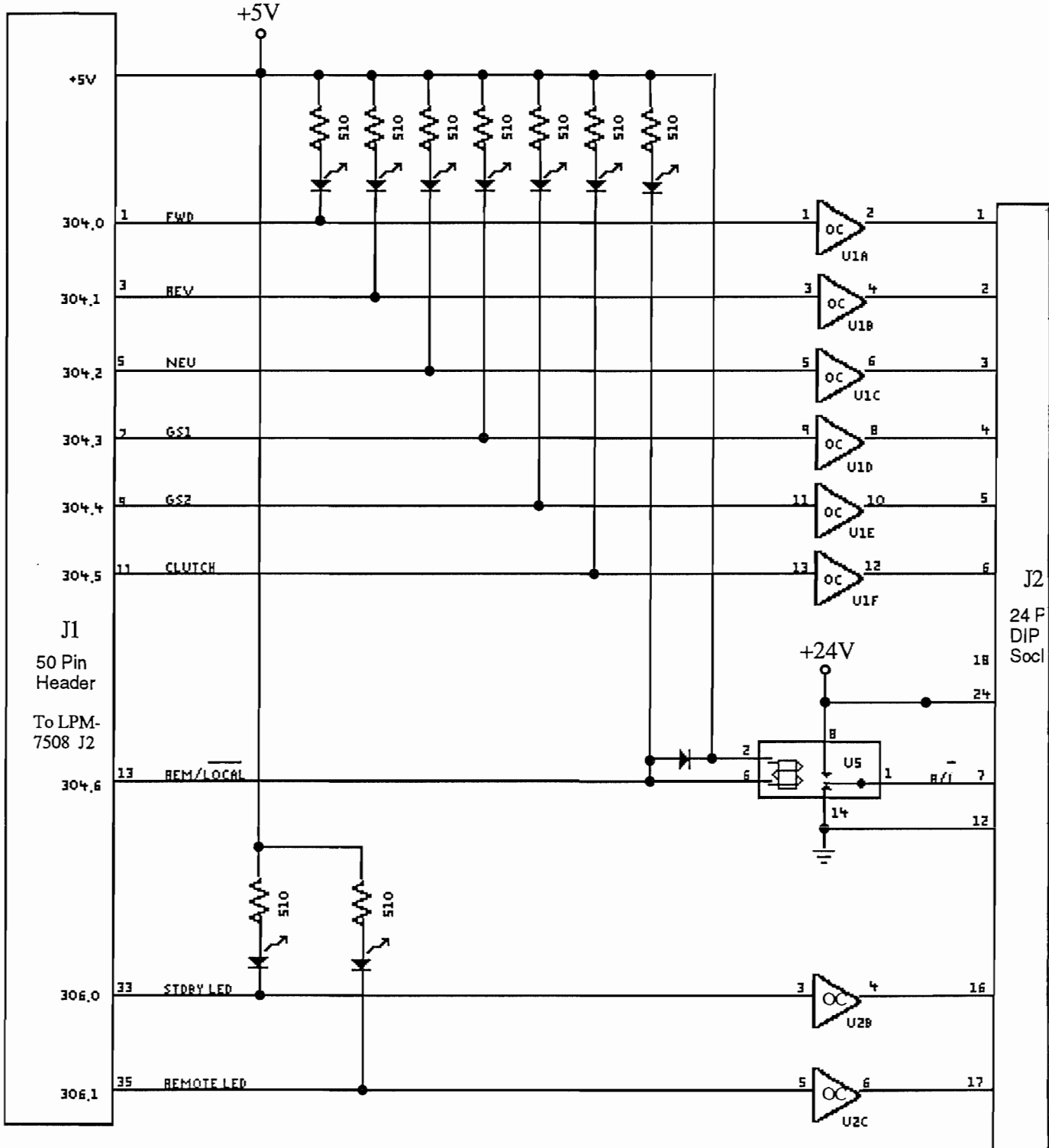
J01-Wire List

From	To
LG-X J11	J01
Pin 1	Pin 1 - VCC +5
Pin 3	Pin 14 - Engine Start
Pin 5	Pin 2 - Engine Kill
Pin 7	Pin 15 - Emerg. Stop
Pin 9	Pin 3 - Return to Dig
Pin 11	Pin 16 - Travel Height
Pin 13	Pin 4 - Dump Height
Pin 15	Pin 17 - Box Online
Pin 17	Pin 5 - Box Online Ind.
Pin 19	Pin 18 - Trans Forward
Pin 21	Pin 6 - Trans Reverse
Pin 23	Pin 19 - Trans Speed Low
Pin 25	Pin 7 - Trans Speed High
Pin 27	Pin 20 - Trans Low Mask
Pin 29	Pin 8 - Trans Declutch
Pin 31	Pin 21 - Sys. OK Ind.
Pin 33	Pin 9 - Comm. OK Ind.
Pin 34	Pin 22 - GND (Digital)
LG-X J4	J01
Pin 2	Pin 10 - Throttle (0-2.5V)
Pin 4	Pin 11 - Brake (0-2.5V)
Pin 6	Pin 12 - Steering (0-2.5V)
Pin 8	Pin 13 - Arm (0-2.5V)
LG-X J5	J01
Pin 2	Pin 23 - GND (Analog)
Pin 4	Pin 24 - Bucket (0-2.5V)

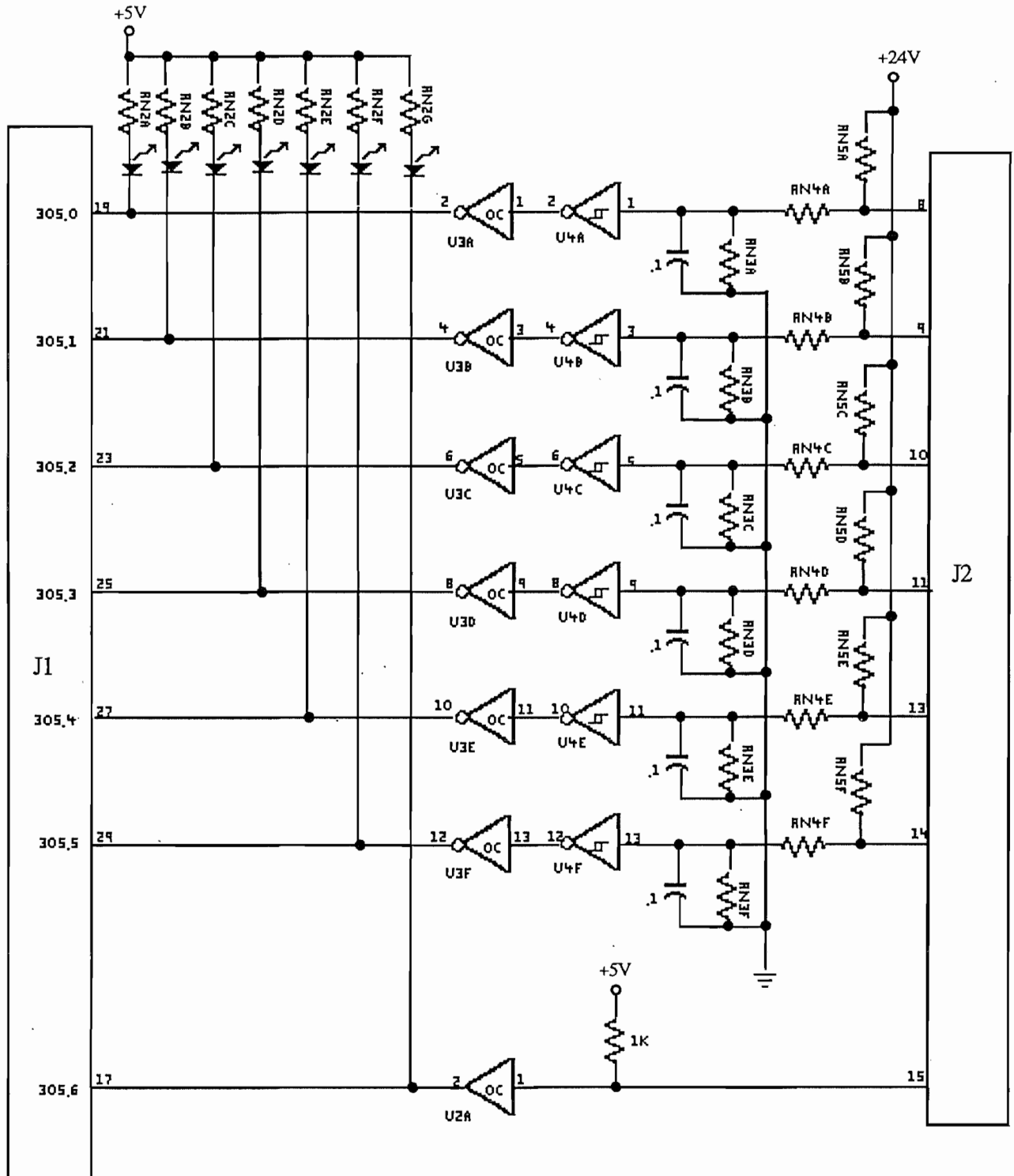
APPENDIX A3

Diagram of LCC Main Interface Board

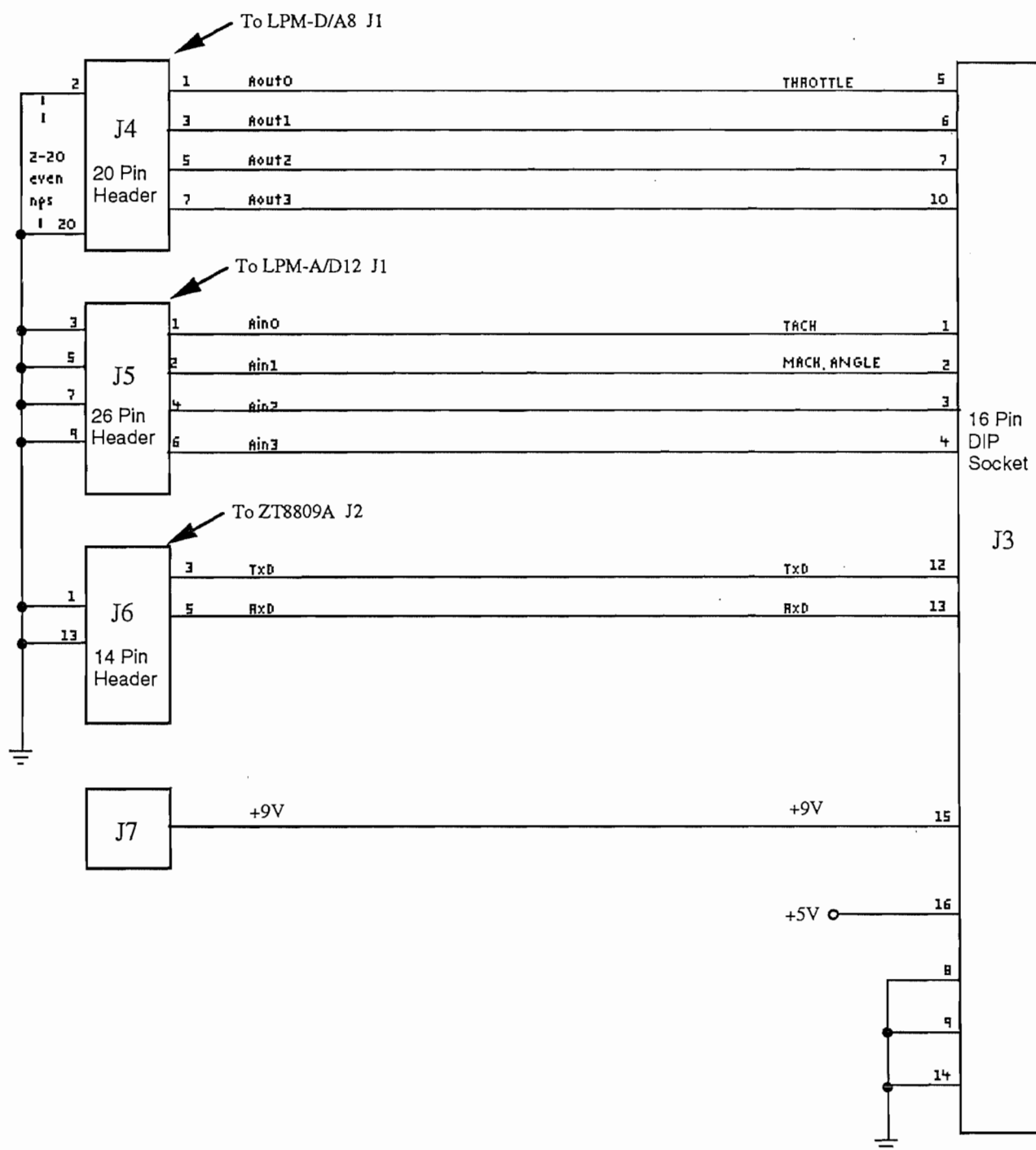
Section I. Digital Output



Section II. Digital Input



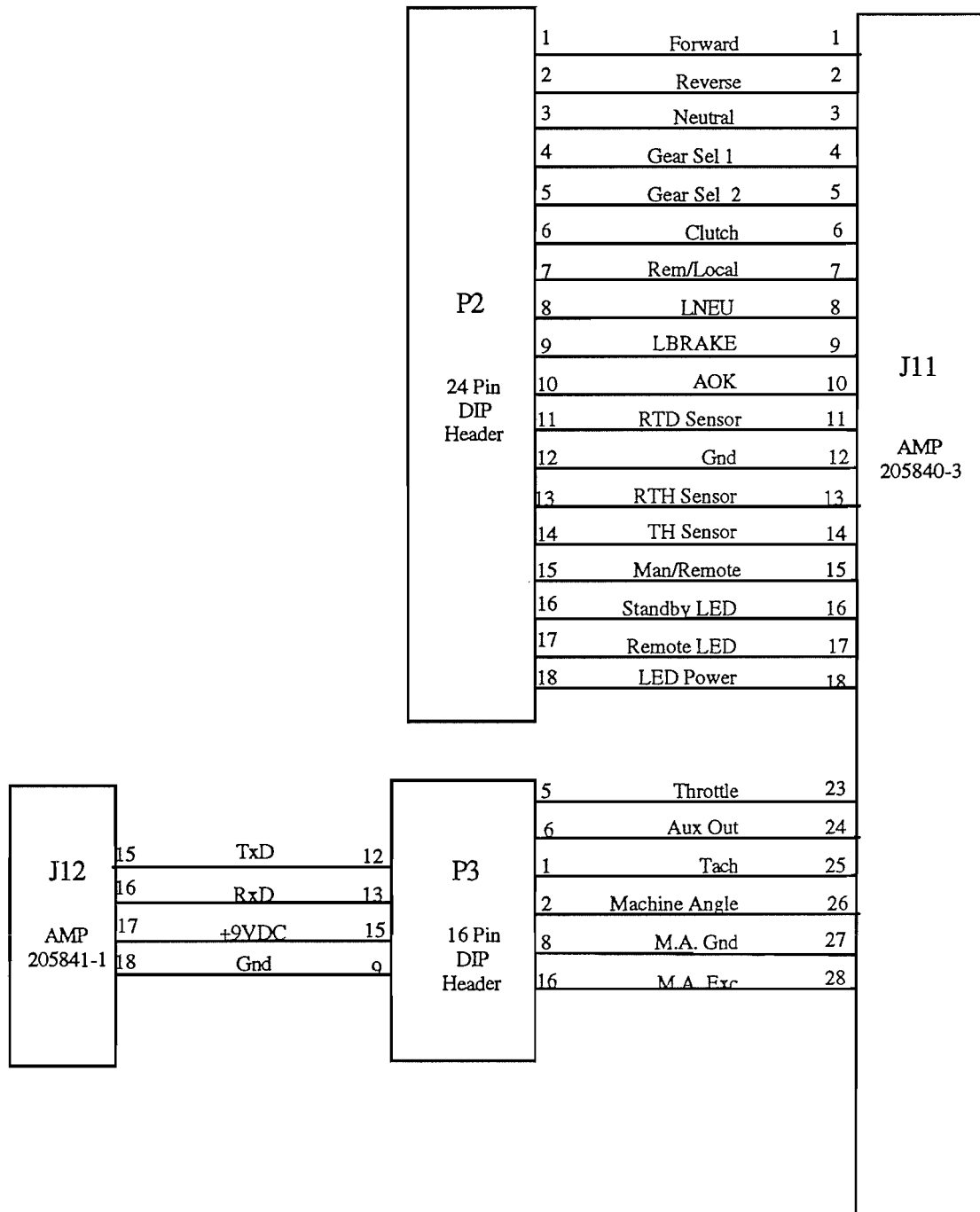
Section III. Analog I/O & Serial I/O

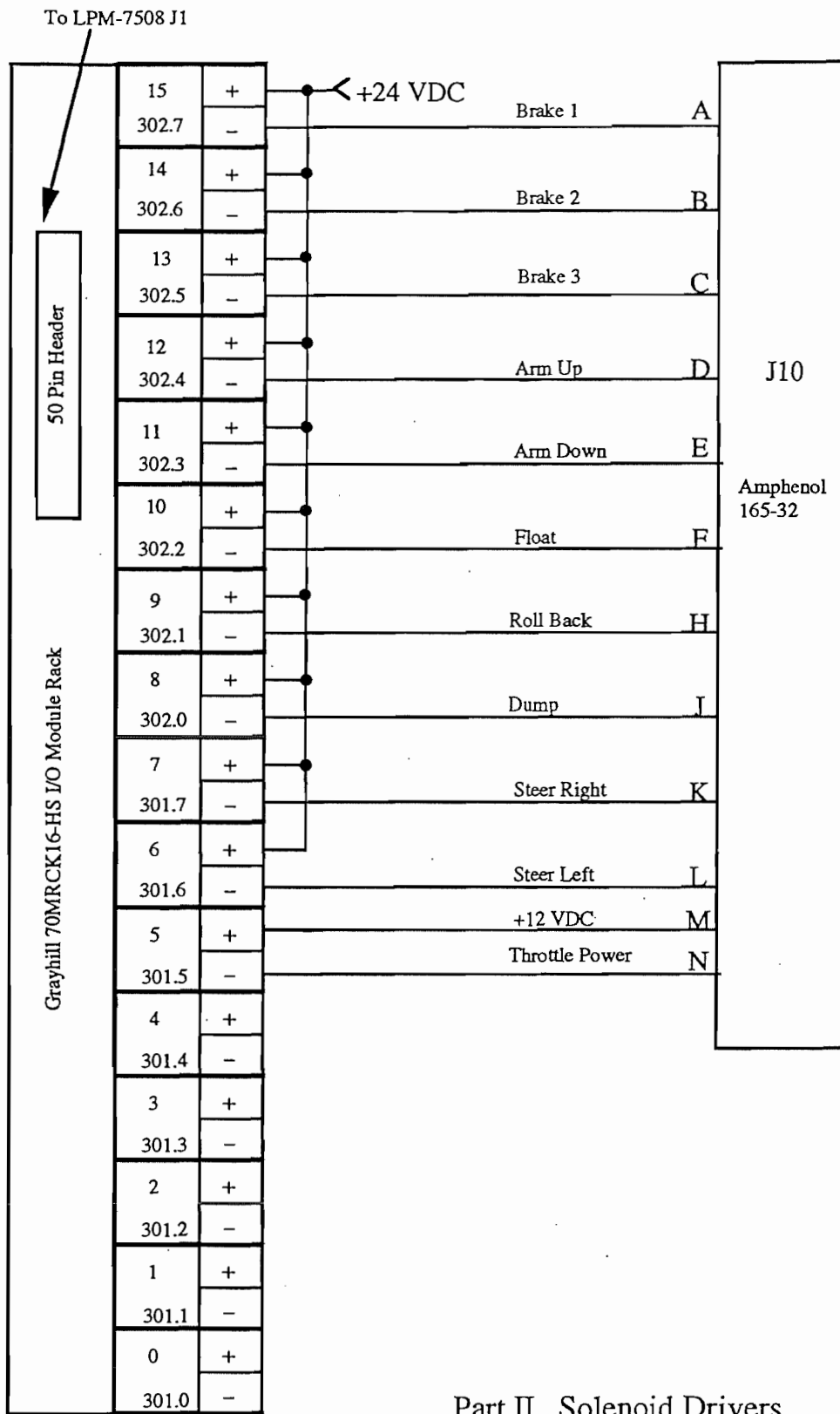


APPENDIX A4

Diagram of LCC Front Panel Interface

Part I. Analog/digital I/O





Part II. Solenoid Drivers

Copyright 2011, AHMCT Research Center, UC Davis



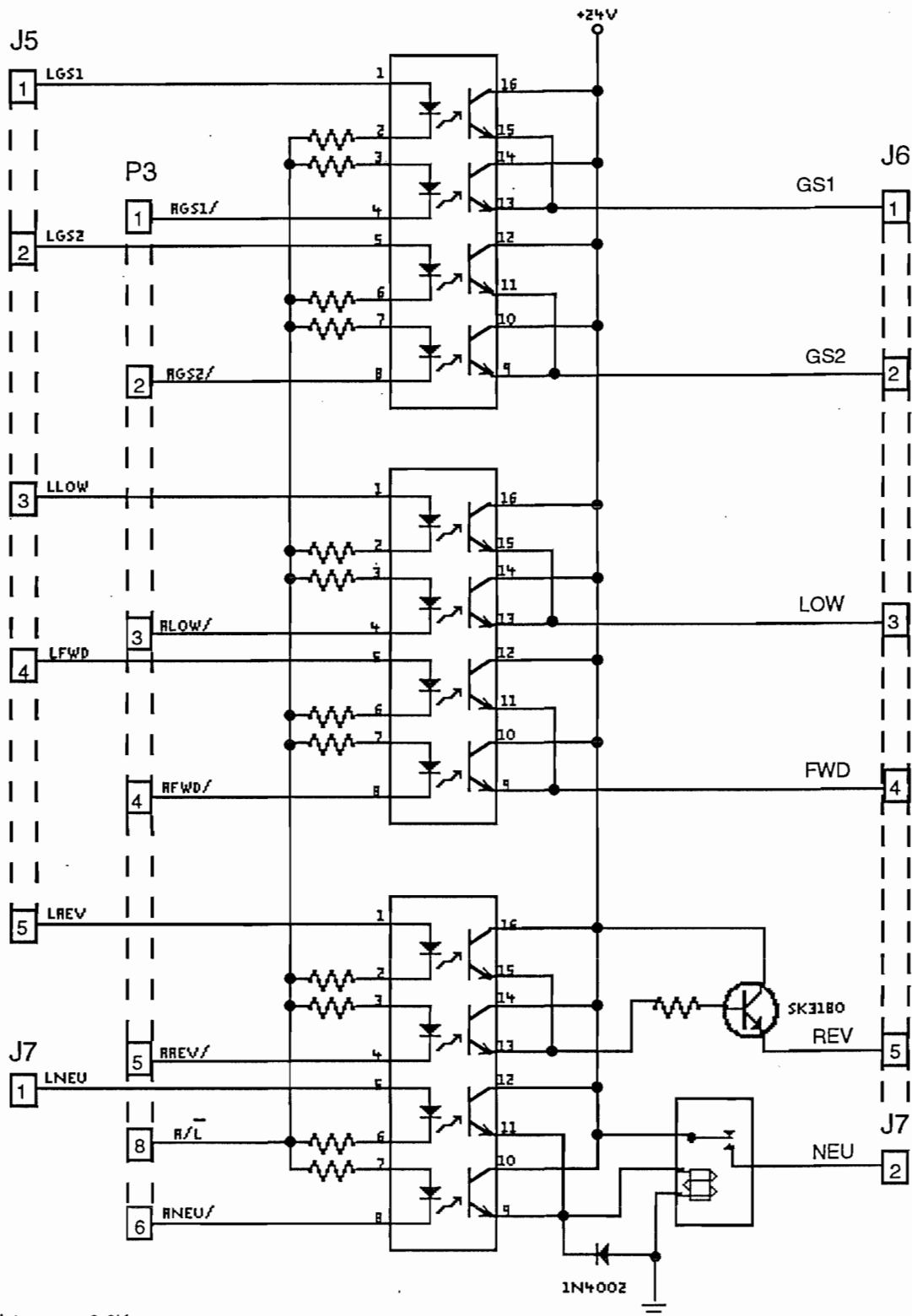
APPENDIX A5

LCC Connection Interface

Signal Name	Direction	Type	Specifications
Fuel Solenoid Valve	Output	Binary	Active-Low
Steering	Output	24v binary	Active high
Brake	Output	24v binary	Active high
Arm	Output	24v binary	Active high
Bucket	Output	24v binary	Active high
Throttle	Output	Analog	0-8 V
Transmission(drive)	Output	Binary	Active-Low
Transmission(reverse)	Output	Binary	Active-Low
Transmission(Gear, Low Bit)	Output	Binary	Active-Low
Transmission(Gear, High Bit)	Output	Binary	Active-Low
Low	Output	Binary	Active-Low
De Clutch	Output	Binary	Active-Low
Return to Dig	Output	Binary	Active-Low
Travel Height	Output	Binary	Active-Low
Dump Height	Output	Binary	Active-Low
Float	Output	Binary	Active-Low
Rattle	Output	Binary	Active-Low
Tachometer	Input	Analog	0-5 V

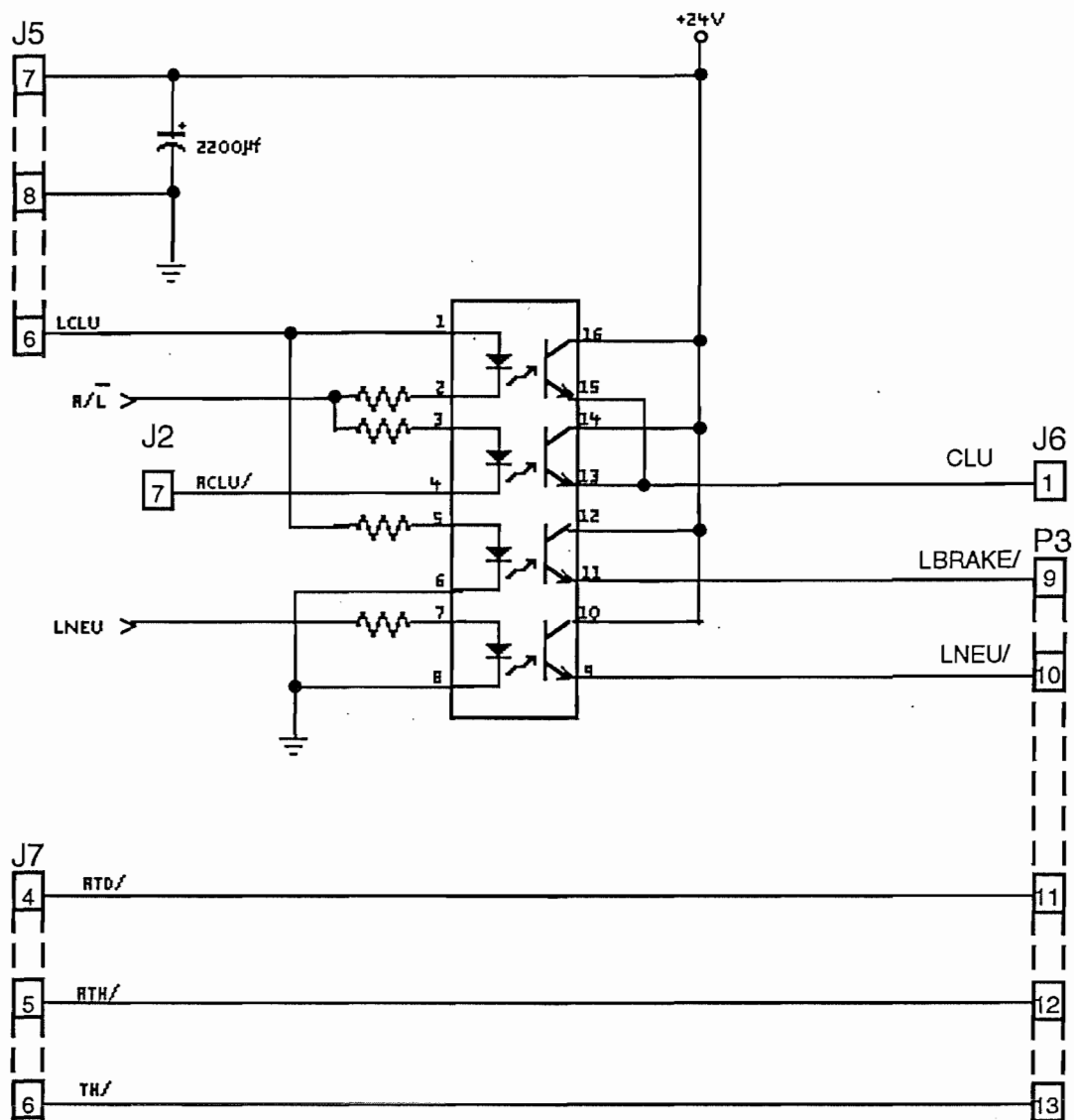
APPENDIX A6

Circuit of Transmission Multiplexer



All resistors are 2.2K

(Part 1)



All resistors are 2.2K

Circuit of Transmission Multiplexer (Part 2)

APPENDIX A7

Communication Protocol

1. OCC COMMAND = KEEP ALIVE COMMAND | VEHICLE COMMAND

KEEP LIVE COMMAND = " = " (bit 1)
 + Checksum (bit 2)
 + Linefeed (10) (bit 3)

VEHICLE COMMAND = " c " (bit 1)
 + DesiredSteering (bit 2 & 3)
 + DesiredBrakes (bit 4 & 5)
 + DesiredThrottle (bit 6 & 7)
 + DesiredArm (bit 8 & 9)
 + DesiredBucket (bit 10 & 11)
 + DesiredGearBitField (bit 12 & 13)
 + DesiredBitFunction (bit 14 & 15)
 + checksum (bit 16 & 17)
 + Linefeed (10) (bit 18)

Where,

DesiredSteering = Steering - H + Steering - L
 ("00" = Full Left, "80" = Center, "FF" = Full Right)

DesiredBrakes = Brakes - H + Brakes - L
 ("00" = No Brakes, "FF" = Full Brakes)

DesiredThrottle = Throttle - H + Throttle - L
 ("00" = Throttle Idle, "FF" = Full Throttle)

DesiredArm = Arm - H + Arm - L
 ("00" = Full Down, "80" = No Action, "FF" = Full Up)

DesiredBucket = Bucket - H + Bucket - L

("00" = Full Dump, "80" = No Action, "FF" = Full Back)

DesiredGearBitField =

b6	b7	+	b5	+	b4	+	b3b2 (Gear)	+	b1	+	b0
x	x		Declutch		Low		0 0 (1)		Reverse		Drive
							0 1 (2)				
							1 0 (3)				
							1 1 (4)				

DesiredBitfunction =

b6	b7	+	b5	+	b4	+	b3	+	b2	+	b1	+	b0
x	x		Dump		Travel		Return		Emergency		Float		Rattle
			Height		Height		To Dig		Stop				

Checksum = Checksum - H + Checksum - L

(XOR of preceding bytes) (0x55 wildcard checksum for testing)

2. LCC MESSAGES = "s" (bit 1)
+ StatusBitField (bit 2)
+ Checksum (bit 3 & 4)
+ Linefeed (10) (bit 3)

Where

StatusBirField = b7-b1 + b0

Spare AllSystemsOK

Checksum = Checksum - H + Checksum - L

(XOR of preceding bytes) (0x55 wildcard checksum for testing)

APPENDIX A8

I/O Address and Bit Definitions for LCC

I/O ADDRESS	SIGNAL	TYPE	CONNECTOR
70H	Throttle	ANALOG OUT	J1 - 19
3C - 3E CH0	Tachometer	ANALOG IN	J1 - 25
3C - 3E CH1	Steering Angle	ANALOG IN	J1 - 26
302H - 7	Brake Level 1	SSR OUT	J2 - A
302H - 6	Brake Level 2	SSR OUT	J2 - B
302H - 5	Brake Level 3	SSR OUT	J2 - C
302H - 4	Arm Up	SSR OUT	J2 - D
302H - 3	Arm Down	SSR OUT	J2 - E
302H - 2	Float	SSR OUT	J2 - F
302H - 1	Roll Back	SSR OUT	J2 - H
302H - 0	Dump	SSR OUT	J2 - J
301H - 7	Steer Right	SSR OUT	J2 - K
301H - 6	Steer Left	SSR OUT	J2 - L
301H - 5	Throttle Actuator Power	SSR OUT	J2 - M
304H - 0	Forward	OUT	J1 - 1
304H - 1	Reverse	OUT	J1 - 2
304H - 2	Neutral	OUT	J1 - 3
304H - 3	Gear Select 1	OUT	J1 - 4
304H - 4	Gear Select 2	OUT	J1 - 5
304H - 5	Clutch Engage	OUT	J1 - 6
304H - 6	Remote/Local	OUT	J1 - 7
305H - 0	Man/Remote	N	J1 - 15
305H - 1	LNEU/	N	J1 - 8
305H - 2	LBRAKE/	N	J1 - 9
305H - 3	AOK	N	J1 - 10
305H - 4	Return To Dig Sensor	N	J1 - 11
305H - 5	Return To Height Sensor	N	J1 - 13
305H - 6	Travel Height Sensor	N	J1 - 14
306H - 0	Standby LED	OUT	J1 - 16
306H - 1	Remote LED	OUT	J1 - 17

APPENDIX B1

Source code list for Operator Control Computer

```
/*.....  
*  
*                               OCC.C  
*   This is a control program for Operator Control Computer developed using Dynamic C   *  
*   with a PC compatible computer and down loaded to Z-world programmable controller   *  
*.....*/
```

```
#INT_VEC    CTC3_VEC    rtc_timer
```

```
#define THREE_SEC    30 /* Real time clock tick at 100 msec */
```

```
#define HUNDRAD_MSEC    1
```

```
#define TRUE    1
```

```
#define FALSE    0
```

```
#define OK    TRUE
```

```
#define BAD    FALSE
```

```
#define ON    TRUE
```

```
#define OFF    FALSE
```

```
#define ON_LINE    TRUE
```

```
#define OFF_LINE    FALSE
```

```
#define NO_ACTION    0X80
```

```
#define FULL_UP    0XFF
```

```
#define FULL_DOWN    0
```

```
#define FULL_BACK    0XFF
```

```
#define FULL_DUMP    0
```

```
#define FULL_LEFT    0
```

```
#define FULL_RIGHT    0XFF
```

```
#define CENTER    0X80
```

```
#define ARM_CENTER    0X80
```

```
#define BUCKET_CENTER    0X80
```

```
#define STEERING_CENTER    0X80
```

```
#define PLAY    0XB
```

```
/* mask values for OCC I/O */
```

```

/* I/O expansion board: */

#define EXP_PRT_A                0xFC00

/* not used */
#define EXP_PRT_B                0xFC01
/* not used */
/* LG I/O: */

#define LG_PRT_A                0x40
#define ENG_START_MASK          1
#define ENG_KILL_MASK           2
#define EMERG_STOP_MASK         4
#define RTN_DIG_MASK            8
#define TRVL_HGT_MASK           0x10
#define DMP_HGT_MASK            0x20
#define CNTL_BOX_MASK           0x40
#define ONLINE_IND_MASK         0x80

#define LG_PRT_B                0x42
#define TRNS_FWD_MASK           1
#define TRNS_REV_MASK           2
#define TRNS_SPD_LO_MASK        4
#define TRNS_SPD_HI_MASK        8
#define TRNS_LOW_MASK           0x10
#define TRNS_DCLCH_MASK         0x20
#define SYS_OK_MASK             0x40
#define COMM_OK_MASK            0x80

/* A/D channel definitions */
#define THRTL_CHNL              8          /* unipoar (0-2.5) */
#define BRK_CHNL                9
#define STR_CHNL                10
#define ARM_CHNL                11
#define BKT_CHNL                12

```

```

#define STATUS_MSG_SIZE 6
#define CMND_MSG_SIZE 18

char send_cnt, rec_cnt;
char send_buf[CMND_MSG_SIZE*2], rec_buf[STATUS_MSG_SIZE*2];
char kp_alive_buf[8];
char rtc_tick, comm_status, comm_timer, cmnd_timer;
char online, old_online, online_state;
char str, throttle, brake, trans, arm, bucket, bit_field;

void send_keep_alive_msg ();
void send_msg ();
void read_inputs ();
void init_io ();
void init_vars ();
char status_msg ();
char numToChat (char);
char charToNum (char);

main ()
{

#nodebug

    init_vars ();
    init_io ();
    while (1)
    {
        rec_cnt = 1;
        ser_rec_s0 (rec_buf, &rec_cnt);    /* get one character */
        while (1)    /* synching into LCC messages (looking for 's') */
        {
            if (rec_cnt == 0)
            {
                if (rec_buf[0] == 's')
                { /* found the header */

```

```

        rtc_tick = 0;          /* ignore the rest of status message */
        while (rtc_tick < 2); /* by allowing the timer tick twice, */
                                /* time to receive 5 more char < 100-200 msec */
        break;                /* got a valid status message */
    }
    else
    {
        rec_cnt = 1;
        ser_rec_s0 (rec_buf, &rec_cnt); /* get one character */
    }
}

if (rtc_tick)                /* send keep alive msg while looking for 's' */
{
    rtc_tick = 0;
    send_keep_alive-msg();
}
}

rtc_tick = 0;
rec_cnt = STATUS_MSG_SIZE;
ser_rec_s0 (rec_buf, &rec_cnt);

while (1)                    /* forever loop */
{
    if (rtc_tick)            /* on each tick do following */
    {
        rtc_tick = 0;
        comm_timer--;
        cmdnd-timer--;
        if (comm_timer == 0) /* comm link timeout */
        {
            /* turn off comm ok light */
            output(LG_PRT_B, inport (LG_PRT_B) & ~COMM_OK_MASK);
            output(LG_PRT_B, inport (LG_PRT_B) & ~SYS_OK_MASK);
            comm_timer = THREE_SEC; /* reset comm. timer */
            comm_status = BAD;
            online_state = OFF_LINE;
        }
    }
}

```

```

        output (LG_PRT_A, inport (LG_PRT_A) & ~ONLINE_IND_MASK);
        break;

    }
}
if (rec_cnt == 0)
{
    if (status_msg () == FALSE) /* decode status message */
        break;
    rec_cnt = STATUS_MSG_SIZE;
    ser_rec_s0 (rec_buf, &rec_cnt);
}
if (cmnd_timer == 0)
{
    /* time to send command */
    cmnd_timer = HUNDRAD_MSEC; /* reset command timer */
    read_inputs ();
    if (online != old_online)
    {
        /* online switch activated */
        if (old_online == ON) /* went through a high to low trans. */
        {
            if (online_state == OFF_LINE)
            {
                /* currently off line */
                if ((brake > (0XFF - PLAY)) && !(TRANS & 0x3))
                {
                    /* brake fully applied and tranmission in N */
                    online_state = ON_LINE;
                    output (LG_PRT_A, inport (LG_PRT_A)
                        | ONLINE_IND_MASK);
                }
            }
        }
        else
        {
            /* currently on line */
            online_state = OFF_LINE;
            output (LF_PRT_A, inport (LG_PRT_A) & ~ONLINE_IND_MASK);
        }
    }
}
old_online = online;

```



```

    }
    if (online_state == ON_LINE)
        send_msg ();
    else
        send_keep_alive_msg ();
    }
}
}
}

/* *****
* the following function will send keep alive
* messages to vehicle to keep communications
* going.
*
***** */

void send_keep_alive_msg ()
{
    while (send_cnt);          /* wait for old message to be sent */
    send_cnt = 4;
    ser_send_s0 (kp_alive_buf, &send_cnt);
}

/******
*
* the following function will send a vehicle
* command according to the state of the inputs.
*
***** */

void send_msg ()
{
    char chk_sum, i;
    chk_sum = 0;
    while (send_cnt);          /* wait for old message to be sent */
    send_buf [0] = 'c';

```

```

send_buf [1] = numToChar (str >> 4 );
send_buf [2] = numToChar (str & 0xF);
send_buf [3] = numToChar (brake >> 4);
send_buf [4] = numToChar (brake & 0xF);
send_buf [5] = numToChar (throttle >> 4);
send_buf [6] = numToChar (throttle & 0xF);
send_buf [7] = numToChar (arm >> 4 );
send_buf [8] = numToChar (arm & 0xF);
send_buf [9] = numToChar (bucket >> 4 );
send_buf [10] = numToChar (bucket & 0xF);
send_buf [11] = numToChar (trans >> 4 );
send_buf [12] = numToChar (trans & 0xF);
send_buf [13] = numToChar (bit_field >> 4 );
send_buf [14] = numToChar (bit_field & 0xF);
for ( i=0; i < 15; i++)
    chk_sum ^=send_buf [i];
send_buf [15] = numToChar (chk_sum >> 4);
send_buf [16] = numToChar (chk_sum & 0xF);
send_buf [17] = '\n';
send_cnt = CMND_MSG_SIZE;
ser_send_s0 (send_buf, &send_cnt);
}

/*****
*
* this functions will read all the input ports
* and saves their state into appropriate
* variables.
*
*****/

void read_inputs ()
{
    /* read all bits */
    bit_field = ~inport (LG_PRT_A) & 0x7F;
    trans = ~inport(LG_PRT_B) & 0x3F;    /* read trans bit */

```

```

/* read and scale analog ports (input rng: 0-4095, command rng: 0-255) */
/* steering, arm and bucket signal allow +- %5 uncertainty around */
/* center and each end. throttle and brake allow %5 around each end */

```

```

throttle = ad_rd ( (int) THRTL_CHNL ) / 16;
if ( throttle < PLAY)
    throttle = 0;
else if (throttle > 0xFF - PLAY)
    throttle = 0xFF;
else
    throttle -= PLAY;
brake = ad_rd ( (int) BRK_CHNL ) / 16
if (brake < PLAY)
    brake = 0;
else if (brake > 0xFF -PLAY)
    brake = 0xFF;
else
    brake -= PLAY;
str = ad_rd ( (int) STR_CHNL ) / 16;
if (STR > CENTER - PLAY && str < CENTER + PLAY)
    str = center;
else if ( str > 0xFF - PLAY)
    str = FULL_RIGHT;
else if (str < PLAY)
    str = FULL_LEFT;
else
    str -= PLAY;
arm = ad_rd ( (int) ARM_CHNL) / 16;
if (arm > CENTER - PLAY && arm < CENTER + PLAY)
    arm = CENTER;
else if ( arm > 0xFF - PLAY)
    arm = FULL_UP;
else if ( arm < PLAY)
    arm = FULL_DOWN;

else

```

```

    arm -= PLAY;
    bucket = ad_rd ( (int) BKT_CHNL ) /16;
    if (bucket > CENTER - PLAY && bucket < CENTER + PLAY)
        bucket = CENTER;
    else if ( bucket > 0xFF - PLAY)
        bucket = FULL_RIGHT;
    else if ( bucket < PLAY)
        bucket = FULL_LEFT;
    else
        bucket -= PLAY;
    /* decode state of "BOX ONLINE" signal */
    online = ( (bit_field & CNTL_BOX_MASK) == CNTL_BOX_MASK);
}

```

```

/******
 *
 * the following interrupt service routine called
 * every 100 msec as a result of rtc timeouts.
 *
 *****/

```

```

interrupt reti rtc_timer ()
{
    rtc_tick++;
}

```

```

/******
 *
 * the following function will initialize
 * the io devices on little giant and I/O
 * expansion module.
 *
 *****/

```

```

void init_io ()
{
    /* setup the digital I/O ports */
    /***** I/O expansion board bit assignments:

```

```

port A:
b0-b7 - spare
port B:
b0-b7 - spare
*****/

exp_init (_1,_1,_1,_1);          /* PPI all inputs */

/***** LG on-board bit assignments:
port A:
    b0 - engine start (input)
    b1 - engine kill (input)
    b2 - emergency stop (input)
    b3 - return to dig (input)
    b4 - travel height (input)
    b5 - dump height(input)
    b6 - control box on-line (input)
    b7 - control box on-line indicator (output)
port B:
    b0 - trans forward (input)
    b1 - trans reverse (input)
    b2 - trans speed low bit (input)
    b3 - trans speed high bit (input)
    b4 - trans low (input)
    b5 - trans declutch (input)
    b6 - system ok ( output)
    b7 - comm ok (output)
*****/

/* init. port A of little giant */
output (0x41, 0xcf); /* mode 3 */
output (0x41, 0x7f); /* all outputs */
output (0x41, 0x03); /* disable interrupt */

/* init. port B of little giant */
output (0x43, 0xcf); /* mode 3 */
output (0x43, 0x3f); /* all outputs */

```

```

    outport (0x43, 0x03); /* disable interrupt */
    /* turn off all LED's */
    outport (LG-PRT_A, 0x7f);
    outport (LG-PRT_B, 0x3f);

    /* setup the serial port 0 to communicate with RF modem */
    ser_init_s0 (4, 8);          /* 9600, 8 bits, no parity, 1 stop */

    /****
        timer 2 and 3 are cascaded. the following calls will cause the
        CTC counters to generate an interrupt every 100 msec
    ****/
    setctc(2,1,180,0);          /* 5 msec */
    setctc(3,2,20,1);          /* 20 times 5 msec */
}

/*****
 *
 * the following function will process a received
 * message from lcc and update appropriate flags
 *
 *****/
char status_msg ()
{
    char chksum, msg_chksum, i;

    chksum = 0;
    for (i = 0; i < STAUS_MSG_SIZE - 3 ; i++)
        chksum ^= rec_buf [i];          /* calculate the message checksum */
    msg_chksum = charToNum(rec_buf [STATUS_MSG_SIZE-3]) << 4
                | charToNum(rec_buf[STATUS_MSG_SIZE-2]);
    if ((rec_buf [0] == 's') && (rec_buf [STATUS_MSG_SIZE-1] == '\n')
        && ((chksum == msg_chksum) || (msg_chksum == 0x55)))
    {
        /* message is a valid lcc message */
        if (comm-status == BAD)

```

```

    {
        comm-status = OK;
        outport (LG_PRT_B, inport (LG_PRT_B) | COMM_OK_MASK);
    }
    if (rec_buf [2] == '1')
    {
        /* all system OK */
        outport (LG_PRT_B, inport (LG_PRT_B) | SYS_OK_MASK);
    }
    else
    {
        /* turn off "all system OK" indicator */
        outport (LG_PRT_B, inport (PG_PRT_B) & ~SYS_OK_MASK);
    }
    comm_timer = THREE_SEC; /* reset c ommunications timer */
    return (TRUE);
}
else
{
    return (FALSE);
}
}

```

```

/*****
*
*   the following function will initialize
*   all the global variables.
*
*****/

```

```

*****/

```

```

void init_vars ()
{

```

```

    char i;

```

```

    rec_cnt = 0;

```

```

    send_cnt = 0;

```

```

    rtc_tick = 0;

```

```

    for ( i = 0; i < CMND_MSG_SIZE*2; i++)
        send_buf [i] = 0;
    for ( i = 0; i < STATUS_MAG_SIZE*2; i++)
        rec_buf [i] = 0;
    kp_alive_buf [0] = '=';
    kp_alive_buf [1] = numTochar ('=' >> 4);
    kp_alive_buf [2] = numTochar ('=' & 0xF);
    kp_alive_buf [3] = '\n' ;
    comm_status = BAD;
    comm_timer = THREE_SEC;    /*comm error if no msg over ONE sec */
    cmnd_timer = HUNDRAD_MSEC;    /* send a command every 100 msec */
    online_state = OFF_LINE;
    online = old_online = OFF;
}

char numTochar (char val)
{
    if (val < 10);
        return (val + '0');
    else
        return (val - 10 + 'A');
}

char charToNum (char chr)
{
    if (chr < 65)
        return (chr - 48);
    else
        return (chr -55);
}

```


APPENDIX B2

Loader Control Computer Source Code List

```
/*      LCC.C LOADER CONTROL PROGRAM WITH RATTLE FEATURE      */
```

```
#include    "stdio.h"  
#include    "conio.h"  
#include    "dos.h"
```

```
/*      8259 Interrupt Controller ports and initialization values      */
```

```
#define      OCW2      0x20  
#define      ICW1      0x13  
#define      ICW2      0x08  
#define      ICW4      0x1d  
#define      ICW1_PORT  0x20  
#define      ICW2_PORT  0x21
```

```
/*      Timer 2 initialization values      */
```

```
#define      TIMER2_INT_NUM 0x0a  
#define      TIMER2_COUNT_REG 0x42  
#define      TIMER2_CONTROL_REG 0x43  
#define      TIMER2_MODE  0xb4  
#define      TIMER2_1MSEC_LSB 0xa9  
#define      TIMER2_1MSEC_MSB 0x04
```

```
/*      COM2 registers and initialization values      */
```

```
#define      COM2_IO_REG  0x2f8  
#define      T2_EOI      0x62  
#define      COM2_INT_NUM 0x0b
```

```

#define COM2_IER      0x2f9
#define COM2_IID      0x2fa
#define COM2_LCR      0x2fb
#define COM2_MCR      0x2fc
#define COM2_LSR      0x2fd
#define COM2_DLL      0x2f8
#define COM2_DLM      0x2f9
#define COM2_IN_IE    1
#define COM2_IO_IE    2
#define COM2_LC_VAL    3
#define COM2_MC_VAL    8
#define DLAB          0x80
#define COM2_9600_MSB  0
#define COM2_9600_LSB  12
#define COM2_EOI      0x63
#define RCV_INT       4
#define XMIT_INT       2

```

```

/* Analog to Digital converter port addresses */

```

```

#define ATOD_MUX      0x3c
#define ATOD_LSB      0x3d
#define ATOD_MSB      0x3e
#define ATOD_BUSY_MASK 0x80

```

```

/*Digital to Analog converter for throttle control */

```

```

#define THROTTLE_DAC  0x70

```

```

#define SOLENOID      0x302 /*High current digital outputs*/
#define SOLENOID2     0x301 /*More high current digital outputs*/
#define XMISSION      0x304 /*Low current digital outputs*/
#define MACHINE_INPUTS 0x305 /*Switches and other digital inputs*/
#define STATUS_OUTPUTS 0x306 /*REMOTE and STANDBY LED'S*/

```

```

/* Machine input bit values */

```

```

#define    LBRAKE      4
#define    LNEU        2
#define    REMOTE_SW    1
#define    STANDBY_MODE_OK 3
#define    REMOTE_MODE_OK 3
#define    REMOTE_MODE_MASK 7
#define    LCC_RTD      0x10
#define    LCC_TH        0x20
#define    LCC_DH        0x40
#define    OCC_RTD      8
#define    OCC_TH        0x10
#define    OCC_DH        0x20
#define    OCC_FLOAT     2
#define    OCC_EMERG_STOP 4
#define    RATTLE        1

```

```

/*    Transmission output bits    */

```

```

#define    GEAR_MASK    0xc
#define    DIRECTION_MASK 3
#define    DIR_NEUTRAL    4
#define    REMOTE_ENABLE 0x40
#define    CLUTCH_ENGAGE 0x20
#define    CLUTCH_DISENGAGE 0xdf

```

```

/*    SOLENOID output bits    */

```

```

#define    ARM_FLOAT    0xc
#define    NO_FLOAT     0xbf
#define    ARM_DOWN     8
#define    ARM_UP       0x10
#define    ARM_STOP     0xe3
#define    ROLL_BACK    2
#define    DUMP          1
#define    BUCKET_STOP  0xfc

```

```

#define BRAKE_0 0x1f
#define BRAKE_1 0x80
#define BRAKE_2 0xc0
#define BRAKE_3 0xa0

```

```

/* SOLENOID2 output bits */

```

```

#define THROTTLE_POWER 0x20
#define STEER_RIGHT 0x80
#define STEER_LEFT 0x40
#define STEER_0 0x3f

```

```

/* STATUS output bits */

```

```

#define STANDBY_LED 1
#define REMOTE_LED 2

```

```

/* Switching parameters for analog functions */

```

```

#define UP_SWITCH_0 140
#define DOWN_SWITCH_0 122
#define RB_SWITCH_0 117
#define DUMP_SWITCH_0 101
#define BRAKE_1_SWITCH 232
#define BRAKE_2_SWITCH 240
#define BRAKE_3_SWITCH 248
#define THROTTLE_OFS 9
#define THROTTLE_SLOPE 3
#define STEERING_SLOPE 12.1

```

```

/* GLOBAL VARIABLES */

```

```

int desired_steering, steering_angle, desired_brake, desired_throttle;
int desired_arm, desired_bucket, desired_gear, desired_direction, loop_timer;

```

```

unsigned t2_oldvect,com2_oldvect;
unsigned com2_input_timer,com2_output_timer,aux_timer,rattle_timer,rtd;
unsigned up_switch=UP_SWITCH_0,down_switch=DOWN_SWITCH_0;
char com2_input_buffer[20],input_block_buffer[21];
char *output_msg="s0172\n\0";
char *out_buf_ptr_;
char *block_char_1;
unsigned input_count,block_count;
unsigned solenoid,solenoid2,xmission;
unsigned rb_switch=RB_SWITCH_0,dump_switch=DUMP_SWITCH_0,brake_level=3;
unsigned arm_height,desired_height,auto_arm,float_flag,rattle_flag;

```

```

void _cdecl _interrupt _far timer2_interrupt(void);
void _cdecl _interrupt _far com2_interrupt(void);
void set_int_vect(void);
void init_8259(void);
void init_timer2(void);
void init_com2(void);
void com2_out(void);
int decode_input(void);

```

```

main()
{
    char com2_;
    int v,s,high_loop,prt_count;
    unsigned arm_sensor;
    printf("set int vect\n");
    set_int_vect(); /*Set DOS interrupt vectors for COM2 and Timer2 */
    printf("init 8259\n");
    init_8259(); /*Initialize 8259 interrupt controller */
    printf("init timer2\n");
    init_timer2(); /* Set timer 2 to produce a 1 msec interrupt */
    printf("init com2\n");
    init_com2(); /* intitalize COM2 serial IO port */

```

```

/* INITIALIZE VARIABLES */

```

```

        solenoid=0;
        block_count=0;
        block_char_1=&input_block_buffer[0];
        com2_input_timer=300;
        com2_output_timer=400;
        aux_timer=5000;
        rattle_timer=0;
        rattle_flag=0;
        prt_count=0;
        out_buf_ptr_=output_msg;
    /*
    /
    /  BEGIN PROGRAM FUNCTIONS
    /
    */

        if((inp(MACHINE_INPUTS))&REMOTE_SW)goto STANDBY_ENTRY_POINT;
            /* If Remote Switch is engaged at reset, goto standby mode */

    /* BEGINNING OF MANUAL MODE */

        MANUAL_ENTRY_POINT:
        printf("MANUAL MODE\n\n");
        outp(COM2_IER,COM2_IN_IE);
        while((inp(MACHINE_INPUTS)&STANDBY_MODE_OK)!=STANDBY_MODE_OK)
        {
            /* Stay in manual mode until Remote switch is engaged and vehicle transmission
            lever is in neutral position. */

        /* disable all remote control functions */
            outp(SOLENOID,0);
            outp(SOLENOID2,0);
            outp(XMISSION,0);
            outp(STATUS_OUTPUTS,0);

        /* Send 'system OK' message to field unit every 400 msec */

```

```

        if(com2_output_timer==0){
            com2_out();
            com2_output_timer=400;
        }
    }

/* BEGINNING OF STANDBY MODE */

STANDBY_ENTRY_POINT:
    printf("STANDBY MODE\n\n");
    /*Set full brakes, transmission neutral, remote mode */
    outp(SOLENOID,BRAKE_3);
    outp(SOLENOID2,0);
    outp(XMISSION,(DIR_NEUTRALIREMOTE_ENABLE));
    outp(STATUS_OUTPUTS,STANDBY_LED); /*turn on STANDBY LED */
    while((inp(MACHINE_INPUTS)&LBRAKE))
    {
        /* Wait for vehicle brake release while continuing to monitor Remote switch. */

        if((inp(MACHINE_INPUTS)&REMOTE_SW)==0)goto MANUAL_ENTRY_POINT;
    }

    /* Wait for receipt of 'keep alive' message while continuing to
       monitor Remote switch and send 'system OK' message. */

    while(v!=1)
    {
        while(block_count==0)
        {
            if((inp(MACHINE_INPUTS)&REMOTE_SW)==0)goto MANUAL_ENTRY_POINT;
        }
        v=decode_input(); /*return value of 1 is 'keep alive msg'*/
        block_count=0;
    }
    v=0;

```

RESTART_ENTRY_POINT:

```
/*Set full brakes, transmission neutral, remote mode */
    outp(SOLENOID,BRAKE_3);
    outp(SOLENOID2,0);
    outp(XMISSION,(DIR_NEUTRALIREMOTE_ENABLE));
    outp(STATUS_OUTPUTS,STANDBY_LED);/*turn on STANDBY LED */
/* Wait for receipt of 'command' message while continuing to
    monitor Remote switch and send 'system OK' message. */
    v=0;
    while(v!=2)
    {
/* return value of 2 indicates valid 'command' msg*/
        if(block_count==0)
        {
            if((inp(MACHINE_INPUTS)&REMOTE_SW)==0)goto MANUAL_ENTRY_POINT;
            if(com2_output_timer==0)
            {
                com2_out();
                com2_output_timer=400;
            }
        }
        else
        {
            v=decode_input();
            block_count=0;
        }
    }

/* BEGINNING OF REMOTE MODE */
```

REMOTE_ENTRY_POINT:

```
printf("REMOTE MODE\n");
com2_input_timer=500;
rtd=0;
arm_height=0;
auto_arm=0;
```



```

float_flag=0;
outp(STATUS_OUTPUTS,REMOTE_LED);    /*Turn on REMOTE LED */

/*Stay in remote mode until Remote switch is deactivated, vehicle brake
    pedal is depressed, or vehicle transmission lever is moved. */
while((inp(MACHINE_INPUTS)&REMOTE_MODE_MASK)==REMOTE_MODE_OK)
{
    outp(ATOD_MUX,1); /*begin machine angle conversion */
    if(high_loop<loop_timer)high_loop=loop_timer;
    loop_timer=0;
    /* If valid 'command' block is not received within 500 msec, goto STANDBY */
    if(com2_input_timer==0){
        printf("INPUT TIMEOUT\n");
        if(v==2)printf("CHECKSUM ERROR\n");
        goto RESTART_ENTRY_POINT;
    }
}
/* Send 'system OK' message every 400 msec */
if(com2_output_timer==0)
{
    com2_output_timer=400;
    com2_out();
}

/* when a new message is received from the field unit, call decode_input */
if(block_count)
{
    v=decode_input();
    /*if it is a valid 'command' message, reset input timer */
    if((v=decode_input())==2)com2_input_timer=300;
    if(v==10)
    {
        printf("EMERGENCY STOP\n");
        goto STANDBY_ENTRY_POINT;
    }
    block_count=0;
}

```

```

/* set gear and direction */
xmission=desired_gear+desired_directionIREMOTE_ENABLE;
while(inp(ATOD_MSB)&ATOD_BUSY_MASK);/*wait for conversion complete */
/*read machine angle from ATOD converter */
steering_angle=((inp(ATOD_MSB)&0x0f)<<8)|inp(ATOD_LSB);
/* disengage clutch if brake level > 1, otherwise engage it */
if(brake_level>1)xmission&=CLUTCH_DISENGAGE;
else xmission|=CLUTCH_ENGAGE;
outp(XMISSION,(char)xmission); /* activate desired transmission bits */

/* check to see if steering angle is within 8 units of correct value.
    If so, disable both steering solenoids, otherwise do nothing. */
s=steering_angle-desired_steering;
if((s<8)&&(s>-8))solenoid2&=STEER_0;
if(auto_arm==2)
{

if(arm_height<desired_height)solenoid=(solenoid&ARM_STOP)|ARM_UP;
    else
if(arm_height>desired_height)solenoid=(solenoid&ARM_STOP)|ARM_DOWN;
    else solenoid&=ARM_STOP,auto_arm=0;
}
arm_sensor=((inp(MACHINE_INPUTS))&(LCC_TH|LCC_DH));
if(arm_sensor==LCC_TH)arm_height=1;
else if(arm_sensor==LCC_DH)arm_height=3;
else
{
    if(arm_height==1)
    {
        if(solenoid&ARM_UP)arm_height=2;
        else if(solenoid&ARM_DOWN)arm_height=0;
    }
    else if(arm_height==3)
    {
        if(solenoid&ARM_UP)arm_height=4;
        else if(solenoid&ARM_DOWN)arm_height=2;
    }
}

```

```

    }

}

if((float_flag)&&!(auto_arm))solenoid=(solenoid&ARM_STOP)|ARM_FLOAT;
if(!(float_flag))solenoid&=NO_FLOAT;

outp(SOLENOID,(char)solenoid);      /* activate SOLENOID bits */
outp(SOLENOID2,(char)solenoid2);

/* send desired throttle position to DAC channel 0 */
outp(THROTTLE_DAC,(char)desired_throttle);
}

/* When one of the conditions necessary for remote operation is not
met, return to the STANDBY mode. */

printf("MACHINE INPUTS %x\n", (inp(MACHINE_INPUTS)&REMOTE_MODE_MASK));
goto STANDBY_ENTRY_POINT;
}

/* com2_out() sends the 'system OK' message to the field unit.
This is not an interrupt driven function due to problems in the ZT8809's
interrupt system. */

void com2_out(void)
{
    out_buf_ptr_=output_msg;
    while(*out_buf_ptr_)
    { /*send characters until NULL char is reached */
        while((inp(COM2_LSR)&0x20)==0); /*wait for XMTR empty */
        outp(COM2_IO_REG,*out_buf_ptr_); /*send character*/
        out_buf_ptr_++;
    }
}

/* This is the timer 2 interrupt service routine. Each of the timer variables
is decremented every millisecond until a value of 0 is reached. */

```

```

void _cdecl _interrupt _far timer2_interrupt(void)
{
    if(com2_input_timer)com2_input_timer--;
    if(com2_output_timer)com2_output_timer--;
    if(aux_timer)aux_timer--;
    if(rattle_timer)rattle_timer--;
    if(loop_timer<10000)loop_timer++;
    outp(OCW2,T2_EOI); /*RESET INTERRUPT FLAG*/
    _asm sti;
}

```

/* This is the COM2 interrupt service routine. When a character is received, it is placed in the input buffer. When the received character is a new line character, or the input buffer is full, the characters received up to that point are transferred to the block buffer and the block count is set to show the number of characters in the buffer. The input count is then set to 0 and the process starts over. */

```

void _cdecl _interrupt _far com2_interrupt(void)
{
    unsigned i;
    if((i=inp(COM2_IID))==RCV_INT)
    {
        com2_input_buffer[input_count]=inp(COM2_IO_REG);/* get character */
        input_count++; /*increment input count */
        if((com2_input_buffer[input_count-1]=='\n')|(input_count>19))
        {
            /* transfer characters to block buffer */
            for(i=0;i<input_count;i++)input_block_buffer[i]=com2_input_buffer[i];
            block_count=input_count; /*set block count */
            input_count=0; /*zero input count */
        }
    }
    else outp(COM2_IER,COM2_IN_IE);
    outp(OCW2,COM2_EOI);
    _asm sti;
}

```

```

    }

/* Set DOS interrupt vectors for COM2 and Timer2 to new service routines */

void set_int_vect(void)
{
    _dos_setvect(TIMER2_INT_NUM,timer2_interrupt);
    _dos_setvect(COM2_INT_NUM,com2_interrupt);
}

/* Initialize 8259 interrupt controller */

void init_8259(void)
{
    unsigned mask;
    _asm cli;
    outp(ICW1_PORT,ICW1);
    outp(ICW2_PORT,ICW2);
    outp(ICW2_PORT,ICW4);
    mask=inp(ICW2_PORT);
    outp(ICW2_PORT,0);
    _asm sti;
}

/* Initialize Timer 2 to produce an interrupt every 1 millisecond */

void init_timer2(void)
{
    outp(TIMER2_CONTROL_REG,TIMER2_MODE);
    outp(TIMER2_COUNT_REG,TIMER2_1MSEC_LSB);
    outp(TIMER2_COUNT_REG,TIMER2_1MSEC_MSB);
}

/* Initialize COM2 serial IO port */

void init_com2(void)

```

```

{
    outp(COM2_LCR,COM2_LC_VALIDLAB);
    outp(COM2_DLL,COM2_9600_LSB); /* 9600 Baud */
    outp(COM2_DLM,COM2_9600_MSB);
    outp(COM2_LCR,COM2_LC_VAL); /* 8 bit characters, 1 stop bit, no parity */
    outp(COM2_IER,COM2_IN_IE); /* enable 'recieve buffer full' interrupt */
    outp(COM2_MCR,COM2_MC_VAL);
    inp(COM2_IO_REG);
}

```

/* The decode_input() function evaluates the received input block and returns a value as follows: 1=valid 'keep alive' message, 2=valid 'command' message, -1=incorrect character count, -2=checksum error, -3=first character was something other than '=' or 'c', 10=Emergency Stop. If the input block is a valid command message, the function decodes it and sets the appropriate bits in the system control variables . */

```

int decode_input(void)
{
    char *ptr;
    unsigned return_val=0,check_sum,i,ary[20];
    float steering_temp,throttle_temp;
    switch(*block_char_1
    {
    case '=': /* 'keep alive message' */
        if(block_count==4)return_val=1;      /*return 1 if char count=4 */
        else return_val=-1;    /* if char count != 4, return -1 */
        break;
    case 'c':    /* 'command message */
        if(block_count!=18)
        {
            return_val=-1; /*return -1 if char count != 18 */
            break;
        }
        ptr=block_char_1;
        ary[0]=*ptr;

```

```

for(i=1;i<17;i++)
{
    /*convert ASCII coded hex into 4 bit hex */
    ary[i]=*(ptr+i);
    if(ary[i]>64)ary[i]=ary[i]-7;
}
/* assemble each pair of hex nibbles into 8 bit binary byte */
for(i=1;i<9;i++)ary[i]=((ary[2*i-1]<<4)&0xf0)|(ary[2*i]&0xf);
check_sum=0; /*compute checksum */
for(i=0;i<15;i++)check_sum^=*(ptr+i);
check_sum^=ary[8]; /* XOR checksum with input checksum */
if(check_sum){ /* if result != 0, checksum error*/
    return_val=-2;
    break;
}

return_val=2; /*if checksum OK, return 2 */
desired_steering=ary[1];
desired_brake=ary[2];
desired_throttle=ary[3];
desired_arm=ary[4];
desired_bucket=ary[5];

/* Check for Emergency Stop Switch */
if(ary[7]&OCC_EMERG_STOP)return_val=10;

/*move gear selection from bits 2&3 to bits 3&4 */
desired_gear=(ary[6]&GEAR_MASK)<<1;
desired_direction=ary[6]&DIRECTION_MASK; /* get desired direction */

/* if neither FWD or REV selected, set direction NEUTRAL */
if(desired_direction==0)desired_direction=4;

/*Joystick function selection.
If the command value is between the switch points, the solenoids
are disabled. If the value exceeds one of the base switch points,
the switch point is moved 2 units toward the center to provide

```

hysteresis and the appropriate solenoid is activated. */

```
if(desired_arm>up_switch)
{
    up_switch=UP_SWITCH_0-2;
    down_switch=DOWN_SWITCH_0;
    solenoid=(solenoid&ARM_STOP)|ARM_UP;
    auto_arm=0;
    float_flag=0;
}
else if(desired_arm<down_switch)
{
    up_switch=UP_SWITCH_0;
    down_switch=DOWN_SWITCH_0+2;
    solenoid=(solenoid&ARM_STOP)|ARM_DOWN;
    auto_arm=0;
    float_flag=0;
}
else
{
    up_switch=UP_SWITCH_0;
    down_switch=DOWN_SWITCH_0;
    solenoid&=ARM_STOP;
}

if(ary[7]&&RATTLE)
{
    if(!(rattle_timer))
    {
        if(rattle_flag)solenoid=solenoid^3;
        else
        {
            rattle_flag=1;
            solenoid=(solenoid&BUCKET_STOP)|ROLL_BACK;
        }
        rattle_timer=200;
    }
}
```



```

    }
}
else
{
    rattle_flag=0;
    if(desired_bucket>rb_switch){
        rb_switch=RB_SWITCH_0-2;
        dump_switch=DUMP_SWITCH_0;
        solenoid=(solenoid&BUCKET_STOP)IROLL_BACK;
        rtd=0; /*terminate return to dig if enabled */
    }

    else if(desired_bucket<dump_switch)
    {
        rb_switch=RB_SWITCH_0;
        dump_switch=DUMP_SWITCH_0+2;
        solenoid=(solenoid&BUCKET_STOP)IDUMP;
        rtd=0;          /*terminate return to dig if enabled */
    }
    else
    {
        rb_switch=RB_SWITCH_0;
        dump_switch=DUMP_SWITCH_0;
        solenoid&=BUCKET_STOP;
    }
}

switch(rtd)
{
    case 0: /*No return to dig action--wait for button press */
        if(ary[7]&OCC_RTD)rtd=1;
        break;

    case 1: /*rtd button pressed, wait for release */
        if(ary[7]&OCC_RTD)break;
        /*determine if bucket should roll back or dump */
        if(!((inp(MACHINE_INPUTS))&LCC_RTD))rtd=2;          /*dump */
        else rtd=3;      /*roll back */

    case 2:          /*bucket must first dump to clear sensor and then roll back */

```

```

if(!((inp(MACHINE_INPUTS))&LCC_RTD))solenoid=(solenoid&BUCKET_STOP)IDUMP;
    else rtd=3;
    break;
case 3: /*bucket must roll back until sensor encountered */
    if(!((inp(MACHINE_INPUTS))&LCC_RTD)
        {
            solenoid&=BUCKET_STOP;
            rtd=0; /* bucket in dig position */
        }
    else solenoid=(solenoid&BUCKET_STOP)IROLL_BACK;
    break;
default:
    rtd=0;
    solenoid&=BUCKET_STOP;
}

switch(auto_arm)
{
case 0: /*No movement--wait for TH or DH to be pressed */
    if(ary[7]&(OCC_THIOCC_DH)){
        auto_arm=1;
        if(ary[7]&OCC_TH){
            if(arm_height)desired_height=0;
            else desired_height=1;
        }
        else if(arm_height<3)desired_height=3;
        else desired_height=2;
    }
    break;
case 1: /*TH or DH pressed, wait for release */
    if(!((ary[7]&(OCC_THIOCC_DH)))auto_arm=2;
    break;
case 2: /*in motion */
    break;
default:

```

```

        auto_arm=0;
    }
    if((ary[7]&OCC_FLOAT)&&!(float_flag))
    {
        float_flag=1;
        if(arm_height)
        {
            desired_height=0;
            auto_arm=2;
        }
    }

```

/*Brake Level Selection

The variable brake_level determines the current braking level. The switching points of that level are adjusted so as to widen the current level to provide hysteresis. If the value of desired_brake is outside of one of these switch points, the value of brake_level is incremented or decremented by 1.*/

```

switch(brake_level)
{
case 3: /* Highest level, check lower switch point only. */
    if(desired_brake<(BRAKE_3_SWITCH-2))
    {
        brake_level=2; /*move to brake level 2 */
        solenoid=(solenoid&BRAKE_0)|BRAKE_2;
    }
    /* otherwise, remain in level 3 */
    else solenoid=(solenoid&BRAKE_0)|BRAKE_3;
    break;
case 2:
    if(desired_brake<(BRAKE_2_SWITCH-2))
    {
        brake_level=1; /* move down to level 1 */
        solenoid=(solenoid&BRAKE_0)|BRAKE_1;
    }

```

```

    }
    else if(desired_brake>=(BRAKE_3_SWITCH+1))
    {
        brake_level=3;          /* move up to level 3 */
        solenoid=(solenoid&BRAKE_0)|BRAKE_3;
    }

    /* otherwise, remain in level 2 */
    else solenoid=(solenoid&BRAKE_0)|BRAKE_2;
    break;
case 1:
    if(desired_brake<(BRAKE_1_SWITCH-2))
    {
        brake_level=0; /*move down to level 0 */
        solenoid=solenoid&BRAKE_0;
    }
    else if(desired_brake>=(BRAKE_2_SWITCH+1))
    {
        brake_level=2; /* move up to level 2 */
        solenoid=(solenoid&BRAKE_0)|BRAKE_2;
    }

    /* otherwise, remain in level 1 */
    else solenoid=(solenoid&BRAKE_0)|BRAKE_1;
    break;
case 0: /* no braking action at all */
    if(desired_brake>(BRAKE_1_SWITCH+1))
    {
        brake_level=1; /*move up to level 1 */
        solenoid=(solenoid&BRAKE_0)|BRAKE_1;
    }

    /* otherwise, remain in level 0 */
    else solenoid=solenoid&BRAKE_0;
    break;
default:brake_level=3;
}

/* Compute throttle position value */

```

```

throttle_temp=255-(desired_throttle-THROTTLE_OFS)*THROTTLE_SLOPE;
if(throttle_temp<0)throttle_temp=0;
if(throttle_temp>255)throttle_temp=255;
desired_throttle=(int)throttle_temp;
solenoid2|=THROTTLE_POWER; /*Apply power to throttle actuator */

/* Compute desired steering angle from input value */
steering_temp=desired_steering*STEERING_SLOPE;
desired_steering=(int)steering_temp;
/* if desired steering differs from machine angle by 40 units
   or more, activate appropriate steering solenoid. */
if((desired_steering-
steering_angle)>40)solenoid2=(solenoid2&STEER_0)|STEER_RIGHT;
   else if((desired_steering-steering_angle)<-
40)solenoid2=(solenoid2&STEER_0)|STEER_LEFT;
   break;
default:return_val=-3;}
return return_val;
}

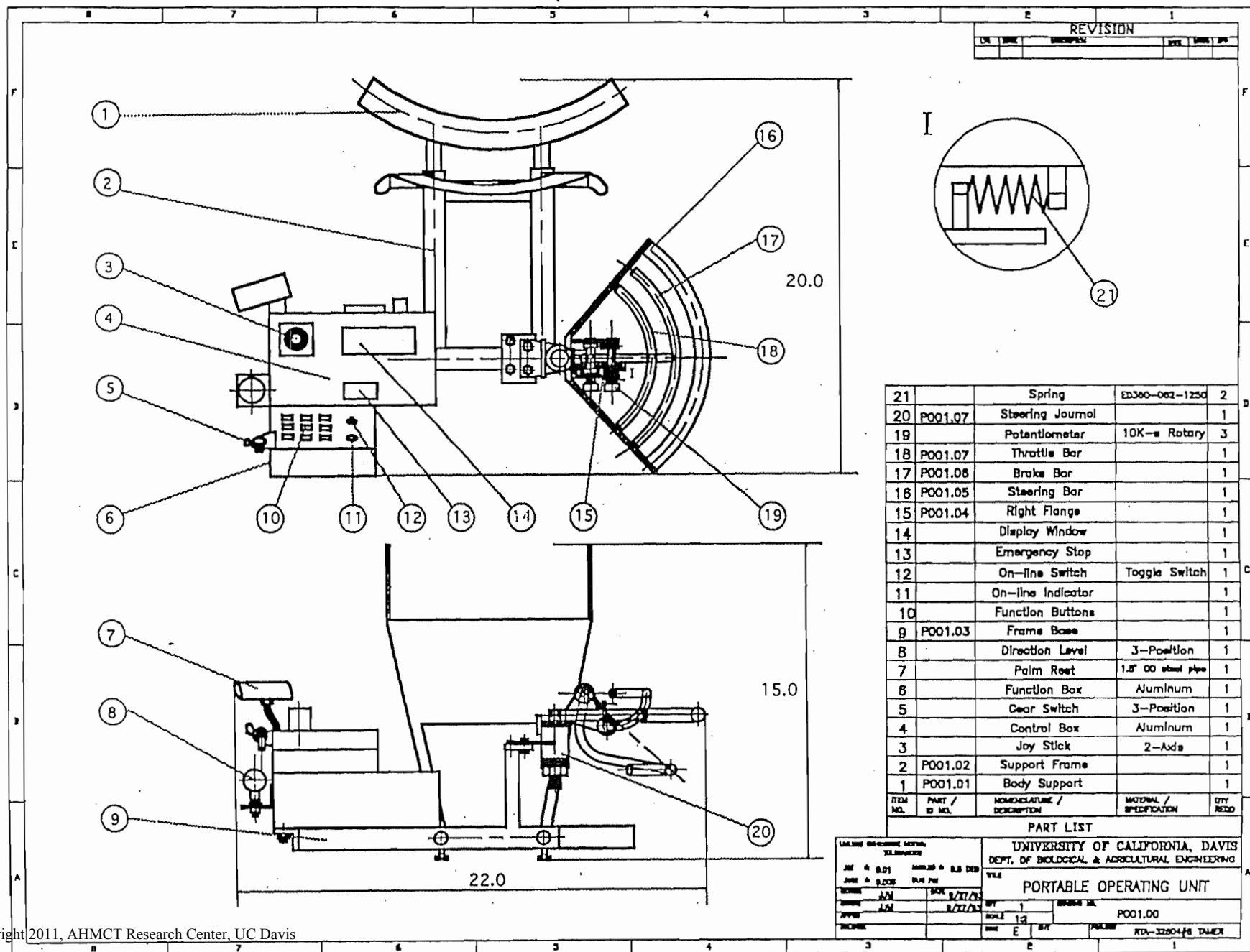
```

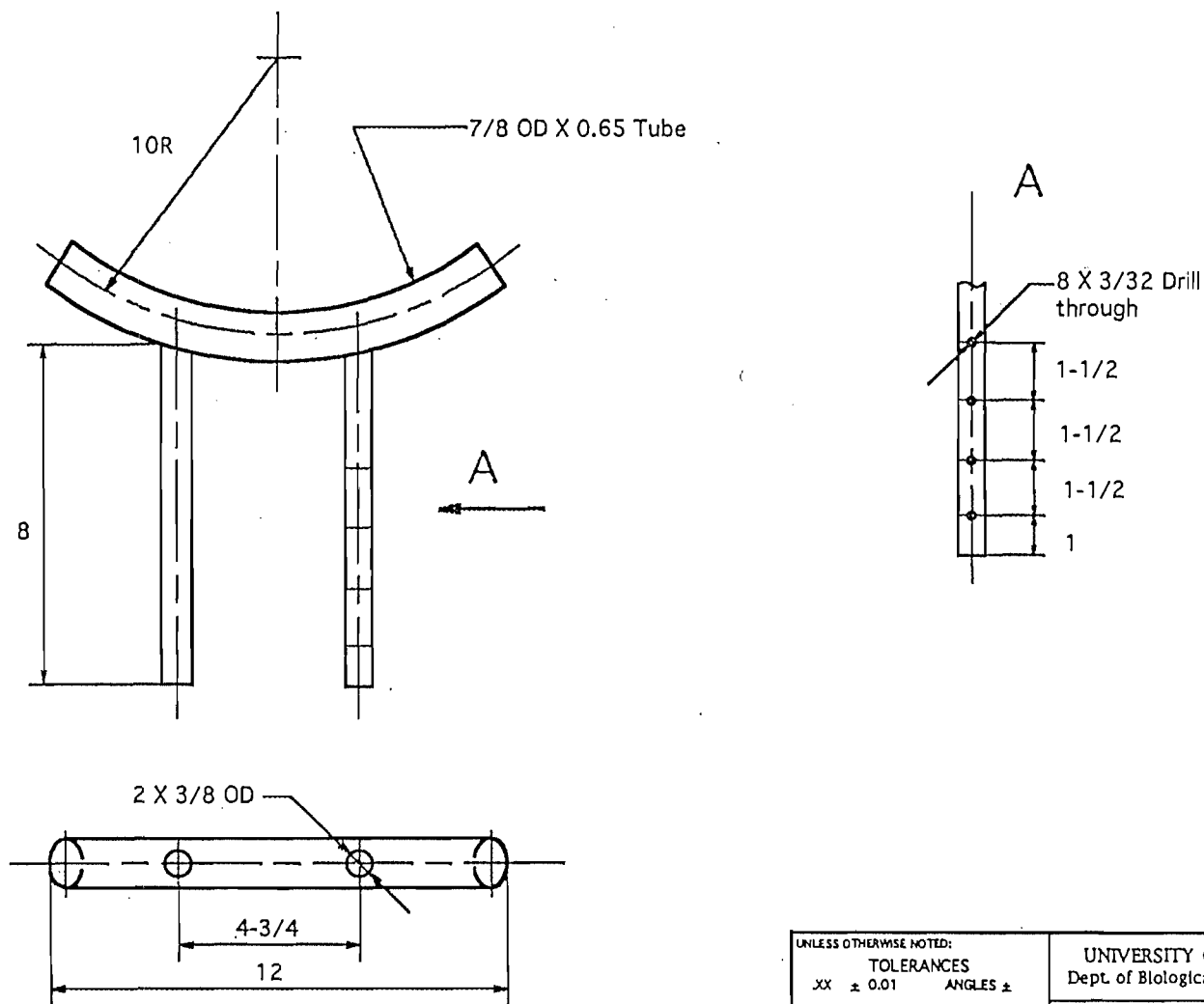
APPENDIX C1

Mechanical Drawings for Portable Unit

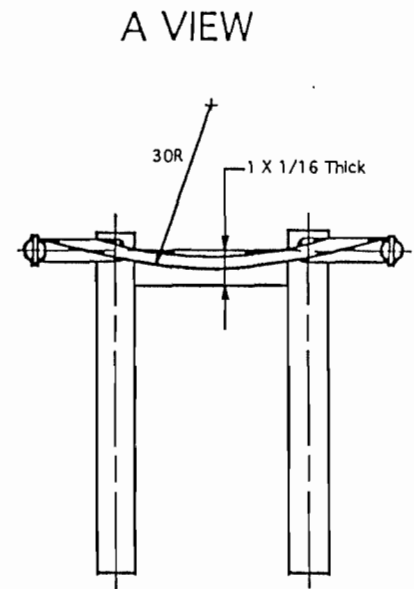
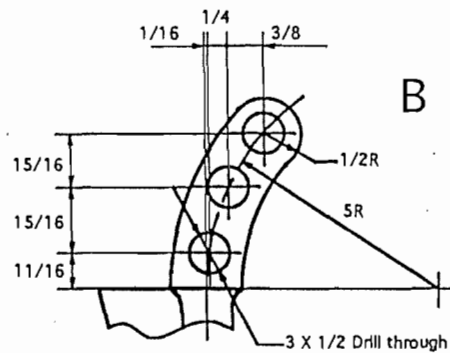
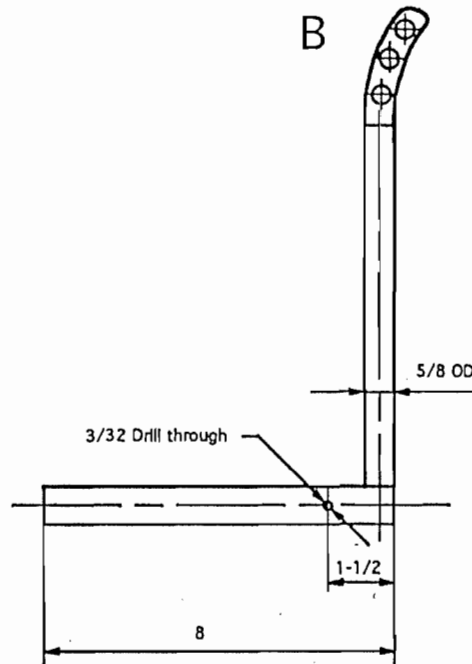
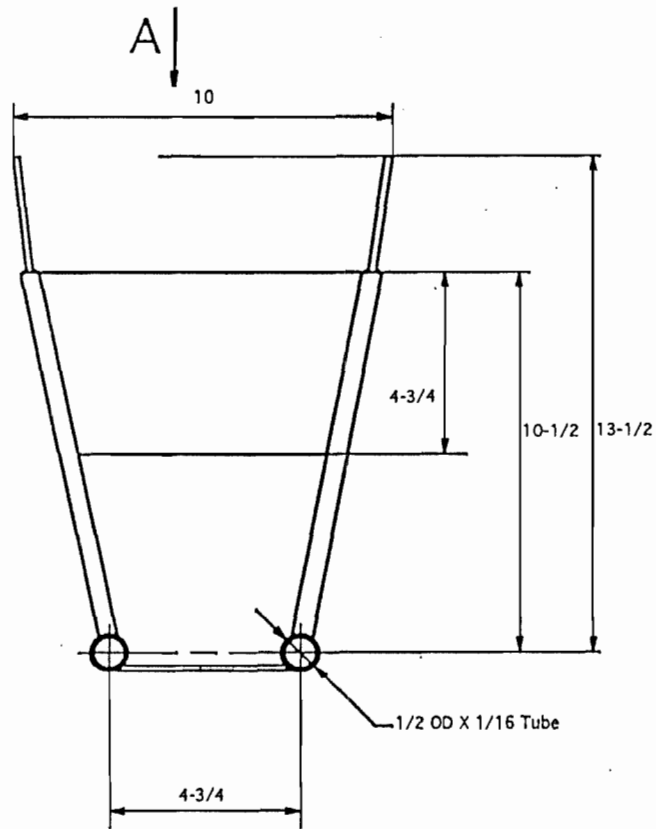
NOTE:

All units in the drawings included in this appendix are in inches. To convert to millimeters, multiply 25.400.



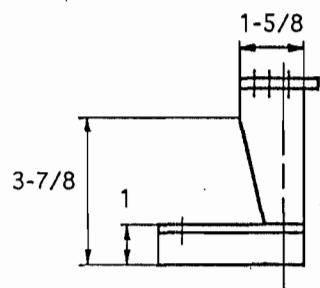
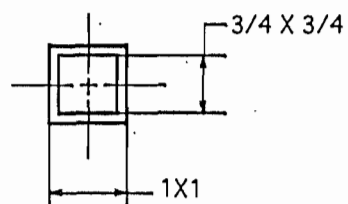


UNLESS OTHERWISE NOTED:		UNIVERSITY OF CALIFORNIA, DAVIS	
TOLERANCES		Dept. of Biological & Agricultural Engineering	
XX ± 0.01	ANGLES ±	TITLE Body Support	
XXX ± 0.005	SUR FIN		
DESIGN JJM	DATE 9/27/93	QTY 1	DRAWING NO.
DRAWN YP, Z	8/16/94	SCALE 1:5	P001.01
APPRV		SIZE A	SHT
RELEASE			PROJECT RTA-32804 #8

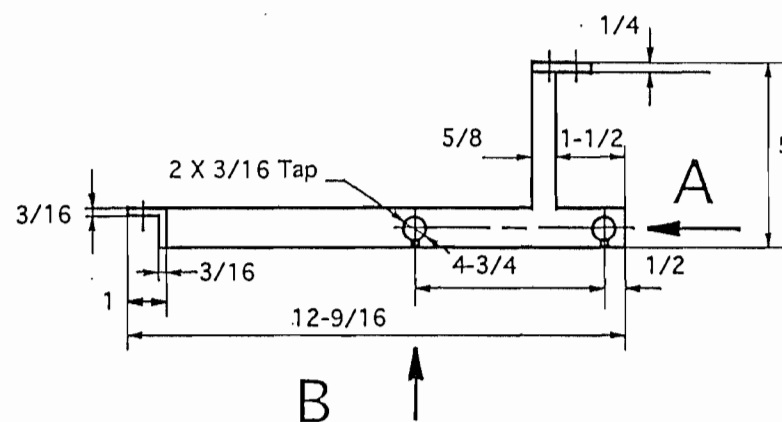
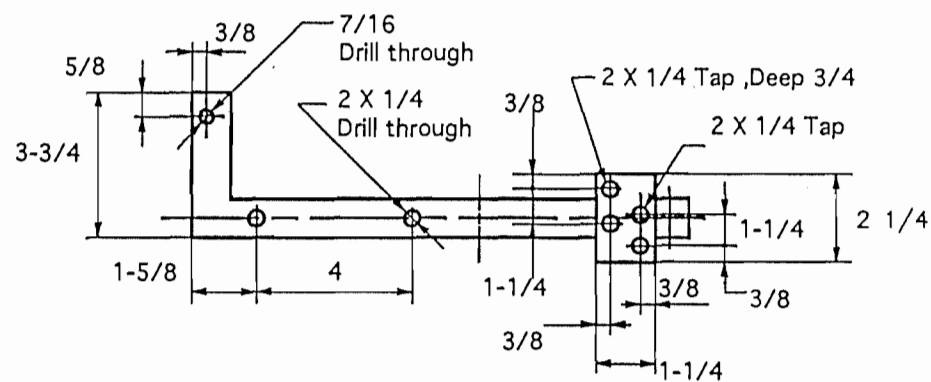
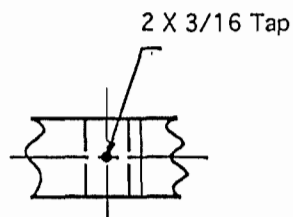


UNLESS OTHERWISE NOTED:		UNIVERSITY OF CALIFORNIA, DAVIS	
TOLERANCES		Dept. of Biological & Agricultural Engineering	
XX ± 0.01	ANGLES ±	TITLE	
XXX ± 0.005	SUR FIN	Support Frame	
DESIGN JJM	DATE 9/27/93	QTY 1	DRAWING NO.
DRAWN YP, Z	8/16/94	SCALE 1:5	P001.02
APPROV		SIZE A	SHT
RELEASE		PROJECT	RTA-32804 #8

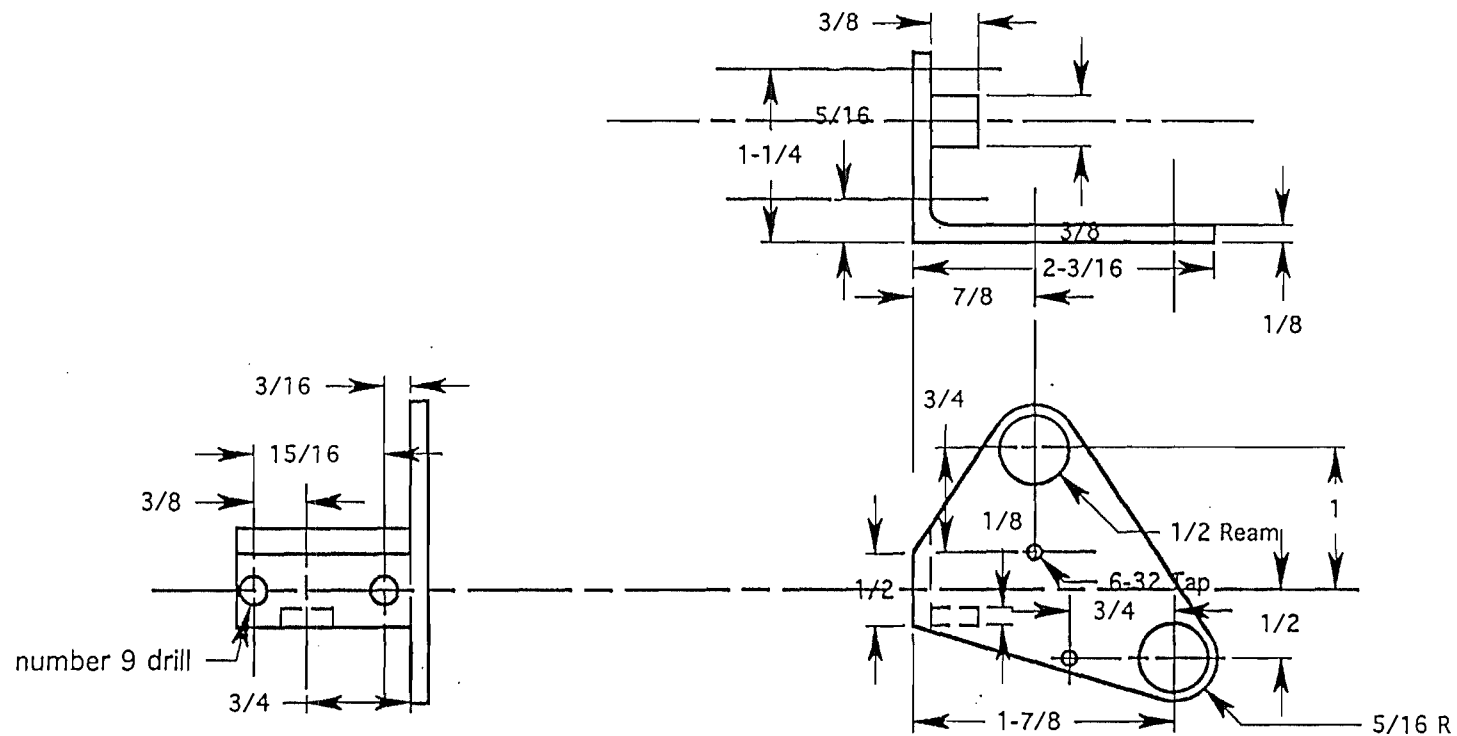
A view



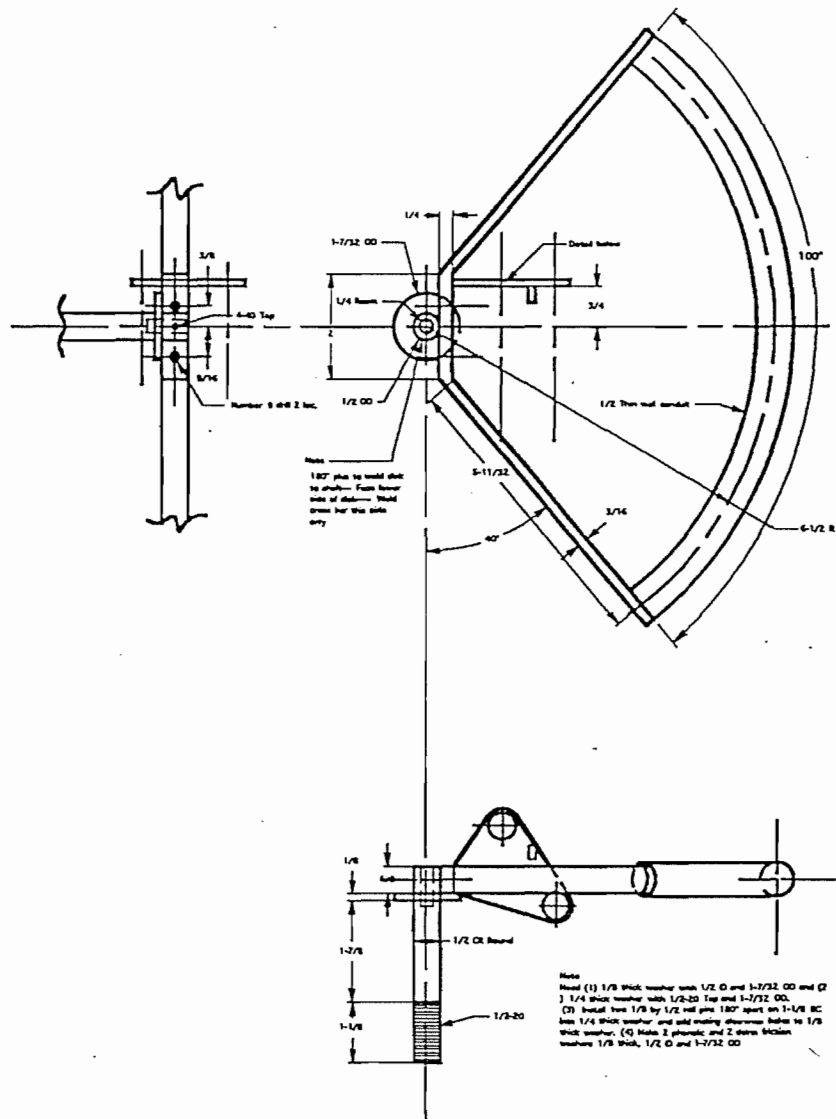
B view



UNLESS OTHERWISE NOTED:		TOLERANCES		UNIVERSITY OF CALIFORNIA, DAVIS	
XX ± 0.01		ANGLES ±		Dept of Biological & Agricultural Engineering	
XXX ± 0.005		SUR FIN		TITLE	
DESIGN JJM	DATE 9/27/93	Frame Base		QTY 1	
DRAWN YP, Z	8/16/94	DRAWING NO.		P001.03	
APPRV		SCALE 1:5		SIZE A	
RELEASE		SHT		PROJECT RTA-32804 #8	



UNLESS OTHERWISE NOTED: TOLERANCES		UNIVERSITY OF CALIFORNIA, DAVIS	
.XX ± 0.01	ANGLES ±	Dept. of Biological & Agricultural Engineering	
.XXX ± 0.005	SUR FIN	TITLE Right Flange	
DESIGN JJM	DATE 9/27/93	QTY 1	DRAWING NO. P001.04
DRAWN JJM		SCALE 1:1.4	
APPRV		SIZE A	SHT
RELEASE			PROJECT RTA-32804 #8

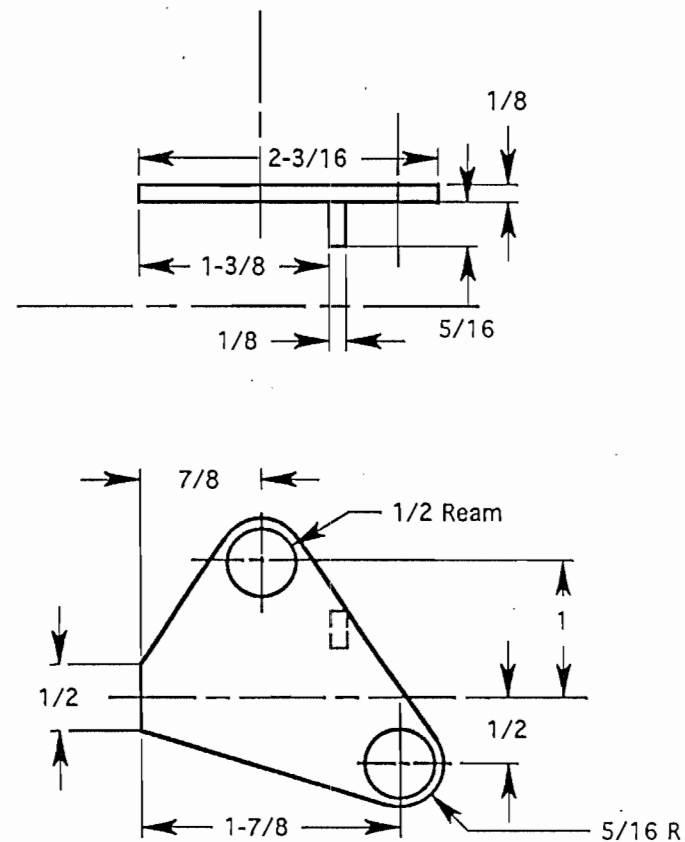
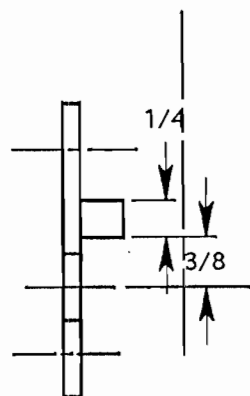


UNLESS OTHERWISE NOTED:	
TOLERANCES	
XX ± 0.01	ANGLES ±
XXX ± 0.005	SUR FIN
DESIGN JJM	DATE 9/27/93
DRAWN JJM	
APPRV	
RELEASE	

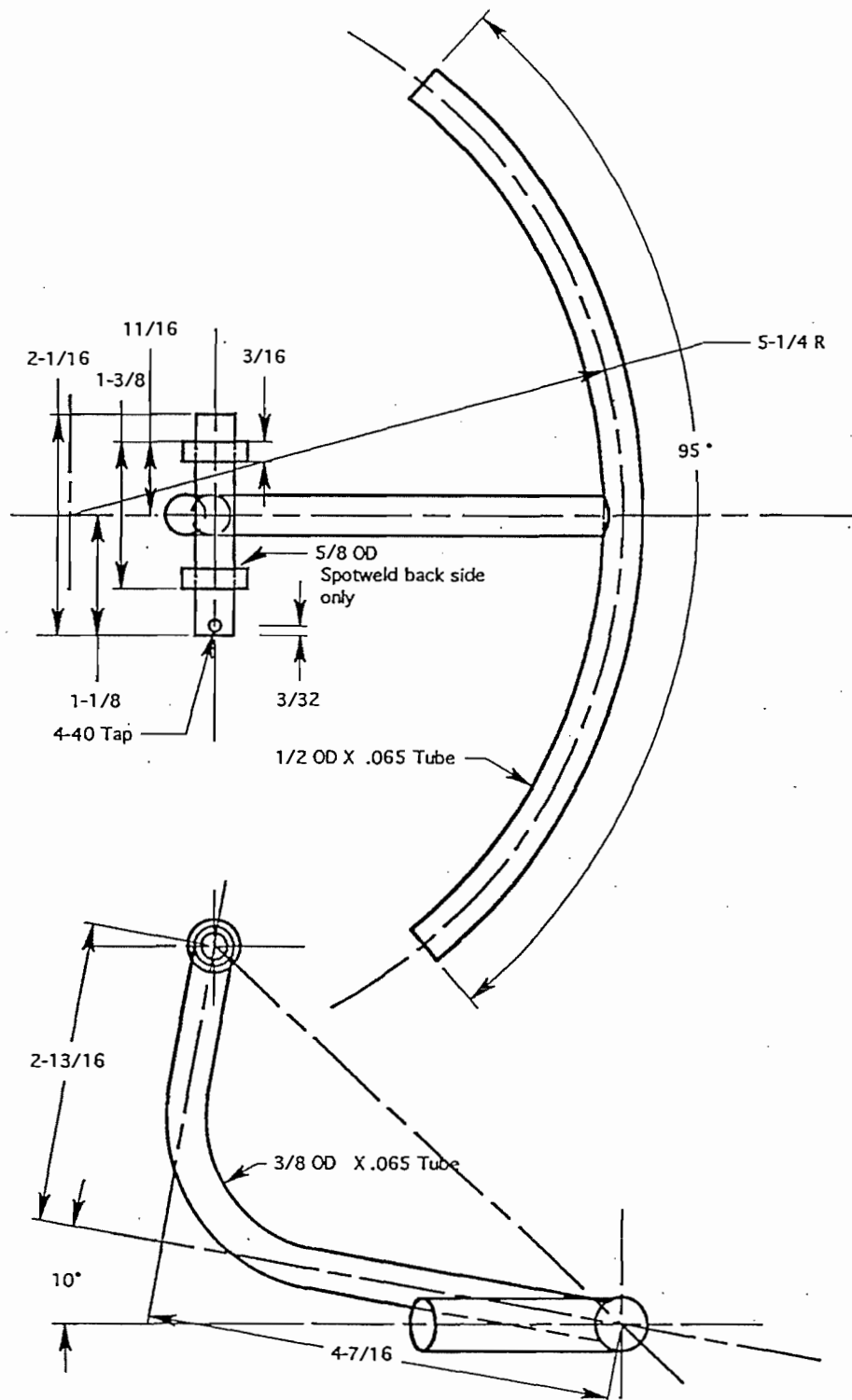
UNIVERSITY OF CALIFORNIA, DAVIS
Dept. of Biological & Agricultural Engineering

TITLE
Steering Bar

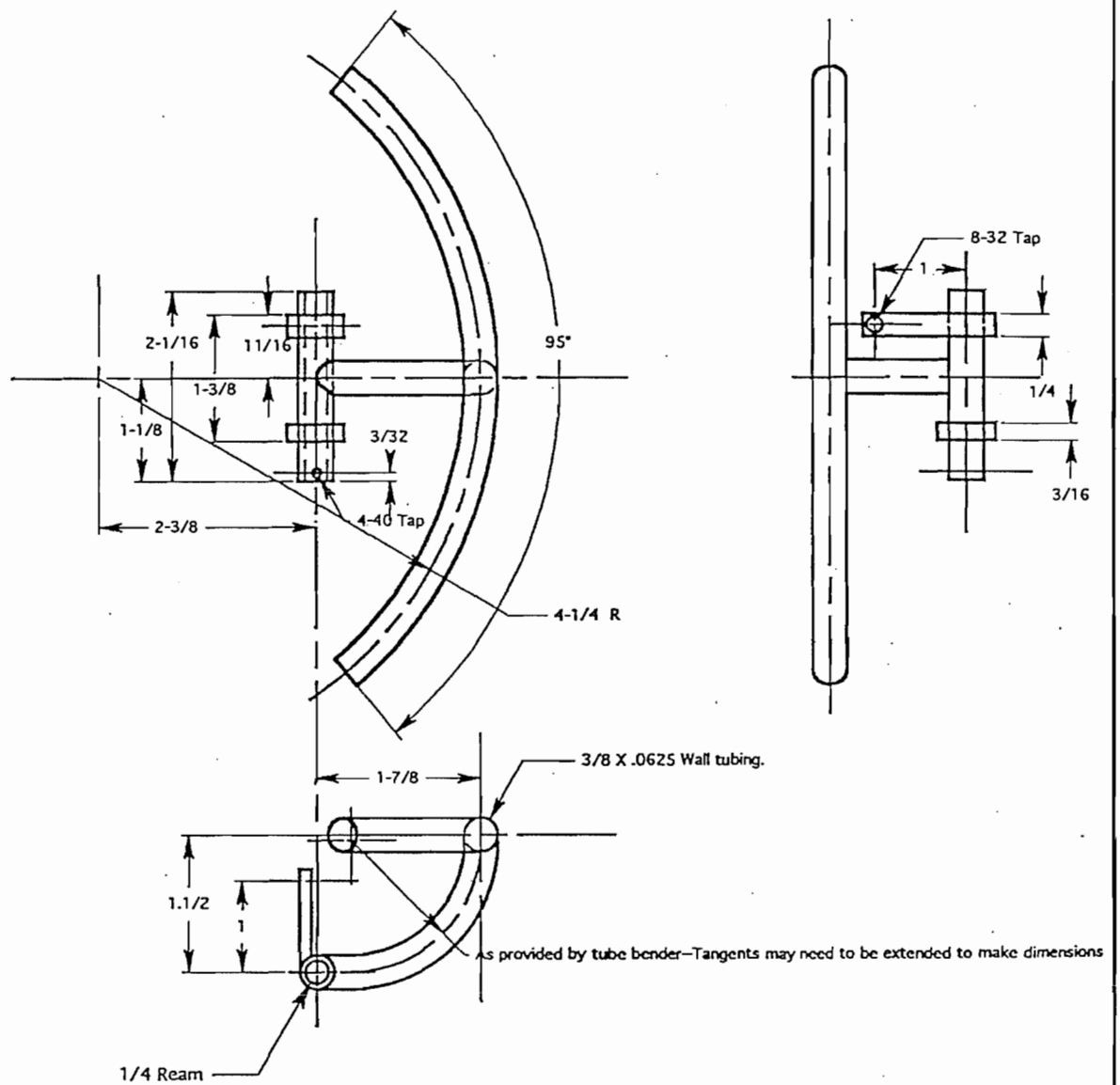
QTY 1	DRAWING NO. P001.05
SCALE 1:3.6	
SIZE A	SHT
PROJECT	RTA-32804 #8



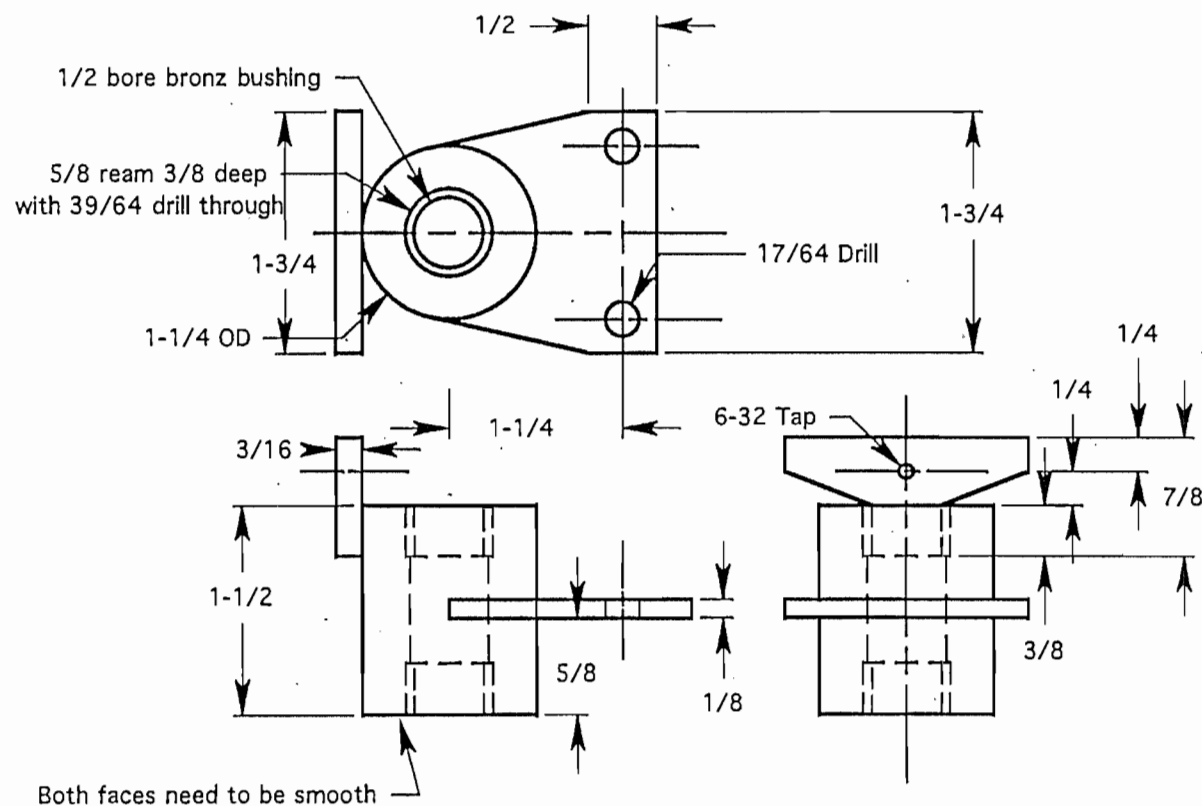
UNLESS OTHERWISE NOTED: TOLERANCES		UNIVERSITY OF CALIFORNIA, DAVIS	
.XX ± 0.01		Dept. of Biological & Agricultural Engineering	
.XXX ± 0.005		TITLE Left Flange	
DESIGN JJM	DATE 9/27/93	QTY 1	DRAWING NO. P001.05.01
DRAWN JJM		SCALE 1:1.4	
APPRV		SIZE A	SHT
RELEASE			PROJECT RTA-32804 #8



UNLESS OTHERWISE NOTED:		UNIVERSITY OF CALIFORNIA, DAVIS	
TOLERANCES		Dept. of Biological & Agricultural Engineering	
XX ± 0.01	ANGLES ±	TITLE Brake Bar	
XXX ± 0.005	SUR FIN	QTY 1	DRAWING NO. P001.06
DESIGN JJM	DATE 9/27/93	SCALE 1:1.7	
DRAWN JJM		SIZE A	SHT
APPRV		PROJECT	RTA-32804 #8
RELEASE			



UNLESS OTHERWISE NOTED: TOLERANCES		UNIVERSITY OF CALIFORNIA, DAVIS	
XX ± 0.01	ANGLES ±	Dept. of Biological & Agricultural Engineering	
XXX ± 0.005	SUR. FIN.	TITLE Throttle Bar	
DESIGN JJM	DATE 9/27/93	QTY 1	DRAWING NO. P001.07
DRAWN JJM		SCALE 1:2	
APPRV		SIZE A	SHT
RELEASE		PROJECT	RTA-32804 #8



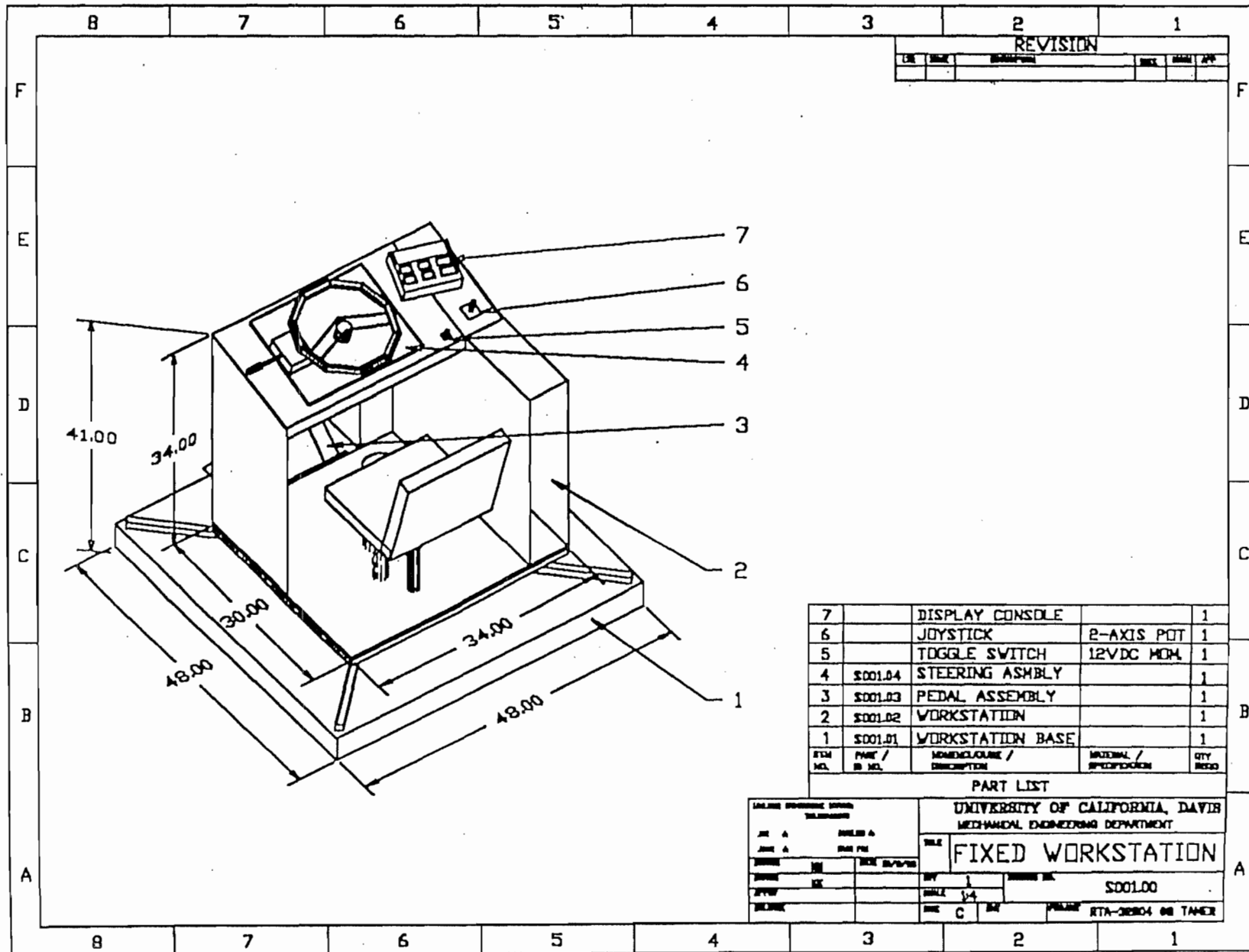
UNLESS OTHERWISE NOTED: TOLERANCES		UNIVERSITY OF CALIFORNIA, DAVIS	
.XX ± 0.01	ANGLES ±	Dept. of Biological & Agricultural Engineering	
.XXX ± 0.005	SUR FIN	TITLE Steering Journal	
DESIGN JJM	DATE 9/27/93	QTY 1	DRAWING NO. P001.08
DRAWN JJM		SCALE 1:1.4	
APPRV		SIZE A	SHT
RELEASE			PROJECT RTA-32804 #8

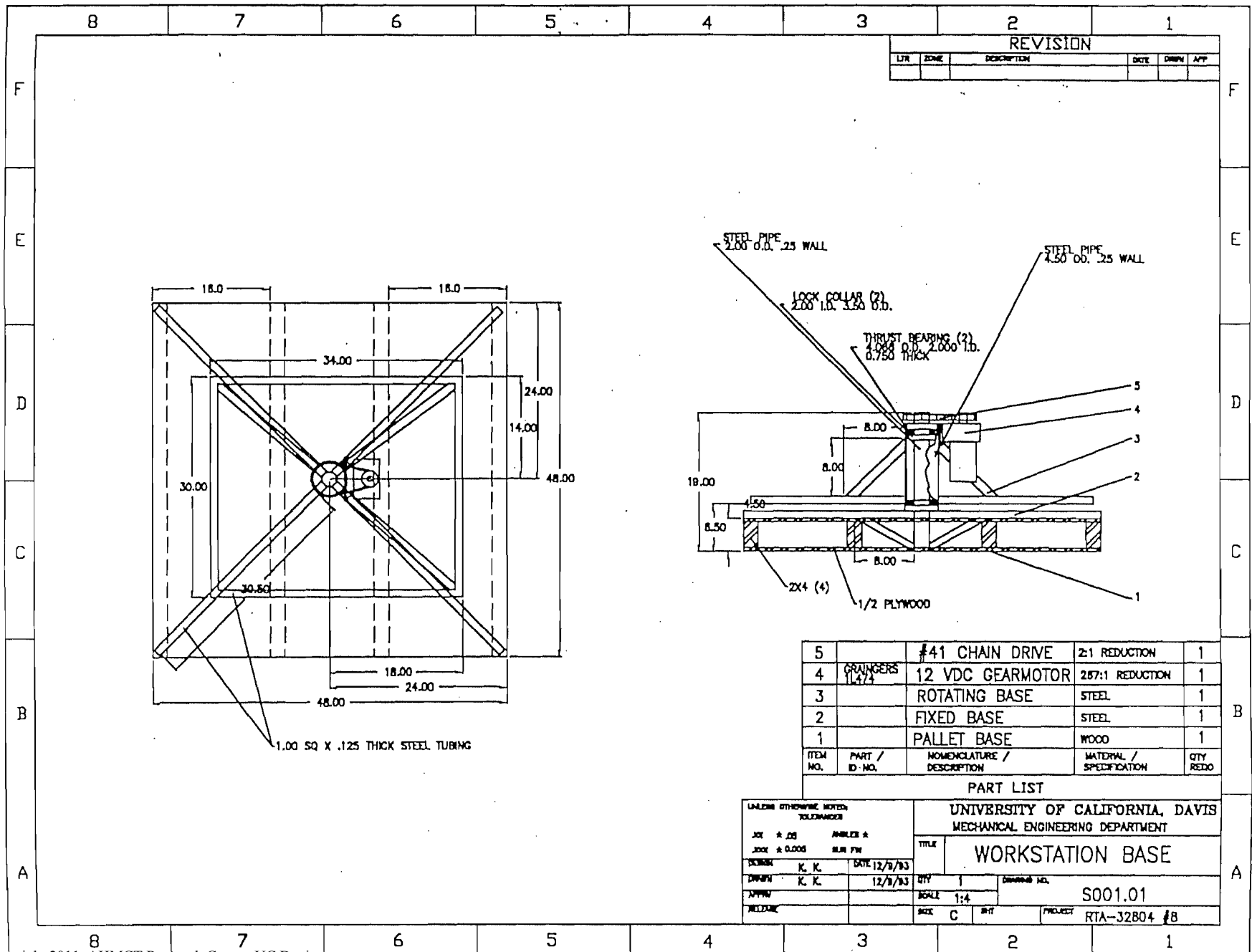
APPENDIX C2

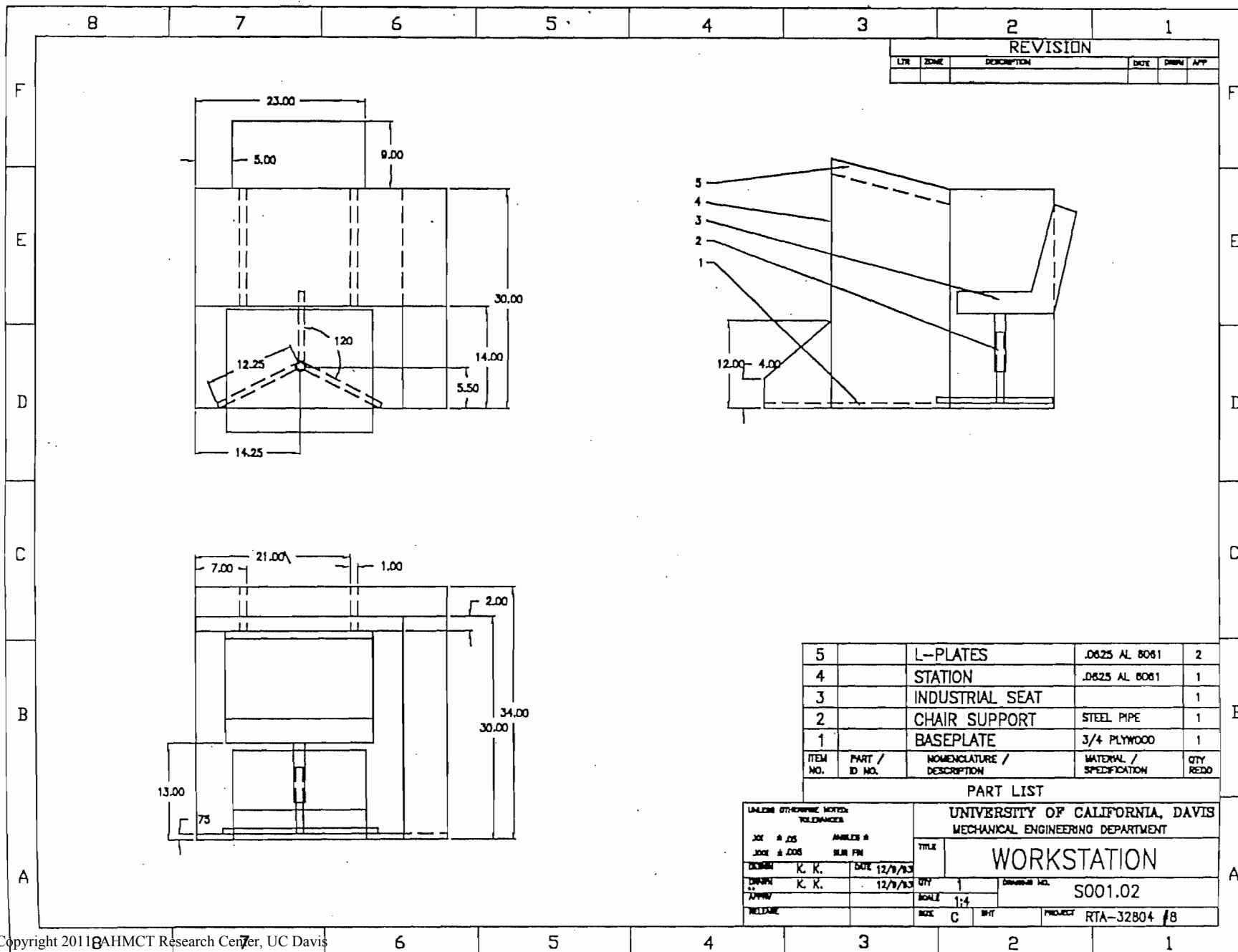
Mechanical Drawings for Stationary Unit

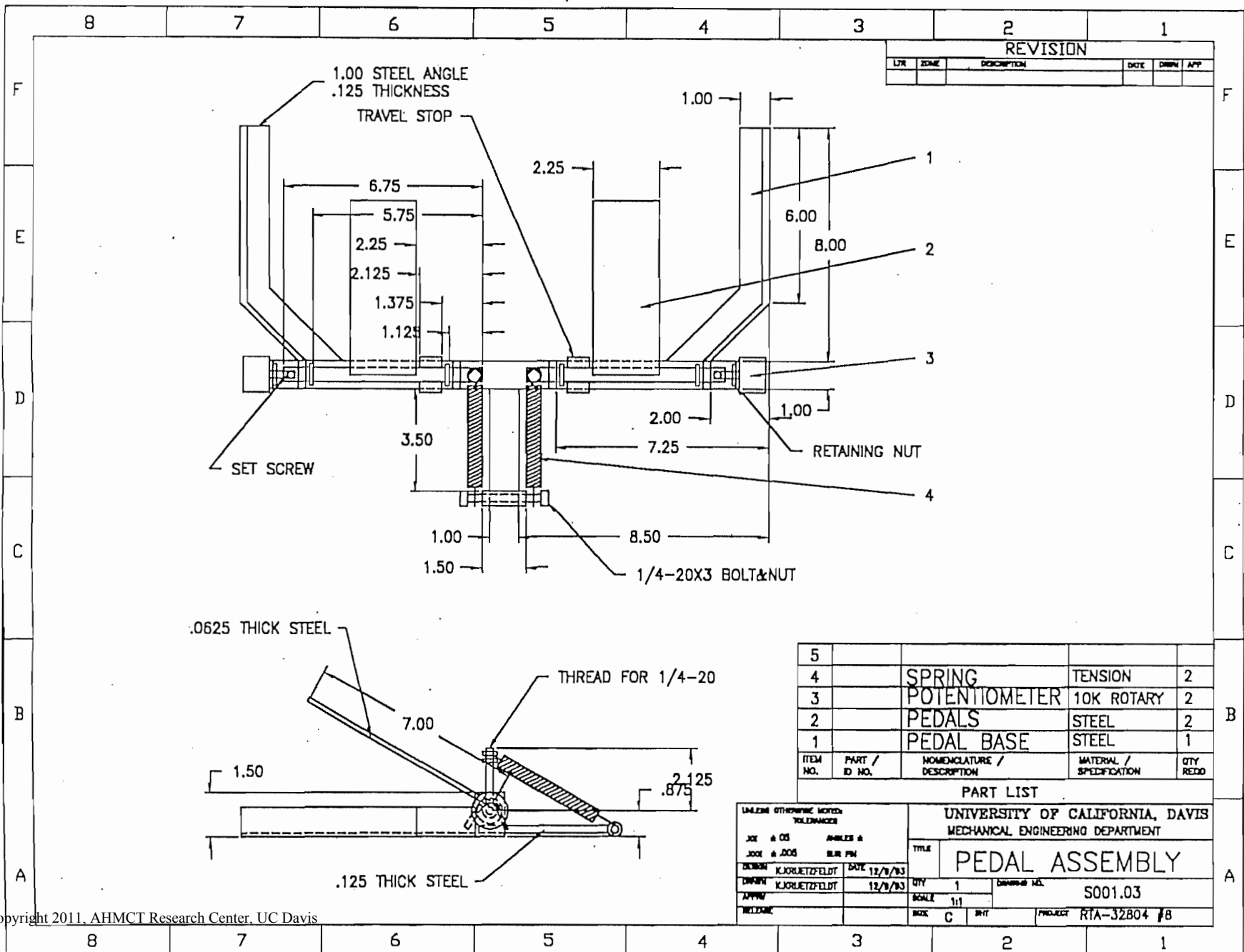
NOTE:

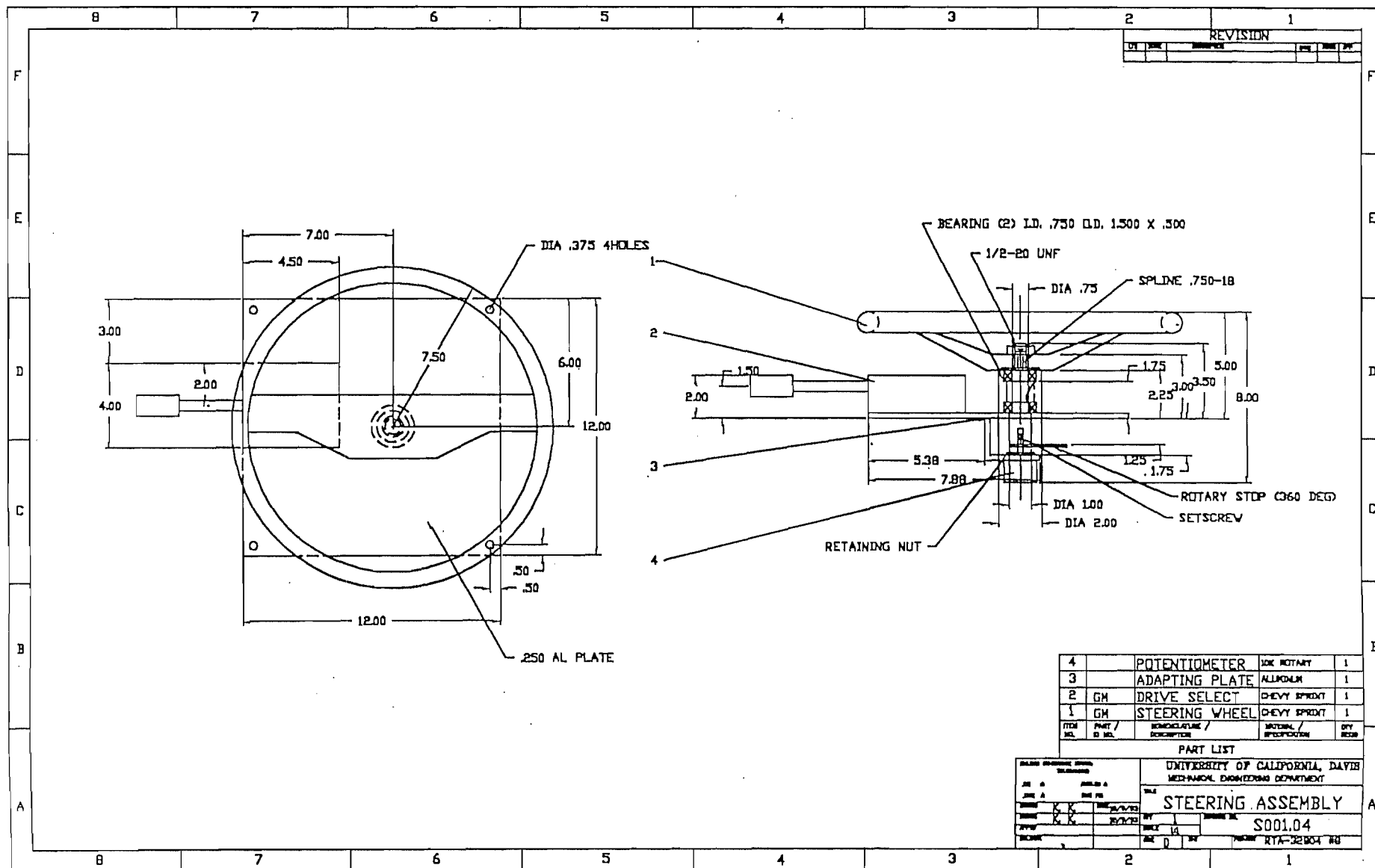
All units in the drawings included in this appendix are in inches. To convert to millimeters, multiply 25.400.











APPENDIX D

Operating Instructions

I. STARTUP

1. Set parking brake.
2. Turn ignition key to on position, wait for case 621 self diagnostic system to perform check.
3. When control panel lights indicate "O.K.", Start endloader.
4. If a warning light persists, consult case operating manual before continuing.
5. Make sure endloader is clear of obstructions and people before continuing. Set bucket on ground.

II. COMMUNICATIONS SET UP

1. Make sure remote operating device is set-up before continuing. If not, refer to remote device set-up section.
2. Set transmission to neutral position.
3. Turn on loader control computer (located behind operators seat) by switching toggle switch to "on" position. Green light will flash once enabled.
4. Pull remote enable switch on (located to right of case diagnostic panel). LED at switch base will light once enabled.
5. Release parking brake and exit endloader.

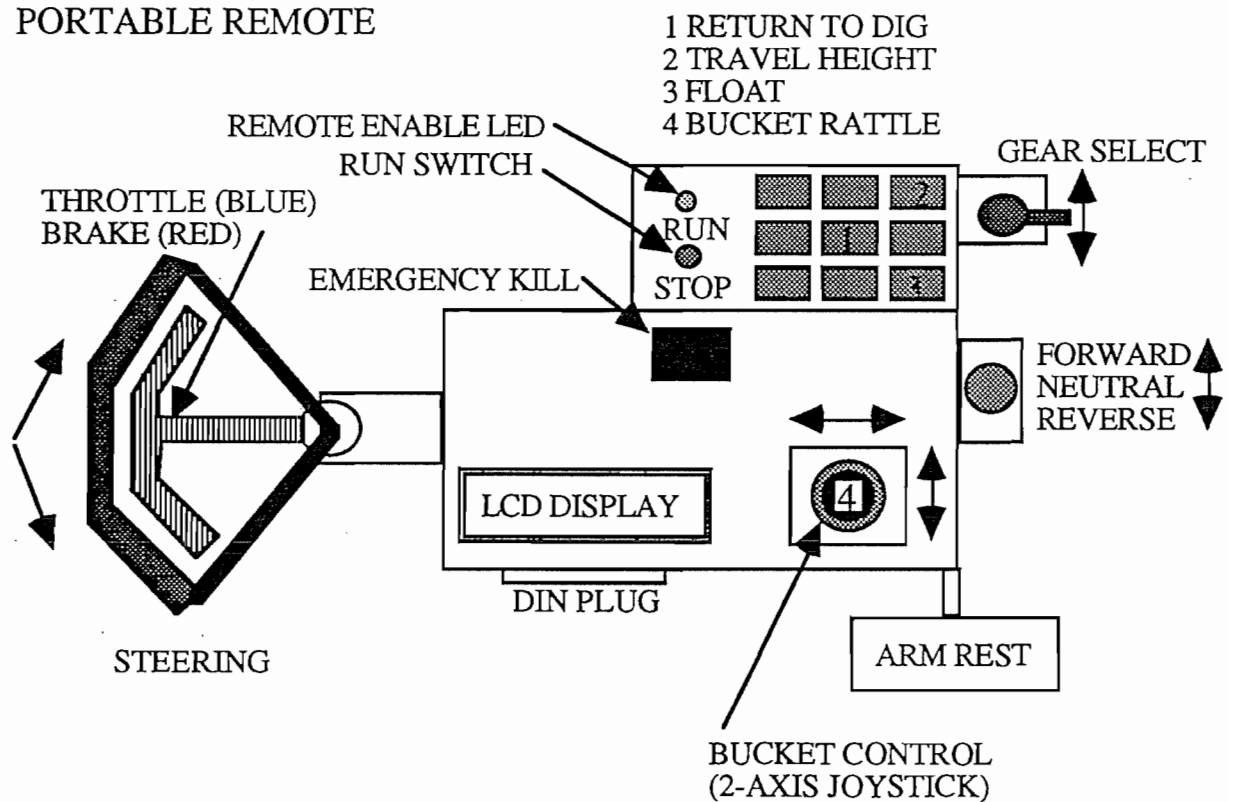
III. REMOTE DEVICE SET-UP

1. Connect the rechargeable battery to the single board computer inside the backpack (to the socket labeled "power"). Turn power on.
2. Remove blue wire bundle from backpack and connect DIN plug to input of remote operating station to be used.

IV. OPERATING THE PORTABLE UNIT

1. Connect blue cable to DIN plug at base of control.
2. Set gear select in neutral.
3. Pull brake lever and hold. Toggle run switch up momentarily and release. Wait for the remote enable LED to light.
3. Release brake lever and begin operations.

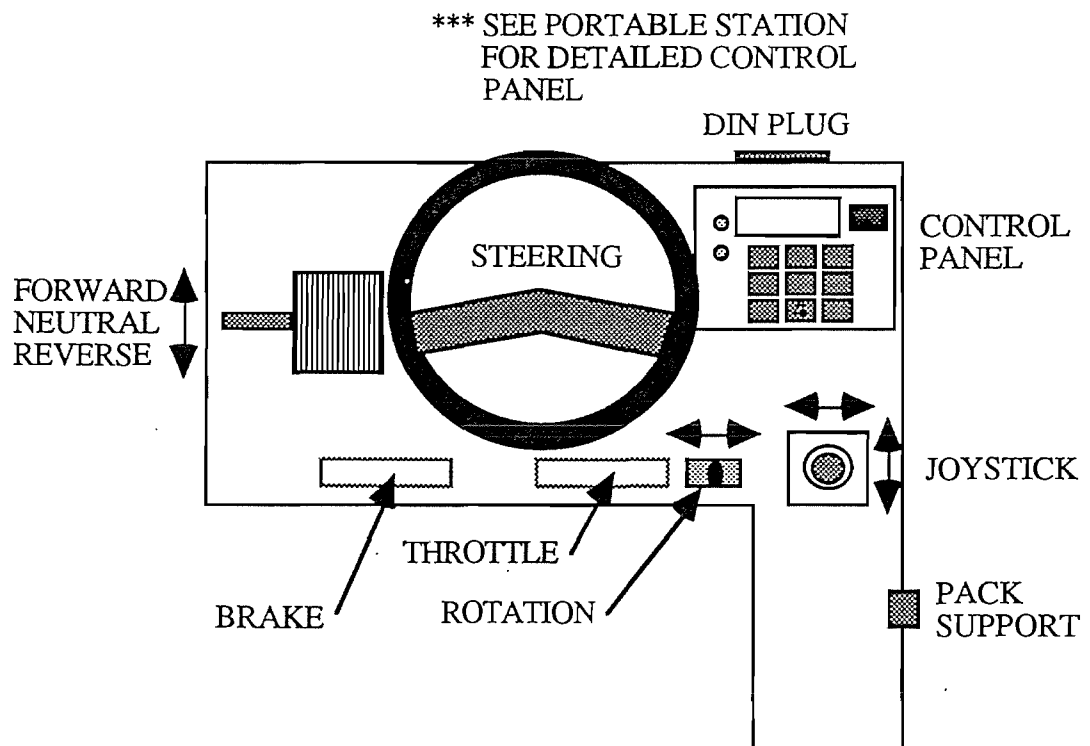
PORTABLE REMOTE



IV. OPERATION ON THE PORTABLE UNIT

- Place backpack on pack support hook on station. Connect blue cable to DIN plug.
- Set gear select in neutral.
- Press brake pedal and hold. Toggle run switch to up position and release. Wait the LED to light.
- Release brake and begin operations. Use rotation toggle switch if necessary to maintain line of sight.

STATIONARY WORKSTATION



V. SHUT DOWN

- Park loader in safe location.
- Toggle run switch to down position. Turn off computer In backpack.
- Board loader. Set parking brake.

- Push in remote switch. Toggle vehicle control computer switch to down position.
- Resume normal loader operations.

VI. TROUBLESHOOTING

In case the remote enable LED fails to light, check the following:

1. ON LOADER

- Parking brake engaged -- release it.
- Gear select lever not in neutral position -- set to neutral.
- Loader Control Computer light not blinking- reset switch.
- Antenna not properly connected on loader cab roof.

2. REMOTE UNITS

- Backpack battery dead or not properly connected.
- DIN plug not securely fastened to cable.
- Gear select lever not in neutral position.
- Brake not engaged while toggling run switch.
- Antenna on backpack not fastened securely.

APPENDIX F

Summary of Survey On Remote Operation By Caltrans Workers

Name	District	Years of Operating loader	Experience on Remote Operating	Time to get used to the remote unit		Comments	Date
				Portable	Stationary		
Mike Salvador	3	6	4 yr..	10 min.	10 min.		6/29-30, 8/20, 8/27/94
Danny Gacia	11	22	0	10 min.	10 min.	Just to say that this idea just might save life	8/10, 12/94
Shawn Hossey	4	25	0	5	5	Prefer joy stick for steering	8/20/94
							8/20/94
Dong Bunqarz	02	5	0	n/a	5	Surprisingly easy to run!	8/20/94
Collinge		0	0		5	Easy to get used to	8/20/94
Diana Elmoer	3	1	0	5	5	Would be good for dangerous work, or where the operator needs to be in an offset position. Great!	8/27/94
Jeff Kisser	04	3	0	5	n/a	Enjoyed running the machine. It's fairly user friendly.	8/27/94
Margarita L. Martinez	03	1	0	n/a	5	A great project.	8/27/94
Robert Pence	03	8	0	n/a	4	Would be good on big rock slides and in bad spots	8/27/94
Duke Picarso	03			n/a	3-5	Discrete steering, too sensitive	8/27/94
Lauren Rampelburg	10	35	/	n/a	<5	It is very neat	8/27/94
Edward Sugg	03	30	/	n/a	5	Good for hazard place	8/27/94
Mark Vukich	02	15	/	10	n/a	Very nice, only took a few minutes to get used to it. Would look nice in a parade	8/27/94