Spring 1983

Microcomputer control of a model robot system using programmable pulse width modulation

Richard Eugene Wainwright

MICROCOMPUTER CONTROL OF A MODEL ROBOT SYSTEM USING PROGRAMMABLE PULSE WIDTH MODULATION

BY

RICHARD EUGENE WAINWRIGHT, 1957-

A THESIS

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF MISSOURI-ROLLA

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

1983

Approved by

Randy H. Moss (Advisor) J. J. Bourquin

C. Y. Ho

ABSTRACT

The hardware design of a model robot system for experimentation in microcomputer control is presented. A special interface board is designed to allow an Apple II Plus microcomputer to control a model robot system which consists of a three-jointed robot with an electromagnet end effector and a conveyor belt, with an optional vision system. A programmable pulse-width modulator is designed to drive the DC motors which operate the robot and conveyor belt. Improvements and additional uses for the model system are discussed.

ACKNOWLEDEMENTS

There are many individuals whom I wish to thank for their contributions and support throughout the course of this project.

I wish to thank Dr. Darrow Dawson and Dr. Jack Bourquin for their time, advice, and encouragement.

I also wish to thank Dr. C. Y. Ho for his instruction and support.

I wish to express my sincere thanks to Dr. Randy Moss for his contribution of time and knowledge, and his patience and invaluable guidance through all the trials and tribulations of this project.

I wish to acknowledge Eric Stelzer and K. F. Shen for their software contributions, Reverand Don Farnsworth for his work on the drawings, and Kathy Matlock for her assistance with the typing of this thesis.

Lastly, I dedicate this thesis to my beautiful wife, Joycelyn, for her love, patience, and sustaining support and to my Lord for his strength and guidance.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

PAGE

LIST OF TABLES

I. INTRODUCTION

Industrial analysts claim that robotics will be one of the main industries of the near future. Industries of the 1980's are already making plans to incorporate even greater numbers of robots in the work place. To maintain and improve this automation, technicians and engineers will have to be trained in the field of Robotics. Acquiring the necessary skills will require both theoretical and practical experience. Due to both the initial high cost of a robot and the high possibility of damage to the robot from carelessness or error during training, a more practical experimental system is required. Such a system should allow the user enough flexibility to test a wide range of software control methods, demonstrate the basic techniques of robot motion, and demonstrate the considerations in hardware/software trade-offs. One possible hardware/software trade-off is the use of a hardware controller to free the computer(s) from unnecessary actuator control, thus allowing more efficient use of computer time in tasks such as complex trajectory generations. The system should also simulate a realistic environment to enable the user to visualize such abstract control concepts as straight line motion, collision avoidance, and interaction of a robot with its surroundings. One such experimental system is presented in this thesis.

A description of the model robot system is presented to

demonstrate its flexibility and versatility. This is followed by the design of an interface board which allows an Apple II Plus microcomputer to control the robot system. Finally the design of a hardware controller, which consists of a manual controller and programmable pulse-width modulators to drive the system's DC motor actuators, is presented. Future experimental uses for the system and possible improvements are discussed.

II. TASK DESCRIPTION

The first requirement in producing a practical experimental system is to define the meaning of a robot system. Presently manufacturers are designing manufacturing units, often referred to as "work cells". Each work cell is designed to be a self-contained system which accepts either raw materials or sub-products and produces sub-products or finished products. If fixed automation is used in this cell then the work cell is only valuable or practical for mass production. If flexible automation such as a robot is used then the work cell can be used for batch manufacturing. In batch manufacturing the cell can produce a limited quantity of a certain item then be easily converted to produce new items or products.

Thus a robot system can be referred to as a work cell and defined as a system consisting of a robot or robots and all necesarry peripherals, such as tools and sensors, to carry out a specific task or function.

Therefore a work cell or system needs to include components, such as a means to transport materials into and out of the cell, and all peripherals or devices associated with the process to be performed. External dependencies must be kept to a minimum for the work cell or robot system to be efficient. Figure 1 shows a block diagram of a possible work cell.

The basic component of any robot system is the robot.

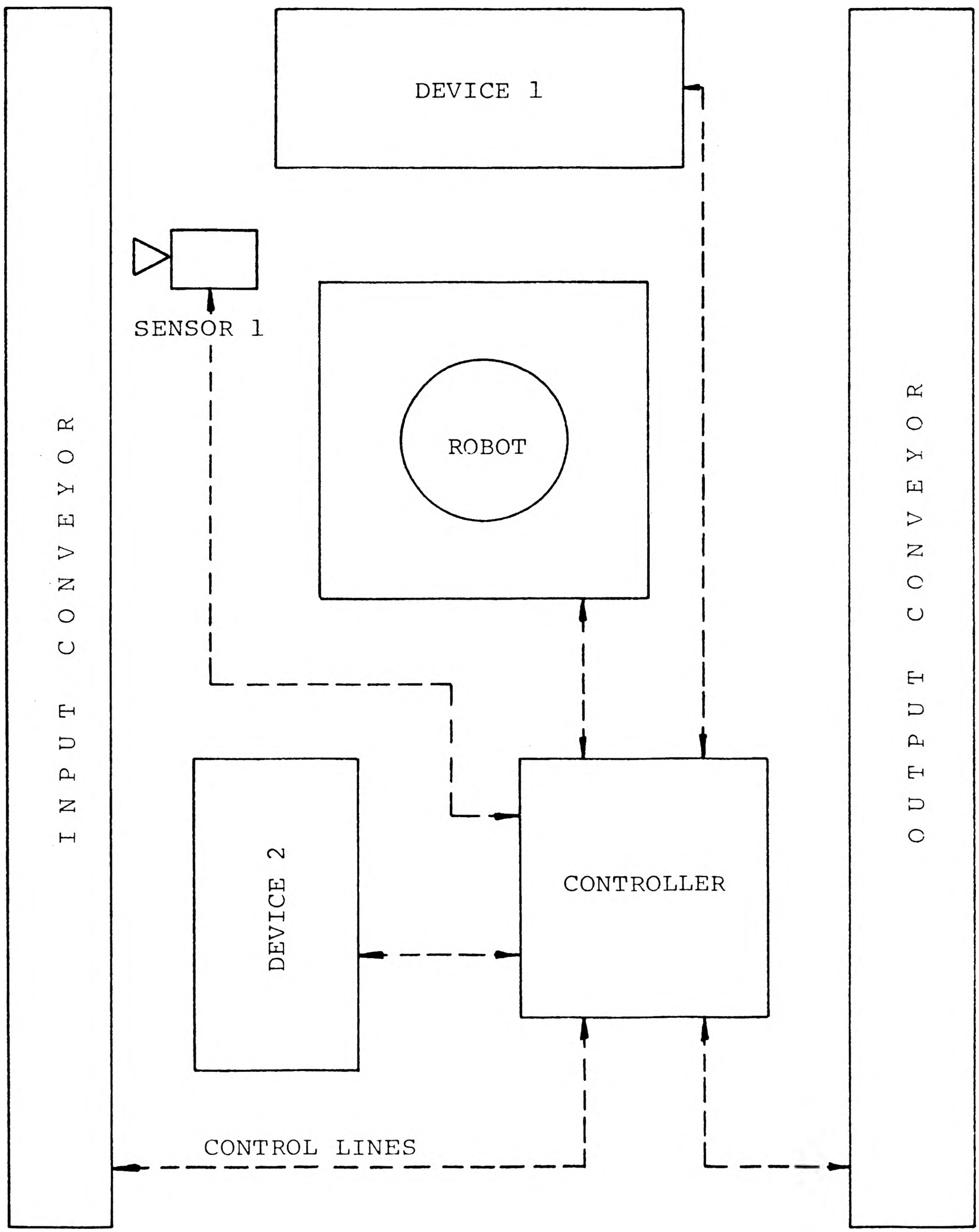


Figure 1. Work Cell

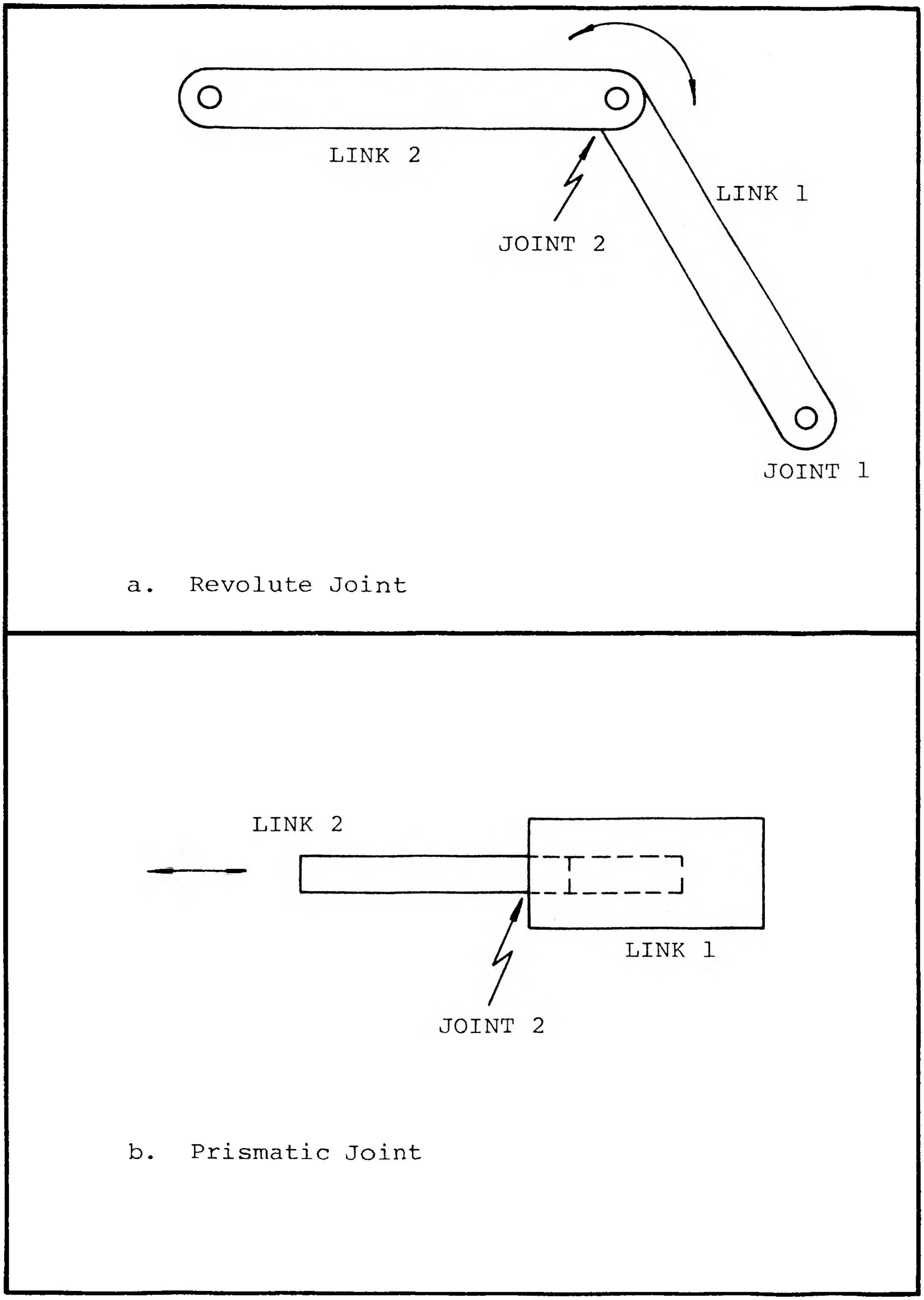
Although many individuals picture robots in the form of mechanical men, the prominent form of robots used in industry is that of an arm with some sort of gripper. Since one of the main objectives of the experimental system is to simulate a real industrial system, the robot used in this system is based on the arm configuation.

The robot arm can be broken down into several distinct parts:

1. Links connected by joints to represent structure and define movement.
2. End effector or gripper.
3. Actuators to control joint movement.
4. Sensors for closed loop feedback.

Each link with its corresponding joint gives the entire structure one degree of motion or freedom. Often, a basic robot uses six joints to allow six degrees of freedom. Three degrees are for major positioning of the end-effector while the remaining three are used for minor positioning and orientation (roll, pitch, and yaw). Basically there are two types of joints -- prismatic and revolute which produce linear and angular motions, respectively. Figure 2 shows the joint-link relationship for both revolute and prismatic joints.

If a basic robot used all revolute joints it could simulate, to a limited degree, the movement of a human using his or her waist, shoulder, elbow, and wrist. One such robot is manufactured by Unimation and is called the PUMA.



a. Revolute Joint

b. Prismatic Joint

Figure 2. Links and Joints

The end effector or gripper is the end point of the arm which is usually some sort of tool, such as a spray gun, or device for picking up other objects. The end effector is also the point which is controlled for position and motion.

Actuators for the robot must also be considered. The most widely used actuators or motors are either hydraulic, pneumatic, electric, or stepper motor servos. Each type of servos has its own advantages and disadvantages. For example, hydraulic servos have accurate control technology available and produce great torques, but yet the fluid lines are very cumbersome, awkward, and prone to leakage, and the compressors are often very noisy.

Sensors on the robot allow accurate control of position, velocity, acceleration, and torque of the arm. Too many sensors add great expense to the robot and complicate the processing of feedback information, possibly to the point of losing real-time control of the robot. Too few sensors on the robot cause accurate control of the robot to be lost, rendering it useless for precise or even practical use. Balance in the use of sensors is thus an important consideration for the robot.

Another component of the robot system is the environment in which the robot or robots work. Here the types of tasks or operations the work cell is to perform determine the environment. For instance if the cell is part of an assembly line, a conveyor may be part of the environment for material transportation. If the robot is to

machine a part on a numerically controlled lathe or mill, then that device would be considered part of the system.

In addition to the sensors on the robot, additional sensors may be needed to monitor the work cell. One such additional sensor is vision which can be used to identify and/or locate objects in the work cell. Another possible sensor would be detectors along a conveyor to monitor product flow. Also if the robot is used in conjunction with other machines, those may need sensors to provide their status.

The final component of the system is the controller. The controller is the device or processor which takes desired characteristics or specifications as its inputs and outputs them as physical or realizable results. For large systems several controllers may be used to orchestrate an entire process with a master controller to coordinate them. Such a control configuration is known as a hierarchy.

The inputs for the controller of a robot system consist of parameters such as actual position of joints and links, their velocities, accelerations, and torques. Desired conditions or operations of other devices or peripherals in the system may also be included as inputs. The outputs of the controller are commands to the actuators of the robot and system, to the sensors, and possibly even to other controllers.

For robotic systems there are basically two types of controllers -- hardware and software. If the controller is

totally implemented in hardware it is sometimes referred to as an analog controller. Such a controller could process its inputs rather quickly but could also entail great costs to build and maintain. Also a total hardware controller would hinder any system development or modifications due to the difficulty and cost of redesigning circuits and reconstructing the controller.

Although a controller could never be totally software, a combination of hardware and software in proper balance can greatly reduce costs and provide flexibilty for change and development of the system. The hardware should be designed to limit the control of the system by software as little as possible. Yet, the hardware should also free the computer from as much time-consuming input/output (I/O) processing as possible, such as actuator control and possibly even some of the signal processing of the data from the sensors. Thus information to and from the computer will be in minimal processing form to decrease computation time and allow more computation time for other concerns such as trajectory generation, collision avoidance, and system error testing. Software gives the system great flexibilty by allowing a system to be easily modified by simply changing the program. On the other hand, if the system grows and evolves too much, some flexibilty and cost savings may have to be sacrificed for the speed of hardware to reduce computation time to levels which allow real-time responses in the system. Therefore, if feasible, space should be designed into the

hardware controller for future expansion of the system.

The design of an experimental robot system requires the consideration of all these details if it is to be a realistic simulation of true-life systems which can provide a practical medium for experimentation and training.

III. MODEL ROBOT SYSTEM

The model robot system fulfills the requirements of a practical experimental system, which are basically to:

1. Provide an inexpensive and easily modifiable physical robot system.
2. Provide a medium for hardware/software control development.
3. Provide training through "hands on" experience.
4. Provide the realization of a system with many of the components used in true-life systems.

One of the initial considerations for the model robot system is the type of materials out of which the physical system is to be constructed. Sophisticated nylon building blocks, structural elements, DC motors, gear trains, and linkages by Fishertechnik (1) are used for the physical system. These system components have enough precision and durability for reliable and repeatable structural movements. The versatility and reusability of these building components allow easy and inexpensive modifications to be made to the system for illustrating various design concepts and configurations. The scope of real-life elements modelled by Fishertechniks provides a wide range of possible system structures with minimal need for development of compatible components. The modest cost of these building elements adds

to their feasibility compared to the high cost of their precision machined metal counterparts.

Similar work using these building materials, for physical simulations, is being done by Shimon Y. Nof, Michael P. Deisenroth, and Wilbur L. Meier (2,3) at Purdue University.

The second concern is the actual characteristics or devices to be contained in the model robot system. This is defined by the intended function of the system. The intended purpose of this model robot system is to sort objects moving down a conveyor belt by identifying the object, picking it off the belt, and placing it in a specified location, such as a bin. To fulfill this function the model robot system consists of a robot, a conveyor belt, and an optional vision system. Figure 3 shows the block diagram of the complete robot system including both the hardware controller and the Apple computer which operate the model robot system.

A. MODEL ROBOT

As in the real world, the most important component of this system is the model robot. The model robot in the system is shown in Figure 4.

Due to the limits on structural integrity, from using the previously described materials, the robot implements only the three major positioning joints. These joints are all revolute joints simulating waist, shoulder, and elbow

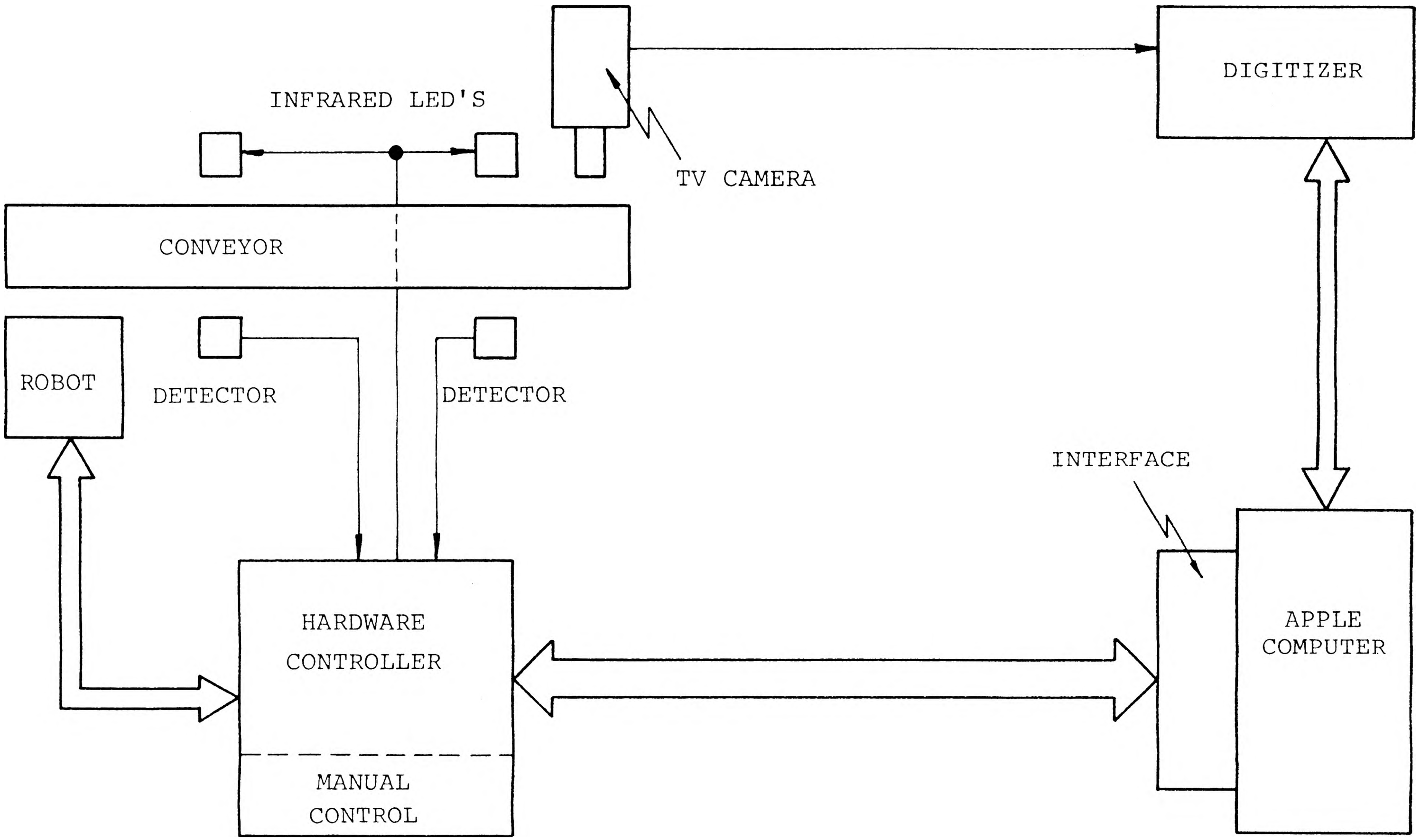


Figure 3. Complete Robot System Block Diagram

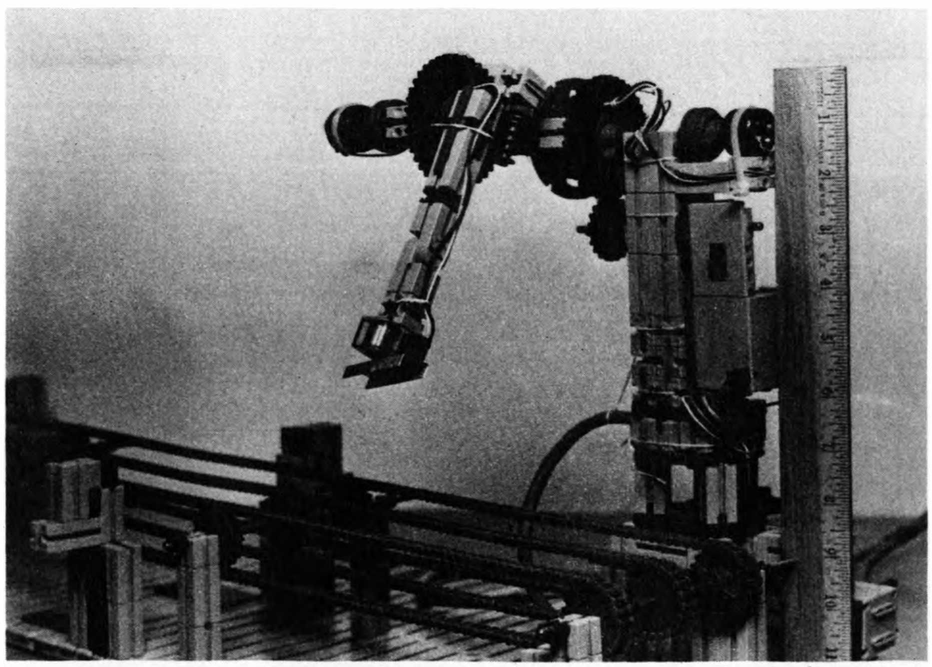


Figure 4. Model Robot Carrying a Load

movements. The actuators for controlling these joints are six-volt DC motors by Fishertechnik. The waist and shoulder joints are driven by the regular size motor (Art.-No. 31039) through a gear box and drive gear pair with a total gear ratio of 480:1. The elbow joint is driven by the mini-motor (Art.-No. 31062) through a worm gear drive with a 560:1 gear ratio.

The end-effector for the model robot is an electromagnet for picking up ferrous objects. The magnet "on" represents the pick or grasp mode while the magnet "off" represents the drop or release mode.

Positional feedback on the robot is achieved through potentiometers mounted on the rotational shaft of each joint. Figure 5 (from (4)) shows how the angular position of the shaft is converted into a voltage by the potentiometer and later converted to a digital word for the computer by an analog to digital converter (A/D).

The overall characteristics of the model robot are as follows:

1. The robot stands 26.5 centimeters from the base to the shoulder axis.
2. The upperarm is 9.5 centimeters from the shoulder axis to the elbow axis.
3. The electromagnet or end-effector is 12 centimeters from the elbow axis.
4. The origin or reference point on the robot, labled "O" in Figure 6 (4), is at the

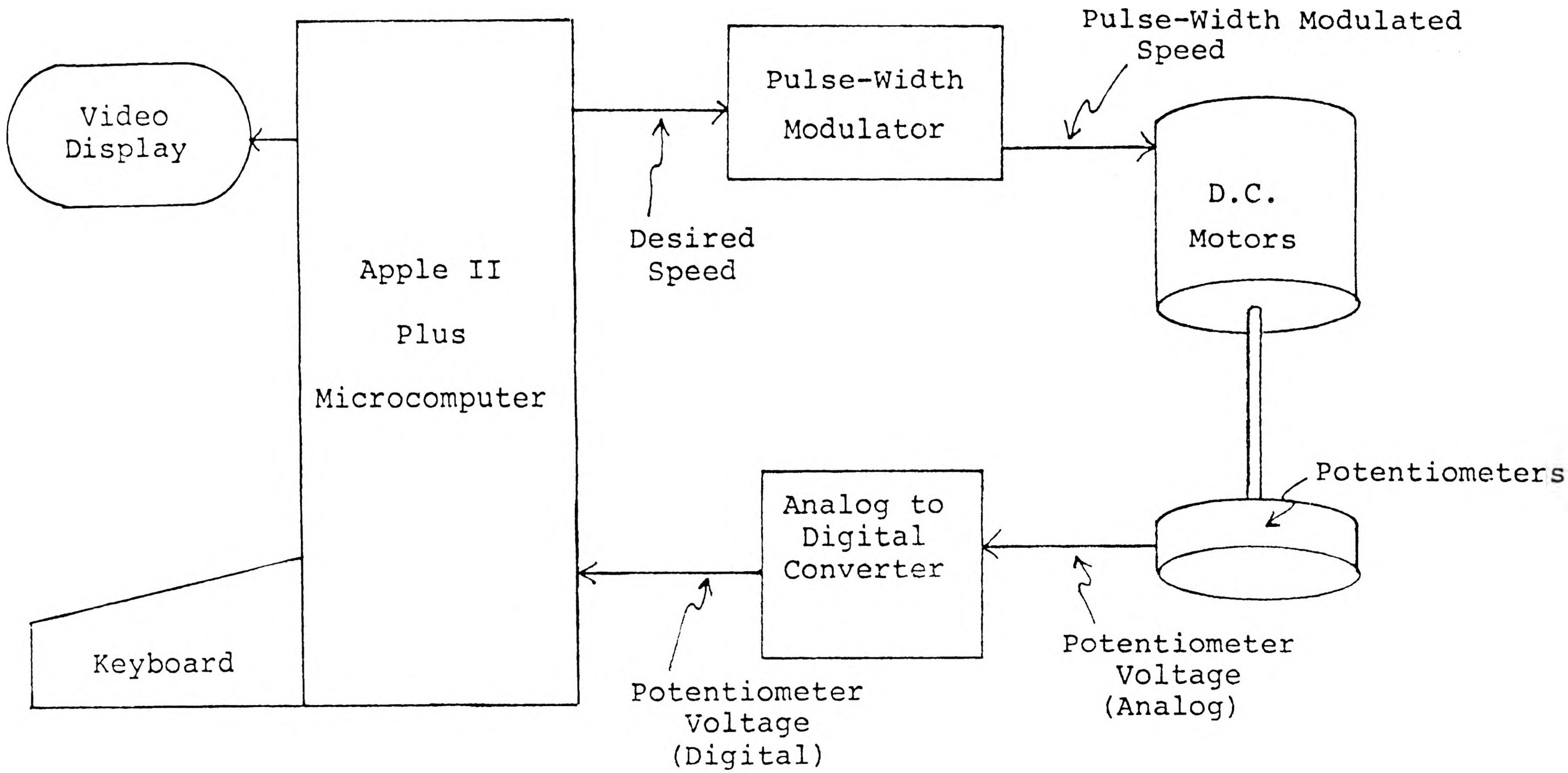


Figure 5. Hardware Block Diagram

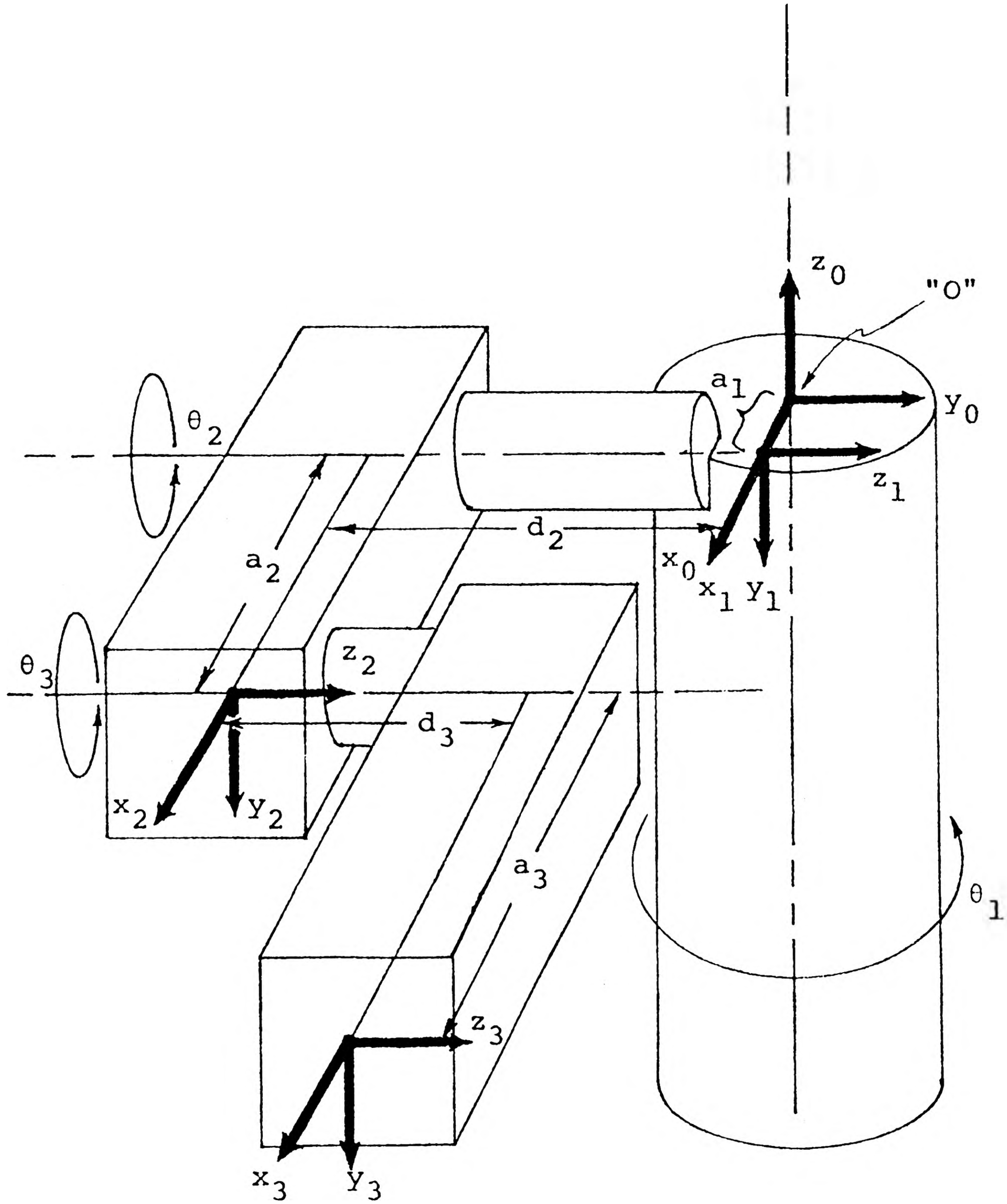


Figure 6. Coordinate Systems and Link Parameters

intersection of the waist and shoulder axes of rotation.

5. The robot has a maximum horizontal reach with respect to the origin of 23 centimeters.
6. The coordinate frames for the links are assigned in the same manner used by Paul (5) and utilized by Stelzer (4) in his robot operating system:
   a. The direction of the z-axis for each joint is defined as the axis of rotation for a rotational joint, or as the direction of movement for a prismatic joint.
   b. The position of the joints are defined such that the x-axis of all the joints are parallel to the x-axis of the base coordinate system. (Depending on how the x-axes are picked, there may be several configurations of the arm which satisfy this condition.)
7. The link parameters θ, d, a, and α are also defined in the manner used by Paul (5) such that the link transfomations on the model robot can be accomplished by (in the order given):
   a. Rotating an angle θ about the z-axis to make the x-axes parallel.

b. Translating a distance d along the z-axis to locate the origin on the common normal between the two coordinate frames.

c. Translating a distance a along the x-axis to bring the origins into coincidence.

d. Rotating an angle α about the x-axis to align the z-axes.

The coordinate frames for each joint and link parameters for the robot are shown in Figure 6 (4). Details on the coordinate frames, the link transformations, and the link parameters which are used for the control of a robot are presented by Paul (5). The values for the link parameters of the model robot are listed in Table I.

The model robot also has physical limits due to structural limits on the robot's joints and transition zones in the potentiometers, which are due to the discontinuity in single turn potentiometers. The structural or mechanical limits in the elbow and shoulder joints are due to regions in those joints which allow physical contact to be made between one part of the robot and another. These physical limits do not include the robot configurations which would allow the robot to make physical contact with its surroundings. The single turn potentiometers used on the model robot do not have any internal stops; so they are free to turn any number of revolutions in either direction. Therefore, the limiting influence contributed by the potentiometers is due to the transition zone, which is the

TABLE I. LINK PARAMETER TABLE

Joint	Joint #	Variable	α	d(cm)	a(cm)
Waist	1	θ_1	-90°	0	1.5
Shoulder	2	θ_2	0°	-6	9.5
Elbow	3	θ_3	0°	3	12

gap between the resistance winding endpoints, and is not due to any internal physical stops in the potentiometers. Figure 7 (an adaptation from (4)) shows how the potentiometer's wiper, which is attached to the joint's rotational shaft, moves between points A and B. As the wiper moves between these two points along the resistive winding, a linear voltage V_w is measured, ranging from 0 to V_r. In the region between points A and B, and not on the resistive winding, V_w is floating. If a floating V_w is used for positional feedback, a significant control error could result. Therefore, this transition region, which encompasses a field of 20 degrees, must be avoided. For the elbow and shoulder joints, this region is avoided by placing it within the mechanical limit region. Since the waist joint has no real physical limit region, the transition zone of the potentiometer will be placed in a region least likely to be used. After these regions of limitation are known, the software controller will have to be programmed to avoid them. Table II lists the ranges of each joint, which are measured from the robot's reference position (joint angles are at zero degrees) shown in Figure 6, available for movement.

B. MODEL CONVEYOR BELT

The conveyor belt is used to move objects from an identification area to an area within the reach of the model robot for sorting. The belt is composed of three linked

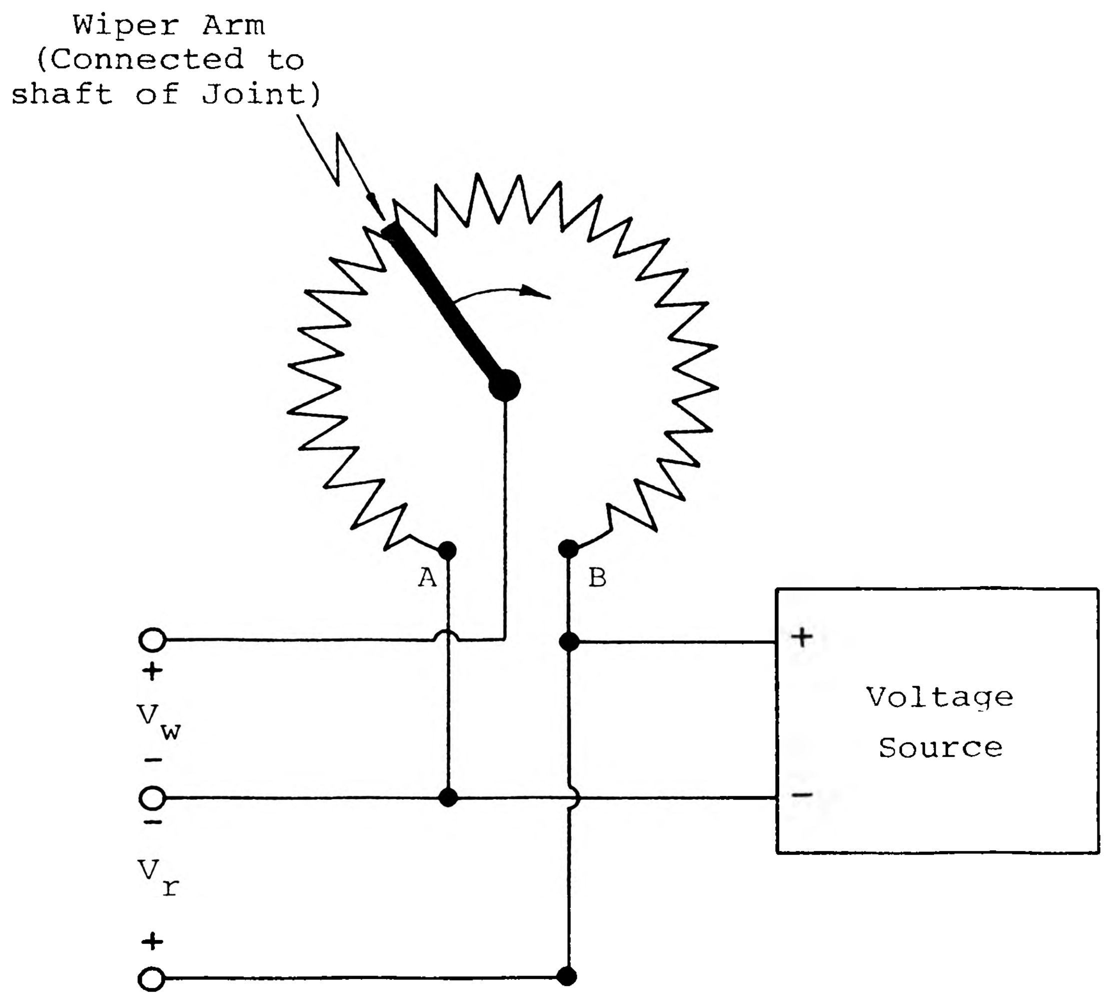


Figure 7. Feedback Potentiometer

TABLE II. ROBOT JOINT RANGES

Joint #	Legal Ranges	Illegal Range
1	0°-165° and 211°-360°	166°-210°
2	0°- 69° and 141°-360°	70°-140°
3	0°-135° and 231°-360°	136°-230°

chains covered with black cloth. The cloth serves to prevent objects from falling between the chains and also gives a solid background for the optional vision system. The conveyor belt has two end supports with two more supports in between giving it a span of approximately 72.5 centimeters. The belt is driven by the regular size Fishertechnik six-volt DC motor through a gear box with a 240:1 gear ratio.

Two pairs of infrared light emitting diodes (LED's) and photodetectors are placed along the conveyor to monitor object position. The first pair is located at the beginning of the conveyor belt to monitor when an object is in the identification area. The second pair is located beside the robot and just outside the robot's reach to monitor when an object is nearby.

Plastic bins are located around the robot and the robot end of the belt for placement of the sorted objects.

C. OPTIONAL VISION SYSTEM

The optional vision system consists of three main components: an ITC Ikegami television (TV) camera, an IVS (Interactive Video Systems) video digitizer, and an Apple II Plus microcomputer. The TV camera is mounted above the identification area on the conveyor belt. When an object passes by the first photodetector, a picture of the object is taken by the TV camera, the picture is converted into binary information by the digitizer, and the binary

information is processed by the microcomputer. The pattern recognition software was written by K. F. Shen and is documented in his thesis "A Computer Vision System For Industrial Robots" (6).

Figure 8 shows the complete system: the TV camera (upper left), the model robot system (lower left), the computer monitor (center), the hardware controller (bottom center) which is discussed later, the Apple computer (to the right of the hardware controller), the digitizer monitor (upper right), and the digitizer (lower right).

Figure 8. Model Robot System

IV. MICROCOMPUTER INTERFACE

After the robot system is designed and built, the type of software and hardware controllers must be chosen. The software controller or computer must be chosen first since it generally is the major component of the controller and its capabilities define the limitations of the system and function load of the hardware controller. Concerns or considerations used in the choice of an optimal computer are:

1. Cost of the computer.
2. Programming languages available.
3. Processing speed of the computer.
4. I/O capabilities of the computer.
5. Developmentability or expandability of the computer.
6. Scope of peripherals available for the computer.

For reasons 1, 4, 5, and 6 the Apple II Plus microcomputer is chosen for the software controller of this system. Another factor in this decision is the fact that the video digitizer system is designed to interface easily to an Apple II Plus microcomputer. Therefore, the Apple computer is serving both as a processor for the vision system and as a controller for the robot system.

To allow the Apple computer to communicate with the robot system, a special I/O interface board was designed and

implemented. Figure 9 shows the finished interface board.

The total functions the interface is to perform are determined by the hardware/software balance considerations and the computer characteristics. Both the hardware/software balance and computer characteristics determine what types of inputs and outputs the computer has to process. The computer characteristics also determine the functions and processing to be done by the interface board due to the unavailability of the function or process by the computer.

Since the Apple computer uses a 6502 eight bit processor, all the I/O data will have to be contained in one byte (eight bit) words.

I/O considerations for the computer include: pulse-width modulation for controlling the DC motors in the robot system, the capability of controlling each device in the system individually, and the ability to monitor the status and condition of each device in the system. Also, since the control of the system is performed digitally, sampling is required. The Apple II Plus computer has no internal, adjustable real-time clock; therefore, the interface must include a real-time clock for sampling.

Thus, the function of the interface board consists of three distinct parts: computer outputs, computer inputs, and a real-time clock.

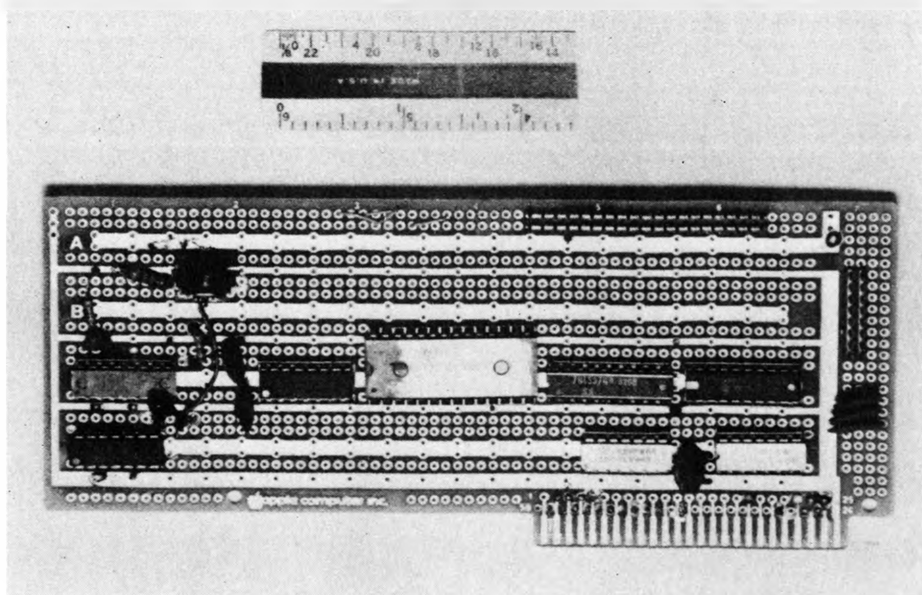


Figure 9. Interface Board

A. COMPUTER OUTPUTS

From the I/O considerations previously mentioned, two types of outputs are used to control the robot system. The main output of the computer is the byte which produces either pulse-width modulation for the control of the system's DC motor actuators or the control signal which turns a device in the system "on" or "off". The other computer output is the device selector.

Pulse-width modulation (PWM), which is discussed in greater detail in Chapter V, is nothing more than turning a motor "on" or "off" for a period of time at a constant frequency. A computer could output 1's for the "on" and 0's for the "off" to drive an actuator through some driving logic directly; but, in the hardware/software balance considerations, this use of computer time was deemed inefficient. Instead, a hardware controller produces the necessary pulse-width modulation under the control of the computer. A control byte is sent to the hardware controller from the computer, which the hardware controller must interpret as either a PWM command for the motors or an on/off command to a device.

1. Control Byte. The control byte, which consists of eight bits of information, contains either the desired speed and direction of rotation for a particular motor or the desired status of a device. Normally, the data for the device selector would also be included in the control byte, but characteristics of the Apple computer allow a separate

method for the device selection.

Since there are only two directions for a motor to turn and only two states (on/off) for a device to be in, the same single control bit is used for both these functions. The most significant bit (d_7) is used for this control bit. If d_7 is a one then a motor is defined to make a forward rotation or a device is defined as being "on". If d_7 is a zero then a reverse direction is defined for a motor or an "off" status is defined for a device.

The remaining seven bits d_0-d_6 contain the desired speed of the motors and are ignored by the other devices. Seven bits allow 128 different speed settings, including a speed of zero. Figure 10 shows the format of the control byte.

2. Device Selector. The Apple II Plus computer has sixteen buffered address lines. This address bus is available to all eight peripheral slots on the Apple. The least significant four lines of the address bus are used for addressing I/O locations within a given slot, sixteen in all (7). The hardware controller is designed to control a total of seven motors or devices. Therefore, only three of these four lines are used for the decoding in the device selection. Lines a_0-a_2 are used to provide the data for the device selector. Line a_3 is used for control logic on the interface board. Thus, the device selector is part of the overall addressing of the slot in which the interface board resides. When the slot of the interface board is addressed

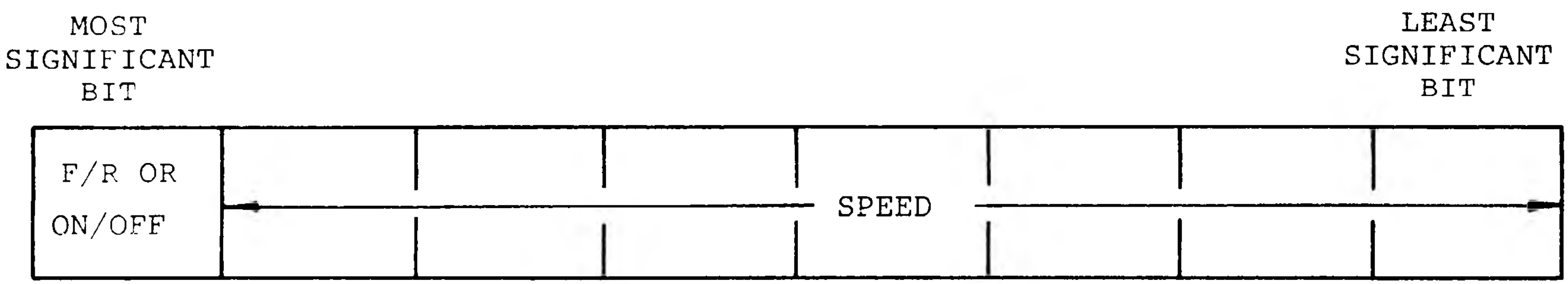


Figure 10. Control Byte Format

to output the control byte, the least significant three bits of the address are stripped off and decoded by a three to eight decoder (74LS138) for the selection of the desired motor or device. This method of device selection optimizes the Apple's memory-mapped I/O feature. Therefore, each motor or device in the robot system looks like a location in memory to the Apple computer.

Table III lists the addresses for the selection of both motors and devices by the device selector when the interface board resides in slot 1.

The computer outputs the control byte by storing it at a particular memory location. A buffer (74LS373) on the interface board places the control byte on a parallel bus to the hardware controller. As previously mentioned, the address of the memory location is decoded and the selected motor or device controller board latches onto the control byte. To prevent undesired device selection when the computer is performing some other function, the device selector (74LS138) is only enabled when the Device Select, Write, and a_3 lines of the computer are all low (zero). The Device Select line, for each peripheral slot on the Apple computer, is only low when that particular slot is addressed. The Write line is low whenever the computer is writing to a memory location. A_3, which is the fourth address line on the Apple's address bus, is low for addresses $C090-$C097 (slot 1). The circuit for the device selector is provided in Appendix A.

TABLE III. INTERFACE BOARD ADDRESSING FOR OUTPUTS

Output Function	Device	a_3	a_2	a_1	a_0	$R/\overline{W}$	Address (Slot 1)
Device Selector	Motor 1	L	L	L	L	L	$C090
	Motor 2	L	L	L	H	L	$C091
	Motor 3	L	L	H	L	L	$C092
	Motor 4	L	L	H	H	L	$C093
	Motor 5	L	H	L	L	L	$C094
	Motor 6	L	H	L	H	L	$C095
	Electromagnet	L	H	H	L	L	$C096
	Clock Clear	L	H	H	H	L	$C097
Start ADC0808	Potentiometer						
	#1	H	L	L	L	L	$C098
	#2	H	L	L	H	L	$C099
	#3	H	L	H	L	L	$C09A
	#4	H	L	H	H	L	$C09B
	#5	H	H	L	L	L	$C09C
	#6	H	H	L	H	L	$C09D
	#7	H	H	H	L	L	$C09E
	#8	H	H	H	H	L	$C09F

B. COMPUTER INPUTS

From the I/O considerations mentioned in the beginning of this chapter, two types of inputs are used to monitor the robot system. There are only two types of inputs because the system will monitor only the status of a device or the position of a motor. Therefore, the inputs are known as the status byte and the position byte.

1. Status Byte. Although the status byte allows for the status of eight devices to be read, only four devices are currently being monitored. The four devices being monitored are: the two photodetectors along the conveyor belt, the end-of-conversion (EOC) flag on the analog to digital converter (A/D), which is used in conjunction with the position byte, and the computer/manual mode switch on the hardware controller.

The EOC flag status bit is placed in the most significant bit position, d_7, since it has the highest priority. The EOC flag must be read before each position byte can be read to ensure the position byte's validity. If d_7 is a one then the data contained in the position byte is valid.

The computer/manual mode switch status is assigned to the least significant bit, d_0. If d_0 is a one then the hardware controller is in the computer mode. If d_0 is a zero then the hardware controller is under manual control.

The next bit, d_1, is assigned to the status of the

first photodetector which is located at the identification area of the conveyor belt. The third bit d_2 is assigned to the status of the second photodetector which is located near the robot. For both of these photodetector status bits, a one signifies that the infrared beam has been broken.

The status byte is stored by a 74LS374 tri-state latch on the interface board and is updated every microsecond. To place the status byte onto the Apple's data bus, the output of the status latch (74LS374) is enabled. The output of the status latch is enabled when $\overline{\text{Device Select}}$ is low, a_3 is low, and the Read/$\overline{\text{Write}}$ line is high. As previously mentioned in the computer output section, an address of $C090 produces a low signal on the $\overline{\text{Device Select}}$ and a_3 lines. The Read/$\overline{\text{Write}}$ line is high whenever the Apple computer is performing a read. Figure 11 shows the format of the status byte.

2. Position Byte. The second computer input is the position byte. As mentioned in Chapter III and shown in Figures 5 and 7, the output voltage of a potentiometer mounted on a joint's rotational shaft is a linear function of the angular position of the joint. To convert the positional data into a form which is understandable to the computer, an eight bit A/D converter (ADC0808) is used.

The output of an eight bit A/D is an eight bit binary number which ranges in value from 0 to 255. If the potentiometers allowed a full range of 360 degrees to be measured, an eight bit A/D would produce a resolution of 1.4

MOST SIGNIFICANT BIT							LEAST SIGNIFICANT BIT
EOC FLAG					SECOND DETECTOR	FIRST DETECTOR	COMPUTER/ MANUAL

Figure 11. Status Byte Format

degrees per bit. Since there is a 20 degree unusable range in the potentiometers, a scaling factor of 340/256 instead of 360/256 must be used in the position calculating software. Since it takes a finite amount of time for the A/D to perform the conversion from the analog input to the digital output, a signal from the A/D is used to tell the computer when the A/D is finished. The signal used is the previously described End-of-Conversion (EOC) flag, which is used as one bit of the status byte. According to the ADC0808 specifications (8), the maximum conversion time takes about 116 microseconds. This conversion time defines the minimum sampling time of the robot's motors.

An important feature of the ADC0808 is the 8-channel multiplexed input. The 8-channel multiplexed input allows a single A/D to perform the conversions for eight separate analog inputs. Thus, for this robot system the interface board could monitor a total of eight motor positions. The main disadvantage of this approach is that the total sampling time for all the joint positions is a multiple of the A/D's conversion time plus the time required to start the conversions for each input used. To start the conversion for a particular analog input on the A/D, the desired input is first addressed and latched. The ADC0808 has a built in three to eight decoder for addressing its inputs. Since each of the motors in the robot system is already addressed by the three lines a_0-a_2 for the device selector (74LS138), these address lines are also used to

address the analog inputs. The ALE (Address Latch Enable) input on the A/D, which is tied to the START input line on the A/D, latches the address for the decoder when it is strobed by a rising pulse edge on the START line. The address is latched and the conversion started when lines a_0-a_2 contain a valid address, a_3 is high, $\overline{\text{Device Select}}$ is low, and the Read/$\overline{\text{Write}}$ line is low as when the memory locations $C098-$C09F (slot 1) are addressed for sending data. Table III lists the addresses which start the conversion for the position byte of each motor.

Originally, the analog input lines coming from the potentiometers on the robot to the A/D on the interface board were contained in a 40-conductor ribbon cable. Upon testing the analog input lines for noise, a two-volt (peak to peak) digital signal was measured on the analog lines. The noise was primarily produced by digital cross-talk from the computer's output lines. The noise was reduced to about a half-volt (peak to peak) by separating the analog and digital lines and by using capacitors to filter out the noise on the A/D's inputs. The analog lines were placed on a twisted pair ribbon cable with one conductor of each pair tied to ground for shielding. Capacitors with a value of .1 microfarads were used on the A/D's inputs since the analog signals from the potentiometers are relatively slow time-varying signals. The small amount of noise that still remains does not seem to affect the operation of the system.

The output of the A/D is also tri-state and must be

enabled to place the position byte on the Apple's data bus. The output of the ADC0808 is enabled when the $\overline{\text{Device Select}}$ is low, a_3 is high, and the Read/$\overline{\text{Write}}$ line is high. Again, if the interface board is in peripheral slot 1, a Read from address $C09F will perform the operation. Table IV lists the addresses for reading both the status byte and position byte inputs.

Therefore, the outputs on the interface board are defined to be when a $\overline{\text{Write}}$ is being performed and the type of output is distinguished by the value of line a_3. During a $\overline{\text{Write}}$, a_3 defines whether the interface board is outputting a control byte to a particular motor or device or outputting an address and start command to the A/D. The inputs on the interface board are conversely defined to be when a Read is being performed and are also distinguished by using the address line a_3. During a Read, a_3 defines whether the data being placed on the Apple's data bus is the status byte or the position byte. Although they are ignored by the control logic, lines a_0-a_2 are all defined to be low for reading the status byte and high for reading the position byte, for convenience.

C. <u>REAL-TIME CLOCK</u>

The purpose of the real-time clock is to provide the computer with an independent clock which will provide the timing for the required sampling. The sampling done in the robot system involves the sampling of the status byte and

TABLE IV. INTERFACE BOARD ADDRESSING FOR INPUTS

Input Function	Device	a_3	a_2	a_1	a_0	$R/\overline{W}$	Address (Slot 1)
Read Status Byte	74LS374	L	L	L	L	H	$C090
Read Position Byte	ADC0808	H	H	H	H	H	$C09F

the position bytes for all the motors (potentiometers) on the robot. The sampling frequency is determined by the conversion times for the A/D, the delay times in the data transfer from the interface board to the Apple's CPU (Central Processing Unit), and the processing time of the controlling software. The conversion and data transfer times are constant, but the processing time for the controlling software is variable.

Since the robot system is designed to be general in nature for control algorithm testing, the clock is adjustable. Adjustment of the clock's period is performed manually by a potentiometer on the interface board.

The actual clock pulses for the real time clock are produced by a dual retriggerable monostable multivibrator (74LS123). As the name implies, each multivibrator has a single stable output level (high or low). If a monostable multivibrator is triggered, the output jumps to the opposite level and remains there for a period of time, which is determined by an RC (resistor-capacitor) time constant. After that period of time, the multivibrator returns to its stable level until it is retriggered. By using two multivibrators with opposite stable outputs for triggering each other, a clock pulse is produced in which the duration of both the high and the low levels are determined by the values of the RC pairs. The multivibrator which determines the duration of the low level is set at about 90 nanoseconds by its RC time constant. The multivibrator which determines

the duration of the high level has a capacitor with a value of one microfarad and a potentiometer for the manual control of the duration. Presently, the duration can be adjusted from 4 to 70 milliseconds.

To optimize the Apple's features, the sampling is interrupt driven. Therefore, each clock pulse interrupts the Apple's CPU and is serviced by a user defined interrupt service routine. The interrupt service routine can contain a counter which will then determine the final sampling frequency. As an interrupt, the clock pulse must be latched and then be cleared. To perform this function a D-type flip-flop (74LS74) is used. The clock pulse from the monostable multivibrators clocks the flip-flop, which has its input (D) tied high. The inverted output ($\overline{Q}$) of the flip-flop is tied to the Apple's Interrupt Request ($\overline{IRQ}$) line, which is a maskable interrupt line. When this line is low, the Apple jumps to the interrupt service routine. $\overline{Q}$ goes low at each low-to-high transition of the clock since D is always high. The flip-flop is cleared ($\overline{Q}$ goes high) by the most significant line from the device selector (74LS138). Thus, to clear the interrupt the Apple must write to memory location $C097 (slot 1) as presented in Table III.

Appendix A shows the schematic for the entire interface board including the circuit for the real-time clock.

V. HARDWARE CONTROLLER

The hardware controller, which is shown in the middle of Figure 8, is the main hardware component of the robot system designed and constructed. As mentioned in the previous chapters, the hardware controller has several distinct and well defined functions.

The major function of the hardware controller is the control of the DC motors (actuators) by pulse-width modulation (PWM) under the supervision of the Apple computer. To perform this function, the hardware controller accepts an eight bit command via a ribbon cable and translates it into a direction and speed using PWM for a specific motor. If the command from the Apple is to turn a device, such as the electromagnetic gripper, in the system "on" or "off", the hardware controller only uses the most significant bit of the eight bit command to execute the function. Chapter IV gives the detailed format of the control byte.

Other functions of the hardware controller include the signal processing of the infrared photodetectors, providing an interconnection point between the model robot system and the Apple computer, providing regulated power to both the model robot system and electronics within the controller, and providing the reference voltages for the system's potentiometers.

The final function of the hardware controller is to

provide an operator with the ability to manually control the model robot system.

For the hardware controller to fulfill the requirements of a practical experimental system, as stated in the beginning of Chapter III, the controller is to preserve maintainability and allow expandibility as it performs the previously mentioned functions. These requirements are fulfilled by making the controller modular and separating the functions into distinct function boards. Each board can contain one or more functions and each board is independent of all other boards. Therefore, each board can easily be removed for maintenance or modification. Common functions, such as the PWM for all the different motors, have identical and interchangeable boards. Inside the hardware controller case each function board has a corresponding edge connector with all the necessary wiring busses and control lines attached. The hardware controller has eight such connectors. Table V lists the function(s) wired into each edge connector with the left-most edge connector in the hardware controller designated as connector one.

To fulfill the function of an interconnection point, a standard DB-25 (P and S) connector is used to join the model robot system to the hardware controller. All the electrical components of the model robot system are placed on two 12-conductor cables which terminate with a DB-25P (male) connector. The female connector (DB-25S) is mounted on the hardware controller's case. Thus, the model robot system

V. ASSIGNMENT OF EDGE CONNECTOR FUNCTION(S)

Connector #	Function
1	Control Waist Joint
2	Control Shoulder Joint
3	Control Elbow Joint
4	No Function
5	No Function
6	Control Conveyor Belt
7	Control Electromagnet and Photodetectors
8	Distribute Wiring Busses and Regulated Power

can be easily disconnected for transport. Details of the pin assignment of the DB-25P connector to the model robot system are presented in Appendix B. The remaining functions for the hardware controller are separated into the following components: the distribution board, the pulse-width modulator boards, the electromagnet and photodetector board, and the manual control.

A. DISTRIBUTION BOARD

The distribution board, which is shown in Figure 12, has four basic functions. One function of the board is to provide a termination point for the ribbon cables coming from the interface board in the Apple computer. The cables have female connectors which plug onto corresponding male headers on the distribution board. The pins on the male headers are tied to either devices on the board itself or to the contacts on the bottom edge of the board, which plug into the edge connector. This allows direct connection to the hardware controller's wiring bus. Appendix C lists the wiring connections at the distribution board's edge connector. Two sources of connections on the distribution board's connector are from the DB-25S (model robot system) and the manual control.

The latter source is for the second function of the distribution board. The hardware controller is designed so that manual control is executed identically to that of the computer by the pulse-width modulator boards. The manual

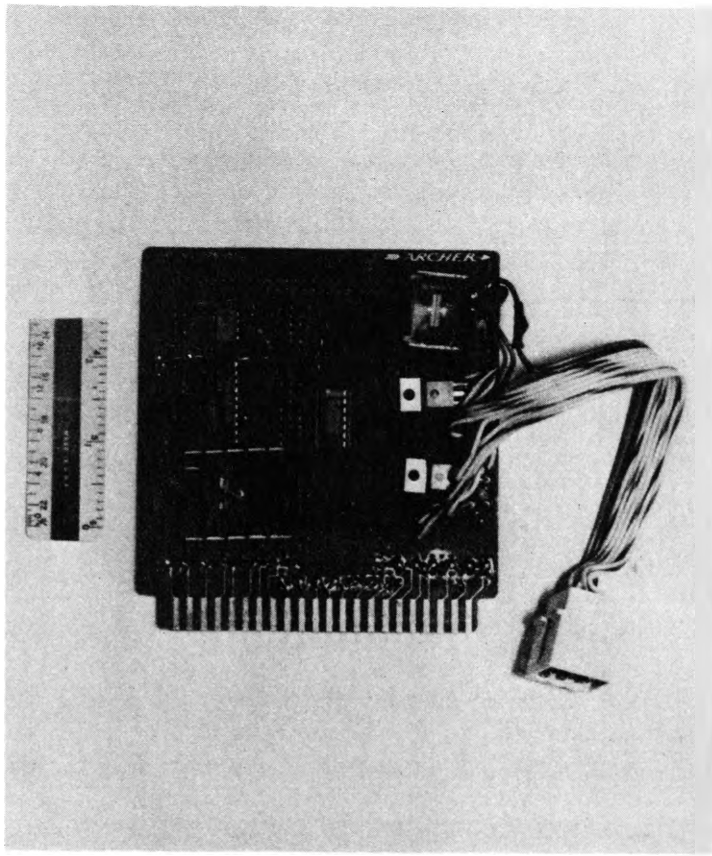


Figure 12. Distribution Board

control, which is presented in greater detail later, has forward/reverse (F/R) control lines coming from switches mounted on the hardware controller's case. These F/R control lines are tied to the F/R control lines from the computer on the pulse-width modulator board. To prevent bus conflicts from occuring these lines are buffered with tri-state latches. The F/R line from the computer is buffered by a tri-state latch on the pulse-width modulator board. The F/R lines from the manual control are buffered by a tri-state latch (74LS374) on the distribution board. The latch (74LS374) is updated every millisecond by an LM555 timer. The output of the manual F/R latch is enabled and the output of the latch on the pulse-width modulator board is disabled when the hardware controller is in manual mode (computer/manual line is low).

Another function of the distribution board is to provide a power-up trigger. When the power of any circuit is first turned on, the state of the devices in the circuit is unknown. Such a state in the hardware controller could cause uncontrolled movement in the model robot system at power-up, which would be very undesirable. Therefore, the power-up trigger is used to clear devices in the hardware controller and initialize the system. Since this is to be done only once at power-up, a single pulse is used. The single pulse is produced by a Schmitt-Trigger (74LS14). The output of the 74LS14 is placed on the power-up trigger control line, which is available at each function board

connector.

The final function of the distribution board is to provide five-volt regulated power to the 7-segment LED (light emitting diode) displays on the manual control, and provide the reference voltages for the potentiometers on the robot and the manual control including the reference voltages for the corresponding A/Ds. To provide the five-volt regulated power, four MC7805 voltage regulators are used. Two MC7805s provide the power for the fourteen possible 7-segment LEDs on the manual control display. One MC7805 provides the power for the reference voltage across all the potentiometer resistance windings on the model robot and also provides the reference voltage used by the A/D (ADC0808) on the interface board. The last MC7805 provides the reference voltages for the speed control potentiometers on the manual control and their corresponding A/Ds (ADC0804's) on the pulse-width modulator boards. The input power to the voltage regulators is provided by an eight-volt, five amp external power supply. Because all the contacts on the distribution board are in use and since a significant amount of current (about .5 to .8 amps) is required to power these devices, a separate cable with a nine-pin connector is used for a regulated power bus. Appendix D shows the complete schematic for the distribution board.

B. PULSE-WIDTH MODULATOR BOARD

The pulse-width modulator (PWM) board is the major function board in the hardware controller. The PWM board's functions include producing a pulse-width modulated signal, providing the speed display on the manual control, and driving the actuators (motors) on the model robot system using the PWM signal under either computer or manual control. Figure 13 shows the pulse-width modulator board.

For a better understanding of the reasoning used in choosing a PWM signal for controlling the motors, a brief background on PWM is presented.

1. Pulse-Width Modulation Theory. For control systems, DC motors are preferred over AC motors because DC motors are capable of developing the highest starting/stall torques, have a wide speed range, simplified drive and control circuitry, and respond quickly to changes in control signals (9). For these reasons, DC motors are useful as actuators in robotic systems.

The torque or angular force of the motor is directly proportional to the current in the motor's windings. The greater the voltage across the motor's terminals, the greater the current through the windings. These concepts are presented quantitatively in the following equations:

$$T=K_T*I \qquad (5\text{-}1)$$

$$V=L*dI/dt + R*I + E_g \qquad (5\text{-}2)$$

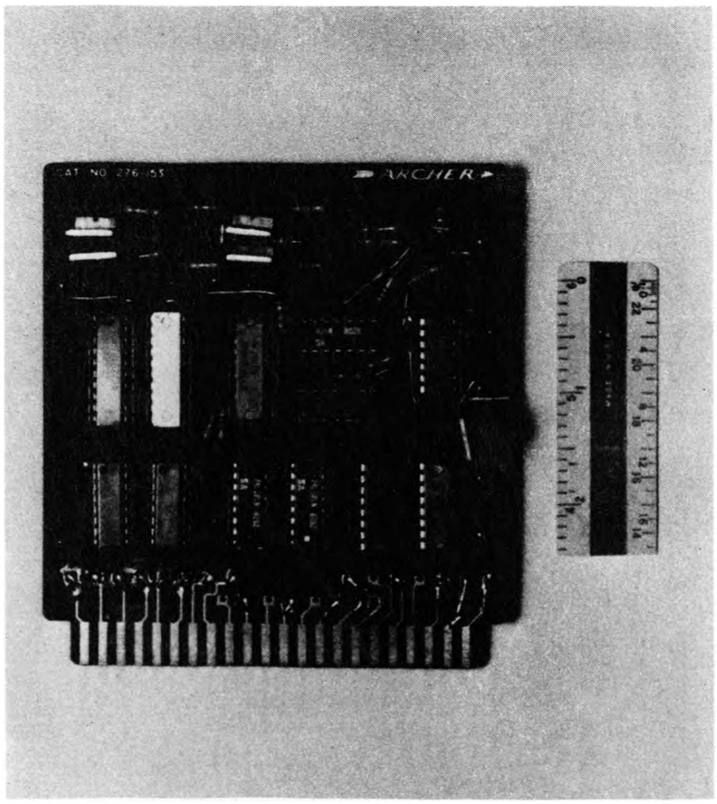


Figure 13. Pulse-Width Modulator Board

where T is the torque (Newton-meters), K_T is the torque constant for a given motor, I is the current (Amperes), V is the terminal voltage (Volts), L is the inductance (Henrys), R is the terminal resistance (Ohms), and E_g is the internally generated voltage (Volts) which is proportional to the motor's velocity (10).

It is also important to note that the speed (angular) is inversely proportional to the torque. The greatest speed is achieved by the motor when there is no load (generated torque is zero) and the greatest torque exists when the motor is stalled (speed is zero). This relationship is shown graphically in Figure 14(a) (from (10)). The speed-torque curve for a motor shifts for different terminal voltages and forms a family of characteristic curves, as is shown in Figure 14(b) (from (10)). Equations 5-3 and 5-4 convey this information quantitatively.

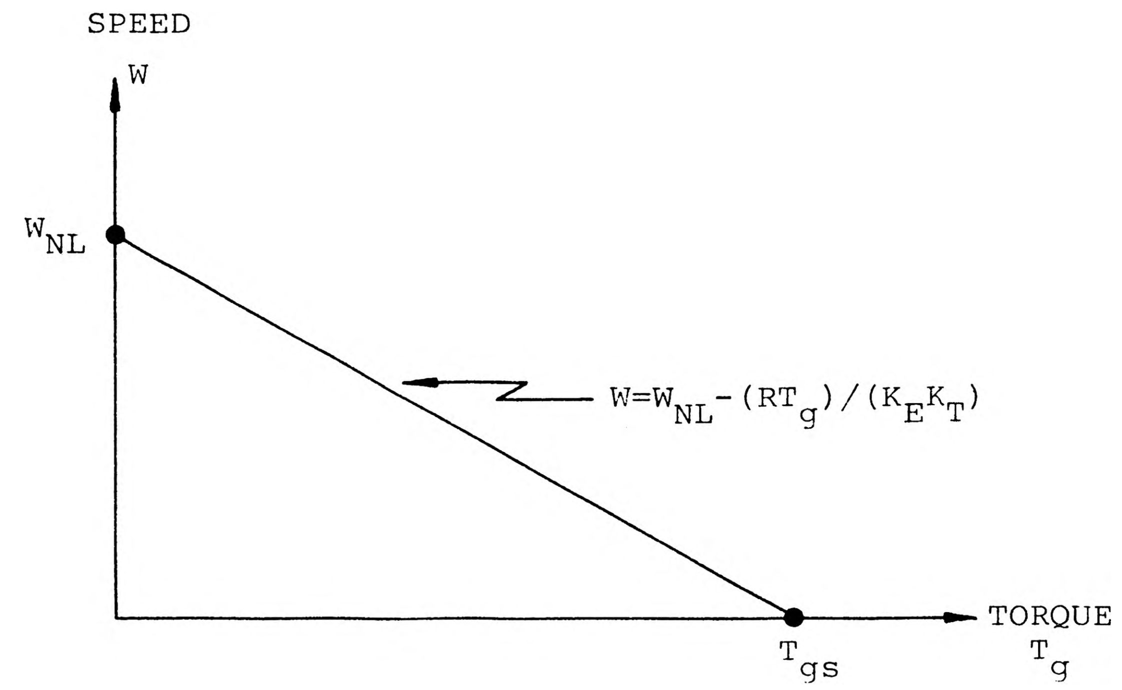
$$W_{NL}=V/K_E \qquad (5\text{-}3)$$

$$T_{gs}=V*K_T/R \qquad (5\text{-}4)$$

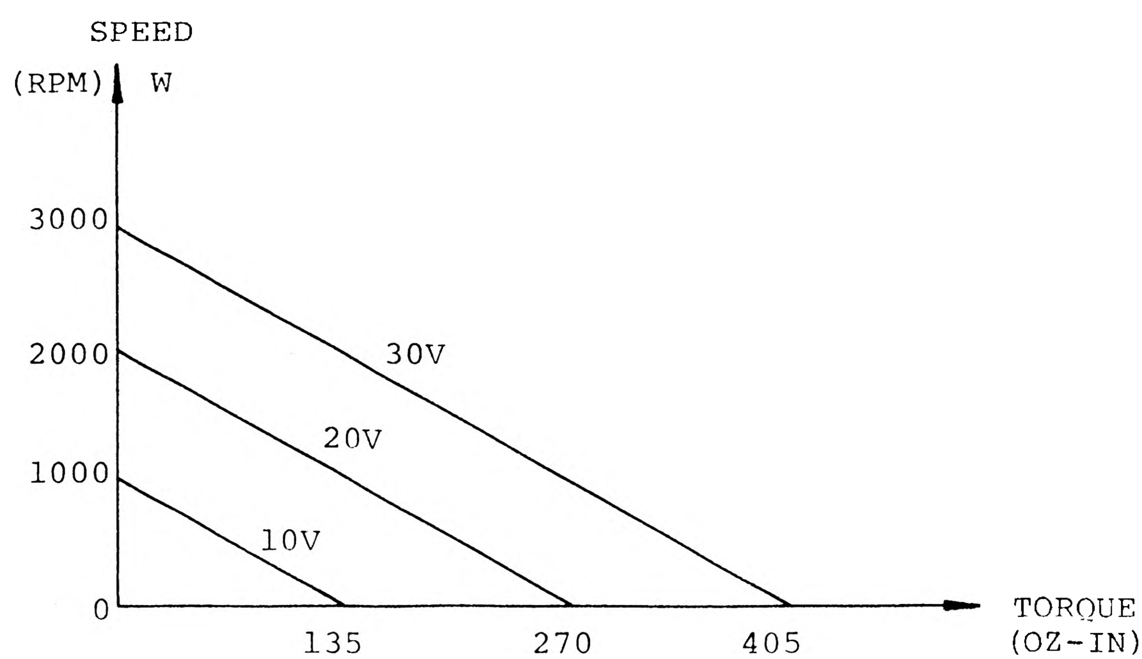
where W_{NL} is the angular speed (RPM) under no-load conditions , V is the terminal voltage, K_E is the motor's voltage constant, T_{gs} is the generated torque at stall, and R is still the terminal resistance (10).

Therefore, the greater the terminal voltage, the higher are the values for the no-load speed and stall torques.

From this information it can be concluded that if the



a. DC Motor Speed-Torque Curve



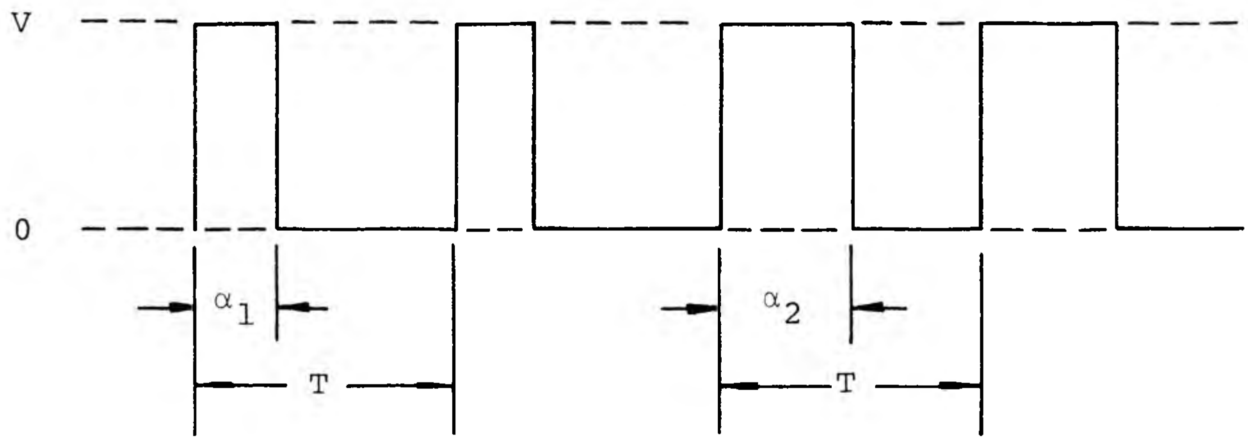
b. Speed-Torque Curves for Terminal Voltages of 10, 20, and 30V

Figure 14. Speed-Torque Curves

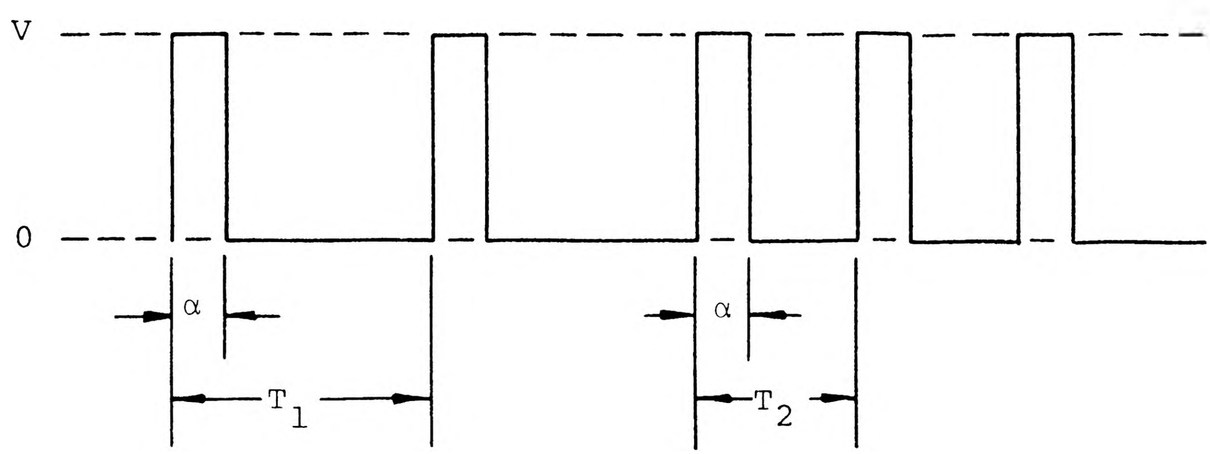
terminal voltage is used to control the speed as equation 5-3 suggests, the torque will follow in direct proportionality as stated by equation 5-4. Small torques at the slow speed ranges is an undesirable attribute. Another problem in using direct voltage control is the dissipation of power by the driving circuitry which is supplying constant power to the motor. Thus, an alternate control method is required. One alternate control method is to use a switching power driver circuit to drive the motors. By switching between a conductive and nonconductive state the driving circuitry spends a reduced amount of time in the high power dissipation region.

Two methods of power switching are pulse-width modulation (PWM) and pulse-frequency modulation (PFM). The differences are shown in Figure 15 (10). Figure 15 shows that for PWM the pulses occur at a constant frequency with a variable pulse width while for PFM the pulses have a constant pulse width and occur at a variable frequency. However, for DC motor control, PFM is not particularly suitable because of the DC motor's frequency response. At the low frequencies the motor rotates in a jerky manner, and at the higher frequencies the inductance of the motor cuts the performance of the motor down to zero. PFM is used to control stepper motors since they rotate in a discrete and uniform manner at every pulse. The frequency of the pulses determines the stepper motor's speed.

The speed of a DC motor is determined by the width of



a. Pulse-Width Modulated (PWM) System



b. Pulse-Frequency Modulated (PFM) System

Figure 15. Pulse-Width and Pulse-Frequency Modulated Waveforms

the pulses when using PWM. In PWM control systems the supply voltage is usually being switched "on" and "off" at a constant frequency, which produces a waveform in the motor current. Figure 16 (10) shows a possible steady-state voltage and current relationship in a PWM system. As the voltage is switched "on" and "off", an effective current, I_{RMS}, and an average current, I_{AV}, are established. The values of the currents are determined by the width and frequency of the pulses.

Another important term used in describing PWM signals is the duty cycle. The duty cycle is the percentage of time a pulse is "on". It is calculated by using the following equation:

Duty Cycle=(Pulse width)/(Period of the pulse) *100 (5-5)

The greater the duty cycle, the larger the current.

Both the torque and speed control performance, which is often limited by power dissipation, are affected by this current. The average motor current (I_{AV}) determines the motor torque as stated in equation 5-1. The power losses in the motor are evaluated by the following equation:

$$P_L = R * I_{RMS}^2 \qquad (5\text{-}6)$$

where P_L is the power loss (Watts) and R is the terminal resistance (10). The relationship between the values of the

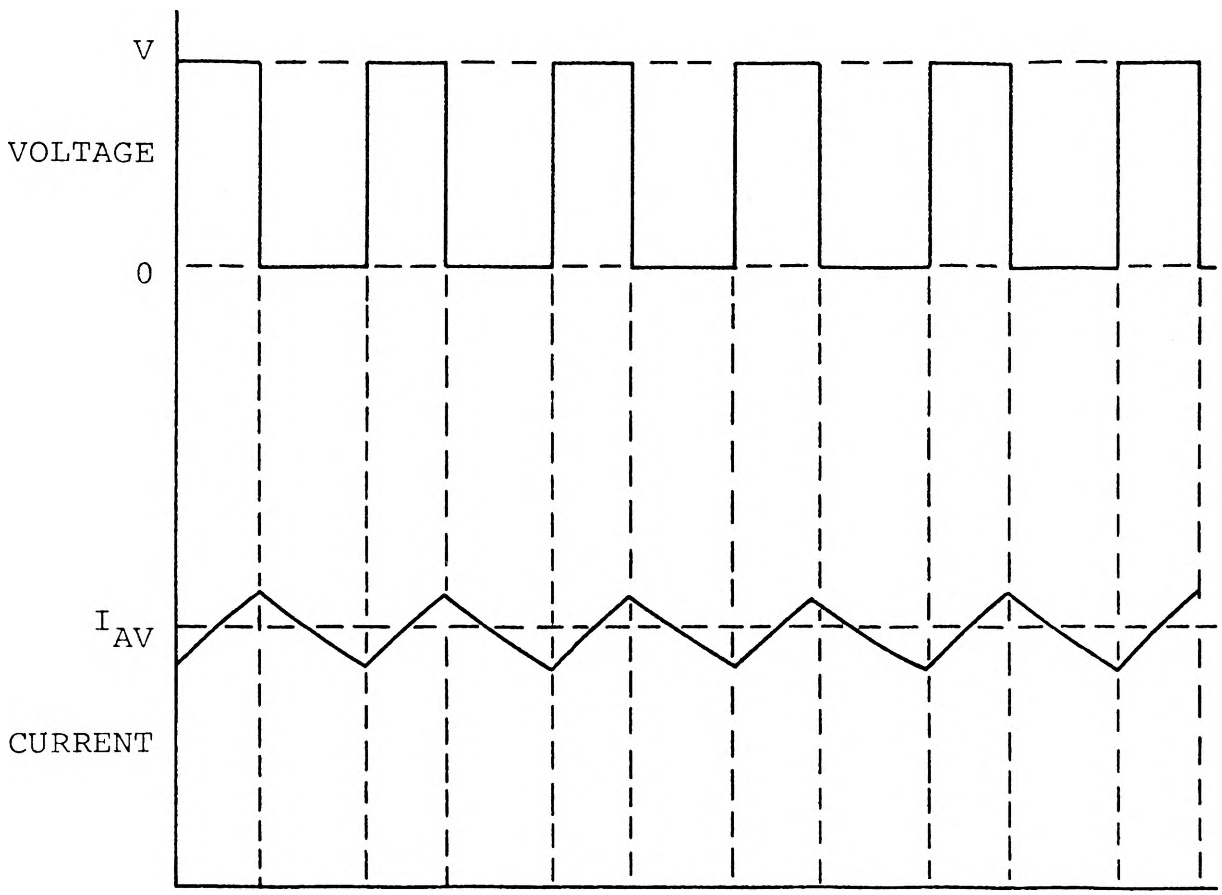


Figure 16. Voltage and Current Relationship in a PWM Control System

RMS current and the average current also has a great influence on the losses in the system. Equation 5-7 (10) is known as the form factor (k) and is one expression to relate these two currents.

$$k=(I_{RMS})/(I_{AV}) \tag{5-7}$$

Substituting equation 5-7 into 5-6 gives:

$$P_L=R*k^2*I_{AV}^2 \tag{5-8}$$

It is clear from equation 5-8 (10) that any value of k greater than one increases the power losses and thereby decreases the performance quadratically.

For PWM, the form factor is dependent on the pulse frequency, and the electrical time constant and inductance of the motor. Therefore, the characteristics of the motor determine the frequency of PWM signal used to minimize the form factor and power losses.

Figure 17 shows the speed-torque curves for the large Fishertechnik motors at several duty cycles and with a pulse-width frequency of 100 Hz (Hertz). The data plotted in Figure 17 and the optimal frequency of 100 Hz was determined by Jeff Arensmeier and Chris Edwall at the University of Missouri-Rolla.

Pulse-width modulation thus provides a method to control motor speed while usually improving the power

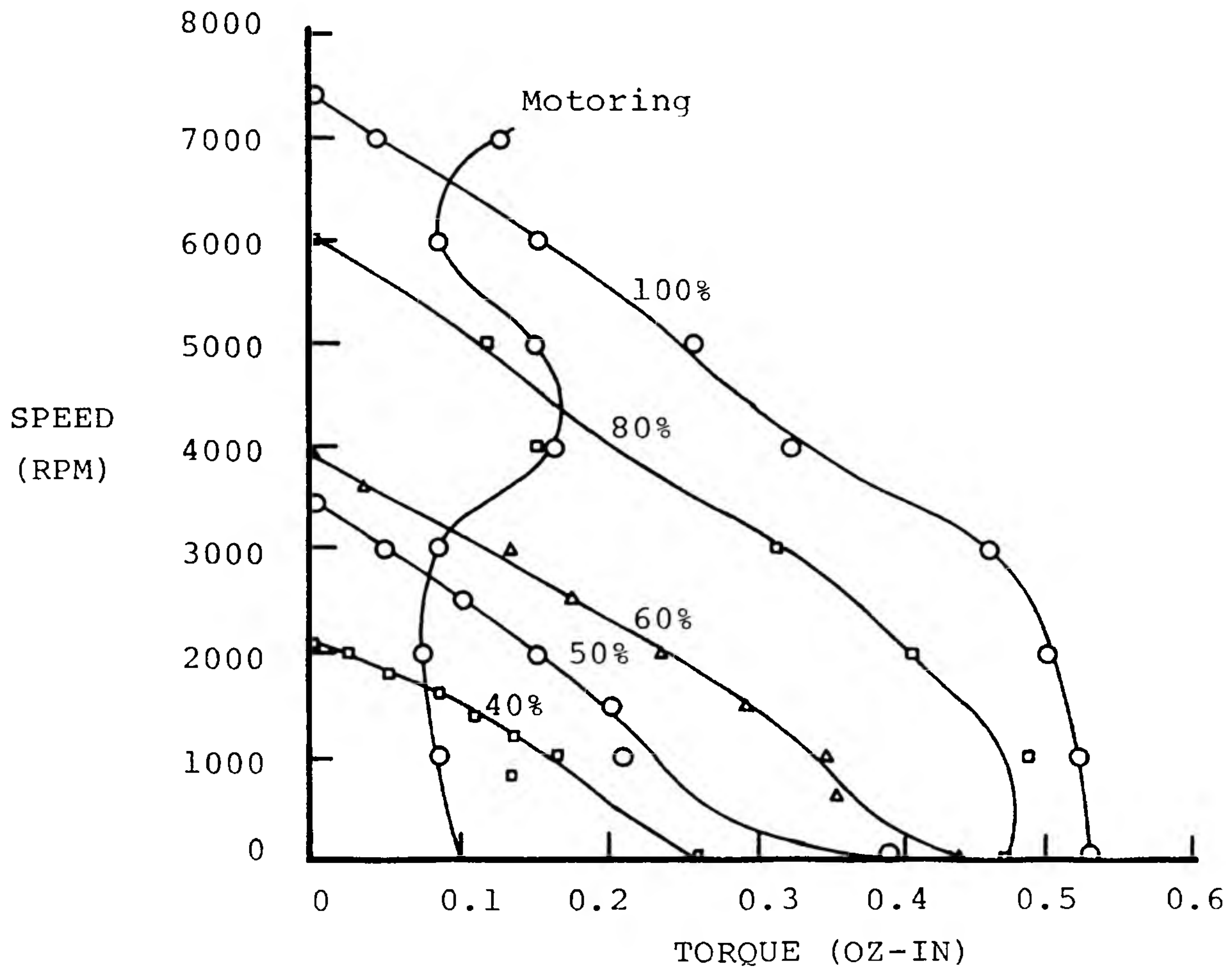


Figure 17. Speed-Torque Curves For the Large Fishertechnik Motors

dissipation.

2. Pulse-Width Modulator. Figure 18 shows a block diagram of the pulse-width modulator board for reference.

Three components on the PWM board, as shown in Figure 18, produce the PWM signal. The components are a digital comparator, a counter, and a clock. Both the comparator and counter are 4-bit devices and are cascaded to handle eight data lines.

The counter (2-74LS163's) is a synchronous binary counter which counts to 256 (eight data lines) continuously. The counter is clocked by one half of an LM556 dual timer. The frequency of the timer or clock is adjusted by a potentiometer at the top of the PWM board. To produce the optimal pulse-width frequency of 100 Hz for the Fishertechnik motors, the clock is set to produce an output freqency of 25.6 kHz (100*256). The eight bit output of the counter is used as the B-input of the digital comparator.

The digital comparator (2-74LS85's) compares the eight bit output from the counter with the eight bit word at its A-input. Whenever the A-input value is greater than the value at the B-input (counter), the A>B output of the comparator goes high. If the B-input is greater than or equal to the A-input, the A>B output goes low. Therefore, the A>B output line is the PWM signal line.

The A-input contains the desired speed data from either the computer or manual control. Since the speed data from the computer is contained only on seven data lines, the

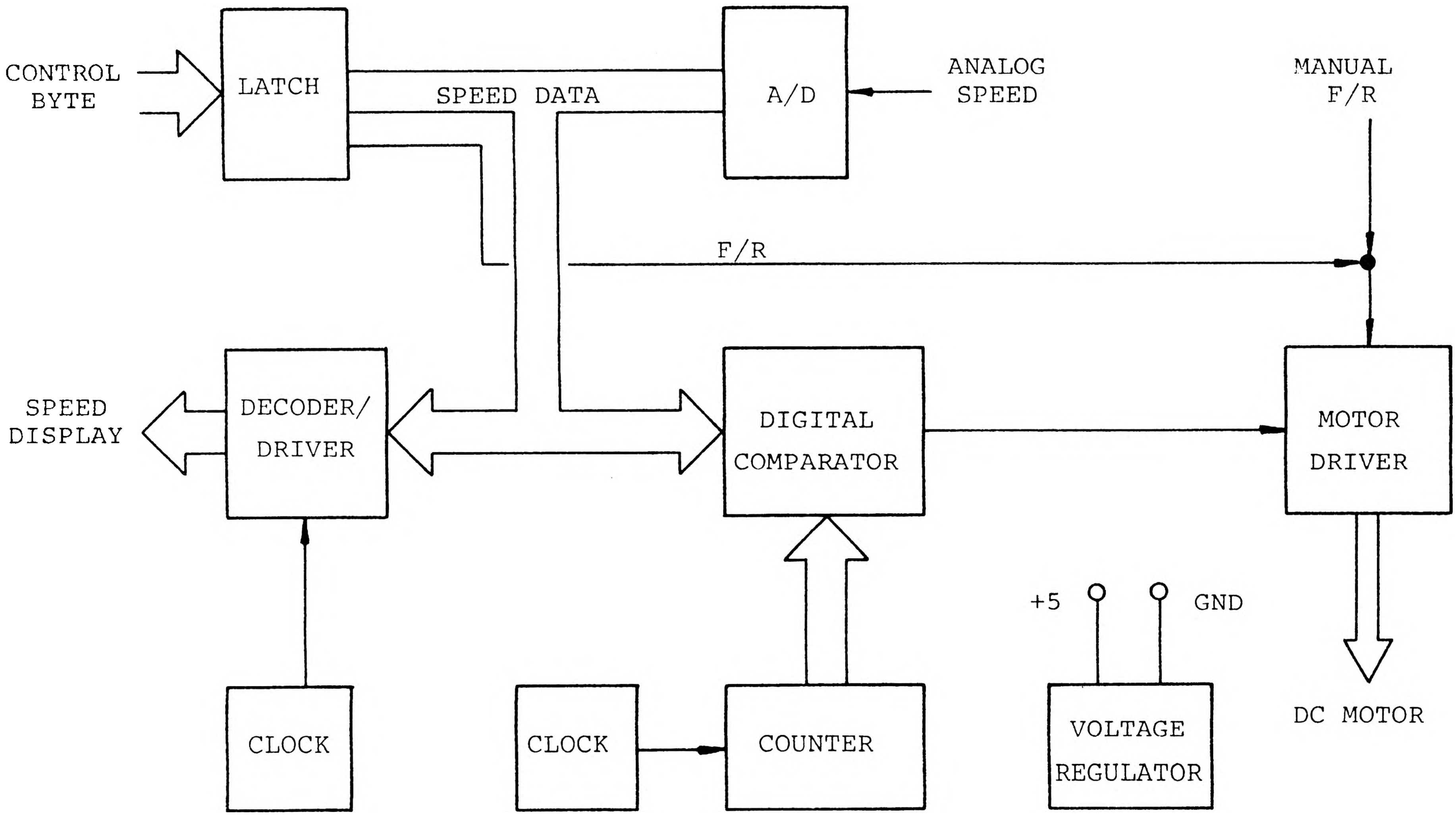


Figure 18. Pulse-Width Modulator Board Block Diagram

least significant input line on the comparator is tied to ground. Thus, there can only be 128 different inputs and 128 different PWM signals.

The A-input from the computer is furnished by a combination latch (74LS273) and tri-state buffer (74LS244). The 74LS273 latches onto the control byte from the computer when its input is clocked by the device selector on the interface board. The output of the latch serves as the input of the tri-state buffer. The output of the latch is initialized to zero at power-up. The output of the buffer (74LS244) is split into the forward/reverse (F/R) line and the speed data tied to the comparator's A-input. The F/R line (the most significant line) is tied to the F/R line from the manual control and the F/R control logic on the PWM board. The output of the tri-state buffer is enabled by the computer/manual mode control line.

The A-input of the comparator from the manual control is furnished by the tri-state output of an A/D (ADC0804) on the PWM board. The A/D receives its input from the wiper on the manual controller's speed potentiometer. The ADC0804 converts the analog speed input from the manual control to an eight bit digital word with the seven most significant bits used by the digital comparator. The ADC0804 has an on-chip clock generator; so no external clock is required. The A/D is wired to operate in the free-running mode so that it is continuously being updated. To insure start-up under all possible conditions, the power-up trigger pulse is used

to start the conversions. The output of the A/D is also enabled by the computer/manual mode control line.

Therefore, both the outputs of the latch through the tri-state buffer and the A/D are tied to the A-input of the digital comparator and are conversely enabled by the computer/manual mode control line to prevent bus conflicts.

3. Speed Display. One other set of devices connected to the comparator's A-input lines is two BCD (Binary Coded Decimal)-to-seven segment hexadeciamal latch/decoder/drivers (MC14495). Their function is to display the speed data from either the computer or the manual control onto two seven-segment LED displays on the harware controller's case. In computer mode the display shows the operator the current speed of each motor and whether the hardware controller is receiving the control byte from the computer. In manual mode the speed display is much more informative. Not only can the operator see exactly at what speed the motor is being driven, but experimentation with the speed can reveal the speed data which pertains to the minimum duty cycle for a particular motor under different loading conditions. Through other forms of experimentation other parameters of the system can be determined.

The MC14495 is externally clocked by the other half of the LM556 dual timer. The clock is currently set to update the display every millisecond.

4. Motor Driver. To convert the PWM signal supplied by the digital comparator to a power signal which can drive

an actual motor, a motor driver circuit is included on the PWM board. The motor driver consists of two parts. The first part of the driver is the control logic, and the second part is a power transistor network.

The control logic determines the direction the motor is to rotate from the F/R control line, turns the power network "on" and "off" according to the PWM signal, and also enables and disables the PWM signal line from the comparator in accordance with the manual control. The control logic consists of NAND gates (7426's) and inverters (74LS04's).

The power transistor network is a power transistor bridge which can conduct through a load in two different directions. The power transistors (TIP 29's and TIP 30's) which drive the DC motors are controlled by signal transistors (2N3906's) which switch the power transistors between cut-off and saturation. The signal transistors are controlled directly by the control logic. The power supplied to the motors by the power transistors is supplied directly from the external eight-volt power supply.

The remaining signal transistors and IC (Integrated Circuit) chips on the PWM board receive their power from an on-board MC7805 five-volt voltage regulator, which receives its input power from the external eight-volt power supply. By placing an MC7805 voltage regulator on the individual PWM boards, digital noise on the PWM boards' power bus from the other function boards is reduced.

Appendix E lists the wiring assignment for the

pulse-width modulator boards' edge connectors.

Appendix F shows the schematic of the pulse-width modulator board. Detailed information on the individual chips is contained in "The TTL Data Book" (11), "The Linear Data Book" (12), and the "Motorola CMOS Data" book (13).

C. ELECTROMAGNET AND PHOTODETECTOR BOARD

The final function board is the electromagnet and photodetector board, shown in Figure 19. As the name implies, the board has two distinct functions.

The first function is the control of the electromagnet on the robot's end effector by computer or manual control. As described in Chapter IV, the most significant bit (MSB) of the control byte defines either the direction of a motor or the state of a device. Since the electromagnet is a device, a one in the MSB defines the electromagnet to be turned "on".

The MSB is latched from the computer by a D-type flip-flop (74LS74) which is clocked by the device selector on the interface board. The output of the flip-flop is tri-state buffered by a hex bus driver (74LS367) and tied to the on/off control line from the manual control (via the tri-state latch on the distribution board) and the input of a NAND gate (7426). The output of the flip-flop is cleared (set to zero) at power-up. The output of the tri-state buffer (74LS367) is enabled by the computer/manual mode control line, so it is only enabled when the on/off line

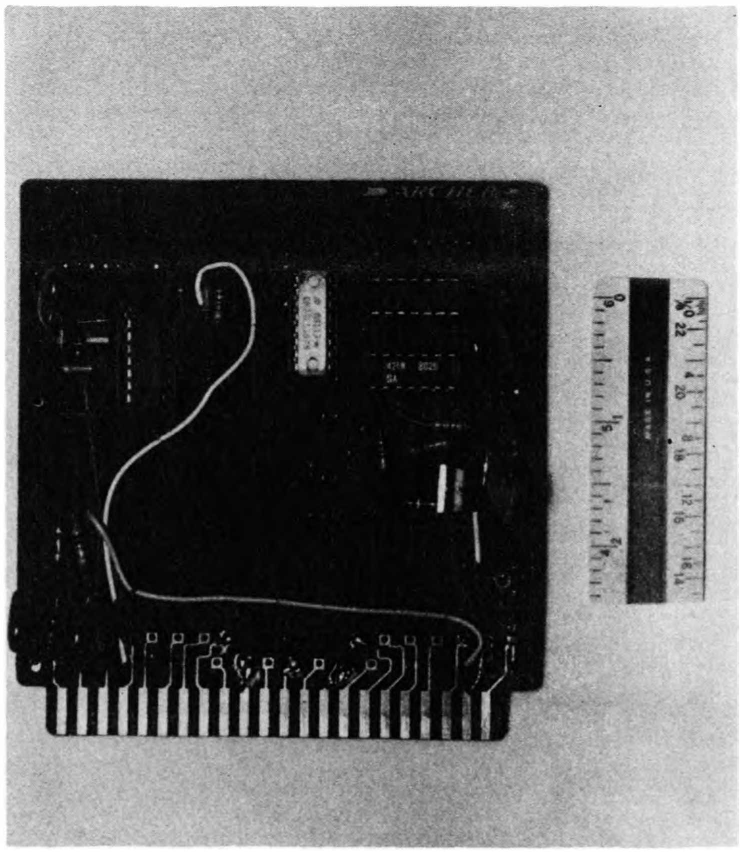


Figure 19. Electromagnet and Photodetector Board

from the manual control is disabled for line conflict prevention. The NAND gate, which has its other input tied high, is the control logic which turns a power transistor (TIP 29) "on" or "off" through a signal transistor (2N3906). When the power transistor is turned "on" by the control logic, current flows through the electromagnet and energizes the coil. A diode (1N4002) is in shunt with the electromagnet to provide a path for the inductive current when the electromagnet is turned "off".

The second function of the electromagnet and photodetector board is to monitor the two infrared photodetectors along the conveyor belt and place the status data on the input lines to the status byte latch on the interface board. The circuit for the photodetectors, which was constructed by Shawn Underwood and Dr. Randy Moss at the University of Missouri-Rolla, provides the power for the infrared LED light sources and the phototransistors used as the detectors.

When nothing blocks the light path between the LED light source and the detector, the phototransistor is in a conductive state and the signal transistors (2N3904's), which provide an output signal, are turned "off" so the output line goes high. When the light path is blocked, the phototransistor goes into cut-off and turns the signal transistors "on" (conducting) so the output line goes low. To have the outputs match the defined format, an inverter (74LS04) reverses the logic for the status byte latch's

inputs. Thus, a broken light beam for a particular photodetector produces a one at the corresponding input of the status byte latch.

Appendix G lists the wiring assignment for the electromagnet and photodetector board's edge connector. Appendix H shows the schematic of the electromagnet and photodetector board.

D. MANUAL CONTROL

The manual control, which is located on the top surface of the hardware controller, is shown in Figure 20. The purpose of the manual control is to provide direct control of the model robot system to the operator. Direct control allows the operator to test motor speeds, experiment with duty cycle responses, and demonstrate movements and configurations on the robot.

The manual control is designed to operate the model robot system through the hardware controller in a manner identical to that of the computer. For the motors both the speed and direction are controlled by the manual control. A potentiomenter provides a speed input to the hardware controller which is displayed on a corresponding pair of seven segment LEDs in hexadecimal. The speed can be varied from a reading of 00 to 7F, which is full speed. The direction of the motors on the robot are controlled by DPDT (double pole-double throw) momentary center "off" switches, while the direction of the motor on the conveyor belt is

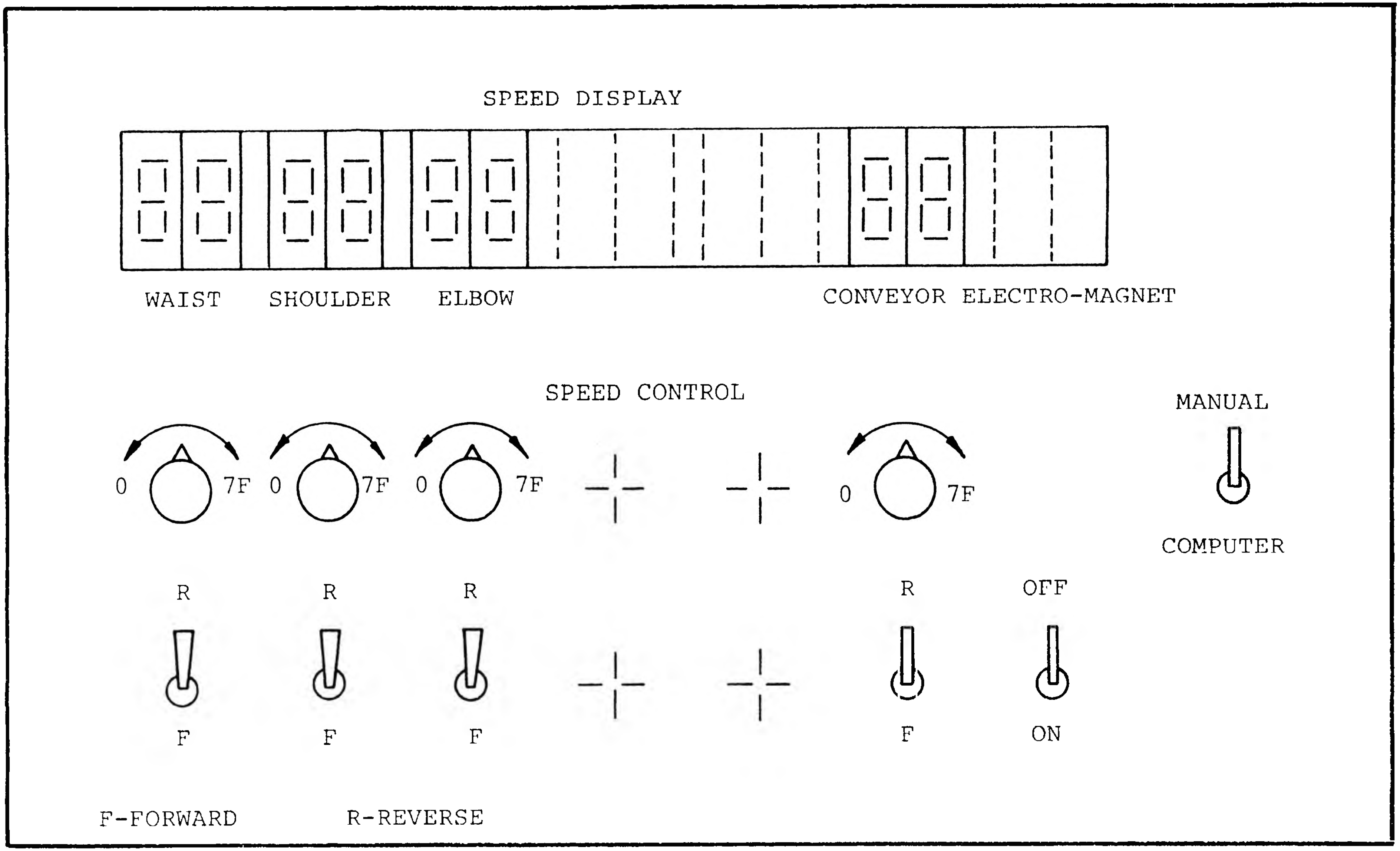


Figure 20. Manual Control

controlled by a SPDT (single pole-double throw) switch since it does not require constant operator supervision. DPDT switches are used for the robot's motor switches because additional control logic is required to turn the robot's motors "off" when the switch is in the center "off" position. The second pole of each of the DPDT switches is ORed (74LS32) with the computer/manual mode control line to enable or disable the PWM signal line (A>B) on the PWM board. When the hardware controller is in manual mode, the PWM signal line is only enabled when the DPDT switch is in either of the "on" positions, which are the forward or reverse positions. When the hardware controller is in computer mode, the PWM signal line is always enabled.

Since the electromagnet has only two states ("on" or "off"), a SPDT switch is used for its control with no display.

The final switch on the manual control is the computer/manual mode switch (SPDT), which supplies the signal for the computer/manual mode control line. The computer/manual mode control line is connected to several components in the hardware controller to provide control logic for the devices on the function boards (as described in their respective sections), and provide the hardware controller's control status to the least significant bit input line on the status byte latch, which resides on the interface board.

The schematic for the manual control is shown in

Appendix I. The inside of the hardware controller with the function boards is shown in Fiqure 21.

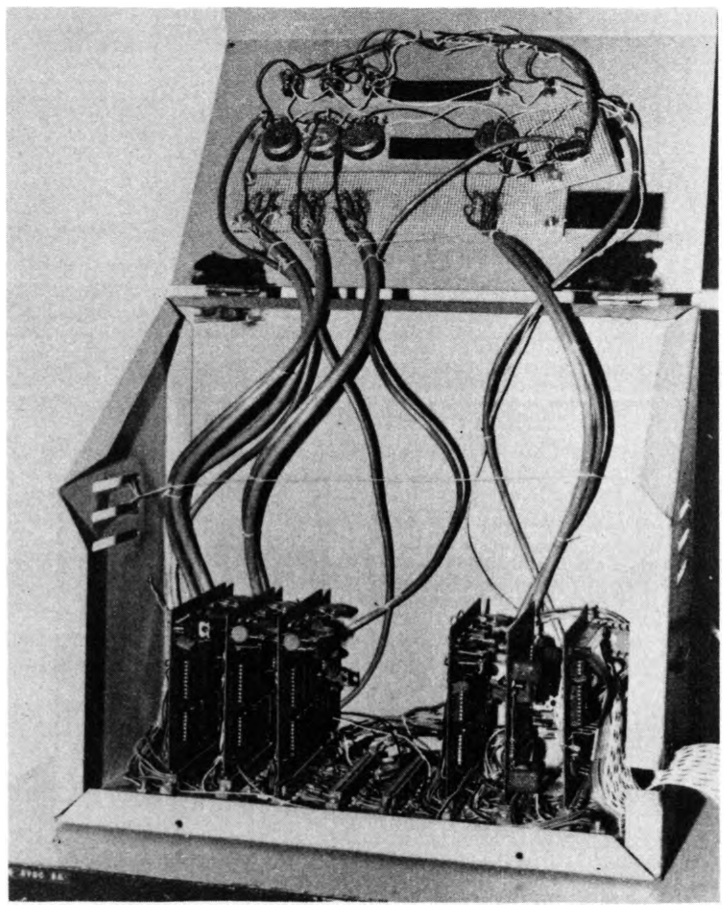


Figure 21. Inside of the Hardware Controller

VI. CONCLUSION

A. SUMMARY

In this thesis the author presents an experimental model robot system with all the necessary hardware components for controlling the system manually or by an Apple microcomputer.

A general purpose model robot system is presented which consists of a model conveyor belt, a three-joint model robot arm, and an optional vision system, all of which can be controlled individually or as a complete interactive system.

An interface board, which provides a communication port between a hardware controller and an Apple computer, is presented with a detailed explanation. The interface board enables the computer to output both control data and signals to the hardware controller and input the feedback information from the model robot system.

A hardware controller, which frees the computer from inefficient actuator control and provides a manual control feature, is presented with a comprehensive analysis of the individual components which comprise it. The hardware controller accepts the control data and signals from the computer and executes them on the model robot system.

To operate the model robot system through the hardware controller by the computer, a robot operating system and control program was developed by Eric Stelzer. Details of the control program and operating system are discussed in

his thesis, "A Control Program and Operating System for a Three Joint Robot Arm on an Apple Microcomputer" (4). Under computer control, the model robot can be directed to move to a particular point designated in Cartesian coordinates (X,Y,Z) or joint angles ($\theta_1,\theta_2,\theta_3$). The move occurrs in real time and is performed with an accuracy comparable to the resolution of the feedback data. Therefore, the large amount of backlash in the robot's gearing is less of a negative influence on the position control than what was initially anticipated. Thus, the resolution of the position potentiometers' A/D is the main limiting influence in the robot's accurate positional control. With the positional control achieved, objects moving down the conveyor belt can be intercepted by the robot and picked-up by the electromagnet without stopping the conveyor belt by allowing for a time delay after the second photodetector beam is broken.

When the model robot system is used in conjunction with the vision system, objects are identified, picked-up, and placed in their designated bins with total continuity. The robot system performs this complex task in real time and with dependable repeatability. Therefore, the criteria of a robot system which can perform a specified function in real time is fulfilled.

The model robot not only demonstrates computer controlled motion but also demonstrates that movements can be performed in several different configurations, as shown

in Figures 22 through 25.

The advantages of pulse-width modulation for velocity control of the robot system's motors is demostrated by the observed wide range of controlled motor velocities and usable minimum motor velocities.

The author believes that the model robot system described in this thesis fulfills all of the requirements of a practical experimental robot system, as summarized at the beginning of Chapter III.

B. IMPROVEMENTS AND FUTURE USES

During the course of developing and testing the model robot system, several areas of improvement and future development were observed. A few areas of improvement involve the removal of some physical limitations, the reduction of digital noise, and the stabilization of power from load variations which affect the motor speeds and the reference voltages.

Physical limitations in the joint ranges and the position control could be reduced and even eliminated in one joint by changes in the feedback loop (potentiometer and corresponding A/D). By using potentiometers without the transition zones discussed in Chapter III (multi-turn potentiometers, for example), the joint range limits could be reduced to the structural limits and even eliminated for the base (waist) joint. If the 8-bit A/D used in converting the analog feedback value to a digital equivalent were

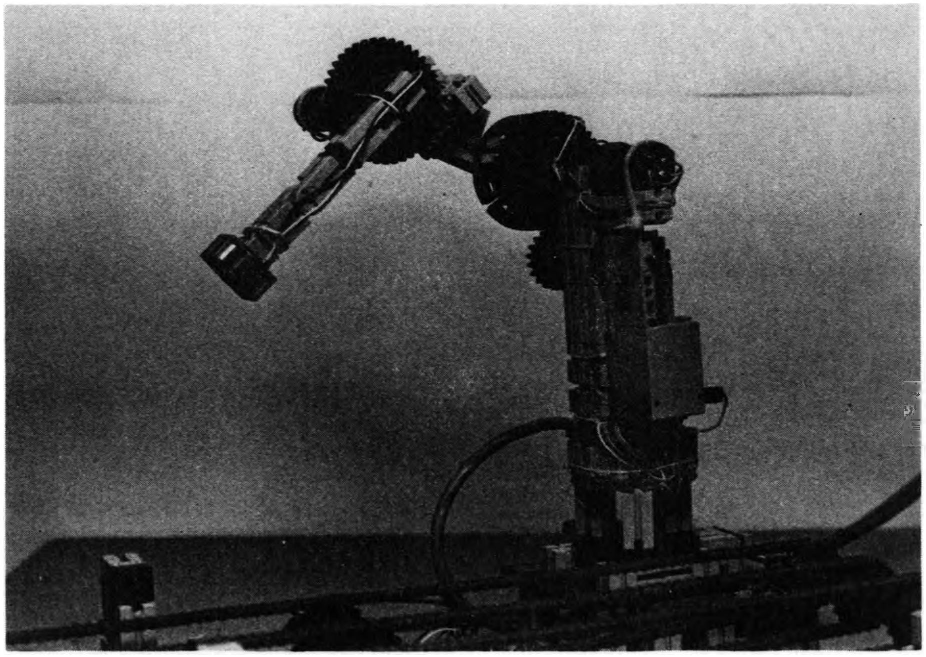


Figure 22. Right-Handed Elbow-Up Configuration

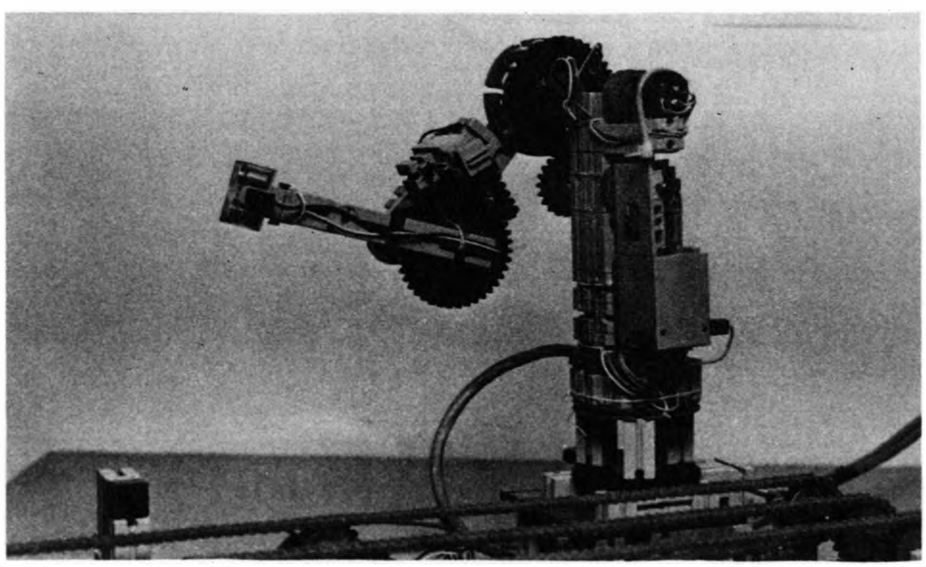


Figure 23. Right-Handed Elbow-Down Configuration

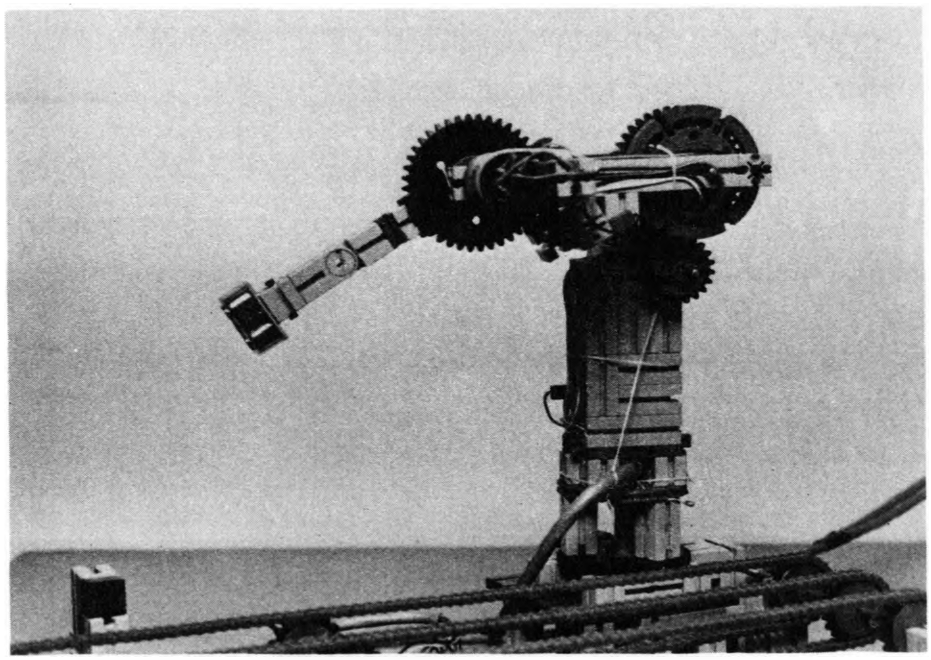


Figure 24. Left-Handed Elbow-Up Configuration

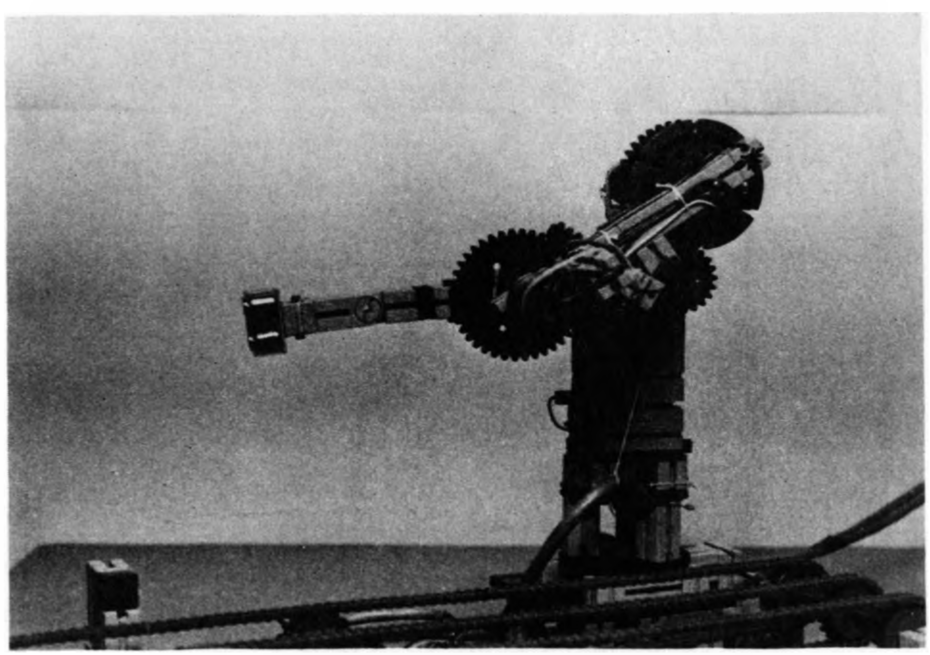


Figure 25. Left-Handed Elbow-Down Configuration

replaced by a 12-bit A/D, the position resolution would improve from 1.4 degrees per bit to .088 degrees per bit. Such an increase in the resolution would place the responsibility for accurate position control mainly on the control software.

One of the main disadvantages of pulse-width modulation (PWM) is the electrical noise generated which can be easily transmitted into low level circuitry. One area of contamination by this electrical noise is the analog lines from the robot's feedback potentiometers. Presently the PWM power lines to the robot system's motors are contained in the same cable as the analog feedback lines. By separating these lines and placing the analog lines in a shielded cable, contamination from electrical noise should be reduced to much lower levels.

Since one power supply presently provides the unregulated power to all the electrical devices, such as the IC (Integrated Circuit) chips and the motors, the output voltage and current of the power supply fluctuates as the load changes. The main cause of the changing load is the continuous turning "on" and "off" of devices and motors in the robot system. The fluctuations in voltage and current affect the performance of both internal and external system components. Although the internal components (inside the hardware controller) are isolated from minor fluctuations by the voltage regulators (MC7805's), major fluctuations could affect the reference voltages and cause erroneous feedback

data from the position A/D. The external components, which include the motors and electromagnet, are the main components affected by power fluctuations. Since they are powered directly by the unregulated power supply, voltage swings cause changing motor speeds and motor torques and cause the electromagnet's field to vary. Although the software compensates very well for these conditions, it would be better to be completely rid of the problem. One possible solution is to provide separate power supplies for the internal and external devices and to use voltage regulators, such as MC7808's, on the power driver circuits to reduce the fluctuations on the power lines to the external devices. This improvement would assist in the elimination of load change sensitivity and thereby provide dependable and consistent power to the robot system.

Future use of the model robot system may include the installment of additional features to enhance the main objective of providing a practical and general medium for experimentation and training.

One possible enhancing feature would be the reconfiguration of the original model robot or the construction of an additional model robot which included at least one prismatic joint as shown in Figure 2. With a prismatic joint, an experimenter or researcher would receive additional experience by the use of the different control algorithms implemented in prismatic joint control.

By replacing some of the DC motor actuators with

stepper motors or by adding the stepper motors as extra devices or joints and constructing pulse-frequency modulator boards for the hardware controller, a control algorithm and a control reponse comparison could be made between the two different electrical actuators and their respective function boards.

Another possible area of enhancement is the addition of or change in feedback sensors. Although in most cases an optical encoder/decoder for a feedback sensor would be too bulky for the model robot joints, the base (waist) joint could support an optical sensor. Such a change would not only remove the joint range limit for the waist joint, since an optical sensor has no transition zone, but would also provide a comparison between the quality and reliability of the feedback data provided by the optical and the potentiometer feedback sensors. Also a comparison between the different processing required for their data could be made. Additional feedback sensors, such as strain gauges on each joint, would provide the controller with torque feedback. Torque feedback could be used to determine proper control values as the motors' loads change with the changing robot configuration and direction, determine if an object to be picked-up exceeds the robot's load limits, or determine if the robot is stalled due to a possible collision.

Since the functions of the hardware controller are separated into distinct function boards, a few changes in the hardware controller's wiring bus and distribution board

would allow each joint of the model robot and each device in the robot system to be controlled by a dedicated microprocessor. Therefore, the model robot system also lends itself to multiprocessor control. An advantage of using multiprocessor control would be the reduction of the Apple computer's workload to allow control schemes of greater complexity to be used in real time.

With the versatility available with this model robot system, the author hopes that continued knowledge and understanding of computer controlled systems and robotics will be gained through its use and application.

BIBLIOGRAPHY

1. Fisher America Inc., 14 Madison Road, Fairfield, New Jersey 07006.
2. Nof, S. Y.; Meier, W. L.; and Deisenroth, M. P.; "Computerized Physical Simulators are Developed to Solve IE Problems," Industrial Engineering, October 1980.
3. Nof, S. Y.; Meier, W. L.; and Deisenroth, M. P.; "Using Physical Simulators to Study Manufacturing Systems Design and Control," Proceedings, 1979.
4. Stelzer, E. H., "A Control Program and Operating System for a Three Joint Robot Arm on an Apple Microcomputer," M.S. Thesis, University of Missouri-Rolla, 1982.
5. Paul, R. P., Robot Manipulators: Mathematics, Programming, and Control, The MIT Press, 1981, pp. 41-54.
6. Shen, K. F., "A Computer Vision System for Industrial Robots," M.S. Thesis, University of Missouri-Rolla, 1983.
7. Apple II Reference Manual, APPLE COMPUTER INC., 1981.
8. Linear Databook, NATIONAL SEMICONDUCTOR, 1980.
9. Murugesan, S., "An Overview of Electric Motors for Space Applications," IEEE Transactions on Industrial Electronics and Control Instrumentation, Vol. IECI-28, No. 4, November 1981, pp. 260-265.

10. <u>DC Motors-Speed Controls-Servo Systems</u>, ELECTRO-CRAFT CORP., 1978.

11. <u>The TTL Data Book</u>, TEXAS INSTRUMENTS INC., 1976.

12. <u>Linear Data Book</u>, NATIONAL SEMICONDUCTOR, 1976.

13. <u>Motorola CMOS Data</u>, MOTOROLA INC., 1978.

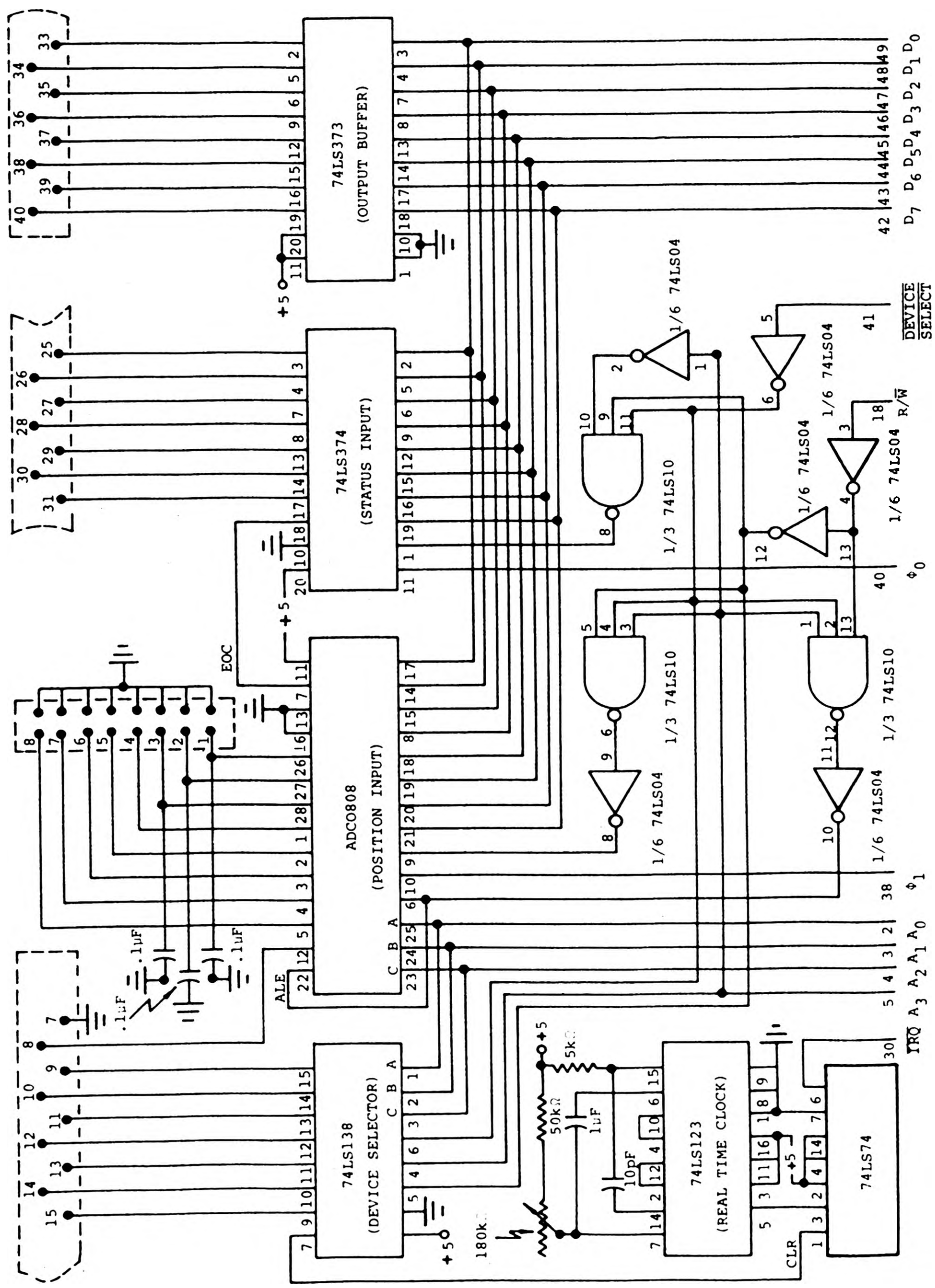
VITA

Richard Eugene Wainwright was born on December 14, 1957 in Monroe, Wisconsin. He received his primary and secondary education in Monroe, Wisconsin. He attended Olivet Nazarene College, in Kankakee, Illinois, where he received his Bachelor of Science Degree Cum Laude in Engineering Physics in May 1980. Since that time he has been pursuing his graduate studies in Electrical Engineering at the University of Missouri-Rolla.

APPENDICES

APPENDIX A

INTERFACE BOARD SCHEMATIC



APPENDIX B

DB-25P PIN ASSIGNMENT

FOR MODEL ROBOT SYSTEM

Pin #	Function	Pin #	Function
1	Shoulder Potentiometer Wiper	14	Shoulder Potentiometer (+) Terminal
2	Elbow Potentiometer GND Terminal	15	Shoulder Potentiometer GND Terminal
3	Elbow Potentiometer Wiper	16	Shoulder Motor
4	Elbow Potentiometer (+) Terminal	17	Shoulder Motor
5	Electromagnet	18	Elbow Motor
6	Electromagnet	19	Elbow Motor
7	Waist Potentiometer GND Terminal	20	Waist Motor
8	Waist Potentiometer Wiper	21	Waist Motor
9	Waist Potentiometer (+) Terminal	22	Second Photodetector
10	First LED (+) Terminal	23	First Photodetector
11	GND for the LED's and the Photodetectors	24	Conveyor Motor
12	Second LED (+) Terminal	25	Conveyor Motor
13	No Connection		

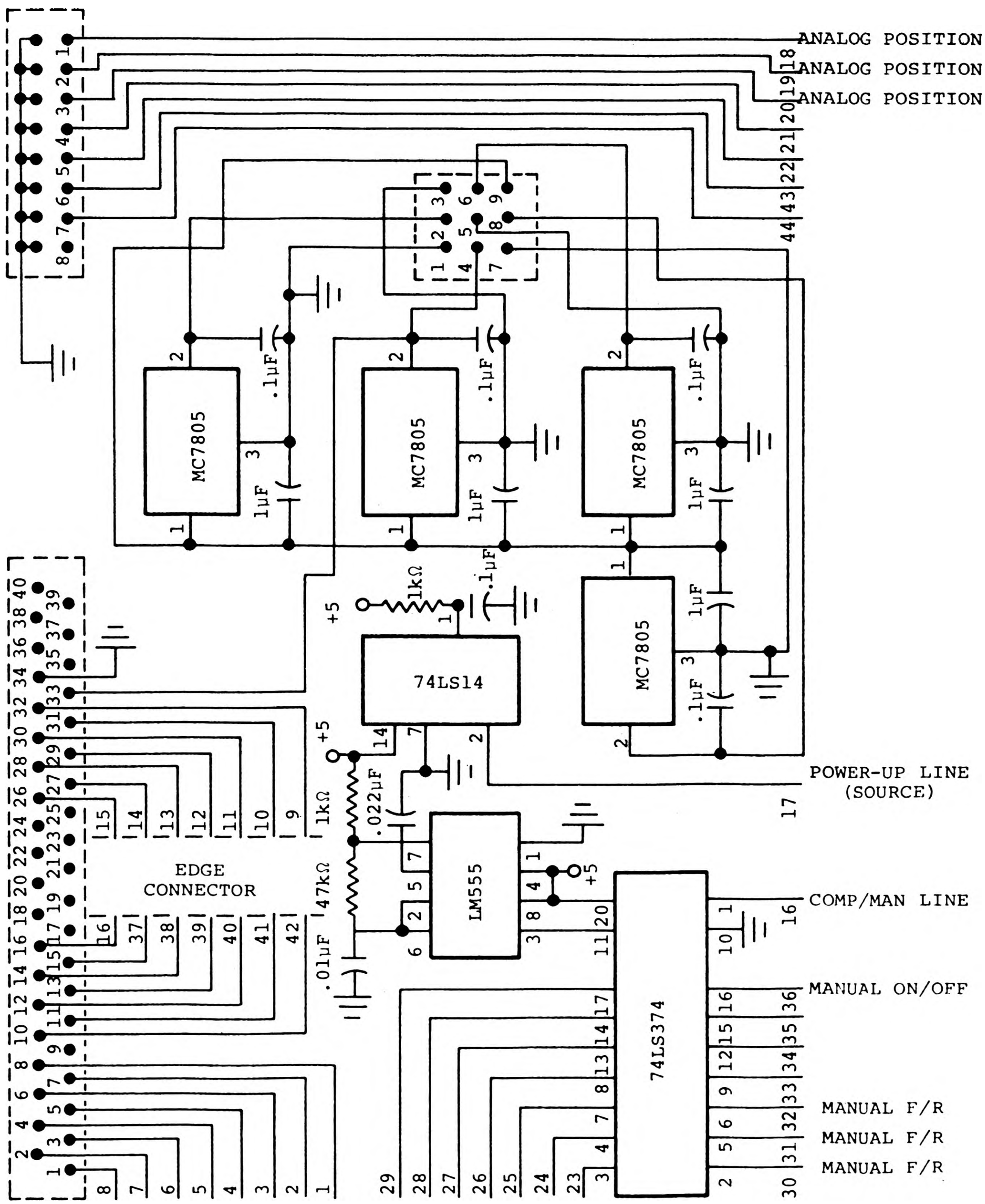
APPENDIX C

WIRING ASSIGNMENT FOR THE DISTRIBUTION BOARD'S EDGE CONNECTOR

Pin #	Function	Pin #	Function
1	Speed Bit #1 (LSB)	23	Manual F/R Input #1
2	Speed Bit #2	24	Manual F/R Input #2
3	Speed Bit #3	25	Manual F/R Input #3
4	Speed Bit #4	26	Manual F/R Input #4
5	Speed Bit #5	27	Manual F/R Input #5
6	Speed Bit #6	28	Manual F/R Input #6
7	Speed Bit #7 (MSB)	29	Manual ON/OFF Input
8	F/R or ON/OFF Bit	30	Manual F/R Output #1
9	Device #1 Selector Line	31	Manual F/R Output #2
10	Device #2 Selector Line	32	Manual F/R Output #3
11	Device #3 Selector Line	33	Manual F/R Output #4
12	Device #4 Selector Line	34	Manual F/R Output #5
13	Device #5 Selector Line	35	Manual F/R Output #6
14	Device #6 Selector Line	36	Manual ON/OFF Output
15	Device #7 Selector Line	37	Status Input #2
16	Computer/Manual Mode Line	38	Status Input #3
17	Power-up Trigger Line	39	Status Input #4
18	ADC0808 In #1 (Waist)	40	Status Input #5
19	ADC0808 In #2 (Shoulder)	41	Status Input #6
20	ADC0808 In #3 (Elbow)	42	Status Input #7
21	ADC0808 In #4	43	ADC0808 In #6
22	ADC0808 In #5	44	ADC0808 In #7

APPENDIX D

DISTRIBUTION BOARD SCHEMATIC



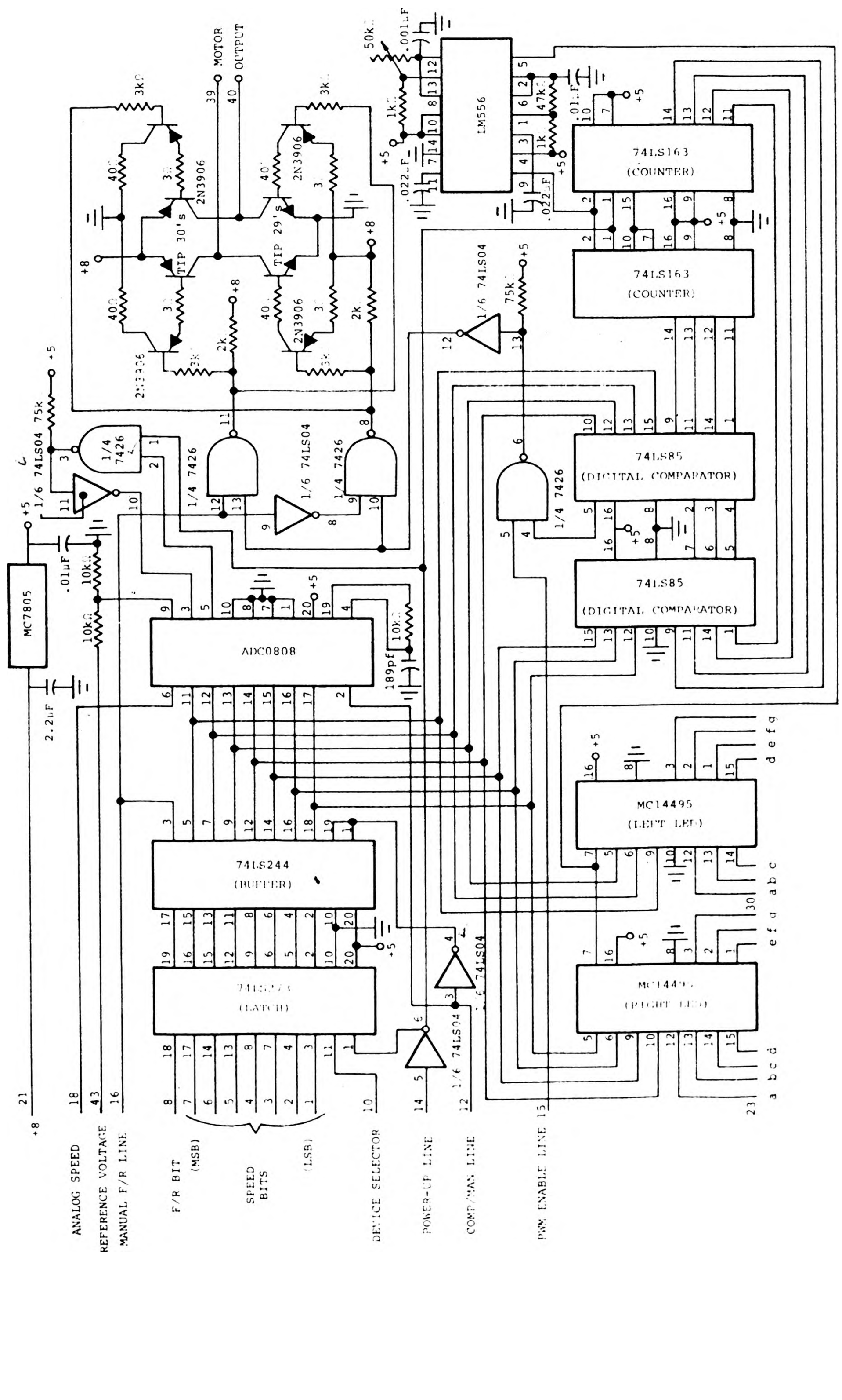
APPENDIX E

WIRING ASSIGNMENT FOR THE PULSE-WIDTH MODULATOR BOARDS' EDGE CONNECTOR

Pin #	Function	Pin #	Function
1	Speed Bit #1 (LSB)	23	Segment a (Right LED)
2	Speed Bit #2	24	Segment b (Right LED)
3	Speed Bit #3	25	Segment c (Right LED)
4	Speed Bit #4	26	Segment d (Right LED)
5	Speed Bit #5	27	Segment e (Right LED)
6	Speed Bit #6	28	Segment f (Right LED)
7	Speed Bit #7 (MSB)	29	Segment g (Right LED)
8	Forward/Reverse Bit	30	Segment a (Left LED)
9	No Connection	31	Segment b (Left LED)
10	Device Selector Line	32	Segment c (Left LED)
11	No Connection	33	Segment d (Left LED)
12	Computer/Manual Mode Line	34	Segment e (Left LED)
13	No Connection	35	Segment f (Left LED)
14	Power-up Trigger Line	36	Segment g (Left LED)
15	PWM Signal Enable Line	37	No Connection
16	Manual F/R Line	38	No Connection
17	No Connection	39	Motor Output (+)
18	Speed Potentiometer Input	40	Motor Output (-)
19	No Connection	41	No Connection
20	No Connection	42	+5 Volts (LED Drivers)
21	+8 Volts (Unregulated)	43	Reference Voltage (A/D)
22	Ground	44	Ground

APPENDIX F

PULSE-WIDTH MODULATOR BOARD SCHEMATIC



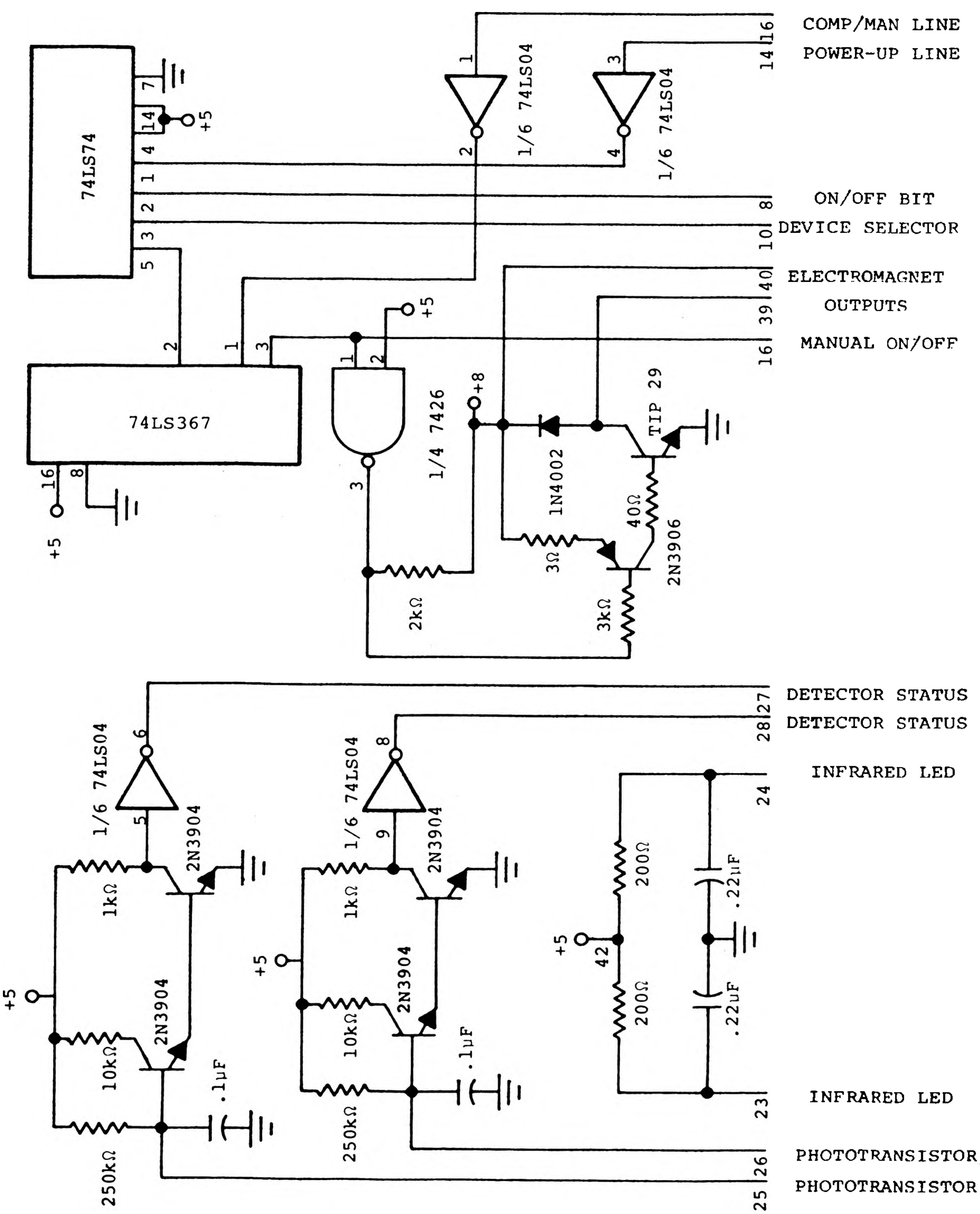
APPENDIX G

WIRING ASSIGNMENT FOR THE ELECTROMAGNET AND PHOTODETECTOR BOARD'S EDGE CONNECTOR

Pin #	Function	Pin #	Function
1	No Connection	23	First LED (+) Terminal
2	No Connection	24	Second LED (+) Terminal
3	No Connection	25	First Photodetector
4	No Connection	26	Second Photodetector
5	No Connection	27	First Detector Status
6	No Connection	28	Second Detector Status
7	No Connection	29	No Connection
8	ON/OFF Bit	30	No Connection
9	No Connection	31	No Connection
10	Device Selector Line	32	No Connection
11	No Connection	33	No Connection
12	Computer/Manual Mode Line	34	No Connection
13	No Connection	35	No Connection
14	Power-up Trigger Line	36	No Connection
15	No Connection	37	No Connection
16	Manual ON/OFF Line	38	No Connection
17	No Connection	39	Electromagnet Output
18	No Connection	40	Electromagnet Output
19	No Connection	41	No Connection
20	No Connection	42	+5 Volts for LED's
21	+8 Volts (Unregulated)	43	No Connection
22	Ground	44	Ground

APPENDIX H

ELECTROMAGNET AND PHOTODETECTOR BOARD SCHEMATIC



APPENDIX I

MANUAL CONTROL SCHEMATIC