A MICROPROCESSOR-BASED PROSTHESIS CONTROLLER FOR USE DURING EARLY WALKING TRAINING OF ABOVE-KNEE AMPUTEES

by

Mark L. Tanquary

S.B., Massachusetts Institute of Technology
(1976)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June, 1978

Signature of Author..
Department of Mechanical Engineering, June 27, 1978

Certified by..
Thesis Supervisor

and..
Thesis Supervisor

Accepted by..
Chairman, Department Committee on Graduate Students

A MICROPROCESSOR-BASED PROSTHESIS CONTROLLER FOR USE DURING EARLY WALKING TRAINING OF ABOVE-KNEE AMPUTEES

by

Mark L. Tanquary

Submitted to the Department of Mechanical Engineering on June 27, 1978 in partial fulfillment of the requirements for the degree of Master of Science.

ABSTRACT

Recent work in the field of above-knee prosthetics at M.I.T. has resulted in the development of a controllable, passive prosthesis. The prosthesis employs a magnetic particle brake to provide a resistive torque about the knee axis to damp the prosthesis motion. A portable controller has been designed and constructed for use with this prosthesis to assist new amputees during early walking training. A microprocessor with a stored program was selected as a basis for the controller. This enables modification of the controller's function by simply changing the control program. The present control program is designed to allow a recent amputee to make a gradual and comfortable transition from a rigid to an articulated prosthesis. Damping profiles are used as input to the controller to provide the desired prosthesis characteristics.

Preliminary testing with an experienced amputee has shown that the system is capable of a wide range of operating characteristics. Plans have been made to further evaluate the system in a hospital environment.

Thesis Supervisor: Woodie C. Flowers
Title: Associate Professor of Mechanical Engineering

Thesis Supervisor: Derek Rowell
Title: Assistant Professor of Mechanical Engineering

ACKNOWLEDGEMENTS

Many people have provided their assistance during the scope of this project. I would like to thank all of them for their part in this work. Among those that have been especially helpful are:

Woodie Flowers and Derek Rowell, for their creative and witty help and guidance in their respective fields of expertise.

The staff and students of the M.I.T. Sensory Aids Development and Evaluation Center, and in particular George Dalrymple, for exercising extreme patience in dealing with me.

Don Darling and Don Grimes, for offering their knowledge and insight into past and present research in the field of above-knee prosthetics.

Dr. Joseph S. Barr, Jr. and Betty Kozlowski, a physical therapist at the Massachusetts General Hospital, for answering what seemed at times, an unlimited number of questions.

Steve Cornell, for participating in the initial system evaluation and helping to discover the system bugs.

And finally, my wife, Carol, whose patience, understanding and humor have enabled me to complete this project while maintaining a healthy outlook on life.

This work was supported by the Rehabilitation Services Agency, under HEW Grant Number 23-P-55854/1.

TABLE OF CONTENTS

Page

LIST OF FIGURES

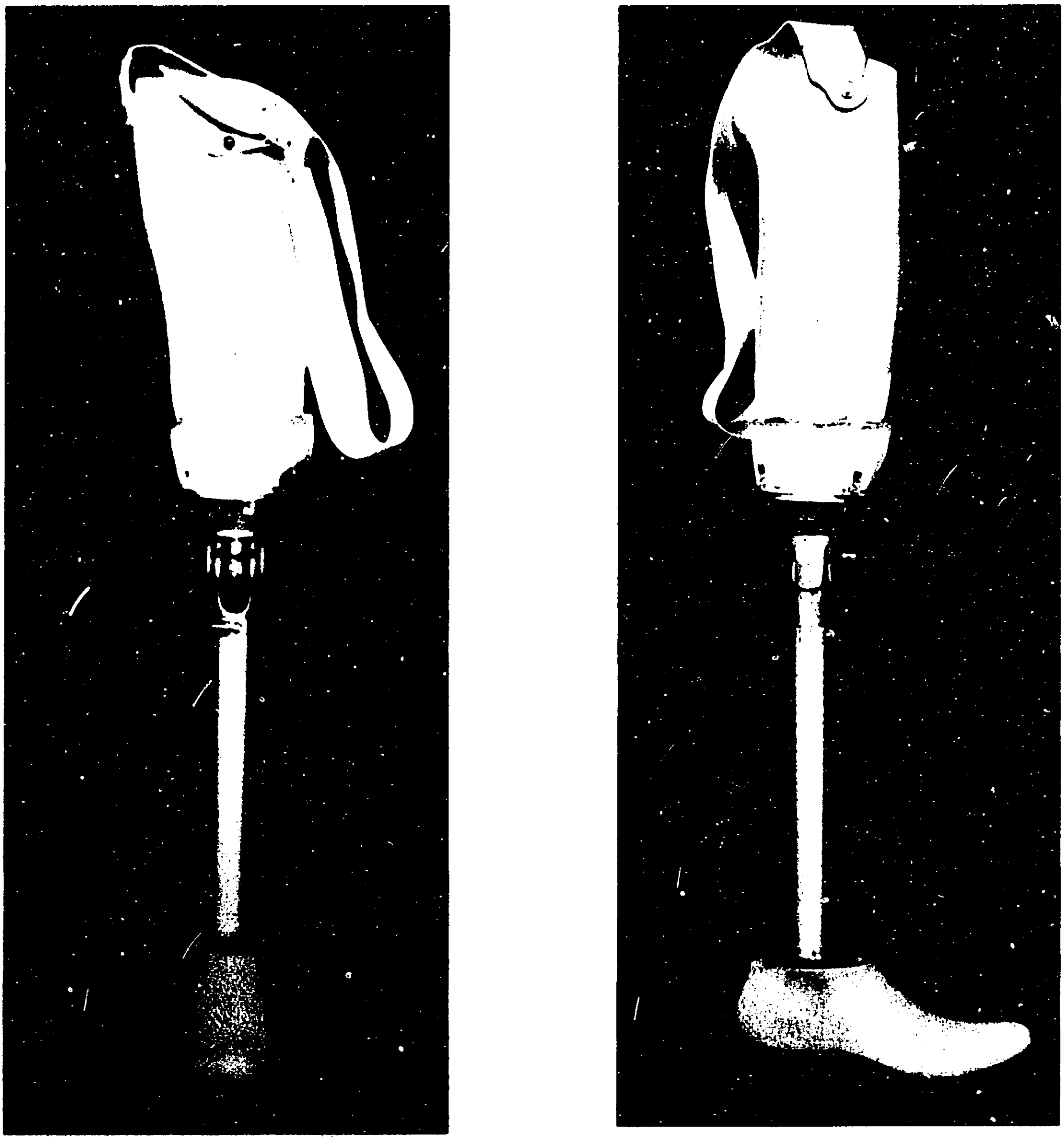
LIST OF FIGURES (con't)

CHAPTER I INTRODUCTION

The loss of a lower limb is a severe experience for any individual from both a physical and emotional standpoint. In the case of an above-knee (A/K) amputee, the ramifications are particularly great. In addition to dealing with an altered body image, the amputee is faced with the problem of reduced efficiency of mobility. The amputee's rehabilitation program plays an important role in helping him to deal with these and other problems.

The ultimate goal of the rehabilitation process is to enable the amputee to return to his former role in society. During this process, the amputee participates in exercises designed to aid his recovery from the effects of the amputation and to strengthen his remaining musculature to serve new functions during ambulation.

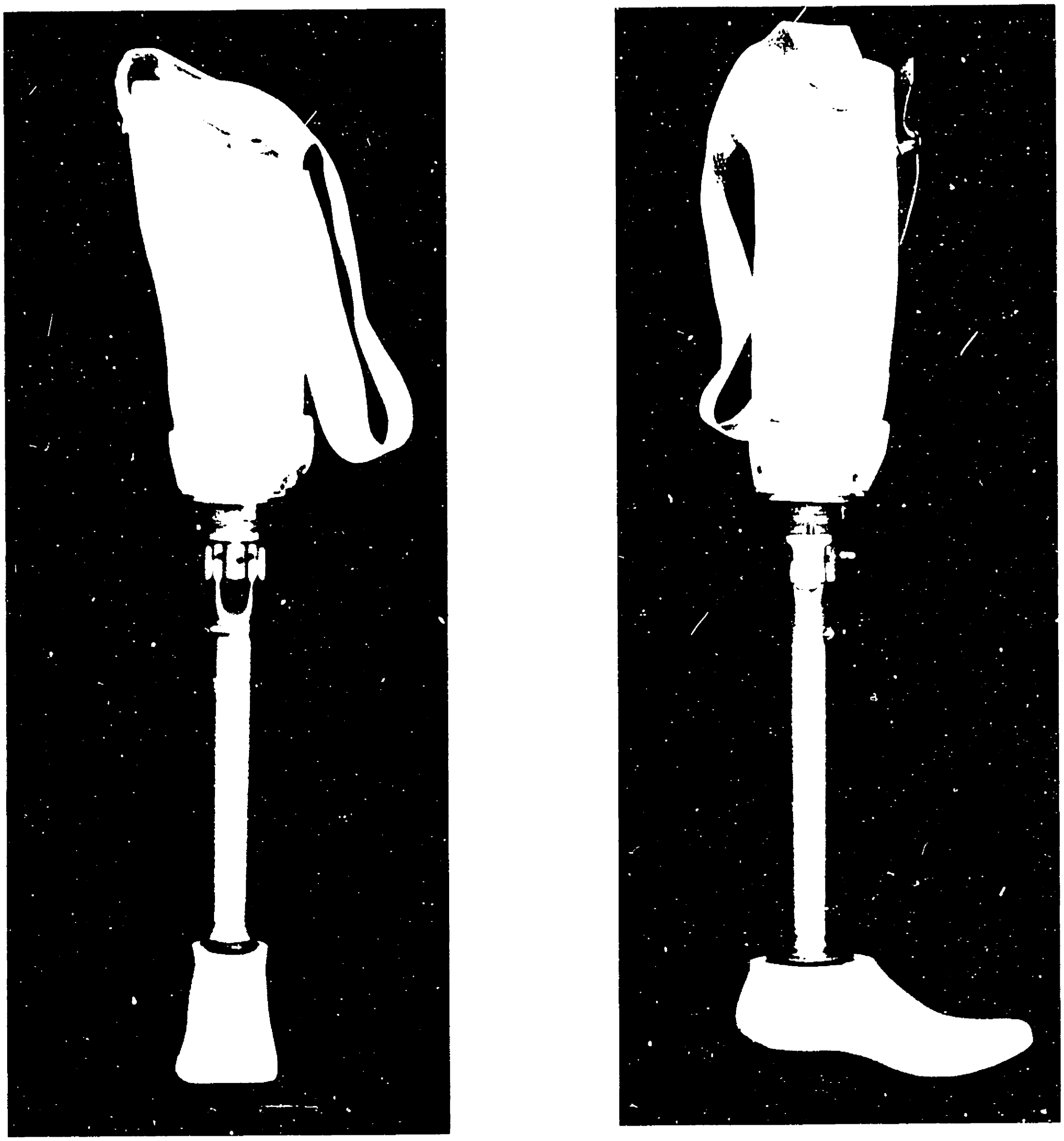
One of the devices that has shown to be of extreme usefulness during rehabilitation is the temporary prosthesis (see Figure 1-1).



Front View

Side View

Figure 1-1 Temporary Prosthesis



Front View

Side View

Figure 1-1 Temporary Prosthesis

It can serve to speed up the healing process of the stump and to shorten the time required for rehabilitation (18). The use of a temporary prosthesis is of increased importance when the amputation was preformed because of vascular disease. It helps to reduce the problems associated with immobility and stimulates circulation (9), (15).

A temporary prosthesis usually consists of a plastic socket made to fit the amputee, a knee unit and a combination ankle and foot. The knee unit commonly used is a single axis device which can be manually locked. It is designed to be reusable and inexpensive. A prosthesis of this form is typically employed for early ambulation training and is retained for use by the amputee until fitted with his permanent or definitive prosthesis.

1.1 The Problem

During early walking training with the temporary prosthesis, one of the most crucial phases is the transition from a rigid to an articulated prosthesis. This transition is important because it enables the amputee to walk in a more energy efficient fashion with a closer resemblance to normal walking motions. Currently, this transition is accomplished by simply unlocking the knee joint and allowing the prosthesis to operate in a pendulum or free-swinging mode. Because of the instability associated with this transition, many elderly amputees never progress to the use of an articulated prosthesis. In other cases, the amputee's first experience with an articulated prosthesis may occur when he receives his permanent prosthesis.

1.2 Approach

The goal of this project was to develop a means of allowing an amputee to make a gradual and confident transition from a rigid to an articulated prosthesis. The approach taken follows along the line of previous research at M.I.T. in the field of above knee prosthetics (3), (4), (5), (6), (8).

Rather than attempt to build a specific prosthesis to perform a desired training function, a versatile prosthesis with a suitable controller was developed. The use of a separate controller provides a means to implement many possible operating characteristics. This enables investigation of many functions without a commitment to a specific hardware design. However, unlike previous work, the prosthesis and controller were designed for portable use outside of the M.I.T. Human Mobility Laboratory. The system shown in Figure 1-2 was developed with the goal of assisting new amputees during the early training process.

This system should enable a smooth transition to an articulated prosthesis during the training process. It should also be of benefit to the prescription process for the amputee's permanent leg. The system presents the opportunity for the amputee to evaluate many knee control schemes before receiving a permanent prosthesis.

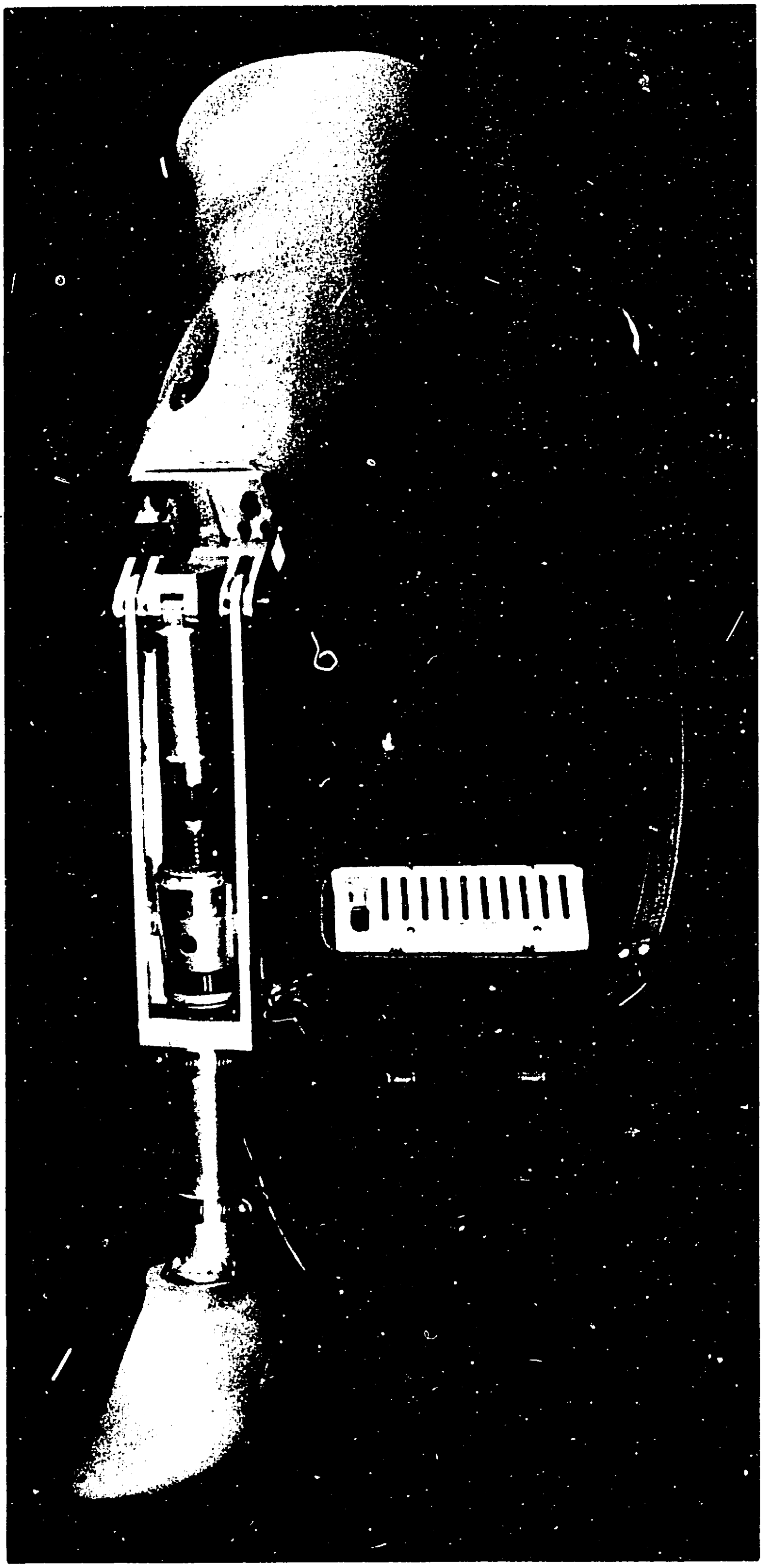


Figure 1-2 Prosthesis-Controller System

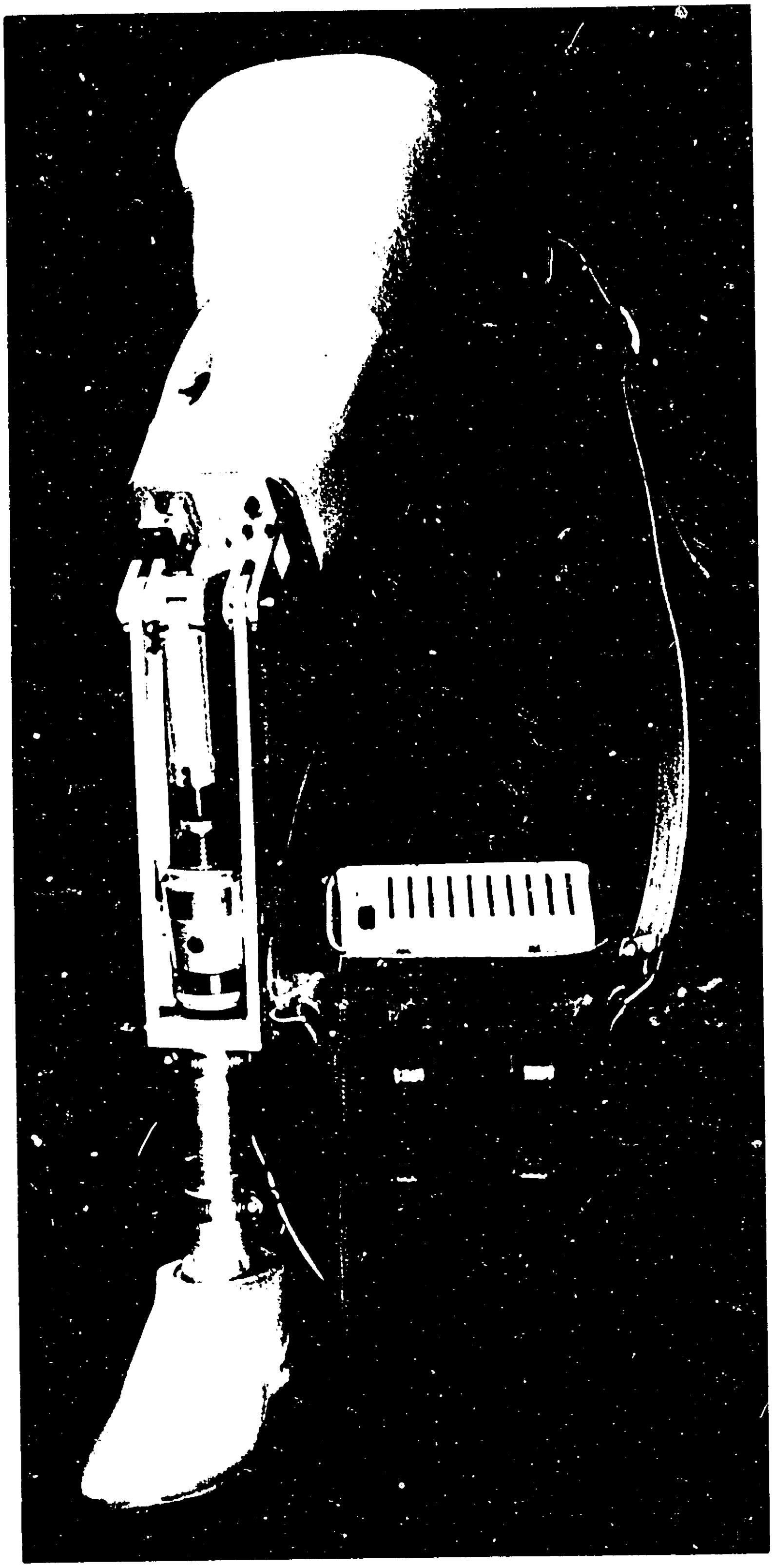


Figure 1-2 Prosthesis-Controller System

1.3 Purpose and Scope of this Thesis

This thesis describes the development and initial evaluation of the prosthesis controller. Chapter II provides a brief introduction to terminology used in describing the walking cycle and an overview of commercially available A/K prosthesis control units. This chapter also includes a summary of the training program for new amputees. The function of the controller and the prosthesis used for its evaluation are described in Chapter III. Chapter IV discusses the microprocessor system employed in the controller and the programs developed for its use. An initial evaluation of the system appears in Chapter V.

CHAPTER II BACKGROUND

2.1 The Walking Cycle

Walking is a cylical process we learn at an early age to use as a method of locomotion. Its complexity can be easily understated. Individuals can alter the basic walking cycle for use over various terrains and conditions at a wide range of speeds. For many individuals this process is also used as an expression of personality and style.

For purposes of discussion and analysis, the basic walking or gait cycle has often been conceptually divided into two phases defined by distinct events during the cycle. For a normal individual, the motion the legs undergo is symmetric and is 180 degrees out of phase relative to one another. Therefore, the cycle of only one leg need be considered. Events during the cycle are usually expressed in terms of the percentage of the total cycle of one leg (see Figure 2-1). Stance phase is defined as the time the leg is in contact with the ground.

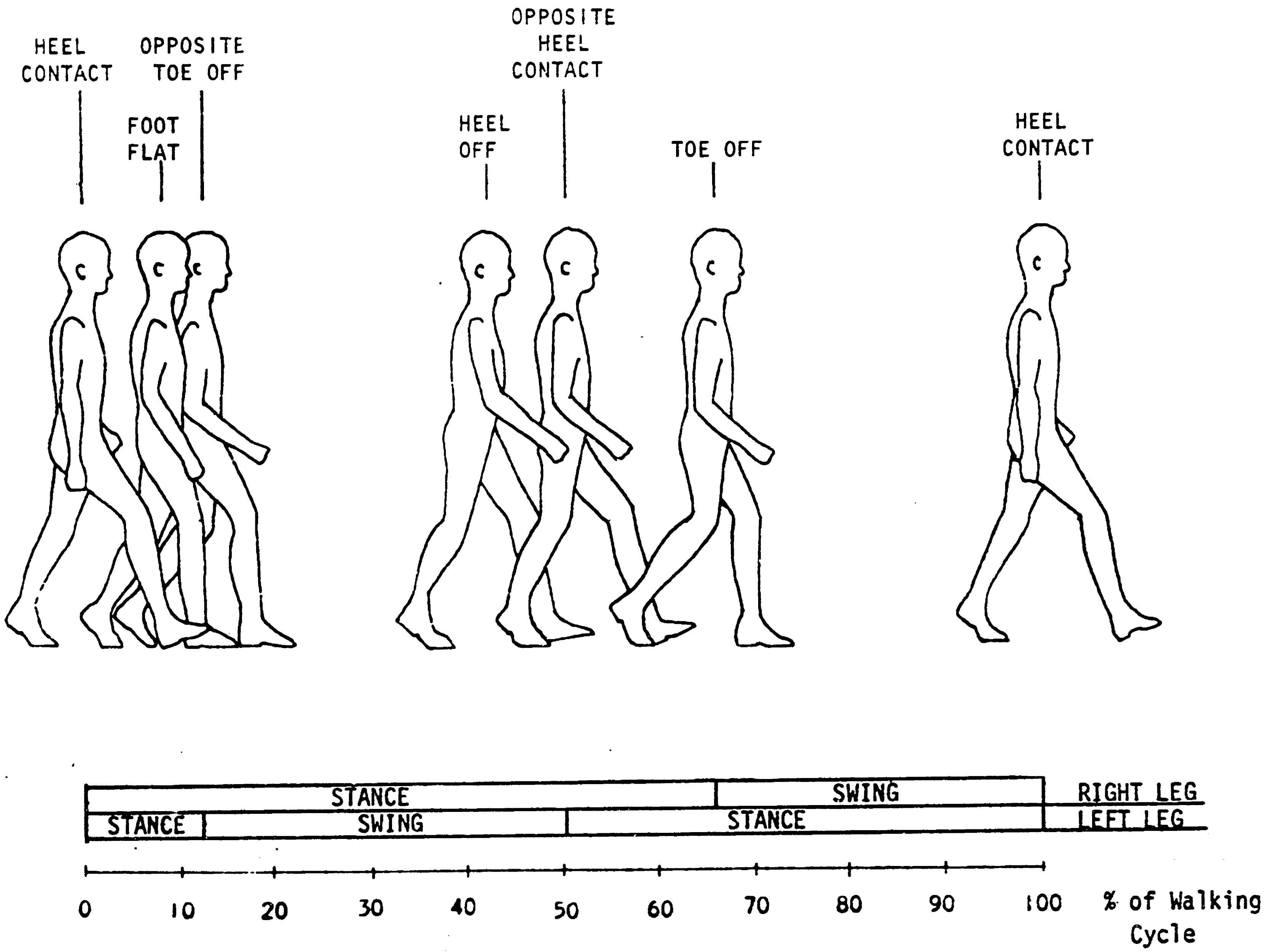


Figure 2-1 The Walking Cycle (from Donath (4))

This comprises approximately 60 percent of the walking cycle. The remaining 40 percent consists of swing phase, when the leg follows through in a forward motion.

Several events have been chosen to define subdivisions within the phases. Starting arbitrarily with heel contact of the right leg as in Figure 2-1, the leg is fully extended. Heel contact signifies the beginning of stance phase. As weight is transferred from the other foot, the ankle flexes until full foot contact is achieved. During this transfer, the knee flexes slightly, then extends to minimize vertical movement of the body's center of gravity. From this point the knee quickly begins to flex and the heel leaves the ground as the hip and ankle provide power to move the body forward. As the knee continues to flex, the right toe leaves the ground and swing phase begins.

During swing phase, the knee flexes until maximum heel rise or flexion is reached. From this point, extension of the hip and gravitational forces act to reverse the velocity of the leg as it extends forward to another heel contact.

During most of the walking cycle the knee acts as an energy absorbing or dissipating device (2). Some positive power is used during stance phase to maintain the height of the body's center of gravity and to initiate flexion into swing phase. The remainder of the required power for the cycle is supplied by torques about the hip and ankle.

2.2 The Above Knee Prosthesis

The goal of an above knee prosthesis is to serve the same functions as the normal leg it replaces. Considerable research and effort have been directed toward the development of the "ideal" prosthesis, but only limited success has been achieved. This is primarily because a passive device has been used to replace both the knee and ankle functions.

A conventional prosthesis, illustrated in Figure 2-2, consists of a basic structure capable of supporting the amputee's weight, an interface to his stump and units to replace the function of the ankle and knee. A quadrilateral socket is usually employed to attach the prosthesis to the stump. Total contact suction and/or a pelvic belt are used as a method of suspension depending on the amputee's preference and stump length.

The majority of above knee amputees use a SACH (Solid Ankle Cushioned Heel) foot and ankle unit as shown in Figure 2-2. It cushions the impact upon heel contact and provides a resistance to dorsiflexion near the end of stance phase. It is a simple design and is preferred by younger, more active amputees. A single axis foot may be used for older amputees who desire greater stability.

There are a number of control units available to help the amputee control the prosthesis during stance and swing phase. Following the guidelines established by the Veteran's Administration (17), control units may be classified into one of three categories:

1. Free swinging
2. Swing phase control
3. Swing and stance phase control

Most of these units have been optimized for normal level walking.

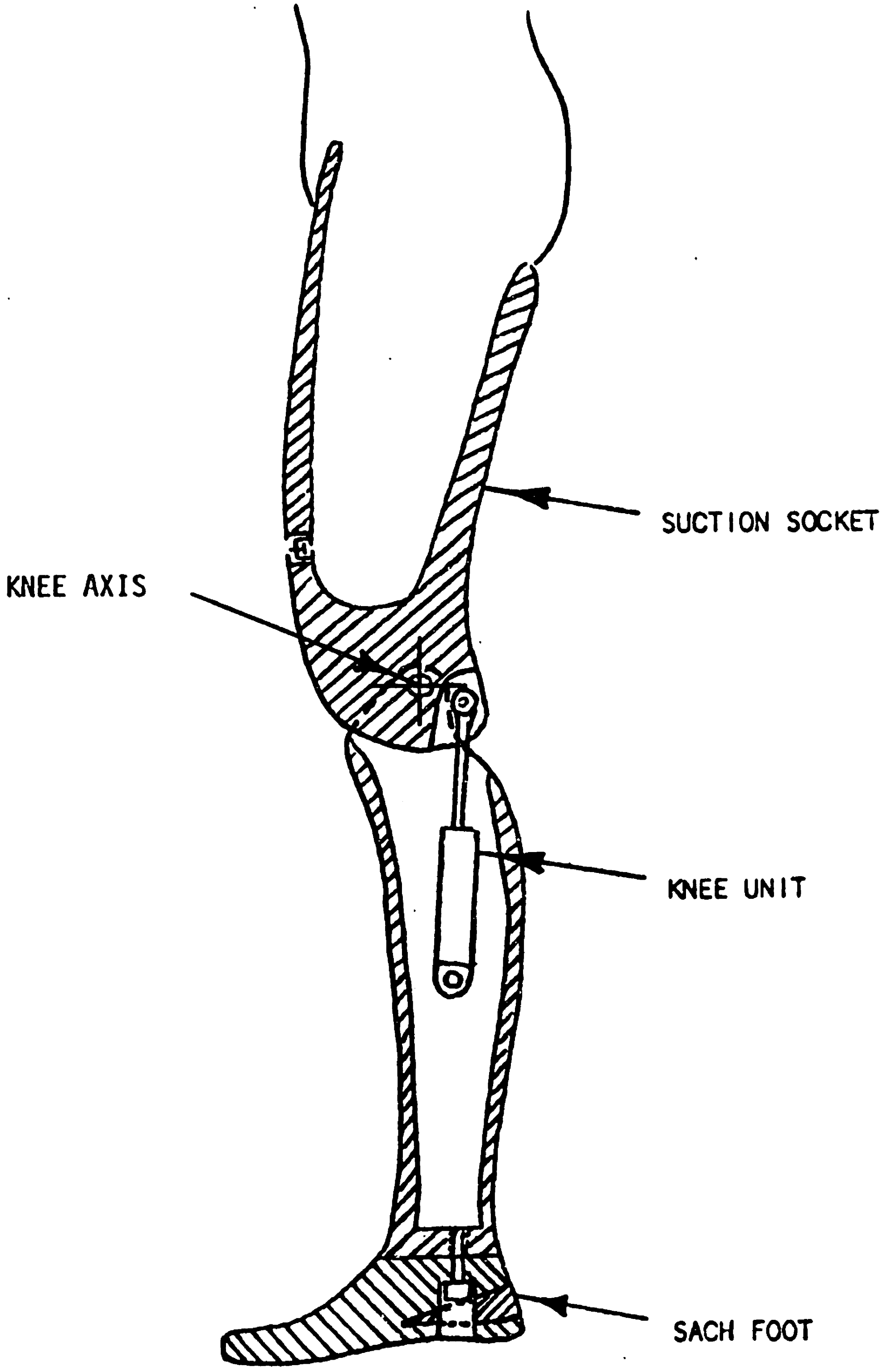


Figure 2-2 A Typical A/K Prosthesis
(From Donath (4))

A free swinging prosthesis provides no resistance to motion except the residual friction present in the knee joint. A prosthesis of this type behaves like a pendulum. Stability during stance phase is provided by a means of a hyperextension stop, as illustrated in Figure 2-3. By positioning the load line ahead of the axis, a moment exists which keeps the prosthesis from buckling during stance.

Swing phase control units provide a resistive torque any time there is relative motion about the knee axis. The term "swing phase" is somewhat of a misnomer because these units also offer resistance during the part of stance phase from heel off to toe off. However, its primary function is to damp motion during swing phase. The damping may be provided by rubbing surfaces or pneumatic or hydraulic dashpots. Rubbing friction can be used to make the resistance constant or proportional to the angular position. Turbulent fluid damping creates a resistance proportional to the angular velocity squared so the prosthesis is responsive to changes in walking speed. A hyperextension stop is usually provided for stance phase control.

The most complex units furnish both swing and stance control. Increased resistance during weight bearing or a polycentric linkage, which changes the instantaneous center of rotation, are used to create enhanced stability during stance. Swing phase control is usually provided by one of the methods described earlier.

In addition to the function of damping the motion of the prosthesis, many control units are fitted with an extension bias spring. This spring acts to store energy during swing phase as the leg approaches

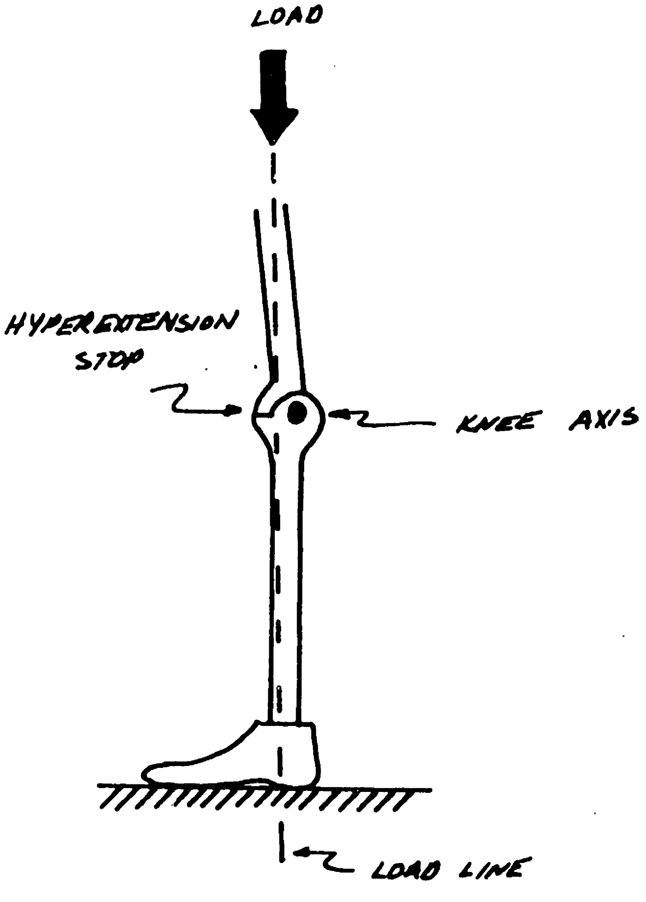


Figure 2-3 Hyperextension Stop

maximum heel rise. This energy is then released so the prosthesis will extend quickly.

Selection of a particular device is based on many factors. The amputee's age, level of activity and stump length are considered as well as the control unit characteristics such as reliability, weight and simplicity. In addition to the components chosen, the prosthesis alignment can also be used to affect its stability. It is possible to trade off stability during stance against the ease of initiating buckling for flexion by adjusting the load line relative to the knee axis. For a more complete description of prosthetic devices and the prescription process, see references (7), (17) and (19).

Despite the number of alternatives available for the construction of A/K prostheses, the final products have tended to fall short of expectations. An above knee amputee requires 20 to 60 percent more energy for level walking than his normal counterpart (1), (16). Also, an amputee's gait is markedly different from that of a normal person. The stance control schemes employed requires the amputee to "vault" over his prosthesis during stance to help keep his center of gravity over the prosthesis.

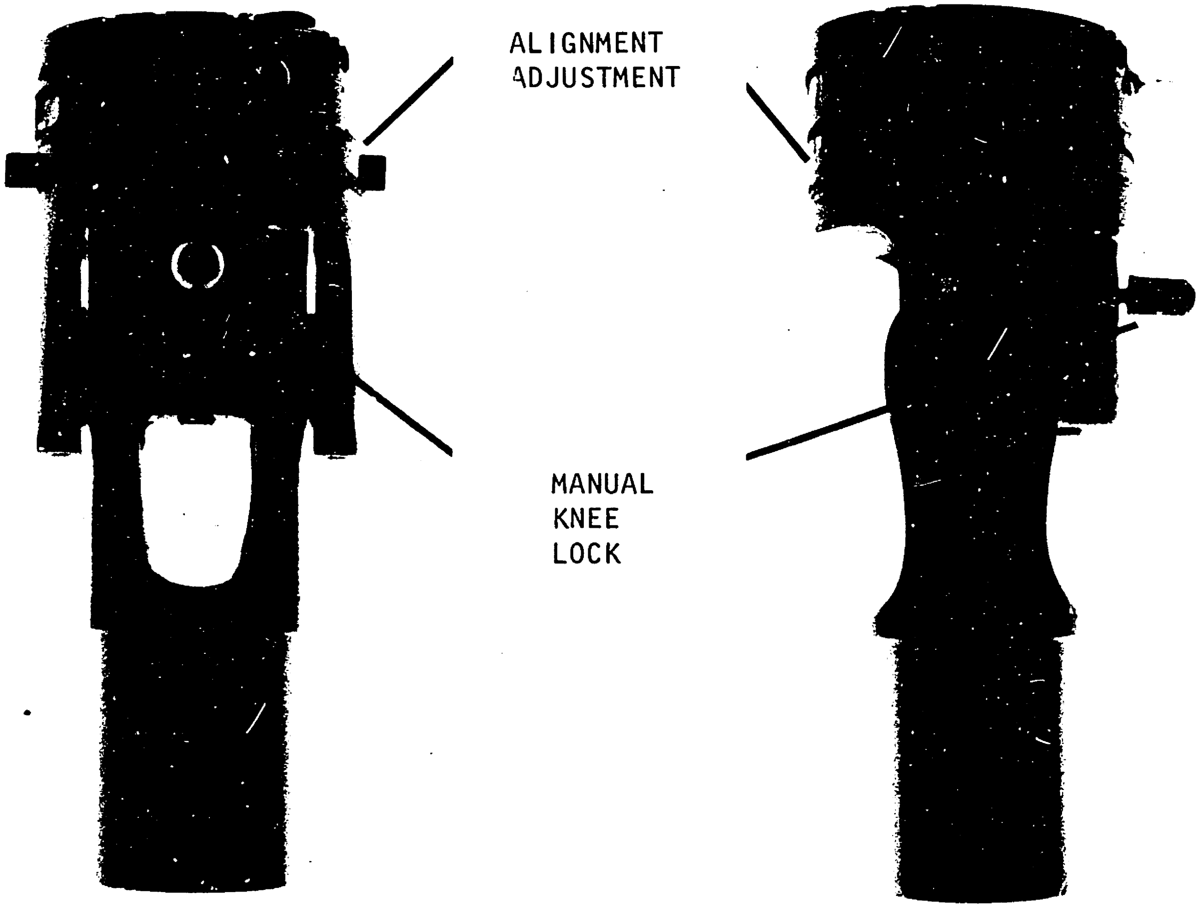
2.3 Early Walking Training of Above Knee Amputees

The use of a temporary prosthesis for early training of amputees has been shown to be beneficial from several standpoints. The temporary socket helps to shrink the stump and promote healing. Psychologically, it reassures the patient that he will be able to regain much of the independence and mobility he had prior to his amputation. By observing the patient's use of his temporary prosthesis, insight can be gained into his needs for prescription of his permanent prosthesis.

Because of its transitory nature, a temporary prosthesis is usually constructed of relatively inexpensive and reusable parts. A typical temporary prosthesis is shown in Figure 1-1. The knee unit employed is usually a free swinging device with a padded hyperextension stop for stability during stance. A manual lock is provided so the prosthesis may be used in a rigid or "peg leg" fashion. A typical temporary knee unit made by U.S. Manufacturing is pictured in Figure 2-4. This model can be preset to lock the prosthesis at full extension when the wearer rises from a sitting position.

The patient is measured for a temporary prosthesis as soon as his physician and the prosthetist feel that the initial swelling of his stump has subsided. Usually the patient will begin training with his prosthesis within two to three weeks of his amputation, depending on the patient's condition. During the first few days of physical therapy, emphasis is placed on applying weight to the prosthesis. Therapy consists of weight shifting exercises and moving around within parallel bars. The third to fifth day is the beginning of gait training. The amputee practices walking with the knee locked using parallel bars or in some cases with the aid of crutches or a walker.

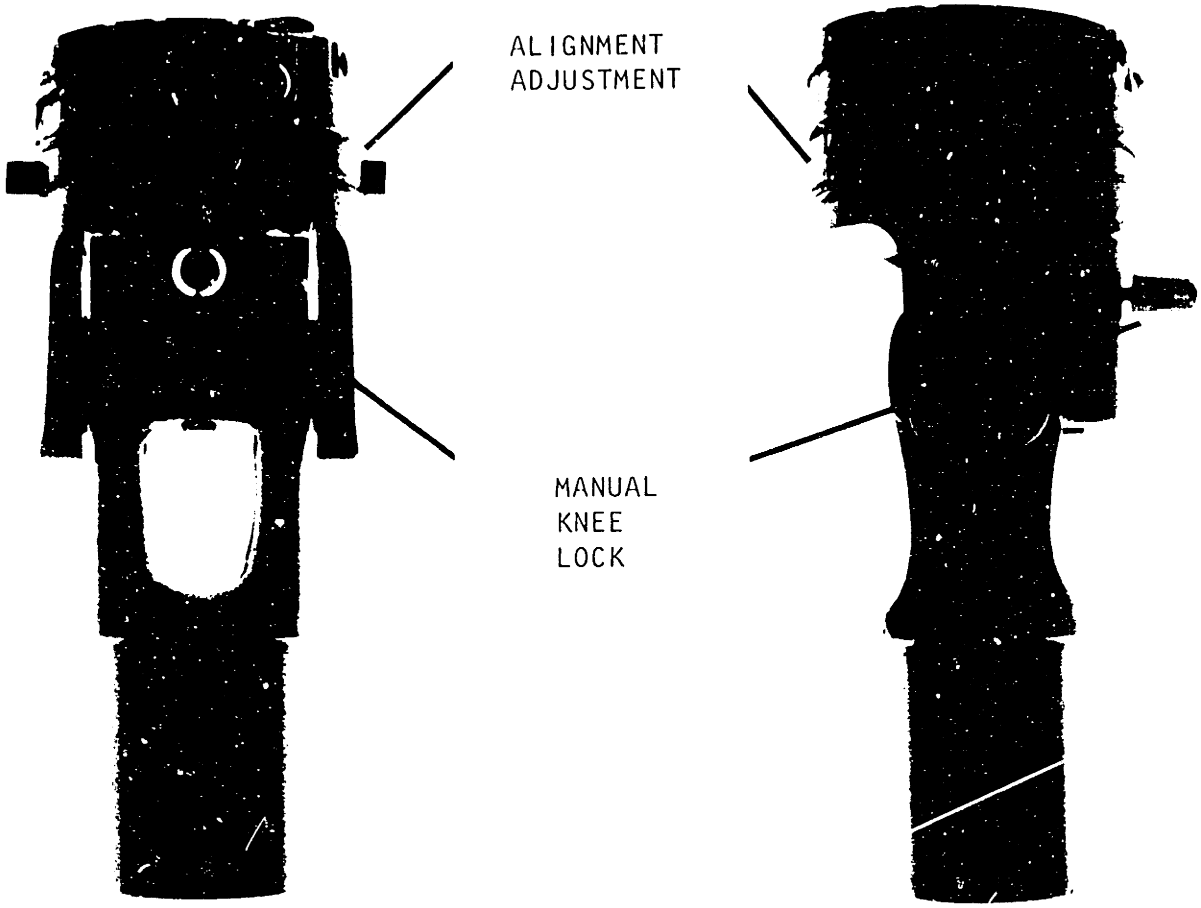
On the third to seventh day, the transition to an articulated prosthesis is made. This is usually brought about by simply unlocking



Front View

Side View

Figure 2-4 Temporary Knee Unit



Front View

Side View

Figure 2-4 Temporary Knee Unit

the knee joint. The amputee repeats previous exercises except this time the knee is free swinging. The unlocked prosthesis introduces new problems. Because of its lack of damping, the heel tends to rise excessively during flexion and the prosthesis stops abruptly when it reaches the end of its travel during extension. Also, once buckling is started for the initiation of swing phase, the prosthesis yields very quickly. Because of these problems, many geriatric patients never progress to this point with a temporary prosthesis.

Training with the free swinging prosthesis continues until the patient is discharged, usually about two weeks after his first training session. When released, younger and more active patients can typically walk with a single crutch or cane. Training is continued on an outpatient basis, two or three times a week. For geriatric patients, discharge may be postponed until receiving training with their permanent prosthesis.

Approximately three to six weeks after training starts, the amputee will be measured for a permanent prosthesis. The time for measurement is determined by the condition of the stump and the amputee's progress during rehabilitation. At this time, the amputee's physician, prosthetist and physical therapist will confer on a prescription for a permanent prosthesis. An additional delay on the order of one to two weeks may occur before the amputee receives his permanent prosthesis. Training with the definitive prosthesis occurs on either an in- or outpatient basis depending on the amputee's age and condition.

Most amputees are prescribed with a prosthesis that provides some form of damping. In light of this, the present training program seems to require an unnatural progression. The amputee starts with a locked prosthesis, makes the transition to a free swinging one, then receives a damped, permanent prosthesis. The major training emphasis occurs while the amputee uses the free swinging prosthesis. The amputee is faced with the challenge of controlling a temporary prosthesis that is usually less stable than the permanent one he will receive. The patient's reaction to this challenge is typically one of frustration.

CHAPTER III PROSTHESIS-CONTROLLER SYSTEM

The purpose of this work was to develop a system that would enable a new amputee to make a gradual and confident transition from a rigid training prosthesis to an articulated permanent prosthesis. Following the approach taken in earlier work at M.I.T. in above knee prosthetics, a separate prosthesis and a portable controller were developed for this purpose.

A conceptual block diagram of the system is shown in Figure 3-1. The amputee's input to the system is a torque about the knee axis produced by movement of the hip and stump. The desired output is a comfortable and cosmetic gait which will allow the amputee to walk in an energy efficient manner. The function of the controller and prosthesis is to generate a resistive torque about the knee axis based on the dynamic state of the prosthesis.

The next section describes the physical and operating characteristics of the prosthesis. The remainder of the chapter discusses

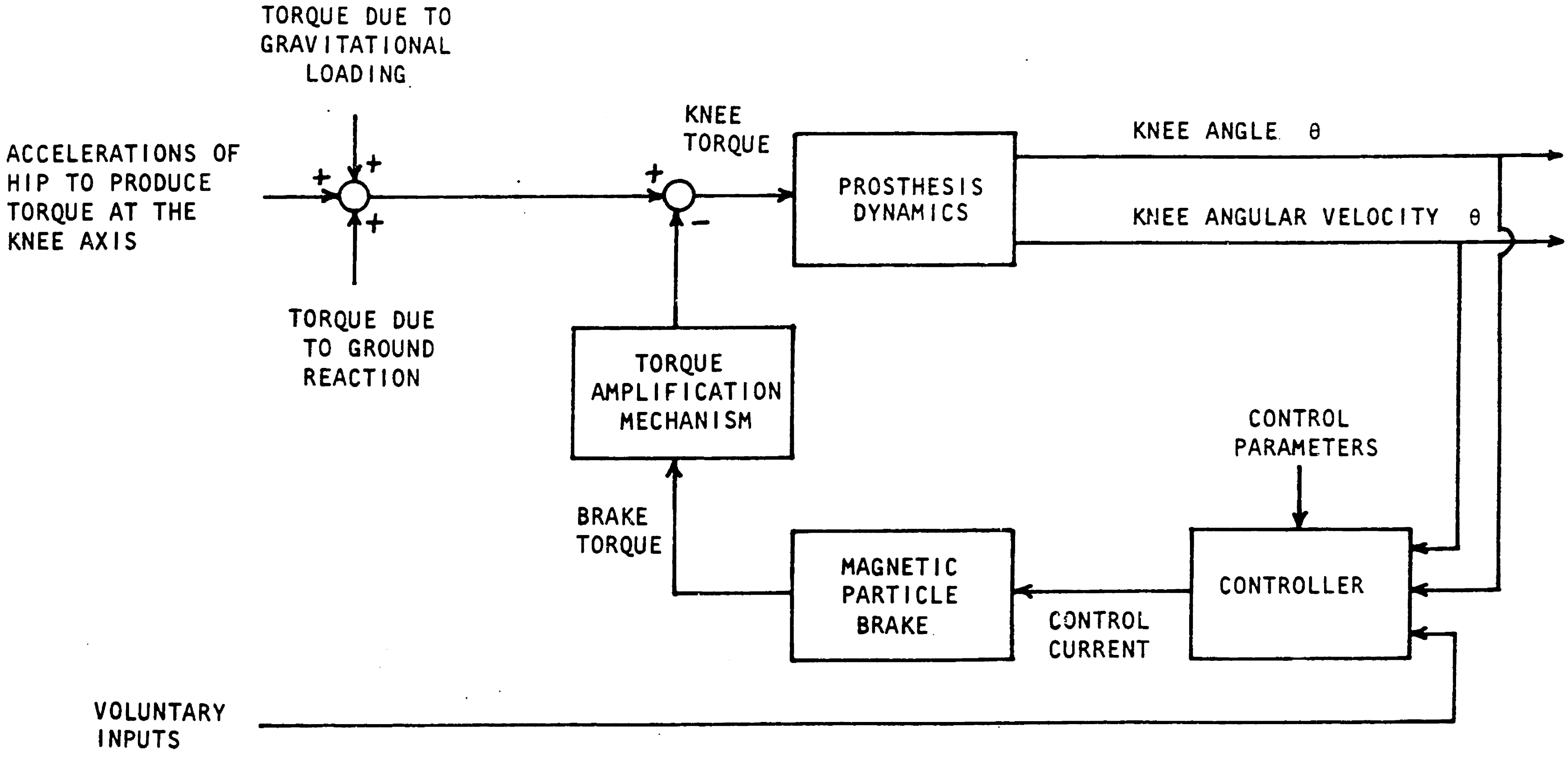


Figure 3-1 Diagram of the Prosthesis - Controller System

the controller and the algorithm used for control of the prosthesis.

3.1 Prosthesis

The prosthesis developed is similar to a previous prosthesis simulator design in that it employs a magnetic particle brake (MPB) as a source of resistive torque. The resistive torque produced is approximately proportional to the input current to the brake. The earlier prosthesis had limited possibilities for portable operation because of its high power requirement. However, the torque the MPB supplies is independent of its rotational velocity, therefore, a smaller torque capacity brake can be used if a larger torque amplification ratio is available to produce a resistive torque at the knee axis.

The original design developed by Lampe (8) employs a two stage reduction with a pulley and cable arrangement in the first stage and a chain drive in the second. The overall "gear ratio" between rotation at the knee and rotation of the brake is 1 to 7.5.

The later prosthesis takes advantage of a low friction recirculating ball nut and screw for torque amplification. Figure 3-2 shows a diagram of the mechanism used. The ball nut and screw serve to transform the torque at the brake shaft into a force directed along the axis of the screw. This force acts on the end of a cantilever to produce a torque about the knee axis. A pair of preloaded ball nuts help to eliminate backlash in the screw. The resultant effective "gear ratio", as defined earlier, is 1 to 40. A stop on the screw shaft allows a small amount of hyperextension for stability during stance phase.

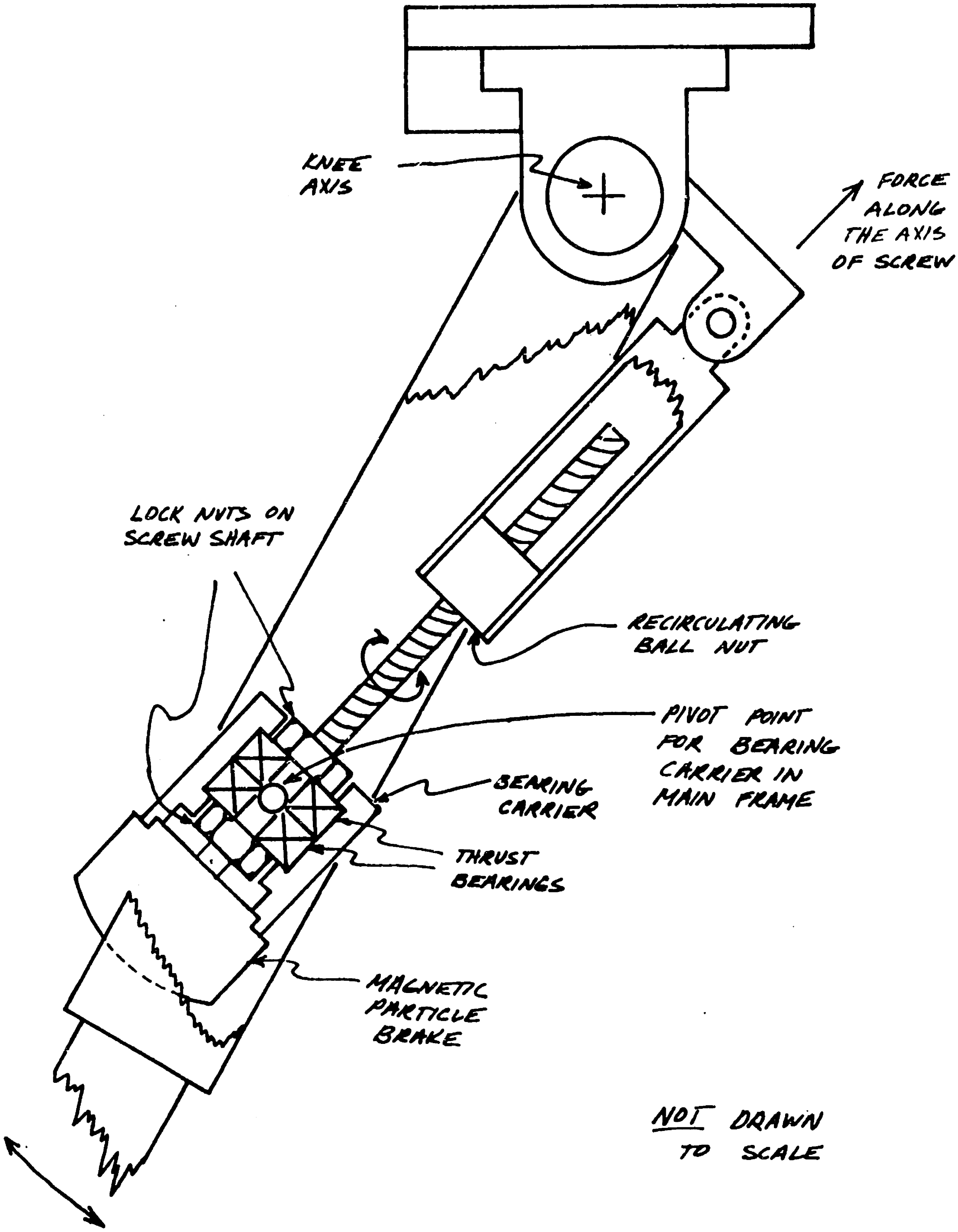


Figure 3-2 Diagram of Torque Amplification Mechanism

The magnetic particle brake chosen for the prosthesis is capable of a maximum torque of 10 in-lb. Full torque output requires a current of 110 milliamps. This results in an effective torque of 400 in-lb at the knee axis.

The prosthesis weighs 3 lbs 3 oz and has an overall length of 16 inches. When complete with a suction socket, SACH foot and shoe, it weighs 8 lbs 6 oz. The top plate provides a standard interface to sockets used by the M.I.T. Knee Project and to temporary sockets used for training.

The prosthesis has been instrumented to provide angular position information using a low-noise conductive plastic potentiometer mounted in line with the knee axis. The possibility of measuring knee torques exists by using the cantilever employed in the transmission as a transducer. Either displacement of the free end or strain in the beam could be measured to provide an indication of torque.

The conceptual design of the prosthesis was done by Prof. Woodie Flowers. Henry Cone, an undergraduate student in mechanical engineering, was responsible for component selection and physical design. It was developed as part of an ongoing program to provide amputees with a controlled prosthesis that is responsive to the individual needs of the user.

3.2 Controller Function

The purpose of the controller is to enable the prosthesis to operate with a large range of characteristics. The goals for the design of the controller were:

- it should be suitable for portable operation,
- a wide range of control parameters should be available so that the prosthesis operating characteristics can be varied from free swinging to heavily damped,
- adjustment of the control parameters should be simple and easily accomplished.

To achieve these goals, a microprocessor system with a stored program was chosen as the basis for the controller. A detailed description of both the components and program used in the controller are discussed in Chapter IV. The remainder of this section explains the controller's function and its use.

The controller's function can be characterized by the control scheme it uses to generate the value for the resistive torque of the prosthesis. The control law used for swing phase control is of the form:

$$T = b(\theta,\ \mathrm{sgn}\ \theta)\ \dot{\theta}^2$$

where θ is the relative angular displacement at the axis of the prosthesis. Figure 3-3 explains the convention used in defining θ. The damping profile, $b(\theta, \mathrm{sgn}\ \theta)$, is a function of the angle and the sign of the angular velocity so that different profiles may be used during flexion and extension.

Velocity squared damping was chosen so that the prosthesis would be responsive to changes in gait speed or cadence. Arguments have been made that this form should be used for ideal gait responsiveness (10), (11). Also, this form simulates the damping provided

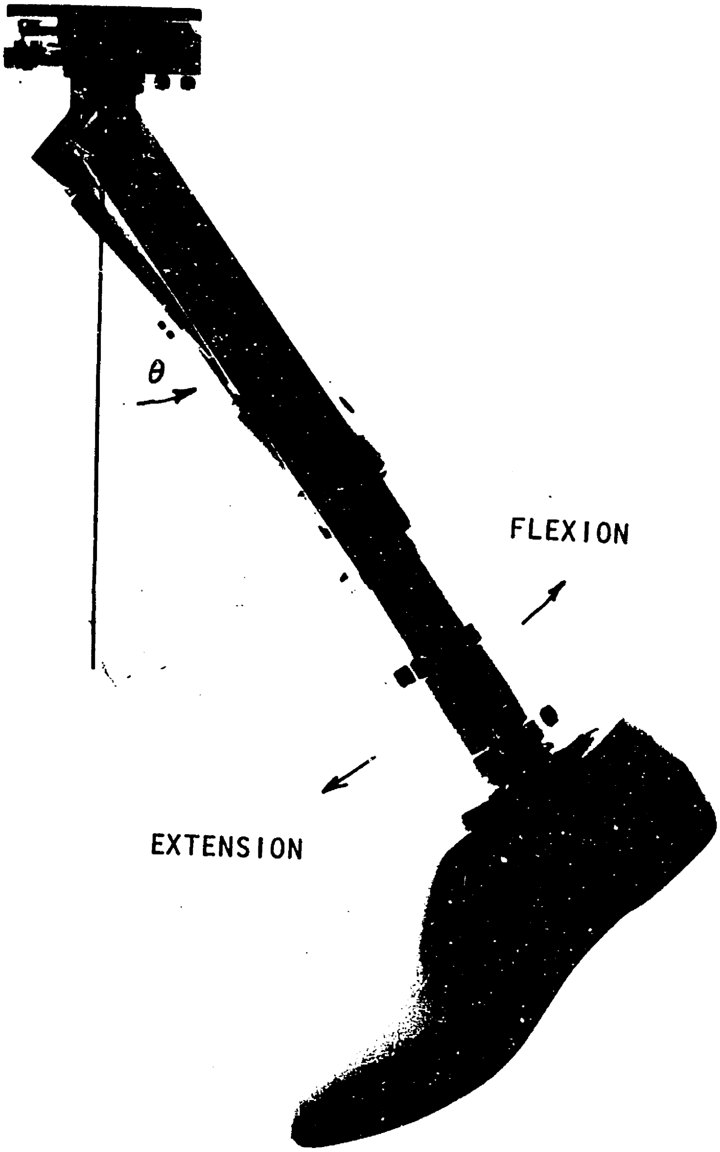


Figure 3-3 Diagram of the Prosthesis Angle

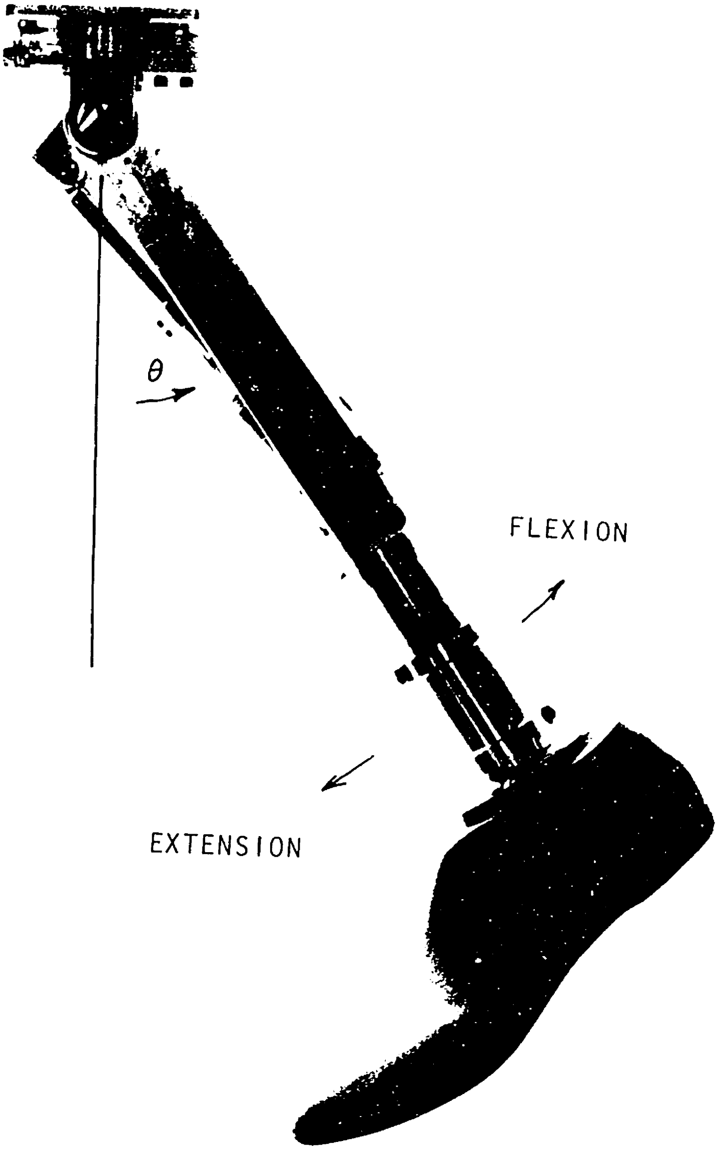


Figure 3-3 Diagram of the Prosthesis Angle

by pneumatic or hydraulic control units (13), (14).

The control unit is designed so that a damping profile may be developed empirically to suit an individual's needs. A series of 10 slide potentiometers (pots) is used to "draw" the damping profile desired. The pots form a graphical display of the profile used by the controller as shown in Figure 3-4. Separate inputs are available for the flexion and extension profiles. The display is arranged so that the damping used during a cycle proceeds from left to right.

When the reset button is depressed, the controller reads the values of the slide pots and creates an internal profile. For angles between those represented by the pots, linear interpolation is used. The result is a piece-wise linear table of 32 values each for both flexion and extension.

The "maximum angle" referred to by the display is an internally set value that determines the largest angle used in the definition of the damping profile (e.g. if maximum angle = 80° , the slide pots represent the damping values at 20° intervals). In the event the prosthesis angle should exceed that value, the damping for the maximum angle is used.

The gain adjustment is used to multiply the entire damping table by a constant factor. This enables the maximum resolution of the slide pots to be used in defining a wide range of normalized profiles.

If more accurate adjustment of the damping profile pots is required, two alternative methods are available for determining the profile. An interface box (see Figure 3-5) can be used with a voltmeter

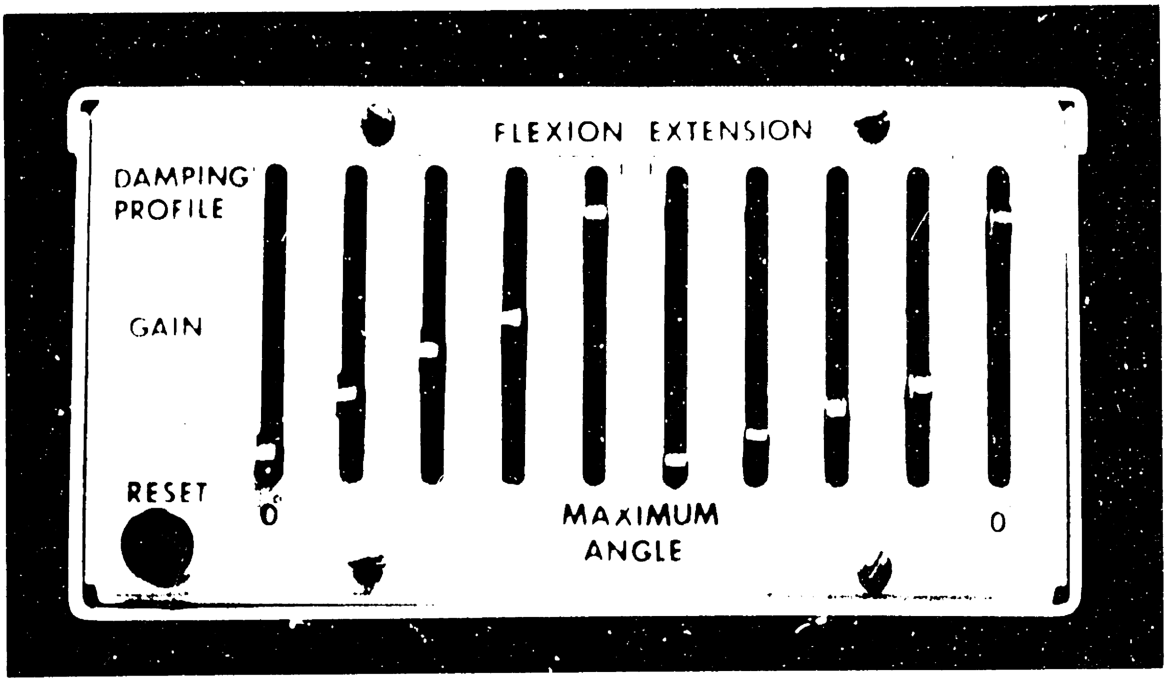


Figure 3-4 Damping Profile Display

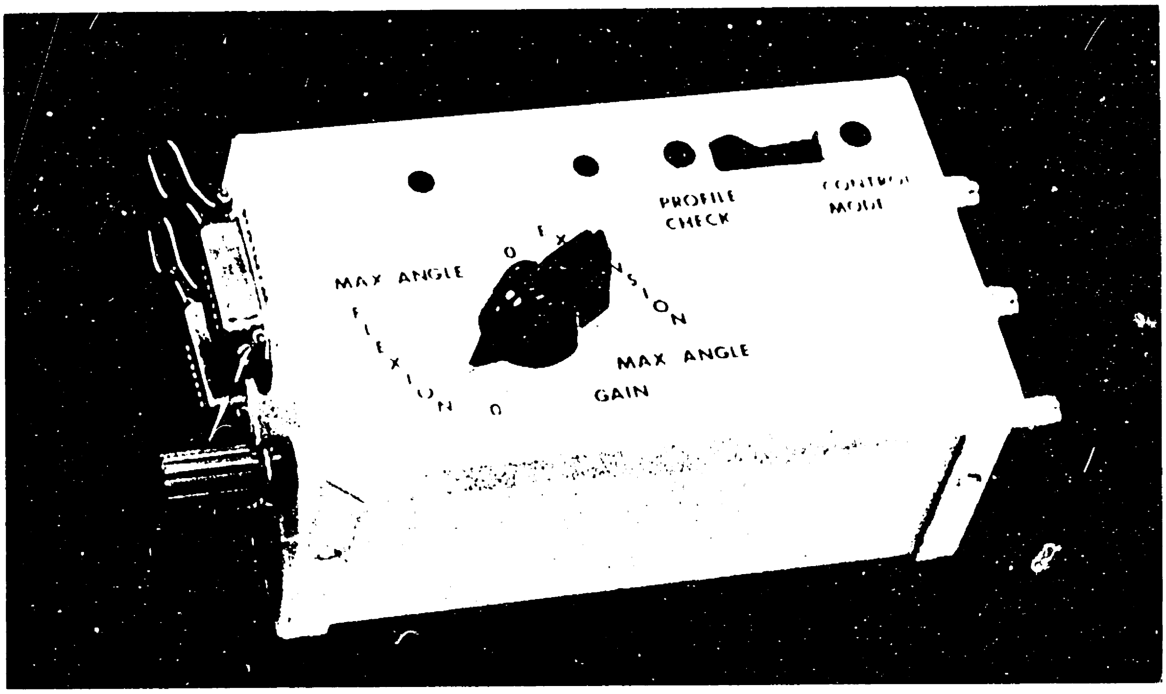
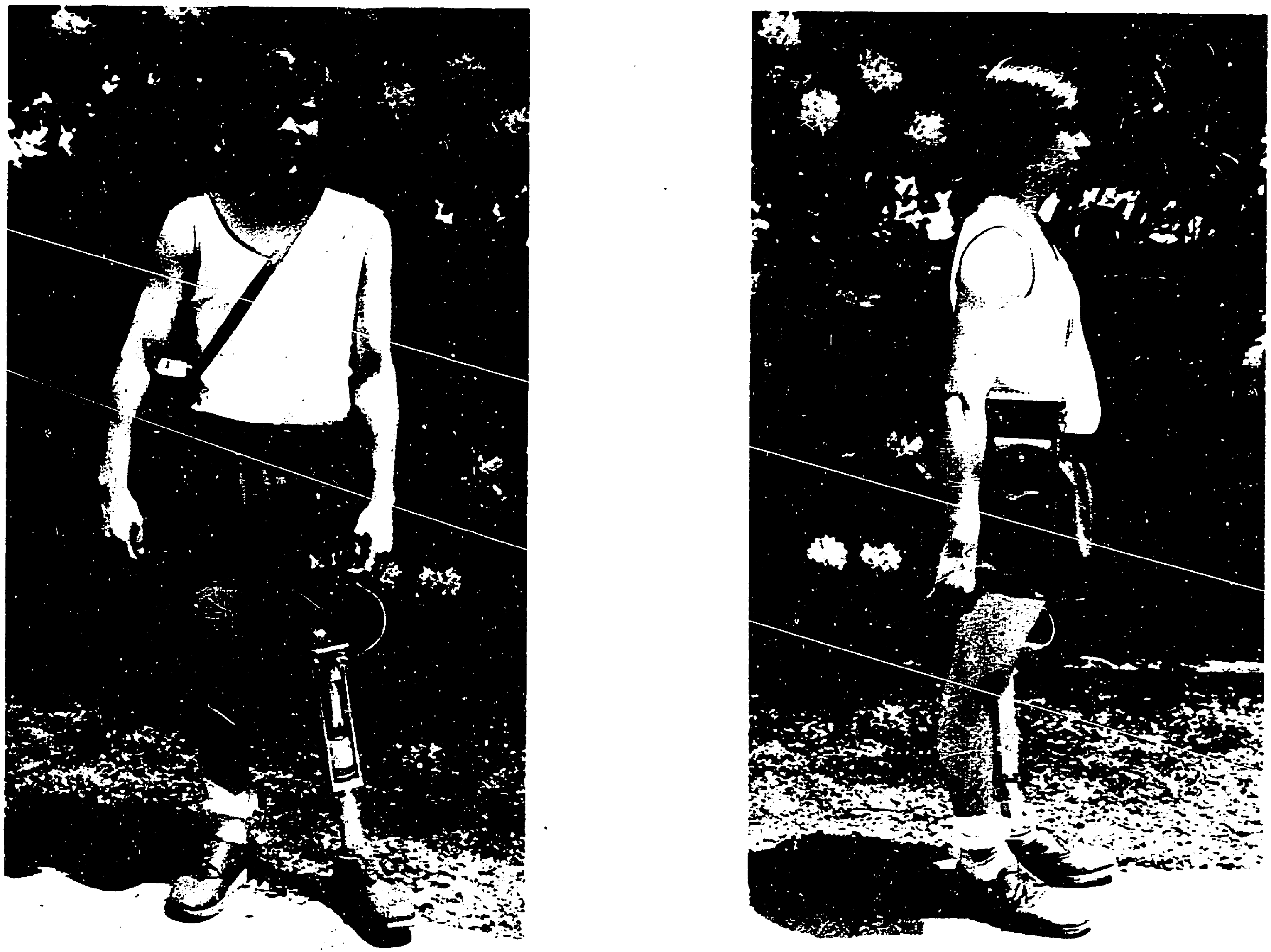


Figure 3-5 Interface Box

to position the profile pots and to set the gain value. This box also provides a means of recording analog data for the prosthesis position and velocity as well as the control signal generated. The control unit also has a serial input/output port that may be used with a data terminal to furnish hardcopy output of the gain and profile settings.

A hand-held pushbutton may be utilized to provide the amputee with voluntary control of the prosthesis. While the button is depressed, the prosthesis is locked at its current position. A "lock" command from the pushbutton overrides the control unit output.

The prosthesis-controller system is designed to be completely portable. A small umbilical is used to transfer information between the prosthesis and the controller. Figure 3-6 shows an amputee wearing both the prosthesis and control unit.



Front View

Side View

Figure 3-6 Amputee Wearing the Prosthesis-Controller System

Front View

Side View

Figure 3-6 Amputee Wearing the Prosthesis-Controller System

CHAPTER IV CONTROLLER DESIGN

Past and present research at M.I.T. has proven the digital computer to be a valuable tool in the development of control schemes for above-knee prostheses. The flexability of a stored-program digital controller, with a suitable interface, frees the investigator of many of the intricacies associated with an analog controller. Through the use of semi-permanent memory, a large variety of control algorithms may be investigated by simply changing the the controller program.

In the design of any digital controller, trade-offs must be made between hardware (electronic components and modules) and software (programming). These trade-offs will directly affect the controller's size, speed, power consumption and versatility. In this work, the approach taken was to demonstrate a concept rather than construct a sophisticated electronics design. Therefore, the trade-offs were made in favor of simplified hardware, at the expense of more complicated software. The net result of this approach was a controller with less than optimal size, weight and power consumption that may be used for a large number of applications.

The remainder of this chapter discusses both the hardware and software design of the controller. In the first section the microprocessor system and the interface electronics are described. The later section explains the program developed to implement the control function.

4.1 Hardware Design

From the beginning of the hardware design, the decision was made to rely on available electronic modules rather than attempt to build a system from basic components. This approach implies that compromises will be made, but it was hoped that a minimum amount of time would be spent debugging the hardware. Once the concept had been demonstrated, this preliminary system would help to define requirements for a second generation design. A block diagram of the complete system is illustrated in Figure 4-1.

4.1.1 Microprocessor System

The Motorola 6800 microprocessor was selected for the controller because of the availability of a development system through the M.I.T. Sensory Aids Development and Evaluation Center. Personnel at the Center have produced several successful intelligent devices utilizing the 6800.

The 6800 microprocessor system used in the controller resides on a single 6.5 by 4.5 inch card made by Wintek Corporation. The basic card provides a minimal system that can be expanded to meet the needs of a particular application. A block diagram of the system used in the

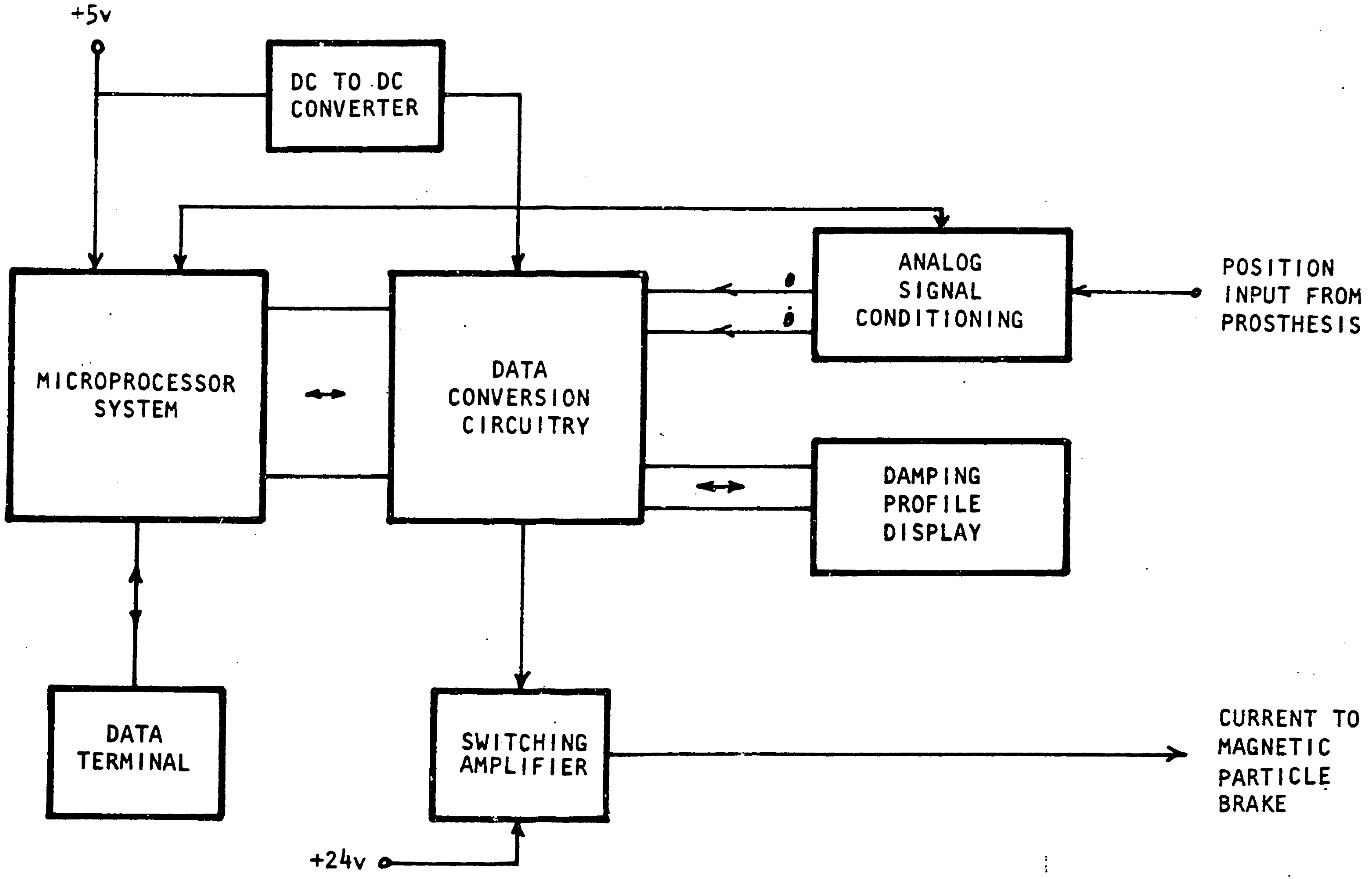


Figure 4-1 Diagram of the Controller Components

controller is shown in Figure 4-2.

The minimum configuration contains a 6800 central processing unit (CPU) and the necessary clock and address decoding circuitry, as well as address and data lines. Provisions exist for the addition of memory, both permanent and volatile, and interfaces to simplify serial and parallel input and output. For the most part, the use of the Wintek module makes the hardware details of the microprocessor system invisible to the user and enables him to concentrate on interface electronics and programming.

The Motorola 6800 is an 8 bit microprocessor with an instruction set well suited to modular, arithmetic programming using subroutines. The CPU contains two general purpose accumulators for arithmetic and logical operations. An index register is available for simplified table processing and a stack pointer allows multiple subroutine calls and last-in-first-out storage on the stack.

The control program is stored in an 1024 byte erasable programmable read only memory (EPROM). The Wintek card is supplied with a socket to facilitate easy removal and insertion of EPROMs for different control programs. Random access memory (RAM) is used for temporary storage while the CPU is in operation. The card has provisions for four 128 byte RAMs. Presently only two RAMs are used.

Two peripheral interface adapters (Motorola, model 6820) are employed for parallel input and output. Each PIA has two 8 bit ports which can be individually programmed, bit by bit, to serve as either an input or output line. The PIA appears to the CPU as a memory location which simplifies

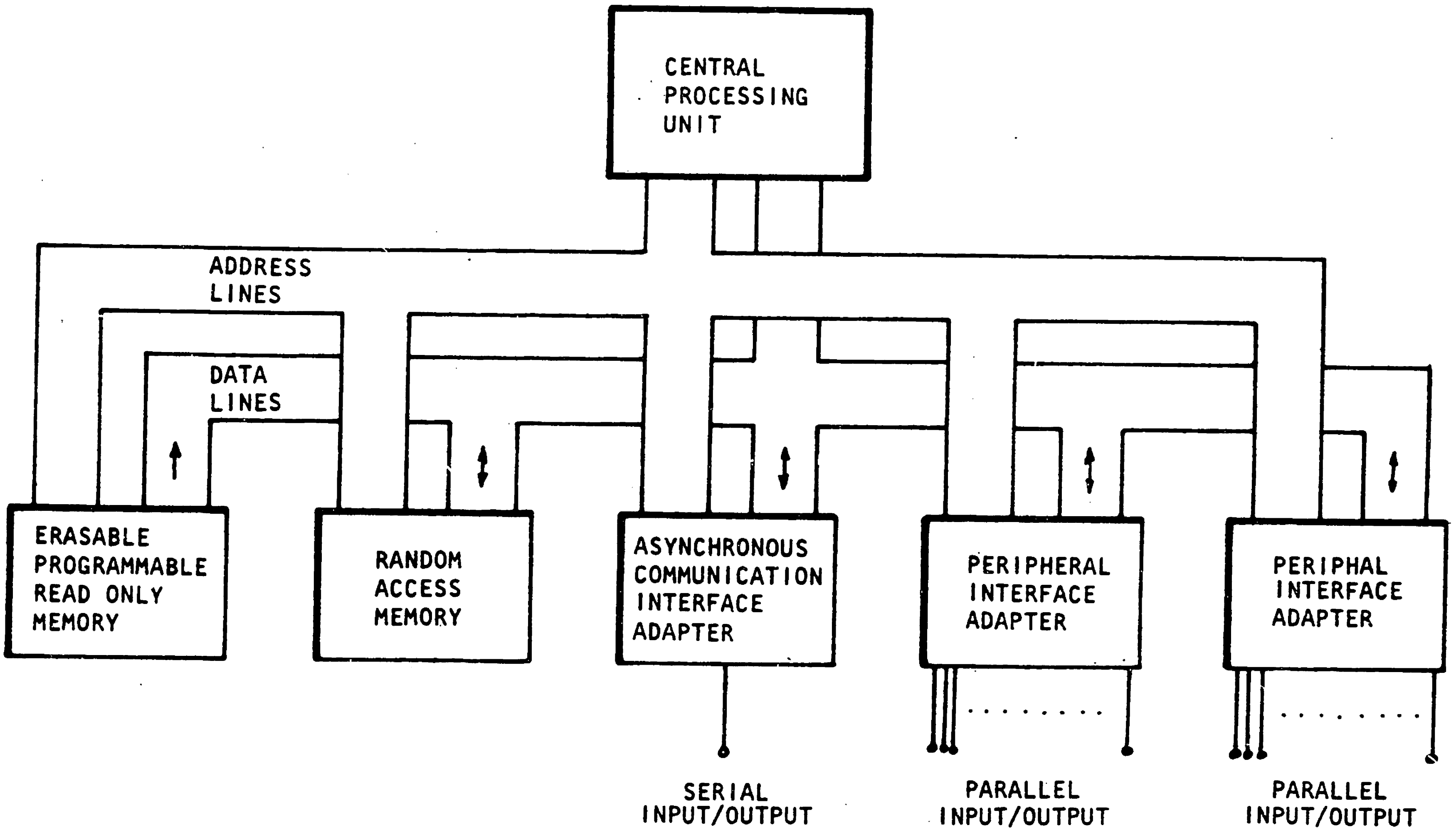


Figure 4-2 Diagram of the Microprocessor System

input/output operations. Serial input/output is established with the CPU using an asynchronous communications adapter (ACIA, Motorola, model 6850). This device translates a parallel received byte into timed high and low signals for data transmission and vice versa. It provides both input and output functions and it also appears as memory to the CPU. A separate interface circuit is furnished so the CPU can communicate with a data terminal using the RS-232C standard at a data rate of 110 characters per second. A memory map for the complete system appears in Appendix A.

4.1.2 Data Conversion Circuitry

An interface must be provided between the PIAs and the analog inputs and outputs. A block diagram of the modules used for this purpose is shown in Figure 4-3.

Starting at the input end, a 16 channel single-ended analog multiplexer (Burr Brown, model 16S) is used to select an input channel. The multiplexer output goes into a sample and hold amplifier (Burr Brown, model SHC 80) which presents a high impedence to the input signals and holds the input value during a data conversion. An 8 bit bipolar analog to digital converter (Burr Brown, model ADC 82) performs the data conversion. Over its input range of ± 5 volts, it can resolve to within ± 1 bit or ± 0.38%. When under microprocessor control, this system can change channels and complete a data conversion within 32 microseconds.

A digital to analog converter (Analog Devices,model 7520) provides an analog output for the controller. It converts an 8 bit input from the microprocessor into a voltage output between 0 and 5 volts with a single

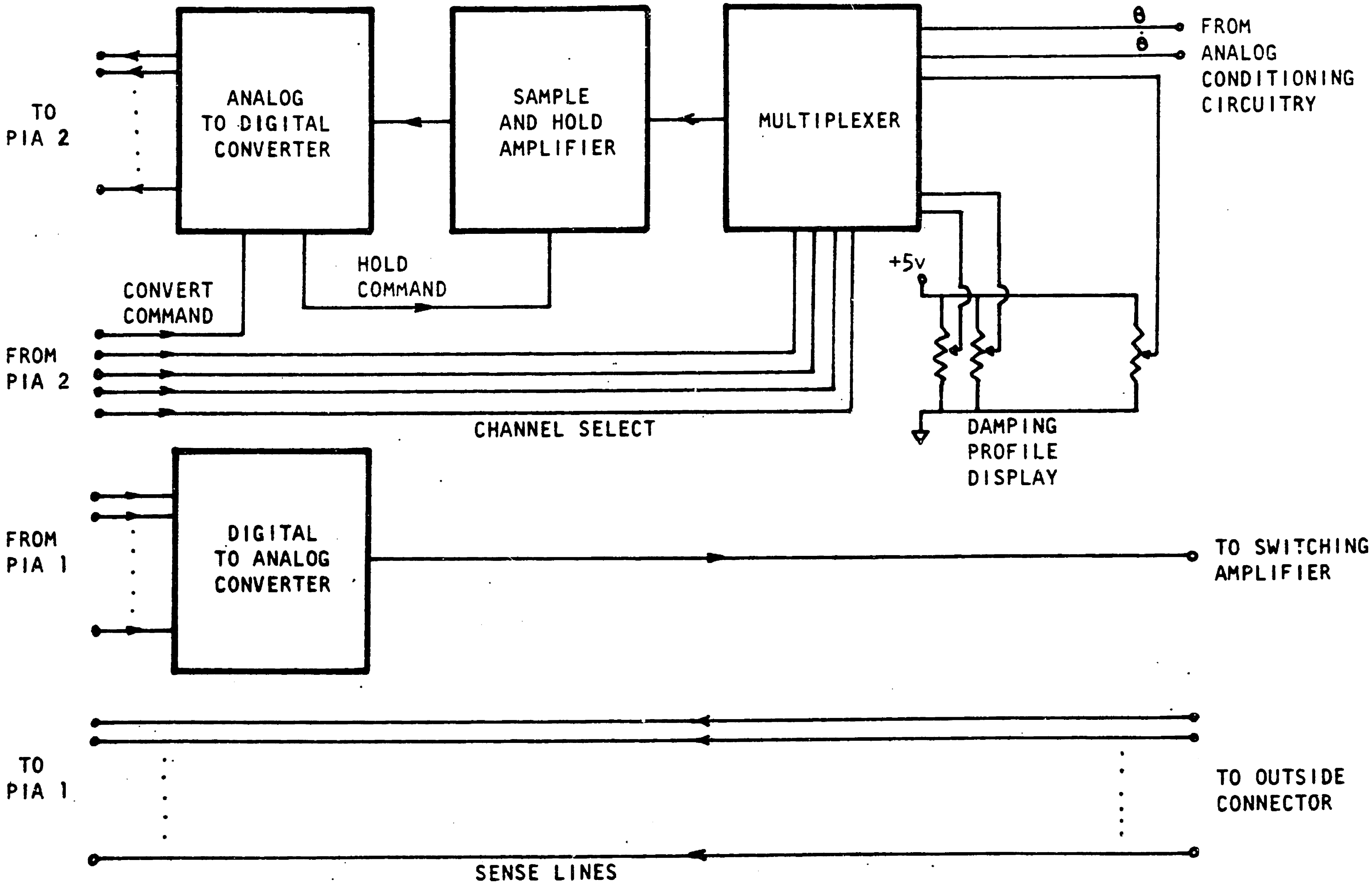


Figure 4-3 Diagram of the Data Conversion Circuitry

bit resolution of 19.5 mV or 0.39%.

4.1.3 Analog Input Signal Conditioning

The function of the analog input electonics is to buffer the inputs and take advantage of the dynamic range of the A/D converter. A psuedo-differentiator is used to derive velocity information from the angular position signal. A break frequency of 1000 Hz was chosen to minimize gain and phase lag error. A precision potentiometer acts as an angular position transducer to produce a low noise signal. A schematic of the input electronics is given is Appendix B.

4.1.4 Controller Power Supply

The modules selected for the microprocessor system and data conversion circuitry require five different supply voltages (-15,-12,+5,+12, +15) at different current levels. (For a future controller, a careful hardware design should eliminate the need for some of these voltages.) This combination of voltages and currents is not easily supplied by series connected batteries, so a single +5 volt supply was chosen from which the other voltages can be derived. A DC to DC converter (Analog Devices, model 940) is used to generate ± 15 volts. The other voltages required are derived using series-pass regulators.

The DC to DC converter operates using the 5 volt supply to create a 20 KHz alternating current. A small transformer is used to step up the voltage which is then rectified and filtered to produce ± 15 volts. Because of its operation, it requires 165 milliamps (mA) at no load resulting in a power consumption of 0.825 watts.

Figure 4-4 illustrates the relative magnitudes of the various currents required for the individual components used in the design

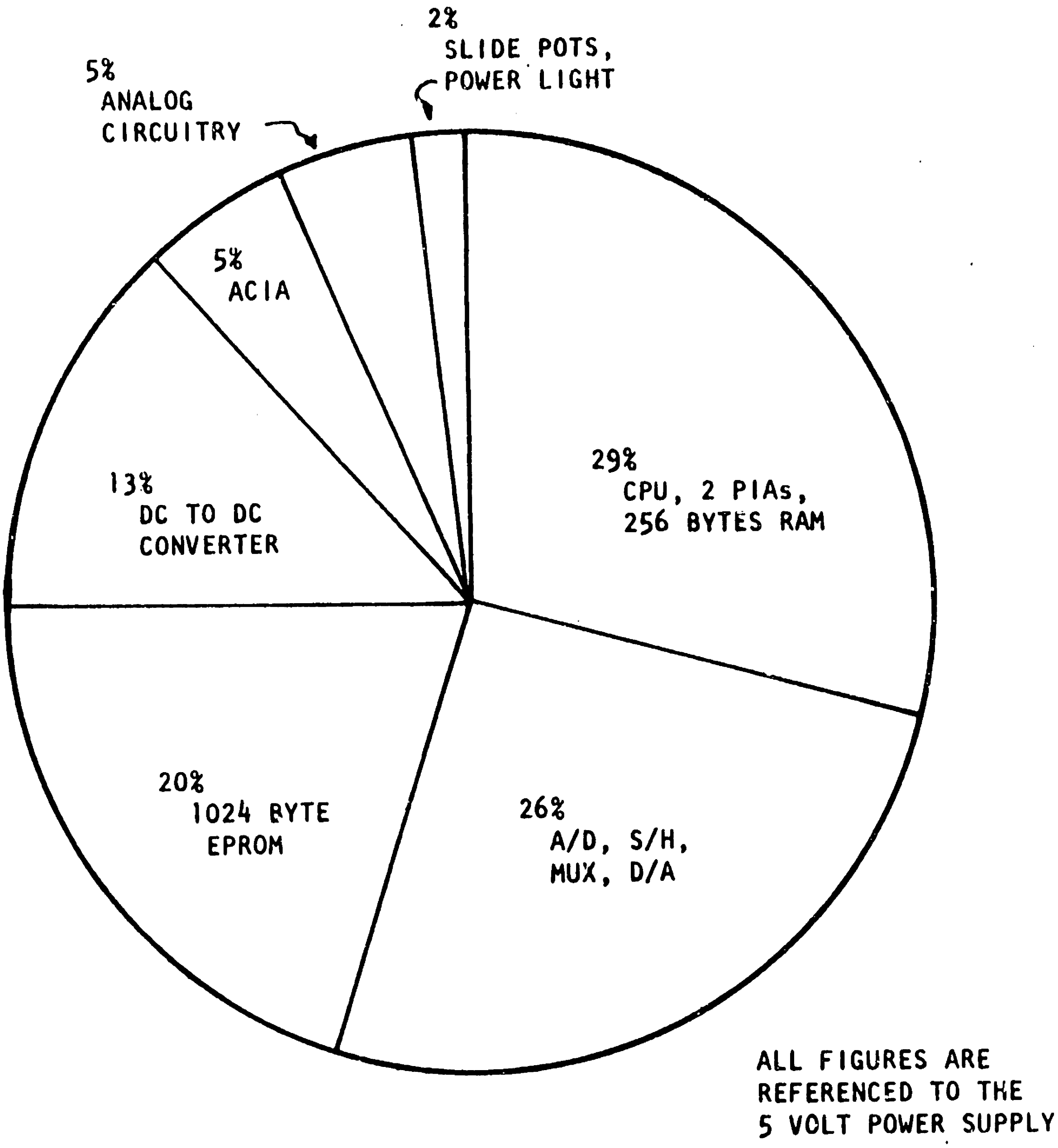


Figure 4-4 Break-Down of the Controller Power Consumption

of the controller. All of the figures shown are referenced to the 5 volt supply and include the inefficiency of the series-pass regulators.

The controller consumes 1.3 amps of current at the nominal 5 volt supply. However, the controller will operate on a supply voltage between 4.75 and 5.5 volts. Over this range, the controller behaves like a resistive load (i.e. a lower input voltage results in a proportionally lower current drain). An 8% difference exists between the current consumed at the maximum and minimum supply voltages. Therefore, to reduce power consumption, the supply voltage should be as close to 4.75 volts as possible.

For portable operation, either lithium or nickle-cadmium batteries may be used. Six primary lithium D cells (Power Conversion Inc., model 550) connected in three series pairs, can be used for 19 hours of continuous operation. Five rechargeable nickle-cadmium D cells connected in series (Eveready, model CH 4) will provide power for about 3 hours of portable operation. At maximum charge rate, 14 hours is required to restore the batteries to full charge. In either case, the batteries are used unregulated because their useable voltage falls within the recommended range for the system components. In the event the controller is to be used as a stationary device, provisions exist to allow operation with an external power supply.

4.1.5 Current Driver for the Magnetic Particle Brake

The brake used in the prosthesis develops a resistive torque proportional to its input current up to a given saturation current.

This device can be modeled as an inductive load with a distributed resistance of 170 ohms. Its electrical time constant is on the order of 20 msec. A second time lag exists between the current input and the output torque.

The ideal way to drive an inductive load is with a current source but this implies an unlimited voltage is available to instantaneously change the current in the inductor. For portable operation, a limited voltage supply must be used instead. A voltage switching amplifier drives the brake to provide the necessary power gain for the output signal from the controller.

A switching amplifier avoids the power dissipation associated with operating the final driver stage transistor in its linear range. It operates by supplying maximum voltage for a part of a fixed period. The ratio of "on" time to the total period determines the average output. The final drive transistor is either in saturation or cutoff mode, both of which dissipate minimum power within the transistor. If the switching period is much smaller than the time constant of the system to be driven, the load effectively filters the pulsed signal to the average value. However, the response time is the same as it would be for a step input with a magnitude equivalent to the averaged input.

The switching amplifier used was designed and built by Ted Fischer, a technical instructor in mechanical engineering. A schematic for the amplifier is given in Appendix C. Its switching frequency is 1600 Hz. Power for both the amplifier and the brake is supplied by four 9 volt alkaline transistor radio batteries. The supply voltage is maintained at 24 volts using a series-pass regulator. Experiments have shown that the four batteries provide enough energy to power the prosthesis for 65 hours of normal level walking.

4.1.6 Packaging

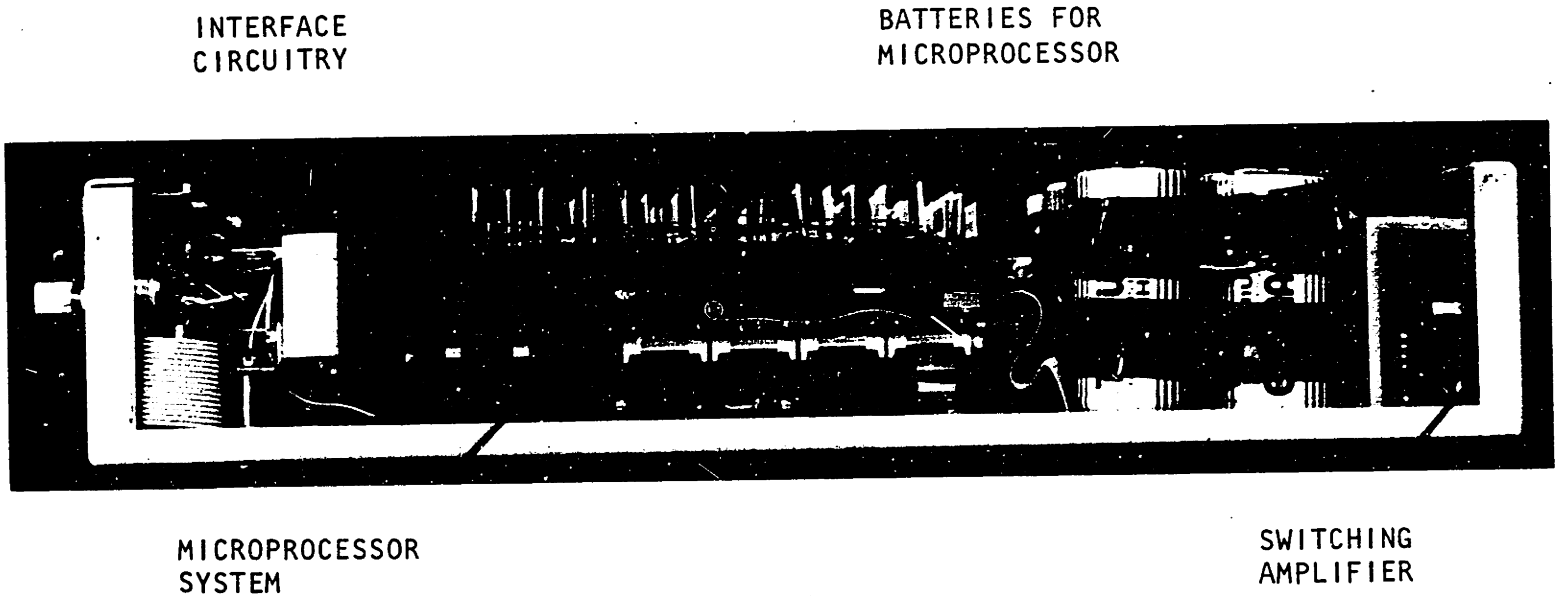
The controller unit contains the microprocessor system, interface circuitry and the current driver stage and their respective power supplies. Figure 4-5 pictures the controller with the cover removed and the components labeled. The unit is 13.5 by 5.5 by 4.0 inches and weighs 5 lbs 6 oz with the leather case. The controller was designed to be carried at the hip with its weight supported by a shoulder strap and a waist belt, as shown in Figure 3-6.

To operate the controller, a 9 pin connector, which serves as a key, is inserted into the power selection outlet. Different keys enable the choice of internal or external power.

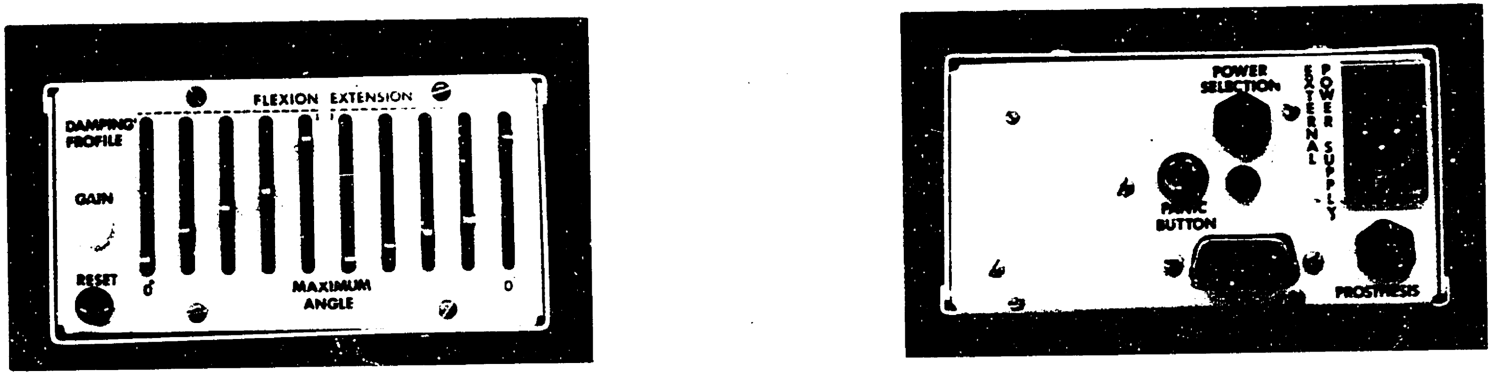
4.2 Control Program

The program currently in the controller serves two functions. It enables the microprocessor to act as a real-time digital controller or as a monitor to record and adjust the controller settings. The function the controller assumes depends on the state of externally accessable sense lines. The connection of an interface device alters the state of the sense lines to make the controller perform the desired function.

Figures 4-6 A and B consist of a flow chart of the program in the controller. A listing of the actual program appears in Appendix D.



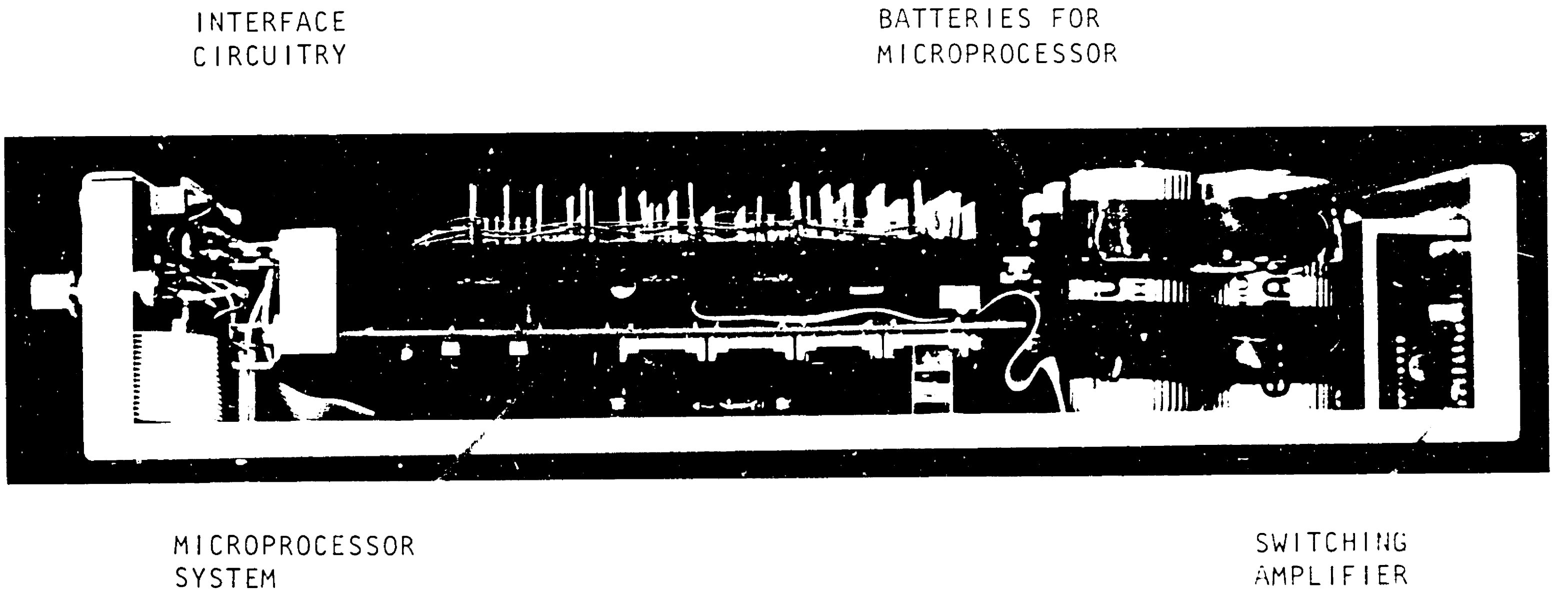
Side View



Top View

Bottom View

Figure 4-5 Controller Packaging



Side View

Top View

Bottom View

Figure 4-5 Controller Packaging

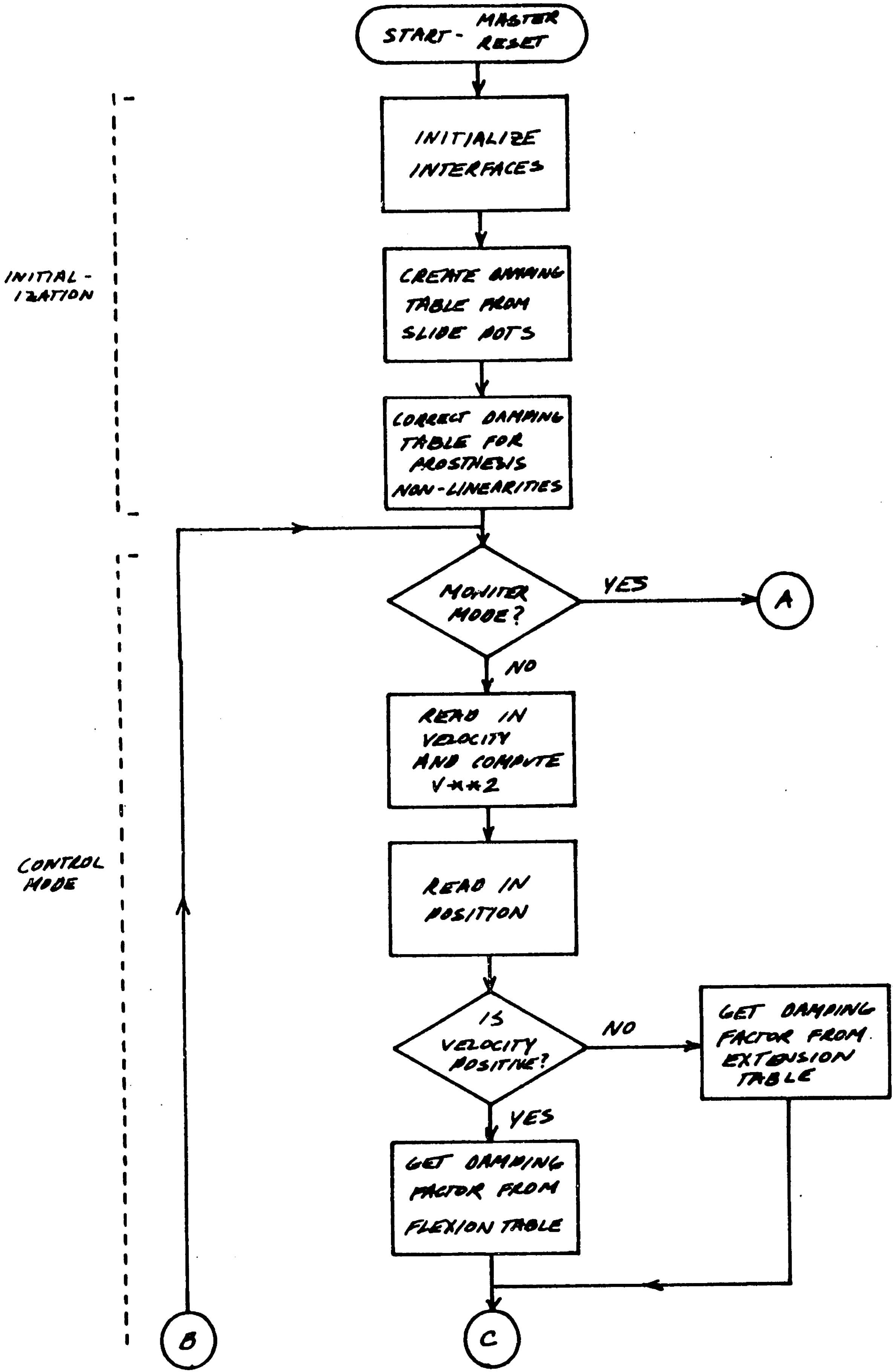


Figure 4-6A Flowchart of Controller Program

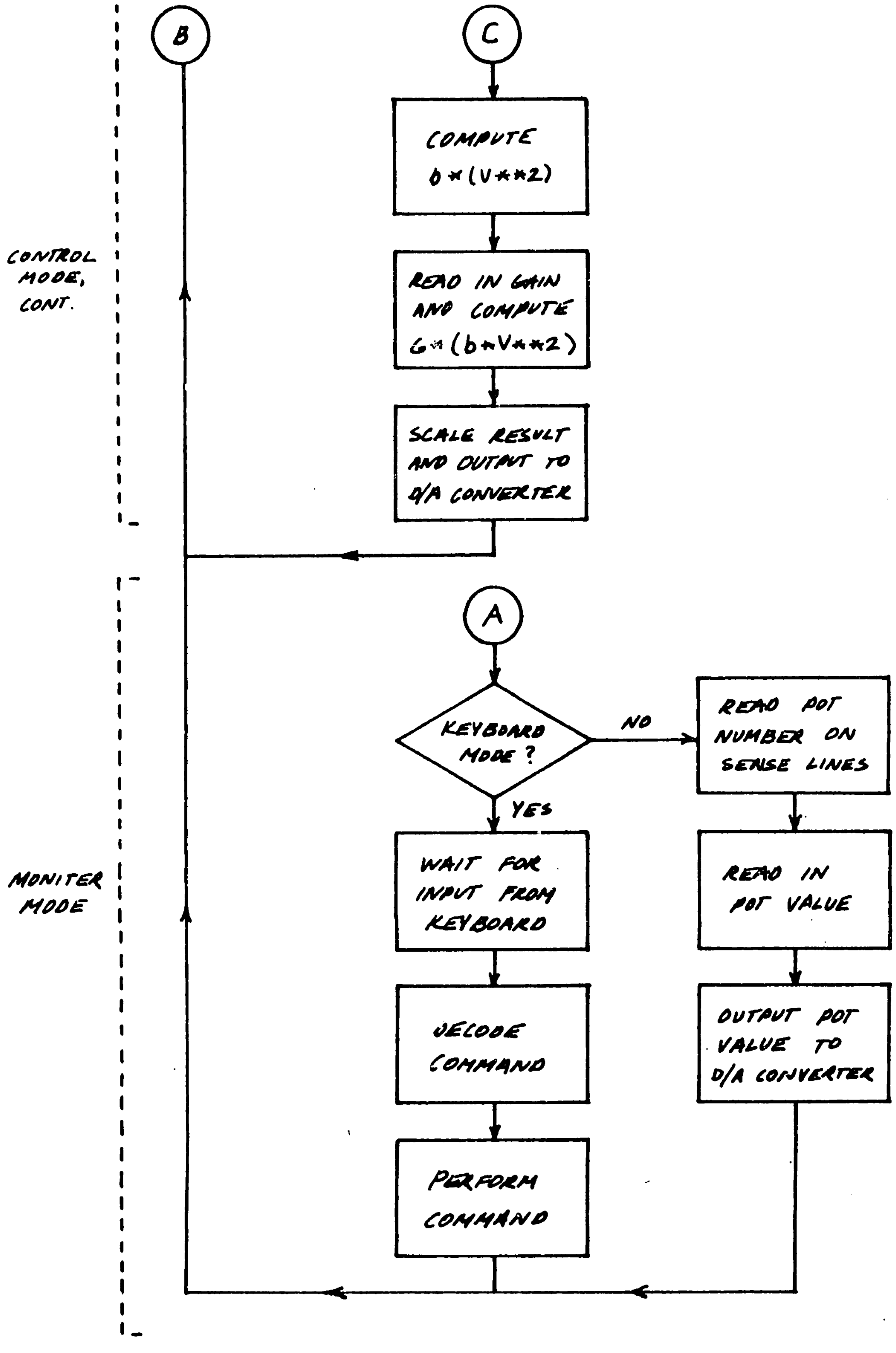


Figure 4-6B Flowchart of Controller Program, cont.

Whenever possible, subroutines were used to simplify the programming task. Two's compliment format was chosen for numeric representation. This allows for a range of integer values from -128 to 127. When required, modified floating point arithmetic was used to increase the range of numbers available during computations. As shown in Figure 4-6 A and B, the program can be divided into three sections: initialization, control loop and monitor function. These portions will be discussed separately.

4.2.1 Initialization

Whenever the reset button is depressed, a complete system reset takes place. This is used for a reset from a power up condition or to input new control parameters. After the interface have been initialized, the slide pot values are read in to create two damping profiles. Two 32 position tables are developed by linearly interpolating between the slide pot settings. The damping coefficients are then modified using a stored table to correct for non-linearities in the torque amplification mechanism used in the prosthesis. The stored table used for correction was computed offline using a minicomputer system.

The damping profiles created are used until the reset button is depressed again. Moving the slide pots while operating in control mode has no effect. While initialization occurs, the prosthesis is rigidly locked. This process requires approximately 80 msec.

4.2.2 Control Loop

After initialization and determination of the control parameters, the main program loop is entered. The first step in this loop is to check the state of the sense lines for a command to enter the monitor mode. Otherwise, the program continues in the control loop. The value of the angular velocity is then read in and is squared using a floating point multiply routine. The position value is obtained and is used in conjunction with the sign of the velocity to determine the proper damping coefficient. The initial torque signal is computed using this coefficient. The value for the gain is then determined and is multiplied by the initial torque signal to generate the final control signal. This allows the overall controller gain to be adjusted while the controller is in operation. The final result is then scaled to the input range of the D/A converter and is sent out to the current driver for the brake. The program continues in this loop until there is a master reset of a command to enter the monitor mode.

The execution time for the complete control loop is 3 to 4 msec, corresponding to an update rate of 250 to 330 Hz. The loop time is not constant because the execution time of the multiply routine varies for different input levels.

4.2.3 Monitor Mode

The monitor program provides access to the control parameters with a data terminal. If connected to a keyboard device, commands exist for printing the values of the slide pots and examining and changing the contents of memory. The commands can be used to document experiment or assist in diagnosing controller problems. In the event a terminal is not available, the program may also be used with the interface box shown in Figure 3-5 to assist in adjusting the control parameters.

CHAPTER V EVALUATION, CONCLUSIONS AND RECOMMENDATIONS

5.1 System Evaluation

Before attempting to evaluate the prosthesis-controller system in a clinical environment, a series of tests were performed to explore the capabilities of the system. The purpose of these tests was to compare the performance of the system to that of a temporary prosthesis as shown in Figure 1-1.

To determine a base-line performance for both the temporary and the prototype prosthesis, a simple test was conducted to estimate the residual damping present when operating in a free-swinging mode. For this test, the mounting plate that would normally be connected to a socket was attached to a surface to enable the prosthesis to behave like a pendulum. A SACH foot was attached to simulate the normal weight distribution. After determining a rest position, the prosthesis was displaced and allowed to return to a new rest position. The results of these tests are illustrated in Figure 5-1. The response of both systems can be characterized as that of a second order system. In the case of the conventional

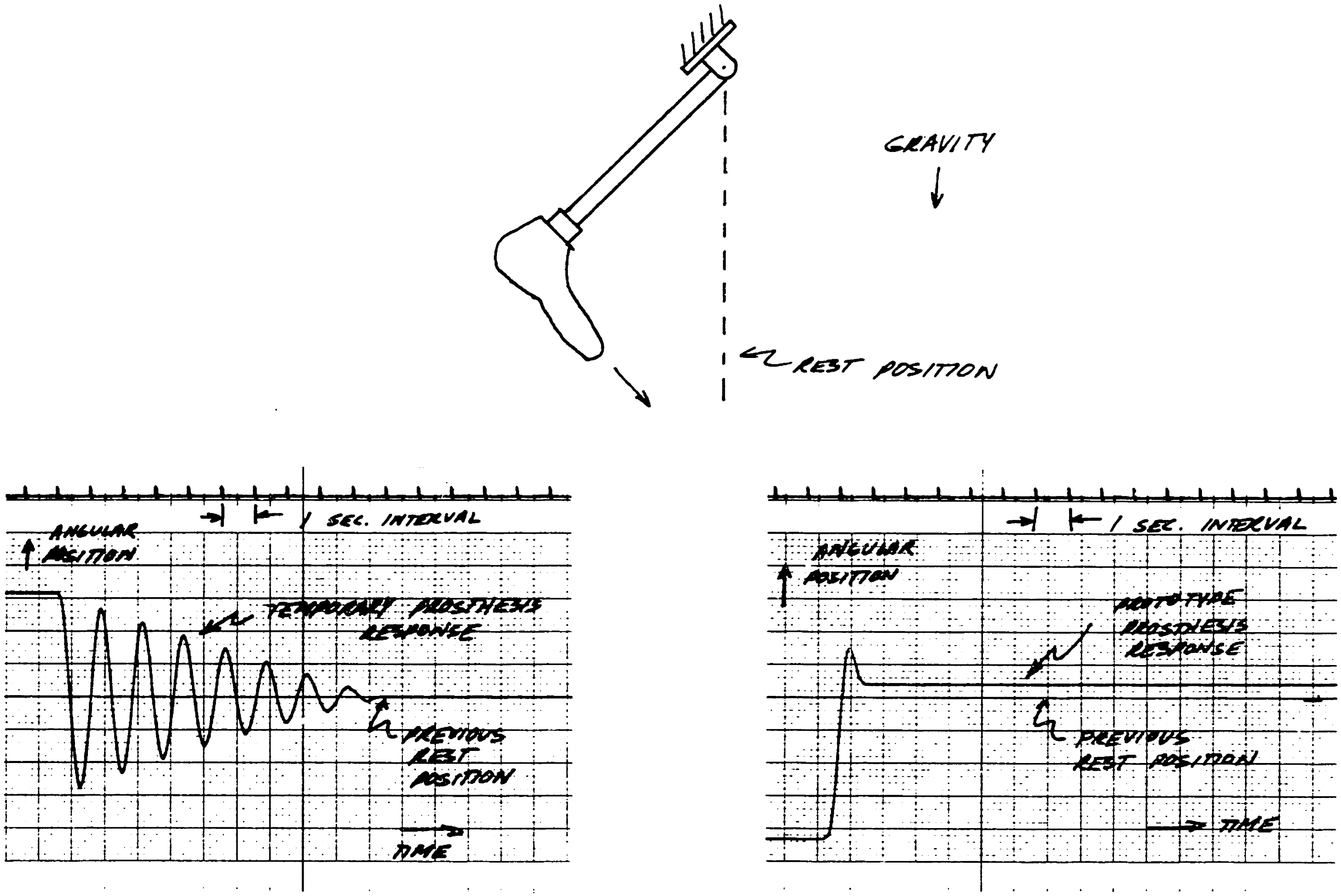


Figure 5-1 Response of the Prostheses to a Step Displacement in Position

training prosthesis, the response shown is very lightly damped and it returns to its previous rest position. In contrast, the prototype prosthesis, with a zero current input to the magnetic particle brake, has only a single overshoot in its response and does not return to its rest position. This implies that some form of stick-slip friction is present, probably due to the residual friction in the brake reflected through the transmission.

Amputee-interactive tests were performed with the assistance of an experienced amputee who has also been involved in the evaluation of other M.I.T. above-knee simulator systems. The subject was Stephen Cornell, a twenty-two year old amputee. At the age of twelve, Steve experienced an amputation due to trauma below the knee of his left leg. Because of infection he was later re-amputated above the knee. His permanent prosthesis employs a Dupaco pneumatic control unit, a SACH foot and a quadrilateral total-contact suction socket.

Steve works as a grocery clerk and occasionally does part-time carpentry work. He is very active and participates in a number of sports. He weighs 135 pounds and is five feet, seven inches tall. Steve is a willing and accommodating test subject.

During all of the tests conducted,the subject was asked to walk with the prosthesis for a short period of time to familiarize himself with its characteristics. After that period, the subject was instructed to walk at what he felt was a comfortable pace while data was recorded. Angular postion and velocity information were transmitted from the prosthesis using a long umbilical cord.

To serve as a basis for comparision, a temporary prosthesis was constructed using a United Manufacturing temporary knee unit. After the initial adjustment period, data was recorded. Two typical cycles of the position and velocity signals are shown in Figure 5-2. The noise present on the velocity trace is due to the analog differentiator used to create the velocity from the position signal. These traces show many of the characteristics normally associated with an undamped prosthesis. The maximum heel rise is 96 degrees and velocities of 10 rad/sec are reached during flexion. The velocity changes abruptly at the end of extension as the prosthesis impacts against the hyperextension stop. During this test the subject stated that the temporary prosthesis seemed to require that he walk quickly. His cadence during this trial was 99 steps per minute.

In an attempt to duplicate the dynamics of the temporary prosthesis, the subject was asked to walk with the prototype prosthesis without the use of the controller. The resultant data is shown in Figure 5-3. During these experiments, the maximum heel rise was approximately 87 degrees with peak velocities as high as 8.5 rad/sec during extension. Once again, the velocity decreases sharply at the end of extension. The subject's pace during this trial was 88 steps per minute.

For the third trial, damping proportional to the angular velocity squared was provided for the prototype prosthesis using the controller. The subject was asked to walk while the proportionality constant was adjusted to what he felt was a comfortable setting. Data was then recorded and two representative cycles appear in Figure 5-4.

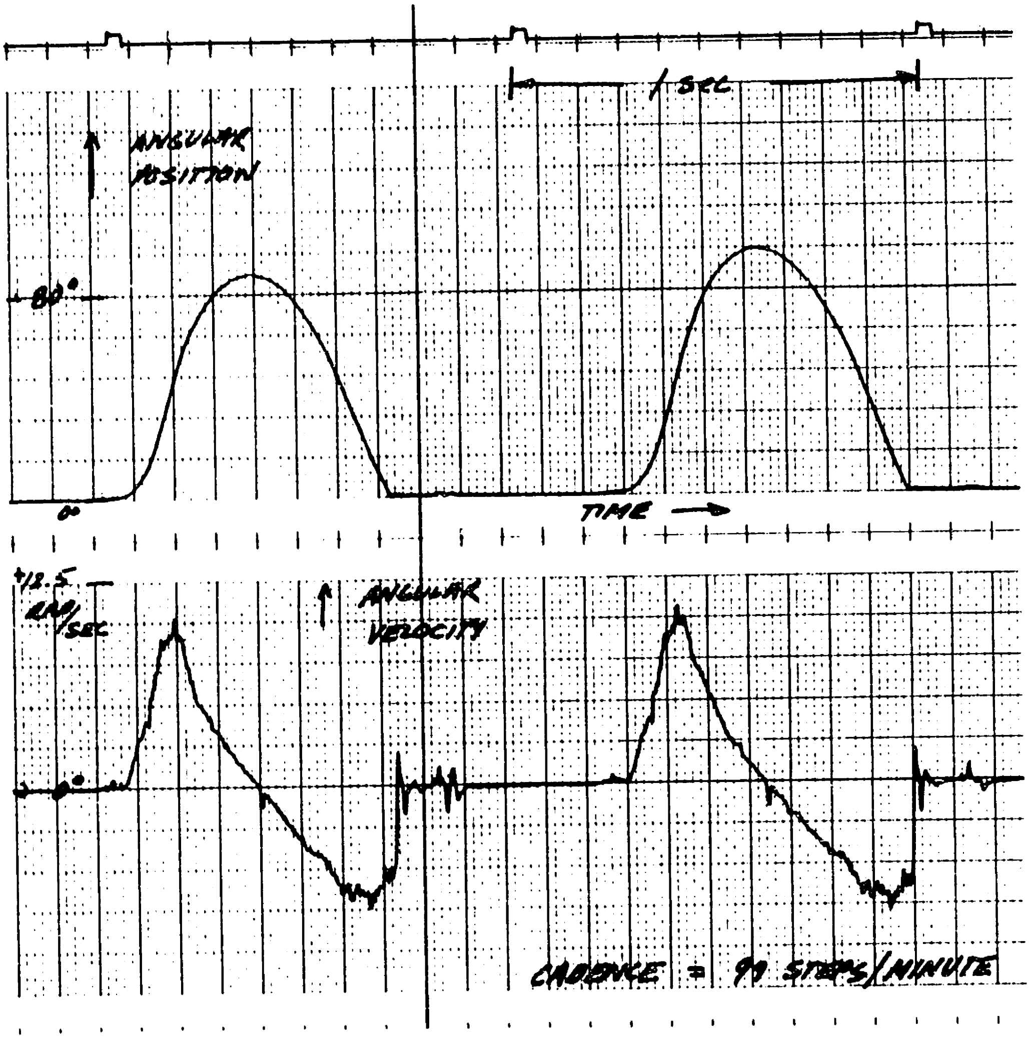


Figure 5-2 Knee Position and Velocity Data: Temporary Prosthesis

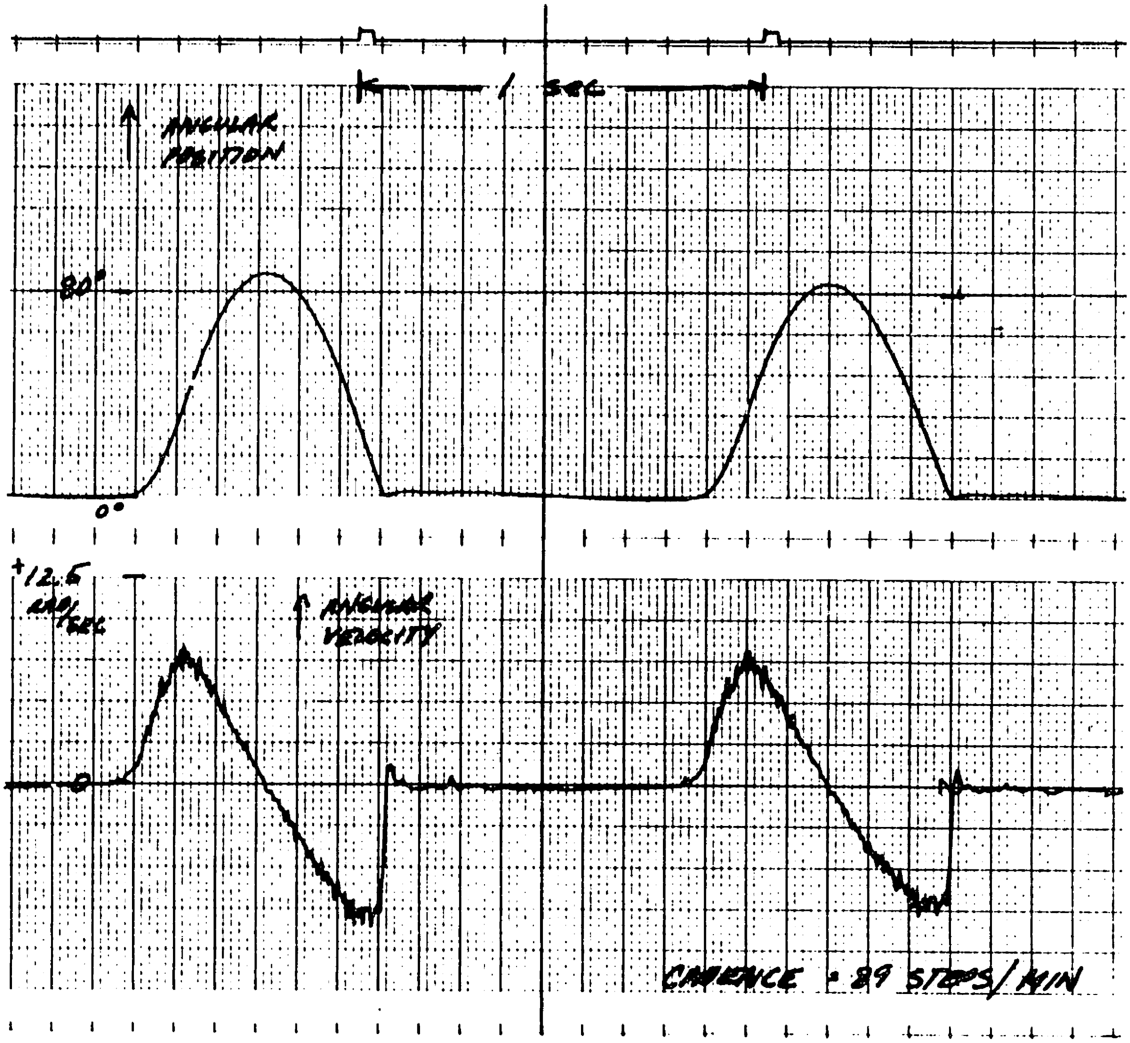


Figure 5-3 Knee Position and Velocity Data:
Prototype Prosthesis Operating in Free-swinging Mode

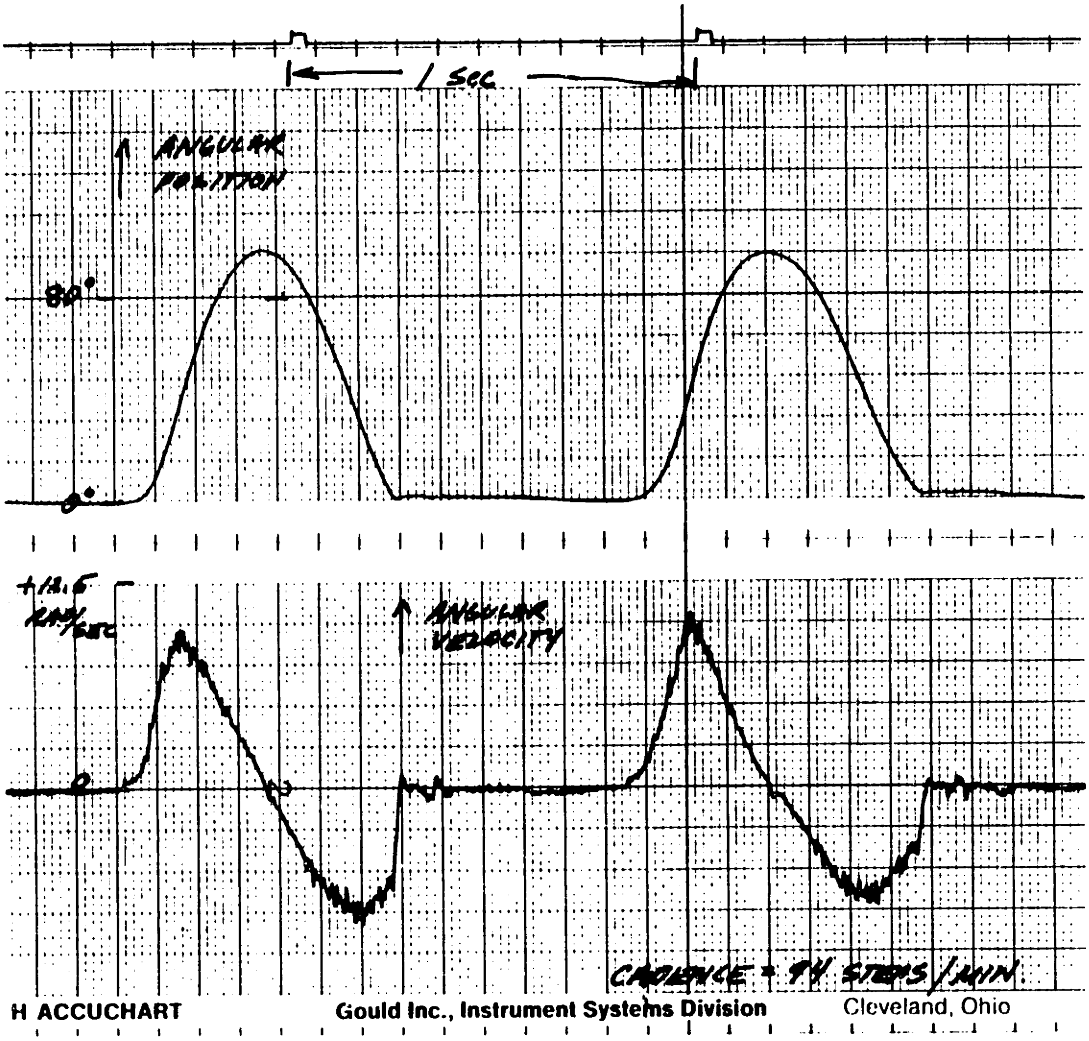


Figure 5-4 Knee Position and Velocity Data: Prototype Prosthesis with Proportional Velocity Squared Damping

The subject chose a proportinality constant of 0.54 in-lbs per $\sec^2$ per rad^2 for the damping and walked at a pace of 94 steps per minute. His reaction to this test was favorable because he said it felt similar to his permanent prosthesis. The velocity trace for this test differs from the previous ones in that the velocity during extension has begun to decrease before the prosthesis hits the hyperextension stop.

To investigate the full capabilities of the controller (i.e. using a profile for the damping rather than a constant) a series of four more tests were conducted. The objective during these tests was to adjust the damping profile to reduce the amount of heel rise and to minimize the velocity at the end of extension so an impact would not occur. The damping profile was modified empirically on the basis of the previous test. In the light of these goals, the third trial proved to be most successful. The results are shown in Figure 5-5. The maximum heel rise was approximately 84 degrees and the velocity approaches zero at the end of extension. However, the subject was somewhat unhappy with the feel of the prosthesis. He said it was difficult to flex and that it required a great deal of effort to make it fully extend before heel contact.

In addition to these tests, the controller and the prosthesis have been tested outside the controlled laboratory environment. On one occassion, the same subject walked approximately three fourths of a mile across the M.I.T. campus. This trip was made through long corridors and included stairs and ramps with a short distance outdoors. Several stops were made along the was to readjust the controller parameters.

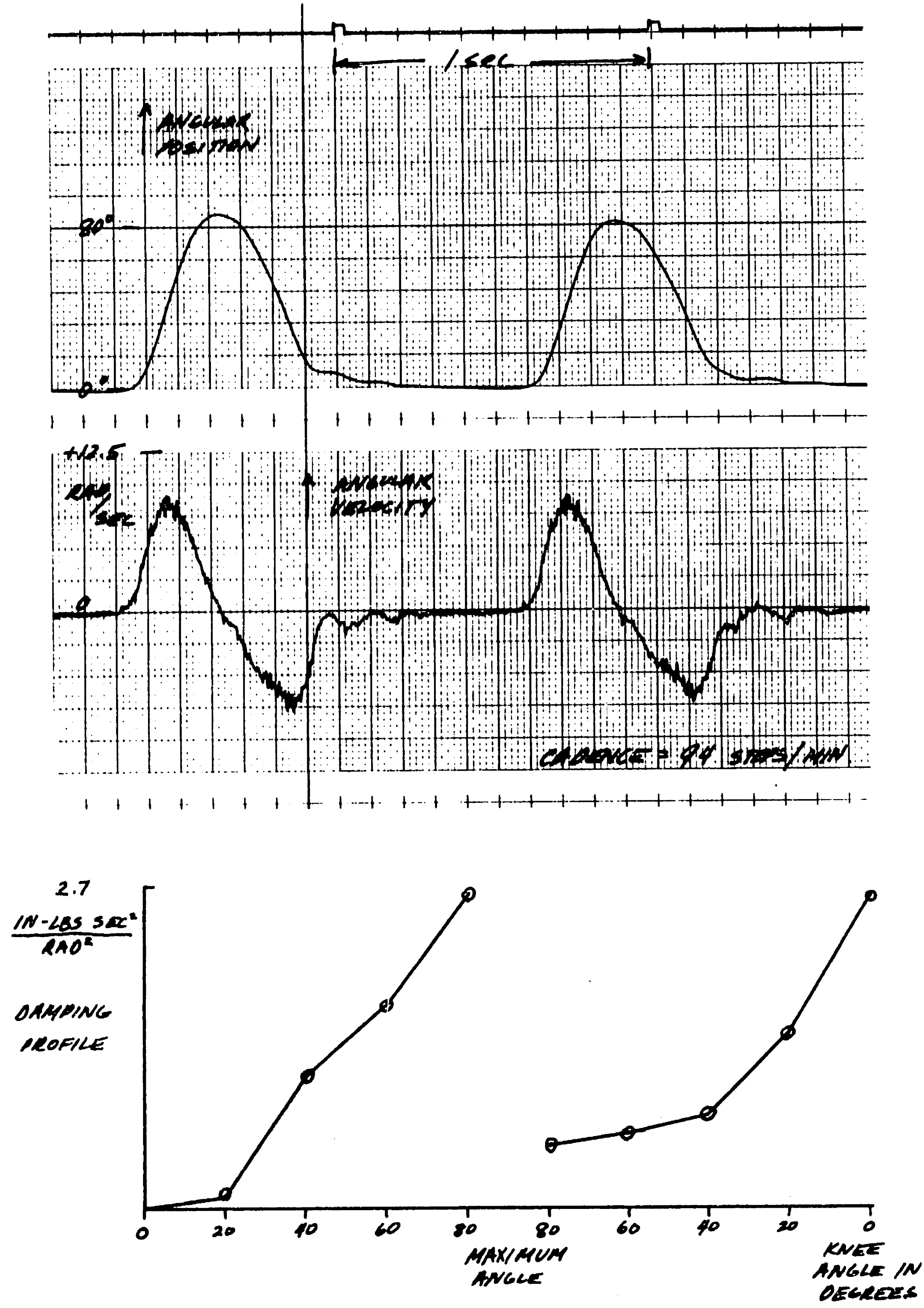


Figure 5-5 Knee Position and Velocity Data:
Prototype Prosthesis with a Turbulent Damping Profile

5-2 Conclusions

The results of the preliminary tests indicated that the controller and prosthesis can be used to simulate a wide range of prosthesis characteristics for a training application. The system has the additional benefit of providing the subject with a means to voluntarily lock the prosthesis on command. Having proved itself successful during the initial trials, arrangements were made to evaluate the system in a hospital environment using recent amputees. Tests were to be conducted during the patient's training session in parallel with the normal training procedure. However, no new amputees were available for experiments during the time provided for this work. Plans have been made to evaluate the system at a later date in the clinical environment.

One of the important factors missing from the preliminary evaluation is a judgement as to how the controller has affected the amputee's overall gait pattern. Hopefully, experiments undertaken in the clinical environment will provide an opportunity for a qualitative judgement by doctors and physical therapists regarding the system's usefulness.

At the present time the microprocessor is operating well below its capacity. Experiments with a previous simulator system have shown that a 60 Hz update rate is sufficient to maintain a continuous response of the prosthesis. The current control program operates at five times that speed, therefore a much more complex control scheme could be used by simply changing the control program. As research progresses in the definition of control schemes for a future multi-mode prosthesis, it should be possible to utilize a microprocessor in the implementation of those functions.

5.3 Recommendations

Several suggestions can be made that would improve the capabilities of the prosthesis-controller system. The possibility of measuring the torque about the knee axis should be pursued so that a more complete investigation may be made into the prosthesis dynamics and to enable closed loop control of the knee torque. Also, the present switching amplifier could be modified to take advantage of the full voltage available for changing the current in the magnetic particle brake. An improved version could sense the current through the brake and use a simple "on-off" control until the current is within a pre-specified tolerance band. At this point, the present switching scheme could be used to maintain the current within the tolerance band.

A number of possiblities exist for other uses of the controller in addition to its role as a training device. During its development and testing it often seemed that the function of the controller was limited only by the surface area of its container available for mounting connectors. However, one must be careful not to attempt functions that may be more appropriately studied with the use of a stationary computer.

Many of the applications within the realm of possiblity involve the use of the serial input/output port. This port, in conjunction with a portable cassette recorder, offers an excellent opportunity for development of a portable data aquisition system. If data compression or averaging techniques are used, the controller alone could serve as a storage device for cummulative totals of gait parameters and prosthesis usage or for isolated events during its operation. The serial port also could be used to establish communication with a larger computer system so that direct data transfer and modification of the controller parameters may take place.

Because of its portable nature, the controller is also well-suited to investigation of control laws to make the prosthesis more responsive to changes in cadence. Tests conducted both inside and outside the laboratory setting have suggested that a longer distance may be required to reach a "steady-state" cadence.

Regardless of the ultimate path taken, the microprocessor should prove itself valuable for the control of a future generation of above-knee prostheses.

REFERENCES

1. Bard, G. and Ralston, H.J., "Measurement of Energy Expenditure During Ambulation with Special Reference to Evaluation of Assistive Devices," Archives of Physical Medicine and Rehabilitation, October 1959.

2. Bresler, B. and Berry, F.R., "Energy and Power in the Leg During Normal Level Walking," University of California, Berkeley, Series III, Issue 15, Reprint March 1953.

3. Darling, D.T., "Automatic Damping Profile Optimization for Computer Controlled Above-Knee Prostheses," S.M. Thesis, M.I.T. May 1978.

4. Donath,M., "Proportional EMG Control for Above-Knee Prostheses," S.M. Thesis, M.I.T., August 1974.

5. Flowers, W.C., "A Man-Interactive Simulator System for Above-Knee Prostheses Studies," Ph.D. Thesis, M.I.T., August 1972.

6. Grimes, D.L., "Stance Phase Control for A/K Prostheses: A Preliminary Study," S.M. Thesis, M.I.T., December 1975.

7. Klopsteg, P.E. and Wilson, P.D., et al., Human Limbs and Their Substitutes, New York, Hafner Publishing Co., 1968.

8. Lampe, D.R., "Design of a Magnetic Particle Brake Above-Knee Prosthesis Simulator System," S.M. Thesis, M.I.T., February 1976.

9. Luckman, J. and Sorenson, K., Medical Surgical Nursing, A Psychophysiologic Approach, Philadelphia, W.B. Saunders Co., 1974, pp. 834 - 846.

10. Mauch, H.A., "The Application of Engineering Technology to the Simulation of Human Motions," Annals of the New York Academy of Science, Vol. 74, Sept. 30,1958, pp. 5 - 13.

11. May, D.R.W., "The Development of a Comprehensive External Prosthetic Knee Control," Proceedings of the Conference on Human Locomotion Engineering, December 1969, pp. 573 - 580.

12. McCollough, N.C., et al., "Some Considerations in the Management of the Above-Knee Geriatric Amputee," Artificial Limbs, Vol. 12, No. 2, Autumn 1958, pp. 28 - 35.

13. Murphy, E.F., "The Swing Phase of Walking with Above-Knee Prostheses," Bulletin of Prosthetic Research, BPR 10-1, Spring 1964.

14. Staros, A. and Murphy E.F., "Properties of Fluid Flow Applied to Above-Knee Prostheses," Bulletin of Prosthetic Research, Spring 1964.

15. Swan, G.S., "The Elderly Amputee," The Lancet, Oct. 28, 1972, p. 926.

16. Traugh, G.H., et al., "Energy Expenditure of Ambulation in Patients with Above-Knee Amputations," Archives of Physical Medicine and Rehabilitation, Vol. 56, February 1975.

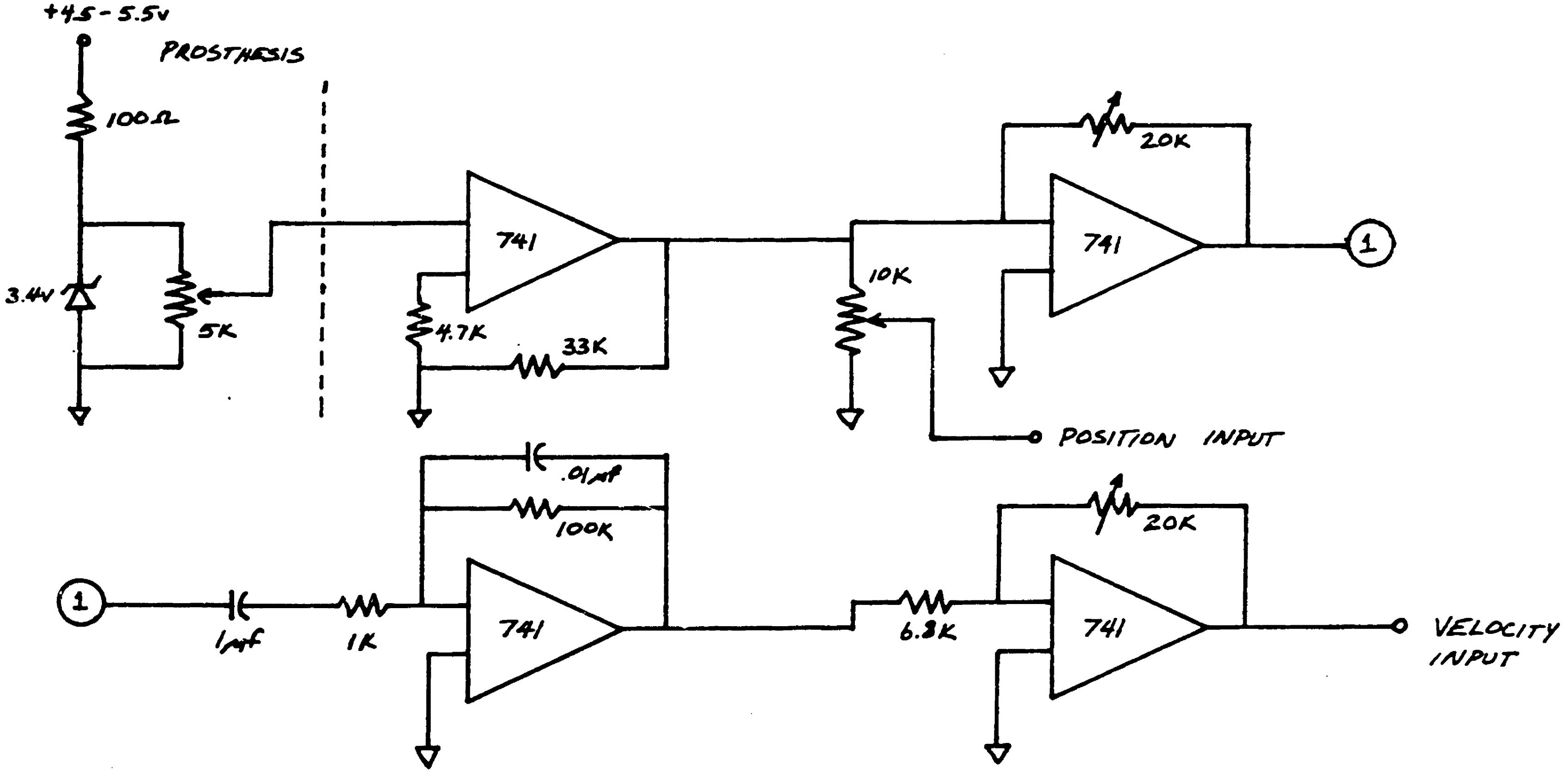
17. Veterans Adminstration, "Selection and Application of Knee Mechanisms," Bulletin of Prosthetic Research, BPR 10-18, Fall 1972.

18. Weiss, Marion, et al., Myloplastic Amputation, Immediate Prostheses and Early Ambulation, Bethesda, National Institute of Health, 1971.

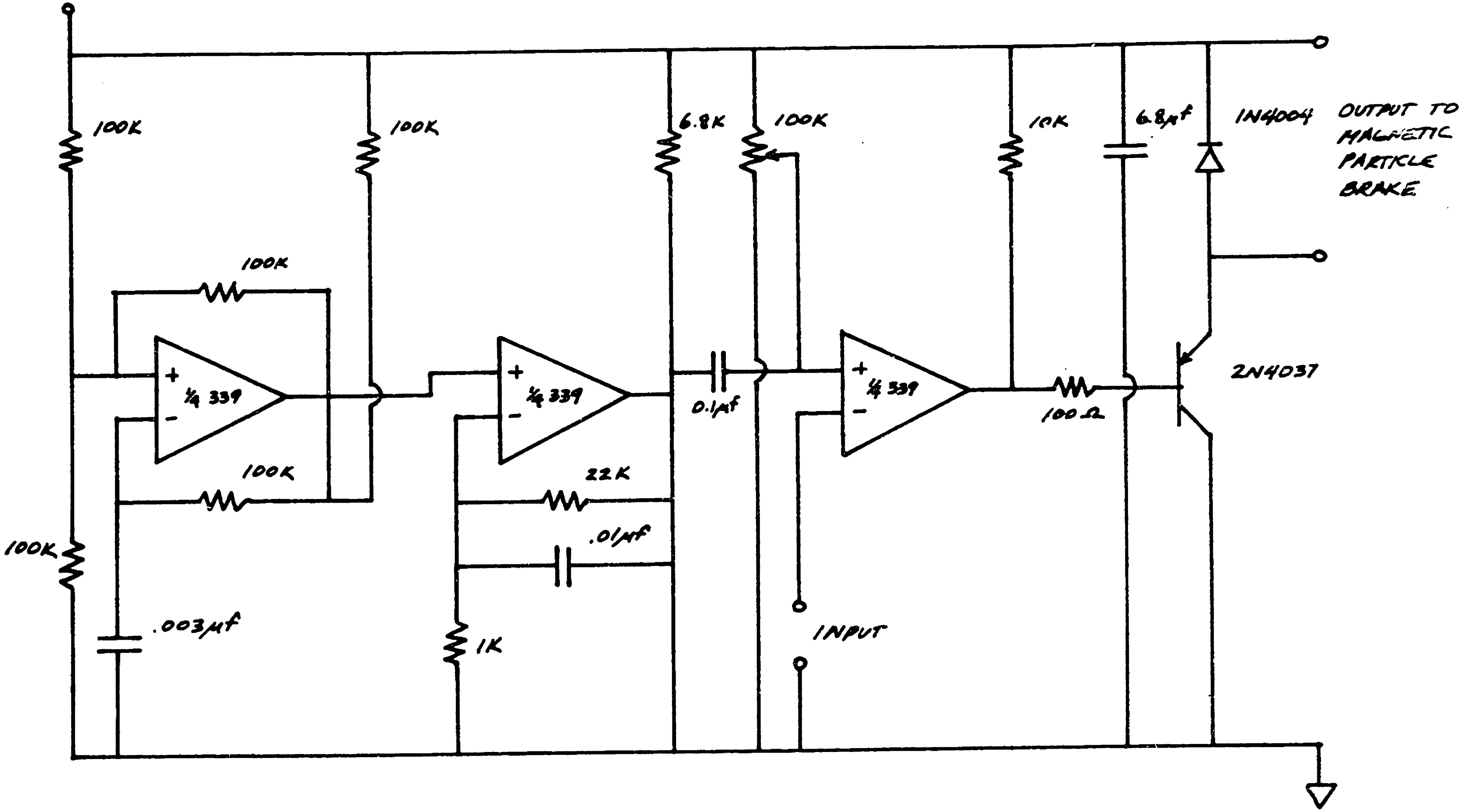
19. Wilson, A.B., "Recent Advances in Above-Knee Prosthetics", Artificial Limbs, Vol. 12, No. 2, 1968, pp. 1-27.

APPENDIX A MICROPROCESSOR SYSTEM MEMORY MAP

Hexidecimal Address	Device
0000-007F	Random Access Memory
ED80-EDFF	Random Access Memory
EE08	ACIA Control/Status Register
EE09	ACIA Data Register
EE10	PIA-1 Output Register A
EE11	PIA-1 Output Register B
EE12	PIA-1 Control Register A
EE13	PIA-1 Control Register B
EE20	PIA-2 Output Register A
EE21	PIA-2 Output Register B
EE22	PIA-2 Control Register A
EE23	PIA-2 Control Register B
FC00-FFFF	Erasable Programmable Read Only Memory



APPENDIX B ANALOG INPUT CIRCUITRY



APPENDIX C <u>SCHEMATIC FOR SWITCHING AMPLIFIER</u>

APPENDIX D:

CONTROLLER PROGRAM LISTING

PRINT ON THE FOLLOWING PAGES IS PARTIALLY ILLEGIBLE

```
PROGRAM FOR BURNING ROM          JUN 13, 1978    X6800-V1D    PAGE    1

                        ;THIS IS THE PROGRAM THAT RESIDES IN THE EPROM
                        ;IT CONSISTS OF TWO PARTS.  THE FIRST PART IS
                        ;A PROGRAM TO MAKE THE MICROPROCESSOR BEHAVE
                        ;LIKE A REAL-TIME CONTROLLER. THE SECOND PART
                        ;ACTS AS A KEYBOARD MONITOR FOR CHECKING
                        ;THE CONTROLLER PARAMETERS AND CHECKING THE
                        ;MEMORY LOCATIONS.
                        ;THIS PROGRAM WAS WRITTEN OVER A LONG PERIOD
                        ;OF TIME FROM JULY 1977 TO APRIL 1978.
                        ;
      0   0                     .ORG     0000
   0  0                 KFLEX   .ZERO    21        ;RESERVE 33 BYTES FOR THE FLEXION TABLE
  21  0                 KEXTEN  .ZERO    21        ;33 BECAUSE 4*7+5
  42  2                 CHANEL  .BYTE    2
  43  0                 SWIVAL  .BYTE    0
  44  0                 COUNT   .BYTE    0
  45  0                 SLYCNT  .BYTE    0
  46  0                 THETA   .BYTE    0
  47  0                 OMEGA   .BYTE    0
  48  0                 Y       .ZERO    2
  4A  0                 Z       .ZERO    2
  4C  0                 ANSWER  .ZERO    2
  4E  0                 PCINTR  .ZERO    2
                        FLAG    .ZERO    0
     EE 10              DACUT   .EQU     0EE10
     EE 11              SENSIN  .EQU     0EE11
     EE 12              CRA1    .EQU     0EE12
     EE 13              CRB1    .EQU     0EE13
     EE 20              ADINPT  .EQU     0EE20
     EE 21              MUXCTL  .EQU     0EE21
     EE 22              CRA2    .EQU     0EE22
     EE 23              CRB2    .EQU     0EE23
      0 48              XHI     .EQU     Y
      0 49              XLOW    .EQU     Y+1
     EE  8              ACIACS  .EQU     0EE08
     EE  8              ACIACR  .EQU     0EE08
     EE  9              ACIADR  .EQU     0EE09
     ED C4              STACK   .EQU     0EDC4
                        ;
                        ;ON WITH THE SHOW....
```

```
PROGRAM FOR BURNING ROM          JUN 13, 1978    X6800-V1D   PAGE    2

     FC  0                .ORG    0FC20
 FC00 8E ED C4            LDS     #STACK  ;INTIALIZE STACK POINTER
                 ;FIRST INTIALIZE ALL PIA'S  HIGH MEANS AN OUTPUT LINE
 FC03 86 FF               LDA A   #0FF
 FC05 B7 EE 10            STA A   DAOUT   ;A1 IS OTPUT TO THE D TO A
 FC08 86 1F               LDA A   #1F
 FC0A B7 EE 21            STA A   MUXCTL  ;B2 IS CHANNEL OUTPUT TO THE MUX
 FC0D 86  4               LDA A   #04
 FC0F B7 EE 12            STA A   CRA1    ;SET UP CONTROL REGISTERS
 FC12 B7 EE 13            STA A   CRB1
 FC15 B7 EE 22            STA A   CRA2
 FC18 B7 EE 23            STA A   CRB2
                 ;
                 ;NOW GET DOWN TO BUSINESS AND INTIALIZE THINGS
                 ;
 FC1B CE  0  0            LDX     #0000   ;INTIALIZE THE INDEX REGISTER
 FC1E 86  2               LDA A   #2
 FC20 97 42               STA A   CHANEL
 FC22 7F  0 4E            CLR     POINTR
 FC25 7F  0 43            CLR     SWIVAL
 FC28 BD FC B0            JSR     SLYPOT
 FC2B BD FC B0            JSR     SLYPOT  ;INTIALIZE THE DAMPING TABLES
                 ;NOW CALL SUBROUTINE TO CORRECT THE DAMPING TABLES
                 ;ACCORDING TO THE NON-LINEARITIES OF THE TORQUE
                 ;LEVER ARM
 FC2E CE  0  0            LDX     #0000
 FC31 BD FE A1            JSR     DAMFIX
 FC34 CE  0 21            LDX     #0021
 FC37 BD FE A1            JSR     DAMFIX
 FC3A 86  1               LDA A   #01
 FC3C B7 EE  8            STA A   ACIACR
                 ;THAT SHOULD BE IT FOR INITIALIZATION
                 ;THIS IS THE MAIN PROGRAM THAT RUNS CONTINUOUSLY

 FC3F 8E FD C4 MAIN       LDS     #STACK  ;INTIALIZE THE STACK POINTER
 FC42 B6 EE 11            LDA A   SENSIN  ;CHECK PIA SENSE LINES
 FC45 2B  3               BMI     OK      ;IF BIT 7 IS HIGH GO IN TO NORMAL
                                          ;OPERATING MODE
 FC47 7E FE E6            JMP     DECODE  ;IF LOW GO DECIDE WHICH MONITER MODE
 FC4A 86  1      OK       LDA A   #01     ;GO GET THE VELOCITY
 FC4C 97 42               STA A   CHANEL
 FC4E BD FC 97            JSR     VALGET
                 ;THE AMPLIFIER FOR THE VELOCITY HAS AN INVERTING
                 ;AMPLIFIER ON ITS LAST STAGE SO ITS NECESSARY TO
                 ;CHANGE THE SIGN ON THE VELOCITY TO GET THE
                 ;CORRECT SIGN FOR DETERMINING WHETHER IT IS
                 ;DURING FLEXION OR EXTENSION. WHEN DOING THIS WE
                 ;MUST BE CAREFUL FOR THE VALUE 80 (-128) WHICH
                 ;THE NEGATE COMMAND DOES NOT HANDLE PROPERLY
                 ;WHEN 80 OCCURS THE INVERSE WILL BE 7F
 FC51 C1 80               CMP B   #80     ;CHECK FOR V =80
 FC53 26  4               BNE     INVERT  ;IF NOT INVERT AS NORMAL
 FC55 C6 7F               LDA B   #7F     ;OTHERWISE USE 7F FOR THE INVERSE
 FC57 20  1               BRA     SAVVEL
 FC59 50         INVERT   NEG B
 FC5A D7 47      SAVVEL   STA B   OMEGA   ;SAVE THE VELOCITY FOR FUTURE USE
 FC5C D7 48               STA B   Y       ;GET READY FOR MULTIPLY
 FC5E D7 4A               STA B   Z
 FC60 7F  0 49            CLR     Y+1
 FC63 7F  0 4B            CLR     Z+1     ;PUT IN ZERO EXPONENTS
```

```
PROGRAM FOR BURNING ROM          JUN 13, 1978    X6800-V1D   PAGE    3

 FC66 BD FC F8            JSR      MLTPY   ;USE THE MULTIPLY ROUTINE
 FC69 96 4C               LDA A    ANSWER
 FC6B D6 4D               LDA B    ANSWER+1         ;PUT THE ANSWER BACK
                                           ;INTO THE MULTIPLY SPOTS
 FC6D 97 48               STA A    Y
 FC6F D7 49               STA B    Y+1
 FC71 7F  0 42            CLR      CHANEL  ;GO GET ANGLE
 FC74 BD FC 97            JSR      VALGET
 FC77 BD FC AB            JSR      FIX     ;IF ITS <0 MAKE IT ZERO
 FC7A D7 46               STA B    THETA   ;SAVE FOR FUTURE REFERENCE
 FC7C 54                  LSR B            ;DIVIDE THE ANGLE BY 4
 FC7D 54                  LSR B            ;TO GET THE OFFSET FOR TABLE
 FC7E 96 47               LDA A    OMEGA   ;IF POSITVE USE THE KFLEX TABLE
 FC80 2C  2               BGE      PLUS    ;IF NEGATIVE ADD 32 TO X TO USE
 FC82 CB 21               ADD B    #21     ;KEXTEN TABLE
 FC84 D7 4F     PLUS      STA B    POINTR+1         ;LOAD THE OFFSET INTO
                                           ;THE POINTER
 FC86 DE 4E               LDX      POINTR  ;PUT TABLE POINTER IN INDEX REGISTER
 FC88 E6  0               LDA B    0,X     ;GET DAMPING VALUE
 FC8A D7 4A               STA B    Z       ;PUT VALUE IN PLACE FOR MULTIPLY
 FC8C 7F  0 4B            CLR      Z+1     ;PUT IN A ZERO EXPONENT
 FC8F BD FC F8            JSR      MLTPY   ;MULTIPLY K TIMES V**2
 FC92 BD FD 5E            JSR      SCALE
 FC95 20 A8               BRA      MAIN    ;CONTINUE IN PROGRAM LOOP
                ;
                ;THESE ARE ALL THE SUBROUTINES
                ;
                ;VALGET RETRIEVES VALUES FROM THE A TO D CONVERTER
                ;AND TAKES CARE OF ALL THE TIMING INFO
                ;THE CHANEL TO BE CONVERTED SHOULD BE PLACED
                ;IN THE LOCATION CALLED CHANEL
                ;THE VALUE WILL BE RETURNED IN ACC B
                ;IN THE PROPER 2'S COMPLEMENT FORM
 FC97 96 42     VALGET    LDA A    CHANEL  ;GET THE CHANEL #
 FC99 8B 10               ADD A    #10     ;TAKE THE CONVERT LINE HIGH
 FC9B B7 EE 21            STA A    MUXCTL
 FC9E 80 10               SUB A    #10     ;TAKE THE CONVERT LINE LOW
 FCA0 B7 EE 21            STA A    MUXCTL
 FCA3 7C  0 42            INC      CHANEL  ;INCREMENT THE CHANEL POINTER
 FCA6 F6 EE 20            LDA B    ADINPT  ;PUT THE RESULT IN ACC B
 FCA9 53                  COM B            ;FIX UP THE CTC STUFF
 FCAA 39                  RTS
                ;
                ;THIS IS A SUBROUTINE THAT CHECKS TO SEE IF THE
                ;VALUE IN ACC B IS NEGATIVE.  IF IT IS IT WILL
                ;MAKE IT = TO 0.  THIS IS FOR USE WITH THE A TO D
                ;FOR SMALL VALUES TO GET RID OF NOISE
                ;
 FCAB 5D        FIX       TST B    ;SET SIGN BIT IF NECESSARY
 FCAC 2C  1               BGE      HENRY
 FCAE 5F                  CLR B
 FCAF 39        HENRY     RTS
                ;
                ;THIS SUBROUTINE IS FOR READING 5 CONSECUTIVE SLIDE POT VALUES
                ;THE STARTING POT IS PASSED VIA THE LOCATION CHANEL
                ;(NOTE SLIDE POT #1 CORRESPONDS TO ADDRESS 2 ON THE MUX)
                ;THE RESULTS WILL BE STORED IN A TABLE THE STARTING
                ;ADDRESS OF THE TABLE SHOULD BE POINTED TO BY THE
                ;INDEX REGISTER.
                ;THE SUBROUTINE WILL AUTOMATICALLY INCREMENT THE
```

```
                      ;CHANEL AND INDEX REGISTER AFTER IT IS DONE WITH EACH POT
                      ;
 FCB0 86 FC           SLYPOT  LDA A   #-4      ;SET COUNTER FOR # OF POTS TO BE READ
                                               ;AFTER THE FIRST ONE.
 FCB2 97 45                   STA A   SLYCNT
 FCB4 BD FC 97                JSR     VALGET   ;GET THE FIRST VALUE
 FCB7 BD FC AB                JSR     FIX      ;CLEAN IT UP
 FCBA D7 43                   STA B   SWIVAL   ;STORE IT FOR THE FUTURE
 FCBC BD FC 97 MORPOT         JSR     VALGET   ;GET SECOND VALUE
 FCBF BD FC AB                JSR     FIX
 FCC2 17                      TBA              ;PUT THE NEW VALUE IN ACC A
 FCC3 D6 43                   LDA B   SWIVAL   ;PUT OLD VALUE IN ACC B
 FCC5 10                      SBA              ;FIND THE DIFFERENCE IN SETTINGS
 FCC6 47                      ASR A            ;DIVIDE BY 8
 FCC7 47                      ASR A            ;TO DO IT SHIFT RIGHT
 FCC8 47                      ASR A            ;PROPOGATING THE SIGN BIT
 FCC9 C6 F8                   LDA B   #-8      ;INTIALIZE LOOP COUNTER
 FCCB D7 44                   STA B   COUNT
 FCCD 16                      TAB              ;PUT THE DIFFERENCE IN ACC B
 FCCE 96 43                   LDA A   SWIVAL   ;GET THE OLD SWIVAL
 FCD0 4D              MORVAL  TST A            ;CHECK IF THE VALUE IS NEGATIVE
 FCD1 2C  1                   BGE     FINE     ;IF NOT GO AHEAD AS PLANNED
 FCD3 4F                      CLR A            ;IF IT IS NEGATIVE PUT A ZERO THERE
 FCD4 A7  0           FINE    STA A   0,X      ;PUT FIXED UP VALUE IN THE TABLE
 FCD6  8                      INX              ;MOVE THE TABLE POINTER
 FCD7 1B                      ABA              ;ADD THE SLOPE TO THE INITIAL VALUE
 FCD8 7C  0 44                INC     COUNT    ;ADD ONE TO THE LOOP COUNTER
 FCDB 2D F3                   BLT     MORVAL   ;IF NOT DONE FILL IN MORE POINTS
 FCDD 7C  0 45                INC     SLYCNT   ;ADD ONE TO # OF POTS CONVERTED
 FCE0 2C  B                   BGE     ENDPOT
 FCE2 F6 EE 20                LDA B   ADINPT   ;PUT THE NEW VALUE IN THE OLD VALUE SPO
                                               T
 FCE5 53                      COM B
 FCE6 BD FC AB                JSR     FIX      ;CHECK TO SEE IF ITS NEGATIVE
 FCE9 D7 43                   STA B   SWIVAL
 FCEB 20 CF                   BRA     MORPOT
 FCED F6 EE 20 ENDPOT         LDA B   ADINPT   ;PUT THE FINAL VALUE IN THE TABLE
 FCF0 53                      COM B            ;CLEAN UP THE CTC
 FCF1 BD FC AB                JSR     FIX      ;CHECK FOR A NEGATIVE VALUE
 FCF4 E7  0                   STA B   0,X
 FCF6  8                      INX              ;MOVE THE TABLE POINTER
 FCF7 39                      RTS
                      ;THIS IS AN 8 BIT MULTIPLY ROUTINE
                      ;NOT TO BE CONFUSED WITH A 2 BIT -----
                      ;IT MULTIPLIES TWO 8 BIT TWO'S COMPLEMENT
                      ;NUMBERS AND ADDS THEIR EXPONENTS
                      ;MUCH OF THIS HAS BEEN LIFTED FROM THE 6800
                      ;APPLICATIONS MANUAL AND FOR THE MOST PART
                      ;USES BOOTH'S ALGORITHM AS SUGGESTED THERE
                      ; Y = MULTIPLIER
                      ; Y+1 = MULTIPLIER'S EXPONENT
                      ; Z = MULTIPLICAND
                      ; Z+1 = MULTIPLICAND'S EXPONENT
                      ;THE MULTIPLIER WILL BE DEMOLISHED IN THE PROCESS
                      ;AND THE MULTIPLICAND WILL ESCAPE UNTOUCHED
                      ;
                      ;
 FCF8 96 4A           MLTPY   LDA A   Z        ;CHECK IF Z=80 (-128)
 FCFA 81 80                   CMP A   #80      ;IF IT IS OVERFLOW WILL RESULT
 FCFC 26  3                   BNE     NOPROB
```

```
PROGRAM FOR BURNING ROM          JUN 13, 1978    X6800-V1D   PAGE    5

 FCFE 4C                 INC A            ;IF Z=80 MAKE IT 81 WHICH WILL
                                          ;FIX THINGS
 FCFF 97 4A              STA A   Z
 FD01 96 49     NOPROB   LDA A   Y+1      ;ADD TOGETHER THE EXPONENTS
 FD03 9B 4B              ADD A   Z+1      ;SAVE THE RESULT WHERE THE
 FD05 97 49              STA A   Y+1      ;Y EXPONENT WAS
 FD07 7F  0 4C           CLR     ANSWER   ;CLEAR OUT THE WORK SPACES
 FD0A 7F  0 4D           CLR     ANSWER+1
 FD0D 7F  0 50           CLR     FLAG
 FD10 CE  0  8           LDX     #8       ;SET UP COUNTER FOR LOOP
 FD13 96 48     GO       LDA A   Y        ;PUT THE MULTIPLIER IN ACC A
 FD15 84  1              AND A   #1       ;MASK OFF TO FIND LSBIT
 FD17 16                 TAB              ;SAVE LSBIT IN ACC B
 FD18 98 50              EOR A   FLAG     ;CHECK FOR 1 TO 0 OR 0 TO 1
                                          ;TRANSISTION
 FD1A 27  F              BEQ     SHIFT    ;IF THE SAME GO TO THE SHIFT ROUTINE
 FD1C 5D                 TST B            ;DOES THE LSBIT=0?
 FD1D 27  6              BEQ     ADDEM    ;YES, ADD MULTIPLICAND TO ANSWER
 FD1F 96 4C              LDA A   ANSWER
 FD21 90 4A              SUB A   Z        ;NO, SUBTRACT MULTIPLICAND
 FD23 20  4              BRA     CONT     ;PUT THE RESULT BACK IN ANSWER
                                          ;GO TO THE SHIFT ROUTINE
 FD25 96 4C     ADDEM    LDA A   ANSWER   ;ADD THE MULTIPLICAND TO THE
 FD27 9B 4A              ADD A   Z        ;ANSWER
 FD29 97 4C     CONT     STA A   ANSWER   ;STORE THE RESULT IN ANSWER
 FD2B 7F  0 50 SHIFT     CLR     FLAG     ;CLEAR THE TEST FLAG
 FD2E 76  0 48           ROR     Y        ;SHIFT THE MULTIPLIER RIGHT ONE
                                          ;BIT SAVING THE LSBIT IN THE CARRY
 FD31 79  0 50           ROL     FLAG     ;PUT THE CARRY BIT OF THE LSBIT IN FLAG
 FD34 77  0 4C           ASR     ANSWER   ;SHIFT THE PRODUCT RIGHT ONE BIT
 FD37 76  0 4D           ROR     ANSWER+1;WITH THE MSBIT REMAINING THE SAME
 FD3A  9                 DEX              ;DECREMENT THE COUNTER
 FD3B 26 D6              BNE     GO       ;IF NOT DONE LOOP BACK
                ;THIS PART OF THE PROGRAM SHIFTS THE MANTISSA TO
                ;MAINTAIN THE MAXIMUM # OF SIGNIFICANT DIGITS
                ;IT WON'T SHIFT A # MORE THAN EIGHT DIGITS AND PADS
                ;THE RIGHT END WITH 0'S
 FD3D C6  0              LDA B   #0       ;SET UP LOOP COUNTER
 FD3F 96 4C     MORE     LDA A   ANSWER   ;LOAD MSBYTE OF ANSWER IN ACC A
 FD41 84 C0              AND A   #0C0     ;TEST 2 MSBITS FOR TWO ZEROS
 FD43 27  6              BEQ     SHIFTL   ;IF TYES, GO TO SHIFT LEFT
 FD45 88 C0              EOR A   #0C0     ;TEST 2 MSBITS FOR TWO ONES
 FD47 27  2              BEQ     SHIFTL   ;IF YES GO TO SHIFT LEFT
 FD49 20  B              BRA     DONE     ;IF NOT, DON'T TOUCH IT YOU'RE DONE
 FD4B 5C        SHIFTL   INC B            ;ADD ONE TO THE LOOP COUNTER TO KEEP
                                          ;TRACK OF THE # OF SHIFTS
 FD4C 78  0 4D           ASL     ANSWER+1;SHIFT LSBYTE LEFT AND SAVE CARRY
 FD4F 79  0 4C           ROL .   ANSWER   ;SHIFT MSBYTE LEFT PUT CARRY ON RIGHT
 FD52 C1  8              CMP B   #8       ;SEE IF THE # HAS BEEN SHIFTED 8 TIMES
 FD54 26 E9              BNE     MORE     ;IF NOT CONTINUE SEARCH PROCESS
 FD56 96 49     DONE     LDA A   Y+1      ;LOAD INTIAL EXPONENT IN ACC A
 FD58 10                 SBA              ;SUBTRACT OFF THE # OF SHIFTS LEFT

 FD59 8B  8              ADD A   #8       ;ADD 8 TO THE EXPONENT BECAUSE
                                          ;WE CUT OFF THE RIGHT 8 BITS
 FD5B 97 4D              STA A   ANSWER+1;PUT THE PROPER EXPONENT BACK
 FD5D 39        ATSALL   RTS
                ;THIS IS A SUBROUITNE THAT TAKES THE PRODUCT OF K*V**2
                ;AND SCALES IT INTO THE RANGE OF 0 TO 256 TO BE OUTPUT
                ;TO THE D TO A CONVERTER
```

```
                   ;IT ALSO CHECKS TO SEE IF THE RESULT IS EQUAL TO 0 OR
                   ;NEGATIVE IN WHICH CASE A ZERO OUTPUT IS SENT TO
                   ;THE CONVERTER
 FD5F 96 4C        SCALE    LDA A   ANSWER   ;GET MANTISSA
 FD60 2C  3                 BGE     GOODY    ;IF NEGATIVE DUMP 0
 FD62 4F                    CLR A
 FD63 20 35                 BRA     DUMP
 FD65 27 33        GOODY    BEQ     DUMP     ;IF ANSWER IS 0 , DUMP 0
 FD67 97 4A                 STA A   Z        ;PUT PRODUCT BACK IN MLTPY
 FD69 96 4D                 LDA A   ANSWER+1
 FD6B 97 4B                 STA A   Z+1
 FD6D 86  C                 LDA A   #0C
 FD6F 97 42                 STA A   CHANEL
 FD71 BD FC 97              JSR     VALGET   ;GO GET GAIN VALUE
 FD74 BD FC A5              JSR     FIX
 FD77 D7 48                 STA B   Y        ;PUT GAIN INTO MLTPY
 FD79 86 FC                 LDA A   #-4      ;LOAD IN EXP FOR GAIN
 FD7B 97 49                 STA A   Y+1
 FD7D BD FC F8              JSR     MLTPY
 FD80 96 4C                 LDA A   ANSWER   ;GET FINAL ANSWER
 FD82 D6 4D                 LDA B   ANSWER+1; GET EXP FOR ANSWER
 FD84 C0  E                 SUB B   #0E      ;DIVIDE BY 2**14 SO
                                             ;ANSWER WILL BE 0 TO 127
 FD86 27 12                 BEQ     DUMP     ;IF ZERO EXP GO DUMP
 FD88 2E  8                 BGT     RGHSHT   ;IF EXP GT 0 GO SHIFT RIGHT
                   ;ANSWER HAS NEGATIVE EXP SO SHIFT LEFT AND CHECK
                   ;FOR ZERO CASE IF SHIFTED TOO FAR.
 FD8A 44           LFTSFT   LSR A
 FD8B 27  D                 BEQ     DUMP     ;IF NOTHING LEFT GO DUMP 0
 FD8D 5C                    INC B
 FD8E 27  A                 BEQ     DUMP     ;IF EXP NOW 0 GO DUMP
 FD90 20 F8                 BRA     LFTSHT
                   ;ANSWER STILL HAS POSITIVE EXP SO SHIFT RIGHT
 FD92 48           RGHSHT   ASL A            ;SHIFT RIGHT
 FD93 2B  A                 BMI     FFDUMP   ;IF ANSWER CHANGES SIGN WE OVERFLOWED
                                             ;SO DUMP FF OUT
 FD95 5A                    DEC B            ;DECREASE EXP
 FD96 27  2                 BEQ     DUMP     ;IF EXP 0 GO DUMP
 FD98 20 F8                 BRA     RGHSHT
 FD9A 48           DUMP     ASL A            ;MAKE INTO 0 TO 256
 FD9B B7 EE 10              STA A   DAOUT
 FD9E 39                    RTS
 FD9F 86 FF        FFDUMP   LDA A   #0FF     ;OVERFLOWED, DUMP MAX VALUE
 FDA1 B7 EE 10              STA A   DAOUT
 FDA4 39                    RTS
                   ;THIS PART OF THE PROGRAM CONTAINS THE
                   ;SUBROUTINES USED IN THE MONITOR
                   ;
                   ;THE SUBROUTINES NEEDED ARE
                   ;FOR THE MOST PART COPIED FROM THE MOTOROLA
                   ;MINIBUG PROGRAM.
                   ;
                   ;BUILD ADDRESS
                   ;TAKES IN ENTRY FROM KEYBOARD TO FORM  A HEX ADDRESS
                   ;LEAVES ADDRESS IN INDEX REGISTER.
                   ;THIS TAKES IN 4 NUMBERS
                   ;
 FDA5 8D  9        BADDR    BSR     BYTE     ;READ IN MSB
 FDA7 97 48                 STA A   XHI
 FDA9 8D  5                 BSR     BYTE     ;READ IN LSB
```

```
PROGRAM FOR BURNING ROM          JUN 13, 1978   X6800-V1D   PAGE    7

 FDAB 97 49              STA A   XLOW
 FDAD DE 48              LDX     XHI      ;PUT ADDRESS IN INDEX REGISTER
 FDAF 39                 RTS
                 ;INPUT BYTE (TWO FRAMES)
                 ;LEAVES BYTE IN ACC A
                 ;
 FDB0 BD FE 37  BYTE     JSR     INCH     ;GO GET A CHARACTER
 FDB3 81 1B              CMP A   #1B      ;IF ITS AN ESC GO BACK TO CONTROL MODE
 FDB5 26  3              BNE     CRCHEK
 FDB7 7E FF 11           JMP     CONTRL
 FDBA 81  D     CRCHEK   CMP A   #0D      ;CHECK FOR A CARRIAGE RETURN
 FDBC 26  4              BNE     NORMAL
 FDBE 7C  0 50           INC     FLAG     ;SET CR FLAG
 FDC1 39                 RTS              ;IF IT IS A CR GO BACK TO THE MEMORY
                                          ;CHANGE PROGRAM
 FDC2 BD FE 4B  NORMAL   JSR     INHEX1   ;WE ALREADY HAVE THE CHARACTER
 FDC5 48                 ASL A            ;LEFT JUSTIFY FIRST PART OF BYTE
 FDC6 48                 ASL A
 FDC7 48                 ASL A
 FDC8 48                 ASL A
 FDC9 16                 TAB              ;SAVE IN ACC B
 FDCA 8D 7D              BSR     INHEX
 FDCC 84  F              AND A   #0F      ;MASK OFF LEFT BITS
 FDCE 1B                 ABA              ;ADD THEM TOGETHER
 FDCF 39                 RTS              ;LEAVE THE RESULT IN ACC A
                 ;
                 ;
                 ;MEMORY DISPLAY AND CHANGE COMMAND
                 ;
 FDD0 8D D3     CHANGE   BSR     BADDR    ;GET ADDRESS OF LOCATION
 FDD2 8D 5B     CHANG1   BSR     OUTCR
 FDD4 8D 5D              BSR     OUTLF    ;GO TO THE NEXT LINE
 FDD6 8D 17              BSR     PRADDR   ;REPRINT THE ADDRESS
 FDD8 8D 4A              BSR     OUTS     ;PRINT A SPACE
 FDDA 8D 46              BSR     OUT2HS   ;PRINT WHATS AT THE ADDRESS
 FDDC 7F  0 50           CLR     FLAG     ;CLEAR FLAG FOR CR
 FDDF 8D CF              BSR     BYTE     ;FIND OUT WHAT USER WANTS NEXT
 FDE1 7D  0 50  NEXONE   TST     FLAG     ;CHECK FLAG FOR CR
 FDE4 2E  6              BGT     CHANG2   ;GO LOOK AT THE NEXT ADDRESS
                                          ;IF IT WASN'T A CR HE WANTS
                                          ;TO CHANGE THE CURRENT ADDRESS
 FDE6 A7  0              STA A   X        ;STORE AT THE ADDRESS WHAT HE TYPED IN
 FDE8 A1  0              CMP A   X        ;CHECK TO SEE IF THE MEMORY CHANGED
 FDEA 26 3C              BNE     WHAT     ;IF IT DIDN'T CHANGE PRINT A ?
                                          ;AND RETURN TO THE CONTROL PORGRAM

 FDEC  8        CHANG2   INX              ;MOVE THE INDEX REGISTER TO POINT
 FDED 20 E3              BRA     CHANG1   ;AT THE NEXT MEMORY LOCATION
                 ;
                 ;SUBROUTINE FOR PRINTING ADDRESSES,
                 ;THE ADDRESS SHOULD BE LEFT IN THE
                 ;INDEX REGISTER

                 ;
 FDEF DF 48     PRADDR   STX     Y        ;PUT THE CONTENTS OF THE INDEX REGISTER
                                          ;IN MEMORY
 FDF1 CE  0 48           LDX     #Y       ;NOW TRY TO PRINT WHAT WAS IN THE
                                          ;INDEX REGISTER
 FDF4 8D 23              BSR     OUT2H
 FDF6 CE  0 49           LDX     #Y+1
```

```
FDF9 8D 1E               BSR     OUT2H
FDFB DE 48               LDX     Y          ;RESTORE THE INDEX REGISTER
FDFD 39                  RTS
                 ;THESE ROUTINES PRINT THE RIGHT AND LEFT HALVES
                 ;OF A SINGLE BYTE OUT

FDFE 44          OUTHL   LSR A              ;PRINT OUT THE 4 MSB
FDFF 44                  LSR A              ;BY RIGHT JUSTIFYING IT IN ACC A
FE00 44                  LSR A
FE01 44                  LSR A
FE02 84  F       OUTHR   AND A   #0F        ;MASK OUT LEFT 4 BITS
FE04 8B 30               ADD A   #30        ;ADD 30 TO CONVERT HEX TO ASCII
FE06 81 39               CMP A   #39        ;IF ITS A # < 9 PRINT IT
FE08 23  2               BLS     OUTCH
FE0A 8B  7               ADD A   #07        ;OTHERWISE ADD 7 TO GET PROPER
                                            ;ASCII FOR A - F
                 ;
                 ;
                 ;THIS PRINTS OUT A PROPERLY PACKED ASCII CHARACTER
                 ;LEFT IN ACC A
FE0C 37          OUTCH   PSH B              ;SAVE ACC B
FE0D F6 EE  8 OUTCH1     LDA B   ACIACS     ;CHECK IF TRANSMIT READY
FE10 57                  ASR B              ;ITS BIT 6
FE11 57                  ASR B
FE12 24 F9               BCC     OUTCH1     ;IF NOT LOOP AND WAIT
FE14 B7 EE  9            STA A   ACIADR     ;OUTPUT THE CHARACTER
FE17 33                  PUL B              ;RESTORE ACC B
FE18 39                  RTS
                 ;
                 ;
                 ;THIS PRINTS OUT 2 CHARACTERS OR ONE BYTE OF DATA
                 ;INPUT ADDRESS IS IN THE INDEX REGISTER
FE19 A6  0       OUT2H   LDA A   0,X        ;GO GET THE DATA
FE1B 8D E1               BSR     OUTHL      ;PRINT LEFT HALF
FE1D A6  0               LDA A   0,X
FE1F 8D E1               BSR     OUTHR      ;PRINT RIGHT
FE21 39                  RTS
                 ;
                 ;THIS PRINTS ONE BYTE AND A SPACE
                 ;INPUT ADDRESS IS IN THE INDEX REGISTER
FE22 8D F5       OUT2HS  BSR     OUT2H
FE24 86 20       OUTS    LDA A   #20        ;LOAD ASCII FOR SPACE IN ACC A
FE26 20 E4               BRA     OUTCH      ;INCLUDES A RTS

                 ;THIS PRINTS A ? FOR DUMB MISTAKES

FE28 86 3F       WHAT    LDA A   #3F        ;LOAD IN A ?
FE2A 8D E0               BSR     OUTCH
FE2C 7E FF 11 C1         JMP     CONTRL


FE2F 86  D       OUTCR   LDA A   #0D        ;LOAD IN ASCII FOR CR
FE31 20 D9               BRA     OUTCH


FE33 86  A       OUTLF   LDA A   #0A        ;LOAD IN ASCII FOR LF
FE35 20 D5               BRA     OUTCH

                 ;SUBROUTINES FOR INPUTTING CHARACTERS
```

```
PROGRAM FOR BURNING ROM          JUN 13, 1978    X6800-V1D   PAGE    9

                       ;
                       ;INPUT 1 CHARACTER IN ACC A
FE37 B6 EE  8 INCH     LDA A   ACIACS  ;CHECK THE RECEIVE BIT
FE3A 47                ASR A
FE3B 24 FA             BCC     INCH    ;LOOP TIL READY
FE3D B6 EE  9          LDA A   ACIADR  ;GET CHARACTER
FE40 84 7F             AND A   #7F     ;RESET PARITY BIT
FE42 81 7F             CMP A   #7F     ;IGNORE RUBOUTS
FE44 27 F1             BEQ     INCH    ;IF RUBOUT LOOP BACK
FE46 7E FE  C          JMP     OUTCH   ;ECHO CHARACTER  INCLUDES RTS
                       ;
                       ;THIS IS FOR INPUTTING A HEX NUMBER
                       ;IT LEAVES THE HEX NUMBER IN ACC A
FE49 8D EC    INHEX    BSR     INCH    ;GO GET AN INPUT CHARACTER
FE4B 81 30    INHEX1   CMP A   #30     ;CHECK FOR SPECIAL CHARACTER,
                                       ;I.E. NOT HEX
FE4D 2B DD             BMI     C1      ;IF YES RETURN TO CONTROL PROGRAM
FE4F 81 39             CMP A   #39     ;SEE IF ITS A NUMBER
FE51 2F  A             BLE     IN1HG   ;IF YES , QUIT
FE53 81 41             CMP A   #41     ;CHECK FOR OTHER NON-HEX, I.E. < A
FE55 2B D5             BMI     C1      ;IF YES GO BACK TO THE CONTROL PROGRAM
FE57 81 46             CMP A   #46     ;CHECK IF > F
FE59 2E D1             BGT     C1      ;IF YES GO BACK TO THE CONTROL PROGRAM
FE5B 80  7             SUB A   #07     ;CHANGE ASCII TO HEX
FE5D 39       IN1HG    RTS
                       ;SUBROUTINE FOR PRINTING OUT THE POT VALUES AS THEY
                       ;APPEAR ON THE FRONT OF THE BOX
FE5E 86  2    POTPRT   LDA A   #02     ;START WITH FLEXION TABLE
FE60 97 42             STA A   CHANEL
FE62 CE  0  0          LDX     #0000
FE65 BD FC B0          JSR     SLYPOT  ;READ IN THE 5 VALUES
FE68 C6  5             LDA B   #05     ;SET UP LOOP TO PRINT 5 VALUES
FE6A CE  0  0          LDX     #0000   ;INTIALIZE INDEX REGISTER TO THE
                                       ;BEGINNING OF THE FLEXION TABLE
FE6D BD FE 22 PFLEX    JSR     OUT2HS  ;PRINT OUT THE DAMPING FACTOR
FE70 DF 48             STX     Y       ;MOVE INDEX REG. SO WE CAN ADD 8 TO IT
FE72 96 49             LDA A   Y+1     ;GET THE LS BYTE
FE74 8B  8             ADD A   #08
FE76 97 49             STA A   Y+1
FE78 DE 48             LDX     Y
FE7A 5A                DEC B
FE7B 26 F0             BNE     PFLEX
FE7D BD FE 24          JSR     OUTS
FE80 BD FE 24          JSR     OUTS
FE83 CE  0 21          LDX     #0021   ;RESET THE INDEX REGISTER
FE86 BD FC B0          JSR     SLYPOT  ;PUT THE NEXT 5 VALUES IN THE TABLE
FE89 CE  0 41          LDX     #0041
FE8C C6  5             LDA B   #05
FE8E BD FE 22 PEXTEN   JSR     OUT2HS
FE91 DF 48             STX     Y
FE93 96 49             LDA A   Y+1
FE95 80  8             SUB A   #08     ;SUBTRACT 8 FOR THIS TABLE
FE97 97 49             STA A   Y+1
FE99 DE 48             LDX     Y
FE9B 5A                DEC B
FE9C 26 F0             BNE     PEXTEN
FE9E 7E FF 11          JMP     CONTRL
                       ;SUBROUTINE TO ADD TORQUE LEVER ARM CORRECTION
                       ;FACTORS.  FACTORS ARE STORED IN A TABLE CALLED FIXTAB
                       ;WITH AN IMPLIED EXPONENT OF -7.
```

```
PROGRAM FOR BURNING ROM          JUN 13, 1978    X6800-V1D   PAGE   10

                      ;WHEN ENTERING THE ROUTINE THE INDEX REGISTER SHOULD
                      ;CONTAIN THE FIRST ADDRESS OF THE DAMPING TABLE TO
                      ;BE FIXED.

 FEA1  8          DAMFIX  INX              ;MOVE X TO FIRST ADDRESS TO BE FIXED
 FEA2 DF 4E               STX     POINTR   ;KEEP DAMPING TABLE ADDRESS IN POINTR
 FEA4 CE FF 3E            LDX     #FIXTAB  ;KEEP FIX TABLE ADDRESS IN THETA,
                                           ;THETA + 1
 FEA7 DF 46               STX     THETA
 FEA9 86 F9               LDA A   #-07     ;SET UP EXP FOR MLTPY
 FEAB 97 4B               STA A   Z+1
 FEAD 86 1F               LDA A   #1F      ;SET UP COUNTER FOR LOOP
 FEAF 97 45               STA A   SLYCNT   ;DO 1F OR 31 MULTIPLIES
                      ;ENTER MAIN LOOP

 FEB1 DE 4E       DAMFX1  LDX     POINTR   ;GET UNFIXED DAMPING COEFF
 FEB3 A6  0               LDA A   0,X
 FEB5 97 48               STA A   Y
 FEB7 7F  0 49            CLR     Y+1
 FEBA DF 4E               STX     POINTR
 FEBC DE 46               LDX     THETA
 FEBE A6  0               LDA A   0,X
 FEC0 97 4A               STA A   Z        ;PUT FIX FACTOR IN MLTPY SPOT
 FEC2 BD FC F8            JSR     MLTPY
 FEC5 96 4C               LDA A   ANSWER
 FEC7 D6 4D               LDA B   ANSWER+1
 FEC9 27  4               BEQ     STORE
                      ;BEFORE PUTTING ANSWER BACK, MAKE SURE IT HAS THE PROPER EXP
 FECB 47          DAMFX2  ASR A
 FECC 5C                  INC B
 FECD 26 FC               BNE     DAMFX2
 FECF DE 4E       STORE   LDX     POINTR   ;GET PROPER ADDRESS TO PUT CORRECTED
 FED1 A7  0               STA A   0,X      ;DAMPING COEFF BACK IN
 FED3 7A  0 45            DEC     SLYCNT
 FED6 27  D               BEQ     QUITS
 FED8 7C  0 4F            INC     POINTR+1
 FEDB 7C  0 47            INC     THETA+1
 FEDE 26 D1               BNE     DAMFX1   ;CHECK FOR OVERFLOW DURING INC
 FEE0 7C  0 46            INC     THETA
 FEE3 20 CC               BRA     DAMFX1
 FEE5 39          QUITS   RTS
                      ;THIS IS A SUBROUTINE TO DECIDE WHICH OF THE
                      ;CONTROL MODES TO BRANCH TO DEPENDING ON THE
                      ;PIA SENSE LINES
                      ; 1XXX XXXX  CAUSES THE PROGRAM TO GO INTO ITS NORMAL MODE
                      ; 011X XXXX  ACCEPT INPUT FROM KEYBOARD DEVICE
                      ; 001X XXXX  CHECK THE SLIDE POT VALUES WITH AN ANALOG
                      ;            VOLTMETER.  BITS ARE THE ADDRESS ON THE MUX
                      ;            TO BE CHECKED
                      ;ACC A ALREADY CONTAINS THE 7 SENSE LINES
                      ;BIT 0 IS IGNORED

 FEE6 48          DECODE  ASL A            ;LOOK AT BIT 6
 FEE7 2B 28               BMI     CONTRL   ;IF HIGH GO INTO KEYBOARD MODE
                                           ;OTHERWISE PUT BITS BACK IN PLACE
                                           ;TO GET MUX CHANNEL FOR POT TO
                                           ;TO BE CHECKED
 FEE9 44                  LSR A
 FEEA 44                  LSR A
 FEEB 84  F               AND A   #0F      ;MASK OFF 4 MSB
```

```
 FEED 97 42     DECOD1  STA A   CHANEL  ;PUT IN PLACE FOR A/D SUBROUTINE
 FEEF BD FC 97          JSR     VALGET
 FEF2 BD FC AB          JSR     FIX
 FEF5 58                ASL B           ;MULTIPLY BY 2
 FEF6 F7 EE 10          STA B   DAOUT   ;DUMP THE RESULT ON THE D/A
 FEF9 7E FC 3F          JMP     MAIN    ;RETURN TO MAIN PROGRAM FOR MORE
                                        ;ORDERS
                ;SUBROUTINE FOR CHECKING THE GAIN SETTING
                ;GAIN PCT IS CHANNEL 12 OR 0C ON THE MUX

 FEFC 86  C     GAINCK  LDA A   #0C     ;GO GET VALUE FOR GAIN SETTING
 FEFE 97 42             STA A   CHANEL
 FF00 BD FC 97          JSR     VALGET
 FF03 BD FC AB          JSR     FIX
 FF06 D7 43             STA B   SWIVAL
 FF08 C   0 43          LDX     #SWIVAL
 FF0B BD FE 19          JSR     OUT2H
 FF0E 7E FC 3F          JMP     MAIN
                ;THIS IS THE CONTROL PROGRAM THAT DECODES THE
                ;INPUT COMMANDS
 FF11 8E ED C4 CONTRL   LDS     #STACK  ;SET THE STACK POINTER TO
                                        ;ITS INTIAL VALUE TO CANCEL
                                        ;ALL SUBROUTINE CALLS THAT
                                        ;HAVE BEEN STACKED UP
 FF14 BD FE 2F          JSR     OUTCR   ;PRINT A CR
 FF17 BD FE 33          JSR     OUTLF   ;PRINT A LINE FEED
 FF1A 86 40             LDA A   #40     ;PRINT A @
 FF1C BD FE  C          JSR     OUTCH
 FF1F BD FE 37          JSR     INCH    ;GET A COMMAND
 FF22 16                TAB             ;SAVE THE COMMAND IN ACB
 FF23 B  FE 24          JSR     OUTS    ;PRINT A SPACE
 FF26 C1 4D             CMP B   #'M     ;MEMORY CHANGE OP
 FF28 26  3             BNE     A1
 FF2A 7E FD D0          JMP     CHANGE  ;GO FLY A KITE
 FF2D C1 50     A1      CMP B   #'P     ;PRINT THE POT VALUES
 FF2F 26  3             BNE     A2
 FF31 7E FE 5E          JMP     POTPRT
 FF34 C1 47     A2      CMP B   #'G
 FF36 26  3             BNE     A3
 FF38 7E FE FC          JMP     GAINCK
 FF3B 7E FC 3F  A3      JMP     MAIN
      FF 3E     FIXTAB  .EQU    .
      FF 3E             .ORG    FIXTAB
 FF3E 7A                .BYTE   7A,75,70,6D,69,67,64,62,60,5F
 FF3F 75
 FF40 70
 FF41 6D
 FF42 69
 FF43 67
 FF44 64
 FF45 62
 FF46 60
 FF47 5F
 FF48 5D                .BYTE   5D,5C,5B,5B,5A,5A,5A,5A,5A,5B,5B
 FF49 5C
 FF4A 5B
 FF4B 5B
 FF4C 5A
 FF4D 5A
 FF4E 5A
```

```
PROGRAM FOR BURNING ROM          JUN 13, 1978    X6800-V1D   PAGE   12

 FF4F 5A
 FF50 5A
 FF51 5B
 FF52 5B
 FF53 5C                   .BYTE   5C,5D,5E,60,61,63,65,67,69,6C
 FF54 5D
 FF55 5E
 FF56 60
 FF57 61
 FF58 63
 FF59 65
 FF5A 67
 FF5B 69
 FF5C 6C
      FF F8                .ORG    0FFF8
 FFF8 FC  0                .DBYTE  0FC00
 FFFA FC  0                .DBYTE  0FC00
 FFFC FC  0                .DBYTE  0FC00
 FFFE FC  0                .DBYTE  0FC00
                   ;THAT'S ALL SHE WROTE.
                           .END
```

```
PROGRAM FOR BURNING ROM           JUN 13, 1978    X6800-V1D   PAGE   13

   EE08 ACIACR      EE08 ACIACS      EE09 ACIADR      FD25 ADDEM
   EE20 ADINPT        4C ANSWER      FD5D ATSALL      FF2D A1
   FF34 A2          FF32 A3          FDA5 BADDR       FDB0 BYTE
     42 CHANEL      FDD0 CHANGE      FDD2 CHANG1      FDEC CHANG2
   FD29 CONT        FF11 CONTRL        44 COUNT       EE12 CRA1
   EE22 CRA2        EE13 CRB1        EE23 CRB2        FDBA CRCHEK
   FF2C C1          FEA1 DAMFIX      FEB1 DAMFX1      FECB DAMFX2
   EE10 DAOUT       FEE6 DECODE      FEED DECOD1      FD56 DONE
   FD9A DUMP        FCED ENDPOT      FD9F FFDUMP      FCD4 FINE
   FCAB FIX         FF3E FIXTAB        50 FLAG        FEFC GAINCK
   FD13 GO          FD65 GOODY       FCAF HENRY       FE37 INCH
   FE49 INHEX       FE4B INHEX1      FC59 INVERT      FE5D IN1HG
     21 KEXTEN         0 KFLEX       FD8A LFTSHT      FC3F MAIN
   FCF8 MLTPY       FD3F MORE        FCEC MORPOT      FCD0 MORVAL
   EE21 MUXCTL      FDE1 NEXONE      FD01 NOPROB      FDC2 NORMAL
   FC4A OK            47 OMEGA       FE0C OUTCH       FE0D OUTCH1
   FE2F OUTCR       FDFE OUTHL       FE02 OUTHR       FE33 OUTLF
   FF24 OUTS        FF19 OUT2H       FE22 OUT2HS      FE8E PEXTEN
   FE6D PFLEX       FC84 PLUS          4E POINTR      FF5E POTPRT
   FDEF PRADDR      FFE5 QUITS       FD92 RGHSFT      FC5A SAVVEL
   FD5E SCALE       EE11 SENSIN      FD2F SHIFT       FD4B SHIFTL
     45 SLYCNT      FCB0 SLYPOT      FDC4 STACK       FECF STORE
     43 SWIVAL        46 THETA       FC97 VALGET      FE28 WHAT
     4B XHI           49 XLOW          48 Y             4A Z

ERRORS:    0
```