National Library of Canada / Bibliothèque nationale du Canada

Canadian Theses Service / Services des thèses canadiennes

Ottawa, Canada
K1A 0N4

CANADIAN THESES / THÈSES CANADIENNES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ MICROFILMÉE TELLE QUE NOUS L'AVONS REÇUE

NL-339 (r. 86/06)

Canada

A MICROCOMPUTER CONTROLLED ABOVE-ELBOW PROSTHESIS USING EXTENDED PHYSIOLOGICAL PROPRIOCEPTION

by

Sebastien
Philippe-Auguste

A thesis

presented to the School of

Graduate Studies and Research

of the University of Ottawa

in partial fulfillment of the requirements

for the degree of

Master of Applied Sciences

in

Electrical Engineering

Ottawa, Canada, 1986

© Sebastien Philippe-Auguste, Ottawa, Canada, 1986.

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-36542-0

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

The University of Ottawa requires the signatures of all the persons using or photocopying this thesis. Please sign below and give address and date.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my academic supervisors Dr. D.T. Gibbons and Dr. M.D. O'Riain for their valuable assistance, supervision and moral support during the course of the program.

I also wish to thank Louis Goudreau, Gordon Evans and all the staff of the Rehabilitation Engineering at the Royal Ottawa Hospital, for their help and support.

This work was supported by the Natural Sciences and Engineering Research Council of Canada.

ABSTRACT

Externally powered upper-extremity prostheses have been made for various levels of amputations. While some degree of success and acceptance of the prostheses have been obtained for below-elbow amputations, prostheses for higher levels of amputation have a much poorer record of user acceptance. The most important reason for the rejection is often ignored and it is the fact that the majority of externally powered multi-functional prostheses are under open-loop control. As a consequence of this, the operation of these prostheses requires intense concentration of their users and constant visual monitoring. In other words no subconscious movement is possible.

This thesis reports the work done on the application of a control scheme, known as "Extended Physiological Proprioception", to a prosthesis. This control scheme makes use of the inherent proprioceptive feedback present within an intact joint to provide closed-loop control.

The ~~prosthesis~~ under consideration is for above-elbow amputees. It has electric powered wrist and elbow joints. The proposed technique is to control the positioning of this prosthesis using the motion of the intact shoulder. The relationship between input (shoulder position) and output (wrist and elbow positions) is set according to a program or linkage stored in a microcomputer memory. A repertoire of linkages are provided to the user so that different tasks can be performed.

It is expected that the use of Extended Physiological Proprioception will help to reduce considerably the necessary user concentration to operate the prosthesis, so that subconscious operation becomes possible.

TABLE OF CONTENTS

lIST OF FIGURES

INTRODUCTION

This report discusses the research work, concerning the design and realization of a prosthesis for an above-elbow amputee. It has been assumed that the amputee has a relatively long arm stump on which the prosthesis can be fixed. The described device will replace part of the arm, the elbow, the forearm, the wrist and the hand.

A prosthesis is a technical product made to serve the amputee. It can never be a total substitute for the lost part of an extremity. The purpose of the device is to provide some functions important to the patient, however it should at the same time be comfortable to wear, easy to use, reliable and fully under the control of its operator. Almost all the above features were present in early prostheses using body power to control the functions of the artificial arm. These prostheses were, in fact, extremely simple; They used either rounding of the back or shoulder shrug to operate a Bowden cable that activated the artificial limb.

Simpson described a body powered prosthesis as follows [1]: " A body powered prosthesis for the above-elbow amputee is usually provided with a flexible elbow and a prehension hook mounted on a friction turntable to enable passive wrist rotation. Both the elbow and the hook are activated by rounding the shoulders to tighten a strap, thus pulling the Bowden cable. A shoulder activated mechanical switch locks the elbow in place allowing further shoulder action to perform the hook opening. Elbow extension uses gravity and the hook is spring loaded in the closed position."

Unfortunately, body powered prostheses had drawbacks and limitations that were difficult, if not impossible, to solve since they were inherent to the use of body power itself. Among the major disadvantages were:

- -the need for excessive harnesses to support and operate the prosthesis.

- -the impossibility of having coordinated motions, as the elbow and hook could not move at the same time.

- -the impossibility of using this kind of prosthesis on subjects with high levels of amputation (among which we include congenital amputations caused by the drug Thalidomide), such as bilateral amelia, since wide amplitude body movements are not, in these cases, available to power the system.

Use of external power for the prosthesis was the obvious solution to these problems. Two major sources of power that have been considered are pneumatic and electric (as a third option electro-hydraulic systems are now also considered, because of their potential advantage of developing high torque in small actuators). However, pneumatic systems never gained popularity in the U.S. because of strict government standards on cannisters of gas that had to be made of steel, which added to the weight [2].

Opposed to compressed gas, electrical power can be stored cheaply, more safely, and with greater density than gas power [3]. Also, the control possibilities offered by electronic circuits have given electrical systems an advantage. Nevertheless, electrical actuators remain heavy and only major progress in the design of electric motors and gearing systems will be able to provide researchers with an adequate tool that can challenge more efficient actuators such as pneumatic or hydraulic ones.

In the coordinated motion of a human upper-extremity, body, shoulder, and arm frequently function together to put the hand in the proper spatial location for doing its work. This involves simultaneous control of several degrees of freedom. With intact subjects, coordinated movements are stored in memory (in a way which is not yet fully understood) and can be performed subconsciously without undue concentration [4]. The problem of controlling simultaneously several degrees of freedom becomes more involved when it has to be performed in a conscious manner. It is believed that, as the number of degrees of freedom (or joints) increases, the complexity in the coordinated control of these joints also increases, but in a non-linear way.

The control of multiple degrees of freedom artificial limbs has been the concern of prosthetics researchers for the past few decades. Various options have been investigated, most of them involve the use of the activity of physiological sites to control the prosthesis. These activities can either be electrical (the low-level signals from biological sites such as myoelectric activity, or EMG) or mechanical (the mechanical displacement of a joint) as exposed below.

When using mechanical input, or biomechanical control, the force and/or displacement of a limb is used to control the motion of the prosthesis. A small displacement of the given limb, also called the input limb, induces a comparable displacement of the prosthesis. A large displacement of the input limb causes a wide amplitude movement of the prosthesis. This mode of control has a lot of advantages. It provides a possibility to feedback the information about force, position and velocity of the prosthetic device directly to the control site, which does not seem possible when myoelectric control is used. Biomechanical control also supplies the prosthesis with a non-corrupted (i.e without noise) type of information which does not require any processing before it can be used by the prosthesis control system, which again, as we shall see, is not the case in myoelectric control. Among the various biomechanical control sites that have been investigated, the best results seem to be obtained when using shoulder flexion-extension or abduction-adduction, or clavicular movements for shoulder disarticulation amputees.

Myoelectric control consists of capturing EMG signals emanating from the muscles and using them to control the prosthesis. Traditionally, designers have approached the control problem on a one-muscle-for-one motion basis; that is, a discrete muscle site is used to control a given joint. The more joints there are the more the muscles sites that are needed. This approach was quite adequate when only one or two degrees of freedom were to be controlled. However, as the number of degrees of freedom increased, the quality of this method stagnated because of the additional levels of skill and attention needed to perform tasks [5].

A more recent approach is to consider the EMG signals from the point of view of a synergetic group of muscles. It is believed that to each limb function (i.e elbow bending, elbow extension, wrist pronation, wrist supination etc.) is associated a defined EMG pattern, thus, if an accurate discrimination between these patterns can be performed each of them can be used to control one degree of freedom of the prosthesis. However, discrimination between the various limb functions to achieve fast control for practical aspects is not simple. Different methods were proposed to solve this problem [6] and they will be exposed in the coming chapter.

Various strategies are available to control an externally powered multifunctional prosthesis but, as we shall discover in the following section, only few of them can pretend to allow transfer of information not only from man to prosthesis but also from prosthesis to man, which is very important.

CHAPTER I

OVERVIEW ON THE CONTROL OF MULTIFUNCTIONAL ARTIFICIAL UPPER EXTREMITIES

1.1 Some proposed control strategies

According to a survey published by D.S. Childress in a paper entitled : " Historical Aspects of Powered Limb prostheses " powered limbs have existed since 1915 [2]. But it is only with the advent of the transistor that electric powered limbs could really be developed and considered as being a potentially useful tool in prosthetic applications. However, as suprising as it may seem, powered prostheses never gained popularity and only a small number of patients actually used myoelectric-controlled electric hands. Passive devices such as the well known hook or aesthetic dummy limbs, are fitted far more often and are better accepted by amputees. The major reason for the rejection of powered prostheses is believed to lie in the control strategy [7].

The most elegant way to control an electric prosthesis would be to use the electrical signals sent by the brain. These signals can be captured at the stump site (for example an above-elbow amputee) where they take the form of Electromyographic (EMG) signals. (EMG signals result from the contraction or expansion of muscular tissues in response to brain stimuli). It is believed that these EMG signals form patterns that are distinguishable with regard to the intended motion. In-depth studies were made on the information content of these signals and algorithms were developed to use computers in the myoelectric control of powered prostheses. Recent advances in technology, such as the development of powerful microprocessors, have made the on-line analysis of the EMG patterns possible. Basically, the information obtained from the analysis of these patterns is used to control the various degrees of freedom of the artificial arm.

There are two well-known approaches that utilize this mode of control. The first , based on the work of Lawrence [8], of Lyman et al [9], or Jacobsen and Mann [10], requires mapping of signals from many (usually more than ten) electrode sites, at each of which the EMG function is strongly correlated with a single limb function. The signals emanating from these sites are weighted and analyzed to determine what action the prosthesis (the artificial arm) should take. This can be done since the forces involved, for example, in lifting an object are transmitted through the musculoskeletal system of the arm and shoulder. The muscles of the upper arm and shoulder are, therefore, engaged in a " symphonic-type activity" related to the particular actions of the arm. It is believed that this activity can still be brought about after amputation, since in a large number of cases, the amputees retain a residual image of the lost limbs, often complete with sensations. The purpose of the multiple electrode sites and decision network is to determine what "tune" the muscles are playing and translate it into the appropriate motion of the artificial arm [11]. It is readily seen from the above description that this approach is not practical, because it requires too many electrode sites and precise positioning of these electrodes each time the prosthesis is to be worn, and also, the system is totally open-loop.

The second approach, based on Graupe et al [6], requires between one and three electrode sites which are used to control five limb functions. In this system, the complete spectral information of the EMG signal is used; which makes it more optimal-linear in terms of information utilization as opposed to the first approach discussed. Function discrimination is achieved using an Auto-Regressive Moving Average (ARMA) identification algorithm. The reported discrimination success rate is 85 percent which is too low for a prosthesis to be acceptable, since this means that 15 percent of the time the amputee will not be in full control of his prosthesis which can result in hazardous situations. Also, uncertainty in operations leads the amputee to avoid the use of the prosthesis.

In the above mentioned approaches, some work was done to provide the amputee with sensory feedback, through the use of electrotactile stimuli. However, these systems can be considered in effect open-loop as this form of display, while useful to some extent for force feedback, is totally inadequate as a means of supplying position information. An amputee wearing an EMG controlled prosthesis would have to rely solely on visual feedback in order to know where his terminal device (the hand) is moving in space. This, in turn, may require considerable concentration by the user and hence, with multijoint control, this may put considerable mental burden on the patient to operate the prosthesis. It must be noted, at this point, that in all systems where the use of so called "unnatural" feedback (such as the one described in the above mentioned example) [7] have been tried, in vast majority of cases, they have been rejected by their users.

Some other problems related to the use of EMG signals are the tremendous amount of signal processing required to extract the desired information. This in turn causes : on-line analysis to be very slow (more than 200 milliseconds) and make self-containment of the prosthesis impossible, because the amount of electronic hardware required is excessive to fit entirely in the prosthesis forcing the user to carry part of it in his pocket or attached to his body.

Fortunately, there are other controlling methods to achieve coordinated motions in multifunctional (or multijointed) artificial prostheses. These methods are certainly less elegant but more promising than myoelectric control. One of them, first proposed by D.C. Simpson [4] and termed by him : " Extended Physiological Proprioception " (e.p.p.), involves the use of an intact joint as the input to the system. The prosthesis is used as a mechanical extension of the body.
In the principle of e.p.p. there is always a one-to-one relationship between the input (the intact joint) and the output (the prosthesis). In this way, the intact joint can be used as a bidirectional channel passing command information from the biological side to the artificial system and returning back to

the central nervous system information about motion, velocity and position of the prosthesis. This principle is supported by many observations. The stick of the blind person and the racquet of the tennis player are examples of mechanical extensions to a natural system. The person learns to associate the position of his body joints with the position of the artificial extension (the stick or the racquet) which then provide him with subconscious control and (extended) physiological proprioception. As Polanyi [4] says :

"... The way we use a hammer or a blind man uses his stick, shows in fact that in both cases we shift outwards the point at which we make contact with the things that we observe as objects outside ourselves. While we rely on a tool or a probe, these are not handled as external objects. We may test the tool for its effectiveness or the probe for its suitability, e.g. in discovering the hidden details of a cavity, but the tool and the probe can never lie in the field of these operations; they remain necessarily on our side of it, forming part of ourselves, the operating person. We pour ourselves into them and assimilate them as parts of our own existence. "

In introducing the principle of e.p.p., Simpson stated as one of its fundamental rules that for the control to be preserved at all time, the input should not be allowed to move faster or further than the prosthesis can follow. The necessity of this feature, termed by Simpson the "unbeatable servo" [12], is to preserve a one-to-one relationship between input and output so that the amputee never has to look at his terminal device (his hand) to know exactly where it is in space. With this, a 100 percent success rate in positioning the prosthesis can be achieved.

In the next section we shall have a look at the advantages and limitations when using the e.p.p. mode of control in the design of a multifunctional above-elbow prosthesis.

1.2 The Above-elbow Prosthesis Control Strategy.

Innovations in microelectronics and computer aided mechanical design (CAMD) have greatly enhanced the field of prosthetics. These technical innovations of the past decade have provided us with extremely powerful tools such as microprocessors, microcontrollers and digital signal processors which are priceless instruments for research and development. However, even with the introduction of these technically sophisticated mechanical (CAMD) and electronic systems, it would be totally unrealistic to think that, today, a "bionic arm" incorpo rating all the movements and dexterity of a human arm could be realized.

In their struggle to find a solution to the replacement of lost limbs, many researchers have created prostheses which are very sophisticated from both technical and aesthetical points of view, and which mimic as closely as possible the natural limbs. For example, in the previous section, the work done on EMG controlled multifunctional prostheses was mentioned. These prostheses, undoubtedly, make use of the near-ideal form of control, but they can be efficiently operated only by the researchers or experts that have conceived them. Unfortunately, these designs remain technical masterpieces with little practical value as they do not provide the amputee with a simple enough tool that can be reliable, easy to learn, and easy to use. As a consequence of this, their level of acceptance among amputees is extremely low. On the contrary, simple prostheses such as the body powered hook have gained extreme popularity among amputees and such devices are still in widespread use.

In the light of these observations, Gibbons and O'Riain stated in the early eighties that; with the available present technology, and in the context of unilateral amputation, an above-elbow prosthesis cannot efficiently act as a total replacement for the lost limb. It could, on the contrary, act as a useful assistant to the remaining (valid) arm for most of the normal daily activities.

- It is amazing how simple daily tasks can be difficult to execute with one hand while extremely easy when done with two. One can think, for example, of the act of drawing a straight line using a pencil and a ruler. The exercise becomes tedious if only one hand is used. The difficulty arises because two different entities, i.e the pencil and the ruler, have to be positioned separately and moved one against the other as the line is drawn. The need for a second hand is obvious in this example, but it can also be noted that the so called "assistant hand" does not have to be skilled, if it can hold the pencil in the upright position or if it can apply pressure on the ruler then the line will be easily drawn. Gibbons and O'Riain also observed that many tasks are often repetitive (such as ironing, sweeping, operating a bench saw or a drill press) and involve the simultaneous use of two or more joints; for example when operating a bench saw coordinated motion of shoulder and elbow is required.

On the basis of these observations Gibbons and O'Riain set the design criteria for our above-elbow prosthesis. These are outlined below:

(i) - The prosthesis will not be conceived to be a complete substitute for the lost limb, its purpose will be to help and assist the valid hand for certain tasks.

(ii) - For security reasons the amputee should always be in full control of the prosthesis and it should provide him with constant position, velocity and force feedback in a so called "natural" manner so that the operation of the device can be performed subconsciously.

(iii)- The prosthesis should be self contained, which means that the control circuitry and power source must all fit in the prosthesis housing so that the amputee does not have to carry any of it in his pockets or attached to his body.

In setting these priorities it was hoped that an easy-to-learn, easy-to-use prosthesis could be realized, which, in turn, would contribute to a better acceptance among amputees.

In the previous chapter it was shown that myoelectric control (EMG) could not, as it now stands, be used efficiently for the control of multijoint prostheses. Lack of "natural-type" feedback and non-ideal positioning success rate (100 % is the ideal one) were the major drawbacks of this mode of control. It must be kept in mind, however, that myoelectric control can be improved (as technology advances) and if it is combined with some form of proprioceptive feedback then it might become the solution for future prostheses. For the time being, using an intact joint to provide control and proprioceptive information for the artificial arm seems to be the best solution. Body powered prosthesis and Simpson's body controlled gas powered prostheses are concrete proof of the validity of this approach. That is why Gibbons and O'Riain decided to use the principle of extended physiological proprioception (e.p.p.) as the control modality for our electrically powered above-elbow prosthesis. However they decided to use a modified form of e.p.p..

In conventional e.p.p., every prosthetic joint should be controlled by a natural joint [13]. Motion of the joints of the prosthesis can be accomplished by moving a specific body joint for each artificial joint being controlled. ~~But~~, it was felt that the coordinated motion of several joints moving simultaneously could require too much concentration from the amputee, which would , in essence, violate the initial objectives. That is why a modification to conventional e.p.p. was proposed. It involves the use of one intact joint to control all the joints of the prosthesis. The motion of the intact joint will result in the coordinated motion of two or more prosthetic joints. Thus the prosthetic joints will be, in effect, "linked" one to each other, and by insuring a one-to-one relationship between input (shoulder) and output (elbow and wrist joints), the amputee will be able to associate the position of his intact limb (shoulder joint) with the position in space of the prosthesis. Thus at all times the amputee will know where his prosthesis is without having to look at it. This will relieve the amputee from having to concentrate too much when performing coordinated motions.

Gibbons and O'Riain not only focused on the practical aspects of the prosthesis but also on the aesthetic point of view. They decided to use the shoulder joint as the input to the system, although other joints were available. The reason behind this is related to the fact that movements of upper-extremities often require the coordinated motion of the hand, elbow and shoulder. Thus, by using the shoulder as the controlling input, the movements of the prosthesis look less mechanical and more natural.

The shoulder flexion/extension is the only parameter used to position the prosthesis. Thus, movements of the hand or elbow not involving that of the shoulder cannot be performed. In this case, if such movements are frequent for a particular individual, clavicular joint movement or chest expansion can be appropriately used as the prime input to the system.

In systems using e.p.p. such as body powered prostheses using Bowden cables, there is only one linkage between input and output which is fixed by the mechanical constraints of the prosthesis. Such prostheses have proven successful because they are easy to use, reliable and fully under the control of their operator [7]. However, it was decided that one linkage was not enough. Thus to overcome the mechanical constraints of the earlier prostheses we decided to put our system under microprocessor control so as to provide a repertoire of movements, also called linkages, to the user. All the amputee has to do is select one of the linkages to perform a given task. Since we are concerned only with unilateral amputation, switching between different linkages will be activated by the intact hand. Because the artificial arm is only an assistant to the valid arm, it is anticipated that it will only be used mainly to position and hold objects while the normal hand is working on them. In addition, it may be used in repetitive tasks. Thus frequent switching between different linkages will not be necessary.

The linkages are stored in the microprocessor memory which allows them to be tailored to the specific needs of the amputee.
The input to the microprocessor is provided by a shoulder electro-goniometer that measures the angle of shoulder flexion/ extension. This parameter is used by the microprocessor to calculate the desired position of the elbow and wrist, as a function of the selected linkage, and drive them accordingly. Thus, for the same shoulder position, the position of the elbow

and wrist will be different for different linkages.

With a variety of input output relationships (or movements) at his disposal, it is believed that the amputee will have to learn to use each of them, just as if they were different tools. However this learning process should not be too demanding on the user. The prosthesis can be envisioned as a tool that can be used in different ways (depending on the selected linkage) for executing different tasks. We all learn to use various tools for various purposes: a racquet is used to play tennis, while a stick is used to play hockey and a bat, to play baseball. All these tools are used as extensions of the body (as the prosthesis is) to do some work and it is fairly easy to learn to use them without any confusion whatsoever. The bat will never be used to play hockey and the stick to play tennis. Thus from this analogy, we can extrapolate that the amputee will quickly learn to associate each linkage with each task and the difference between them will become obvious. The linkage that allows him to drink from his cup of tea will not be confused with the one that permits him to iron and so on.

In addition to the various linkages provided to the amputee it was decided to include, in our system, the "unbeatable servo" feature proposed by Simpson. This feature helps to keep a one-to-one relationship between input and output so that the input can never go faster or further than the output can. It is essential for the amputee to know without a visual check, but rather with proprioceptive feedback, whether or not his prosthesis can perform the required task. Simpson implemented this feature by connecting a cable between the system output and the input joint. This system continuously prevented the input from exceeding the output capabilities of the prosthesis. In our system due to mechanical and cosmetic constraints it was not possible to implement this important feature. Instead, a vibration sent by the microprocessor to the shoulder will warn the user that the servo has been "beaten". The user can then make the appropriate adjustments by either slowing down his movement or backtracking a little until the vibration stops, meaning that the output has caught up with the input. In this case the use of an "unnatural" type of feedback such as the vibration, is not in contradiction with what we have said before; rather it complements it because we are not using this kind of feedback for position sensation or information, we are just using it as an on/off display that warns the amputee.

Having introduced the concept behind the control of our above elbow prosthesis we shall present, in the following chapter, the prototype prosthesis together with its electronic hardware and software.

CHAPTER II

THE PROSTHESIS HARDWARE

2. The Mechanical Parts of the Prosthesis

In the previous chapter the controlling concept and the design criteria of our prosthesis were laid down. We determined that extended physiological proprioception was the most appropriate mode of control for multijoint prostheses. We also noted that simplicity of utilization, position proprioception (i.e position sensation) and reliability in positioning the joints of the prosthesis were among the most important features for the acceptance of the device by amputees. Following these guidelines, a prototype prosthesis was constructed. We chose the option of designing an above-elbow prosthesis rather than a below-elbow or shoulder disarticulation prosthesis. However, in these cases, the procedure would have been similar. Our choice was influenced by the fact that the above-elbow prosthesis has a mechanical level of complexity in-between the other two. However, from the controlling electronics and software point of view there is not much difference between these three options.

In this study our goal is really to show that the use of a microprocessor could make e.p.p. and the "unbeatable servo" features possible. The present above-elbow prosthesis makes use of an electric elbow and an electric wrist rotator both controlled by modified e.p.p., and a prehension mechanism that is controlled by EMG signals taken from the biceps and triceps muscles. The hand grasp is driven in a proportional manner by a system developed at the Rehabilitation Institute of Montreal [14].

As illustrated in figure 2.1 the prosthesis is composed of five major parts, which are :

1) The hand with its prehension mechanism.
2) The wrist unit
3) The elbow unit
4) The shoulder Goniometer
5) The microprocessor and controlling electronics

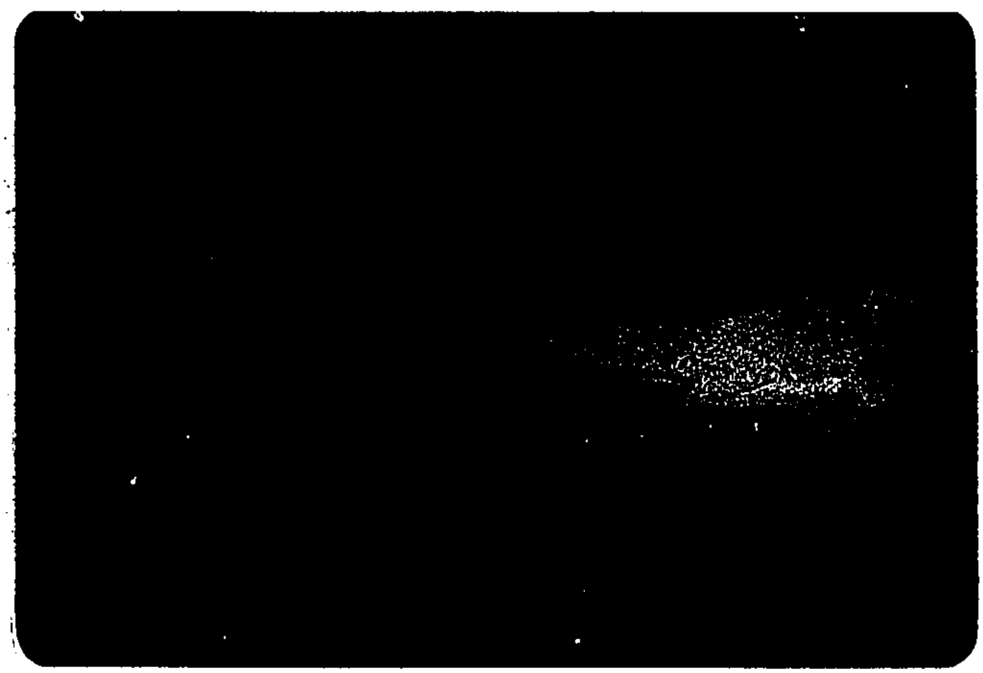


Fig. 2.1 The Above-Elbow Prosthesis.

It should be noted that the hand, wrist rotator and elbow were not designed by us. They are off-the-shelf components that have been sold for years and have been fitted on many amputees, hence they have proven their effectiveness and reliability. However, some of them were modified so that they could be adapted to our needs.

In the next section we have included a brief description of these electromechanical elements, for completeness.

2.1 The Hand Unit

This hand was designed and manufactured by Otto Bock Orthopedic Industries, a German prosthetics company. It is a simple pinch prehension, i.e. the fingers cannot be controlled independently. The operation of the hand is controlled by EMG signals from biceps (the hand opening) and triceps (hand closing) muscles. Because hand grasp requires only force, not positioning, it is appropriate to use EMG as a simple on/off control. Also, the hand grasp is driven in a proportional manner by a system developed by the Rehabilitation Institute of Montreal (RIM) [14]; This system named Analog Proportional Control, is such that to any increase of the rectified myoelectric signal there is a corresponding proportional increase of the DC power available to the motor. Thus, the subject can control the motor torque by varying the level of contraction in the control muscles (here, the biceps and triceps). This enables him to control the speed of the hand opening and closing and also the prehension force [15].

An advantage with this system is its flexibility of utilization, because it can also be operated using a mechanical displacement as the (controlling) input, instead of EMG. As it stands now, sensory feedback of the prehension force is not provided. However, if we decided later on to use a mechanical displacement as the input, then we will be able to provide feedback at the same time. This hand has been fitted on below elbow amputees at the RIM and seems to give good satisfaction to its users.

2.2 The Wrist Rotator Unit

The wrist rotator unit (see figure 2.2) was also designed by the Otto Bock company. It provides 360 degrees rotation, but the microprocessor was programmed so as to restrict its range of rotation from full pronation (palm down) to complete supination (palm up) in order to make the movements of the hand look more natural. Unfortunately this unit does not provide us with wrist flexion/extension but this option is not indispensable since it can be adequately replaced by shoulder and elbow motions.

We have incorporated in the wrist unit a linear potentiometer to measure the angular rotation and provide the microprocessor with position information. The drive to the wrist unit is bidirectional and microprocessor-controlled allowing fast and accurate positioning of the terminal device (i.e the hand).

Fig.2.2 The Otto Bock wrist unit.

2.3 The elbow unit

A variety of elbow units were available on the market. The most popular ones are: the Variety Village elbow (VV elbow), The Boston elbow, and the Utah arm. We decided to use the Boston elbow, because the two others had serious disadvantages. The VV elbow, developed for children, was too weak for our purposes. The VV elbow could not move when fitted with a wrist rotator unit and a prehension mechanism. There is no need to say that it could not lift any weight.

The Utah arm, on the other hand, was the latest designed among these three elbows, but its price, about five times that of the Boston elbow (in 1982) made it an inaccessible development tool. Also, it was not flexible enough to accept the various modifications that we had to apply to our prototype elbow.

The Boston elbow was developed by the Liberty Mutual Assurance Company in collaboration with The Massachusetts Institute of Technology (MIT) Boston (see figure 2.3 (a)). This elbow is currently available for above-elbow amputees. Off-the-shelf it provides : one degree of freedom (i.e. elbow flexion/extension) controlled by EMG signals taken from the biceps and triceps muscles. We mentioned in the previous chapters that we were not going to use EMG to control our prosthesis (except for the hand prehension), but rather e.p.p.. Thus we did not make use of the electronics supplied with the Boston elbow, we designed our own. In fact, just the mechanical parts and the electric motor of the Boston elbow were used in our system.

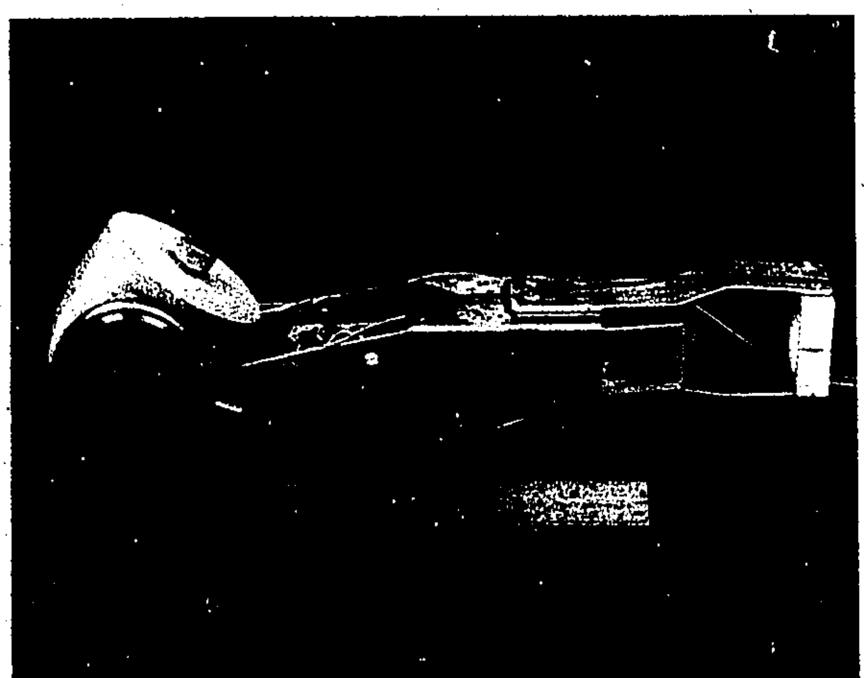


Fig. 2.3 (a) The Boston elbow.

We also took advantage of the empty cavity present in the fore-arm housing to place the controlling electronics, the micro-processor and the battery pack making the prosthesis self-contained. Some minor mechanical modifications were required to be able to place a potentiometer in the elbow housing to provide feedback position information of the elbow joint to the micro-processor.

The Boston elbow offers a lot of features that makes it attractive [1]:

1) High speed - 145 degrees in about one-and-half second.
2) Free swing for ease in walking - the elbow swings freely 30 degrees or locks directly to motor.
3) Length is adjustable to small or large amputees.
4) Reverse locking clutch - arm locks unless motor is being driven.
5) high torque - 6.1 Newton-meters.
6) Can support 23 kg at 30 cm.
7) Overriding clutch protects the arm during falls.
8) routine maintenance is required only once per year.

The hand was fitted on the wrist rotator and they were both secured on the Boston elbow. Our above-elbow prosthesis was then ready to be tested. Figure 2.3b gives a general representation of the prosthesis as it would look on an amputee.

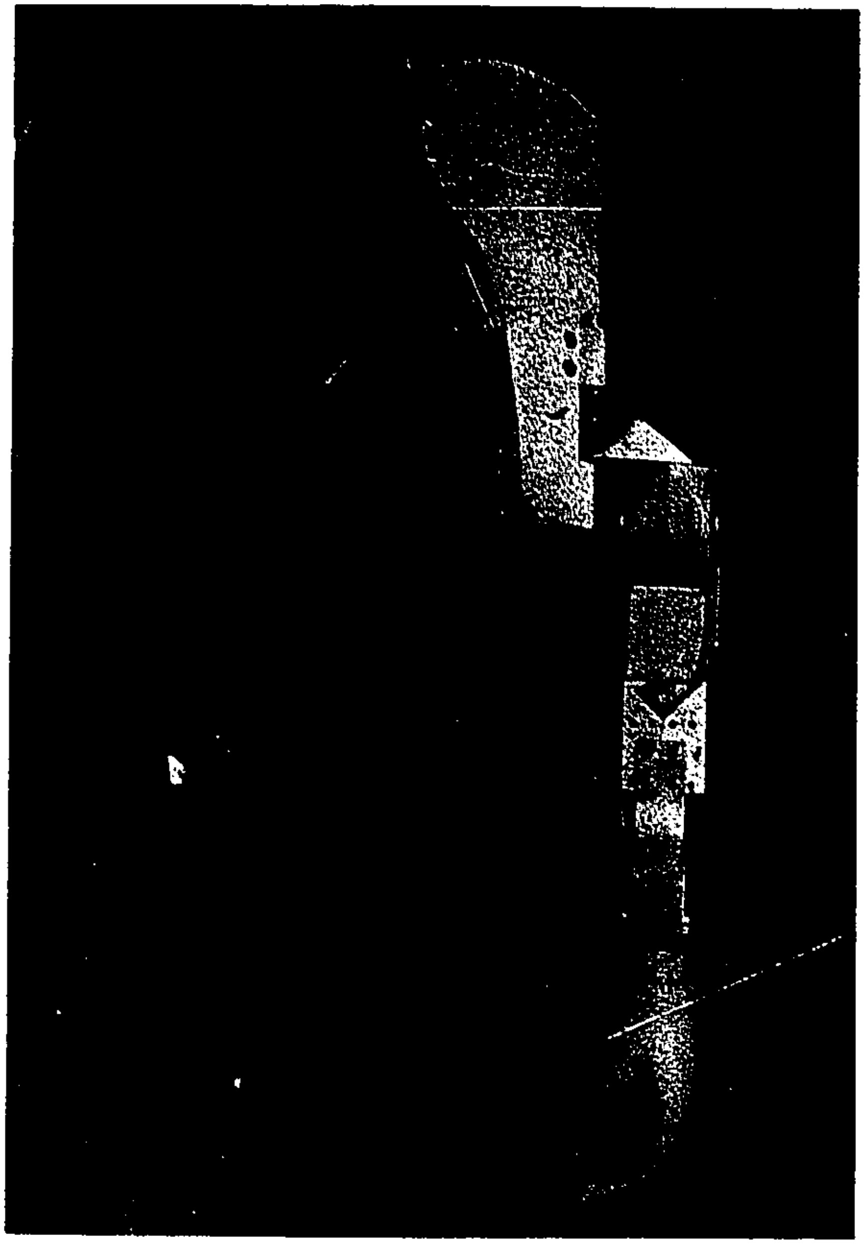


Fig. 2.3 (b) The prosthesis as it would be worn by an amputee.

2.4 The Goniometer

We have developed, in conjunction with the Royal Ottawa Regional Rehabilitation Centre, a shoulder goniometer. This goniometer is unobtrusive and can be an integral part of the harness of the prosthesis (see figure 2.4). The potentiometer that measures shoulder flexion/extension, fits under the shoulder, between the upper arm and the trunk (see figure 2.4). The two arms of the goniometer are maintained parallel to the upper arm and trunk respectively. A sliding rod mechanism ensures that shoulder movement is not constrained. The goniometer has a range of motion of about 128 degrees which is more than enough to allow for a complete elbow flexion/extension movement or a full pronation/ supination twist of the wrist.

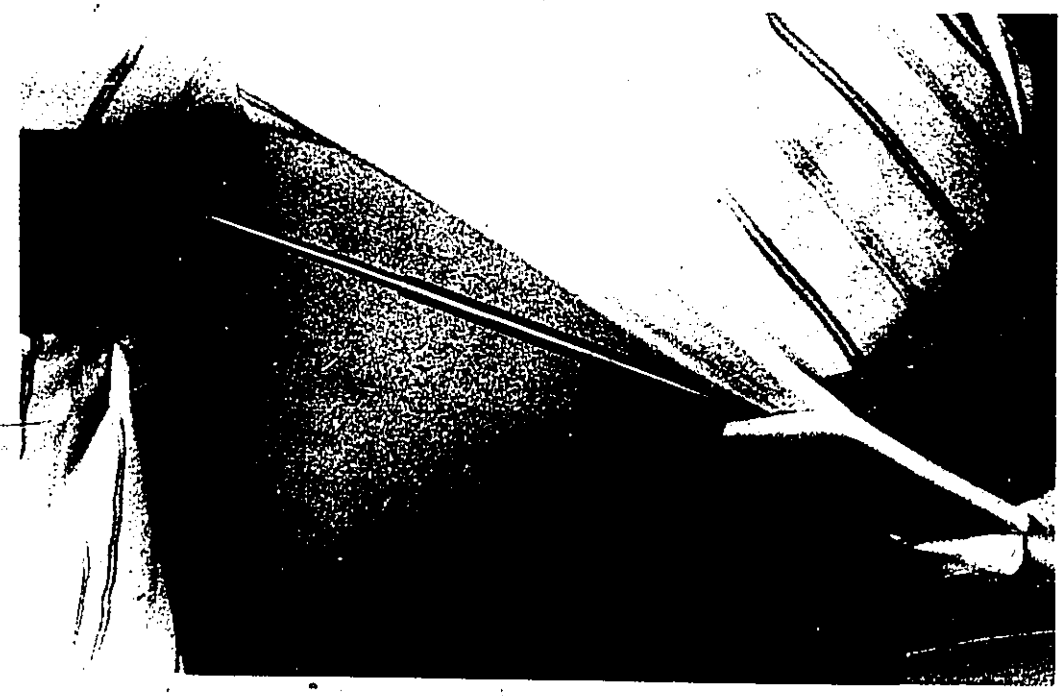


Fig. 2.4 The goniometer

Many attempts were made to incorporate into the goniometer a braking mechanism that would allow the "unbeatable servo" feature to be realized. This braking mechanism should be such that if at any time the output capabilities of the system are exceeded, the microprocessor would output a braking command. This command would remain until the system has reached the desired output demanded by the input, or if the shoulder movement (the input) is attempted in the opposite direction to that which caused the braking. This implies that the braking mechanism has to be bidirectional with fast application and release times. Unfortunately such mechanical design is neither easy to realize nor aesthetical. Instead we realized a software brake that does exactly the required task but still lacks the "natural-type" feedback that would be achieved with a hardware braking mechanism.
-In an intact subject, when his arm is not able to lift a heavy weight, for example, the joints are forced by the weight to stop in a certain position. This situation will automatically and subconsciously alert the subject, and he will be able to react to correct this aborted attempt. This is exactly what a hardware braking mechanism would provide.

For closed-loop controlled motion to be made possible, input and output position information have to be fed to the microprocessor; but since these signals are analog, they will have to digitized before they can be processed by the microprocessor. The next section will discuss the data acquisition hardware used in our system, the microprocessor used, and the various interfaces needed to link all these elements.

2.5 The controlling Electronic Hardware

In this section we will discuss the data acquisition interface required to monitor:

- The input to the system (the shoulder angle)
- The output of the system (the position of the prosthesis for feedback information).

There are essentially three important data inputs to the microcomputer. The first one is the goniometer, which is a one turn potentiometer (see figure 2.5.1). The maximum allowable displacement of this potentiometer is 128 degrees.

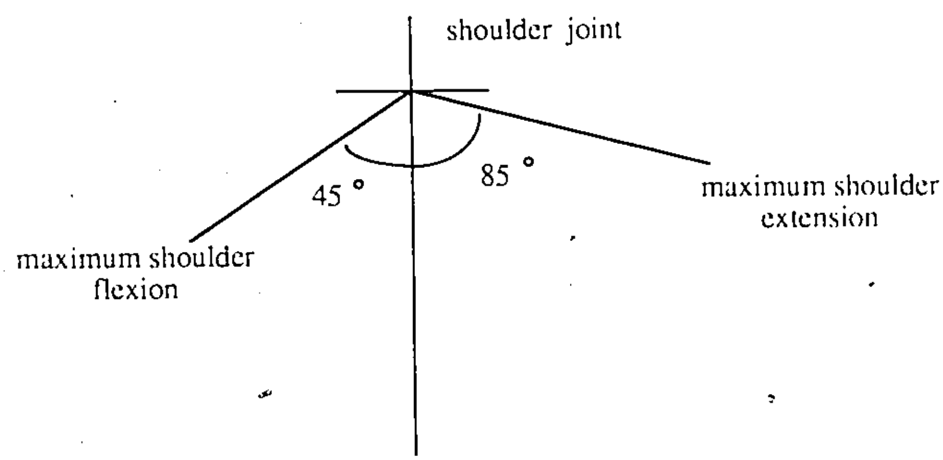


Fig. 2.5.1 Schematic diagram of the goniometer.

It is to be supplied by the 0 to 5 volts power rail of the system, hence wherever the reference is set, the range of variation of the voltage will be about 1.8 volts from position A to position B. The second and third inputs to the microprocessor are the ones coming from the elbow and the wrist (which also have one turn potentiometers). The elbow potentiometer is also allowed to move freely for 128 degrees for wide amplitude flexion/extension movement, whereas the wrist is allowed 180 degrees rotation for proper pronation (palm down) and supination (palm up) movements. All these potentiometers are read continuously and their signals fed to an analog to digital converter (ADC), hence a four-to-one line multiplexer had to be placed at the front end of the ADC.

Since our control system should consume as little power as possible to allow for, long hours of operation, all integrated circuits should be of MOS/CMOS type. It was also decided that only off-the-shelf components were going to be used so as to keep down the price of the electronic hardware.
The dual 4 channel analog multiplexer MM 54HC4042 (from National Semiconductor) was readily available. It featured eight channels all together, very low quiescent current (80 microamperes maximum), and fairly low "on resistance" 50 ohms when Vcc = Vee = 5 volts (see [16] for details).
The range of operating temperatures of this device (from -30 to +40 degrees Celsius) together with its frequency response near 100 MHz, were well above those required for prosthetic applications. Thus this device was considered adequate to be used

in our system.

The next link of the data acquisition system is the analog to digital converter (ADC). An 8-bit converter was proposed assuming that the precision of the least significant bit (LSB) would be far in excess of that furnished by the rest of the system. By this we mean the following :

- firstly, the capability of the electric motors to stop within a certain range from the position requested by the controlling input (This depends on the inertia of the motors and their relative loading).
- secondly, the accuracy of the potentiometers in monitoring the position of the joints.

An in-depth look at the problem revealed that an 8-bit ADC was more than enough in the controlling loop. The variety of analog to digital converters available is extremely large today. Thus our choice was oriented towards an inexpensive but nonetheless "intelligently" conceived device. We chose an 8-bit A/D converter using the successive approximation technique (the Analog Devices AD 7574 see [17] for more information) because it satisfied our basic needs. Some of the interesting features of this converter, from the point of view of our prosthetic system, were that it was readily compatible with an 8-bit microprocessor (which we are using), and that it used three-state logic for its output data bits, thus allowing direct connection to the microprocessor data bus or input/output ports. Also this ADC (AD 7574) was designed to operate as a memory mapped input device. It could be operated as static RAM, ROM or as slow memory. The conversion time 15 microseconds, power dissipation of 30 milliwatts, and low cost, also influenced our choice. Unfortunately, as is usual, this ADC had some drawbacks. The one that we were concerned with was the very low impedance (approximately 10 kilc-ohms see [17] for more details) of the analog input, which would inevitably cause a loading of the previous elements of the link (i.e. the multiplexer and the potentiometers) thus inducing voltage drops along the analog path. In the worst case these drops could induce an ADC error of two LSB. Thus, a buffer was needed at the front end of the ADC. However one should bear in mind that the board space is extremely limited, since the control circuit and power pack must all fit in the empty cavity located in the forearm of the prosthesis. The addition of any extra element reduces both the available space and available power. Thus extreme care was taken in the design in order to optimize these two parameters. The device stability with respect to temperature was considered acceptable. In any case, if any offset or nonlinearity would appear in the conversion due to temperature, autocalibration techniques could be applied using the microprocessor. At this stage, we shall ignore the errors due to temperature changes assuming that their influence is not as critical as compared to that due, for example, to the low input impedance of the ADC which is a constant source of error.

Before moving on to the next stage, which is the microprocessor, it might be interesting to note that as the stand-by quiescent current seems to be fairly high (approximately 5 milliamperes maximum) a pulsed mode operation of the ADC will be considered in order to cut-down on power consumption. In this mode, the ADC will be switched on by the processor only when a conversion is required, then it will be powered down again.

Once analog incoming signals have been digitized, they must be processed so that valuable information can be extracted and utilized for control purposes. The processing device will be discussed in the following paragraphs.

The brain of our system was chosen to be the Intel 8751 which is a high performance single chip 8-bit microcontroller, intended for use in sophisticated real-time applications such as robotics. The 8051 family is composed of three different versions : 8031/ 8051/8751. We chose the 8751 which is the only one, of that family that has a programmable EPROM of 4K-bytes as an integral part of the device. Thus the 8751 is well suited for development, prototyping (which is exactly what we are doing), and low-cost, low-volume production. In the following section our aim will be to introduce some of the interesting features of this microcontroller. Additional information is available in [18].

The Intel 8751, as a development oriented device, is an extremely powerful tool. It helps to simplifying both software and hardware which is one of the main reason why it was chosen to be the microcontroller of our prosthesis. At the time when it was decided that a microcontroller was to be used in the system, various features, that were to be present in the controller, were set down. These required features were directly related to our design constraints which were in essence :

- to minimize components (due to lack of space). The microcontroller had to have at least 2K_bytes of on board EPROM (no external memory was to be accepted), two or more interrupt lines with selectable levels of priority, two timers, and finally minimum hardware interface with peripheral devices (which must be TTL/MOS compatible).

- To minimize power, so that a full-day's use on a single battery pack was possible. A CMOS or hybrid device was required having a power down or "stand by" mode that could be used when the processor was not doing any useful work, so that power could be cut to a minimum.

- To minimize execution time so that real-time processing was possible. This was an important factor since the response of an artificial limb should be close to real time in order to achieve, not only a good technical operation but also a natural one. Thus a high frequency clock was needed for fast instruction execution together with built-in functions such as multiply or divide that are usually time consuming when realized in software. Also simple instructions for input/output (I/O) were required, since the controller would be practically executing at every instant in time I/O operations. Finally, since the control algorithm (to be discussed in the next chapter) makes use of a timer to control the drive to the joint actuators, another timer was needed to control possible power down mode of various peripherals and/or to be used to synchronize the operation of a new joint if the number of degrees of freedom of the prosthesis was increased.

As an answer to these challenging constraints, the Intel 8751 provided us with all the needed features and more: on a single die the 8751 microcontroller combines CPU; non-volatile 4K (8-bit) erasable programmable read-only memory (EPROM); volatile 128 (8-bit) RAM memory; 32 I/O lines; two 16-bit timer/event counters that can be read and reloaded on the fly; a five-source, two-priority-level, nested interrupt structure; on-chip oscillator and clock circuits. For completeness, a block diagram of the internal architecture of the 8751 is shown on figure 2.5.2

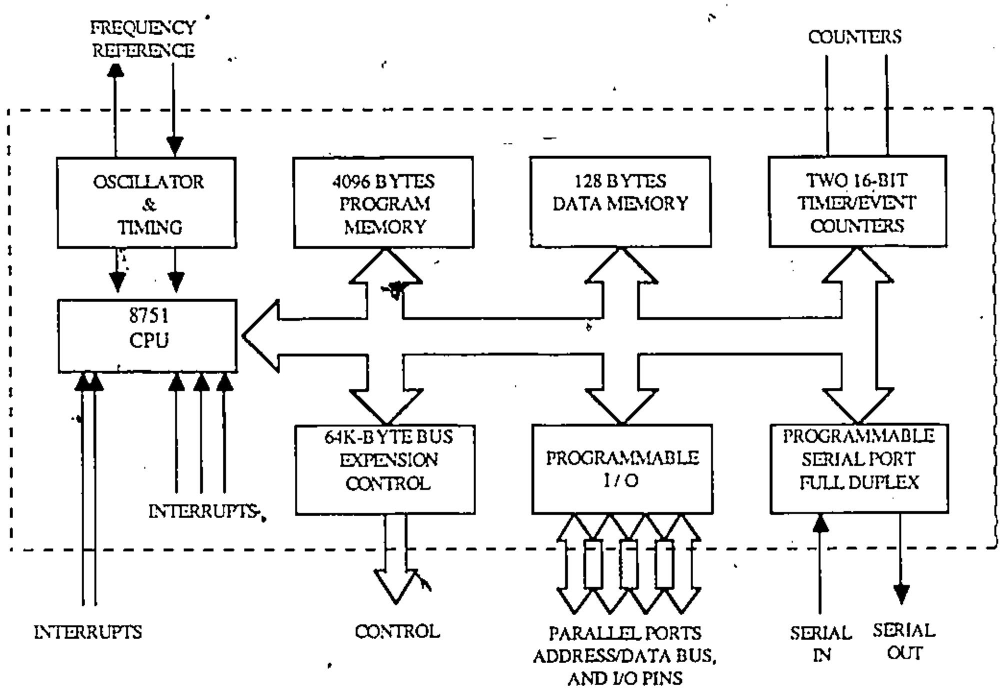


Fig. 2.5.2 The 8751 internal architecture.

The CMOS technology allowed us to have approximately 25 Milliwatts of power consumption which could be reduced to less than 5 Milliwatts on power down mode. This is, as we think, good for a device having so many on-board components. Notice that all input/ output ports can sink/source one TTL load. The 8751 was used with a 12 MHz clock giving an average execution time of 1 to 2 microseconds for each instruction except for multiply and divide which require 4 microseconds.

The control algorithm did not require more than a thousand lines (approximately) of code written in assembler language, thus computation remained within "real-time" limits, and the prosthesis reactions were almost instantaneous.

As far as input/output operations were concerned, the manipulation of data was extremely easy. Most instructions could be used for I/O operations. There are 32 I/O lines that can be accessed as 32 individually addressable bits or as four parallel 8-bit ports addressable as ports 0, 1, 2, and 3. In a single 8-bit port each bit can be set individually as an input or output port, which greatly simplifies bit operations and hardware connections. We shall see later on, how extensive use of this feature took place in the allocation of lines to various control units.

All the above mentioned features, combined with a total of 111 possible instructions, make the programming of the 8751 very easy. We have also appreciated the eight general purpose registers (R0 - R7) which allowed easy manipulation of data. We also used the 16-bit timer/counters that provided sufficient flexibility in the way they could be programmed and sufficient numbers of bits to set a count or division factor to the desired length.

Before concluding this presentation of the Intel 8751 one important aspect remains undiscussed : the effect of varying temperatures on the specifications of the integrated circuit. This microcontroller has its ratings guaranteed for temperatures from 0 to 70 degrees Celsius under bias and from -65 to +150 under storage conditions. These temperature ranges are acceptable for us since our design is an introductory one and we are not yet concerned with temperature extremes. Also, a careful look at the design reveals that the prosthesis was programmed to execute repetitive tasks such as those encountered around the house or in workshops where obviously the temperature is maintained above 0 degrees and below 70 Celsius. If the amputee is to go in open air, his prosthesis will usually be off or on stand-by mode, which should prevent him from having any problems. However, tests will be made in an artificial environment with extreme temperatures to be able to set the guaranteed ratings of the design.

It is a fact that more powerful microprocessors with larger processing power, larger memory and larger I/O capabilities were available, but we found the Intel 8751 more than sufficient for our prosthetic application. We have chosen to use it primarily for its small size. No other supporting hardware was required besides a crystal to supply the clock pulses. This compactness is extremely important since we wanted ~~to~~ place all controlling electronics in the forearm housing of the Boston elbow, making the system self-contained. A prosthesis of any kind is certainly more attractive for an amputee when he does not have to carry any parts of it in his pockets or attached to his body.

The prosthesis electronics also contain the motor drive which is composed of discrete TMOS transistors (see [19] for specifications on these transistors) and some CMOS combinational logic. The TMOS transistors were mounted in a H switch configuration (see figure 2.5.3) so that they could provide a bidirectional drive to the elbow and wrist motors.

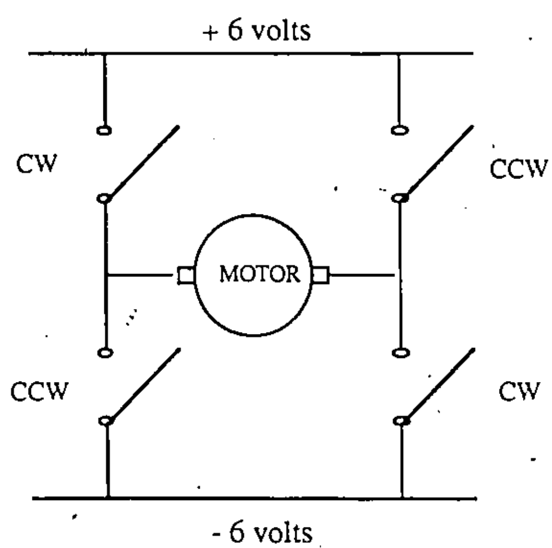


Fig. 2.5.3 TMOS transistors mounted on H switch configuration.

These power TMOS transistors were used instead of the usual bipolar complementary pair (NPN, and PNP) for two major reasons :

- Their "on voltage" was as low as 150 millivolts, thus voltage losses could be minimized.
- They could be driven by standard CMOS logic gates, since they required only a negligible amount of current at their gate input. Thus no buffered logic gates were needed to drive these power transistors, which is not the case when bipolar power transistors are used.

The combinational logic was necessary to provide system security, and energy conservation. When the ends of the rotation of the joints are reached, the motor drive is automatically switched off. Greater details on circuit diagrams and integrated circuits used are given in appendix A.

Having introduced and justified the need for the various elements composing the controlling electronics, we can now have a look at the complete system. In the next chapter we will see how this controlling system is connected with the electromechanical parts of the prosthesis and the outside world (i.e. the amputee himself.

CHAPTER III

3. The System

Figure 3.1 is a block diagram of the microcontroller interfaced with all the peripherals i.e. the ADC, the multiplexer, the logic to the motor drive, and the linkage selection switch.

There are three external analog sources of information to the microcontroller. The first two are from the elbow and wrist joint potentiometers, they provide position information on these joints. The third source is from the shoulder goniometer which provides the position information of the amputee's shoulder joint.

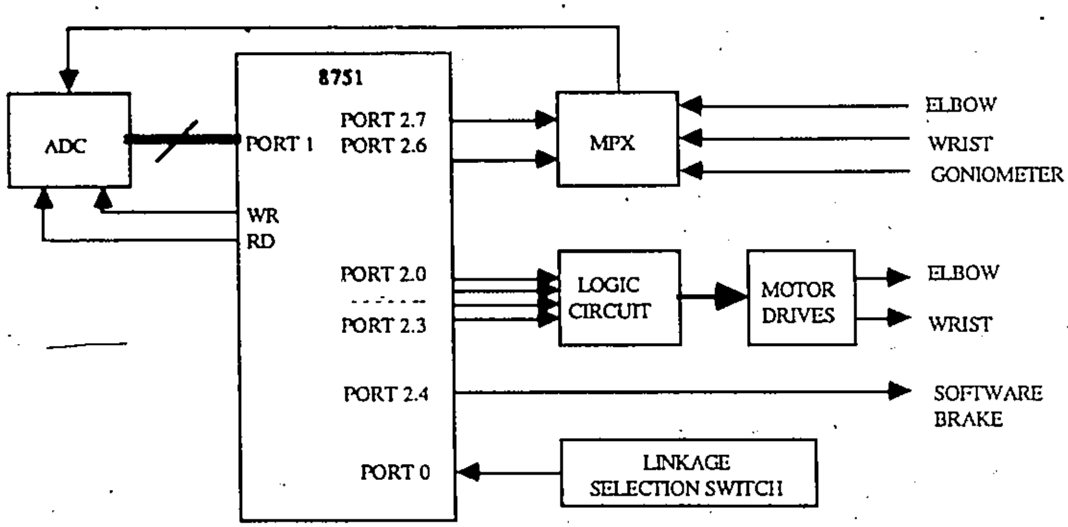


Fig. 3.1 Block diagram of the controlling hardware.

These analog signals are passed on to an analog multiplexer which is under software control. Upon request from the processor, one of these signals is selected and passed to the analog/digital converter where it is converted into an 8 bit digital value.

The input/output (I/O) ports of the microcontroller are used, rather intensively, for communication with the outside world. However, due to the fact that so many I/O lines were available we decided to allocate to every I/O port a particular task so as to ease debugging of the circuit. These ports use three-state logic which allows them to be connected to different devices at the same time, but we will not use this feature. Thus ports 0 and 1 were defined as input data ports:

The lines of port 0 are connected to a three bit switch which is under patient control. Using this switch the patient can select one of the eight different linkages that has been custom-made for him. The bit allocations are as follows.

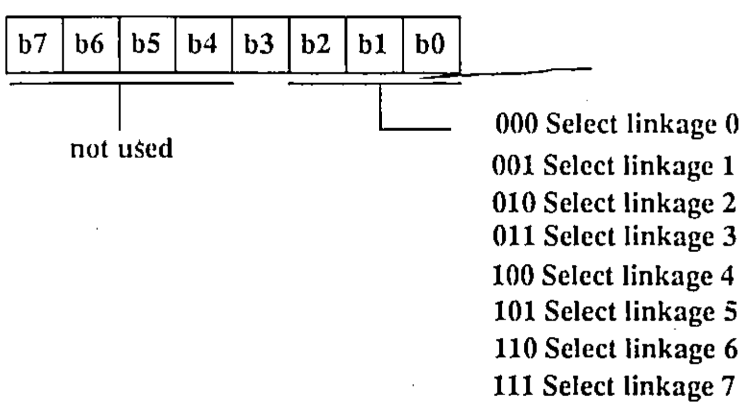


Fig. 3.2 : port 0 bit allocations.

Bits 1.0 to 1.7 of port 1 are directly connected to the output lines of the ADC, as shown in figure 3.3.

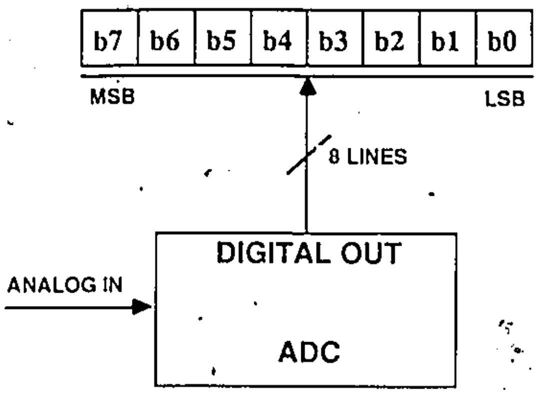


Fig. 3.3 Bit allocation for port 1.

The remaining two ports, which are port 2 and port 3, are mostly used as control lines to the various units. These ports are configured as output ports.

Port 2 controls the channel selection of the multiplexer, and also provides the pulse-width modulated drive, and direction of motion to both the elbow and wrist joints. The bits of this port are used as in figure 3.4.

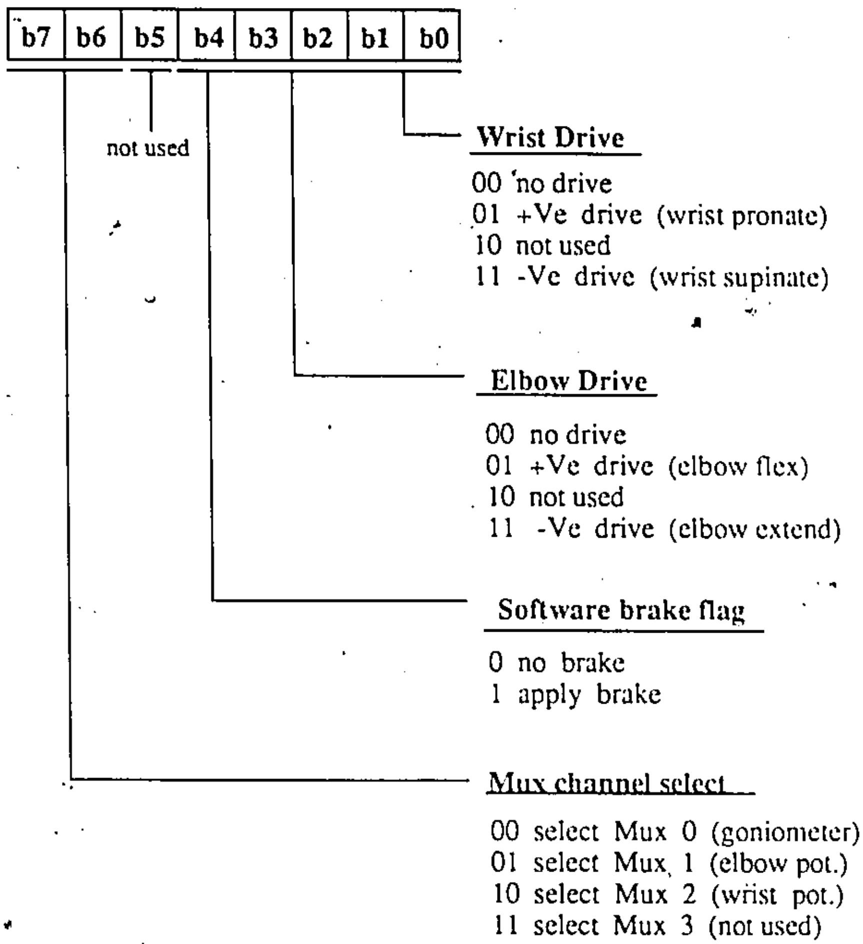


Fig 3.4 : port 2 bit allocations.

One line (2.1) is, at present, used to implement the "software brake" feature discussed in the previous chapter. If ever the input is moved too fast or too far for the output to follow, the system automatically responds by sending a vibration to the shoulder of the amputee warning him that he is about to "beat" the servo. Line 2.1 is set when the software brake is in operation, otherwise it is cleared.

Only two lines of port 3 are used, they are : the read (RD), and the write (WR) lines. They are connected directly to the RD and WR inputs of the analog to digital converter. Figure 3.5 gives the bit allocation of port 3.

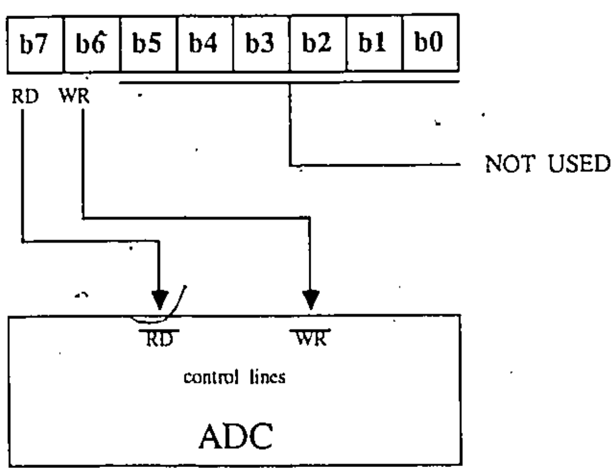


Fig. 3.5 : port 3 bit allocations.

A modular design was applied when conceiving this system. Each module was designed and tested separately so as to insure that it was bug free. Even though there was not much space available, the system was equipped with protection circuitry against overcurrent, noise, and voltage polarity reversal. Open circuit and battery discharge detection will also be provide in future devices.

In this section the controlling electronics were presented from a system point of view with the various modules that performed a given task. However, a sequential and coordinated operation of these modules is needed to achieve a closed-loop control of the prosthesis. In the following chapter the basic control software is discussed.

CHAPTER IV

THE CONTROLLING SOFTWARE

4.1 The Control Algorithm

The algorithm that was used to control the prosthesis is illustrated in figure 4.1. As can be seen in this flowchart, the program runs in an infinite loop. The sequence of steps shown is always executed in that order. Here are the actions taken each time the loop is entered:

1. The shoulder position (the input) is read.
2. The linkage selection switch is read.
3. Knowing the shoulder position and the selected linkage, the microcontroller can determine the new desired position of the elbow and wrist joints.
4. The actual position of the elbow and wrist joints are read and the microcontroller determines the difference (or error) between the desired position and the present one. From this error, the necessary drive to the motor of the joints are computed.
5. The computed drive is applied to the motors for a fixed time interval "T".
6. The program returns to the beginning and the same process starts over again.

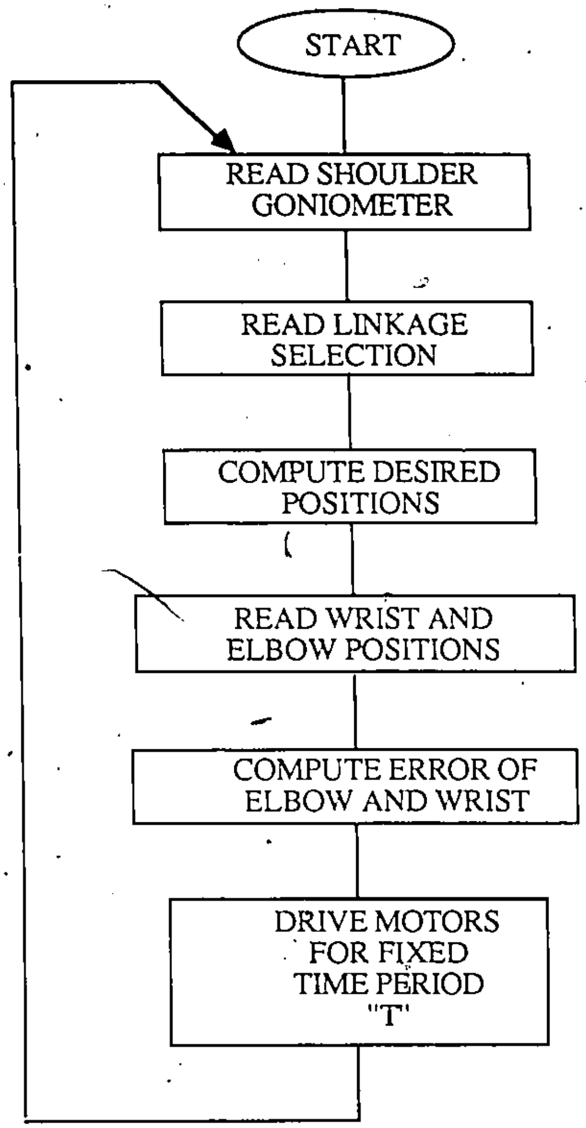


Fig. 4.1 Flowchart of the prosthesis control program.

This software was written, at design stage, in a modular fashion so that each module could be independently tested, debugged and optimized (see in appendix B the flowchart of the prosthesis controlling program and the elbow and wrist error evaluation routines). Thus, the operational version of the software had the following architecture :

- a main module taking care of steps 1, 2, 5, and 6.
- five submodules performing the tasks of steps 3 and 4.

These submodules were introduced because it was believed that the separation of elbow and wrist routines would lead to more flexible programming. The same algorithm is used to control the elbow and the wrist except that the unbeatable servo feature is not applied to the wrist. The reason behind this decision was that after investigation it was apparent that the wrist was always less likely to fail to respond to a command than was the elbow. The risk of overloading or "beating" the wrist servo was minimal. It must be kept in mind that the prosthesis was conceived to be a faithful assistant to the intact arm and if any hard work is to be performed, most of it should be done by the intact arm. This philosophy gave us a fair guarantee that the wrist would not fail to respond adequately to the input.

The hardware timer of the 8751 is used to establish a time base (T). During each of these time base periods, the following steps are executed : The microcontroller reads the shoulder position (the input) then reads the linkage selection switch. The value of the switch gives the offset to a look-up table where the corresponding <u>desired position</u> of the elbow and wrist are stored.

Different look-up tables are used for elbow and wrist joints. Having obtained the desired position of the elbow and wrist, the actual position of these two joints are read from the feedback potentiometers. The controller then determines the difference between where the joints are and where they are supposed to be. This difference is also known as the "error" between input and output in control theory.
The error is then used :

1) to check if the elbow joint is about to be beaten by the input, in which case the software brake is applied to avoid the loss of the one-to-one relationship between input and output and also to prevent a waste of useful power in a stalled motor.
2) to compute the drive to the joint motors.

Having computed the required drive, the problem remains to transform this value into an appropriate signal to drive the motors. The problem that we were confronted with was how to relate the error signal to the applied drive. A traditional method is to use a "bang-bang" type of applied drive. This drive technique is illustrated in figure 4.2. Essentially, the motor is driven in one direction or the other, there is no proportionality between the computed error and the applied drive.

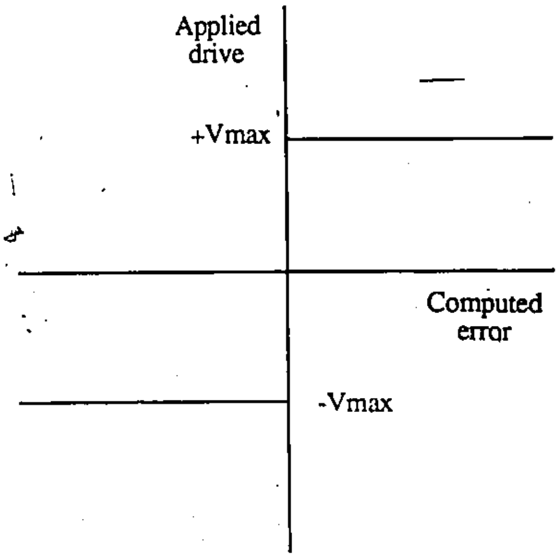


Fig. 4.2 Bang-Bang type of drive.

Another possibility is linear analog control, which consists of increasing the DC voltage available to the motor proportionally to the computed error. A relation as given by figure 4.3 (a) could be applied.

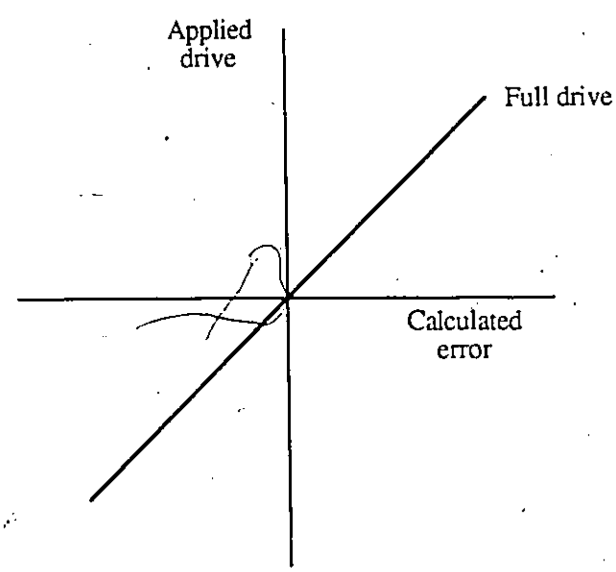


Fig. 4.3 (a) Linear type of drive.

A third method consists of applying pulses with a duty cycle proportional to the calculated drive (or error). In this case the average level of energy supplied to the motor is controlled by the width of the pulse. This is known as pulse-width modulation. This method is widely used to drive motors in robotics and prosthetics. However there is still a controversy about its use. It was reported by Ulen, and Wagner [20] that Pulse-Width Modulation was not more efficient in terms of power consumption than linear analog control. On the contrary, advocates of this driving modality such as Lozac'h et al have found its utilization very efficient in prosthetic systems [14]. It is a fact that Pulse-Width Modulation (PWM) has some advantages over linear analog control as Lozac'h described it [14] :

" The electrical energy applied to a motor is transformed into mechanical energy only when it reaches a certain percentage of the nominal energy. This percentage is a function of the load on the system. Inability to reach this percentage of the nominal energy due to loading may be the cause of a current drain through the motors, although no displacement of the driven device is observed. In the case of a prosthesis, where the energy storage is limited, this unnecessary power consumption will undesirably shorten the daily usage time."

From the above argumentation it can be seen that when linear analog control is used the voltage applied to the motor will not, for a certain time, reach the percentage of nominal energy (see figure 4.3 (b)). Thus, the motor will not move. The amount of energy supplied to the motor, for that amount of time, is completely wasted.

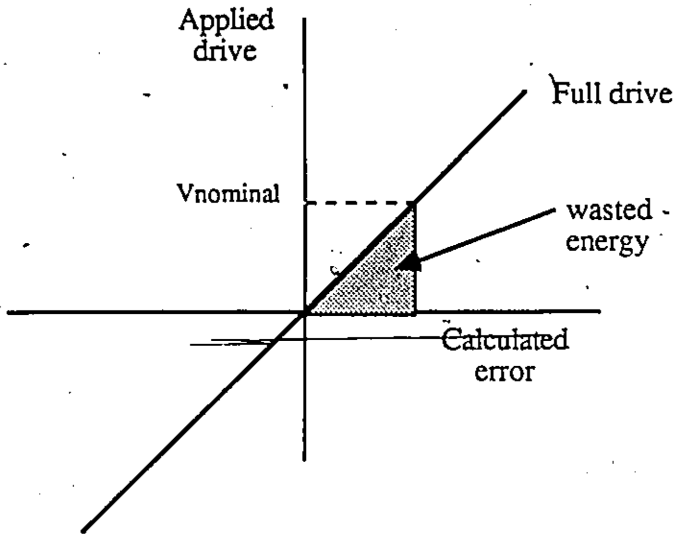


Fig. 4.3.(b) Linear type of drive.

The principle behind Pulse-width Modulation (PWM) is as follows : A waveform of amplitude Vmax, fixed period "T" and variable width "W" is applied to the motor (see figure 4.4).

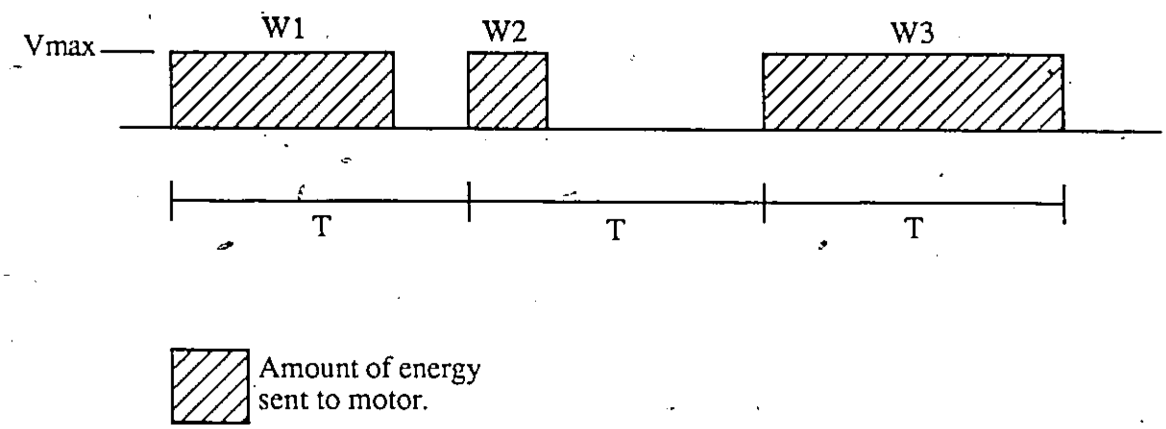


Fig. 4.4 Pulse-Width Modulation.

By varying the width of the pulse, the energy sent to the motor and hence the speed can be controlled. It usually takes many cycles and hence many on/off switchings of the voltage (Vmax) to get a significant rotation from the motor, especially if it is geared. It must be noted that there is a minimum width (Wmin), around 15 % of the time period T, below which the motor will not respond, and a maximum width (Wmax), around 90 % of the time period T, above which the motor will respond as if it was DC driven with a voltage V = Vmax.

In the present study it was discovered that PWM also had some drawbacks. Among the most serious ones were the difficulty of fine positioning and the excessive generation of noise in the electrical system. The problem of fine positioning came from the fact that the minimal amount of energy (W = 15% of T) that could be sent to the motor to make it move was sometimes too much, especially when the motor was already in motion. This problem is inherent in Pulse-Width Modulation since the amplitude of the applied voltage is always maximum, causing abrupt and step-like motion of the motor. If the desired position falls in between a step there will always be an error between the present position and the desired one. The obvious solution to this problem would be to use a reduced voltage so that the motor could be accurately guided to the desired position. But this would be in contradiction with the use of Pulse-width Modulation.

Although our design was well protected against noise, the sharp voltage edges present, when switching the pulses on and off, were generators of high frequency noise which heavily corrupted our digital system. This in turn would sometimes induce errors in the digitization process.

In the light of these observations we decided to use a method that would combine the advantages of analog control which provides the joint motors with a clean DC signal free of switching noise, and the advantages of PWM which guarantees that the motor will always be provided with sufficient voltage to start. A simple relation of the form of figure 4.2 seems to be sufficient. Hence whenever the computed error is non zero, the applied drive is maximum. Further investigations with respect to the system under consideration showed that some modifications had to be made to the above simplified relationship as shown in figure 4.5.

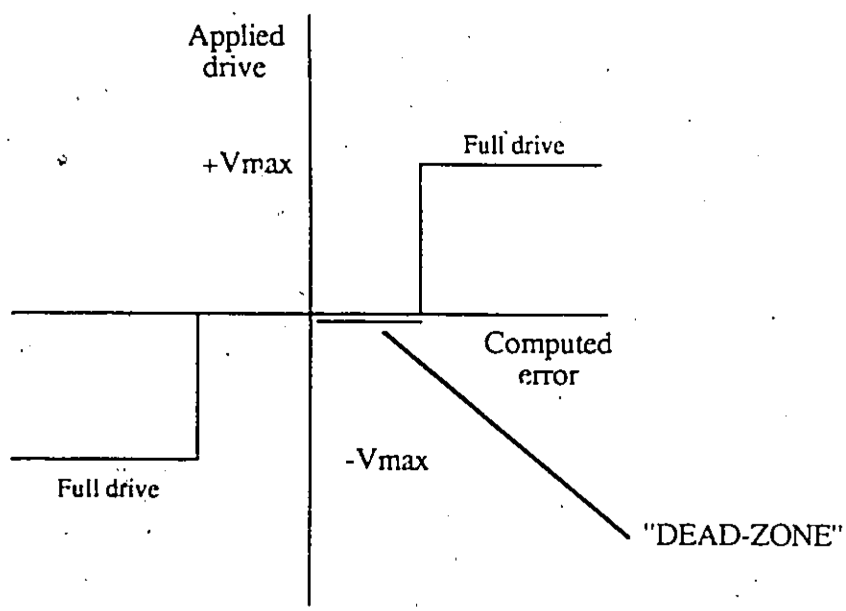


Fig. 4.5 Bang-Bang drive with a dead zone.

Due to digital rounding and quantization errors, potentiometer inaccuracy, and possibility of slight unintentional movements of the patient's shoulder, the desired position could be computed to have changed although no real shoulder movement was performed. This would result in undesirable jitter (or even oscillations) at the terminal device. Thus a "dead zone" is required to eliminate the jitter and possible hunting. Any computed drive falling in the dead zone will be ignored.

An apparent disadvantage when using this modified form of Pulse-width modulation is that it is not possible to operate the prosthesis at a different speed. However we have determined empirically that even with conventional Pulse-width Modulation or with linear analog control, the speed of the prosthesis was in general constant and close to the maximum possible whether large or small angles were to be traversed. Also due to the fact that the prosthesis is much slower than the physiological arm, the amputee will, in most cases, use the maximum speed of his prosthesis.

Equation (1) is used to calculate the error (or the drive) making use of the desired position and the present one.

$$ERROR = Xd - Xp \qquad (1)$$

Where : Xd = the desired position

Xp = the present position

The sign of the error will determine the direction of the drive. The minimum value of the error for it to remain in the "dead zone" was determined to be 5 (0000 0101). Any value greater than that would cause a drive to be applied for at least one time period T; which is the time it takes to go once around the software loop. If after this time "T" the error is less than "5" the drive is switched off, otherwise it is left on for another time "T" and so on. This time period "T", which is fixed at 20 milliseconds, was chosen to balance the need for a fast response of the prosthesis to the input from the amputee, and the necessity to allow the hardware sufficient time to react to the drive signals. Too small a time period would lead to excessive and useless computations since before the inertia of the system allowed the joint to move, the microcontroller would have had the time to go many times around the controlling loop, reading again and again the same desired and present positions. Too large a time period would lead the amputee to notice an appreciable time delay between his movements and those of the prosthesis. Also, hunting would be possible as the joints could pass the desired position before the program turned off the drive. A time period of 20 milliseconds was found to be adequate as it provided fast response of the prosthesis and a reasonable amount of computation.

The input-output relationship of the prosthesis was stored in memory so that the amputee could be provided with a variety of different movements also, called linkages. The next section exposes the principles behind these linkages.

4.2 The Linkages

The primary input to the system is the shoulder. The outputs of the system are the elbow and and wrist joints. In the case of linked motions, input and output have to be uniquely related by a one-to-one relationship. Regarding shoulder, elbow and wrist movements as angular displacements from a fixed reference, it is possible to represent the input/output relationship by:

$$B = Fi(\Theta)$$

Where Θ is the shoulder angle (the input angle) and B is the wrist or elbow angle (the output angle). The function Fi acting on Θ is set by us. It is, indeed, one of the possible linkages. This function Fi can, in theory, take the shape of any of the curves drawn on figure 4.6. Each of these curves would be the graphical representation of a movement. However the input/output given by curve (c) cannot be used in our prosthetic system. It is readily seen that more than one output can correspond to a single input. This fact, when translated to our system, means that to one shoulder position can correspond many different prosthesis positions. Such a situation can lead to extreme confusion as the amputee would never know exactly where his prosthesis would end up. This, obviously, is not tolerable. Thus in our system, there will always be a one-to-one relation between input and output. The only way to have a different output for the same input is by changing the linkage.

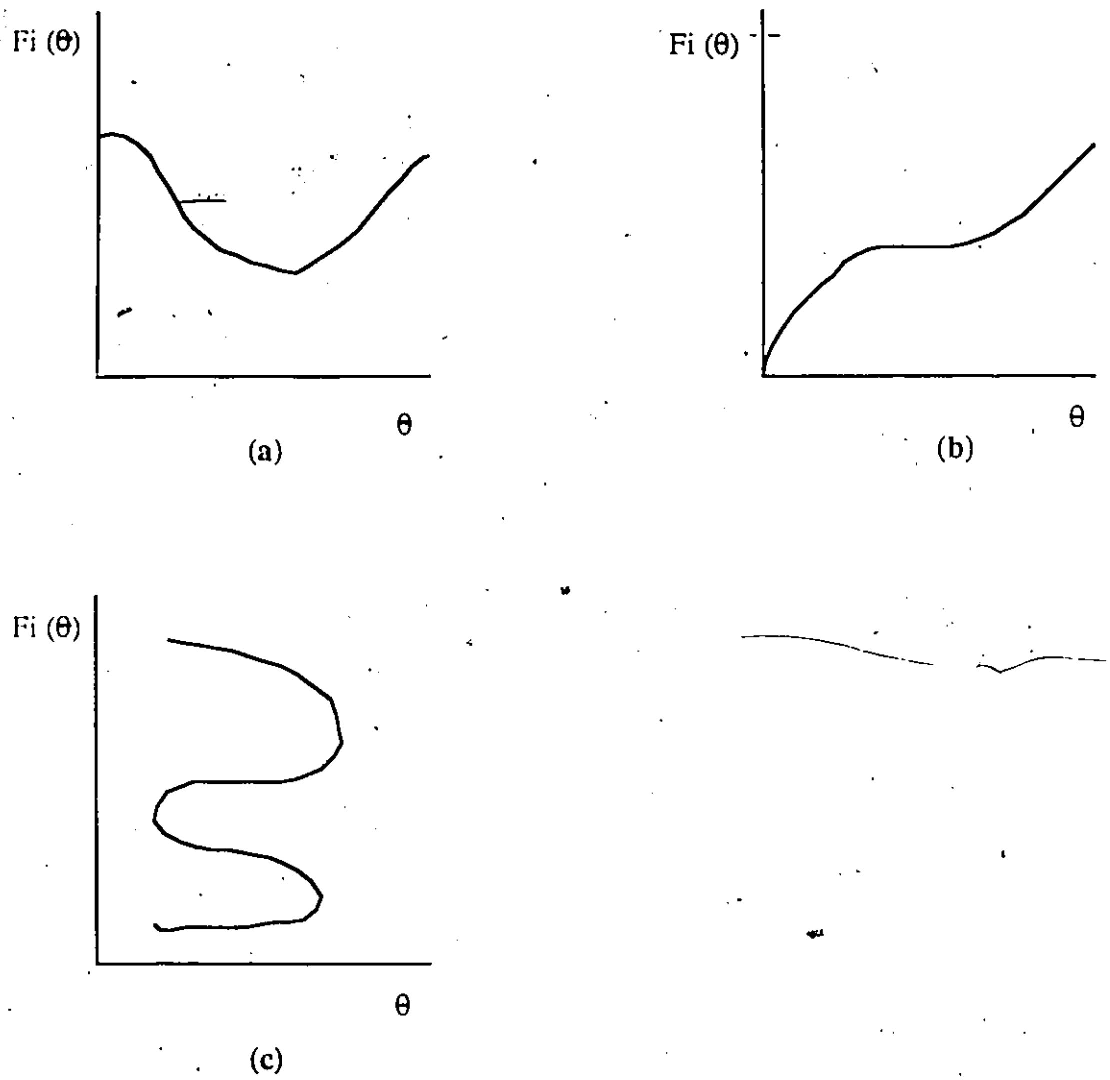


Fig. 4.6 Some possible input/output relationships.

It is usually very difficult to obtain an analytical expression for the function Fi, because coordinated motions are rarely functions with a closed form. In the case where an analytical expression could be found for all the functions Fi, there would be different ways to implement the linkages on a microprocessor-controlled prosthesis. The simplest way would be to store in the microprocessor memory all the functions Fi. Then when a certain task needs to be performed, the amputee selects the corresponding linkage. The processor finds the required Fi and calculates the output joint positions (B) as a function of the shoulder input (θ). This approach is very elegant but suffers from a number of problems.

- computation time would be tremendous, inducing delays in the prosthesis response. This is not acceptable in real-time prosthetic applications.
- analytical expressions for the Fi's are not easy to obtain.

Another way to implement the linkages is to use look-up tables. This method can be applied whether or not an analytical expression is available for Fi. If it is, the function is solved for all the possibles input θ, and the resulting B's are stored in the microprocessor ROM. If no analytical expression is available for Fi, an experimental approach can be applied as follows :

- The joints of the prosthesis are manually guided to perform the movement and strategic points are recorded at the same time. The movement can than be completely reconstructed, afterwards, using these points. This method is often referred to as "teaching the movement to the prosthesis". Here again to every θ the corresponding B is known, and thus it can be stored.

This solution would obviously remove the time delay previously mentioned, as the value of Fi(θ) would be immediately available. However this solution would require an excessively large memory. Assuming that each joint : shoulder, elbow, wrist is allowed 128 degrees of rotation and that one degree is mapped onto one word (8-bit), 2K of ROM would be required to store eight different movements ((128x8)x 2 joints). Thus half of the memory would be used just to store look-up tables.

At an early stage it was decided that eight different linkages should be sufficient for an amputee. However, as he gets used to his prosthesis the need for more functions or flexibility might emerge. Thus we decided to cut down the memory requirements of the look-up tables.

A way of reducing the large memory requirements was proposed by Gibbons and O'Riain [1], it consist of piecewise linearizing the function Fi. However an improvement to this method will be proposed here.
Formerly it was proposed to split Fi into eight equal regions, as shown in by figure 4.7. Each of these regions had a width of 16 degrees. To allow a piecewise linear reconstruction of the function the assumption had to be made that no sharp slope changes occurred within a 16 degrees envelope of the shoulder displacement.

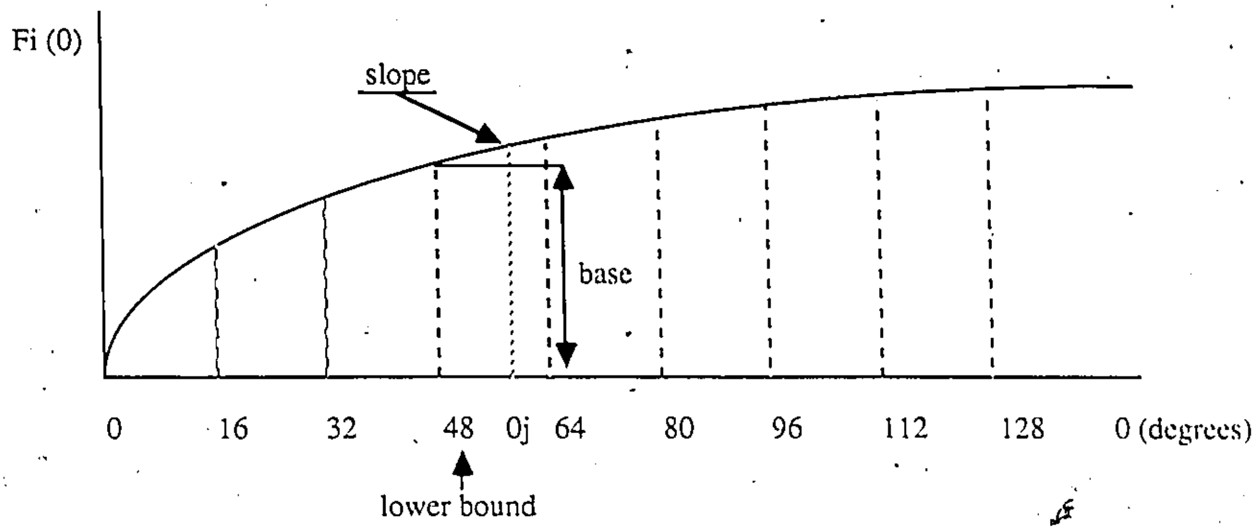


Fig. 4.7 The linearization of a curve.

Recent observations and tests have shown that a curve of the type given by figure 4.7 is less frequent then those given on figure 4.8. This proves that the functions Fi are often, in nature, piecewise linear. Rather than fixing the number of regions at eight, we propose now to leave it as a function of the slope changes of the curve. Thus the number of regions per linkage will vary. However, the maximum number of regions of a curve will be set at eight, as this number will usually provide a very good approximation to the original curve.

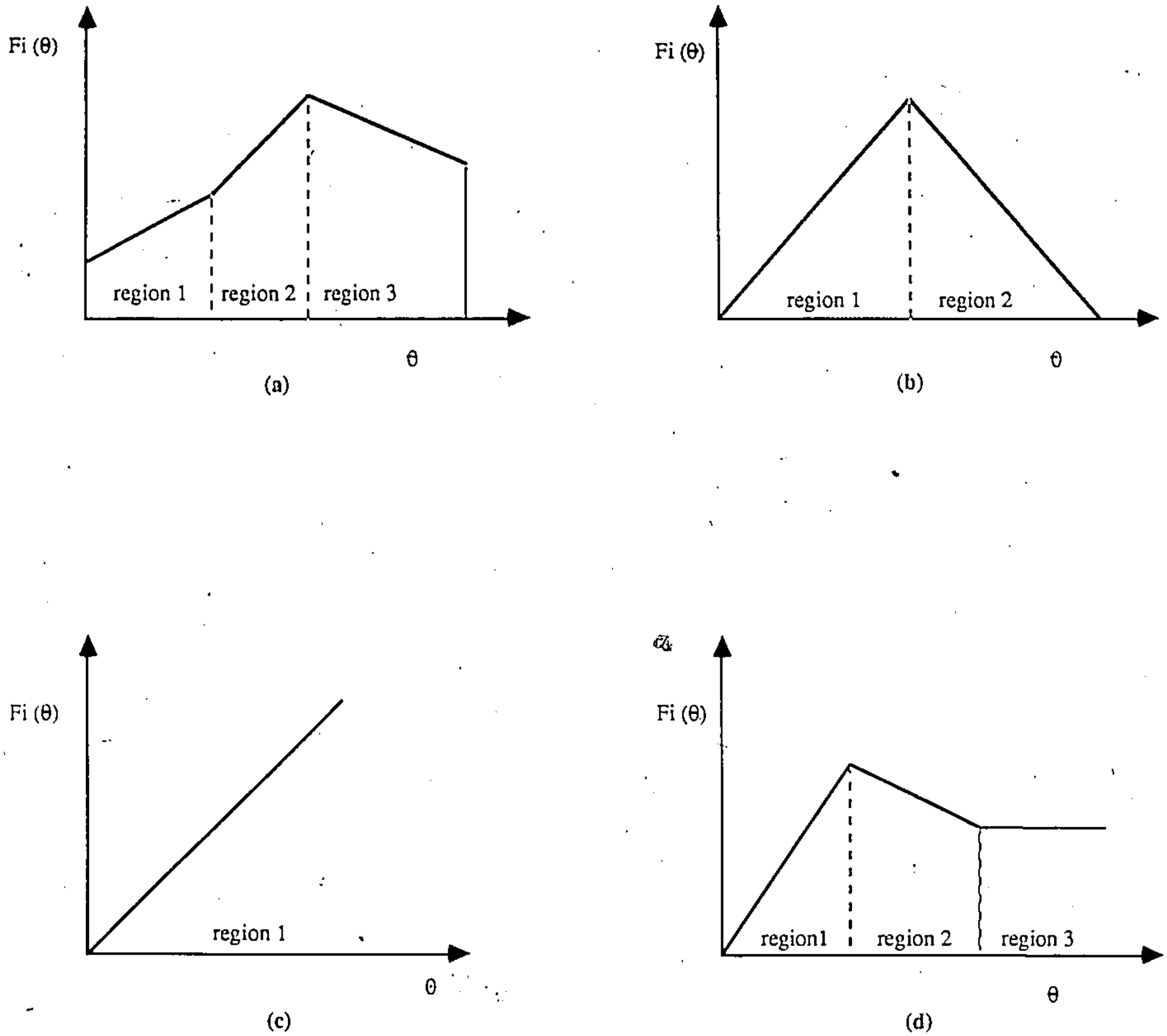


Fig. 4.8 Examples of common input/output curves.

The value Fi(θ) can be approximated for each region by the equation (2)

$$Fi(\theta) = a(\theta j - \theta k) + C \qquad (2)$$

Where a = slope

C = base

θk = angle at the beginning of the K^{th} region (also called the lower bound of the K^{th} region)

θj = input angle that falls in the K^{th} region

From equation (2) it can be seen that only three parameters have to be stored in memory, these are the slope, the beginning of each region and the base of each region. Assuming an average of four regions per linkage (notice that it is usually less) a total of 12 bytes per linkage per joint is expected. For two joints 24 bytes are required per linkages. Hence, 8 linkages can be stored in 192 bytes. This is 10 times less than what was required when no piecewise linearization was done. But, because the slope was not always an integer value and no floating point or fixed point arithmetic was to be programmed, we had to introduce some correction factors for each region to make sure that the accuracy, when computing equation (2), was maintained. This increased the byte requirements per region to four, and the total memory storage for the linkage table to 256 bytes, which is still far less than the 2K byte required if no piecewise linearization is applied.

4.3 An Example of a Linkage Table

In this section an example is given of how a linkage table is implemented. The movement that we would like to simulate is that of an above-elbow amputee using his prosthesis to pick up a cup of tea and bring it to his mouth. It is assumed that the prosthesis will be used by an amputee who is approximately 1.80 meter (6 feet) tall. The subject is seated in front of a table.

To model the prosthetic arm, the following assumptions were made :

L1 = 25 cm (10 in.) (distance between shoulder and elbow joints)

L2 = 30 cm (12 in.) (distance between elbow joint and the palm)

L3 = 22.5 cm (9 in.) (distance between shoulder joint and table)

H = 5 cm (2 in.) (height of the handle of the cup from table)

Figure 4.9 gives a graphical representation of the prosthesis when it has reached the cup.

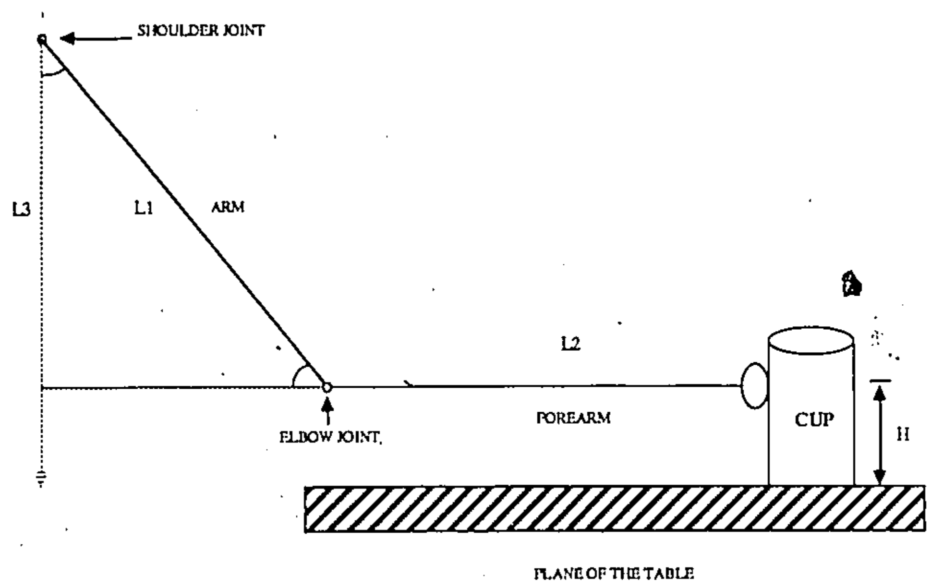


Fig. 4.9 The prosthesis position when the cup is grasped.

The motion is broken down and illustrated on figure 4.10 (a)-(e). From (a) to (c) the amputee extends his shoulder which causes the elbow to extend and the wrist to rotate from a position of palm up towards a position of palm down. When the hand reaches the cup, the shoulder, upper arm, and forearm are in the position given by figures 4.10 (c) and 4.9. The wrist has rotated from palm up (0 degrees) to palm half way between up and down (90 degrees) and will stay there as the movement continues. The biceps contracts inducing prehension: the cup is grasped. Now further extension of the shoulder will induce a flexion of the elbow (notice that the elbow joint changes direction as the slope of the elbow curve changes sign, as shown in figure 4.11). However, the wrist will remain in the same position (that is, holding the cup upright), until the mouth is reached (figure 4.10 (e)).

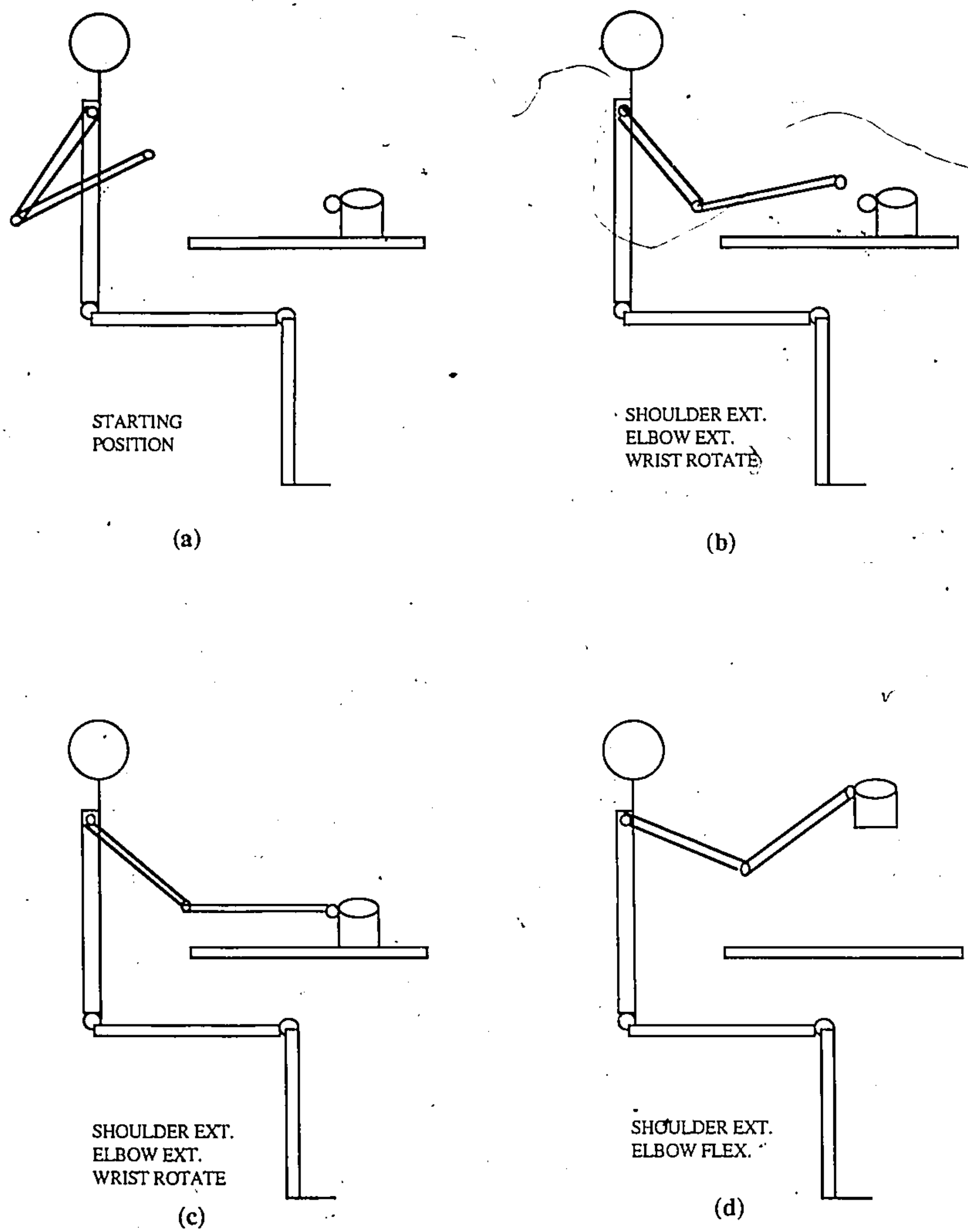
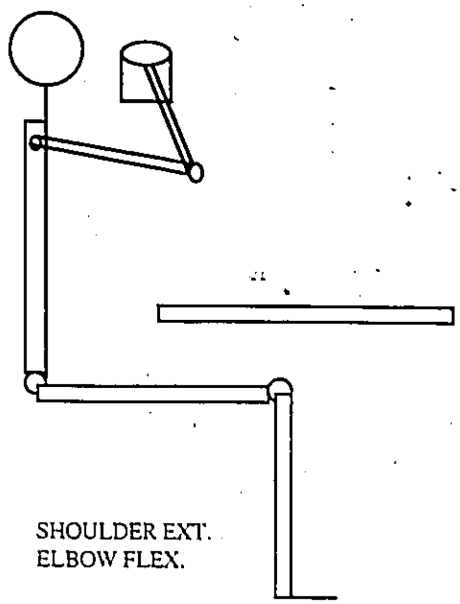


Fig.4.10 The decomposed motion of : reaching a cup and bringing it to the mouth.



(e)

Fig. 4.10 The decomposed motion of : reaching a cup and bringing it to the mouth.

The input/output relationship, or the function Fi, for this particular coordinated motion is given on figure 4.11.
On the abscissa (X-axis) we have the angular displacement of the shoulder and on the ordinate (Y-axis) we have the output angles for elbow and wrist.

The linkage was constructed empirically by observation of a natural human arm performing the same movement. The joint motions were carefully decomposed so that angles could be recorded at strategic points and the input/output curves implemented for elbow and wrist. As can be seen in figure 4.11, these curves are simple and already piecewise linear. Thus, to construct the linkage table for this movement we need only to store : the

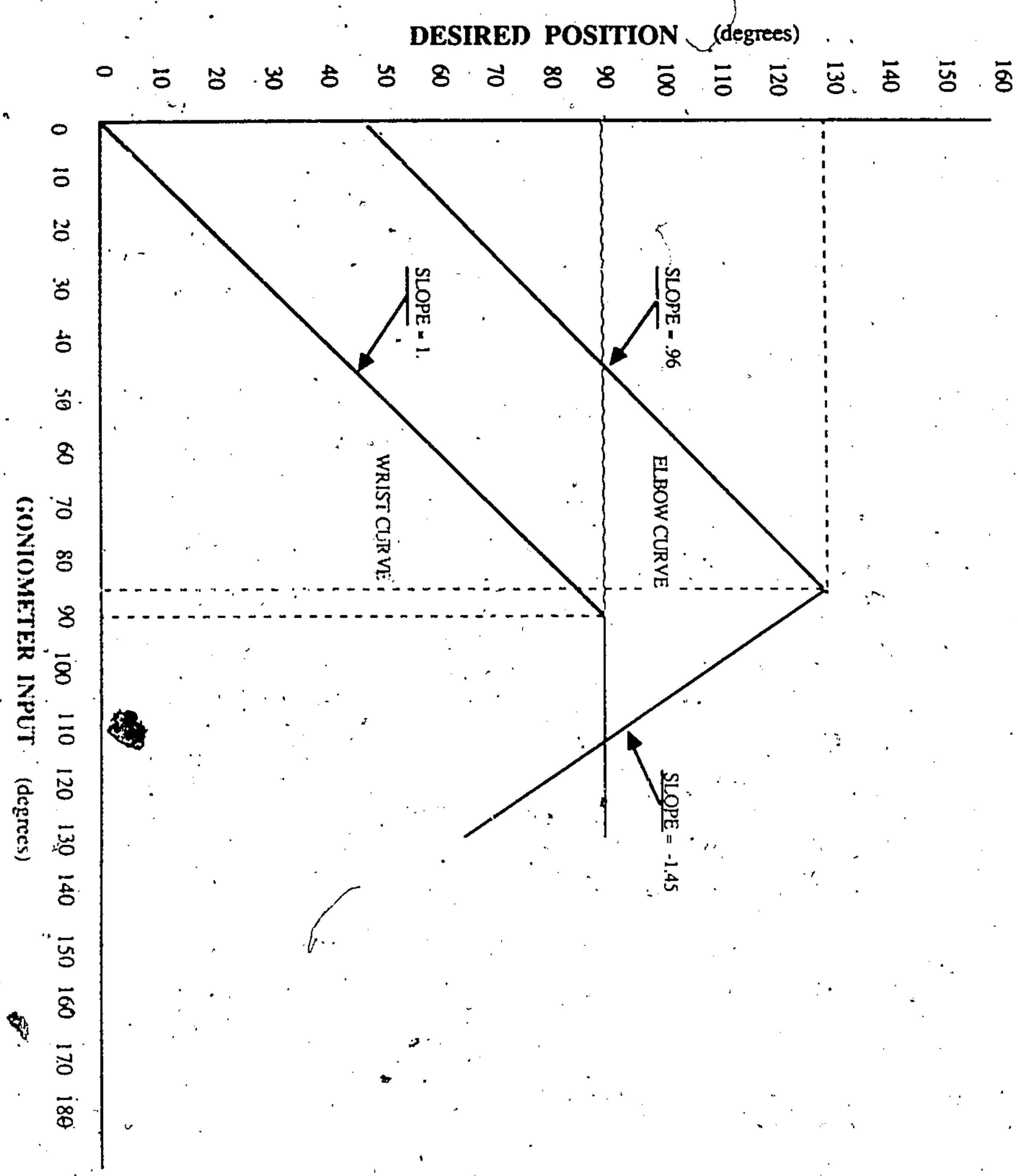


Fig. 4.11 : Input / output relationship for elbow and wrist.

smallest angle at the beginning of each K^{th} region (also called lower bound of the region) the slope, the base, and the correction factor.

The lower bound, the base and the correction factor are all initially integer values. They can readily be converted to 8-bit unsigned binary values. However, the slope is initially a real value. We decided to convert it to a 5-bit integer value (see figure 4.12). Five bits are more than enough to represent the slope which rarely is greater than unity. The remaining three bit are used as control bits.

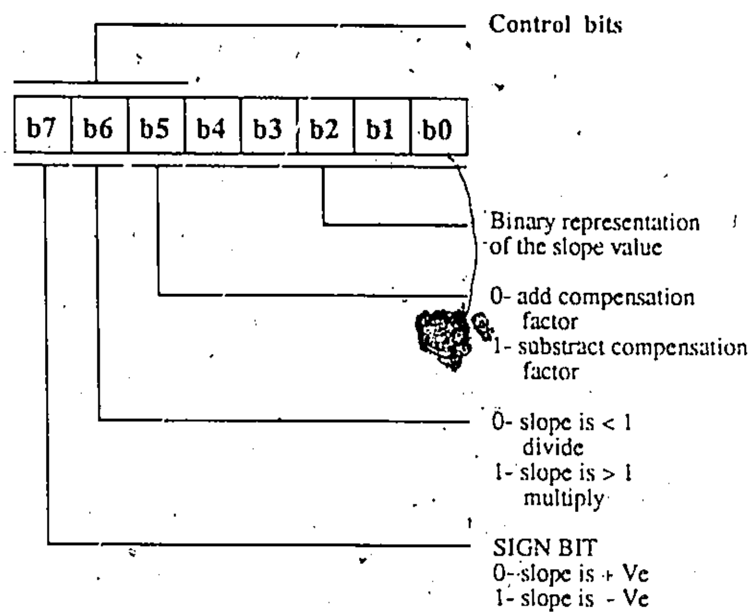


Fig. 4.12 Binary representation of the slope :
5 bits for magnitude
3 bits for control

To be able to compute the desired position $B = Fi(\theta)$, the slope, the lower bound of the region, and the base are obviously needed (see equation 2).

$$Fi(\theta) = a(\theta j - \theta k) + C \qquad (2)$$

a = slope

C = base

θk = lower bound of the Kth region.

However, the correction factor is needed to adjust the result of the operation $a(\theta j - \theta k)$ only in the case where the slope was not, initially, an integer value. In this case, the value of the slope is converted to the nearest integer value (up or down) and bit five (see figure 4.12) is set accordingly to compensate for the adjustment made to the slope value. When no correction is needed the correction factor is zero and the status of bit 5 of the slope is irrelevant. When the correction factor is not zero then the status of bit 5 will determine whether correction will take place up or down. This method proved to be simple and efficient when tested. It made our programming easy since no arithmetic (fixed point or floating point) was required to perform the calculations.

Using the above mentioned approach, we constructed the linkage table of the movement that enables the amputee to " reach for a cup and take it to his mouth" (see table 4.13). Both wrist and elbow have only two regions of operation, which simplifies greatly the table. However, tables for more complex movements are usually longer but they will not be more complicated than the one given on table 4.13.

BINARY VALUE	HEX VALUE	MEANING OF THE ENTRY
0000 0000	0 0	Region 0 lower bound (0 deg.)
0110 0001	6 1	Region 0 slope (1)
0010 1101	2 D	Region 0 base (48 deg.)
0000 0101	14	Region 0 correction factor (20D)
0101 0000	5 0	Region 1 lower bound (85 deg.)
1010 0001	A 1	Region 1 slope (-1)
0111 1011	7 B	Region 1 base (130 deg.)
0000 0010	0 2	Region 1 correction factor (2)

Linkage table for the elbow.

BINARY VALUE	HEX VALUE	MEANING OF THE ENTRY
0000 0000	0 0	Region 0 lower bound (0 deg.)
0010 0001	2 1	Region 0 slope (1)
0000 0000	0 0	Region 0 base (0 deg.)
0000 0000	0 0	Region 0 correction factor (0)
0101 0101	5 5	Region 1 lower bound (90 deg.)
0100 0000	4 0	Region 1 slope (0)
0101 0101	5 5	Region 1 base (90 deg.)
0000 0000	0 0	Region 1 correction factor (0)

Linkage table for wrist.

Table 4.13 The linkage tables for the movement of reaching a cup and bringing it to the mouth.

CHAPTER V

PERFORMANCE OF THE SYSTEM

5.1 General Comments

In this research work, our main goal was to prove that the control strategy known as "Extended Physiological Proprioception" first proposed by Simpson [4], could be used to control an electric prosthesis. There are various reasons why we decided to use electrical actuators for our above elbow prosthesis, as opposed to Simpson, who decided to use pneumatic actuators and carbon dioxide to power his prosthesis. Indeed, his design was very successful, and is still considered as the reference when powered prostheses are mentioned, but it had the following limitations:

1. The prosthesis was carbon dioxide powered, this required the carrying of a heavy compressed gas cylinder. Although the weight of the cylinder was compensated for by relatively light actuators, the carbon dioxide had a short life and replenishment presented some difficulties as compressed gas is not a common household item. Also, the activation of carbon dioxide powered joints tends to be noisy and attracts the attention of persons nearby.

2. The input/output relationships were fixed mechanically and were therefore difficult to change.

3. Some movements required complex input combinations which were difficult for the amputee to produce.

On the basis of the above considerations, we designed a prosthesis that used the "Extended Physiological Proprioception" as the control modality, but, as opposed to Simpson, all actuators were electrically powered. Nickel-cadmium rechargeable batteries have the best power-weight ratio (see table 5.1) of the common power sources for prostheses, have a long life during normal use, and are easily recharged using domestic electricity [3].

A American Institute for Prosthetics Research bottle pack with integral regulator, 112 g CO2.
B Aluminium alloy with regulator attached 31 g CO2.
C Steel cylinder with regulator attached 120 g CO2.
D Nickel cadmium battery 12 V, 5 Ah.

<table>
<tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th></tr>
<tr><td>Mass of bottle, in g</td><td rowspan="2">590</td><td>130</td><td>420</td><td>---</td></tr>
<tr><td>Mass of regulator, in g</td><td>100</td><td>100</td><td>---</td></tr>
<tr><td>Mass of gas, in g</td><td>112</td><td>31</td><td>120</td><td>---</td></tr>
<tr><td>Total mass, in g</td><td>702</td><td>261</td><td>640</td><td>260</td></tr>
<tr><td>Energy stored, in kJ</td><td>0.50</td><td>0.13</td><td>0.52</td><td>13</td></tr>
</table>

Fig. 5.1 Comparaişon of some of the existing types of energy store.

The use of electricity as the main source of power also gives the advantage of being able to use a microcomputer to control, and supervise the activities of the prosthesis. Hence, the input / output relationship are not fixed mechanically any more. They can be varied to the convenience of the amputee. Eight different movements are provided, but if they are to be modified, only the software has to be altered. This flexibility greatly enhance the use of the prosthesis, and makes it more attractive, from this point of view, than Simpson's design. The use of the microcomputer also gave us the opportunity to provide a fail-safe product to the amputee. The unbeatable servo feature and the power supplied to the electronic circuit are constantly monitored by the microcomputer. If, at any moment in time the prosthesis cannot respond to a command or the batteries are getting dischaged, or there is an open circuit in the wiring, the amputee is alerted of the malfunctioning of the circuit. He then can take the necessary action to solve the problem.

In our design, the problem of having to provide complex input combinations in order to position the terminal device (the hand) was also overcome by using the modified form of e.p.p., that is: by using only one joint to control the complete prosthesis (instead of having one intact joint to control every prosthetic joint as Simpson did) we could be sure that, whatever the level of complexity of the movement, as long as it is one of the preprogrammed linkages, the level of difficulty and thus the concentration to perform this movement would always be the same. The microcomputer takes care of activating each joint at the right moment and makes sure that the coordinated motion is well performed. This certainly takes off the burden on the amputee, who can concentrate on something else.

Tests have been performed with the prototype prosthesis strapped to the side of the arm of various investigators while performing a variety of daily living and occupational tasks. These have demonstrated the effectiveness of e.p.p. and its advantages over open-loop control systems.

The tests were performed as follows :

An adaptation time of five to ten minutes was given to the investigator, then with his eyes blindfolded he was asked to place the prosthetic hand in different positions in space, also he was asked to specify where it was. Not more than five minutes were required for him to get used to one linkage. Having remembered the length of each joint (which was set to same length as his own natural joints for convenience) he could, as a function of his shoulder position, tell immediately where the prosthetic hand was. The sensation of proprioception was continously present. It is not possible to say that the investigator had the same proprioceptive sensations with the artificial hand than he had with his physiological one, but the result was the same : no visual feedback was needed to know where it was in space.

Switching between different linkages was not a problem as long as a certain learning time was given to the investigator. This learning time was required only once per experimenting session and strangely recalled the "warming up" of a tennis player that swings his racket back and forth before a game in order to, once again, merge the racket and himself into a single entity. It is believed that the amputee after wearing his prosthesis a couple of times will get used to the linkages and no further "warming up" time, or adaptation period, will be required; the movements and reactions of the prosthesis will become his.

From a practical point of view, the prototype prosthesis should be quite efficient. As an assistant to a valid arm and hand, the prosthesis helped to position and hold in place various objects. When used in repetitive tasks that involved the simultaneous movement of shoulder and elbow (e.g. movements that are similare to that involved when painting a wall) the prosthesis was a faithful replacement to a physiological arm.

In the exercise of reaching a cup or a can and taking it to the mouth (this example was illustrated in the previous chapter), the prosthesis revealed to be able to mimic closely the movements of a natural arm. Even though smoothness and dexterity of the human arm were not quite there. specially when performing small amplitude movements, as the error (Xd - Xp) gets bigger than the "dead zone", the elbow suddenly jumps out of it. These jumps are sometimes too jerky which shows a lack of smooth and fine positioning of the prosthetic elbow. It is believed that this is partly due to the relative age of the motor* of our Boston elbow that has been intensively used in the past years for prototyping. With a new motor this jerky behavior it expected to disappear.

* See Appendix D

The speed of the artificial elbow is less than that of a biological one, but we found its average of 120 degrees/second to be sufficient, as it gives a bit more time to direct and position the terminal device.

Before commenting on the software, it would be interesting to have a look at some specifications on the basic hardware.

With all the components working all the time, the average power consumption of the controlling system was about 10 milliamperes. Most of the power is taken by the ADC (5 milliamperes) and the microcontroller (5 milliamperes see [21] for CMOS equivalent version of 8751). However, when the microcontroller is in the stand-by mode its power consumption is reduced to 1 mA .
-The joint actuators consume about 1.2 to 1.5 amperes (maximum) when in motion. Most of the actuation power is taken by the elbow joint, which in return provides a good lifting strength. We also appreciated the reverse locking clutch on the elbow motor that allows it to carry a weight without actually having to drive the motor. From our own measurements four hours of normal daily activities can be accomplished on one set of batteries. The batteries will recharge in 15 minutes. Thus ,for a normal day of 8 hours of work a single recharge during the day, say at lunch time, should be sufficient.

In running the various tests most of the attention was basically focussed on the elbow joint, the reason is that it does most of the hard work. However the wrist and the hand units gave complete satisfaction. The wrist speed (90 degrees/second) force, and fine positioning were adequate for our design. The hand unit, as previously mentioned is just a pinch prehension device, individual movements of the fingers is not provided. This is common practice in prosthetics as the individual coordination of the fingers is extremely difficult and is commonly not used.

The important criteria to judge a prosthetic terminal device are: the pinch force and the closing and opening time. In our case both were good. The prehension force was a lot greater than that of an average human being, and the opening and closing time were always less or equal to one second.

From a software point of view the design was completely bug free. 600 lines of codes in assembler language were used. With the 12 mega-Hertz crystal giving an average execution time of 1 microsecond, no delay could be noticed. The prosthesis responded instantaneously to the input. The only limiting factor was the mechanical inertia. Certainly the use of look-up tables helped greatly in speeding up the system, since this minimized the calculations.

5.2 The Microprocessor Controlled Above-Elbow Prosthesis : a challenging commercial product.

As it is now our above-elbow prosthesis is a good prototype. However there are too many attractive features in this device for it to remain a simple prototype. To our knowledge there are no other operating microprocessor controlled above-elbow prosthesis using e.p.p., which is, at the moment, the most adequate way to control a prosthesis. If we compare our design to what is commercially available on the market today, there is no doubt that we have a competitive product in our hands. There are essentially two major manufacturers that develop and sell above-elbow prostheses. The first one is the Liberty Mutual Assurance company that sells the Boston elbow, and the second one is the University of Utah.

In the Liberty Mutual prosthesis, the mechanical parts and the electric motor are exactly the one that we have used. The difference lies in the control strategy. The Liberty Mutual elbow is controlled by myoelectric signals from the biceps and triceps muscles. The technique used is called proportional myoelectric control. A small amount of tension on the biceps will cause the elbow to move slowly. Increased muscle activity will move it faster.

The opening and closing of the terminal device, which can be either a Greifer or an electric hand, is controlled by the shoulder via cables.

The Liberty Mutual system, like ours, is simple, easy to learn and easy to use, however it suffers from two major problems :

- only one degree of freedom is provided (not counting the hand prehension), thus no coordinated motion is possible.
- no form of feedback is provided to the amputee, which makes this system totally open-loop. Thus the operation of

this device requires constant visual monitoring from the user.

The other group that has developed and sells arm prostheses is the one at the university of Utah. Qualitative judgement of this prosthesis was difficult as it was not available for tests. In any case, it also makes use of proportional myoelectric control to operate the various degrees of freedoms, but here again, the system runs in open-loop since no feedback is provided to the amputee. The Utah arm is certainly more advanced and flexible than the Liberty Mutual Boston elbow as it provides a possibility to have a myoelectric controlled wrist and terminal device (hand or Greifer) or passive wrist rotation and cable operated terminal device. But here again, there is the difficulty of executing coordinated motion of shoulder, elbow, and wrist.

From the above discussion it can be concluded that our microprocessor-controlled device has strong arguments to challenge the available products on the market. Our control strategy has proven to be among the most efficient ones. We strongly believe that with more financial support and technical assistance this prosthetic arm could become a valid commercial product.

GLOSSARY

The following are some of the important prosthetic and scientific terms used in this thesis.

Abduction	: to move or pull part of the body (in the context of this paper: the shoulder joint) away from the median axis.
Adduction	: to pull a part of the body (in the context of this paper : the shoulder joint) toward the median axis.
Amelia	: see bilateral amelia.
Bilateral amelia	: bilateral amelia, designates an individual with amputations on both sides of the body.
Desarticulation (shoulder)	: A shoulder desarticulation amputee is a patient that has an amputation at the shoulder joint.
EMG	: stands for "Electromyographic" signals. Electrical signals resulting from the contraction or expension of muscular tissue.
e.p.p.	: stands for "extended physiological proprioception". Controlling concept, first intro-

	duced by Simpson, and used by him to control artificial limbs. It is based on extending the proprioceptive sensations present in intact joints to the artificial ones.
Goniometer	: instrument used to measure angles.
Griefer	: terminal device used to replace the hand, it is just composed of a pinch prehension mechanism (i.e there are no independent finger movements).
Linkage	: relationship existing between the input and output of the prosthesis.
Myoelectric	: same as EMG.
Multifunctional prosthesis	: A prosthesis having many degrees of freedom; that can perform various movements.
Optimal-linear	: term used, in this context, to characterize the fact that all the frequencies of the given signal are used as useful information.
Pronate	: to rotate (the hand or forearm) so that the palm faces down or toward the body.
Pronation	: see pronate.
Proprioception	: position sensation, see proprioceptive.

Proprioceptor	: sensory end organ in the muscles, tendons, etc. that are sensitive to the stimuli originating in these tissues by the movement of the body or its parts.
Proprioceptive sensations	: stimuli produced in body tissues, as muscles or tendons, and received there by the proprioceptors.
Supinate	: to rotate (the hand or forearm) so that the palm faces upward or away from the body.
Supination	: see supinate.
Synergetic	: working together; cooperating.
Terminal device	: commonly used in prosthetics to designate the artificial hand or what replaces it (e.g. hook, Griefer etc...).
Unbeatable servo	: concept introduced by Simpson to prevent the input of a servo mechanism to go faster or further than the output.

REFERENCES AND BIBLIOGRAPHY

[1] J.A.Agopyan, D.T.Gibbons, "A Microprocessor Controlled Above-Elbow Prosthesis", Master's thesis, Concordia University, Montreal, Quebec, Canada, june 1983.

[2] D.S.Childress, "Historical Aspect of Powered Limb Prostheses", Clinical Prosthetics & Orthotics, Vol.9, No.1, pp.2-13, 1985.

[3] S.R.Montgomery, "External power systems for the upper extremity", in MURDOCH, G. (Ed.) "Prosthetic and orthotic practice", Edward Arnold, London, 387-397.

[4] D.C.Simpson, "The choice of control system for the multimovement prosthesis: extended physiological proprioception (e.p.p.) ", in "The Contol of Upper-Extremity Prostheses and Orthoses", Eds. P.HERBERTS, R.KADEFORS, R.MAGNUSSON, and I.Petersen, pub. C.C.Thomas, 1974, ch.15.

[5] R.W.Wirta, D.R.Taylor, F.R.Finley, "Pattern recognition arm prosthesis: a historical perspective- A final report", Bull. of Prosthetics Research, pp. 9-35, Fall 1978.

[6] D.Graupe, J.Magnussen, A.A.Beex, "A Microprocessor System for Multifunctional Control of Upper-Limb Prostheses Via Myoelectric Signal Identification", IEEE Trans. AC, Vol. AC-

23, No.4, pp. 538-544, 1978.

[7] D.T.Gibbons, M.D.O'Riain, "An Above-Elbow Prosthesis Employing Programmed Linkages", proposed paper to IEEE Trans.BME, 1986.

[8] P.Lawrence, "Computer design and simulation of a myoelectric system for controlling a multifunctional prosthetic hand", in Proc. Symp. on Pattern Recognition And Image Processing, 1972.

[9] T.Lyman, A.Freedy, and M.Solomonow, "Studies toward a practical computer-aided arm prosthesis system", Bull. Prosth. Res., pp. 213-225, Fall, 1974.

[10] S.C.Jacobsen and R.W.Mann, "Control systems for artificial arms", IEEE Con. on Systems, Man and Cybernetics, Nov. 1973.

[11] R.W.Wirta and D.R.Taylor,Jr., "Development of a multiple-axis myoelectrically controlled prosthetic arm", Adv. Extern. Control Human Extremities: Proc. 3rd Int. Symp. on External Control of Human Extremities (Dubrovnik,1969), pp.245-253, Yugoslav Comm. Electron. Automat., Belgrade, 1970.

[12] D.C.Simpson, "The control and supply of a multimovement externally powered upper limb prosthesis", Proc. 4th Intl. Symp. on External Control of Human Extremities, Belgrade,

247-254, 1973.

[13] J.A.Doubler and D.S.Childress, "An analysis of extended physiological proprioception as a prosthesis-control technique", Jour. Rehabilitation Research, 21:1 pp. 5-18, 1984.

[14] Y.Lozac'h, E.D.Sherman, G.Gingras, " A New Versatile Myoelectric Proportional Control Unit". Bulletin of Prosthetics Research - Fall 1971.

[15] H.W.Ladd, Y.Lozac'h and M.D.O'Riain, "Development of Myoelectrically Controlled Prostheses and Orthoses at the Rehabilitation Institute of Montreal", Canadian Clinical Engineering Conference, pp.75-77, 1977.

[16] National Semiconductor, "Linear Data Book". MM54HC4042 ANALOG MULTIPLEXER pp. 4-361 to 4-366.

[17] "Analog Devices", Data Book. AD7574 ANALOG to DIGITAL CONVERTER VOL. 1, pp 10-151 to 10-158.

[18] Intel MCS-51, "Family of Single Chip Microcomputers User's Manual".

[19] Motorola, "TMOS Power MOSFET Data Book", MTM25N05 TMOS POWER FIELD EFFECT TRANSISTORS.

[20] E.Ulen, and M.Wagner, "Myoelectric Control for Pneumatic

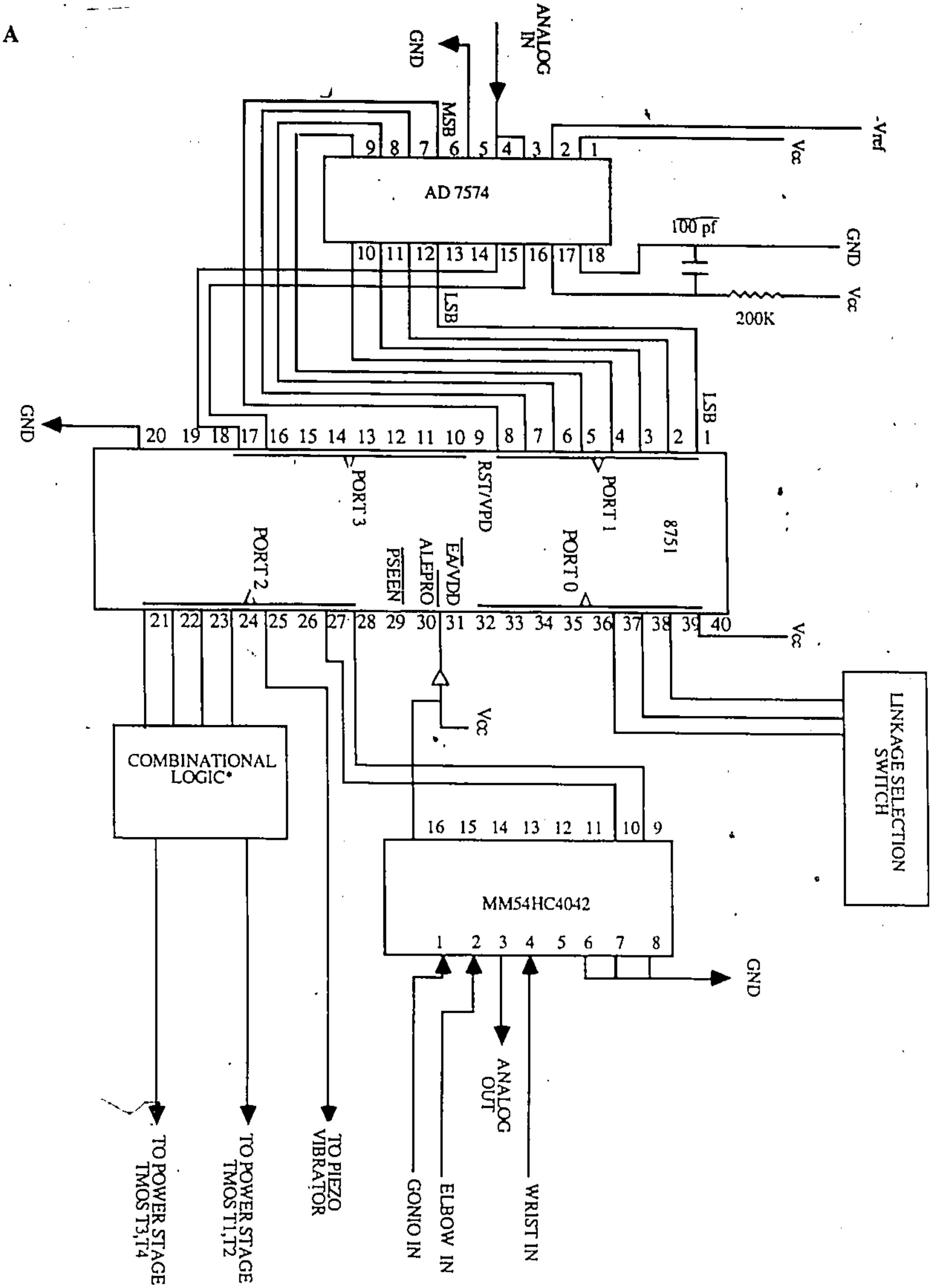
Prosthesis, Annual Report, Rehabilitation Institute of Montreal, Vol. v, Oct. 1969.

[21] NEC Electronics (EUROPE) GmbH, "1982 CATALOG", see 8748 specifications for stand-by power consumption pp. 361-370.

[22] W.E.Snyder, "Industrial Robots: Computer Interfacing and control", Industrial Robots Series, pp. 76-80, 1985.

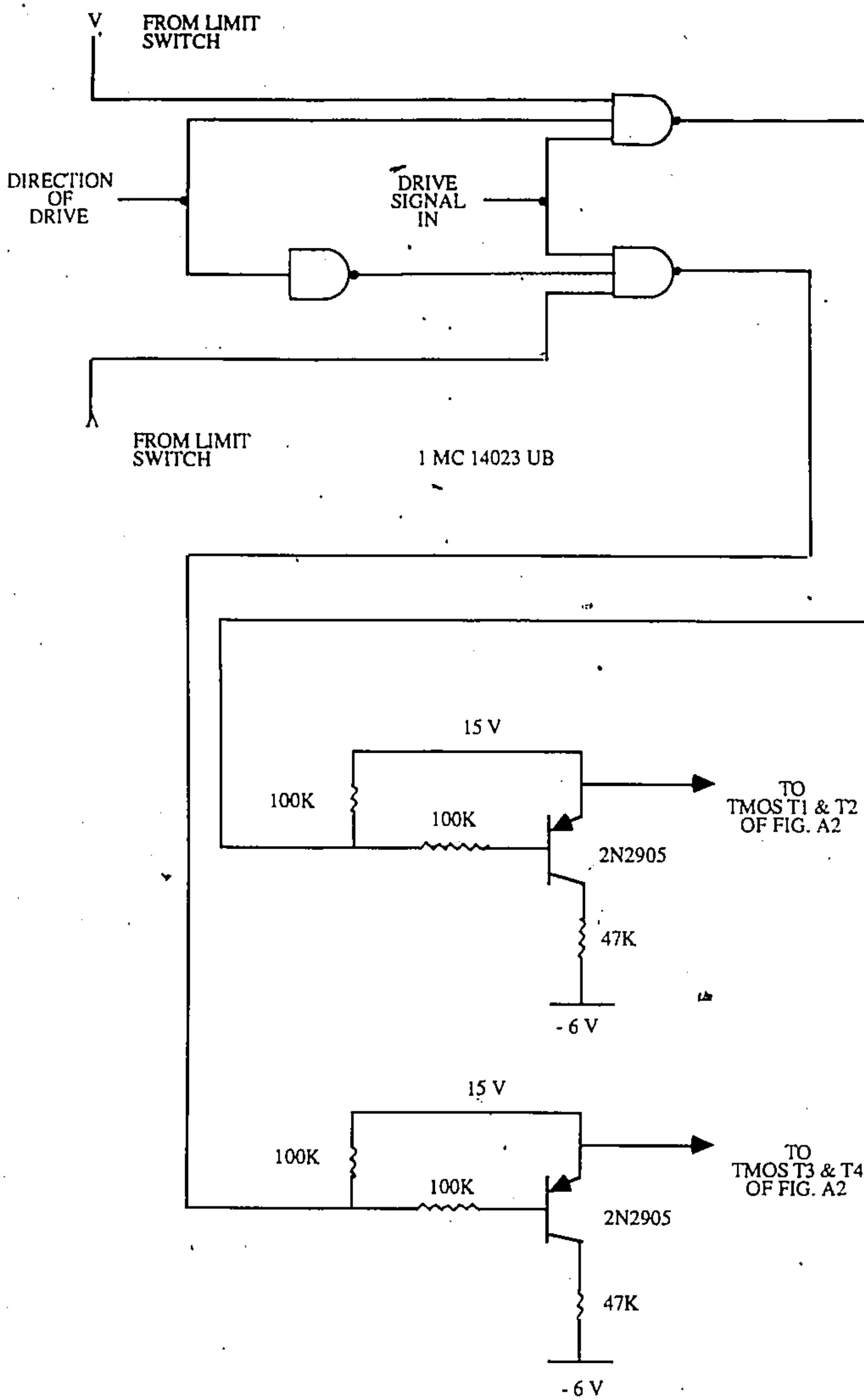
[23] B.C.Kuo, "Incremental Motion Control: Step Motors and Control Systems", Univesity of Illinois at Urbana-Champaign, pp. 6-7, 1979.

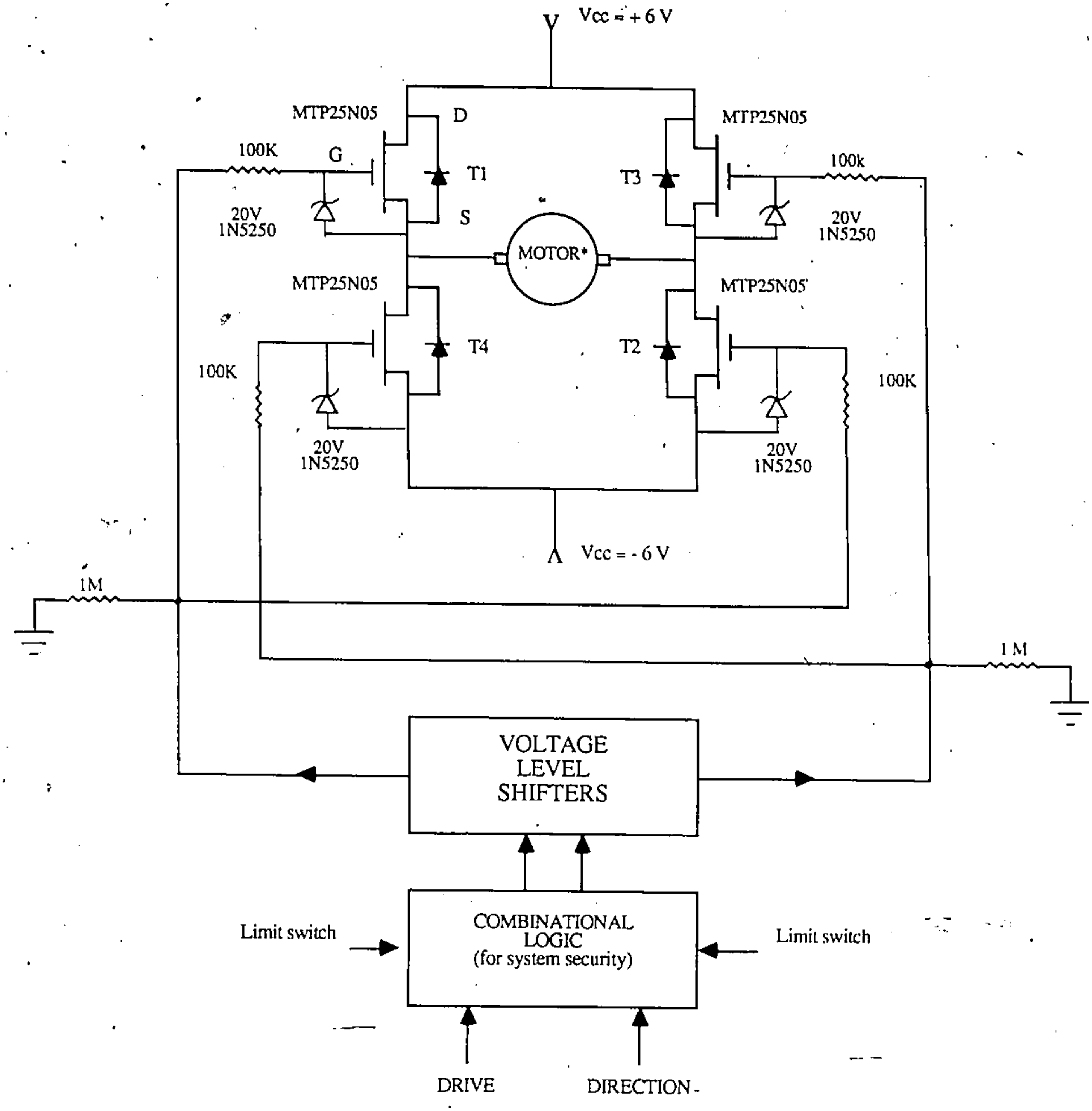
Appendix A



A.1 The controlling circuit.
*See page 88-A

COMBINATIONAL LOGIC





A. 2 Power stage and logic circuit
* Larcom D-1500-C12/K014

Appendix B

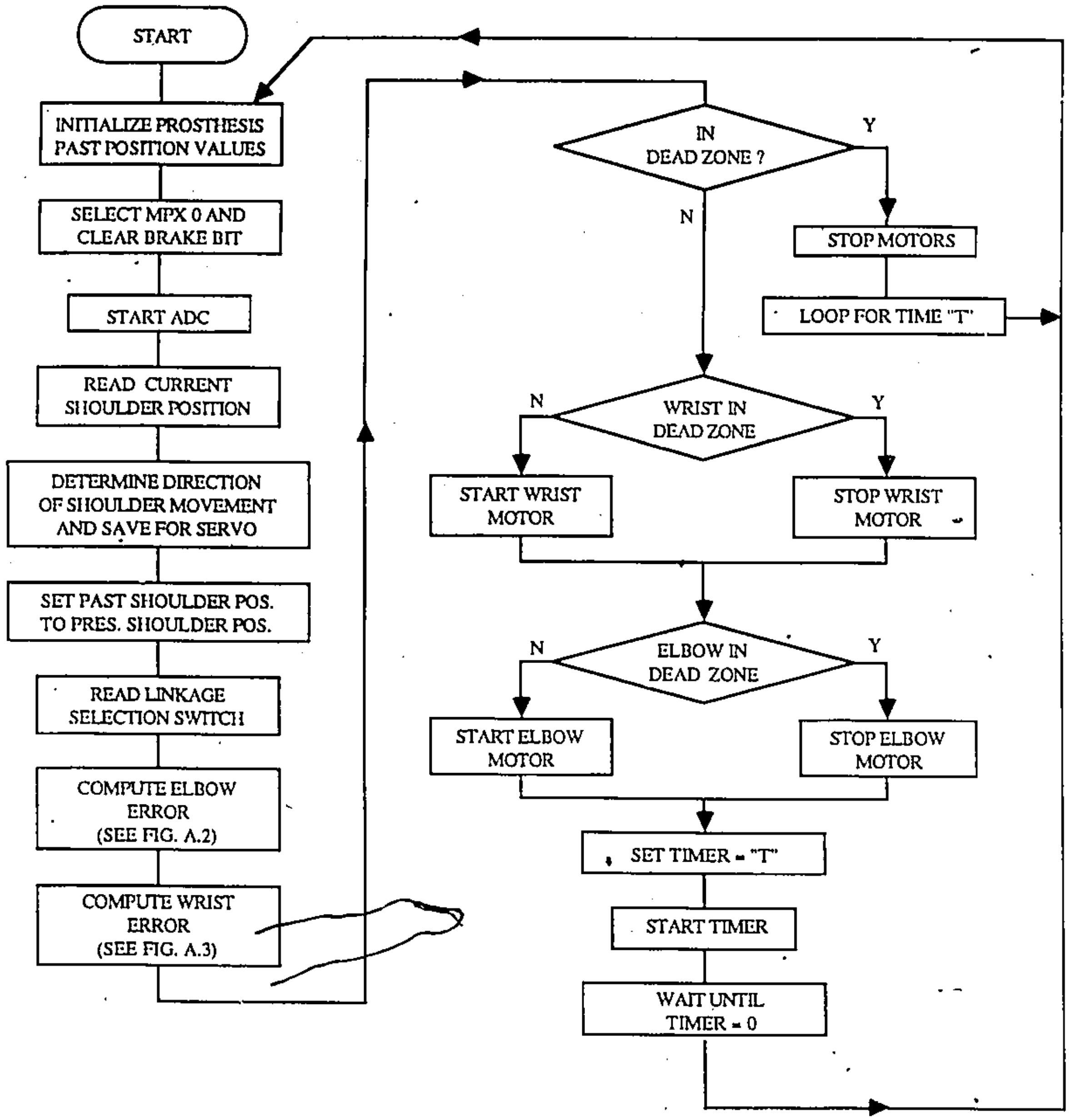


Fig. B.1 Flowchart of the prosthesis control algorithm.

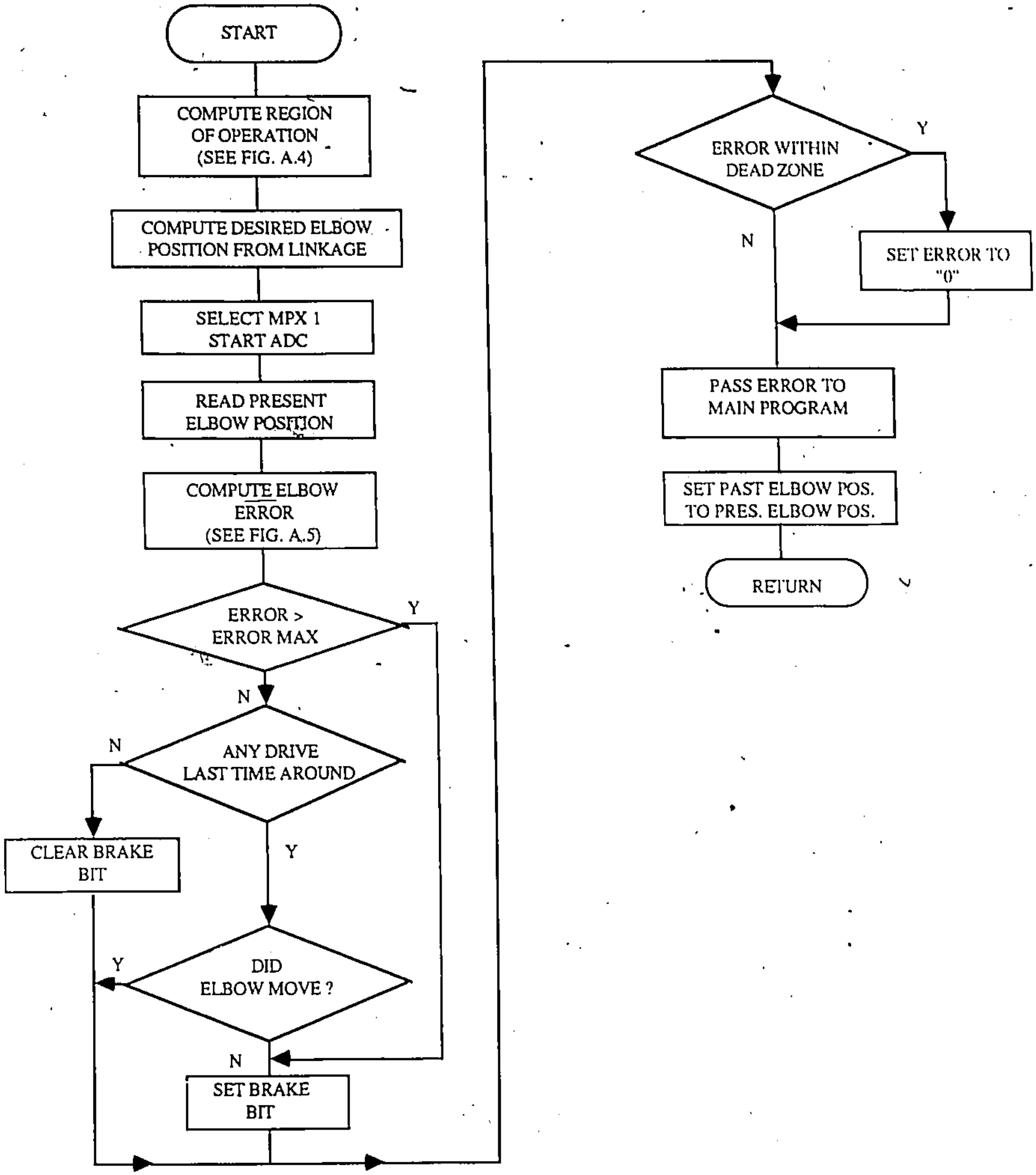


Fig. B.2 Flowchart for the computation of the elbow error.

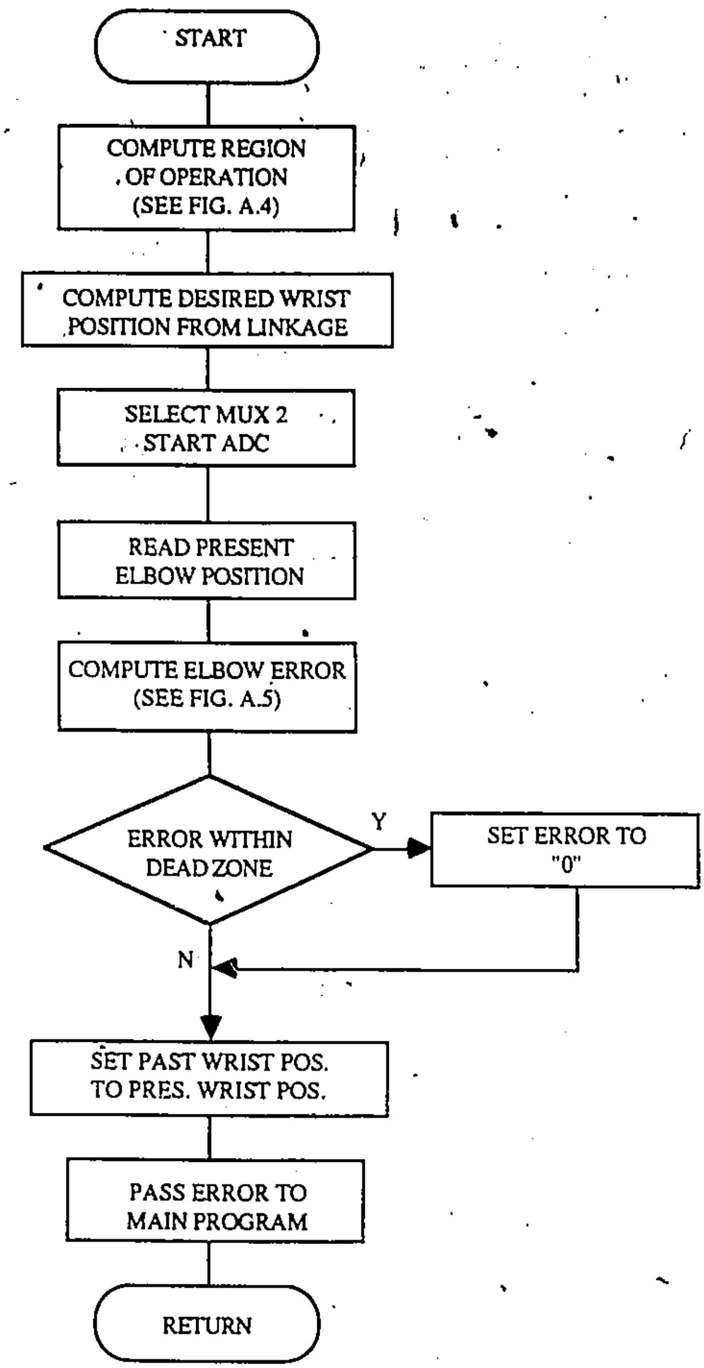


Fig. B.3 Flowchart for the computation of the wrist error.

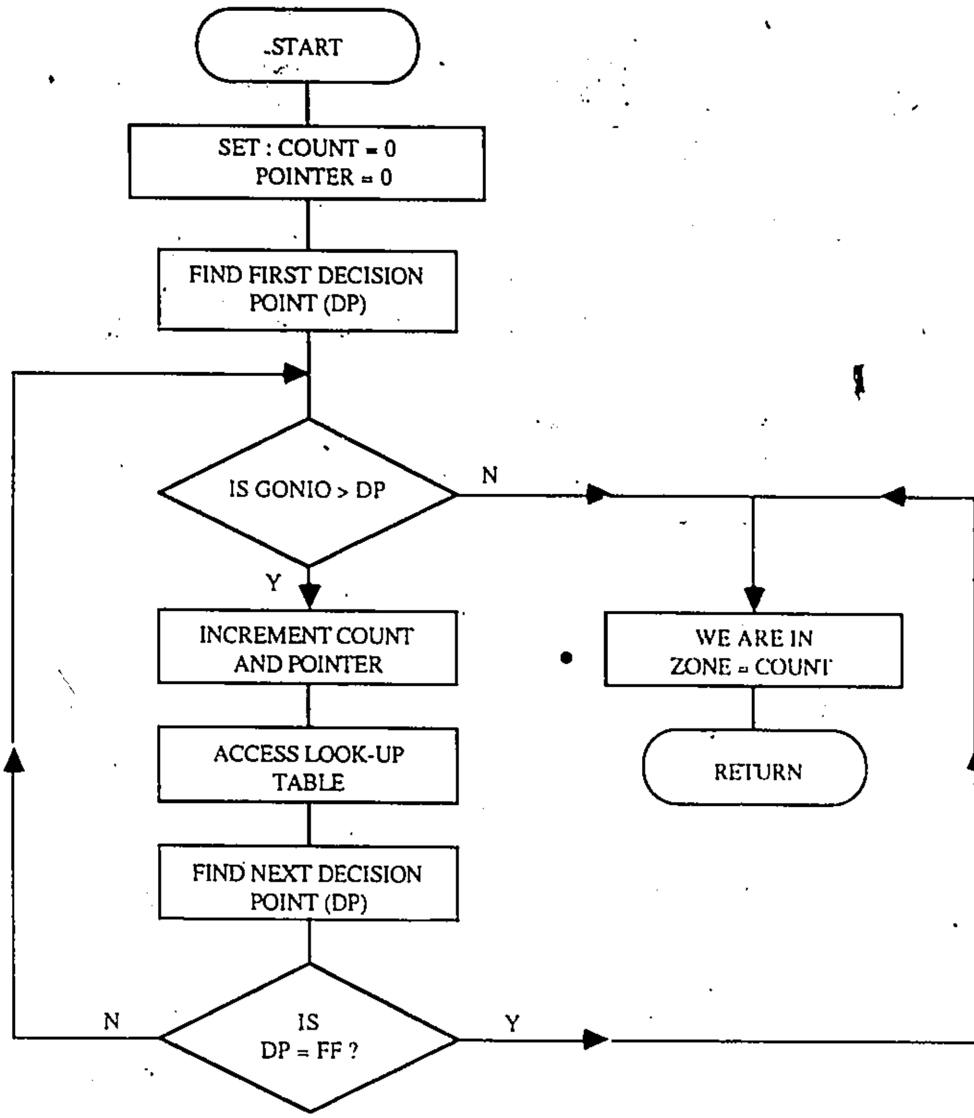


Fig. B.4 flowchart for the computation of the operating region within the look-up table.

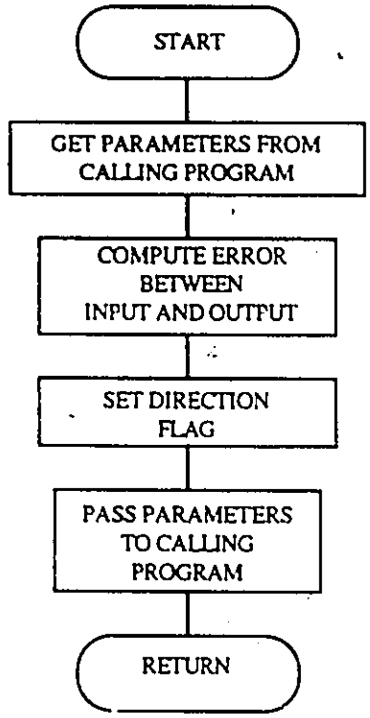


Fig. B.5 Flowchart for the computation of the error between input and output.

APPENDIX C

PROGRAM FOR THE CONTROL OF THE PROSTHESIS.

```
                    EXTRN  CODE (ELBOW, WRIST, LOOP)

                           RAM ADDRESSES

GONIO       DATA    56D
GOPAST      DATA    57D
ELPAST      DATA    58D
DIRECT      DATA    59D
WRPAST      DATA    63D
DRIVLO      EQU     0DFH
DRIVEH      EQU     0B1H
LINK        DATA    72D
ELDRIV      DATA    73D
WRDRIV      DATA    74D
ELDIR       DATA    75D
WRDIR       DATA    76D
;
;           ------------------------------------------
;           CSEG
            PUBLIC      TABLES,WTABLE
            ORG         00H
            AJMP        BEGIN
;
            ORG         01BH
            CLR         A
            RETI
;
;           ------------------------------------------
;
            ORG         30H
;
            DB          00H,41H,00H,00H,0FFH,00H,41H,00H,00H,7FH,00H
                        80H,00H,0FF,00H,40H,50H,00H,50H,41H,50,00H,
                        0FFH,30H,35H,3EH
;
WTABLE:     DB          30H,35H,3EH
;
            DB          00H,02H,00H,00H,0FFH,00H,41H,00H,00H,7FH,0C1H
                        80H,00H,FFH,00H,40H,00H,00H,0FFH
;
TABLES:     DB          4AH,4FH,58H
;
BEGIN:      MOV         SP,#50D
            MOV         GOPAST,#00
            MOV         ELPAST,#00
            MOV         WRPAST,#00
            MOV         JUICE,#00        ;INITIALISE ALL STORAGES
```

```
         CLR      P2.0            ;REMOVE BRAKE
;
DEBUT:   CLR      P2.7            ;SELECT MUX 00 I.E. GONIO
         CLR      P2.6

         CLR      P3.6            ;CLR WR LINE TO RESET ADC
         NOP
         MOV      R0,#100D
DELAY0:  NOP
         DJNZ     R0,DELAY0
         SETB      3.6            ;START CONVERSION (WR IS  NOW
                                  ;HIGH)
         MOV      R0,#80D
DELAY1:  MOV      A,#25D          ;SET  DELAY TO  ALLOW  FOR
                                  ;CONVERSION
         MOV      B,#0AH
         MUL      AB
         DJNZ     R0,DELAY1
         NOP
         SETB     P2.6            ;SET MUX TO 01 TO READ ELBOW
         CLR      P3.7            ;CLR PORT RD SO THAT DATA IS
                                  ;LATCHED OUT
         MOV      P1,#0FFH        ;SET PORT1 AS AN INPUT PORT
         MOV      GONIO,P1
         MOV      A,GONIO
         SETB     P3.7            ;LINE RD BECOMES HIGH
         CLR      P3.6            ;RESET ADC WITH WR TO LOW
         CLR      C
         SUBB     A,GOPAST        ;CHECK IF GONIO IS G.E. TO
                                  ;GOPAST
         JC       BACKWAR
         MOV      DIRECT,#01D     ;YES GONIO IS G.E TO GOPAST
         AJMP     NEXT
BACKWAR: MOV      DIRECT,#00      ;NO SET DIRECT. TO 0
NEXT:    MOV      GOPAST,GONIO    ;SET PAST POSI. TO PRESENT
                                  ;ONE
         MOV      P0,#0FFH        ;SET PORT0 AS AN INPUT PORT
         ;
         MOV      A,P0            ;READ IN VALUE OF LINKAGE
         ANL      A,#07           ;USE ONLY FIRST THREE BITS
         MOV      LINK,A          ;PUT LINKAGE VAL. IN LINK
         NOP
         ACALL    ELBOW           ;CALL ROUTINE THAT COMPUTES
                                  ;ELBOW ERROR
         ACALL    WRIST           ;DO THE SAME FOR WRIST
         CLR      C
         MOV      A,ELDRIV        ;CHECK IF WE ARE IN DEAD ZONE
         ADD      A,WRDRIV
         JNZ      GOON            ;NOP THEN KEEP GOING
         SETB     P2.5            ;YES STOP ELBOW MOTOR
         SETB     P2.3            ;YES STOP WRIST MOTOR
         ACALL    LOOP            ;WASTE 20 MS
         AJMP     DEBUT           ;GO BACK TO THE TOP
         NOP
GOON:    MOV      IP,#08D         ;SET PRIORITY LEVELS
```

```
        MOV     IE.#10001000     ;ENABLE INTERRUPT FOR TIMER 1
        MOV     TMOD.#10H        :SET TIMER 1 AS 16 BIT TIMER
        MOV     A.ELDIR          ;PUT ELBOW DIRECTION IN ACC
        MOV     C.ACC.0
        MOV     P2.4.C           ;SET DIRECT OF DRIVE OF ELBOW
        MOV     A.WRDRIV         :PUT WRIST DIRECTION IN ACC
        MOV     C.ACC.0
        MOV     P2.2.C           ;SET DIRECT OF DRIVE OF WRIST
        MOV     A.WRDRIV         :IS WRIST DRIVE ZERO?
        JNZ     NOTZIP
        SETB    P2.3             ;IT IS STOP WRIST MOTOR
        AJMP    COUDE            :GO TO ELBOW
NOTZIP: CLR     P2.3             ;START WRIST MOTOR
COUDE:  MOV     TL1.#DRIVLO
        MOV     TH1.DRIVEH       ;SET TIMER TO 20 MS
        MOV     A.ELDRIV         :IS ELBOW DRIVE ZERO?
        JNZ     OK               ;IF IT IS NOT GOTO OK
        SETB    P2.5             :IT IS TURN ELBOW OFF
        MOV     JUICE.#00        ;SET JUICE FLAG TO NODRIVE
        AJMP    SUITE1
OK:     CLR     2.5              ;NO TURN ELBOW MOTOR ON
        MOV     JUICE.#01
SUITE1: MOV     A.#01
        SETB    TR1              ;START TIMER
WAIST2: JZ      CIRCLE
        AJMP    WAIST2           ;WAIT FOR TIME TO ELAPSE
CIRCLE: AJMP    DEBUT            :GO BACK TO THE BEGINING
        END
```

ELBOW ROUTINE

```
;
        THIS SUBROUTINE CALCULATES THE ELBOW DRIVE

         PUBLIC ELBOW
         EXTRN  CODE(REGION,LINKAGE,DRIVE)
;
        RAM ADDRESSES
;
ELDIR     DATA     75D
ELDRIV    DATA     73D
ARM       DATA     58D
ELPAST    DATA     59D
OVERFL    DATA     79D
JUICE     DATA     62D
BRAKE     DATA     80D
RANGE     EQU      07D
DIRECT    DATA     60D
ELZONE    DATA     64D
ELBWL     DATA     65D
ELMINI    EQU      05D
;
;         ----------------------------------------
;
COUDE     SEGMENT  CODE
          RSEG     COUDE
;
ELBOW:    MOV      R1,#00        ;SET FLAG TO 0 - ELBOW CALLING
          ACALL    REGION        ;CALL ROUTINE REGION
          MOV      ELZONE,R1     ;USE R1 TO PASS PARAM BACK TO
                                 ;CALLER
          MOV      R0,#00        ;SET R0 TO ZERO -ELBOW CALLING
          ACALL    LINKAG        ;CALL ROUTINE THAT COMPUTES ELBOW
                                 ;DESIRED POSITION
          MOV      OVERFL,#00    ;INITIALIZE OVERFLOW FLAG
          SETB     P2.6          ;SET MUX TO 01 READ ELBOW
          CLR      P3.6          ;RESET ADC
          NOP
          SETB     P3.6          ;START CONVERSION- SET WR LINE
          MOV      R0,#0FFH
DELAY3:   MOV      A,#25D
          MOV      B,#0AH        ;SET DELAY LOOP FOR CONVERSION
          MUL      AB
          DJNZ     R0,DELAY3
          SETB     P2.7
          CLR      P2.6          ;PREPARE MUX FOR WRIST READING
          CLR      P3.7          ;LATCH DATA
          MOV      P1,#0FFH      ;SET PORT1 AS AN INPUT PORT
          MOV      ARM,P1        ;READ IN VALUE OF ARM
          MOV      R3,ARM
          MOV      A,R4
          MOV      R2,A
          MOV      R4,ELPAST     ;PASS PARAMETERS USING REGISTERS
```

```
        ACALL   DRIVE           ;CALL ROUTINE THAT COMPUTES ELBOW
                                ;DRIVE
        MOV     ELDIR,R3        ;PASS PARAMS USING REGISTERS
        CLR     A
        ADD     A,JUICE         ;ANY DRIVE APPLIED LAST TIME
                                ;AROUND ?
        JNZ     PARIS           ;NO DRIVE BEFORE
        CLR     P2.4            ;FREE BRAKE !
        MOV     BRAKE,#00       ;CLR BRAKE FLAG
        AJMP    DEAD            ;GOTO THE DEAD ZONE TEST
PARIS:  MOV     A,ARM           ;THERE WAS A DRIVE APPLIED LAST
                                ;LAST TIME AROUND THE LOOP
        SUBB    A,ELPAST        ;DO ACC - ARM-ELPAST
        JNC     COMPAR
        CPL     A
        INC     A               ;ADJUST RESULT TO ACC-CPL(A) + 1
COMPAR: CLR     C
        SUBB    A,#RANGE        ;HAVE WE MOVE MORE THAN RANGE ?
        JNC     DEAD            ;OK GOTO DEAD TEST
FREIN:  CLR     A
        MOV     BRAKE,#01       ;SET BRAKE FLAG
        SETB    P2.4            ;APPLY BRAKE
DEAD:   MOV     A,R4            ;PLACE DRIVE VALUE IN ACC
        CLR     C
        SUBB    A,#ELMINI       ;CHECK IF COMPUTED DRIVE IS
                                ;GREATER THAN MINIMUM
        JC      NODRIV
        MOV     ELDRIV,R4       ;WE ARE NOT IN DEAD ZONE
                                ;SET ELBOW DRIVE - DRIVE
        AJMP    FIN
NODRIV: MOV     ELDRIV,#00      ;WE ARE IN DEAD ZONE SET DRIVE
                                ;TO ZERO
FIN:    NOP
        RET
        END
```

```
;                        WRIST ROUTINE
;
;       THIS SUBROUTINE CALCULATES THE WRIST DRIVE
;
          PUBLIC WRIST
          EXTRN  CODE (REGION,LINKAG,DRIVE)
;
         RAM ADDRESSES
;
WRISTP    DATA      83D
WRPAST    DATA      63D
WRDIR     DATA      76D
WRDRIV    DATA      74D
WRZONE    DATA      67D
WRIMIN    EQU       07D
WRIMAX    EQU       63D
;
;         ---------------------------------------
;
POIGNE    SEGMENT CODE
          RSEG    POIGNE
;
WRIST:    MOV       R1,#01          ;SET FLAG, WRIST IS CALLING
          ACALL     REGION          ;FIND OUT WHICH REGION WE ARE IN
          MOV       WRZONE,R1
          MOV       RO,#01          ;PASS PARAMS TO LINKAGE ROUTINE
          ACALL     LINKAG          ;CALL ROUTINE THAT COMPUTES
                                    ;THE DESIRED POSITION
          MOV       A,R4
          MOV       R2,A            ;PASS PARAMETERS
          CLR       P3.6            ;RESET ADC
          NOP
          SETB      P3.6            ;START CONVERSION
          MOV       RO,#80D
 DELAY4:  MOV       A,#25D
          MOV       B,#0AH
          MUL       AB
          DJNZ      RO,DELAY4
          CLR       P2.7
          CLR       P2.6            ;SET THE MUX FOR GONIO READING
          CLR       P3.7            ;LATCH DATA OUT
          MOV       P1,#OFFH        ;P1 AS AN INPUT PORT
          MOV       WRISTP,P1       ;STORE-IN WRIST POSITION
          CLR       P3.6            ;RESET ADC
          MOV       A,#EOH          ;DO A SOFTWARE SWITCH
          CLR       C
          SUBB      A,R2            ;CHECK IF DESIRED POSI IS >4.35V
          JC        LIMIT
          MOV       R3,WRISTP
          MOV       R4,WRPAST       ;PASS PARAM TO ROUTINE
          ACALL     DRIVE           ;CALL ROUTINE THAT COMPUTES THE
                                    ;DRIVE
          MOV       WRDIR,R3        ;GET DIRECTION OF DRIVE
```

```
        CLR     C
        MOV     A,R4

        SUBB    A,#WRIMIN       ;CHECK IF WE ARE IN DEAD ZONE
        JC      LIMIT
        MOV     WRDRIV,R4       ;GET PARAMETERS BACK FROM DRIVE
        MOV     WRPAST.WRISTP   :SET PAST POSI TO PRESENT POSI
        AJMP    FINI
LIMIT:  MOV     WRDRIV,#00      ;WE ARE IN THE FORBIDEN ZONE
                                :SET DRIVE TO ZERO
FINI:   NOP
        RET
        END
```

```
;
                     SUBROUTINE REGION
;
;          THIS SUBROUTINE FINDS THE REGION OF OPERATION WITHIN THE
           LOOK-UP TABLE.
;          ------------------------------------------------------------
;
           PUBLIC REGION
           EXTRN  CODE (TABLES,WTABLE)
;
          RAM ADDRESSES
;
COUNT      DATA      91D
LINK       DATA      72D
ZONE       DATA      92D
GONIO      DATA      56D
;
;          ------------------------------------------------------------
;
AREA       SEGMENT
           RSEG      AREA
REGION:    MOV       COUNT,#00        ;CLEAR COUNT
           MOV       A,R1             ;CHECK WHICH ROUTINE IS CALLING
           JNZ       WRISTO           ;IF R1=1 WRIST IS CALLING
           MOV       A,LINK           ;GET LINKAGE SELECTION
           MOV       DPTR,#TABLES     ;POINT TO LINKAGE TABLE
           AJMP      GOHEAD
WRISTO:    MOV       A,LINK
           MOV       DPTR;WTABLE
GOHEAD:    MOVC      A,@A+DPTR
           ADD       A,#4D            ;A= No OF ENTRY IN TABLE
           MOV       RO,A             ;RO POINTS TO NEXT REGION
           MOV       DPTR,#00         ;SET DATA POINTER TO ZERO
           MOVC      A,@A+DPTR        ;A=NEXT LOWER BOUND
           CLR       C
           SUBB      A,GONIO          ;DO LOWER BOUND (ZONE#?)-GONIO
           JC        SEARCH           ;KEEP ON SEARCHING
           MOV       R1,#00
           RET
AGAIN:     MOV       A,#4
           ADD       A,RO             ;NOW A IS POINTING TO THE NEXT
                                      ;REGION
           MOV       RO,A             ;UPDATE POINTER
;
           MOVC      A,@A+DPTR        ;A=NEW LOWER BOUND
           SUBB      A,#OFFH          ;ARE WE AT THE END OF THE TABLE ?
           JZ        OVER             ;IF TRUE GOTO OVER
;
           MOV       A,RO             ;SET A TO NEW LOWER BOUND AGAIN
           MOVC      A,@A+DPTR
           CLR       C
           SUBB      A,GONIO
           JC        SEARCH
OVER:      MOV       R1,COUNT         ;WE ARE IN REGION COUNT
```

```
        RET
SEARCH: INC     COUNT
        AJMP    AGAIN
        END
```

```
;                         SUBROUTINE LINKAG

          THIS ROUTINE IS CALLED BY THE ELBOW AND THE WRIST ROUTINES,
          IT  CALCULATES  THE  DESIRED  POSITION  USING  THE  PRESENT
          POSITION VALUE, THE OPERATING REGION AND LOOK-UP TABLES.
;
;         ------------------------------------------------------------
;
          PUBLIC LINKAG
          EXTRN  CODE (TABLES,WTABLE)
;
LINK      DATA      72D
ZONE      DATA      92D
GONIO     DATA      56D
LEVINF    EQU       04D
;
;         ------------------------------------------------------------
;
LIEN      SEGMENT CODE
          RSEG    LIEN
;
LINKAG:   MOV       A,RO            ;CHECK WHICH ROUTINE IS CALLING
          MOV       ZONE,R1         :UP-DATE ZONE OF OPERATION
          JNZ       SECTAB          ;IF A=O ELBOW IS CALLING
          MOV       A,LINK          ;GET LINKAGE SELECTION
          MOV       DPTR,#TABLES    ;POINT A BEGINING OF TABLE
          AJMP      ICI
SECTAB:   MOV       DPTR,#WTABLE
          MOV       A,LINK
ICI:      MOVC      A,@A+DPTR
          MOV       RO,A            ;RO=ADDRESS OF TABLE I
          MOV       A,ZONE          :SET PROCEDURE TO GET OFFSET IN
                                    ;TABLE
          MOV       B,#LEVINF       ;MOVE INFO LEVEL INTO B
          MUL       AB
          NOP                       ;WE NOW HAVE OFFSET IN TABLEI
          MOV       R1,A
          MOV       A,RO            ;A = ADDRESS OF TABLE I
          ADD       A,R1            :A = STARTING ADDRESS+OFFSET
          MOV       RO,A            ;A CONTAINS STARTING ADDRESS OF
                                    :ZONE OF OPERATION
          MOV       DPTR,#00        ;SET DPTR TO 00 = BASE
          MOVC      A,@A+DPTR       :A = LOWER BOUND OF REGION
          MOV       R6,GONIO        ;R6 = GONIO = INPUT
          XCH       A,R6
          SUBB      A,R6            ;A=GONIO-LOWER BOUND =EPSI
          MOV       R6,A            ;SAVE EPSILON IN R6
          CLR       C
          INC       RO              ;RO POINTS TO SLOPE
          MOV       A,RO
          MOVC      A,@A+DPTR
          MOV       R3,A
          MOV       C,ACC.6         ;DIVIDE OR MULTIPLY ?
          JC        MULTI           ;IF SIXTH BIT OF SLOPE SET
```

```
                                        ;GOTO MULTIPLY
          ANL       A.#1FH
          MOV       B.A                 ;B IS NOW EQUAL TO SLOPE
          MOV       A.R6                ;A = EPSILON
          DIV       AB                  ;SLOPE.6 NOT SET DIVIDE
          MOV       R4,A                ;R4 =  (EPSI/SLOPE)  INTEGER
                                        ;PART
          MOV       A,B
          RLC       A                   ;A = 2A
          MOV       R7,A                ;SAVE 2A IN R7
          MOV       A,R3                ;GET SLOPE IN A
          ANL       A,#1FH
          NOP
          SUBB      A,R7                ;DO A= 2A-SLOPE
          JNC       HAITI
          INC       R4
          AJMP      HAITI
;
MULTI:    MOV       R7,A
          ANL       A,#1FH              ;AGAIN MASKOUT SLOPE CONTROL
          MOV       B,A
          MOV       A,R6
          MUL       A,B                 ;DO EPSILON x SLOPE
          MOV       R4,A                ;R4 = (EPSI x SLOPE)
HAITI:    MOV       A,#2D
          ADD       A,R0                ;A = POINTER TO CORRECT. FACTOR
          MOV       R1,A
          MOVC      A,@A+DPTR           ;A = CORRECTION FACTOR
          MOV       R5,A                ; SAVE C. F. IN R5
          JZ        SIMPLE              ;ON ZERO DON'T ADJUST RESULT
          MOV       B,A                 ;B = CORRECTION FACTOR
          MOV       A,R6                ;A = EPSILON
          DIV       AB                  ;A = EPSILON/C.F.
          MOV       R2,A
          MOV       A,B
          RLC       A                   ;A = 2A
          CLR       C
          XCH       A,R5                ;DO A=CORRECTION FAC., R5= 2A
          SUBB      A,R5                ;DO CORRECTION FAC. - 2A
          JNC       SKEEP
          INC       R2                  ;ROUND UP THE RESULT IN R2
SKEEP:    MOV       A,R7                ;A = SLOPE
          MOV       C,ACC.5             ;CHECK WHETHER BIT FIVE OF SLOPE
                                        ;IS SET'
          JC        SUBSTRA             ;IF C SET SUBSTRACT
          ;IF C RESET SLOPE WAS UNDER ESTIMATED THUS ADD
          ;
          MOV       A,R4                ;A=EPSI(/)OR(x) SLOPE
          ADD       A,R2                ;A=EPSI/x SLOPE - CORRECT. QUANTITY
          AJMP      THERE
SUBTRA:   MOV       A,R4
          CLR       C
          SUBB      A,R2                ;A=EPSI/xSLOPE-CORRECT. QUANTITY
THERE:    MOV       R4,A                ;SAVE A IN R4
```

```
SIMPLE:   MOV     A,R3           ;A =SLOPE

          INC     RO             ;RO = POINTER TO BASE
          MOV     C,ACC.7        ;CHECK BIT 7 OF SLOPE FOR.SIGN
                                 ;OF SLOPE
          JC      NEGA           ;IF ACC.7 IS SET SLOPE IS NEGATIVE
          MOV     A,RO
          MOVC    A,@A+DPTR      ;GET BASE
          ADD     A,R4           ;A=(EPSIxSLOPE +- RESULT) +BASE
          AJMP    DONE
NEGA:     CLR     C
          MOV     A,RO           ;A = POINTER TO BASE
          MOVC    A,@A+DPTR      ;A = BASE
          SUBB    A,R4           ;A=BASE-(EPSI*SLOPE+- RESULT)
DONE:     MOV     R4,A           ;DESIRED POSITION IS PASSED BACK
                                 ;TO CALLER THROUGH R4
          RET
          END
```

```
;                    SUBROUTINE DRIVE
;
        THIS SUBROUTINE COMPUTES THE ERROR BETWEEN INPUT AND
        OUTPUT. IT MAKES USE OF THE PRESENT POSITION AND THE
        DESIRED ONE. IT IS CALLED BY THE ELBOW AND WRIST
        ROUTINES.
;       ---------------------------------------------------------
;
        PUBLIC DRIVE
;
        RAN ADDRESSES
;
DESIR   DATA    84D
PRESEN  DATA    85D
ROTATE  DATA    86D
OVERFL  DATA    87D
;
;       ---------------------------------------------------------
;
PULSE   SEGMENT CODE
        RSEG    PULSE
;
DRIVE:  CLR     C
        MOV     DESIR,R2
        MOV     PRESEN,R3
        MOV     A,DESIR         ;PUT DESIRED POSI IN ACC
        SUBB    A,PRESEN        ;DO : DESIRED-PRESENT
        JC      BAKWAR
        MOV     ROTATE,#1       ;DIRECTION IS CCW
        AJMP    VAZI
BAKWAR: CLR     C
        MOV     A,PRESEN        ;DO : PRESENT-DESIRED
        SUBB    A,DESIR
        MOV     ROTATE,#0       ;DIRECTION IS CW
VAZI:   MOV     OVERFL,#00
        MOV     R4,A            ;PASS DRIVE VALUE USING R4
        MOV     R3,ROTATE       ;PASS DIRECTION BACK TO CALLING
                                ;PROGRAM
        RET
        END
```

SUBROUTINE LOOP

```
;
;                   SUBROUTINE LOOP

        THIS ROUTINE LOOP IS USED AS A TIME BASE WHEN NO
        DRIVE IS APPLIED TO THE MOTOR. THIS ROUTINE LOOPS
        FOR 20 MILLISECONDS AND RETURNS TO THE MAIN PGR.
        -------------------------------------------------
;
;
        PUBLIC LOOP
;
JUICE   DATA     3EH
        -------------------------------------------------
;
;
        LOOP1    SEGMENT   CODE
        RSEG     LOOP1
;
LOOP:   SETB     P2.5            ;STOP ELBOW MOTOR
        SETB     P2.3            ;STOP WRIST MOTOR
;
        MOV      IP,#08D
        MOV      IE,#88H         ;ENABLE INTERRUPT FOR TIMER #1
        MOV      TMOD,#10H       ;SET TIMER #1 AS A 16 BIT TIMER
        MOV      RO,#5           ;SET LOOP COUNTER TO 1
START:  MOV      A,RO
        JZ       DONE            ;WHEN COUNT IS REACHED STOP
        MOV      TL1,#0DFH       ;DF-CPL[20]
        MOV      TH1,#0B1H       ;B1-CPL[4E]
        SETB     TR1             ;START TIMER
        DEC      RO              ;DECREMENT COUNTER
        MOV      A,#01
MINUTE: JZ       START
        AJMP     MINUTE
        NOP
DONE:   MOV      JUICE,#00       ;CLR JUICE FLAG I.E THERE WAS NO
                                 ;DRIVE APPLIED THIS TIME
        RET
        END
```

APPENDIX D

The Boston elbow makes use of a D.C. motor for the elbow actuation. It is often believed that for robotics applications, such as the one under consideration, a step motor would be more appropriate. The main reason behind this is the fact that the control required to operate a step motor is in general much simpler than that required to operate a D.C motor [22], because feedback is not required. Position information is inherent in a system using a step motor. However we do not think that this kind of motor would be as good as a D.C. motor in our system for the following reasons :

1) The step motor suffers from many known disadvantages [23]:
   - Fixed step angle or increment of motion; lacks flexibility in step resolution
   - Low efficiency with ordinary driver; low power to weight ratio (lower than a comparable D.C. motor)
   - relatively high overshoot and oscillation in step response
   - Limited ability to handle large and non-constant inertia loads

2) The lack of feedback is both the greatest attribute and the greatest drawback of the step motor. When a motor is moving rapidly, in a slewing mode and the control pulses suddenly stop, the motor may easily have enough foward momentum to carry it all the way past the desired

stopping point around to the next stable point. This could be the cause of a positioning error highly related to the relative loading of the arm. Certainly this is not acceptable in prosthetic applications.

3) In order to provide the elbow with passive holding force, gears would have to be used which , inevitably, would lead to the same dead zone problem (i.e. a certain margin must be left for the play of the gears) encountered with our D.C. motor coupled to gears.

All the above mentioned facts are arguments that convinced us that the utilization of a step motor for elbow actuation would not lead to any significant improvement in the performance of the prosthesis.