A DISTRIBUTED FUZZY LOGIC CONTROLLER FOR A PROSTHETIC HAND

MOHD YAZED AHMAD

FACULTY OF ENGINEERING UNIVERSITY OF MALAYA
KUALA LUMPUR

JAN 2006

A DISTRIBUTED FUZZY LOGIC CONTROLLER FOR A PROSTHETIC HAND

MOHD YAZED AHMAD

A Dissertation Submitted to Engineering Faculty University of Malaya as Partial Requirement for the Degree of Master of Engineering Science

FACULTY OF ENGINEERING UNIVERSITY OF MALAYA
KUALA LUMPUR

JAN 2006

UNIVERSITI MALAYA
ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Mohd Yazed Bin Ahmad (IC/Passport No: 790817-07-5445)

Registration/Matric No: KGA030037

Name of Degree: Master of Engineering Science

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

A Distributed Fuzzy Logic Controller for a Prosthetic Hand

Field of Study: Biomedical Engineering

I do solemnly and sincerely declare that:

1) I am the sole author/writer of this Work;
2) This Work is original;
3) Any use of any. work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained,
6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature: Date: 12/07/2006

Subscribed and solemnly declared before, Date: 12/07/2006

Witness's Signature ~~Date~~

Name: AHMAD HAFIZ BIN KADIR

Designation: Mr.

ABSTRACT

A Fuzzy Logic with distributed control monitoring (DCS) system is implemented to control multiple degree-of-freedom (DOF) prosthetic fingers. There are four fingers with 3-DOF and a thumb with 4-DOF. Five identical microcontrollers programmed with Fuzzy Logic Controller (FLC) and a System Handler are employed to control and monitor the fingers and the thumb to replicate the desired hand actions of the grasp, the key pinch, the pulp to pulp pinch, the tripod pinch, and the open hand. Each finger is equipped with position sensors at the pivot joints and a tactile-pressure sensor at the fingertip. The finger movements are programmed to follow given set points and stopped whenever an obstacle is encountered and the pressure of the tactile sensor exceeds a specified limit. This allows the fingers and thumb to wrap round an object without crushing it. DC motors with reduced gear heads are used as actuators and they are driven by H-Bridge switches. Input signals to the switches in the form of Pulse Width Modulation (PWM) and direction signals are generated by the microcontrollers. The signals represent control action of the FLC. Membership functions of the FLC were tuned and the rules were formed to obtain the desired response. Distributed control is implemented by connecting all finger microcontrollers to a main microcontroller that can be integrated with the Brain Computer Interface. The overall system was constructed and tested successfully to control the prosthetic hand.

ACKNOWLEDGEMENT

I would like to convey my thanks and gratitude to those who have supported and encouraged me throughout the preparation of this thesis. Thanks to my supervisor Professor Dato' Goh Sing Yau for his enthusiastic support, guidance, and advice. To the BCI group and all staff members at Faculty of Engineering, University of Malaya thank you very much. I also would like to thank Professor Hew Wooi Ping for his guidance and technical advice especially at the beginning of my research.

Last but not least, I would like to express my heart-felt gratitude to my beloved family for their encouragement and advice, and also to all my friends whose names are not mentioned here. Thanks to all of you.

Table of contents

TABLE OF FIGURES

TABLE OF TABLES

NOMENCLATURES AND ABBREVIATIONS

BCI: Brain Computer Interface

ECG: Electrocardiogram

EEG: Electroencephalogram

EMG: Electromiogram

FLC: Fuzzy Logic Controller

P: Proportional

PI: Proportional-Integral

PID: Proportional-Integral-Differential

MF: Membership Function

INTRODUCTION

1.0 Introduction

This thesis work is part of a larger project to develop a Brain Computer Interface (BCI) system at University of Malaya to enable control and communication using brain signals. An electrically actuated prosthetic hand is one of the devices in this system. A hand controller is designed to operate the prosthetic hand using the BCI system. Such a system will give disabled or paralyzed people a chance to be independent and be able to do certain tasks by their own will. In this type of application, the hand controller must be intelligent enough to enable the separate fingers to move in a manner that will adapt the hand geometry for the intended hand task.

1.1 BCI system overview

1.1.1 BCI in general

The BCI is an interfacing of the brain signals to a computer in a way that the computer can recognize the intensions of the subject [1]. An Electroencephalogram based BCI provides a new communication channel between the human brain and a computer [1,2]. Specific information from neuron activities is obtained through analyzing and classifying this signal [1,2]. A typical BCI consist of an Electroencephalogram amplifier (EEG Amplifier), a Data Acquisition System (DAQ), a computer with Digital Signal Processing software and device. The device can either be a program inside the computer like cursor movement program or it can be external appliances like wheelchair or prosthetic devices. Subjects who suffer from severe motor impairments such as late stage of Amyotrophic Lateral Sclerosis (ALS), severe cerebral palsy, head trauma and spinal injuries may use such a BCI system as an alternative form of communication [1,2].

1.1.2 A BCI system developed at University of Malaya

We aim to develop a portable BCI system to control devices like a prosthetic hand and to switch on remote electrical appliances [55]. The BCI system will be designed such that the power consumption is low and the number of electrodes used is minimal. The *brain signal classifier*, which is the core unit of this BCI system, will run in a remote computer. The structure of the BCI system is shown below.

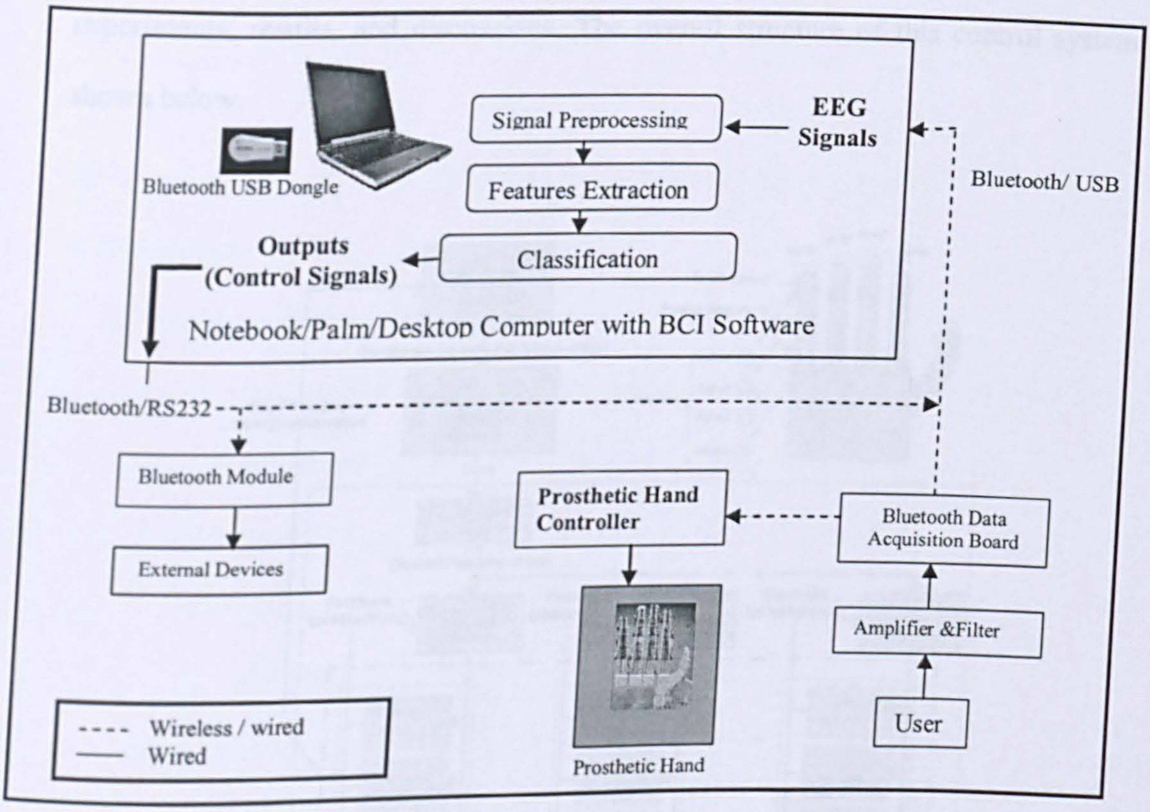


Figure 1.1: The BCI system developed at University of Malaya

This BCI system consists of four main parts: The electroencephalogram (EEG) amplifier, the data acquisition (DAQ) system, the Brain signal classifier and the devices (i.e. a prosthetic hand and LED's that represent switching states of devices). EEG electrodes that are placed on the scalp of the subject pick up the brain signals.

Conductive paste is applied between the electrodes and the scalp to improve electrical conductivity. The EEG signal is filtered, amplified and transmitted to the remote computer. The signal processing, feature extraction and classification of the EEG data are carried out in the remote computer. The results of the classification are output as control signals to the prosthetic hand controller to operate it or to remote LED's representing other devices. A Graphic User Interface (GUI) on a display allows the subject to select and activate the desired tasks. This thesis describes the development and implementation of the prosthetic hand motion controller using Fuzzy Logic with distributed control monitoring system. It includes control analysis, design methodology, experiments, results, and discussions. The overall structure of this control system is shown below.

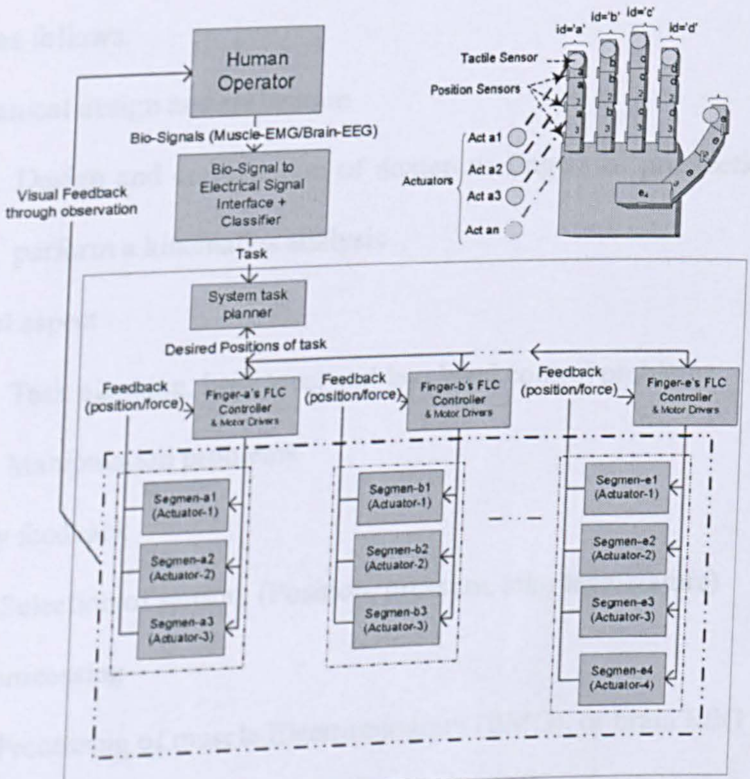


Figure 1.2: Structure of the prosthetic hand control system

1.1.3 An Overview of prosthetic hand developments

The prosthetic hand has been introduced a long time ago. In 218 BC a Roman General Marcus Sergius had his left hand amputated, he was then fitted with an iron replacement to hold his shield and was able to return to battle [3]. Over the centuries, the pace of development has been limited due to the lack of technological progress and small number of amputees. It was not until the American Civil War, that the first artificial hands were actuated by harnessing the motion of the body. In the early 20th century, it was possible to exploit external sources by electrical actuation [3]. The prosthetic hand design is slowly modified until the shape resembles that of a natural human hand.

Development of a prosthetic hand involves a multidisciplinary study [12]. The study can be classified as follows:

1. Mechanical design and realization
   - Design and construction of dexterous fingers of prosthetic hand and to perform a kinematics analysis
2. Control aspect
   - Task planning, high level and low level control problems
   - Manipulation problems
3. Sensory feedback
   - Selection of sensors (Position, pressure, slip, temperature)
4. Signal processing
   - Processing of muscle Electromiogram (EMG), or brain EEG signals.
5. Cosmetic
   - Enhancement of appearance and looks.

A prosthetic hand is necessary for amputees not only to improve their looks but also as a tool to facilitate their daily life activities. If this kind of hand can mimic and functions effectively like the human hand it will also become very useful in other areas of applications like remote surgery and hazardous handling [5]. These are among the things that motivate the development and improvement of prosthetic hands. Nowadays with continued advancements in technologies, and the development of tissue engineering, a well functioning prosthetic hand that also looks like a real hand has become possible. Also the improvement in living standards has made the prosthetic hands more affordable.

1.1.4 The prosthetic hand developed at University of Malaya

An anthropomorphic prosthetic hand mechanism that is to be used as a device in the BCI system was designed [53, 54, 55]. The prosthetic hand consist of four fingers with 3-*degree-of-freedom* (DOF) and a thumb with 4-DOF. The fingers and the thumb are mounted on a plate that follows the human hand arrangement. Actuations of segments of the fingers are performed using geared DC motors. A tendon cable is used to link each motor to the associate finger segment. The overall structure of the prosthetic hand is shown in the figure below.

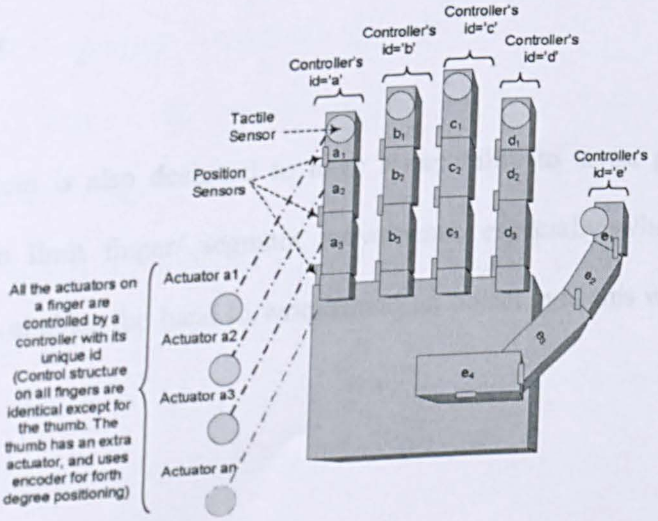


Figure 1.3: The prosthetic hand

The segments of the fingers are controlled in such a way that they move to form the desired geometry to mimic the motion of the human hand. Control system that is applied to control this hand is discussed in detail in the following chapters of this thesis report.

1.2 Distributed FLC Implementation (Methodology)

Fuzzy Logic with Hierarchical control method is selected to control the multi DOF prosthetic hand. The Fuzzy Logic method is found to be the most suitable to resolve the uncertainty behaviors and the complex control modeling of multi-fingered hand [16, 17, 25]. This method uses linguistic terms that can be easily understood by human whose operates the hand.

With the Hierarchical control, every microcontroller of a finger is treated as a slave unit. The microcontroller runs FLC algorithm, and performs closed loop control to position segments of a finger. Five units of slave microcontroller are employed to control all the fingers and the thumb. At one level higher, there is a master microcontroller which functions to coordinate all the slaves. The mater also allows another higher-level integration. This feature enables the incorporation of a BCI to the prosthetic hand motion controller.

This control system is also designed to have a capability to sense pressure. A tactile sensor is used to limit finger/ segment movements, especially when an obstacle is encountered. This enables the hand to wrap round an object, and this widens its function and capability.

For hardware implementation, embedded microcontrollers from Microchip Technology Inc. are selected. PIC18F4431 is used as slave microcontroller and PIC18F720 as master microcontroller.

For inter-microcontrollers communications, USART with RS232 protocol is employed. Special interrupt service routines are written for both master and slaves microcontrollers to serves commands from master to slave, and from BCI to master. The whole system is tested to perform five typical hand tasks; the grasp, the key pinch, the pulp to pulp pinch, the tripod pinch, and the open hand.

1.3 Thesis Objectives

The main objectives of this thesis are:

1. To develop a *prosthetic hand motion controller* for a multi fingered prosthetic hand.
2. To enables control and communication between the prosthetic hand controller to the *developed BCI system*.

1.4 Thesis Outline

This thesis has been organized in the following way:

- In Chapter 1, the overview of BCI and the part concerned for the BCI system is highlighted.
- Chapter 2 reviews previous works on control strategies to dexterous/robotic manipulators.
- In chapter 3, the prosthetic hand mechanism, fingertip trajectory, and actuation system are discussed.

- In chapter 4, Fuzzy Logic Control and Distributed monitoring / Hierarchical System to the multi-fingered prosthetic hand are described.
- Chapter 5 discusses hardware & software development.
- Chapter 6 shows the performance of the implemented controller. It discusses time responses, trajectory of a dexterous finger and the result of overall integration "Hand to BCI".
- Chapter 7 concludes the finding and achievement of this thesis. It also includes suggestions and possible improvement for future works.

LITERATURE REVIEW

2.0 Introduction

The implementation of a Prosthetic hand controller involves multidisciplinary studies. Reviews are grouped into sections to clearly see what are the best components, approaches and methods that most suit for the control of a prosthetic hand. The first section reviews the recently developed prosthetic hand to see the trend and how far the electrical control has been developed. The next sections extend the review into more specific topics that discuss the structure of multiple plants control and also the identification of components for electrically actuated hands. The most important section, the methods of control is discussed in section 2.2. Conclusions are then made at the end of this section.

2.1 Prosthetic Hand and Robotic Hand Developments

Earlier developments of prosthetic hands focus on creating a hand mechanism without fully considering the control strategy. The earliest design solely depends on mechanical design itself with actuation relying on human muscles [3, 12]. This approach limits dexterity and the capability of the hand. The main reason was that the technology of electronics, microcontrollers and the actuators were still at the early stage during that period. For a compact system like a prosthetic hand controller, the use of such technology was not practical. These controllers and actuators consume high power making the overall system inefficient. However, the advancement of semiconductors, microcontrollers and actuators technologies in recent years has made compact electrical actuation control become possible. Most of the parts are now designed to be smaller and lighter.

The development of prosthetic and robotic hands shares lots of similarities in terms of designs and technology applied. In fact the robotic researches have contributed a lot to the advancement and improvement to prosthetic hand systems [18]. This literature review considers both the prosthetic & robotic hand developments. The followings are among the famous hands developed recently.

i. **Belgrade/USC hand,**. It attempts to mimic human hand dexterity and functionality. It has four degrees of freedom on each finger. McHenry implements the conventional PD control with the feedbacks of joint position sensing and contact force sensing [22].

ii. **The Stanford/JPL hand.** Its design had been motivated by anthropomorphic considerations as well as kinematics and control issues. *Clifford S. Loucks and Victor J.Johnson* use strain gauges to sense tendon tension and six axis force/torque sensors placed on finger tips for contact force sensing. The conventional PD control method has been applied on this hand [11].

iii. **The Hirzinger hand** incorporates joint position sensing, joint torque sensing, tactile sensing, temperature sensing, and a vision system in the hand. By far, the Hirzinger hand has the most number of sensors (28 per finger). The sensing used by Hirzinger hand offers simplicity of the arrangement, compact sizes, better resolution, and fewer wires [19].

iv. **The Utah/MIT hand** was originally developed at the University of Utah by the Center for Engineering Design, led by Dr. Steve Jacobsen. In 1987, Motion Control incorporates joint position sensing, joint torque sensing and a tactile sensing suite with conventional proportional control to the hand. Microprocessor technology with computer interface was also been introduced that allows the prosthetist or wearer to fine-tune the adjustment to achieve maximum performance [18].

v. **DIST-Hand** is a modular robotic system made up of 4-degrees of freedom (dof) identical fingers. Each finger is actuated by a system of six tendons routed through pulleys and driven by five DC motors. Giuseppe Casalino & Fabio Giorgia from University Genova Italy introduced Embedded FPGA-based control with distributed control architecture. Parallel processing in that system provides solution to complex algorithm calculation. The FPGA-based performs 100 times faster than the leading DSPs [21]. Controller of DIST-Hand employs a multilayered hierarchical structure as depicted in Figure 2.1.

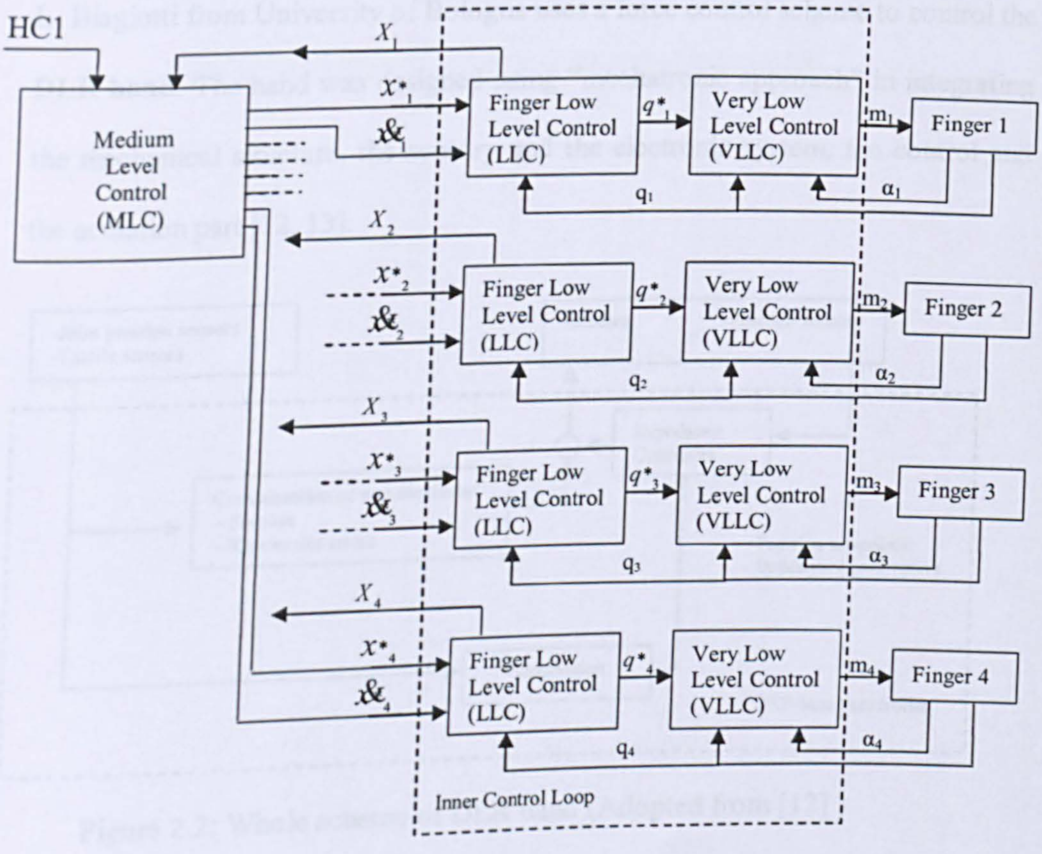


Figure 2.1: The architecture of DIST Hand control system (Adopted from [21]).

vi. **Shadow Hand** – Developed by Shadow Robot Company, the hand is a modular general-purpose human form dexterous manipulator. It uses pneumatic actuator with on/off control [20].

vii. ***GIFU* hand**, is a Multi-fingered Robot Hand that possesses dexterous manipulation capability. Yingjie Yin implements Hybrid Control to the hand. In order to obtain sub-optimal solutions, biologically inspired techniques might be very powerful [23].

viii. **NTU hand**, Han-Pang Huang controls the dexterous hand (NTU-Hand) using force feedback. A DSP-based bilateral control is used to link the hand to the operator [5].

ix. L. Biagiotti from University of Bologna uses a force control scheme to control the **DLR hand**. The hand was designed using "mechatronic approach" in integrating the mechanical structure, the sensory and the electronic system, the control and the actuation part [12, 13].

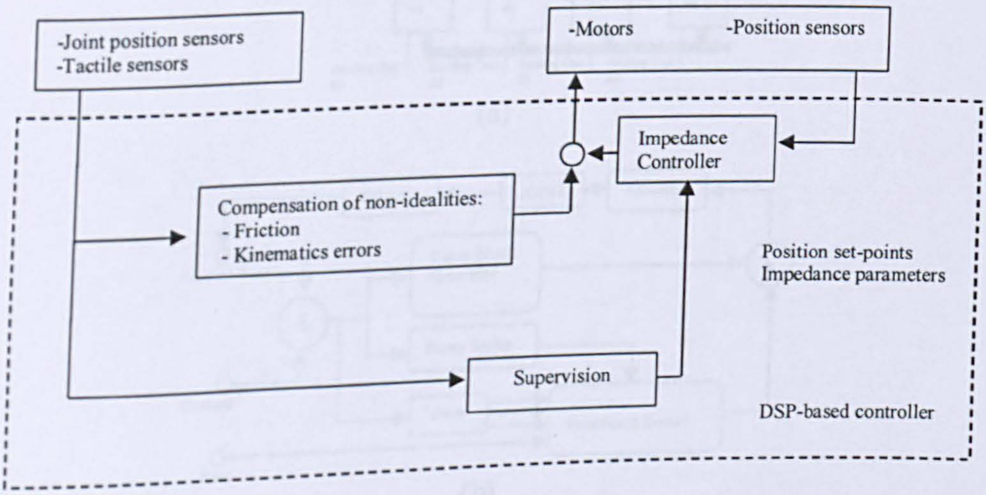


Figure 2.2: Whole scheme of DLR hand (Adopted from [12])

Each finger is integrated with distributed sensory equipment in order to allow application of control procedure taking into account local and structural compliance [12].

2.1.1 Control structure of a multi-fingered hand

This section reviews the aspect of *microcontroller-actuator* interfacing. Key factors in this review are the hierarchical structure, data line/communication, and the actuation approach.

I. *Steven M. Spano & Nicholas Bourbakis* use a method of Fuzzy Blocks control scheme to control a Multi Finger Robotic Hand. The interface system as shown in Figure 2.3 (a) below is capable of activating four DC motors in forward and reverse bias mode [8].

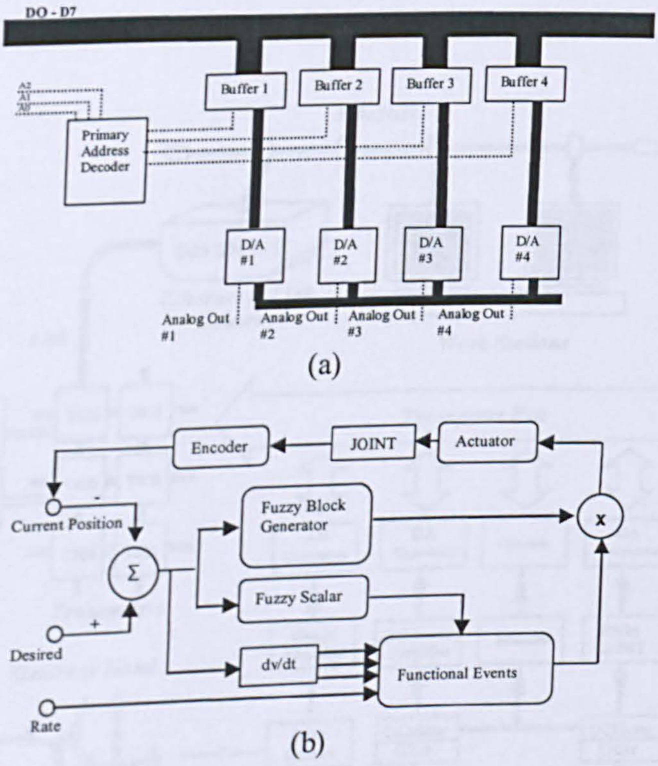
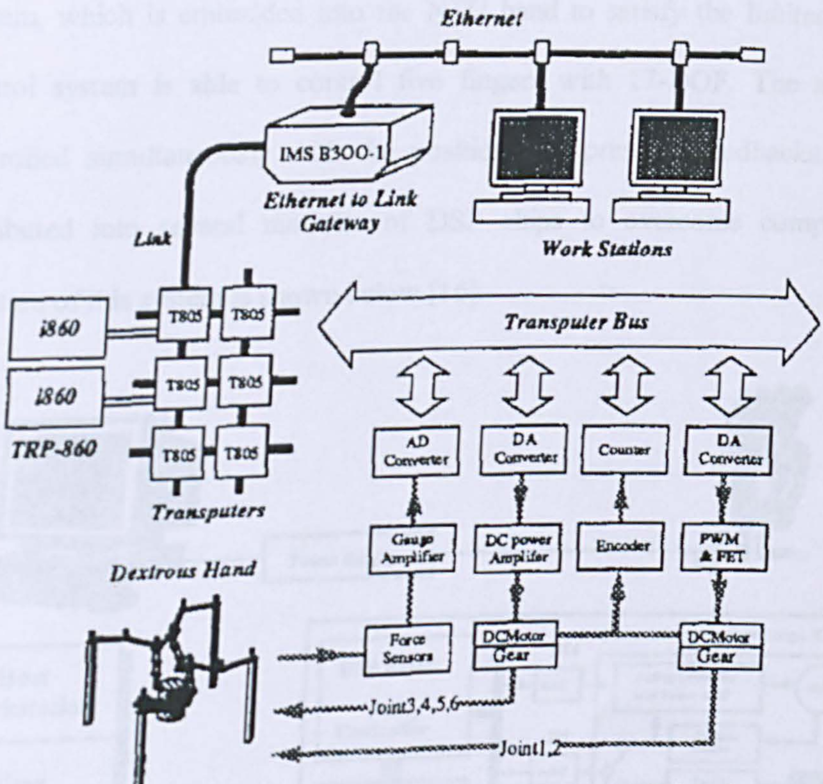


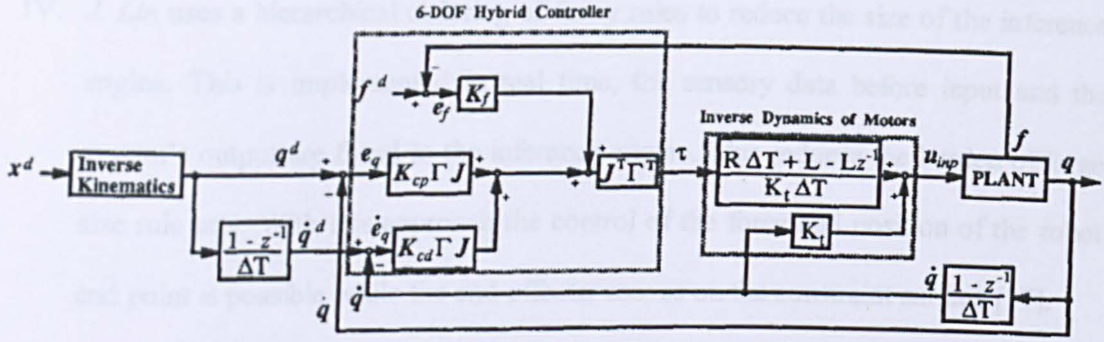
Figure 2.3: (a) the Interface for the hand, and (b) the block diagram of the controller (Adopted from [8])

The control software of this scheme has the capability of independently adjusting control efforts (output voltage) of ach motor. The control system has been implemented on an IBM compatible computer and it is claimed that this system provides excellent results and cost effective.

II. *Tetsuo Namima and Hideki Hashimoto* perform grasping and manipulating objects using a 6-axis hybrid controller approach. The control is on 24-DOF robotic hand with force and torque sensors, equipped to detect forces and moments at the fingertip. They use data from the force torque sensor to obtain appropriate position and force reference in control. The object parameters like friction and stiffness are estimated [15]. The control system structure for the hand and the 6-DOF hybrid controller for each finger are shown in Figure 2.4 below.



(a) The Interface for the 24-DOF robotic hand



(b) 6-DOF hybrid controller for each finger

Figure 2.4: (a) Interface for the hand and (b) 6-DOF hybrid controller for each finger

(Adopted from [15])

III. *Li-Ren Lin and Han-Pand Huang* developed a special design compact control system, which is embedded into the NTU hand to satisfy the limited space. The control system is able to control five fingers with 17-DOF. The actuators are controlled simultaneously with the position and pressure feedbacks. Control is distributed into several modules of DSP chips to overcome complexity. The structure of this system is shown below [16].

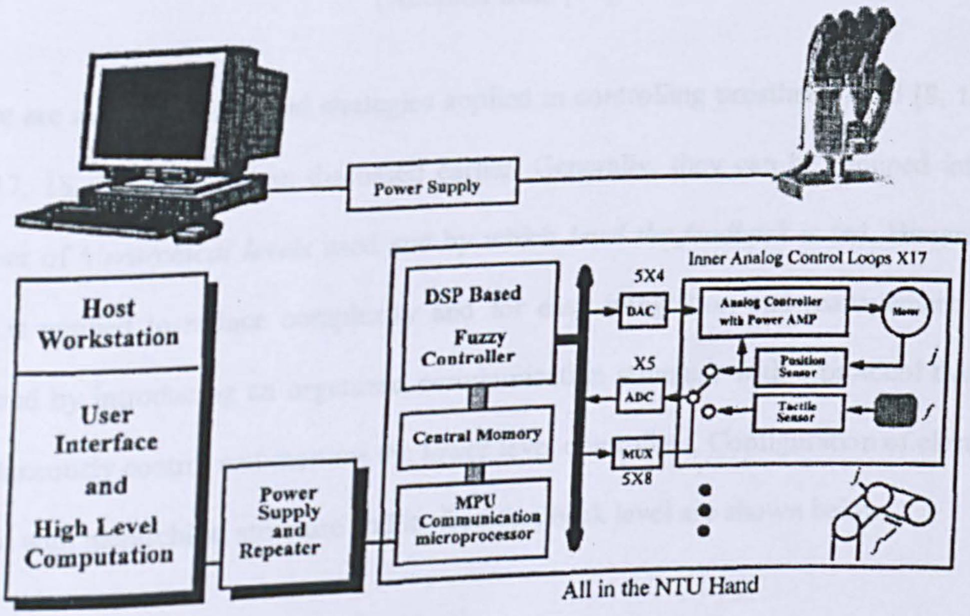


Figure 2.5: Control Structure implemented to the NTU Hand (Adopted from [16])

IV. *J. Lin* uses a hierarchical ordering of fuzzy rules to reduce the size of the inference engine. This is implemented in real time, the sensory data before input and the system's output are fused to the inference engine. This reduces the burden of large size rule sets. With this approach the control of the force and position of the robot end point is possible while the end-effecter moves on the constraint surface [17].

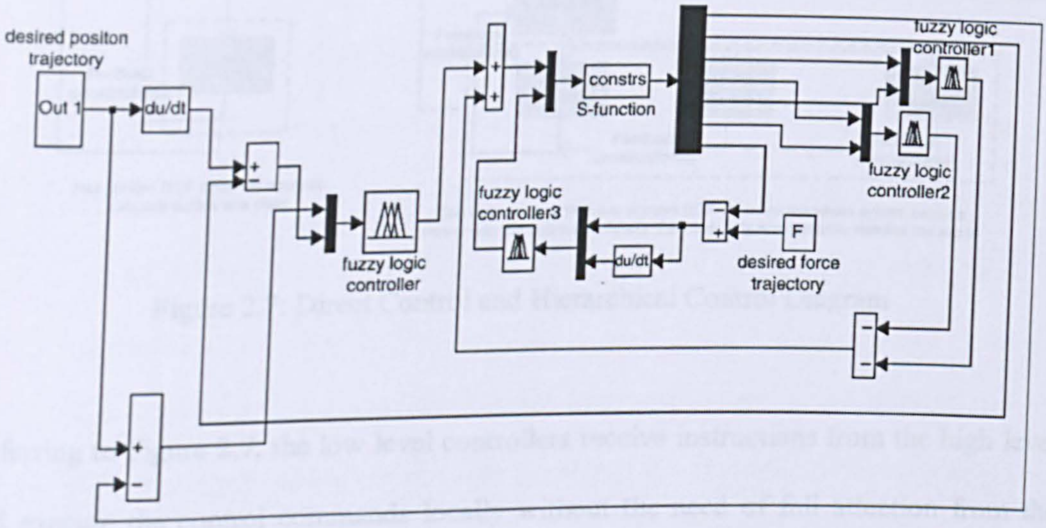


Figure 2.6: Block diagram of the hybrid position/force control system

(Adopted from [17])

There are many methods and strategies applied in controlling prosthetic hand [8, 13, 15, 16, 17, 18, 21, 23] as been discussed earlier. Generally, they can be grouped into the number of *hierarchical levels* used and by which *level the feedback* is fed. Hierarchical level is applied to reduce complexity and for easy integration and management. It is achieved by introducing an organized communication structure with a protocol that can simultaneously control and monitor the lower level controllers. Configuration of electrical control with hierarchical structure and its feed feedback level are shown below:

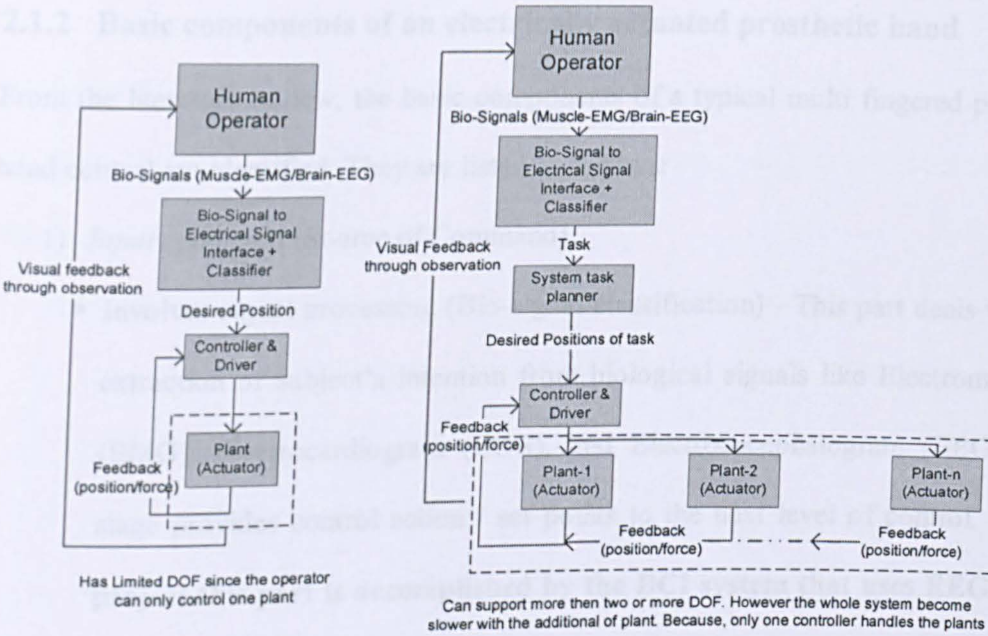


Figure 2.7: Direct Control and Hierarchical Control Diagram

Referring to Figure 2.7, the low level controllers receive instructions from the high level and execute the control commands locally without the need of full attention from the upper level controller. The lower level controllers at any moment would be able to respond back to the upper level whenever necessary. This approach reduces overhead communication activities and improves speed of the control since not all feedback information is given back to the main controller during task execution.

2.1.2 Basic components of an electrically actuated prosthetic hand

From the literature review, the basic components of a typical multi fingered prosthetic hand control are identified. They are listed as follows:

1) *Input command* (Source of Command)
   - Involves signal processing (Bio-signal classification) - This part deals with the extraction of subject's intention from biological signals like Electromyogram (EMG), Electrocardiogram (ECG), and Electroencephalogram (EEG). This stage provides control action / set points to the next level of control. **In this project this part is accomplished by the BCI system that uses EEG signal from the subject.** For testing purposes the BCI command is imitated using computer or microcontroller.
2) *Coordination of local controllers* (High level part in hierarchical control)
   - Involves communication and task planning, dealing with high level and low-level control.
3) *Local control* (low-level)
   - Manipulation problems using (P, PI, PID or advance control like fuzzy, Neuro fuzzy and etc.)
4) *Feedbacks from sensors/transducers* (position, pressure, slip .etc)
5) *Actuations* (DC motor/ Pneumatic/ Shrink cable)

Hardware consideration

The implementation of the multi-fingered prosthetic hand control involves hardware and software.

From the literature review, the following parts are typically incorporated:

1) Microprocessors/Microcontroller (Speed, size, build-in module, power consumption)
2) ADC and DAC for signal conversion
3) Actuator drivers (Drive method – switching/PWM, Linear)
4) Feedback circuitries (Amplification, filtering, signal conversion)

There are many approaches of implementing a control algorithm for multi-fingered prosthetic hand. Han Huang Pang [5] in 1996 used a DSP chip to execute his control algorithm. The approach requires external ADC and external digital to analog converter DAC for data conversions. Power control for actuators is established by Power Amplifiers. This method consumes space and high power loss at the drivers. In 2003 Giuseppe Casalino from University of Di Genova, Italy implemented hand control system inside FPGA but still use external ADC. However, he used PWM method to control the actuator power which improves driver efficiency.

Recently Microchip Semiconductor introduced embedded microcontrollers that are designed to perform various task including control. Among the controllers, the PIC164431 was found to be very suitable to control multiple plants. It has embedded most of the components inside a compact chip. The use of this embedded chip may reduce power consumption and space.

2.2 Local Control Methods

In this sub review, *local control* refers to low level control that is implemented inside microcontrollers to control position and motion of segments of prosthetic fingers. Lots of control strategies are available nowadays like P, PI, PID, Fuzzy Logic, and Hybrid control [12, 22, 23]. Advanced control methods like Neuro Fuzzy and Adaptive control have also been introduced [16, 17]. Any method that offers solution to a particular control problem still has some weaknesses when dealing with uncertainties of plant behaviors. A good control performance with minimal drawbacks is achievable by properly studying and understanding the characteristics of the plant and the controller. Control parameters like plant behaviors and the aims of the implementation play a major role in developing a controller.

Fuzzy Logic offers simplicity and solves a lot of difficulties that is normally faced by conventional control. Problems like exact system model, indefinite and inconsistent system behaviors are easily tackled by Fuzzy Logic controller [16, 17, 25]. The reason for this is that Fuzzy Logic uses human knowledge (intuitive approach) in handling control problems. The use of Fuzzy Logic technique in controlling a multi-fingered prosthetic hand is believed to be very helpful since it can simplify and overcome the complex modeling problems. The comparison between conventional approach and Fuzzy Logic control is shown below.

Control Method/Strategy	Advantages	Disadvantages
Fuzzy Logic	- Use intuitive approach that employ human knowledge - Does not require exact model	- Involve rules evaluation that slowdown decision and computation.
Conventional (PI,PD,PID)	- More precise (PI,PD,PID)	- Require an exact system model for a better control performance

Table 2.1: Comparison between conventional and fuzzy logic control method

The negative side of using FLC is that, additional input parameters to the FLC may results an exponential growth to the number of rules [16]. This becomes worse when a big number of membership functions are used. The decisional loops would run slower and this degrades performance of the FLC system. The big rules also affect the centroid computation (deffuzzification). This problem however is tolerable with the use of Sugeno Deffuzzification technique. The Sugeno Deffuzzification technique simplifies the centroid computation by introducing single tone values.

2.3 Conclusion

From literature review that has been conducted the following conclusions are made:

1. Hierarchical/multilevel structure is used to control motion of the prosthetic hand that has been developed at the University of Malaya. From top to bottom, which is from BCI to finger segments, the control structure is break into three physical layers. They are the main frame, master, and slave layers. FLC is performed at the lowest layer.

2. Fuzzy logic control strategy is used to control motion for every segment of a finger. This method is found to be the most suitable to resolve the uncertainty behaviors and the complex control modeling of a multi-fingered hand. This method uses linguistic terms that can be easily understood by user.

3. The embedded chip PIC18F4431 is used as a local micro-controller/slave micro-controller, and for the master the PIC18F8720 is used.

4. A tactile sensor is used to limit movement of finger/segment especially when an obstacle is encountered. This enables object to wrap round an object and widens the hand function and capability.

THE PROSTHETIC HAND

3.0 The hand mechanism

The human hand consists of multiple degrees of freedom (DOF) that enable it to perform various tasks. It is difficult and costly to implement and control a hand with such a large number of DOFs. In the present study, only the most important DOF is considered.

The hand had been built by a group of biomedical and mechanical engineering student [52, 56]. Four fingers have been designed to have 3-DOF and a thumb with 4-DOF. Parallel plates were used to represent the three fragments of a finger, which are the distal, proximal and middle segments. For clarity, the structure of the hand and the fingers are depicted below.

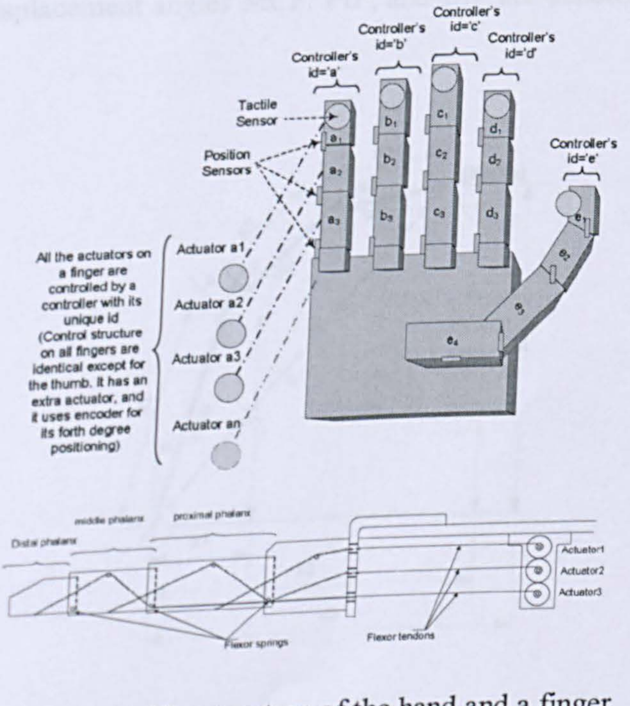


Figure 3.1: The structure of the hand and a finger

DC motors with the combination of reduced gear heads are used to actuate finger segments. Each finger is equipped with position sensors at the pivot joints and tactile-pressure sensors at fingertips. Five identical controllers with identical control structure are employed to monitor and control the fingers. A flexible nylon string links each segment to a separate actuator. Counter back springs are used to return the segments back to their rest position.

3.1 Trajectory of a finger tip

Simplified skeleton structure representing a finger is used to study fingertip trajectory. The purpose is to find the relationship between displacement angles and the trajectory of segments such that a certain position can be obtained by adjusting the angles. The adjustment of these angles (also known as position control) would be discussed in the next chapter.

Referring to Figure 3.2; a1, a2, and a3 represent the length of the distal, proximal, and phalange. The displacement angles MCP, PIP, and DIP are denoted by ɵ1, ɵ2, and ɵ3 respectively.

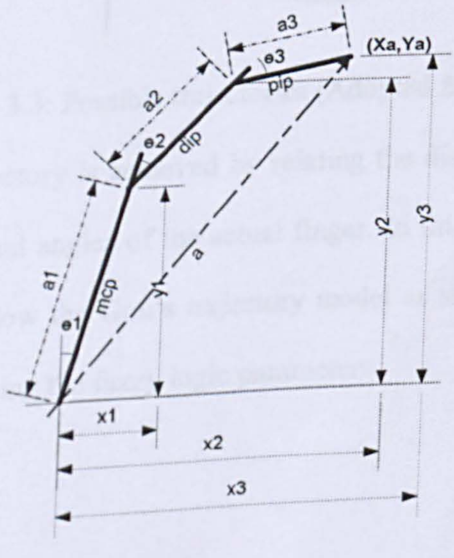


Figure 3.2: Skeleton model

Applying trigonometric rules

$$x1 = a1 \cdot \sin(\theta 1)$$
$$x2 = a1 \cdot \sin(\theta 1) + a2 \cdot \sin(\theta 1 + \theta 2)$$
$$x3 = a1 \sin(\theta 1) + a2 \sin(\theta 1 + \theta 2) + a3 \sin(\theta 1 + \theta 2 + \theta 3) \quad \text{Eq 3.1}$$

$$y1 = a1 \cdot \cos(\theta 1)$$
$$y2 = a1 \cdot \cos(\theta 1) + a2 \cdot \cos(\theta 1 + \theta 2)$$
$$y3 = a1 \cos(\theta 1) + a2 \cos(\theta 1 + \theta 2) + a3 \cos(\theta 1 + \theta 2 + \theta 3) \quad \text{Eq 3.2}$$

The trajectory of the finger tip can be represented by

$$y3 = \frac{(a1 \cdot \cos(\theta 1) + a2 \cdot \cos(\theta 1 + \theta 2) + a3 \cdot \cos(\theta 1 + \theta 2 + \theta 3)) \cdot x3}{a1 \sin(\theta 1) + a2 \sin(\theta 1 + \theta 2) + a3 \sin(\theta 1 + \theta 2 + \theta 3)} \quad \text{Eq 3.3}$$

With the three degree of freedom finger, various trajectories are possible as shown on the picture below [4].

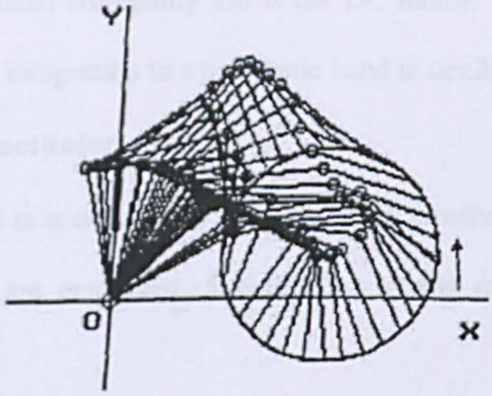


Figure 3.3: Possible trajectories (Adopted from [4])

A human mimicked trajectory is achieved by relating the displacement angles $\theta 1$, $\theta 2$, and $\theta 3$ to the displacement angles of the actual finger. In this design the displacement angles is designed to follow the Gou's trajectory model as shown on Figure 3.4 [39]. This is achieved by adjusting the fuzzy logic parameters.

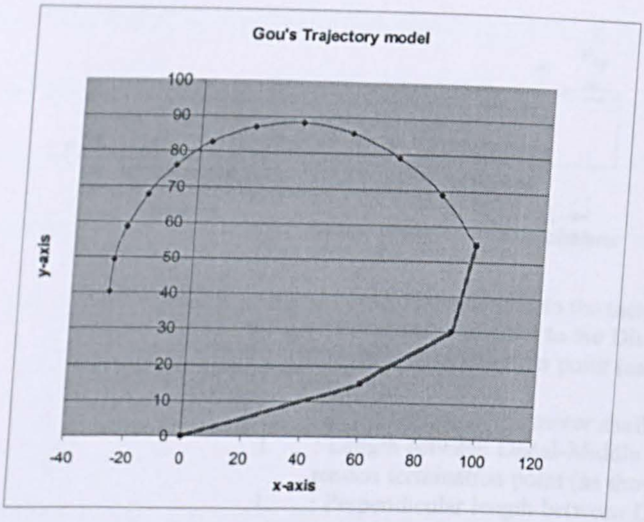


Figure 3.4: Gou's finger trajectory

3.2 Actuation

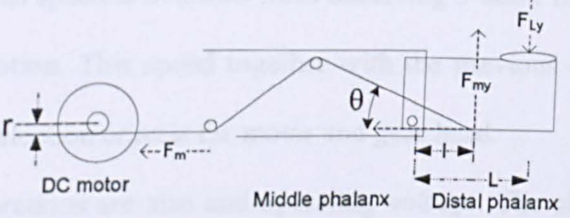
Possible actuation methods for a prosthetic hand includes; pneumatic, DC motor, and artificial muscle. The most commonly use is the DC motor. This method is relatively easier to control and its integration to a prosthetic hand is flexible.

3.2.1 DC motor as actuator

Geared dc motor is used to actuate every segment of the prosthetic fingers including the thumb. 16 DC motors are employed. Specification of the motors is determined as follows:

1. The motor must be able to withstand 400g load on fingertip, this load is within typical handling for an adult finger [40, 41]. From this load and some other parameters of a finger, the motor torque is calculated.

   Considering there is no friction, motor torque (T_m) is $T_m = F_m \times r$, where (F_m) is the force vector and (r) is the distance perpendicular to the force towards the axis of rotation, please refer to Figure 3.5.



F_{Ly} : Force perpendicular to the tactile sensor surface
F_{my} : Force perpendicular to the Distal phalanx back plate at the tendon termination point (as shown on the above figure).
F_m : String tension at the motor shaft.
l : Length between Distal-Middle phalanx joint to Distal tendon termination point (as shown on the above figure).
L : Perpendicular length between Distal-Middle phalanx joint to the direction of F_{Ly}
r : Diameter of the motor shaft.

Figure 3.5: Force estimation on a middle finger

Vertical moments as in position on Figure 3.5:

Momen due to load on fingertip

$$M_{Ly} = F_{Ly} \times L$$

Momen due to the string pulls by the motor

$$M_{my} = F_{my} \times l \qquad F_{my} = F_m Sin(\theta)$$

To make the segment to move upward M_{my} must be bigger then M_{Ly} (M_{my}>M_{Ly}).

$$F_m Sin(\theta) \times l > F_L \times L$$

$$F_m > \frac{F_L \times L}{Sin(\theta) \times l}$$

The minimum torque required is:

$$T_m > \frac{F_L \times L}{Sin(\theta) \times l} \times r \ ,$$

By using L, l, r, θ values for the middle finger from (Appendix-4A), the minimum torque is determined.

T_m > 0.036189 Nm

2. To mimic the human hand motion, the speed of actuator is important. Referring to angular speed of a finger, for an adult hand, a speed around 4 rad/sec is

considered. This speed is obtained from observing 3-adult fingers during normal hand grasp motion. This speed together with the previous calculated torque is used as main selection criteria for motor and gear head.

3. Others considerations are size and operating voltage. The size must not be too bulky and the operating voltage is around typical power supply range which is 12V.

From the criteria that have been discussed, a geared DC motor model from Faulhaber (1331T012S-motor, 1/14-gear type, 159:1-reduction ratio) is selected [Appendix-3A and Appendix-3B]. For the Thumb's 4th DOF actuation, the same motor is selected but with additional of (IE2-512) magnetic encoder [Appendix-3C].

Calculations based on motor parameters are carried out to see DC motor operating performance. First we determine torque at the motor shaft that is required to drive the gear to produce a torque greater than 0.0362 Nm (this 0.0362 Nm is calculated previously in section 3.2.1). The efficiency of the gear head is 60% [Appendix-3B], means that 40% of the power developed by the motor would be lost in the gear head. According to the manufacturer operational manual [44] the simplest method of accounting for gear head losses is to increase the torque requirement and make the calculations as if the gear head were 100% efficient. In this case we increase the torque by 40% (0.0362*1.4 = 0.04704 Nm). Torque reflected to the motor is therefore 0.04704/159 Nm = 0.000296 Nm or 0.04188 Oz.in. We use this unit in order to match to the units in the characteristic graph provided by the DC motor manufacturer. This is a minimum torque required to enable a segment to move under 400g load on the middle finger.

The motor must be able to support speed of the geared shaft to be not less than 4 rad/sec. At this condition, the motor speed is 4*159=639 rad/sec or 6073.35 rpm. Referring to the Figure 3.6, the torque capability at this speed is bigger than the critical limit (0.55>0.041887) Oz.In or (0.003909 > 0.04704) Nm.

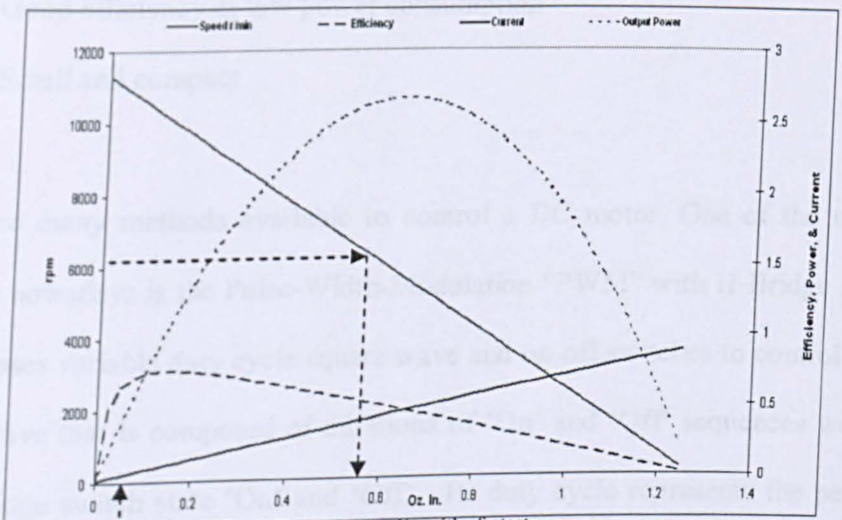


Figure 3.6: Characteristic of the DC motor

The motor therefore operates at approximately 90% of its no load speed and about 10% to 30% of its stall torque. This is the typical operating range for motor applications. This operating range gives good efficiency and also results in a better life characteristic [43, 44].

3.2.2 The actuator driver

Another important part of motion control is the actuator driver. Here the driver translates speed and directional signal from a microcontroller into motor input (i.e the power supply to the DC motor). The signal is a control action resulting from the execution of the control algorithm by the microcontroller. The control signals are in the form of PWM signal for the motor speed and high/low signal for the motor direction.

The following factors have been taken into considerations in designing the driver:

1. It has to operate within the range of motor current and voltage.
2. Capable to perform bidirectional drive
3. Low noise operation
4. Good efficiency & low power consumption
5. Small and compact

There are many methods available to control a DC motor. One of the most famous methods nowadays is the Pulse-Width-Modulation "PWM" with H-Bridge switch. This method uses variable duty cycle square wave and on-off switches to control power. The square wave that is composed of durations of 'On' and 'Off' sequences would trigger the H-bridge switch state 'On' and 'Off'. Its duty cycle represents the percentage of power delivered to the load.

The equation for this phenomenon is:

Output Voltage (Actuator supply),

$$V_{out} = V_{in} * [T_{on}/(T_{on}+T_{off})]$$

$$= V_{in} * (T_{on}/T)$$

Where, T = Switching period ($T_{on}+T_{off}$),

T_{on} = Switch_On time

T_{off} = Switch_Off time

V_{in} = Main supply

For the directional control, there are four switches that form the H-Bridge switch. Depending on which pair of the diagonal switches is active the polarity of the load source would be determined and this allows motor direction control. This is clearly shown by the Figure 3.7.

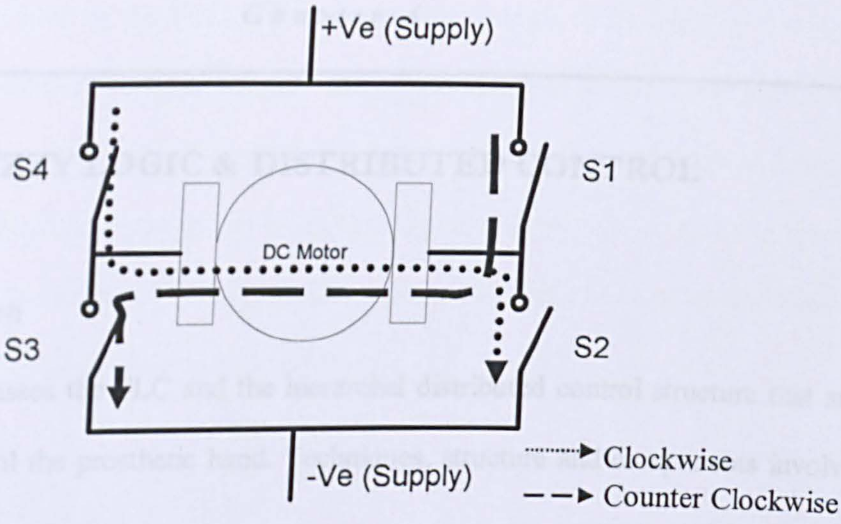


Figure 3.7: The H-Bridge Switch

The power delivered to the motor is determined by how long the switch is ON. Additional logical circuit is normally added to this arrangement that allows easier microcontroller integration. From a survey, an integrated H-bridge switch "LMD18201" is found suitable to drive the selected motor model. In this early design stage a low cost package is used. This chip supports up to 3 Amps with its operating voltage ranges around 12 to 24-Volts.

3.3 Conclusion

At the beginning of this chapter, study of the finger mechanism helps us to see its possible trajectories. The conventional actuation by using DC motor is selected to actuate the hand; this method is relatively easier and inexpensive. The motor operating region is determined for its optimum operation. Hand motion control is accomplished by the use of H-bridge drivers to operate the DC motors; this allows low signal interfacing to control the DC motor.

Chapter 4

FUZZY LOGIC & DISTRIBUTED CONTROL

4.0 Introduction

This Chapter discusses the FLC and the hierarchal distributed control structure that are employed to control the prosthetic hand. Techniques, structure and components involve are described.

4.1 Overview of the system

As mentioned in introduction, the prosthetic hand control system consists of five separate identical microcontrollers called slave. Each salve microcontroller is programmed to execute a fuzzy logic control algorithm together with a system handler. The main purpose is to enable hierarchical integration and allows the actuators to be controlled locally. Those slave microcontrollers are coordinated by another microcontroller called 'Master'. The structure of this arrangement is shown on Figure 4.1 below.

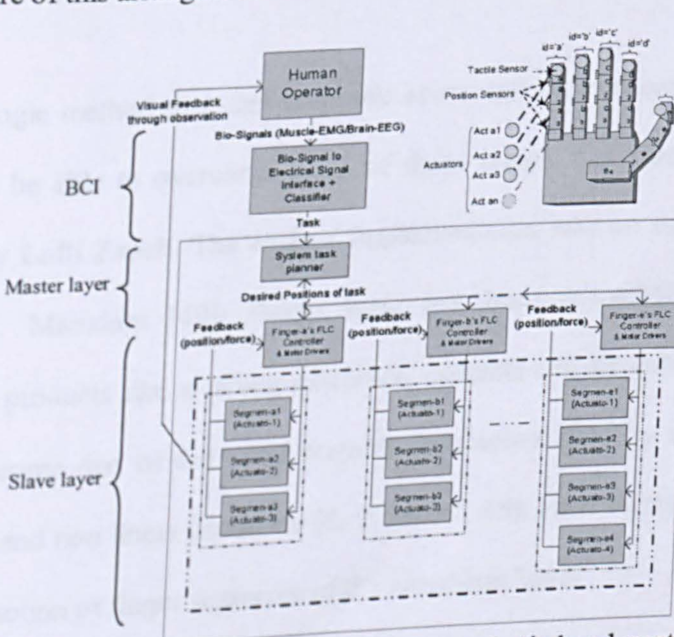


Figure 4.1: Proposed structure for the prosthetic hand control system

The control parameter in the FLC is the displacement angle. This angle is measured between two adjacent segments of a finger using potentiometer placed at the joint. Each joint that has active segment to move is equipped with a potentiometer. However, on forth DOF thumb a magnetic encoder is use instead of the potentiometer, this is due to special arrangement on this segment. Voltage divider concept is applied on potentiometers to give a proportional values related to displacement.

4.2 Fuzzy Logic Controller (FLC)

The purpose of the FLC is to overcome difficulties in controlling multiple degrees-of-freedom (DOF) that the typical conventional controllers have difficulty dealing with, mainly due to imperfect modeling of a system [25, 46, 9]. The conventional controllers like PID requires a good system model before it can effectively control the system. This is hard to achieve for a multi-DOF prosthetic hand. The existences of non-uniform friction at joints and tendons and also unknown load conditions during prosthetic hand operation have made the modeling of the system rather difficult if at all possible [8]. A robust controller that could overcome or tolerate those problems is therefore necessary.

A Fuzzy Logic method that uses linguistic terms and experiences inherent to humans is believed to be able to overcome some of those issues. The Fuzzy Logic concept was proposed by Lofti Zadeh. The earliest implementation was on steam engine control by Ebrahim H. Mamdani [49]. Fuzzy logic has been successfully applied in many commercial products like washing machines, cameras and air conditioners. The method has now become one of the more popular approaches used by researchers in solving inconsistent and non linear problems [8, 9, 25, 47, 48]. Here Fuzzy logic is employed to control the motion of finger segments of the prosthetic hand.

4.2.1 Constructing a Fuzzy Logic Controller

A typical FLC system is shown below. The arrangement is similar to that of basic conventional control system. The difference is that the FLC makes use of linguistic terms and human experiences to determine control actions.

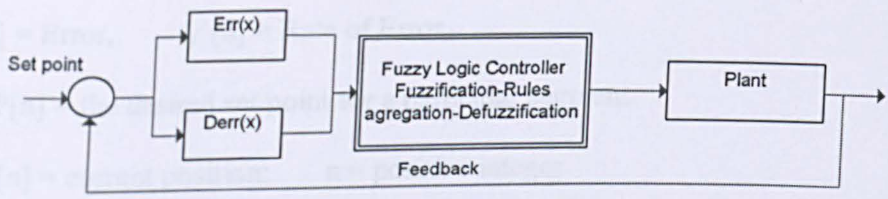


Figure 4.2: A typical Fuzzy Logic Control system

The FLC block possesses a few steps called *Fuzzification*, *Rules evaluation, Aggregation* and *Deffuzzification*. *Fuzzification* translates the crisp input value(s) into sets of active linguistic term(s) together with its fuzzy value. The *Rules evaluation & Aggregation* determine what the Fuzzy Logic output should be for the Fuzzified inputs by referring to the *Rules*. Finally, *Deffuzzification* translates the results of *Rules Aggregation* into a crisp value. This value is a control action for the system that is then applied to the "plant" In this case a DC motor. The details about these FLC steps are described below.

4.2.1.1 Defining Fuzzy sets/Membership Function (MF)

Fuzzy sets/membership functions (MF) are defined over the whole possible range of the variables called the Universe of Discourse. The purpose of MF is to represent the input and output variables in linguistic terms, such that a control decision by *word* can be carried out. In this thesis, as mentioned earlier the control parameter is displacement angle. As input variables to the FLC this angle is translated into Error and Rate of Error, in which the error is the different of (set angle / set point) to the current angle. The relation of the FLC variables is show below.

$$y[n] = SetP[n] - Pos[n] \ldots\ldots\ldots\ldots \text{Eq 4.1}$$

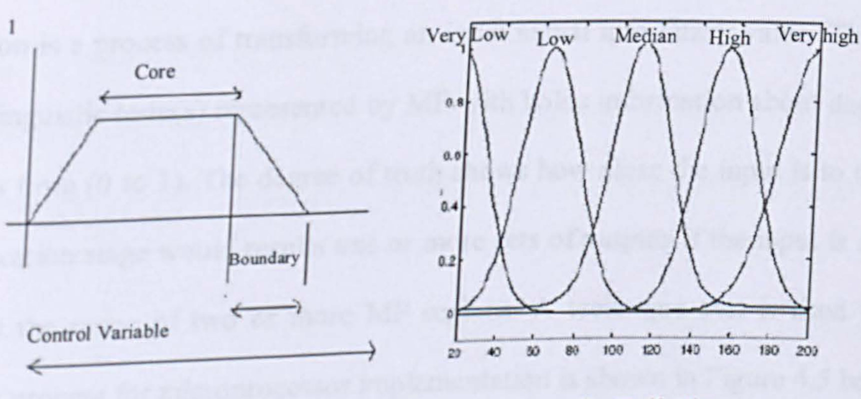
$$y'[n] = y[n] - y[n-1] \ldots\ldots\ldots\ldots\ldots \text{Eq 4.2}$$

y[n] = Error, y'[n] = Rate of Error,

SetP[n] = the desired set point for a particular segment

Pos[n] = current position; n = positive integer

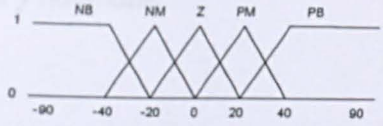
A MF can be represented by any shape like triangular, trapezoidal or bell-shape as shown in Figure 4.3 below [50]. The vertical axis represents Fuzzy value, while horizontal axis represents control variable.



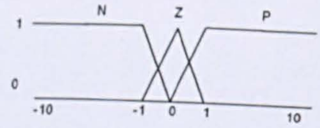
(a) Trapezoidal (b) Bell-Shape

Figure 4.3: Shapes of membership

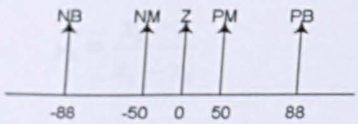
In this thesis trapezoidal and triangular shapes are used. This triangular shape is comparatively easier to be represented inside a microcontroller. MFs used in this thesis are shown on Figure 4.4.



$y_m[n]$ (Error)



$y'_m[n]$ (Rate of Error)



$Y_m[n]$ (PWM duty cycle)

n= Positive integer
m= Membership number

	m
NB	1
NM/N	2
Z	3
PM/P	4
PB	5

Figure 4.4: The proposed Membership Functions

4.2.1.2 Fuzzification

Fuzzification is a process of transforming an input signal into Fuzzy value. The Fuzzy value is a linguistic term(s) represented by MF with holds information about degrees of truth ranges from (0 to 1). The degree of truth shows how close the input is to the MF. This fuzzification stage would results one or more sets of outputs if the input is overlap or within in the range of two or more MF regions. A technique that is used for the fuzzification process for microprocessor implementation is shown in Figure 4.5 below.

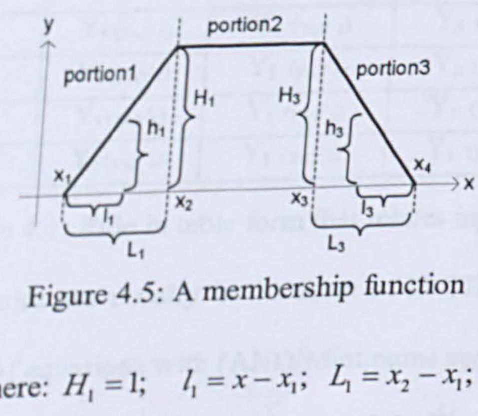


Figure 4.5: A membership function

$h_1 = \frac{l_1}{L_1} H_1$ Where: $H_1 = 1$; $l_1 = x - x_1$; $L_1 = x_2 - x_1$; x is input

Using x and y notations;

$$h_1 = \frac{x - x_1}{x_2 - x_1}$$

$$y_1 = \frac{x - x_1}{x_2 - x_1}$$

Applying the same procedure for the portion2 and portion3 in the figure 4.4 we obtain

$$y_2 = 1 \ , \qquad y_3 = \frac{x_4 - x}{x_4 - x_3}$$

The Fuzzified value for all input membership functions can be represented as

$$y = \begin{cases} \dfrac{x - x_1}{x_2 - x_1} \text{........;} & x_1 \le x < x_2 \\ 1 \text{.......................;} & x_2 \le x \ge x_3 \\ \dfrac{x_4 - x}{x_4 - x_3} \text{.........;} & x_3 < x \le x_4 \end{cases}$$

4.2.1.3 Rules Evaluation & Aggregation

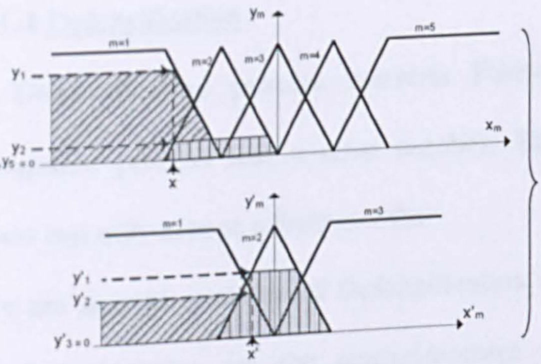
This part makes use of the Fuzzified results to determine a control action based on rules that have been specified. This process produces another set of linguistic terms involving output MF.

Err (Input2)		DErr (Input1)		
		y'_1	y'_2	y'_3
	y_1	$Y_5\,(y_1,y'_1)$	$Y_5\,(y_1,y'_2)$	$Y_4\,(y_1,y'_3)$
	y_2	$Y_5\,(y_2,y'_1)$	$Y_4\,(y_2,y'_2)$	$Y_3\,(y_2,y'_3)$
	y_3	$Y_4\,(y_3,y'_1)$	$Y_3\,(y_3,y'_2)$	$Y_2\,(y_3,y'_3)$
	y_4	$Y_3\,(y_4,y'_1)$	$Y_2\,(y_4,y'_2)$	$Y_1\,(y_4,y'_3)$
	y_5	$Y_2\,(y_5,y'_1)$	$Y_1\,(y_5,y'_2)$	$Y_1\,(y_5,y'_3)$

Output
$Y_{m\,max}$
$Y_{5\,max}$
$Y_{4\,max}$
$Y_{3\,max}$
$Y_{2\,max}$
$Y_{1\,max}$

Table 4.1: Rule in table form that relates input and output MFs

In Fuzzy Logic rules are usually in the form of IF-THEN statements. However, here rules in the form of equations with (AND/Minimum) and (OR/Maximum) operators are used. This is clearly shown on the Figure 4.6 and the Firmware code below.



Aggregation

$$aggr = \begin{cases} Y_5 = (y1 \vee y'1) \wedge (y1 \vee y'2) \wedge (y2 \vee y'1) \\ Y_4 = (y1 \vee y'3) \wedge (y2 \vee y'2) \wedge (y3 \vee y'1) \\ Y_3 = (y2 \vee y'3) \wedge (y3 \vee y'2) \wedge (y4 \vee y'1) \\ Y_2 = (y3 \vee y'3) \wedge (y4 \vee y'2) \wedge (y5 \vee y'1) \\ Y_1 = (y4 \vee y'3) \wedge (y5 \vee y'2) \wedge (y5 \vee y'3) \end{cases}$$

$\wedge = \max imum\ function$ / or operator
$\vee = \min imum\ function$ / and operator

Figure 4.6: Rules Evaluation & Aggregation

The firmware code for (Rules Evaluation & Aggregation) of the FLC using Basic programming language is shown below.

```
'==================== "Rules evaluation & Aggregation" =======
Rules:

MF5out1[0] = 0
MF4out1[0] = 0
MF3out1[0] = 0
MF2out1[0] = 0
MF1out1[0] = 0

MF5out1[0] = (MF1in1[0] MIN MF1in2[0]) MAX (MF5out1[0])
MF5out1[0] = (MF1in1[0] MIN MF2in2[0]) MAX (MF5out1[0])
MF5out1[0] = (MF2in1[0] MIN MF1in2[0]) MAX (MF5out1[0])

MF4out1[0] = (MF1in1[0] MIN MF3in2[0]) MAX (MF4out1[0])
MF4out1[0] = (MF2in1[0] MIN MF2in2[0]) MAX (MF4out1[0])
MF4out1[0] = (MF3in1[0] MIN MF1in2[0]) MAX (MF4out1[0])

MF3out1[0] = (MF2in1[0] MIN MF3in2[0]) MAX (MF3out1[0])
MF3out1[0] = (MF3in1[0] MIN MF2in2[0]) MAX (MF3out1[0])
MF3out1[0] = (MF4in1[0] MIN MF1in2[0]) MAX (MF3out1[0])

MF2out1[0] = (MF3in1[0] MIN MF3in2[0]) MAX (MF2out1[0])
MF2out1[0] = (MF4in1[0] MIN MF2in2[0]) MAX (MF2out1[0])
MF2out1[0] = (MF5in1[0] MIN MF1in2[0]) MAX (MF2out1[0])

MF1out1[0] = (MF4in1[0] MIN MF3in2[0]) MAX (MF1out1[0])
MF1out1[0] = (MF5in1[0] MIN MF2in2[0]) MAX (MF1out1[0])
MF1out1[0] = (MF5in1[0] MIN MF3in2[0]) MAX (MF1out1[0])

RETURN
'==================== "END" =======
```

4.2.1.4 Defuzzification

The Defuzzification process converts Fuzzy values obtained from the previous *Aggregation* process into a crisp number. This is necessary since the next control process can only accept a finite number.

There are several methods of Defuzzification technique. The well known methods in control engineering are the centroid/centre of gravity technique (COG) and the weighted-average. The weighted-average or also known as Sugeno-Defuzzification technique is found to be most suitable for an 8-bit microprocessor implementation since this technique offers simple computation while still giving a comparable result to the COG. The equation for the Sugeno-defuzzification is shown below:

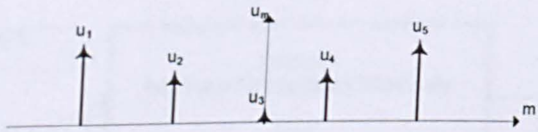


Figure 4.7: Output MF for Sugeno-Deffuzzification

$$U=\left(\frac{(Y_1\times u_1+Y_2\times u_2+\ldots+Y_m\times u_m)}{(Y_1+Y_2+\ldots+Y_m)}\right) \quad \text{(Equation 4.1)}$$

$$U=\left(\frac{\sum_{1}^{m}(Y_m u_m)}{\sum_{1}^{m}(Y_m)}\right) \quad \text{(Equation 4.2)}$$

u_m= Output MFs.

Y_m= Fuzzy values from previous stage of "Rules evaluation & Aggregation".

U = A crisp value representing FLC output.

The crisp value 'U' is PWM duty cycle representing power delivered to a DC motor to move a finger segment.

4.2.2 Sharing Fuzzy Logic routine

The previous control block can be modified to handle more than one plant/actuator. For a finger, it consists of several segments that utilize three or more similar actuators as shown on Figure 4.8 below. Directly applying the previous control block without modification would result in redundancy that consumes unnecessary space, power and cost. This can be overcome by simultaneously sharing the FLC block for one finger. This means a unit of microcontroller is used to control and monitor a finger/ thumb.

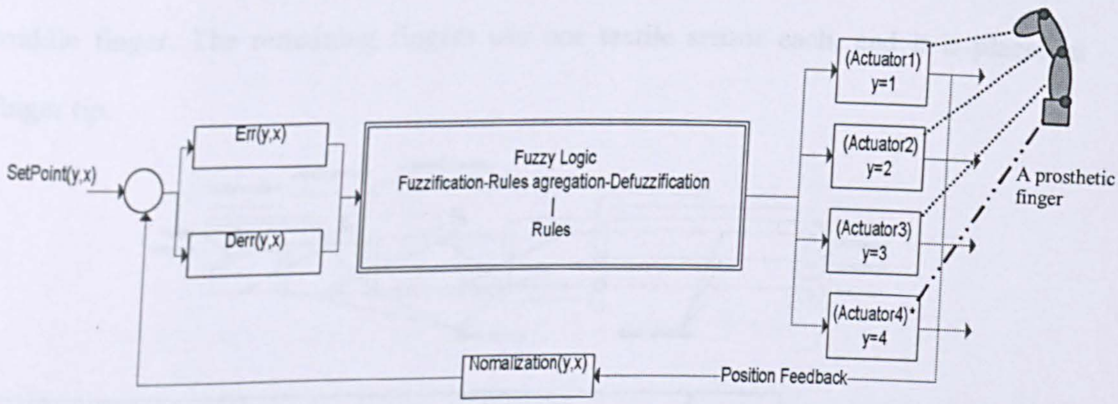


Figure 4.8: The proposed control system to control one finger

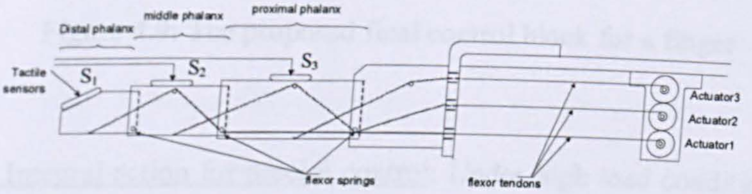
This configuration not only reduces space and power consumption but also offers a systematic wiring for feedback and actuators.

4.2.3 Shape adaptation & safety

Depending solely on *position feedback* is insufficient for a prosthetic hand control, especially when performing tasks that involve interaction with objects. The hand is expected to closely follow the shape of the held object and also maintain a certain grip force when holding the object. For safety reasons, the actuated segment has to stop if an obstacle is encountered. This prevents damage to the object and also to the system.

The human hand uses Mechanoreceptors to sense contact and also pressure on skin. This signal is sent to the brain, and from experience in mind human can easily handles an object, for example holding an egg. To mimic some of these features tactile sensors together with rules are employed on the prosthetic hand.

The following rules (in the table below) are applied for shape adaptation. The S_3, S_2, S_1 represents pressure on distal, middle and proximal phalanx respectively as shown on the figure below. At the moment, the three sensors configuration is only applied to the middle finger. The remaining fingers use one tactile sensor each, and it is placed at finger tip.



$S_3>S_{3limit}$	$S_2>S_{2limit}$	$S_1>S_{1limit}$	Actuation Distal	Actuation Middle	Actuation Proximal	Case
0	0	0	1	1	1	4
0	0	1	0	0	0	1
0	1	0	0	0	1	2
0	1	1	0	0	0	1
1	0	0	0	1	1	3
1	0	1	0	0	0	1
1	1	0	0	0	1	2
1	1	1	0	0	0	1

Case	Actuator (1, 2, 3)
1	All are off
2	Only Act 1 is on
3	Only Act 3 is off
4	All are on

Table 4.2: Sequences for shape adaptation

Table 4.2 describes the possible conditions. Without any load (case 4) all the actuators are set to active. When pressure at S_3 exceeds the threshold value S_{3limit} (case 3) the actuator of that segment is turned to off. The same rule is applied for the next segment 2. In any situation if $S_1>S_{1limit}$ all the actuator will be off.

4.2.4 A practical Fuzzy Logic Controller for the prosthetic hand

Several more modifications are made to the control block on section 4.2.2 to make it more practical and precise in executing tasks. This includes Additional Integral action and Normalization.

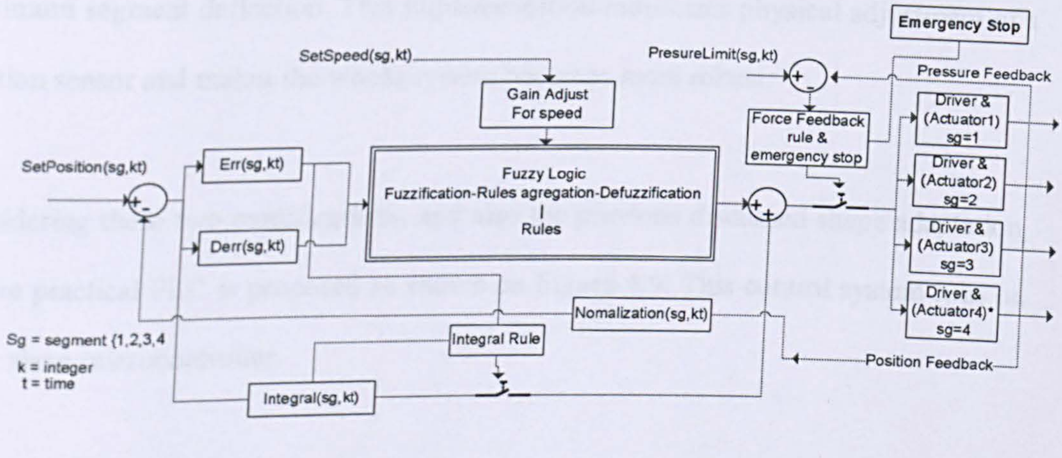


Figure 4.9: The proposed final control block for a finger

1. Additional Integral action for precise control: Under high load condition PD type of FLC cannot eliminates zero steady state error. Reduction of this error is significant when a precise control is desired. Hence, integral action is included into the FLC. This action is implemented by accumulating previous error values and it is then scaled and added to the FLC output to increase the PWM duty cycle. This increases power of the actuator that then allows it to further adjust the particular segment towards its set point. The integral action will only be activated when the manipulator (i.e. the controlled segment) fails to reach its targeted set point after certain period of time. This implementation is depicted as Integral Rule on Figure 4.9. This rule is adjustable during calibration.

2. Normalization: The purpose of Normalization is to have a standard input for the FLC. Regardless of different voltage level from position sensor at referent point, the controller capable to use this input signal to represent the actual position of the segment. However, the sensor must be able to give a voltage change from reference point to its maximum segment deflection. This implementation minimizes physical adjustment of a position sensor and makes the whole system becomes more robust.

Considering these two modifications, and also the previous discussed shape adaptation, a more practical FLC is proposed as shown on Figure 4.9. This control system runs on every slave microcontroller.

4.3 Distributed Control

To effectively control a system consisting of duplicate identical sub-systems, the use of a central unit is very helpful. This approach is called distributed control. Distributed control has been widely used in industry to monitor and control large number of plants. It is also implemented in robotic systems [16, 21]. In the robot system, the scattered controllers on the robot body are linked to a main controller unit such that all instructions from upper level are only sent to the main controller. This idea reduces complexity and easy system management. A basic distributed control system is shown on Figure.4.10. It normally consists of minimum of 2 level networks.

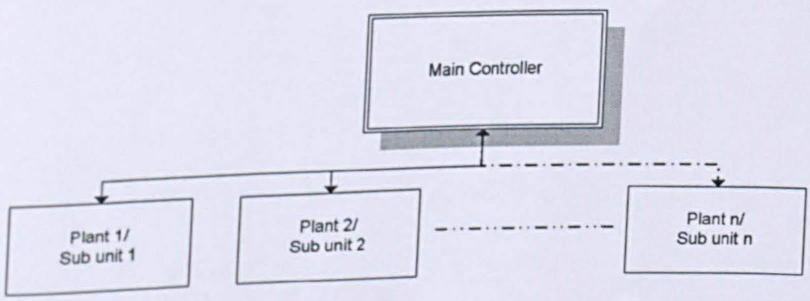


Figure 4.10: Basic Distributed control system

For the prosthetic hand control system, each sub-system (which is the slave microcontroller) is programmed to always listen to the commands given by the main microcontroller. A command is executed when a match id with valid format found in the command code. The topology of the implemented distributed control is shown in the Figure 4.11.

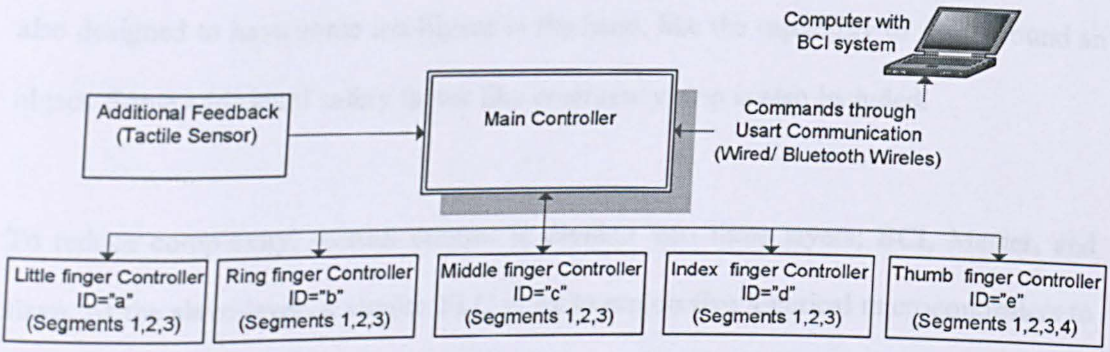


Figure 4.11: The Distributed control for the prosthetic hand

The use of this control topology is very helpful in making the overall system to be more systematic and integrate-able to the other system such as BCI unit.

4.4 Conclusion

The design of a practical FLC for the UM-Prosthetic hand has been presented. Typical PD type of Fuzzy Logic is designed to simultaneously handle three to four identical actuators. Construction of the PD type FLC to control position/angle of a segment is described. This involves identification of membership function, fuzzification, rules evaluation and rules aggregation and finally defuzzification. This hand control system is also designed to have some intelligent to the hand, like the capability to wrap around an object. Some additional safety factor like emergency stop is also included.

To reduce complexity, overall control is divided into three layers; BCI, Master, and slave. At the slave layer, a similar FLC is let to run on five identical microcontrollers to control all the fingers and the thumb. These microcontrollers are coordinated by Master microcontroller, which is located one level higher. This integration not only reduces complexity but it is also more manageable and easy to be integrated to with the BCI.

IMPLEMENTATION

5.0 Introduction

Here the implementation of hardware and software for the prosthetic hand controller is discussed. In this part the theoretical aspects that have been discussed earlier is translated into a practical form.

5.1 The controller hardware system

A controller board is developed to realize the Distributed Fuzzy Logic Control system. Figure 5.1 depicts the controller board. It is composed of the following components:

1 Master microcontroller (PIC18F8720)
2 Slave microcontrollers (PIC18F4431)
3 RS232 communication
4 DC motor drivers
5 Data acquisition

PIC microcontrollers from Microchip Semiconductor Inc. are used as processor units. A PIC18F8720 is used as a master microcontroller and four PIC18F4431 as slave microcontrollers. The first PIC offers high memory space suitable for high memory demanding applications. It has also embedded in the chip with two "Universal Serial Asynchronous Receiver Transmitter" (USART) modules. These features make it most suited for monitoring and coordinating.

For the slave microcontroller, the chip has specially designed with modules that can handle motion control. It is embedded with most of the control components like Pulse Width Modulator ("PWM" for power control), Analog to digital converter ("ADC" to convert analog into digital signal), Universal Serial Asynchronous

Receiver Transmitter ("USART" as serial communication), and Quadrature Encoder Interface ("QEI" to encode signal). This integration in the form of modules into a single chip greatly improves execution time, consumes low power and takes up small space.

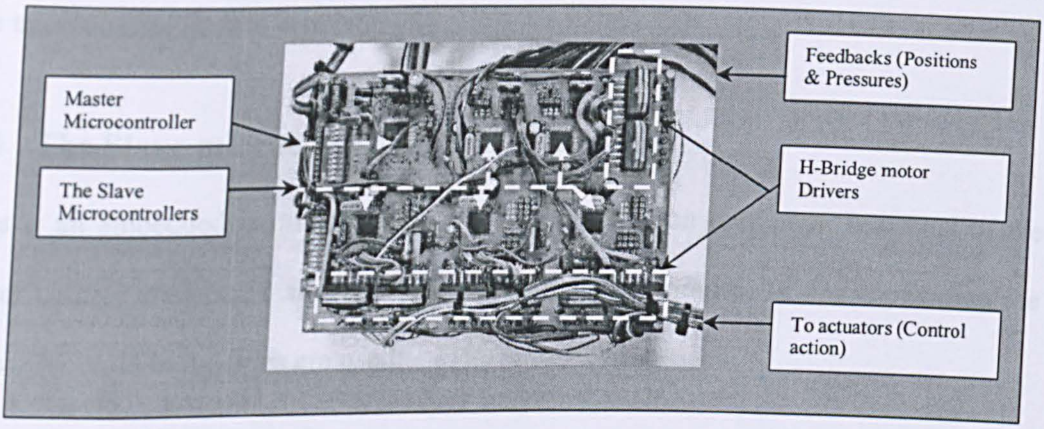


Figure 5.1: The Controller board prototype

Communication between master and slaves is established through USART interface with RS232 protocol. H-bridge switches with PWM power control are used to drive the actuators. The whole circuit of this arrangement is shown in appendix-1A.

5.2 The Master microcontroller (PIC18F8720)

The role of the master microcontroller is to be a front-end microcomputer that allows BCI-FLC communication. It coordinates and monitors the distributed fuzzy logic controllers. When executing a task, the master distributes set points to all slaves. Control is then accomplished locally by every slave units.

The built in USART modules are used to link the 'upper' to the 'low' level part of the distributed control structure (which is discussed in chapter 4). These USART modules are operated in Asynchronous mode under RS232 protocol. For the 'upper level' PC-Master, the configuration is (Baud Rate = 115200, Data Bits = 8, Parity = No Parity, Stop Bits = 1). For the 'low level' Master-Slaves, the configuration is (Baud Rate = 9600, Data Bits = 8, Parity = No Parity, Stop Bits = 1). These parameters are set by adjusting the internal registers of the microcontroller. For the baud rate setting, its register modification has to follow certain conditions because it depends on oscillator frequency and microcontroller internal logic circuitries. This is discussed in detail inside the microcontroller data manual.

5.3 The Slave microcontroller (PIC18F4431)

This is an embedded microcontroller designed for motion control. It has most of the motor control elements that are embedded as built in modules. In this application the following build in modules are used:

1 PWM signal generator.
2 USART communication module.
3 ADC module.
4 Motion Feedback module (Quadrature Encoder Interface - QEI)

The operating mode and its configuration for every module listed above are discussed in detail in the following sub-topics.

5.3.1 PWM signal generator

Pulse width modulation PWM is a method of controlling power by adjusting PWM duty cycle. There are 4 PWM channels available in this PIC. Here, letter 'x' is used to designate PWM channel. Referring to the PIC datasheet, the duty cycle is adjusted by changing 10-Bit duty cycle registers represented by CCPRxL:CCPxCON<5:4>.

A proper relationship between the registers and the effective PWM duty cycle output is determined by:

PWM_Duty_cycle= CCPRxL:CCPxCON<5:4>*Tosc*[TMR2_prescale_value])

Where the *Tosc* is 1/(oscillator frequency), and the *TMR2_prescale_value* is a constant pre-scale value set in TMR2 pre-scale register.

Here PWM frequency is set to 22 kHz. This frequency is set beyond the human audible range. This fast switching frequency is necessary for smooth PWM output.

The PWM frequency is calculated as follows:

PWM_freq = 1 / PWM_Period,

PWM_Period = [(PR2) + 1]*4*Tosc*(TMR2_prescale_value).

5.3.2 USART communication module (PIC18f4431)

This module is quite similar to the Master USART module. Its configuration is set to be the same as the second module of Master's USART configuration (Baud Rate = 9600, Data Bits = 8, Parity = No Parity, Stop Bits = 1). All slaves' TX and RX pins are linked to formed one TX and one RX line. This two nodes TX and RX is connected to the master's RX and TX line respectively. This connection is clearly shown in figure 5.3. This configuration requires all the slaves' TX lines to be set to high impedance if not in use, and at any moment only one TX line is allowed to be active to prevent data collision.

5.3.3 ADC module

The build in ADC module is a high speed 200Ksps 10-bit resolution. It offers 9 channels with simultaneous two samplings and auto conversion capability. The ADC has nine registers for configuration, control, and result buffer.

The registers are as follows:

Registers	Address Location
• A/D Result High Register (ADRESH)	FC4h
• A/D Result Low Register (ADRESL)	FC3h
• A/D Control Register 0 (ADCON0)	FC2h
• A/D Control Register 1 (ADCON1)	FC1h
• A/D Control Register 2 (ADCON2)	FC0h
• A/D Control Register 3 (ADCON3)	F9Ah
• A/D Channel Select Register (ADCHS)	F99h
• Analog I/O Select Register 0 (ANSEL0)	FB8h
• Analog I/O Select Register 1 (ANSEL1)	FB9h

Table 5.1: Registers and the address associated with the ADC module

The first three to four ADC channels are used for position feedbacks and the remaining channels for tactile sensing.

5.3.4 Motion Feedback module (Quadrature Encoder Interface - QEI)

This module is part of control component for forth DOF thumb's position control. Position feed back on this segment is obtained from encoder that is located at one end of the motor shaft. This encoder produces two square wave signals. The QEI module in the PIC microcontroller converts the two signals into a value representing position of the motor shaft. An example of the two square wave signals QEA and QEB is shown below:

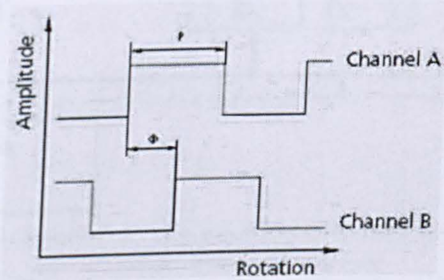


Figure 5.2: Signals from Encoder

This combination of continuous square wave signals contains information about speed and position of the motor shaft. The PIC18F4431 has the capability of monitoring this sequence. The internal QEI logic detects the leading edge of the QEA and the QEB phase, generates a count pulse and updates the position counter. The counter acts as an integrator for tracking distance traveled.

5.4 Communication

The communication system is a vital element for a distributed control system. Here the RS232 protocol is applied on the embedded USART module. The RS232 protocol is a commonly used communication method for microcontroller-computer interface. This protocol is easy to use, and most microcontrollers nowadays support this type of communication. The connections of RS232 protocol designed to support the distributed control of the prosthetic hand is shown below.

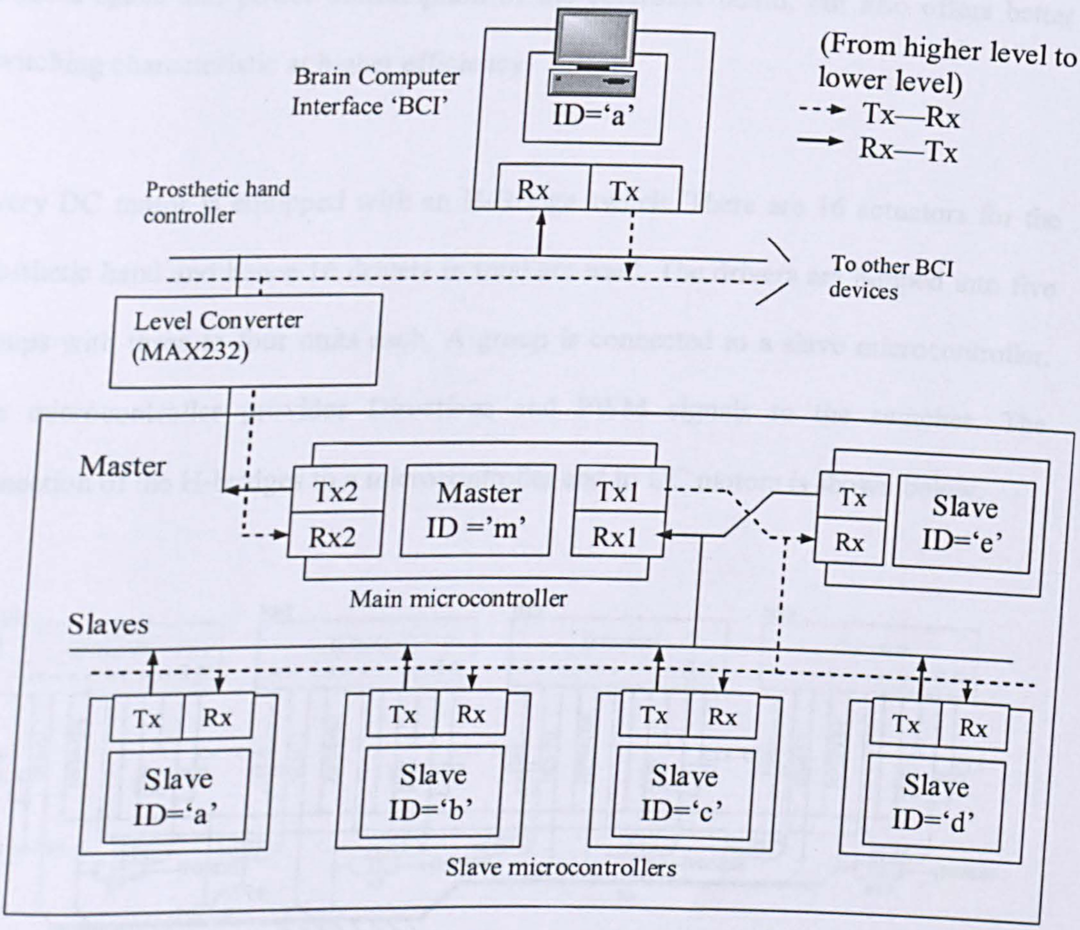


Figure 5.3: Communication of the prosthetic hand controller

Level converter MAX232 is used in this communication system to provide a proper voltage level for microcontroller-computer communication. The level converter is not used for master-slave communication since the microcontrollers are placed close to each other.

5.5 DC motor driver

H-bridge switches are used to drive the actuators with PWM power control. The integrated H-Bridge switch from National semiconductor LMD18200 is selected. This chip was built using multi technology processes that combine bipolar and CMOS control circuitry with DMOS power devices on the same monolithic structure [42]. This compact H-bridge switch eliminates the need of conventional method by connecting four separate switches to perform a similar function. The use of this chip is not only reduces space and power consumption of the controller board, but also offers better switching characteristic at higher efficiency.

Every DC motor is equipped with an H-Bridge switch. There are 16 actuators for the prosthetic hand and hence 16 drivers in total are used. The drivers are lumped into five groups with three to four units each. A group is connected to a slave microcontroller. The microcontroller provides Directions and PWM signals to the switches. The connection of the H-bridges to a microcontroller and to DC motors is shown below.

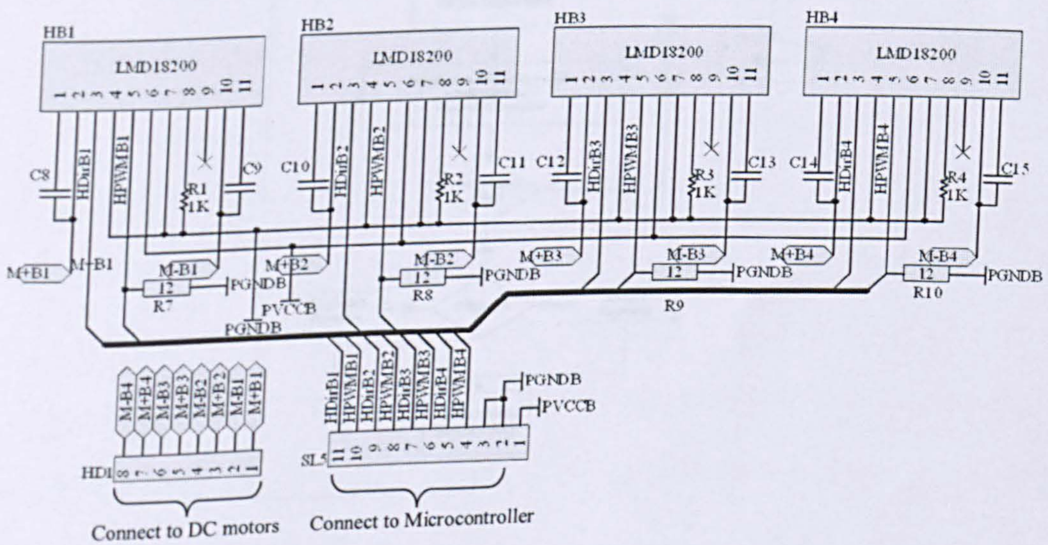


Figure 5.4: Connections of H-bridge switches for actuators of a finger

Detail interconnections of the H-bridge switch is clearly shown on Appendix1A

5.6 Firmware and computer interface

5.6.1 The Firmware

Firmware is developed using MPLAB IDE that is provided for free by the PIC microchip manufacturer. Basic and PIC microprocessor Assembly languages are used in developing the firmware. The codes written are translated into machine codes with the aid of the PICBASIC PRO Assembler. The machine codes are then burned into the flash program memory of the PIC microcontroller. The simplified block diagrams of firmware algorithms for the master and slaves microcontrollers are shown on Figure 5.5 and Figure 5.6.

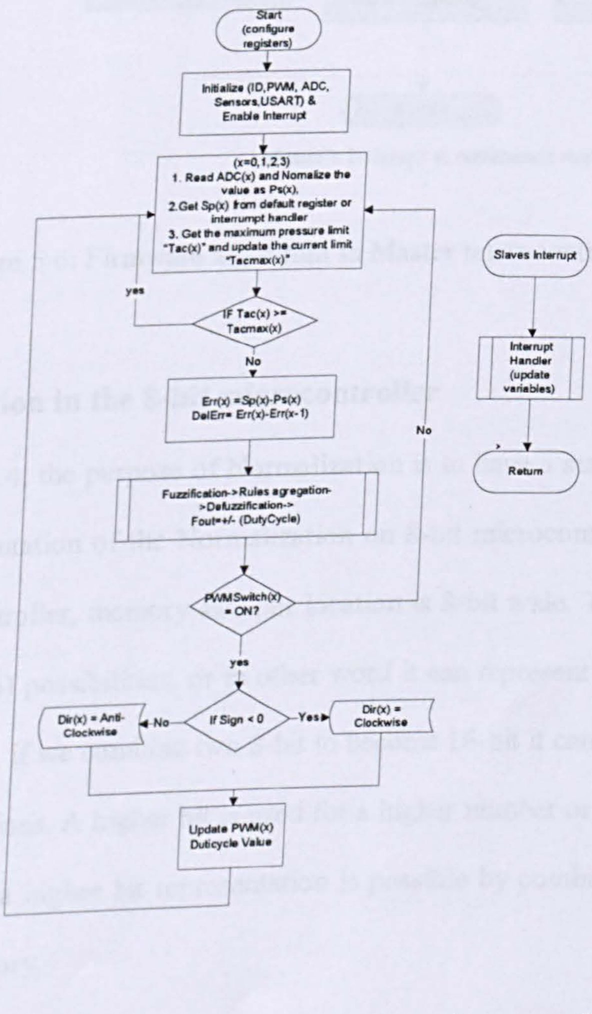


Figure 5.5: Firmware algorithm in a Slave microcontroller

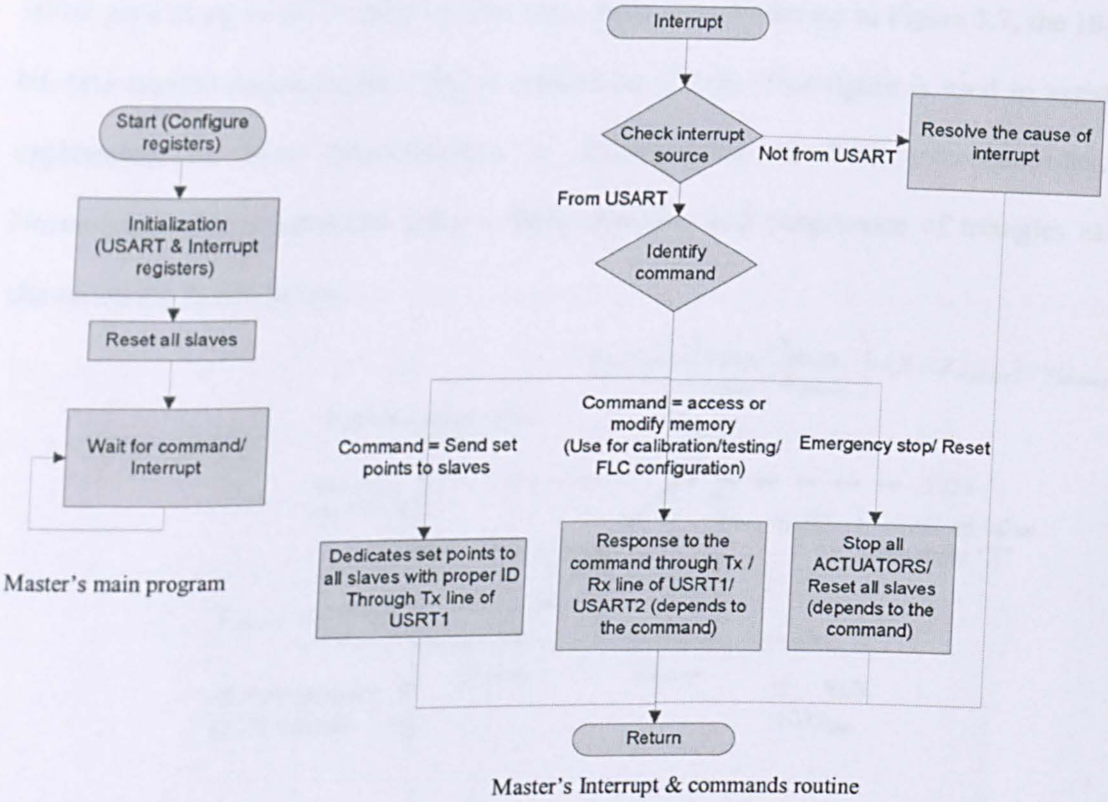


Figure 5.6: Firmware algorithm in Master microcontroller

5.6.2 Normalization in the 8-bit microcontroller

As mentioned in 4.2.4, the purpose of Normalization is to have a standard input for the FLC. Here, implimentation of the Normalization on 8-bit microcontroller is discussed. In an 8-bit microcontroller, memory size per location is 8-bit wide. This 8-bit wide can cover up to ($2^8 = 256$) possibilities, or in other word it can represent integers from 0 to 255. In the same way, if we combine two 8-bit to become 16-bit it can cover up to ($2^{16} = 65536$) possible locations. A higher bit is used for a higher number or higher resolution. Under 8-bit memory, a higher bit representation is possible by combining two or more units of the 8-bit memory.

In the FLC system, Normalization receives input from 10-bit wide ADC register. This 10-bit give us up to ($2^{10} = 1024$) to represent input data. Referring to Figure 5.7, the 10-bit data representation of the ADC is applied on X axis. This figure is used to assist explanation on how Normalization is accomplished on 8-bit microcontroller. Normalization is constructed using a linear function and congruence of triangles as shown on the figure below.

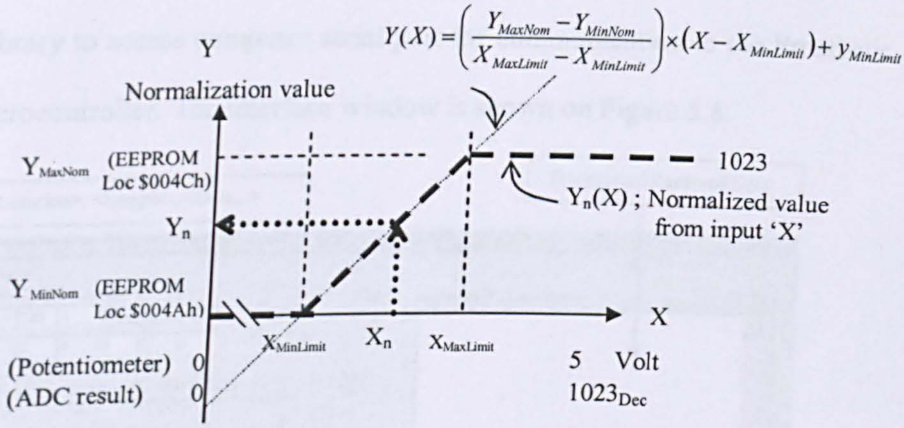


Figure 5.7: Normalization of feedback

Rotation of a finger segment is limited to a certain angle similar to the human hand. Hence, we have minimum and maximum values that are allowed for the ADC which are $X_{MaxLimit}$ and $X_{MinLimit}$. On FLC side, we have a range of numbers acceptable as its input. The range is remarked as Y_{MaxNom} and Y_{MinNom}. The reason why only the values within this range are permitted is that to prevent error on the FLC system. These values $X_{MaxLimit}$, $X_{MinLimit}$, Y_{MaxNom}, and Y_{MinNom} are stored in EEPROM of the microcontroller. On the Figure 5.7, the "tick dark dashed line" labeled as $Y_n(X)$ is a Normalization function. This function is shown below;

$$Y_n(X) = \begin{cases} Y_{MinNom} \ldots\ldots\ldots\ldots\ldots\ldots\ldots ; X \le X_{MinLimit} \\ m \times (X - X_{MinLimit}) + Y_{MinLimit} \ldots\ldots ; X_{MinLimit} \le X \ge X_{MaxLimit} \\ Y_{MaxNom} \ldots\ldots\ldots\ldots\ldots\ldots\ldots ; X \ge X_{MaxLimit} \end{cases}$$ (the linear function)

Where; $m = \frac{Y_{MaxNom} - Y_{MinNom}}{X_{MaxLimit} - X_{MinLimit}}$

5.6.3 The computer interface

A Windows based computer interface was developed for easy customization of the prosthetic hand controller and also for validating the Distributed Fuzzy Logic Control algorithms.

The program was written and compiled using Visual basic. This program uses a RS232 dynamic link library to access computer serial port for communication to the Prosthetic hand master microcontroller. The interface window is shown on Figure 5.8.

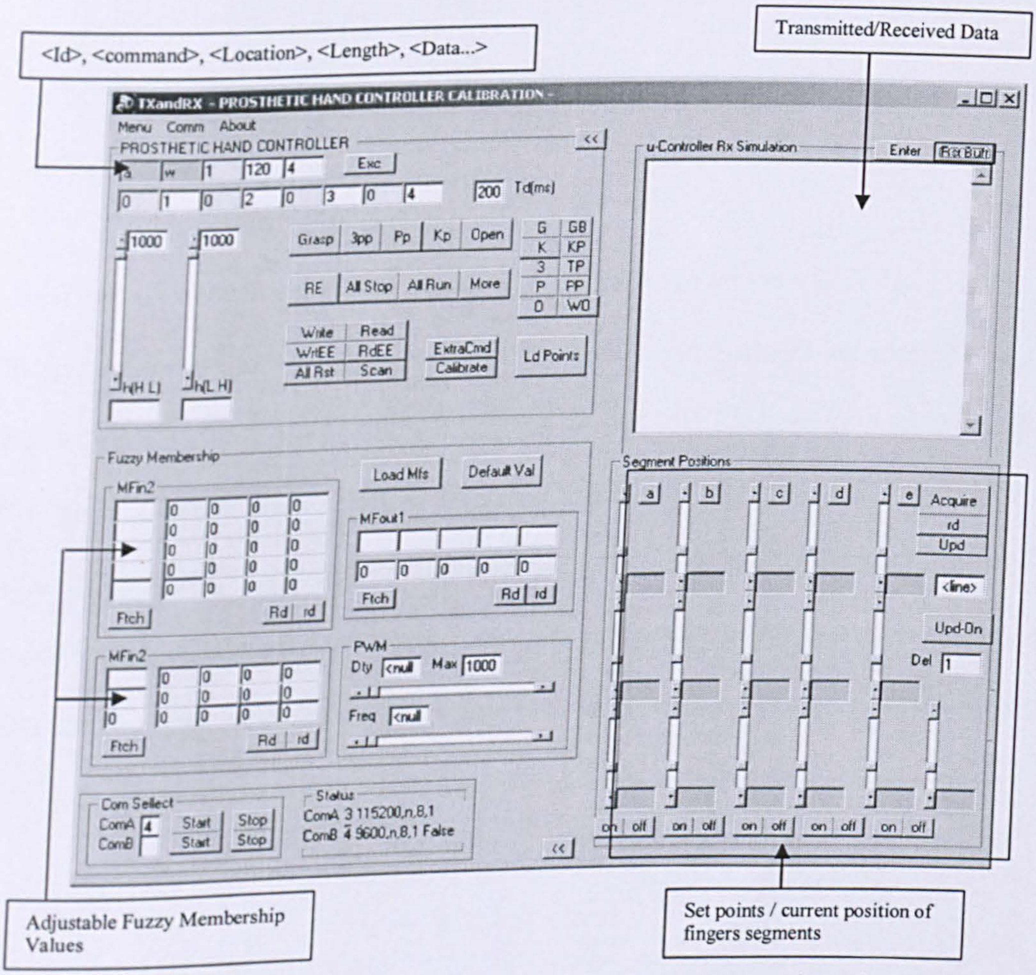


Figure 5.8: The Windows front-end microcomputer interface

Commands used to control the prosthetic hand controller are listed in Appendix-2A. Any computer or device that has RS232 communication can use the commands. This low cost interfacing using serial RS232 protocol is implemented to ensure that the FLC system is general enough to interface to any computer or devices that may run the BCI system.

RESULTS & DISCUSIONS

6.0 Introduction

Tests were conducted to validate the control system for the prosthetic hand.

The tests are as follows:

1. The prosthetic hand controller with the BCI
2. Finger responses under different set points.
3. Finger response under different load.
4. Finger Trajectory
5. Finger-Object interaction

The first test shows the overall integration of the hand and the BCI. Using a PC as the Brain Computer Interface platform, the Prosthetic hand is controlled through the RS232 communication line. Five hand tasks were performed. The second test examines the FLC in positioning segments of a finger. In the third experiment, the system was tested again to see how it responses under different load conditions. The finger trajectory was tested in experiment number four. In the last experiment, which is to test Finger-Object interaction a finger was allowed to move with its segments sequentially to wrap around an object. All of the tests are discussed in the following sub-topics below.

6.1 Test of complete system (Integration of prosthetic hand controller with the BCI)

The complete system, (consisting of a BCI processing unit, the prosthetic hand controller and the multi-fingered prosthetic hand) was tested to perform the five desired hand tasks. The tasks are grasping, pulp-to-pulp pinch, tripod-pinch, key-pinch and reset position. For every hand task execution it involves simultaneous positioning control over the whole fingers segments. Motion of every segment is limited either by its individual final set point or by the pressure limit sensed at the fingertips. In this experiment the control components hardware and software (which include the RS232 communication, FLC, the system handler, and the distributed control) were tested. Photos of the hand perform the tasks are shown below.

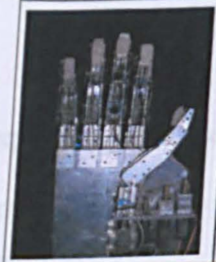
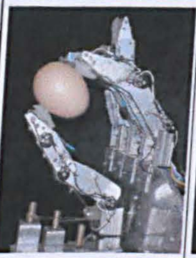
	Reset Position	Grab	Tripod Pinch	Pulp-to-pulp	Key pinch
Tasks →					
Set points (a1,a2,a3,a4) (b1,b2,b3,b4) (c1,c2,c3,c4) (d1,d2,d3,d4) (e1,e2,e3,e4)	100, 100, 100, - 100, 100, 100, - 100, 100, 100, - 100, 100, 100, - 100, 100, 100, 1	400, 400, 400, - 400, 400, 400, - 400, 400, 400, - 400, 400, 400, - 270, 318, 0, 50	100, 100, 100, - 100, 100, 100, - 160, 160, 370, - 160, 160, 320, - 50, 117, 31, 145	100, 100, 100, - 100, 100, 100, - 100, 145, 100, - 160, 120, 340, - 60, 159, 31, 115	400, 447, 512, - 447, 485, 458, - 450, 493, 530, - 252, 441, 310, - 152, 297, 130, 50
Listed are the maximum displacement angles for the manipulator to move, if there is an obstacle and the pressure exceed a certain threshold value sensed at fingertips the finger will stop					

Figure 6.1: The five hand tasks

This results show that the hand controller capable to communicate with the BCI system and performs the designated hand tasks.

6.2 Finger responses under different set points

A finger was tested to see how the FLC perform. Set points were sent from a computer using serial communication port through the master microcontroller. In this case only one microcontroller was active. The finger motion is recorded by continuously taking displacement angles on each joint. Graph that shows the segments responses is plotted as shown below.

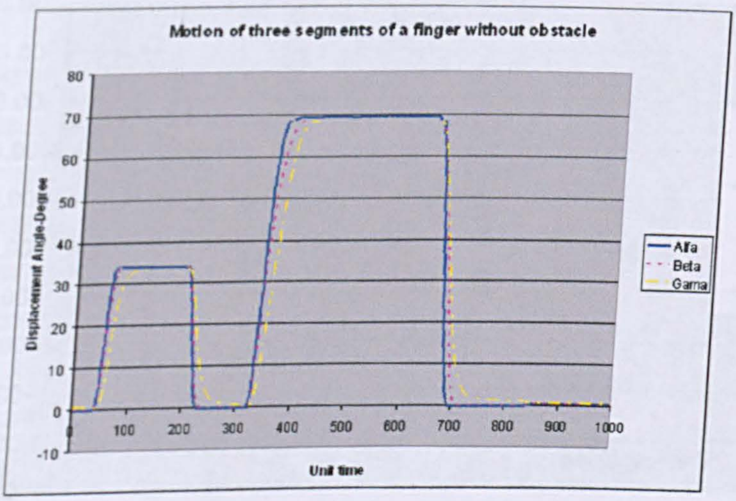


Figure 6.2: Segments responses

In this experiment the fuzzy logic controller was tested to simultaneously move the finger segments from one position to another position. The segments were able to move close to the specified set point as shown on the graph.

6.3 Responses of a finger under different loads

To show the capability of the finger to operate under different loads the following test has been conducted. A finger was let to run repeatedly under same set points but under different loads (from 0 to 400g). The responses for position were recorded.

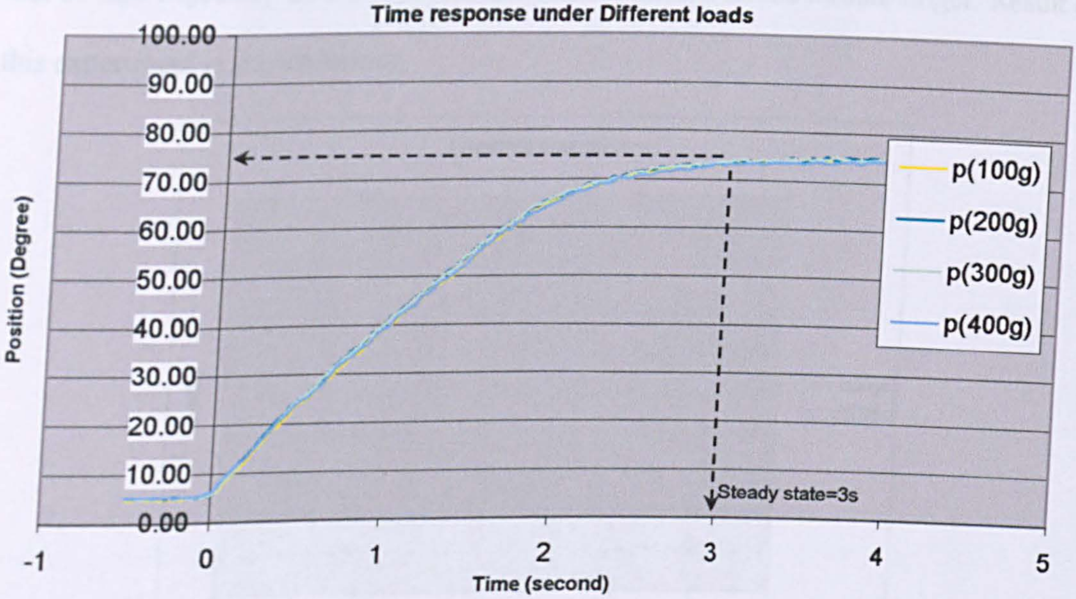


Figure 6.3: Finger responses under different loads (0-400g)

The responses of position are consistent although the load was changed. The maximum applied load was limited by the maximum motor torque allowed by the motor. This load variation test indicates that a load change on finger segments does not much affect its motion response and trajectory. The main factor that supports this stability is the capability of the motor driver to supply high current and also due to high torque handling capability possessed by the actuators. Load variation in the prosthetic hand system is mainly caused by inconsistent friction that is due to surface contact between the segment plates and also friction and elastic behavior of the harness cables. Time taken to reach the set point is about 3 second; this time is acceptable for a hand motion.

6.4 Finger trajectory

The advantage of electric controlled prosthetic fingers is its flexibility to give trajectory suitable for hand motion. Here Gou's standard trajectory is used as a comparison model. The fuzzy logic controller was programmed to follow the trajectory similar to that of Gou trajectory model [1]. The test was conducted on the middle finger. Result of this experiment is shown below.

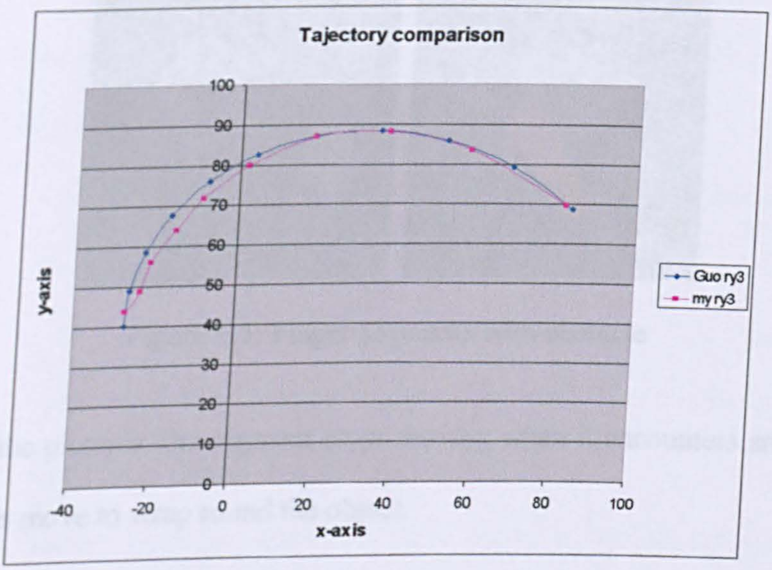


Figure 6.4: Trajectory of the middle finger compared to the Gou's trajectory model

The trajectory obtained from the prosthetic finger is close to the Gou's trajectory model.

6.5 Finger-Object interaction

Here the same procedure as the second experiment was applied, however the set points were set to maximum. An object is placed in the path of the segments.

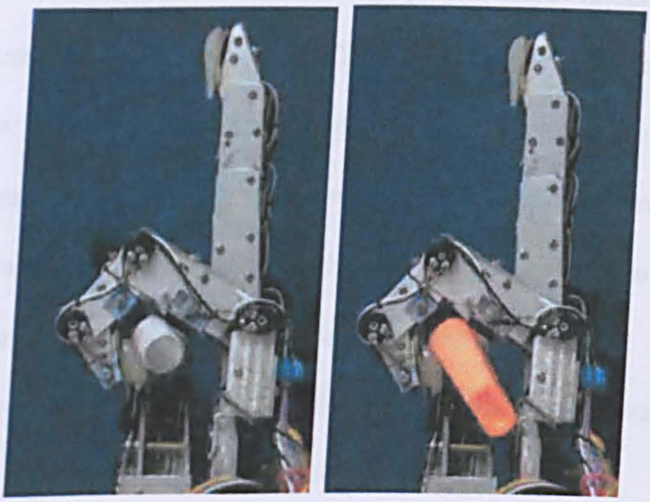


Figure 6.5: Finger segments with obstacle

As shown in the pictures, the segment stops moving when it encounters an object. The other segments move to wrap round the object.

CONCLUSIONS

1. A FLC with distributed monitoring system was developed for a BCI system to control a prosthetic hand. The finger movements are programmed to follow given set points and stopped whenever an obstacle is encountered and the pressure of the tactile sensor exceeds a specified limit. This allows the fingers and thumb to wrap round an object without crushing it.
2. The control system consists of the following components:
   - Five unit slave microcontrollers programmed with fuzzy logic to control the four fingers and the thumb.
   - One unit master microcontroller to coordinate all the slave microcontrollers and to interface to the BCI. The master microcontroller acts as the front end to the BCI system.
   - Sixteen H-bridge motor drivers to drive DC motors using PWM power control method.
3. The following tests were carried out to validate the control system for the prosthetic hand.

   They are:
   - The prosthetic hand controller with the BCI
   - Finger response under different set points
   - Finger response under different loads
   - Finger trajectory

4. The prosthetic hand control board was implemented using the embedded microcontrollers. This method requires small spaces and also consumes relatively low power since much of electronic components are properly connected and integrated in compact chips.
5. The use of distributed control method over the FLC microcontrollers for the prosthetic hand has enabled easy integration of the microcontrollers to other component such as the BCI system.

Suggestion for future research

1. This hand controller can be extended to control wrist in the future by just adding another unit of similar slave microcontroller to the control network.
2. The slave microcontrollers and the DC motor drivers should be rearranged to an optimum position such way feedbacks and motor driver signal possess least effect of interference. Besides, all the signal lines must be properly protected from external noise.
3. Self calibration is possible by integrating more sensors together with minor modification to the firmware. This self calibration is necessary to reduce errors and make the system more reliable.
4. Implementation of smaller unit control hardware is possible by using smaller packages of microcontrollers and H-bridge switches. However, this implementation requires special and relatively expensive tools.

References:

[1] C. Guger, A. Schlögl, C. Neuper, D. Walterspacher, T. Strein, and G. Pfurtscheller, "Rapid Prototyping of an EEG-based brain-computer interface (BCI)," IEEE Trans. Rehab. Eng., Vol. 9, 2001, 49-58.

[2] Ramoser, H.; Muller-Gerking, J.; Pfurtscheller, G., "Optimal spatial filtering of single trial EEG during imagined hand movement" Rehab. Eng., IEEE Trans., Vol. 8, Issue 4, Dec 2000, 441-446.

[3] Peter K., " The Intelligent Hand", IEEE Review, September 2000, 31-35.

[4] Iberall, T., Sukhatme, G., Beattie, D., Bekey, G., "Control Philosophy and Simulation of a Robotic Hand as a Model for Prosthetic Hands", Intelligent Robots and Sys. IROS '93. Proc. IEEE/RSJ Int. Conf., Jul 1993, 824-831.

[5] Han-Pang Huang, Ya-Fu Wei, "Control of Dextrous Hand Master with Force Feedback", Robotics and Automation, Proc. IEEE Int. Conf., Vol.1, May 1998. 687-692.

[6] Buss, M., Hashimoto, H., "Intelligent Control for Human-Machine System", Mechatronics, IEEE/ASME Trans., March 1996, 50-55.

[7] L. Biagiotti, "Advance Robotic Hand: Design and Control Aspects", Ph.D. Thesis, University of Bologna, Italy, year 2002.

[8] Spano, S.M., Bourbakis, N., "Implementation of a Multi-fingered Robotic Hand Using a Method of Fuzzy Blocks", Intelligence and Sys., IEEE Int. Joint Symposia, May 1998. 262-267.

[9] Kumbla, K.K., and Jamshidi, M., "Control of Robotic Manipulator Using Fuzzy logic", Fuzzy Sys. IEEE World Cong. on Comp. Intelligence. , 1994, 518-523.

[10] Iberall, T., Sukhatme, G., Beattie, D., Bekey, G., "Control Philosophy and Simulation of a Robotic Hand as a model for prosthetic Hand", Intelligent Robots and Sys. IROS '93. Proc. of the (1993) IEEE/RSJ Int. Conf., Vol. 2, July 1993. 824-831.

[11] Loucks, C., Johnson, V., Boissiere, P., Starr, G., Steele, J., "Modeling and Control of the Stanford/JPL Hand", IEEE, 1987, 573-578.

[12] Biagiotti, L., Lotti, F., Melchiorri, C., and Vassura G., "Mechatronic Design of Innovative Fingers for Anthropomorphic Robotic Hands", DEIS – DIEM, University of Bologna, 2002

[13] Biagiotti, L., Lotti, F., Melchiorri, C., and Vassura, G., "How Far Is the Human Hand? A Review on Anthropormorphic Robotic End-effectors", University of Bologna, 2002

[14] Shadow Robot Company, "Design of a Dextrous Hand for advanced CLAWAR Applications", (www.thirdwave.de/tech/actuators/mckibben**hand**.pdf)

[15] Namima, T., Hashimoto, H., "Dextrous Manipulation Using a 24 DOF Robotic Hand", Intelligent Robots and Sys '94. IEEE/RSJ/GI Int. Conf., Vol. 2, Sep 1994. 1446-1453.

[16] Li-Ren Lin and Han-Pand Huang, "Integrating Fuzzy Control of the Dexterous National Taiwan University (NTU) Hand", Mechatronics, IEEE/ASME Transactions, Vol. 1, No. 3, Sept 1996, 216-229.

[17] J. Lin, "Hierarchical fuzzy logic controller for a flexible link robot arm performing constrained motion tasks", IEE Proc. Control Theory App, Vol. 150, No. 4, July 2003, 355-364.

[18] Iversen, E., Sears, H.H., Jacobsen, S.C., "Artificial Arms Evolve from Robots, or Vice Versa?", Control System Meghazine, IEEE Vol.25, Feb 2005, 16-18.

[19] Amit V. Upasani, Chetan Kapoor, Delbert Tesar, "Survey Of Available Sensor Technology For Robotic Hands", Proc. of DETC99 ASME Design Engineering Technical Conferences Sep., Las Vegas, Nevada., 1999, 1-7.

[20] "Shadow Hand's Documents", http://www.shadow.org.uk/index.shtml, 2005

[21] Casalino, G., Giorgi, F., Turetta, A., Caffaz, A., "Embedded FPGA-based Control of a Multi-fingered Robotic Hand", IEEE Int. Conf., Vol. 2, Sept 2003, 2786 – 2791.

[22] Michael Charles MacHenry, "Slice-based Path Planning", Ph.D thesis,University of Southern California, 1998.

[23] Yingjie Yin, Zhiwei Luo, Svinin, M., Hosoe, S., "Hybrid Control of Multi-fingered Robot Hand for Dexterous Manipulation", Sys, Man and Cyb, IEEE Int. Conf., Vol. 4, Oct 2003, 3639-3644.

[24] Tipsuwan, Y., Mo-Yuen Chow, "Fuzzy Logic Microcontroller Implementation for DC Motor Speed Control", Industrial Electronics Society, IECON '99. 25th Annual Conf. of the IEEE, Vol. 3, Dec 1999, 1271-1276.

[25] Reza L., and Dongyoon H., "Fuzzy Logic Based Compensation of Friction in Low Speed Motion Control" Pro. Int. IEEE Sym. on Intelligent Control. Mexico City, 2001, 337-342.

[26] Paul-l-Hai L., and John C. (1994), "Comparison on Fuzzy Logic and PID Controllers For A DC Motor Position Controller", Conf. Record of the IEEE, Oct. 1994. 1930–1935.

[27] Zadeh L. A., "Fuzzy sets", Informant Control, Vol. 8, 1965, 338-353.

[28] Zadeh L. A., "Outline of a new approach to the analysis complex system and decision processes", IEEE Trans. Syst, Man Cybern, Vol. SMC-3, 1973, 28-244.

[29] Haessig D. A. and Friedland B., "Modelling and Simulation of friction", *ASME Journal of Dynamic Systems, Measurement and Control,* Vol. 113, 1991, 354-362.

[30] Armstrong-Helouvry B., "Control of Machines with Friction", Norwell MA: Kluwer Academic Publishers, 1991.

[31] Parspour N. and Hanitsch R., "Fuzzy Controlled D.C Motor for Medical Application", Pro. 20th Int. IECON '94 Conf., Vol. 2, 1994, 1310-1314

[32] Iberall T., Sukhatme G., Beattie D., and Bekey, G.A., 'Control Philosophy and Simulation of a Robotic Hand as a Model for Prosthetic Hands', Pro. Int. IEEE/RSJ Conf. on Intelligent Robots and System in Yokohama, Japan, 1994, 824-831

[33] Massa B., Rocella M., Carrozza M. C., and Dario P., 'Design and Development of an Underactuared Prosthetic Hand', Pro. Int. IEEE Conf. on Robotic and Automation. in Washington DC, 2002, 3374-3379

[34] Chow M., Tram H., 'Application of fuzzy logic controller for spatial load forecasting', IEEE Trans. Power System., Vol. 12, no. 3, 1997, 136-1366.

[35] Chow M., Zhu J., Tram H., 'Application of fuzzy multi-objective decision making in spatial load forecasting', IEEE Trans. Power System., Vol. 13, no. 3, 1998, 1185-1190

[36] Hess D. P., Soom A., 'Friction at lubricated line contact operating at oscillating sliding velocities', Tribology, Vol. 112, no. 1, 1990, 147-152.

[37] Sung-Woo Kim and Ju-Jang Lee, "Resolved Motion Rate Control of Redundant Robots using Fuzzy Logic", IEEE Trans., 1993

[38] Jeff C. Becker and Nitis V. Thakor, "A Study of the Range of Motion of Human Fingers with Application to Anthropomorphic Design", IEEE Tran. on Biomedical Engineering, Vol. 35. No. 2, February 1988.

[39] G. Guo, J. Zhang, W.A. Gruver, "Optimal design of a six bar linkage with one degree of freedom for an anthropomorphic three-jointed finger mechanism", Proc. Inst. Mech. Eng., Vol. 207, 1993, 185-190.

[40] Francisco J. Valero-Cuevas, "Predictive Modulation of Muscle Coordination Pattern Magnitude Scales Fingertip Force Magnitude Over the Voluntary Range", J Neurophysiol, Vol.83, Mar 2000, 1469 - 1479.

[41] Per Jenmalm, Seth Dahlstedt, And Roland S. Johansson, "Visual and Tactile Information About Object-Curvature Control Fingertip Forces and Grasp Kinematics in Human Dexterous Manipulation", J Neurophysiol, Vol. 84, Dec 2000, 2984 – 2997.

[42] "LMD18200 Datasheet", www.national.com, 2005

[43] "Faulhaber DC Motor and Gear Head datasheet", www.faulhaber.com, 2005

[44] "Faulhaber Motor selection guide", www.faulhaber.com, 2005

[45] Muhammad H. Rashid, "Power Electronics Circuit, Devices, and Applications", 2nd Edition, Prentice Hall, Eaglewood Cliffs, 1993, 172-175.

[46] Guo-Qing Wang, Qi-Xian Zhang, "Fuzzy-Reasoning-Based Control for the Dexterous Hand", Ind. Tech., Proc. IEEE Int. Conf., Dec 1994, 616-620.

[47] Kim, S.-W., Lee, J.-J., Sugisaka, M., "Inverse Kinematics Solution Based on Fuzzy Logic for Redundant Manipulators", Intelligent Robots and Sys. IROS '93. Proc. IEEE/RSJ Int. Conf., Vol.2, July 1993, 904-910.

[48] Ramos, M.C., Jr., Koivo, A.J., "Fuzzy Logic-Based Optimization for Redundant Manipulators", Fuzzy Sys., IEEE Trans. Vol.10, Aug 2002, 498-509.

[49] Bart Kosko, "Fuzzy Engineering", Prentice Hall, 1997

[50] Venkataraman Raja Gopalan, "A Region-Sensitive Fuzzy Logic Controller for pH", Dissertation (M.Eng.), National University of Singapore, 1999.

[51] Pedrycz, W. "Fuzzy Control and Fuzzy System", second extended edition, Research Studies Press LTD, Taunton, Somerset, England. 1993

[52] N. H. Muhammad, "Fabrication and testing of prosthetic hand", Academic exercise report (B. Eng), Department of Biomedical Engineering, University of Malaya, 2005

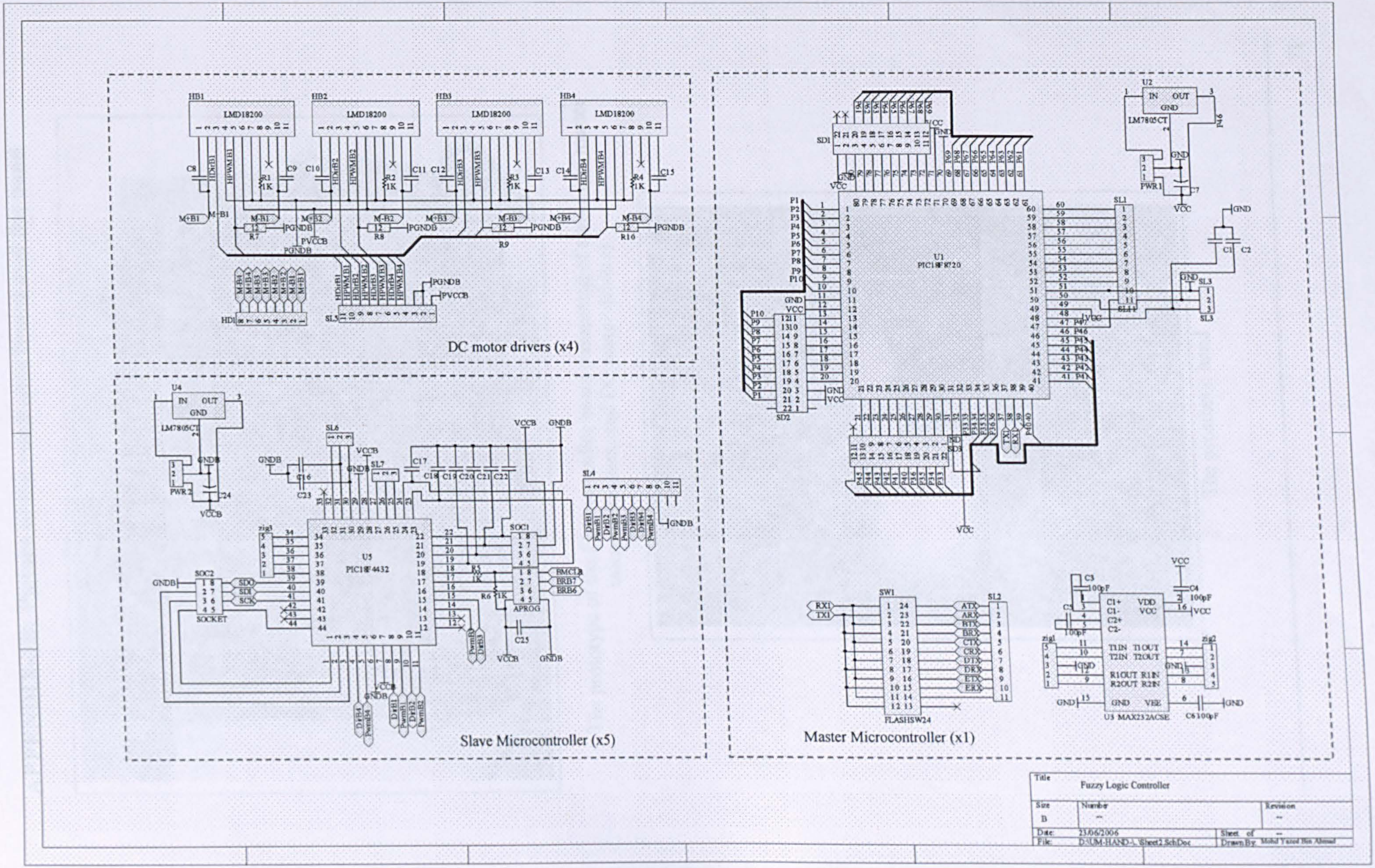
[53] A.M. Yazed, S.Y. Goh, W.P. Hew, "Control of a Three Degree of Freedom Prosthetic Finger Using Fuzzy Logic", Proc. of the 3rd Technical Postgraduate Symposium 2004.

[54] A.M. Yazed, S.Y. Goh, W.H. Ping, "Control of Prosthetic Hand using Fuzzy Logic", IFMBE Proc., Kuala Lumpur International Conference on Biomedical Engineering, Vol. 7, 2004, 163-165.

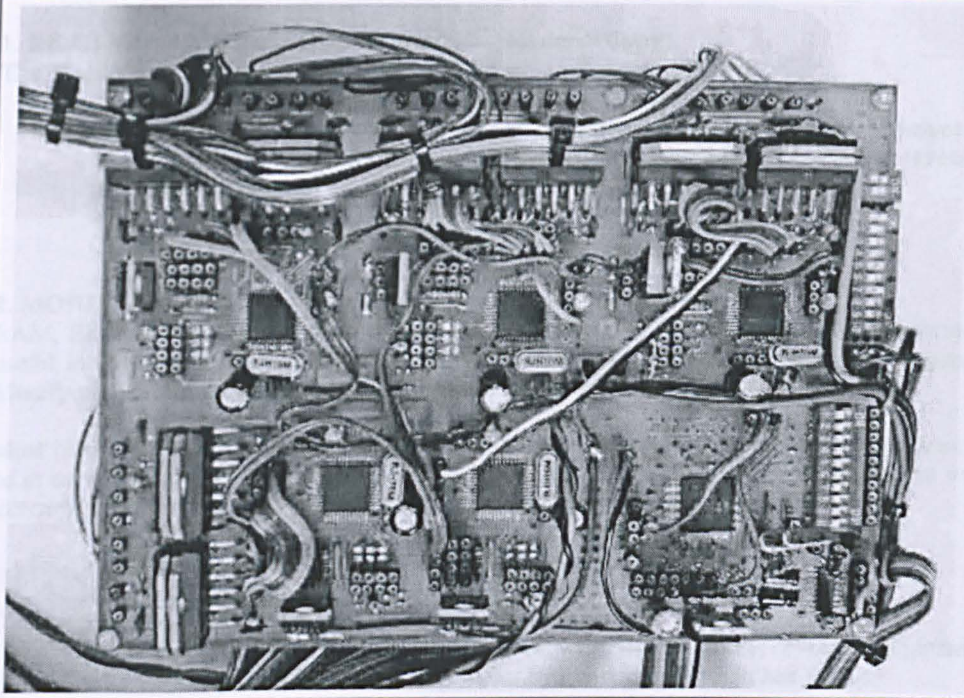
[55] S.Y. Goh, S.C. Ng, Y.M. Phang, X.Y. Yong, E. Lim, A.M. Yazed and Y. Shuhaida, "A brain-computer interface for control of a prosthetic hand", Journal of Science and Technology in the Tropics, Vol. 1, 2005, 35-41.

[56] F.W. Lai, "Development of a multifingered prosthetic hand mechanism", Dissertation report (M. Eng.), Faculty of Engineering, University of Malaya, 1998

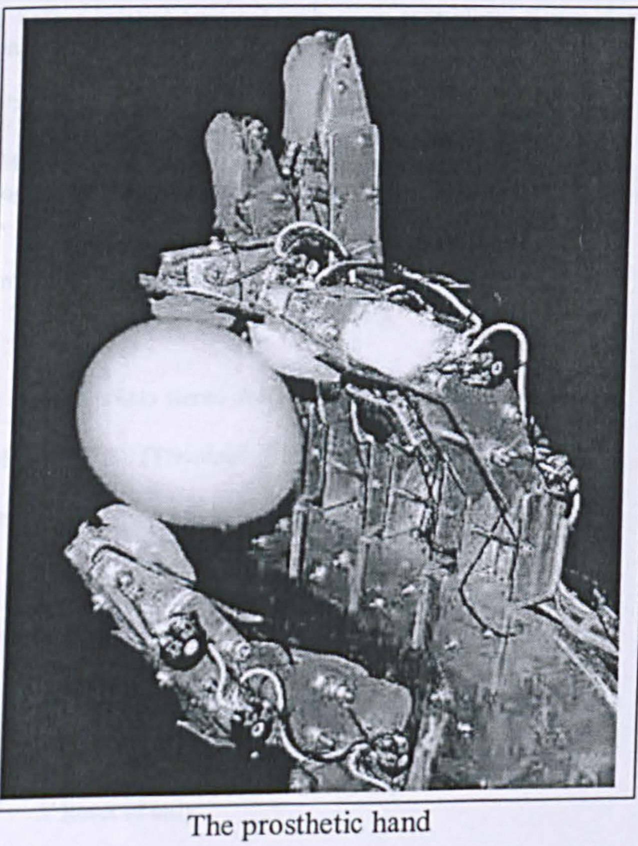
APPENDIX-1A: CIRCUIT DIAGRAM



APPENDIX-1B: Picture of the controller board and the hand



The prototype of the FLC controller board (consisting of slaves and master microcontrollers, and DC motor drivers)



The prosthetic hand

APPENDIX-2A: COMMANDS

1. READ TACTILE SENSORS (Terminal ->Master->Slaves)

The Tactile sensors can be read simultaneously using main controller

>> *HAD* // H-device, AD-Analog to digital

Master reads all current tactile sensors and sends the corresponding decimal values representing pressures to the requested terminal. This sequence is repeated until the system is reset / off. Reset is either hardware reset or by a command through USART communication.

>>*HRT* //H-device, RT-reset command

Command to reset all controllers that is also terminated HAD execution.

2. MODIFY MEMORY (Terminal -> Master->Slaves/ Slaves)

RAM, EEPROM, and registers on Master and Slave controller are accessible and can be modified. It is useful in minimizing controller code size since similar commands are used to modify system memory. Modifying memory locations would reflect variable values.

Most important variables like PWM frequency, Set points, and Fuzzy membership functions are fixed to be at certain locations. And to access or modify variables, a similar command is used but with different memory location. This implementation reduces determination loops and improves speed.

>>*HC*<Id><Command><Start><Address><size><Termination>
Id = {*'a', 'b', 'c', 'd', 'e', 'm'*}
Start = {'s'}
Command = {*'U'*-ReadEE; *'V'*-WriteEE; *'w'*-write flash memory; *'r'*-read flash memory}
Address = 2bytes memory location Higher byte and lower byte in hex format
Size = number of bytes to be read or write {*1, 2, 3...14*}
Termination = {*'e'*}

3. BYPASS MASTER (Terminal -> Master->Slaves/ Slaves)

This command bypasses the master controller that enables direct communication from the upper controller (controller before master) terminal with slave controllers. Every slave controller is directly accessible as its commands through master controller by using

>>*HC<'ID'><Slave Command>*

4. PREPROGRAMMED MOTIONS (Terminal -> Master->Slaves)

There are five preprogrammed commands use to perform tasks; key-pinch, tripod-pinch, pulp-to-pulp, hand open, and grasp.

>>*HP*<Command><Termination>
Command = {*"GB", "PP", "TP", "KP", "WO"*}
Termination = {*'Z'*}

Master controller will read set points stored at EEPROM for the particular task.

5. STOP, RUN, RESET (Terminal -> Master->Slaves)

Stop actuator(s) by setting PWM value(s) to be equal to zero.

```
>>HCOF          // stop all
>>HC<Id>F       //stop only id
```

Use current PWM value(s)

```
>>HCON          // on/run all
>>HC<Id>N       //on/run only id
```

Reset

```
>>HCRT    // Reset master then reset all slaves
>>HCRE    // Reset all slaves only
>>HC<Id>R // Reset id only
```

Id = {*'a', 'b', 'c', 'd', 'e'*}

Enable status for position values - 73 -
Set variable Pen.2 = 1 at location Program memory \$143 / (1, 67) $_{Dec}$.
or set EEPROM loc \$0A0 / (0,160) $_{Dec}$

Enable Integral action
Set EEPROM loc \$90 / (0, 144) $_{Dec}$ <x x x x x x x x> → x=1 enable, x=0 disable
This would change value of variable "Iflg" after reset/reboot the microcontroller unit.

Enable tactile sensors as motion limiter
Set EEPROM loc \$76 / (0, 118) $_{Dec}$ <x, x, x> → x= 1 enable, x=0 disable
Internal variables would only change after reset/reboot the microcontroller unit

Important variables

Flash Memory (SLAVES)

Name, Type/Size, Base Location
SP word [5] \$016E 'Current Set Points (1st .., 2nd .., 3rd Set Points)
Er word [5] \$0180 'Current Errors (1st .., 2nd .., 3rd Segments)
DEr word [5] \$018A 'Current Error Rates (1st .., 2nd .., 3rd Error Rates)

EEPROM (SLAVES)

Error membership functions 'MF' (Listed are the trapezoidal point of the MF)
MF1in1 \$0000 –to- \$0007
MF2in1 \$0008 –to- \$000F
MF3in1 \$0010 –to- \$0019
MF4in1 \$0018 –to- \$001F
MF5in1 \$0020 –to- \$0029
Del Error membership functions
MF1in2 \$0028 –to- \$002F
MF2in2 \$0030 –to- \$0037
MF3in2 \$0038 –to- \$003F

EEPROM (MASTER)

Preprogrammed Set Points (8 bytes for each finger – 1st, 2nd, 3rd, 4th segment)

	Final Position Memory Location				
Finger	Grasp "GB"	P-Pulp "PP"	Tripod "TP"	Key-Pinch "KP"	Rest/Open "WO"
a	\$0	\$28	\$50	\$78	\$A0
b	\$8	\$30	\$58	\$80	\$A8
c	\$10	\$38	\$60	\$88	\$B0
d	\$18	\$40	\$68	\$90	\$B8
e	\$20	\$48	\$70	\$98	\$C0

APPENDIX-3A: Actuator's technical data (DC motor, and Gear head, Encoder)

FAULHABER

NEW

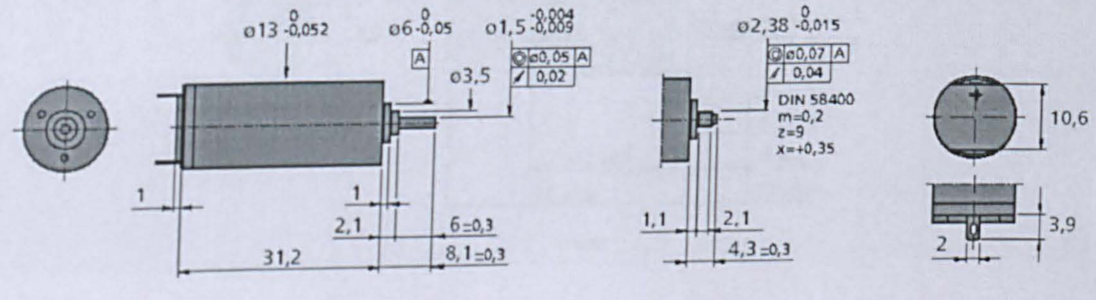
DC-Micromotors 3,2 mNm

Precious Metal Commutation

For combination with
Gearheads:
14/1, 15/3, 15/5
Encoders:
IE2

Series 1331 ... SR

		1331 T	006 SR	012 SR	024 SR	
1 Nominal voltage	U_N		6	12	24	Volt
2 Terminal resistance	R		2,83	13,7	52,9	Ω
3 Output power	$P_{2\,max.}$		3,11	2,57	2,66	W
4 Efficiency	$\eta_{max.}$		81	80	80	%
5 No-load speed	n_o		10 600	9 900	10 400	rpm
6 No-load current (with shaft ø 1,5 mm)	I_o		0,0220	0,0105	0,0055	A
7 Stall torque	M_H		11,20	9,90	9,76	mNm
8 Friction torque	M_R		0,12	0,12	0,12	mNm
9 Speed constant	k_n		1 790	835	439	rpm/V
10 Back-EMF constant	k_E		0,56	1,20	2,28	mV/rpm
11 Torque constant	k_M		5,35	11,4	21,8	mNm/A
12 Current constant	k_I		0,187	0,087	0,046	A/mNm
13 Slope of n-M curve	$\Delta n/\Delta M$		946	1 000	1 070	rpm/mNm
14 Rotor inductance	L		70	310	1 100	µH
15 Mechanical time constant	τ_m		7	7	7	ms
16 Rotor inertia	J		0,71	0,67	0,63	gcm²
17 Angular acceleration	$\alpha_{max.}$		160	150	160	·10³rad/s²
18 Thermal resistance	R_{th1} / R_{th2}	6 / 25				K/W
19 Thermal time constant	τ_{w1} / τ_{w2}	5 / 190				s
20 Operating temperature range:						
– motor		– 30 ... + 85 (optional – 55 ... + 125)				°C
– rotor, max. permissible		+125				°C
21 Shaft bearings		sintered bronze sleeves				
22 Shaft load max.:						
– with shaft diameter		1,5				mm
– radial at 3000 rpm (3 mm from bearing)		1,2				N
– axial at 3000 rpm		0,2				N
– axial at standstill		20				N
23 Shaft play:						
– radial	≤	0,03				mm
– axial	≤	0,2				mm
24 Housing material		steel, black coated				
25 Weight		19				g
26 Direction of rotation		clockwise, viewed from the front face				
Recommended values						
27 Speed up to	$n_{e\,max.}$		12 000	12 000	12 000	rpm
28 Torque up to	$M_{e\,max.}$		3,2	3,2	3,2	mNm
29 Current up to (thermal limits)	$I_{e\,max.}$		0,81	0,37	0,19	A



1331 T ... SR

1331 E ... SR
for Gearheads 15/...

FAULHABER

Planetary Gearheads

0,3 Nm

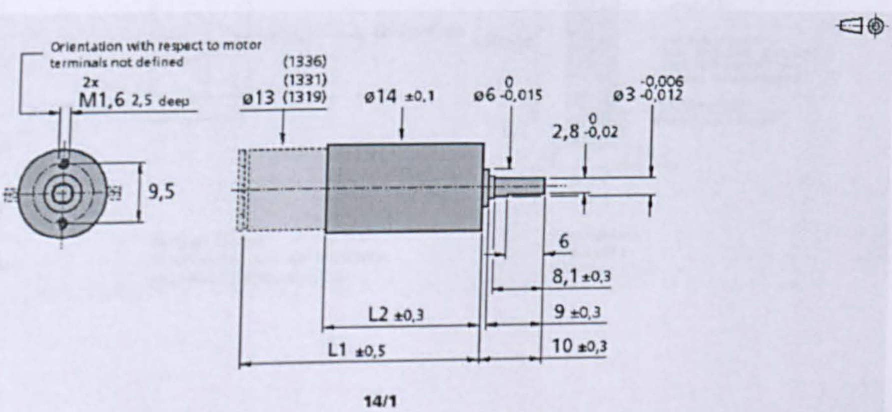
For combination with
DC-Micromotors:
1319, 1331, 1336

Series 14/1

	14/1
Housing material	metal
Geartrain material	all steel
Recommended max. input speed for:	
– continuous operation	5 000 rpm
Backlash, at no-load	≤ 1°
Bearings on output shaft	preloaded ball bearings
Shaft load, max.:	
– radial (6,5 mm from mounting face)	≤ 20 N
– axial	≤ 5 N
Shaft press fit force, max.	≤ 5 N
Shaft play (on bearing output):	
– radial	≤ 0,02 mm
– axial	= 0 mm
Operating temperature range	– 30 ... + 100 °C

Specifications

reduction ratio (nominal)	weight without motor	length without motor L2	length with motor 1319 T L1	length with motor 1331 T L1	length with motor 1336 U L1	output torque continuous operation M max.	output torque intermittent operation M max.	direction of rotation (reversible)	efficiency
	g	mm	mm	mm	mm	mNm	mNm		%
3,71 :1	17	20,9	34,1	45,9	50,9	200	300	=	90
14 :1	20	25,0	38,2	50,0	55,0	300	450	=	80
43 :1	24	29,2	42,4	54,2	59,2	300	450	=	70
66 :1	24	29,2	42,4	54,2	59,2	300	450	=	70
134 :1	27	33,3	46,5	58,3	63,3	300	450	=	60
159 :1	27	33,3	46,5	58,3	63,3	300	450	=	60
246 :1	27	33,3	46,5	58,3	63,3	300	450	=	60
415 :1	30	37,4	50,6	62,4	67,4	300	450	=	55
592 :1	30	37,4	50,6	62,4	67,4	300	450	=	55
989 :1	30	37,4	50,6	62,4	67,4	300	450	=	55
1 526 :1	30	37,4	50,6	62,4	67,4	300	450	=	55
2 608 :1	34	41,5	54,7	66,5	71,5	300	450	=	50
4 365 :1	34	41,5	54,7	66,5	71,5	300	450	=	50
5 647 :1	34	41,5	54,7	66,5	71,5	300	450	=	50



14/1

Encoders

Magnetic Encoders

Features:
64 to 512 Lines per revolution
2 Channels
Digital output

Series IE2 – 512

		IE2 – 64	IE2 – 128	IE2 – 256	IE2 – 512	
Lines per revolution	N	64	128	256	512	
Signal output, square wave		2				channels
Supply voltage	V_{DD}	4,5 ... 5,5				V DC
Current consumption, typical (V_{DD} = 5 V DC)	I_{DD}	typ. 6, max. 12				mA
Output current, max. [1]	I_{OUT}	5				mA
Pulse width	P	180 ± 45				°e
Phase shift, channel A to B	Φ	90 ± 45				°e
Signal rise/fall time, max. (C_{LOAD} = 50 pF)	tr/tf	0,1 / 0,1				µs
Frequency range [2], up to	f	20	40	80	160	kHz
Inertia of code disc	J	0,09				gcm²
Operating temperature range		– 25 ... + 85				°C

[1] V_{DD} = 5 V DC: Low logic level < 0,5 V, high logic level > 4,5 V: CMOS and TTL compatible
[2] Velocity (rpm) = f (Hz) x 60/N

Ordering Information

Encoder	number of channels	lines per revolution	in combination with:
IE2 – 64	2	64	DC-Micromotors series 1336 ... C, 1516 ... SR, 1524 ... SR, 1717 ... SR, 1724 ... SR, 1727 ... C, 2224 ... SR, 2342 ... CR, 2642 ... CR, 2657 ... CR, 3242 ... CR, 3257 ... CR, 3863 ... C Brushless DC-Servomotors series 1628 ... B, 2036 ... B, 2444 ... B
IE2 – 128	2	128	
IE2 – 256	2	256	
IE2 – 512	2	512	

Features

These incremental shaft encoders in combination with the FAULHABER DC-Micromotors and Brushless DC-Servomotors are used for indication and control of both shaft velocity and direction of rotation as well as for positioning.

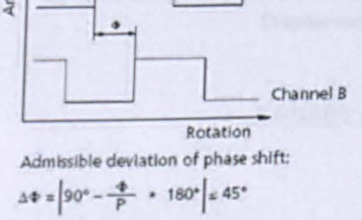
The encoder is integrated in the DC-Micromotors SR-Series and extends the overall length by only 1,4 mm. Built-on option for DC-Micromotors and Brushless DC-Servomotors.

Hybrid circuits with sensors and a low inertia magnetic disc provide two channels with 90° phase shift.

The supply voltage for the encoder and the DC-Micromotor as well as the two channel output signals are interfaced through a ribbon cable with connector.

Details for the DC-Micromotors and suitable reduction gearheads are on separate catalogue pages.

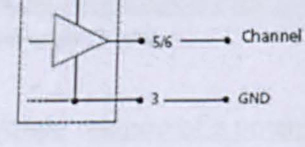
Output signals / Circuit diagram / Connector information



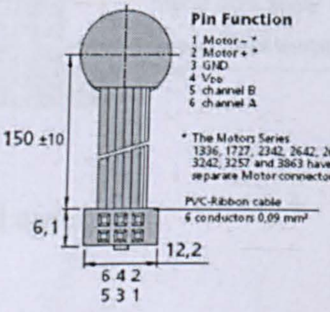
Admissible deviation of phase shift:

$$\Delta\Phi = \left| 90° - \frac{\Phi}{P} \cdot 180° \right| \leq 45°$$

Output signals
with clockwise rotation as seen from the shaft end



Output circuit
Note: Motor terminal resistance increases by approx. 0,4 Ω



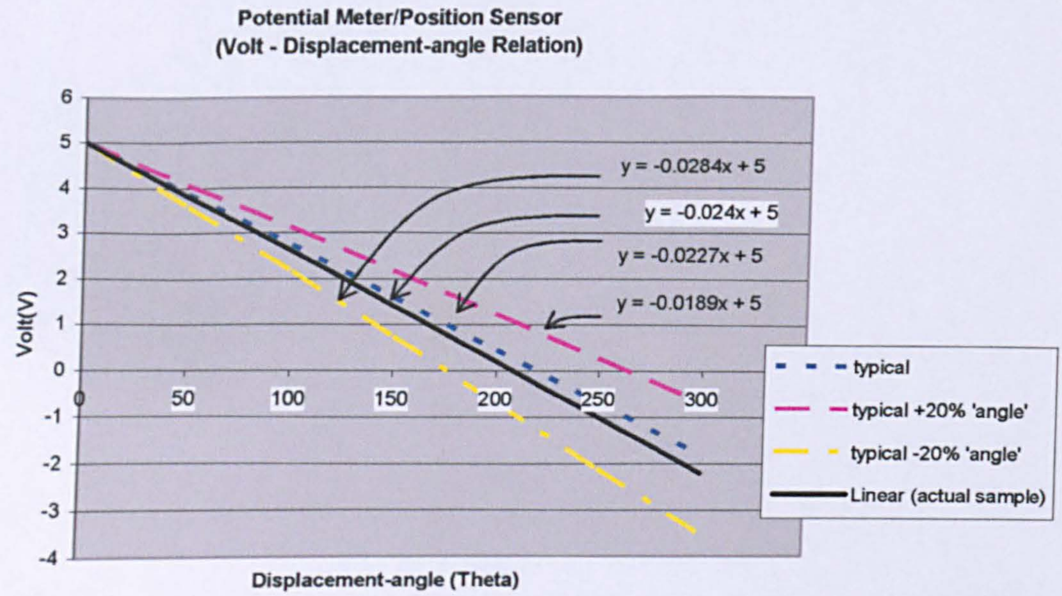
Connector
DIN-41651
grid 2,54 mm

Nominal Voltage	12	Volts DC
Armature Resistance	13.3	Ohms
No Load Speed	11300	Rpm (before gear)
No Load Current	0.015	Amps
Max. Efficiency	77%	
Max. Power	2.66	W
Gear head reduction	159:1	

Parameters of the DC motor

l (mm)	L (mm)	L/l	Theta(deg)	rad	r(mm)	r(m)	Fyl(g)	Fyl(N)	Tm(Nm)	Fm(N)
49.82	76.58	1.54	30.00	0.524	3.00	0.0030	400.00	3.92	0.036	12.06

Middle finger parameters for minimum torque estimation



Voltage-Angle relation of a potential meter