IMAGE SENSOR CLASSIFICATION SYSTEM FOR PROSTHESIS CONTROL

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Daniel Jason Schwandt

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Electrical and Computer Engineering

December 2010

Fargo, North Dakota

North Dakota State University
Graduate School

Title

Image Sensor Classification System

For Prosthesis Control

By

Daniel Jason Schwandt

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

ABSTRACT

Schwandt, Daniel Jason, M.S., Department of Electrical and Computer Engineering, College of Engineering and Architecture, North Dakota State University, December 2010. Image Sensor Classification System for Prosthesis Control. Major Professor: Dr. Mark J. Schroeder.

This paper proposes the integration of an image sensor classification system into the control scheme for prosthetic limbs. The goal is to minimize the mental strain experienced by brain-computer interface (BCI) users when performing routine tasks. In particular, the focus is to investigate the capabilities of image classification as a method for identifying and classifying objects. Incorporating this process into prosthesis control will allow more fluid and natural movements while minimizing the mental strain of multiple commands by the user. A popular imaging and classification software package was evaluated by exposing it to various images under different conditions. Results indicate that an image sensor classification system is suitable for prosthetic limb control in many situations; however, the system performance is reduced in low light and clustered object conditions. It is therefore feasible to consider the integration of image sensor classification and camera-in-hand technology into EMG- and BCI-controlled prosthetics.

ACKNOWLEDGMENTS

I am very grateful to take this opportunity to thank Dr. Mark J. Schroeder for being my advisor as I pursued a master's degree. I cannot thank him enough for the guidance, encouragement and wisdom he gave me during my graduate studies. He has shown me the joys of teaching, the rewards of research, and the importance of life.

I would like to thank Dr. Jacob Glower for serving on my graduate committee. I appreciate the comments and help he has provided me over my college career.

I would like to thank Dr. Subbaraya Yuvarajan and Dr. Majura Selekwa for serving on my graduate committee. I have always enjoyed discussing many different aspects of engineering and mathematics with these two gentlemen and look forward to future discussions.

Finally, I would like to thank my family for their support and encouragement during my time pursuing a master's degree.

DEDICATION

To my beautiful wife, Tara,

and to my awesome family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

Natural movements with simplistic controls are essential when designing prosthetic hands. There are several stages and approaches for controlling prosthetics. The initial stage is to collect a signal from the body. There are primarily two methods for accomplishing this. First, electrodes can be placed on the surface of the skin to collect myoelectric signals which initiate the prosthetic hand to react [1]-[4]. Alternatively, signals can be collected neurologically with invasive electrodes located along the nervous system [5]-[9]. The second stage is to identify these signals using a classification system such as an artificial neural network (ANN) [10]-[13] or fuzzy logic classifier [14]-[16]. Finally, these classified signals are used as specific input controls for the prosthetic hand to perform multiple commands [1], [3], [17]-[20].

In order to design a successful prosthetic, the device needs to have multiple degrees of freedom [3], [17]-[21], simplistic and intuitive controls [1], [19] and a responsive feedback system [1]. These introduce multiple problems however. First, in order to give the prosthetic multiple degrees of freedom, the user must voluntarily initiate the device for each movement. This produces an unnatural relationship with the device and further causes great mental strain on the individual. Thus, the user rejects the device entirely [1]-[2], [19]. Similarly, if the device can only utilize minimal controls from the user, this causes limitations in the performance of the device. Thus, the user also rejects this design [1], [17], [19]. Prosthetics need to essentially replace the missing limb, mimicking the natural and carefree movements of the limb.

Although there have been great advances in the field of myoelectric analysis and control for prosthetics, one area that has not been fully investigated is image sensor classification. Image sensors can be incorporated into the design of a prosthetic hand to help the individual control the device. These sensors can identify unique objects that are encountered on a daily basis and activate the prosthetic to perform a certain automated task related to such objects. For example, grasping a glass is an action that is performed multiple times a day by everyone. The user could initiate the movement of the prosthetic to the glass. The image sensor could identify the glass and its location. Then, the system could automate the prosthetic to grasp the glass and bring it back to the user. This system would need minimal input from the user, allowing less mental concentration for the mundane everyday tasks.

Using image, position and force sensors along with a strong myoelectric signal classifier could enable a more naturally controlled prosthetic. Why should the user need to control every action of the prosthetic? Automating multiple everyday tasks would allow the user to have natural movements with minimal thinking about controlling the device. Figure 1 illustrates a comparison between typical prosthetics and this image controlled prosthetic.

The purpose of this paper is to introduce the integration of image sensors into prosthetics in order to improve their control structure and minimize the constraints on the user. To establish the foundation needed to understand this topic, a brief history of the development of prosthetic control will be discussed in Chapter 2. Chapter 3 provides an introduction to the National Instruments software, Vision Builder AI, which was used to implement the classification structure. The methodology of the derivations and its

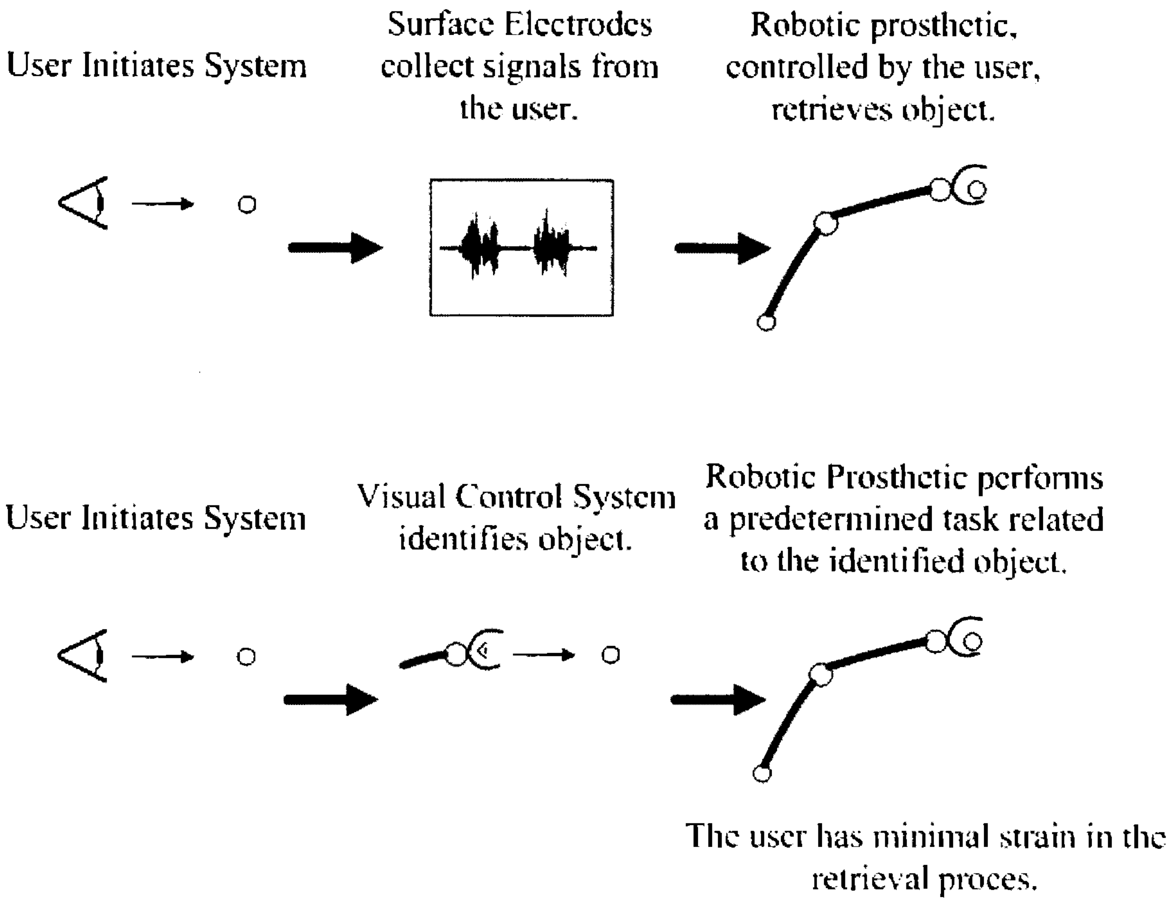

Figure 1. Comparison between typical prosthetics and visual based prosthetics.

implementation are discussed in Chapters 4 and 5, respectfully. The results are discussed in Chapter 6 and Chapter 7 provides the concluding remarks.

CHAPTER 2. BACKGROUND INFORMATION

2.1. Introduction

This chapter presents the details involved with acquiring electromyographic (EMG) signals from the body. There are primarily two methods for retrieving signals from the body. The first is to place electrodes on the surface of the skin to collect myoelectric signals to initiate the prosthetic to react. The second is to collect signals neurologically from invasive electrodes located along the nervous system or directly connected to the brain. The system connected between the brain and a device is called a Brain Computer Interface or BCI. Naturally, there are many ways to collect signals from the body. Only a few major methods will be introduced.

In addition, this chapter provides the details involved with how the acquired signals are analyzed and used for prosthetic control. There are multiple approaches for recognizing and classifying signals including artificial neural networks (ANN) [10]-[15], fuzzy logic classifiers [14]-[16], self organizing neural networks and genetic algorithms.

Visual-based robotic control is also discussed in this chapter. This method illustrates the prosthetic's mechanical process of retrieving objects by means of a camera-to-object relationship.

2.2. Description of Surface Electrodes

Surface electrodes are used to collect many signals from the body on the exterior of the skin. They are commonly used in the clinical setting to collect electromyographic (EMG) signals, electrocardiographic (ECG) signals and electroencephalographic (EEG) signals. EMG surface electrodes measure the biopotentials of muscle groups flexing and relaxing [23].

Surface electrodes are generally made from particular types of polymers or elastomers and then adding a fine carbon or metal powder to make them electrically conductive. The location the electrode will be placed is generally cleaned to limit the impedance between the electrode and the skin. However, to allow for easy and quick application, the electrodes can also be made with a prepaste AgCl gel [23]. The main advantage of surface electrodes is their ability to collect signals noninvasively [9]. This makes them easily replaceable and limits the possibility of infection. Unfortunately, the distance between the source and the electrode generates a noticeable amount of noise. Thus, the signals that are generally used are the larger signals like the ECG and EMG [9].

2.3. Description of Invasive Electrodes

Implantable electrodes provide a direct path to record the electrical signals from the brain, nerves and muscle fibers [5]-[9]. These electrodes are either surgically implanted or inserted into the muscle tissue with needles [23]. In the brain or nervous system, microelectrodes can be implanted to interact with individual cells. Neuron cells carry electrical impulses along their axons when an action potential is elicited. These action potentials can then be measured and used to interact with prosthetic devices [23].

The advantage of implantable electrodes is that there is far less noise associated with the signals and there are millions of signals to use [9]. Direct communication with the brain and neurons can provide the possibility of advanced control and dexterity. However, direct communication is difficult to achieve since the electrodes must be surgically implanted into the body, bringing forth multiple difficulties and risks [25].

2.3.1. Needle Electrodes

Needle electrodes provide direct connection with the electrical signals from the nerves and muscle fibers. Concentric bipolar electrodes are the most common needle electrodes. The recording and reference electrodes are made from two thin metallic wires that are incased in either a larger canula or hypodermic needle. The other common type of percutaneous electrode is the unipolar needle electrode. The needle is manufactured using a thin metal wire insulated entirely by a layer of Teflon except for the very tip. Since this is a unipolar needle, it requires a second reference electrode to create a closed electrical circuit. It can either be another unipolar electrode or a surface electrode placed near the first [23].

2.3.2. Microelectrodes

Microelectrodes are designed to interact with individual biological cells by inserting an ultra fine tip into the cell. In order to minimize cell damage, the electrode tip must be very small compared to the dimensions of the cell. These electrodes are used commonly in neurophysiological studies to record the action potentials of individual cells. These electrodes are typically manufactured in three different ways: (a) glass micropipettes, (b) metal microelectrodes, and (c) solid-state microprobes [23]. These electrodes can provide excellent signal-to-noise ratios because of their ability to record the action potentials of individual neurons.

2.3.3. Chronic Electrode Implants

Chronic electrodes can be surgically implanted to interact directly with the neurons of the brain. There are two typical designs for chronically implanted electrodes. The first and most widely used are microwire electrodes [33]. The individual electrodes are essentially designed similar to the needle electrode except on a miniature scale. Their structure consists of an array of conducting metal wires coated with an insulating layer except for their tips. Their simple design is their primary advantage being easily fabricated compared to the more sophisticated silicon arrays.

Silicon arrays are the next generation of chronic electrodes. With a more complex design, silicon micromachined electrodes offer the possibility to minimize the foreign body response which allows more flexibility for the placement of the electrodes. Advancements in silicon manufacturing allow for complete control over the designs of the electrode's size, shape, texture and placement of individual recording electrodes on the chip. Silicon based electrodes also allow the recording circuit to be integrated directly onto the chip itself for better signal acquisition [33].

2.3.4. Problems with Invasive Electrodes

Although invasive electrodes offer a wide selection of benefits, such as being able to directly measure signals from neurons, there are disadvantages with them. Whenever the skin barrier of the body is broken there is a possibility of infection. Brain surgery is a difficult procedure within itself. Long term use of implantable electrodes has shown that the body can consider these electrodes as foreign objects and does its best to control the area where the electrode was implanted by creating a layer of scar tissue between the electrode and the neurons [33].

2.4. Details Involved With Signal Classification

Signals are collected from the body with single or multiple electrodes, either on the surface or invasively [15]-[16]. There are multiple approaches on how to use these signals for prosthetic control. In order to use these signals to control a prosthetic device with multiple degrees of freedom, the signals need to be mapped to the particular muscle or neuron group that initiated the command [15]-[16]. Figure 2 illustrates how a digitized EMG signal is classified by an artificial neural network (ANN) system.

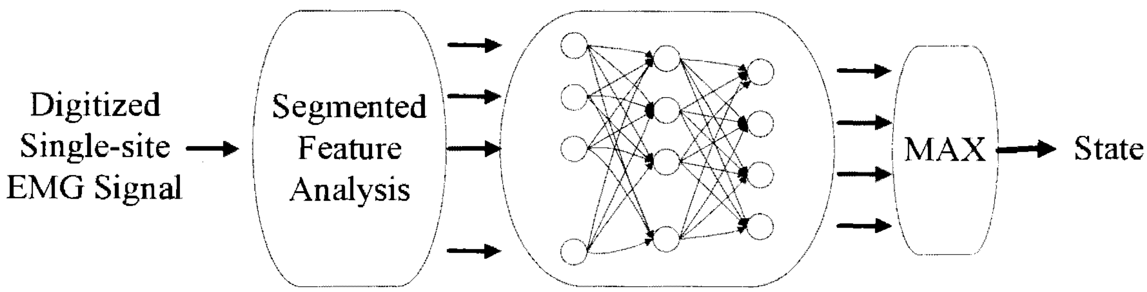

Figure 2. ANN EMG classification strategy.

2.4.1. Artificial Neural Networks

The human brain is a structure of interconnected neurons that communicate with each other through electrical impulses. This structure provides the basis for artificial neural networks [26]-[28]. These neural networks can be used as a form of parallel computing [26] to mathematically model the complex relationships between inputs and outputs [15] or as a pattern recognition tool [16]. Essentially, it is a mathematical function mapping the relationship between a set of inputs *(X)* to a set of outputs *(Y)*.

$$f : X \rightarrow Y \tag{2.1}$$

Artificial neural networks create a system of interconnected neurons in individual layers. The first layer is the input layer, where information is then sent to the second layer via synapses. Synapses are numerical weights which store coefficients that manipulate neuron calculations. The second layer is a hidden layer. Multiple hidden layers can exist to model more complex systems. The final layer is the output layer. Figure 3 shows an example architecture of an artificial neural network. The synapse weights are adjusted to minimize the output error during the training phase.

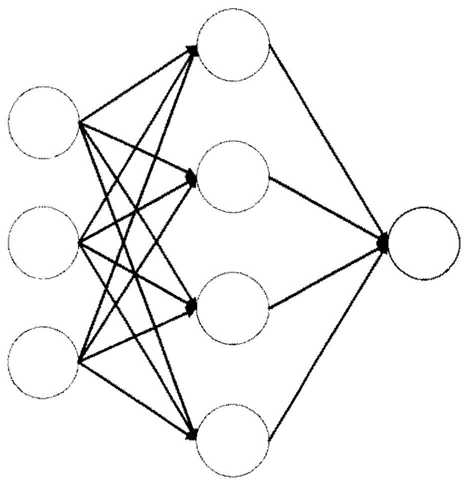

Figure 3. A simplified view of a neural network.

Artificial neural networks can be used to learn particular information which will then be used to perform mathematical tasks. The two primary types of learning are supervised and unsupervised learning. In supervised learning, the system is given sets of training data *(X, Y)* to learn. The system then learns to recognize the function *(f)* that relates the input *(X)* to the output *(Y)*. This can be used for pattern recognition or classification. In unsupervised learning, the system is given some input data *(X)* and the cost function *(f)*. The goal is to find the output *(Y)* while minimizing the cost function *(f)*; essentially trying to determine how the data is organized. This can be used for filtering and general estimation problems.

Supervised learning of artificial neural networks can be used to classify signals from the body. The training can be set up so the user performs particular tasks while the system learns the signals that are being created. Thus, the system is determining the function that relates the action to the signal.

2.4.2. Self Organizing Maps

Self organizing maps are a unique style of trained artificial neural networks that use unsupervised learning to produce a low dimensional representation of the input samples. These maps differ from artificial neural networks in that they use a neighborhood function to maintain the topological properties of the input data. This model has also been commonly called Kohonen maps. Self organizing maps typically consist of one dimensional or two dimensional grids of connected neurons or nodes by weights. Initially, the weights are typically given sample values evenly spaced throughout the grid. The training consists of competitive learning where the neuron with the weights that most closely match the input wins. This causes the neuron and the neurons closest to it to change their weights towards the input. The mathematical function of this process of competing neurons and adjusting weights $(W_v(t))$ is

$$W_v(t+1) = W_v(t) + \theta(v,t)\alpha(t)(X(t) - W_v(t)) \qquad (2.2)$$

where the monotonically decreasing learning coefficient is $\alpha(t)$and the input vector is $X(t)$. The neighborhood function, $\theta(v,t)$, depends on the distances between the winning neuron and the adjacent neuron [47].

Self organizing maps can be used as a signal identification tool. The identified unique signals can be mapped to a particular task.

2.4.3. Fuzzy Logic Classification

Fuzzy logic is a mathematical model for approximation. This is an essential tool when dealing with extremely complex systems, such as the human brain and nervous system. Fuzzy logic incorporates degrees of truth instead of just using true or false as in binary logic [29]-[32]. An excellent narrative of fuzzy logic was given by P. Hajek in the *Metamathematics of Fuzzy Logic.* "In a narrow sense, fuzzy logic ... is a logical system which aims at a formalization of approximate reasoning [29]."

2.5. Brain-Computer Interface (BCI)

Brain-Computer Interfaces (BCI) systems provide direct neural communication links from the electrical signals of the brain to controllable devices [25]. In order to use these systems, the signals need to be linked to the particular action that the user intended. Thus, most BCI systems require a fair amount of training for the user to understand the thoughts needed to control the device [40]. BCI systems have had many advancements in recent years and also provide a promising future for direct communication with the brain. In the near future, BCI systems will be able to work flawlessly with patients to provide natural moving prosthetics [41].

2.6. Visual Based Robotic Control

The use of cameras has become an interesting topic of discussion for robot control. There are multiple ways to use cameras as control strategies. The two primary methods are the fixed camera approach and the camera-in-hand or eye-in-hand approach [37]-[39]. The fixed camera system uses multiple cameras in fixed locations creating a coordinate system with images of both the robot and its surroundings [37], [39]. The camera-in-hand approach uses a camera mounted on the robotic arm, which supplies visual feedback to the system of the environment (Fig. 4). This mounted camera provides more flexibility to view a particular environment than the fixed camera system [37]-[39].

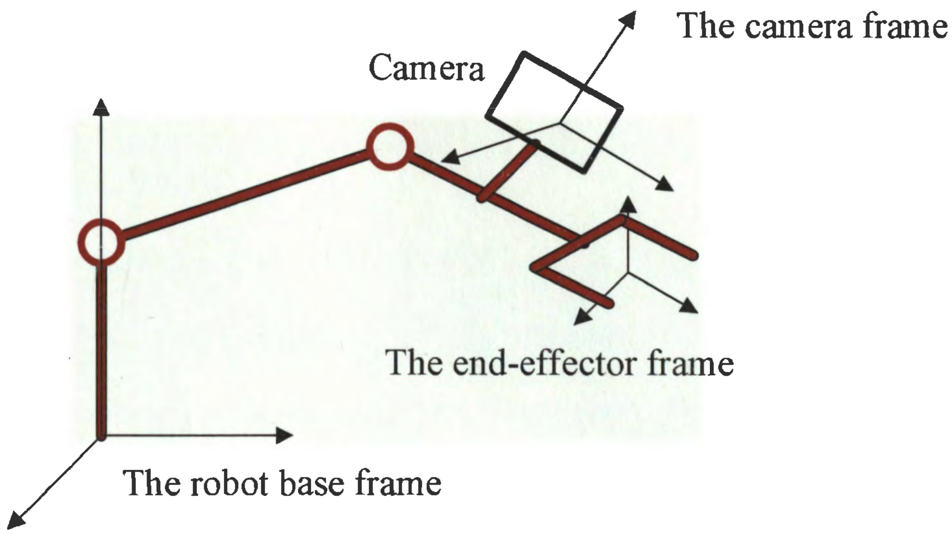

Figure 4. Camera-in-hand setup for visual control.

The camera-in-hand approach provides a natural method for prosthetic control. This system incorporated into a prosthetic can be used for both identifying objects and the retrieval process of locating and positioning the prosthetic in relation to the desired object.

CHAPTER 3. AN INTRODUCTION TO VISION BUILDER AI

3.1. Introduction

This chapter provides a brief introduction and explanation of the National Instruments Vision Builder AI software. Vision Builder AI has two separate views, the Configuration and Inspection Interfaces. The Configuration Interface is where the software is created and modified to perform the desired operations. The Inspection Interface is the viewable junction connecting the Configuration Interface to the readily available information created.

3.2. Configuration Interface

The Configuration Interface, shown in Figure 5, displays four primary areas: Main window, Overview window, State Configuration window, and Inspection Steps Palette. These windows are used to create an inspection.

1. Main window – displays either the image being processed or the state diagram for the inspection. The Main window can be used to modify the state diagram, define regions of interest within the displayed image and configure various step parameters.
2. Overview window – displays a small view of either the image being processed or the state diagram for the inspection.
3. State Configuration window – displays the selected state's list of steps in the inspection. The user programs each state (from window 2) in this window by selecting objects from window 4.

4. Inspection Steps palette – Lists and explains the steps used to create the inspection. These steps can be conformed to what is needed.

1	Main Window	3	State Configuration Window
2	Overview Window	4	Inspection Steps Palette

Figure 5. Vision Builder AI Configuration Interface.

3.3. Inspection Interface

The Inspection Interface of Vision Builder AI is used to run the configuration created in the Configuration Interface. It consists of three main areas shown in Figure 6: the Results Panel, the Display Window, and the Inspection Statistics Panel.

1. Results panel – Displays the Inspection States and steps. It also shows measurements or comments and the pass/fail results.
2. Display window – Displays the image under inspection.

3. Inspection Statistics Panel – displays indicators that show the yield (ratio between PASS and FAIL), active verses idle time, and processing time of the inspection.

1 Results Panel 2 Display Window 3 Inspection Statistics Panel

Figure 6. Vision Builder AI Inspection Interface.

3.4. Creating a Configuration for Inspection

This section will explain some of the different possibilities the software provides. The software is set up in two levels. First, the high level architecture is the inspection state diagram, which functions like a flowchart with branching and decision making abilities. The inspection proceeds from state to state depending on whether the transition or branch is met or proceeds with the default transition. The second level consists of the individual states which run a series of subroutines if and only if their prior transitions were met.

3.4.1. Inspection State Diagram

The inspection state diagram provides easy solutions to create new states and the transitions that connect them. The transitions follow Boolean logic which can be triggered from a variable declared in the previous state. For example, a transition could have a constraint of: if *Question* is *True* proceed through the flowchart, otherwise use default transition. There are a variety of possible constraints that can be used and declared within a state.

3.4.2. Individual State Subroutines

Subroutines can be created within the individual states to produce a variety of possibilities. Vision Builder has provided a great number of pre-coded steps available to use. These steps can be combined and customized to meet the needs of the particular software requirements. The steps on the palette are classified into eight categories; Acquire Image, Enhance Image, Locate Features, Measure Features, Check for Presence, Identify Parts, Communicate and Use Additional Tools. Each of these categorical tabs on the palette have their own unique operations.

- Acquire Image – this tab is used for interfacing with cameras to acquire the necessary images. There is also an option to simulate image acquisition from a declared folder.
- Enhance Image – this tab is used to alter the attributes of the acquired images. There are options to enhance the image's features, filter noise, and extract color planes. One step can calibrate the image to perform measurements in real world

units. One major step is the ability to create a region of interest within the image. This can be used to specify areas wanting to be investigated.

- Locate Features – this tab is used to located features within the image. These features can be used to find edges, straight edges and circular edges in a region of interest. Match pattern is a tool used to locate features within the image that match a pattern that has been defined. Geometric matching connects the features collected from the edge data. One of the most helpful steps is the detect objects step. This can be used to locate objects with homogenous intensities. It can be fine tuned to collect important information while disregarding useless noise.
- Measure Features – this tab has multiple features to measure attributes within an image. After an enhancement or location has been used within the software, measurements can be recorded. Combining several find edges steps could produce the geometric measurements of an object, such as length, width, area, etc. There are also measurement features to document light and color intensities.
- Check for Presence – this tab can be used to measure the intensity differences of a particular area within the image. Comparing shades of the object with its background environment.
- Identify Parts – this tab can identify characters, numbers and barcodes. This provides countless options for systems that need to read information. Such as part numbers.
- Communicate – this tab is used to read or write information from or to files.

- Use Additional Tools – this tab is the miscellaneous tab and has a variety of features to use. For instance, time delay functions, set the system status to pass or fail, initialize a counter, locate information within a stack, request information from the user and many more.

3.5. Pattern Matching and Classification Methodologies

This section describes the theory behind several of the matching and classification methods available in National Instruments AI software.

3.5.1. Pattern Matching Techniques

The pattern matching technique primarily uses normalized cross correlation to determine the mathematical similarities between a given sample image and an undefined image. This subroutine can be set up to return the empirical score of the cross correlation similarities. If the measurement is acceptable, the unknown image can be categorized as the given image.

This software uses a normalized cross correlation method for determining a template or pattern within an overall image, identifying feature's distances measured by

$$d_{f,t}^2(u,v) = \sum_{x,y}[f(x,y) - t(x-u,y-v)]^2 \tag{3.1}$$

(where f is the given image and the $\sum_{x,y}$ is of the boundaries containing the feature t's position at u,v). Expanding the equation gives

$$d_{f,t}^2(u,v) = \sum_{x,y}[f^2(x,y) - 2f(x,y)t(x-u,y-v) + t^2(x-u,y-v)] \tag{3.2}$$

Assuming that $\sum t^2(x-u,y-v)$ is a constant and approximating $\sum f^2(x,y)$ as a constant, then the remaining cross correlation

$$c(u,v) = \sum_{x,y} f(x,y)t(x-u,y-v) \tag{3.3}$$

is a measurement of the quantified similarities between the given image and the features. This information creates a template which is then used to match features within an unknown image for classification [44].

3.5.2. Classification Methods

There are two classification schemes used in this software, Nearest Neighbor and Support Vector Machines. Both use a two tiered approach of training on a sample set and classifying these sets into categories.

Nearest Neighbor classification is an intuitive system for categorizing data. This system defines objects based on their closest distances of a given trained sample within the featured space. Using the formula below to find the minimum distance from an input feature X of unknown class C_j is defined as

$$d(X, C_j) = \min_i d(X, X_i^j) \tag{3.4}$$

where $d(X_1, C_j)$ is the distance between X and X_i^j. X, of unknown class, is then assigned to the class of its nearest neighbor defined by the following classification rule.

$$X \epsilon \ Class\ C_j, if\ d(X, C_j) = \min_i d(X, C_i) \tag{3.5}$$

A second classification method used in the software is the Support Vector Machine method which uses a form of supervised learning to generalize a large set of trained samples into support vectors which are fewer and more manageable for computation. This method is more mathematically complex than distance based classification methods, but utilizes a more generalized approach and can exploit a large sample set since the set will be broken down into supporting vectors. This method compares the unknown sample with the supporting vectors of the given trained sample to determine the class. The unknown sample x is classified using the following formula.

$$x = \text{sgn}(\sum y_i a_i K_i(x_i, x) + b) \tag{3.6}$$

Here, y_i is the class association (-1 or +1), a_i is the weighted coefficient, K is the kernel function, x_i is the number of supporting vectors and b is the distance of the separation boundary of each class [43].

CHAPTER 4. DERIVATION OF CONTROL METHODS

4.1. Introduction

The work presented in Chapters 2 and 3 provides an excellent foundation for creating and implementing a new approach for robotic prosthetics control. The format for the next three chapters will be first, the development of the methods, then implementation of this method and finally a discussion of the results.

4.2. Determining the Control Method

The area of prosthetic control has had many advancements throughout the years. Originally within the area of signal acquisition, myoelectric signals were only collected from muscle tissue using surface electrodes. Now, entire electrode systems can be fully implanted to acquire and classify signals from the neurons in the brain and can even stimulate the surrounding tissue.

In the area of myoelectric signal control through the use of surface electrodes there has been a great deal of innovation. However, the functionality of the device is dependent on how much control is given to the user. This presents two alternative solutions. The first solution is a prosthetic device that can perform multiple tasks with great dexterity. However, the control system has to heavily rely on the user to input multiple commands over a long period of time. This very strenuous on the user. The second option is to give the user only a few control options. This is quite limiting to the possible functionality of the device [1], [2], [17]-[19]. This is also not ideal.

An initial idea was to use artificial neural networks to classify unique myoelectric signals from the arm. These unique signals would be collected from individuals as they tried to perform a task during training sessions. Training sessions are when users repeatedly try to perform tasks and the signals created are documented and linked to their respective task. This type of system would allow the user to react to an object, creating a unique myoelectric signal. The system would then classify the signal and perform the appropriate action based on the classification (Figure 7).

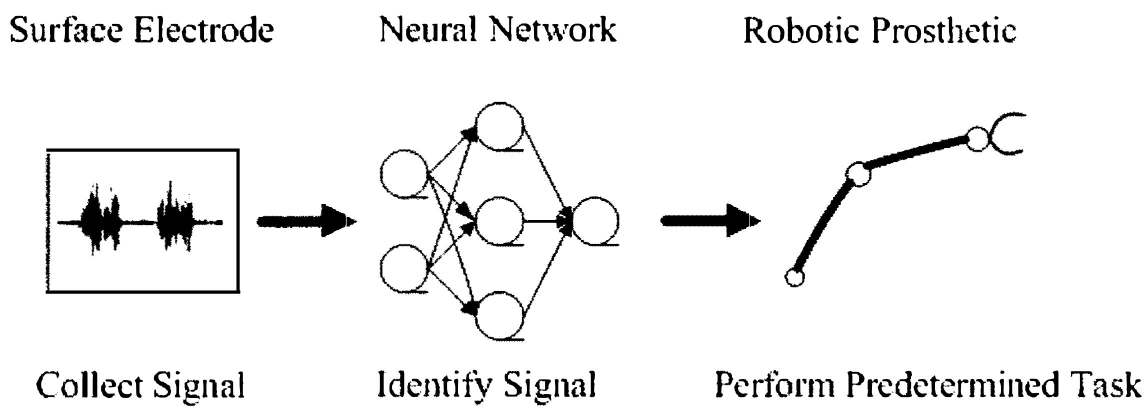

Figure 7. Original idea of the control structure.

Some background research revealed that the use of artificial neural networks for myoelectric prosthetic control had quite extensively been developed. Then, considering the need for a highly functional prosthetic device and also the need for a control system that has minimal mental strain on the user, the addition of image sensors incorporated into the control structure for prosthetics were examined.

Image systems that can identify unique objects and use that information to initiate a predetermined task were investigated. This idea was engineered from the thought of daily activities usually consisting of repeated common tasks, such as reaching for a glass of water. An image sensor based control system for prosthetic devices that can identify several objects that are encountered on a daily basis and perform a predetermined action defined by that object could reduce the mental strain on a user to perform those mundane tasks.

This system was set up so the user could initiate the device to acquire an image of the object. The system would identify the object and ask if the user would like to retrieve the object. The user would respond and the system would react appropriately. The implementation of this system would be quite fruitful given that the system can identify unique everyday objects and perform a particular predetermined task based on that object. However, considering that the user initiates the prosthetic to react toward an object, this implies that the user has already decided they would like to retrieve this object regardless of what the object is.

This thought introduced the idea of determining the proper action regardless of the object since the user already specified the desire to retrieve the object. This system's only concern would be the objects width, unless there was more than one object.

Once the user initiates the prosthetic system to react to an object, the system determines the number of objects. If there is only one object, the system proceeds to determine first, if the object is able to be grasped, then the object's width and angle of trajectory. If there is more than one object, the system asks the user to define which object to grasp and then proceeds as if there is only one object.

This method of assuming the user wants to retrieve an object when they initiate the prosthetic gives the user more freedom to worry about other details during the day than the voluntary act of guiding the prosthetic to retrieve the object.

4.3. Method Objectives and Set Up

The high level objective is to create a program that can improve the control structure of robotic prosthetics. The software chosen to accomplish this task is National Instruments Vision Builder AI.

Two different control approaches are considered. First, the specificity approach possesses the ability to identify unique objects from each other and to then react according to said object. For instance, this approach should decipher the difference between a pen and a ball. The second method is the capabilities approach, broadening the possibilities of identifying nonspecific objects and reacting to their geometric shape.

NI Vision Builder AI can utilize real time image acquisition using smart cameras or it can simulate image acquisition by selecting previously created images from a specified folder. The simulation method was chosen. Therefore, a library of images was created by using a digital camera to take single pictures of a variety of common objects and combinations as shown in Table 1. The images were used to determine if the vision acquisition software can be implemented to decipher unique objects. The images were taken in a controlled environment using a white background and several light intensities. The structure of the image was organized with either one or multiple objects in a simple or clustered arrangement. Future testing could incorporate a variety of backgrounds to determine effectiveness.

Table 1. List of common objects and quantities.

Quantity of Objects Within an Image	Number of Images Including:		
	Pens or Pencils	Utensils	Soda Can
1 Object	9	15	4
2 Objects	6	6	N/A
3 Objects	3	6	N/A
4 Objects	3	3	N/A
Cluttered Objects	3	5	N/A

The procedure was broken up into two steps; the training procedure and the testing procedure. During the training procedure, the system was taught to recognize objects by using a collection of categorized images of the individual objects at multiple light intensities and angles. This provided one category or object with multiple frames of reference for comparison. During the testing procedure, the system was tested on images with single objects and combinations of objects. This system would give an empirical accuracy score for each tested object, 0 – 100%, which would be used to determine how closely the tested object matches the trained objects. The score could then be integrated into a system that would retrieve the object. For example, eight images were taken of a single pen at different positions and light intensities. Figure 8 shows two of the trained images and two of the tested images. The system was trained on four of these images, giving the training system multiple frames of reference for the object. Then the testing

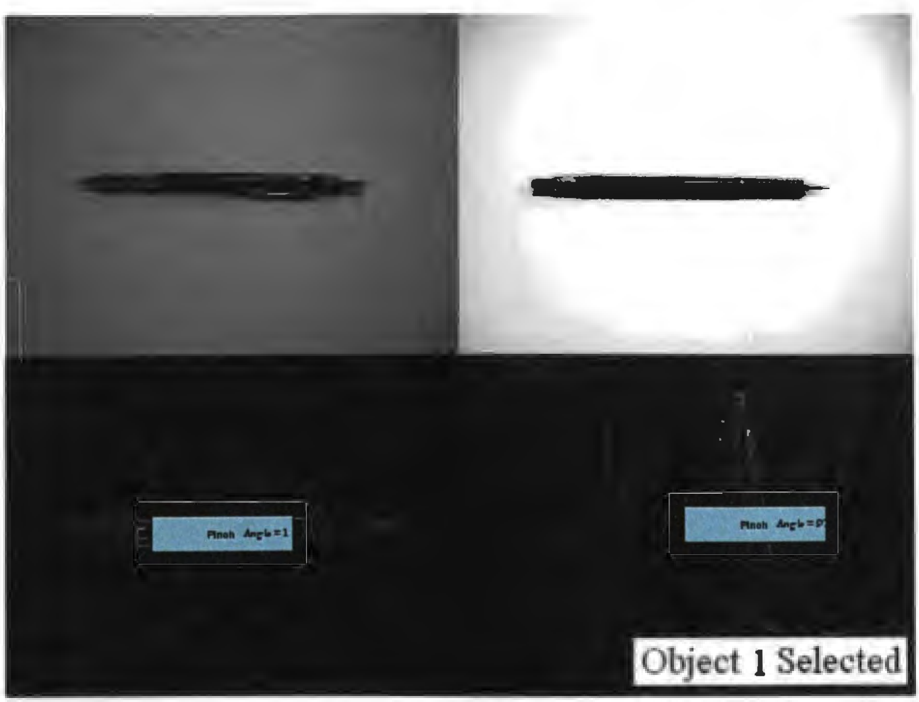

Figure 8. Trained (top images) and tested (bottom images) objects.

system would decipher the remaining four images and compare the reference object to the unidentified objects. If the matching accuracy score was greater than 80%, the system would continue to grasp the object.

CHAPTER 5. IMPLEMENTATION OF CONTROL METHODS

5.1 Introduction

Chapter 5 presents how the methods from Chapter 4 are implemented into the design of prosthetic control. There are multiple approaches and tools for using these ideas. Only a few will be discussed.

5.2. Software Architecture

National Instruments Vision Builder AI provides an excellent platform to analyze images, implement various design options and integrate with software that can be used to control robotic arms, such as LabVIEW [43].

A simplistic software interface is necessary for the designs of the prosthetic which require only minimal input controls from the user to perform the appropriate actions. The user simply has to initiate the system. The software then identifies how many objects are within the viewable frame. If there are multiple objects in the viewable frame, the program indicates the objects and requests which object the user would like to reach for. If there is only one object in the viewable frame the program passes through to reaching for the object. Figure 9 illustrates the high level software architecture and Figure 10 displays the actual state diagram from the Vision Builder software.

There are two control schemes that need to be considered in determining the recognition strategy to use. First, how specific is the software at recognizing objects? Second, do the objects even need to be recognized? If the user has declared the desire to retrieve an object, what it is, is no longer relevant. What is relevant are the attributes of this object. These two different control schemes are further discussed below.

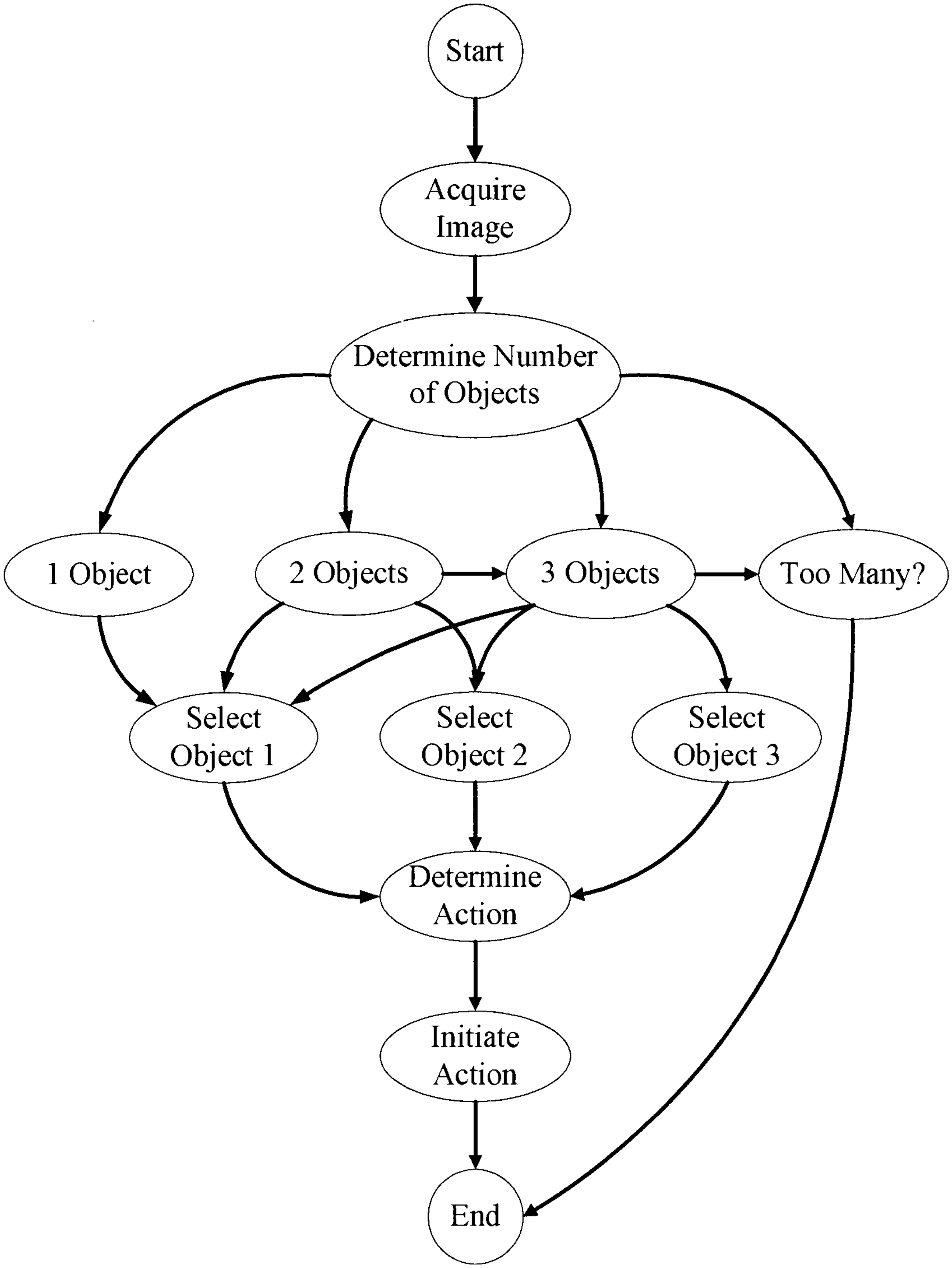

Figure 9. Software architecture.

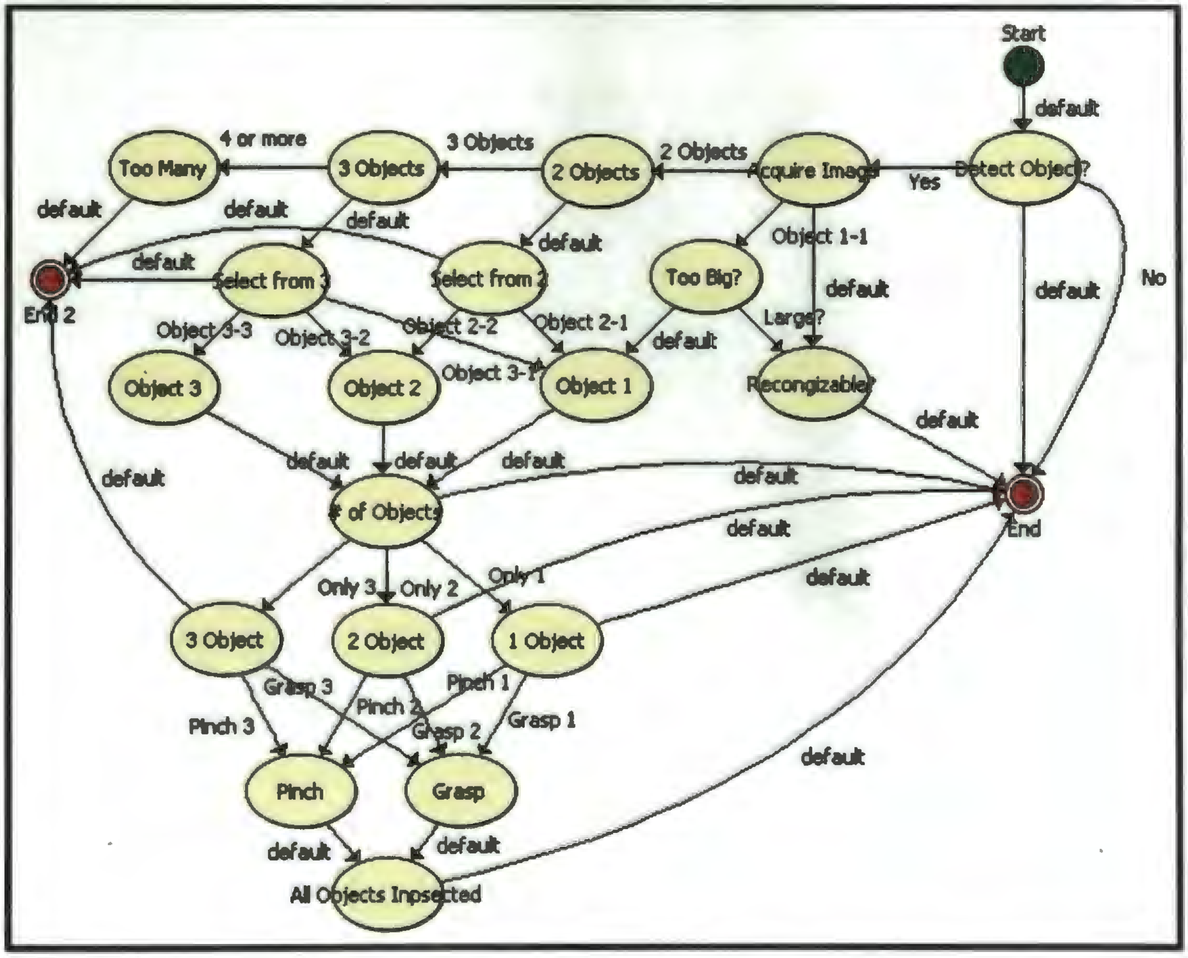

Figure 10. State diagram from the NI Vision Builder.

5.2.1. Specificity Approach

The specificity approach identifies unique objects and classifies them into specific categories that have predetermined tasks associated with them. For instance, when a pencil is identified, the program would initiate the prosthetic to reach out and grab the pencil in a predetermined way. This has a different category than reaching for a cup.

This implementation uses a library of collected images of chosen objects to classify objects during the real-time operations. A great variety of objects with multiple reference points and varied light intensities are needed to provide the most in depth range of classifiable objects. Each object is classified into categories of similar actions. Table 2 below illustrates several different action categories and their associated items. Obviously, the types, number of objects and actions can be drastically expanded. Also, more detailed movements can be implemented.

Table 2. Classifiable objects and actions.

Pinch	Grasp	Clutch	Hook
Pen	Cup	Door Knob	Door Handle
Pencil	Soda	Baseball	
Knife	Phone		
Fork			
Spoon			

During the real-time operations, images are captured from the camera after the user has declared the prosthetic to reach for an object. In order to compare these new images with the library of categorized images, the software must first enhance these new images so that useful information can be extracted. The images are filtered to block out unusual disturbances and color enhancements are used to illuminate the outlines of the objects.

With this information the software locates the objects, disregarding the small disturbances that were not filtered out, and compares them to the library of collected images to find a match. The software gives the object an empirical score (0 – 100) based on how closely the tested object matches the reference object from the library. If this score is above a certain threshold, the software declares what the object is and proceeds to perform the appropriate action. If there were multiple objects within the viewable frame, the software identifies and requests that the user choose which object they would like to retrieve. If the object is unrecognized, the software will not proceed to retrieve the object but display an error indicating that it is unrecognizable.

CHAPTER 7. DISCUSSION AND CONCLUSION

7.1. Discussion of Results

It is understandable to assume that better lighting conditions and more contrast between the objects color and their environment will naturally improve the performance of both the capabilities approach and specificities approach. This could be accomplished by using a camera with better sensors and the addition of a light flash. However, a visible flash during each iteration of the software would cause an annoyance to the user. This could be solved by using a camera and flash that was outside the visual spectrum, such as infrared spectrum.

The specificity approach had several instances where the software would identify an object as a different object. Typically this would not cause an issue since the identified object was quite similar to the actual object. For instance, a pen was identified even though it was a pencil. This risk is acceptable. However, in the rare instance where a pen was identified and the actual object was a soda can (this did not happen, but it is within the realm of possibilities during real-time operations), the software would initiate the prosthetic to reach for and pinch the "pen." This would obviously crush the soda can. Several simple abort functions could be incorporated into the design. The first would allow the user to abort any and all functions at any point during real-time operations. The second would be a stop or freeze function whenever the prosthetic's force sensors retrieve abnormal information; the system would stop all functions and report an error message. Continuing with the rare instance of an identified "pen," the moment the prosthetic begins to close its grasp around the "pen," which is really a soda can, the force sensors retrieve resistance information that is not accurate with attributes of a pen, it would stop all operations and display an error message.

Each of the issues do not pose a great threat to either the operability of the prosthetic device or the user's wellbeing. If any issue is encountered during real-time operations, the software would simply reflect an error message and stop all operations.

It is recommended that a hybrid combination of both the capabilities approach and the specificity approach be developed to provide the best of both worlds accentuating the benefits and limiting the defects.

7.2. Concluding Remarks

Modem technology is sustained or destroyed by the will of consumers. Products need to both simple and complex. The interface between device and controller has to be intuitive and lackadaisical. A product may have a revolutionary idea, but if it fails to meet the most basic ergonomics, it will be unsuccessful. Alternatively, if a product has an amazing interface with consumers, but severely lacks in depth or operations, it will also fail.

A process incorporating camera-in-hand technology into prosthetic control was developed to simplify the user's mental strain on routine tasks. Comparing the capability classification approach to the specificity object classification approach, the former provides an easier system for the user to utilize given that no prior library of information is needed. However, the addition of a library of images allows the specific classification approach to produce less classification errors. Both the capability and specific classification approaches provide less mental strain on users compared with the typical myoelectric controlled prosthetics since fewer commands are needed from them. This work shows proof of concept that either approach can provide users with a more natural moving prosthetic device. It is recommended that a hybrid version of both the capability and specific approaches be combined to greatly increase its versatility.

Further work could be done to broaden the program's ability to react with more objects and perform more predetermined tasks. Also, the hardware can be improved for better precision in positioning and more accurate results. This system would allow a more natural moving prosthetic device, where the user has to use little voluntary brain power to perform tasks. Another interesting topic would be the addition of nonvisual feedback systems to the user. This could include vibration or heat stimulators that allow the user to feel how hard and what types of objects they are grasping.

BIBLIOGRAPHY

[01] Cipriani, C., Zaccone, F., Micera, S., & Carrozza, M. C. (2008). On the Shared Control of an EMG-Controlled Prosthetic Hand: Analysis of User-Prosthesis Interaction. *IEEE Transactions on Robotics*, 170.

[02] Carrozza, M. C., Massa, B., Micera, S., Lazzarini, R., Zecca, M., & Dario, P. (2002). The Development of a Novel Prosthetic Hand-Ongoing Research and Preliminary Results. *IEEE/ASME Transactions on Mechatronics*, 108.

[03] Carrozza, M. C., Suppo, C., Sebastiani, F., Massa, B., Vecchi, F., Lazzarini, R., et al. (2004). The SPRING Hand: Development of a Self-Adaptive Prosthesis for Restoring Natural Grasping. *Autonomous Robots*, 125-141.

[04] Huang, Y., Englehart, K., Hudgins, B., & Chan, A. D. (2005). A Guassian Mixture Model Based Classification Scheme for Myoelectric Control of Powered Upper Limb Prostheses. *IEEE Transaction on Biomedical Engineering*, 1801-1811.

[05] Lowery, M., Weir, R. F., & Kuiden, T. A. (2006). Simulation of Intramuscular EMG Signals Detected Using Implantable Myoelectric Sensors (IMES). *IEEE Transactions on Biomedical Engineering*, 1926-1933.

[06] Liu, X., McCreery, D. B., Carter, R. R., Bullara, L. A., Yuen, T. G., & Agnew, W. F. (1999). Stability of the Interface Between Neural Tissue and Chronically Implanted Intracortical Microelectrodes. *IEEE Transactions on Rehabilitation Engineering*, 315-326.

[07] Linderman, M. D., Santhanam, G., Kemere, C. T., Gilja, V., O'Driscoll, S., Yu, B. M.,. (2008). Signal Processing Challenges for Neural Prostheses. *IEEE Signal Processing Magazine*, 18-28.

[08] Stieglitz, T., Schuettler, M., & Koch, K. P. (2005). Implantable Biomedical Microsystems for Neural Prostheses. *IEEE Engineering in Medicine and Biology Magazine*, 58-65.

[09] Micera, S., Citi, L., Rigosa, J., Carpaneto, J., Raspopovic, S., Pino, G. D., (March 2010). Decoding Information From Neural Signals Recorded Using Intraneural Electrodes- Toward the Development of a Neurocontrolled Hand Prosthesis. *Proceedings of the IEEE*, 407-417.

[10] Huang, Y., Englehart, K. B., Hudgins, B., & Chan, A. D. (2005). A Gaussian Mixture Model Based Classification Scheme for Myoelectric Control of Powered Upper Limb Prostheses. *IEEE Transactions on Biomedical Engineering*, 1801-1811.

[11] Liu, Y.-H., Huang, H.-P., & Weng, C.-H. (2007). Recognition of Electromyographic Signals Using Cascaded Kernel Learning Machine. *IEEE/ASME Transactions on Mechatronic* , 253-264.

[12] Iftime, S. D., Egsgaard, L. L., & Popovic, M. B. (2005). Automatic Determination of Synergies by Radial Basis Function Artificial Neural Networks for the Control of a Neural Prosthesis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 482-489.

[13] Adamczyk, M. M., & Crago, P. E. (2000). Simulated Feedforward Neural Network Coordination of Hand Grasp and Wrist Angle in a Neuroprosthesis. *IEEE Transactions on Rehabilitation Engineering*, 297-304.

[14] Ajiboye, A. B., & Weir, R. F. (2005). A Heuristic Fuzzy Logic Approach to EMG Pattern Recognition for Multifunctional Prosthesis Control. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 280-291.

[15] Karlik, B., Tokhi, M. O., & Alci, M. (2003). A Fuzzy Clustering Neural Network Architecture for Multifunction Upper-Limb Prosthesis. *IEEE Transactions on Biomedical Engineering*, 1255-1261.

[16] Momen, K., Krishnan, S., & Chau, T. (2007). Real-Time Classification of Forearm Electromyographic Signals Corresponding to User-Selected Intentional Movements for Multifunction Prosthesis Control. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 535-542.

[17] Huang, H.-P., & Chen, C.-Y. (1999). Development of Myoelectric Discrimination System for a Multi-Degree Prosthetic Hand. *IEEE International Conference on Robotics & Automation*, 2392.

[18] Pons, J. L., ROCON, E., & CERES, R. (2004). The MANUS-HAND Dextrous Robotics Upper Limb Prosthesis: Mechanical and Manipulation Aspects. *Autonomous Robots*, 143-163.

[19] Englehart, K., & Hudgins, B. (2003). A Robust, Real-Time Control Scheme for Multifunction Myoelectric Control. *IEEE Transactions on Biomedical Engineering*, 484.

[20] Petroff, N., Reisinger, K. D.. & Mason, P. A. (2001). Fuzzy-Control of a Hand Orthosis for Restoring Tip Pinch, Lateral Pinch, and Cylindrical Prehensions to Patients with Elbow Flexion Intact. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 225.

[21] Englehart, K., Hudgins, B., & Parker, P. A. (2001). A Wavelet-Based Continuous Classification Scheme for Multifunction Myoelectric Control. *IEEE Transactions on Biomedical Engineering*, 302.

[22] Chan, F. H., Yang, Y.-S., Lam, F. K., Zhang, Y.-T., & Parker, P. A. (2000). Fuzzy EMG Classification for Prosthesis Control. *IEEE Transactions on Rehabilitation Engineering*, 305.

[23] Enderle, J., Blanchard, S., & Bronzino, J. (2005). *Introduction to Biomedical Engineering*. San Diego: Elsevier Academic Press.

[24] Gronborg, R. (2007, Jaunary 9). *New Outstanding Disposable Concentric Needle Electrode from Ambu® Neuroline*. Retrieved July 19, 2010, from Ambu: http://www.ambu.com/COM/News_-_Events/NewsShow.aspx?M=News&PID=226&NewsID=9

[25] McFarland, D., & Wolpaw, J. (2008). Brain-Computer Interface Operation of Robotic and Prosthetic Devices. *IEEE Computer Society*, 52-56.

[26] Gurney, K. (1997). *Neural Networks*. London: UCL Press.

[27] Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press.

[28] Abdi, H., Valentin, D., & Edelman, B. (1999). *Neural Networks*. Thousand Oaks, California: Sage Publications, Inc.

[29] Hajek, P. (2001). *Metamathematics of Fuzzy Logic*. Norwell, MA: Kluwer Academic Publishers.

[30] Wang, P., Da, R., & Kerre, E. (2007). *Fuzzy Logic: A Spectrum of Theoretical and Practical Issues*. Berlin: Springer.

[31] Ross, T. (2004). *Fuzzy Logic with Engineering Applications*. West Sussex, Englend: John Wiley & Sons, Ltd.

[32] Gerla, G. (2001). *Fuzzy Logic*. Norwell, MA: Kluwer Academics Publishers.

[33] Polikov, V., Tresco, P., & Reichert, W. (2005). Response of brain tissue to chronically implanted neural electrodes. *Journal of Neuroscience Methods*, 1-18.

[34] Thompson, J. (2010). *Electromyography.* Retrieved July 19, 2010, from Austin Peay State University: http://www.apsu.edu/thompsonj/Anatomy%20&%20Physiology/2010/2010%20Exam%20Reviews/Exam%203%20Review/CH%2009%20Electromyography.htm

[35] Kindlmann, G., Normann, R., Badi, A., Bigler, J., Keller, C., Coffey, R., et al. (2003). Imaging of Utah Electrode Array, Implanted in Cochlear Nerve. *NIH Symposium on Biocomputation & Bioinformation.* Bethesda, Maryland: Scientific Computing and Imaging Institute (SCI).

[36] Medgadget, E. (2009, April 28). *Read/Write Brain Electrodes Getting Smaller and Smarter.* Retrieved July 19, 2010, from Medgadget: http://medgadget.com/archives/2009/04/readwrite_brain_electrodes_getting_smaller_and_smarter.html

[37] Wang, H., Liu, Y.-H., & Zhou, D. (2008). Adaptive Visual Servoing Using Point and Line Features With an Uncalibrated Eye-in-Hand Camera. *IEEE Transactions on Robotics*, 843-857.

[38] Muis, A., & Ohnishi, K. (2005). Eye-to-Hand Approach on Eye-in-Hand Configuration Within Real-Time Visual Servoing. *IEEE/ASME Transactions on Mechatronics*, 404-410.

[39] Kelly, R., Carelli, R., Nasisi, O., Kuchen, B., & Reyes, F. (2000). Stable Visual Servoing of Camera-in-Hand Robotic Systems. *IEEE/ASME Transactions on Mechatronics*, 39-48.

[40] Moore, M. (2003). Real-World Applications for Brain-Computer Interface Technology. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 162-165.

[41] Jackson, A., Moritz, C. T., Mavoori, J., Lucas, T. H., & Fetz, E. E. (2006). The Neurochip BCI: Towards a Neural Prosthesis for. *Upper Limb Function*, 187-190.

[42] Hollmer, M. (2009, June 10). *Brain-Computer Interface Developed at Brown, Begins New Clinical Trials.* Retrieved July 19, 2010, from Brown University News: http://news.brown.edu/pressreleases/2009/06/braingate2

[43] NI, C. (2010). *NI 1764 Smart Camera.* Retrieved July 19, 2010, from National Instruments: http://sine.ni.com/nips/cds/view/p/lang/en/nid/205959

[44] Lewis, J. P. (1995). *Fast Normalized Cross-Correlation.* Vision Interface. Interval Research, Palo Alto CA.

APPENDIX A. ASPECTS OF SOFTWARE ARCHITECTURE

The section provides a description of the primary aspects of the software program *GeneralClassifier.vbai*. Each state was created in the Configuration Interface window of National Instruments Vision Builder AI. Many of the minute details will not be discussed. Figure 15 below displays the high level architecture of the software.

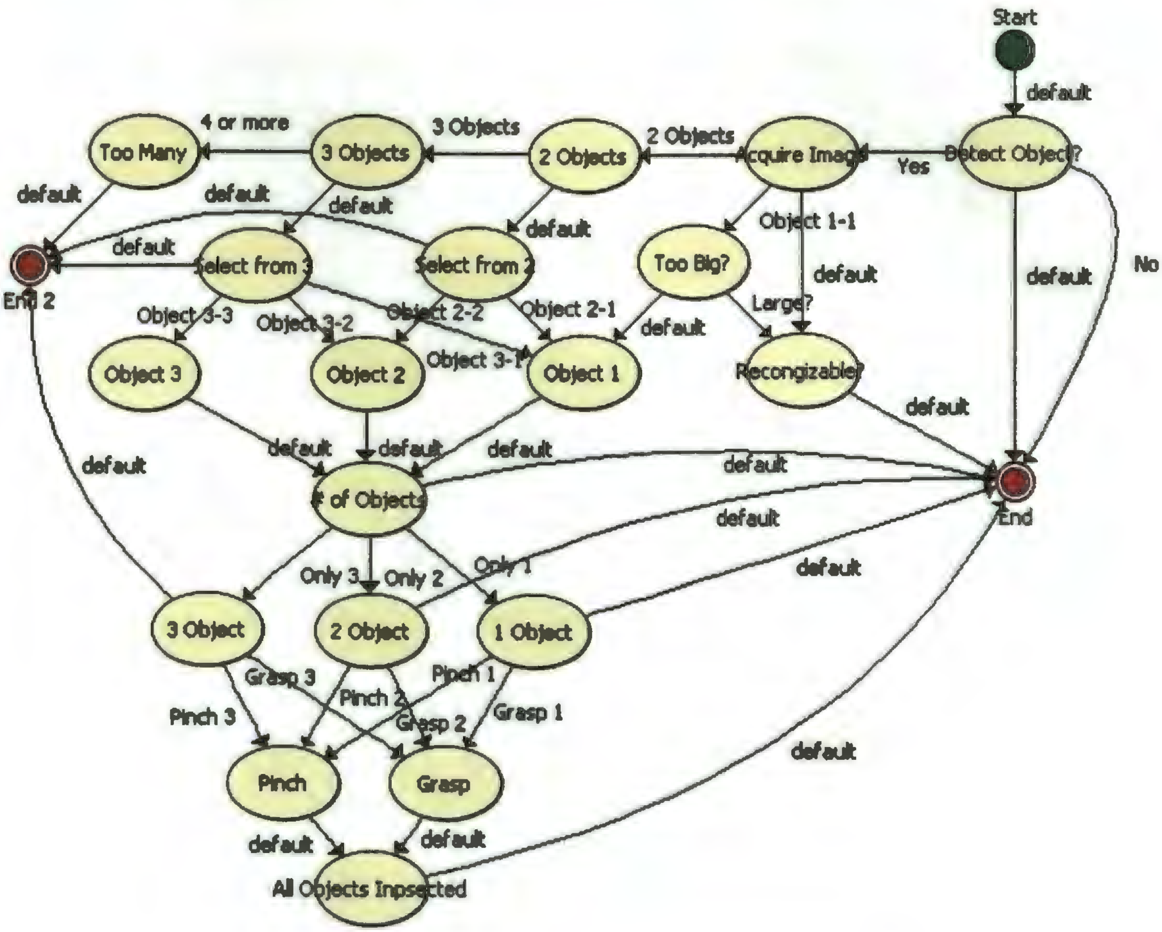

Figure 15. State diagram from the NI Vision Builder.

The software begins with asking whether or not the user would like to retrieve an object. If yes, the software continues to the Acquire Image state. If no, the software simply ends.

The Acquire Image state is a complex state with intricate subroutines. This state begins with acquiring an image, either by use of a camera or simulation. Once an image is acquired, the software begins with filtering the image to enhance the desired features by softening the background noise and intensifying the attributes. This allows the software to determine how many objects are within the viewable frame. This subroutine can be set up to ignore minor imperfections within the image. If objects were collected, a counter is initialized to one to keep track of how many objects are within the frame. Also, the first object's coordinates are made into a global variable to be able to retrieve at a later date. Using the object's coordinates, an "Object 1" icon is displayed on the first object within the viewable frame.

If there were two objects available, the system would move to the 2 Objects state that extracts the second object's coordinate data and makes that data into a global variable. Also, this state displays a similar icon "Object 2" onto the second object. If there were three objects, a similar state would also progress.

In the Select from 2 state, the user would be asked to choose which object they would like to retrieve in the event of multiple objects within the viewable frame.

The next state and its multiple related states (Object 1, Object 2 and Object 3), depending on which object is chosen to retrieve, displays an icon "Selecting This Object." Also within this state, a global variable is set to the appropriate selected object.

The number of objects state is used to create a region of interest around the desired state. It selects the appropriate object using the global variable set up in the previous state.

The next states, 1 Object, 2 Object and 3 Object, are used to determine the geometric features of the specified object. This state uses the global variables set up previously determining which object to analyze and its coordinate data. Using this information to create more specified regions of interest, a subroutine is set up to determine the width between two accented borders of the object. Figure16 illustrates this action. Depending on whether there was only one object identified or multiple objects identified, an individual circumference determination routine was given. If only one object was identified, the sample frame was broadened to allow for an objects more robust circumference as opposed to a frame with multiple objects.

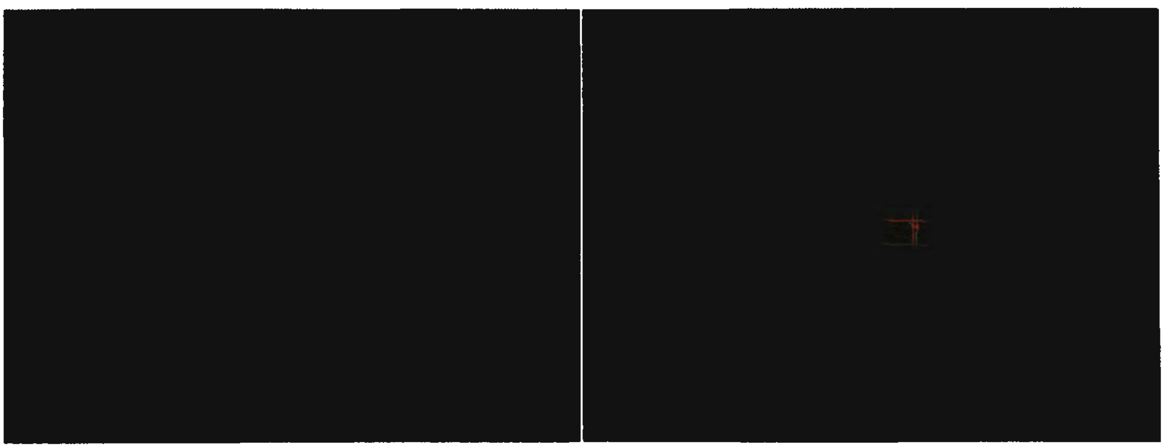

Figure 16. Width determination routine.

The Pinch or Grasp states were set up to simply reflect the possibility of using a robotic arm to grasp the object since no hardware was considered for this procedures.

In addition, the specificity approach uses a subroutine set up to classify objects. This subroutine is where the library of trained images is collected and the empirical score is derived. The images of the objects are collected and categorized and the system is then trained to recognize images with similar attributes. The number of trained images and categories could expand greatly. The empirical accuracy score is given to the tested images as compared to the trained object. For example, a pen would relate to the pen category with .954 accuracy rate. This is then used to proclaim that this is a pen, since the accuracy score is above a certain threshold.