Reduction of Dimensionality of a Cellular Actuator Array for Driving a Robotic Hand

By

Kyu-Jin Cho

Bachelor of Science in Mechanical Engineering, Seoul National University, 1998

Master of Science in Mechanical Engineering Seoul National University, 2000

Submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

Massachusetts Institute of Technology February 2007

© Massachusetts Institute of Technology 2007. All rights reserved.

Author				
		1	Pepartment of M	echanical Engineering
			-	Jan. 30, 2007
Certified by .			marries of a Vian	
			· 1	H. Harry Asada
		F	Ford Professor of M	echanical Engineering
				Thesis Supervisor
Accepted by.			· · · # · · · · · · · · · · · · · · · · · · ·	
				Lallit Anand
		Chairman, Dep	artment Committee	on Graduate Students
MASSACHUSETTS INSTITUTE				
OFTEURNOLDGY				
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				
JAN 0:3 2008				
	ARCHIVES			
LIBRARIES	• • • • • • • • • • • •			

Reduction of Dimensionality of a Cellular Actuator Array for Driving a Robotic Hand

By Kyu-Jin Cho

Submitted to the Department of Mechanical Engineering on Jan. 30. 2007 in partial fulfillment of the requirements for the degree of Doctor of Philosophy

ABSTRACT

In an attempt to explore an alternative to today's robot actuators, a new approach to artificial muscle actuator design and control is presented. The objective of this research is to coordinate the multitude of artificial muscle actuator axes for a large DOF (degree of freedom) robotic system based on dimensionality reduction.

An array of SMA actuators is segmented into many independently controlled, spatially discrete volumes, each contributing a small displacement to create a large motion. Segmented Binary Control is proposed where each segment is controlled in an on-off manner, creating a stepper-motor like actuator. This overcomes hysteresis and other nonlinearities of the actuator material. The segmented cellular architecture of SMA wires is extended to a multi-axis actuator array by arranging the segments in a two-dimensional array. The multi-axis control is streamlined and coordinated using a grouping of segments called *C-segments* in order to activate multiple links of a robot mechanism in a coordinated manner. This allows control of large DOF with a small number of controls.

The proposed approach is inspired by the segmented architecture of biological muscles and synergies, a strategy of grouping output variables to simplify the control of large number of muscles. Data from various hand postures are collected using data glove and used in creating the C-segment design that is capable of performing the given postures.

A lightweight Robotic Hand with 16 DOF is built using shape memory alloy actuators. This hand weighs less than 1kg including 32 SMA actuators and control circuitry. Eight C-segments that are ON-off controlled are used to create sixteen given postures. In the future, this approach can be applied to applications where the control signal is inherently limited due to limited amount of information that can be extracted or transferred to the robot, such as brain machine interface and tele-operation.

Thesis Supervisor: H. Harry Asada Title: Ford Professor of Mechanical Engineering

Acknowledgements

First of all, I would like to thank Prof. Asada for his guidance and support over the last few years. With his valuable insights and enthusiasm, he constantly encouraged me to explore new and challenging topics which sometimes lead me to achieve exciting results. His constant encouragement to go beyond the boundaries will be a valuable lesson and something that I'll try to do throughout my career.

I also would like to thank my committee members, Prof. Kamal Youcef-Toumi, Prof. J.J. Slotine and Dr. Karl Iagnemma, for their valuable inputs to this work.

It was great to have the DLab members that made my life at MIT much more pleasant. Especially, Binayak Roy and Arin Basmajian with whom I've shared a lot of fun stories. Other members of the group, Eric Wade, Phil Shaltis, Melissa Barbagelata, Devin McCombie, Tao Cheng, Jin-Oh Han, Levi Wood, Manas McMennon, Chris Brown, Tom Secord, Lael Odner, Josiah Rosmarin and Jun Ueda, thank you guys for the all the discussions in variety of topics from research to life

I would like to thank Dr. Sokwoo Rhee, alumni of our lab, for his encouragement and advice. His advice based on his experience in the lab helped me a lot. I owe a lot to him.

I believe I was able to survive my first year playing squash with my squash group every Friday. I thank Kyungyoon Noh and his wife, and Jongyoon Kim and other members of the squash group for all the hours that I had sweated on the court with them.

The second and third year at MIT was unique, living with my roommates, Sangyoon Min, Hyungjoon Cho and Hyunkyu Kim. You guys were like my brothers.

I also would like to thank members of the KGSAME. KGSA soccer team and the red racquets tennis club.

Most importantly, I would like to thank my Mom, Dad, my brother and my sister-in-law for their support, love and belief in me. I owe this thesis to them.

The most important person of my life, of course, is my wife Min-young and I cannot thank her enough for her love and support throughout the years.

Table of Contents

Chapter 1. Introduction
1.1. Artificial Muscle Actuators
1.2. Coordination of multi-axis actuator system10
1.3. Thesis goal and overview11
Chapter 2. Coordination using artificial muscles
2.1. Introduction14
2.2. Background15
2.3. Actuator Level Coupling
2.4. Fundamental characteristics of Actuator level coupling
2.4.1. Coupling coefficient and coupling theory of the actuator material
2.5. Summary24
Chapter 3. Cellular Actuator Arrays25
3.1. Introduction25
3.2. Background
3.3. Segmented Architecture of SMA Actuators / Segmented Binary Control26
3.3.1. Segmented Binary Control27
3.3.2. Verification of segmentation through local activation
3.4. Segmentation architecture for multi-axis system
3.5. Summary
Chapter 4. Algorithms for Generating the C-segment Architecture
4.1. Introduction
4.2. Problem Formulation
4.3. Constraints in C-segment design
4.3.1. Limited length of sum of rows of W matrix
4.3.2. Constraint on H matrix
4.3.3. Non-Negativity Constraint37
4.3.4. Transformation of negative components
4.4. Principal Component Analysis40
4.5. Nonnegative Matrix Factorization42

4.6. Comparison of PCA and NMF	42
4.6.1. Case Study using 6 Dimensions	44
4.7. Binary Algorithm for Implementation of SBC	46
4.7.1. Binary NMF	46
4.8. Improved solution using simulated annealing	47
4.9. Summary	50
Chapter 5. Application to an Anthropomorphic Robot Hand	51
5.1. Data collection of postures	51
5.2. Transformation of joint angles to actuator displacement	53
5.3. C-segmentation Design for MHI Grasps	55
5.4. Summary	60
Chapter 6. Hardware Implementation of Cellular Actuator Array and Experimental Resul	ts61
6.1. Shape Memory Alloy Actuator	61
6.2. Implementation of SBC using thermoelectric modules	62
6.2.1. Basic Structure	62
6.2.2. Single-Axis Setup	63
6.2.3. Ten-Axis Actuator Array	64
6.3. Implementation using Joule Heated SMA	65
6.3.1. Concept	65
6.4. Experimental Results	68
6.5. Hardware Implementation of Robotic Hand with Cellular Actuator Array	74
6.6. Implementation with TED driven SMA	75
6.7. Implementation using Joule Heated SMA	82
6.8. Summary	84
Chapter 7. Conclusion	86
7.1. Future Areas of Research	86
7.1.1. Brain Machine Interface	86
7.1.2. Application to new artificial muscle actuators	87
7.1.3. Optimization of the design	87
7.1.4. Trajectory following using desired postures	88

REFERENCES		89
APPENDIX A: F	Proof of the theorem 1	94
APPENDIX B: I	Heuristic approach of generating the C-segments and it's application to a r	obotic
hand		96
APPENDIX C: N	Mechanical design of the robotic hand	105
APPENDIX D: I	Drive circuitry of the C-Segmentation board	115

List of Figures

Fig. 1-1 Overview of the Process
Fig. 2-1 Concept of dimensionality reduction using C-segments
Fig. 2-2 Coupled and independent segments of m axis array20
Fig. 2-3 (a) Postures of m axes, (b) All On and OFF states for all the m axes, (c) The common
segments21
Fig. 3-1 Schematic of multi-axis artificial muscle
Fig. 3-2 (a) Traditional SMA heating method and (b) Segmented Binary Control. N is the number
of segments and n is the number of austenite segments (c) Phase transition diagram of SMA.
Diagram shifts with change in stress
Fig. 3-3 Minimum segmentation of single axis
Fig. 3-4 Two-dimensional segmentation of multi-axis SMA actuator system
Fig. 4-1 Representation of single desired output using C-segment Matrix and a Encoding vector
Fig. 4-2 Mathematical formulation of the C-segment design problem
Fig. 4-3 W matrix and corresponding design
Fig. 4-4 Example of implementing single component from PCA. Single component is equivalent
to single coordinate axis41
Fig. 4-5 Implementation of the different encodings on the antagonistic actuator. Just like the C-
segmentation design, negative values can be treated by using the antagonistic actuator41
Fig. 4-6 Comparison of mean joint angle error and standard deviation for NMF and PCA results
Fig. 4-7 Results of using PCA
Fig. 4–8 Results of using NMF
Fig. 4-9 Comparison of errors for NMF and PCA
Fig. 4-10 Sigmoid function for binary constraint
Fig. 4-11 Comparison of solutions found by NMF and applying SA to the solutions
Fig. 4-12 Flow chart of the binary algorithm
Fig. 5-1 Points of angle measurements in the cyberglove
Fig. 5-2 Pictures of 15 MHI Hand Grasps being taken while wearing the cyberglove. The
graphics in the box shows the captured postures, while the larger pictures are the actual hand.53
Fig. 5-3 Cross-section of a index finger of the robotic hand: SMA actuator is used as a flexor to
bend the finger and SMA spring is used as an extensor to provide bias force that pulls the finger
back to full extended posture when the SMA actuator is not activated54
Fig. 5-4 C-segment design in the actuator
Fig. 5–5 Eight C-segment Postures generated from the dataset
Fig. 5-6 Comparison of desired posture(left) and generated posture(right)
Fig. 5-7 mean of the error of each joint created by the actuator with clustering. (below)
Standard deviation of the error
Fig. 5-8 Two principal components shown as a robotic hand postures
Fig. 5-9 C-segmentation design of the actuator60
Fig. 6-1 Experimental results of a strain-temperature graph of 0.015" diameter SMA actuator
wire
Fig. 6-2 Schematic of one segment of an actuator array. Thermoelectric module on a grooved
substrate board drives the segment of SMA actuator wire
Fig. 6-3 Setup for single axis minimum segmentation
Fig. 6-4 Result of the unit step displacement experiment
Fig. 6-5 Top view of ten-axis SMA actuator array prototype. Segmentation architecture is

shown with the heat sink design	65
Fig. 6-6 Electrical posts used for Joule heated SBC	66
Fig. 6-7 Forming C-segments	66
Fig. 6-8 Grouping segments in parallel to form a C-segment	67
Fig. 6-9 Problem of serial connection	67
Fig. 6-10 Implementation of C-segment with Joule Heated SBC	68
Fig. 6-11 Experimental Setup of Single Axis Joule-Heated Segmented Binary Contr	ol of SMA
actuator	68
Fig. 6-12 Sensitivity Analysis of SMA actuator at various loading conditions	69
Fig. 6-13 Experiment of heating up and cooling with various current	70
Fig. 6-14 Experimental result of activation under different loading conditions wit	h different
currents for 1 sec	71
Fig. 6-15 Slope of the SMA wire under different loading conditions	72
Fig. 6-16 Cooling Temperature to martensite start temperature	73
Fig. 6-17 Cooling time to 10% of maximum displacement	73
Fig. 6-18 Solid Works model of Robotic Hand (Exploded View)	74
Fig. 6-19 Picture of the prototype robotic hand	75
Fig. 6-20 Schematic of one segment of an actuator array. Thermoelectric module on a	a substrate
board drives a segment of SMA actuator wire	76
Fig. 6-21 Different views of multi-axis actuator array system that drive 12 SMA actua	tors77
Fig. 6-22 Peltier board showing 96 peltier elements connected using a PCB board th	at is wired
and connectors placed on one side	78
Fig. 6-23 Cover plate for the heatsink and the springs that apply pressure on to the he	at sinks 78
Fig. 6-24 Connection board that routes 96 peltier elements into 24 drivers	79
Fig. 6-25 Procedure of assembly of peltier board and heat sink on to the actuator syst	em79
Fig. 6-26 Other side of the robotic hand system	80
Fig. 6-27 Features of the actuator system	81
Fig. 6-28 View of the final assembled robotic hand	82
Fig. 6-29 Robotic hand showing the electrical posts on the left, and the size of the ro	botic hand
compared with the real human hand on the right	83
Fig. 6-30 PCB board layout of the C-segmentation design implemented on PCB board.	84
Fig. 7-1 Application of dimensionality reduction to brain machine interface	87
Fig. B-1 Procedures of segmentation design for the postures given in table 1. (a)	Actuators
divided into two decoupled groups and actuators {6, 7} are totally coupled. (b) Common
segments for actuators $\{1,2\}$, $\{3,4\}$ and $\{5, 6, 7\}$ are designed. (c) Common set	gments for
independent postures of actuators {3, 5} are designed (d) Minimum number segment	ation found
for each segments	99
Fig. B-2 Hand postures used in the design of the actuator	100
Fig. B-3 Minimum segmentation design for 10-axis array actuators driving robotic har	ıd 102
Fig. B-4 Activation patterns of the Actuator Array. Shaded segments represent the s	egments to
be activated to generate different postures.	103
Fig. B-5 Experimental result with the ten axis actuator array: Normalized displacem	ents of (a)
Axis 7 and (b) Axis 10 for all 14 hand postures.	104
Fig. D-1 Schematic of Circuit Diagram for single C-segment	115
Fig. D-2 Voltage requirement for different MOSFET gates	116
Fig. D-3 Voltage limiting zener diode	116
Fig. D-4 Shadow voltage showing up at the gate of the connected segments	117
Fig. D-5 Blocking diode paired with zener diode	117

Chapter 1. Introduction

In recent years, humanoid robots with over 30 degrees of freedom have been developed. We have seen Honda Asimo, Sony Qrio and Hubo walking, shaking hands, dancing and throwing balls. Even though this current generation of humanoid robots can walk, run and dance, they are still far from the actual human or human-like robots that we see in the movies. Humans have hundreds of muscles that are coordinated to create all different kinds of movements, whereas the current humanoid robots are limited to less than around 40 degrees of freedom. Next generation of robots will have much larger degrees of freedom, will use compliant materials and will have more natural movements. For the next generation of humanoid robots to have large degrees of freedom close to that of a human, a new type of lightweight, compact and high power to weight ratio actuators must be available. Not surprisingly, many researchers are developing new types artificial muscle actuators that have characteristics similar to biological muscles. Material characteristics of these artificial muscle actuators are similar to biological muscles. But for using large number of these actuators, architectural characteristics of the biological muscle as well as the material characteristics are important. The goal of this thesis is to create an effective and efficient method of controlling and coordinating the large degrees of freedom actuation system by exploiting the architectural characteristics of the biological muscles.

1.1. Artificial Muscle Actuators

Biological muscles are lightweight and compact. Large number of muscle can be integrated into a small volume. A human hand for example, has roughly 50 muscles that drive over 21 joints. Biological muscles also have a unique cellular structure that departs from traditional electromechanical actuators in many ways. A muscle consists of a vast number of cellular units or building blocks, i.e. sarcomeres and muscle fibers. Output force and displacement are aggregated effects of the individual building blocks. Force and strain characteristics are tuned to specific loads by combining those cellular building blocks in series and parallel [1]. Rather than using gearing and transmission for load matching, muscles can be tailored to a range of loads, satisfying specific force and displacement requirements. Over the last few decades the materials community has been striving to develop new materials for artificial muscle actuators. Similar to biological muscles, they are compact and lightweight and has a high power to weight ratio. Among others, conducting polypyrrole actuators have shown great promise with respect to strain and stress [2,3,4]. It is still difficult to produce reliable conducting polymer actuators with a long life span, large stress and high strain [5]. However, polypyrrole actuators point in the future direction of artificial muscle actuators. Dielectric elastomers, on the other hand, can create an extremely large strain with excellent power efficiency [5]. Requirements for a preloading mechanism and an extremely high applied voltage have been a bottleneck to this technology. Although these new actuator materials are promising, a number of technical issues including reliability must be overcome before they become usable for broad applications.

Most of the theoretical results to be developed in this thesis will be applicable to diverse actuator materials, including conducting polymers and elastomers as well as shape memory alloy (SMA) and piezoelectric devices. For practical embodiment and demonstration of the concept, we will use SMA since it is a mature actuator material proved to be reliable. Among other artificial muscle actuators, SMA actuators have distinctive advantages in terms of stress and power density. Despite its limitations [5], SMA has the highest stress and superb power density. Many researchers have already developed various robotic devices using SMA actuators [6-10]. One of the major difficulties in controlling SMA actuators is prominent hysteresis and nonlinearities [11]. In the past various control strategies have been developed to cope with the difficulty of SMA actuators [12-16].

1.2. Coordination of multi-axis actuator system

Coordination of multi-axis movements has been an important research issue in both robotics and biomechanics. For example, it is known in biomechanics that the five fingers of a human hand are highly coordinated and coupled in their joint movements [17, 20]. Although 19 joints are involved in a single hand, only a limited number of distinct finger postures are needed for performing daily chores [18]. Based on this fact, most prosthetic arms and robotic hands developed so far have much fewer degrees of freedom, yet they can perform a class of simple tasks [19]. This coupled behavior is also used for recognition of hand motion and gesture understanding. The search space of possible hand postures is significantly reduced by

considering this coupled behavior as constraints [20, 21]. Macroscopic coordinated behavior has been studied by many investigators.

1.3. Thesis goal and overview

The goal of this thesis is to provide a new method for controlling artificial muscle actuators. The new approach will simplify the control of the actuators by exploiting the unique characteristics of muscle-type actuators. Large number of actuators will be controlled with a few number of binary control input signals. Nonlinearity of the material and the hysteresis will be avoided by using a stepper motor like control. Fig. 1-1 shows the overview of the process of designing the actuator system for driving the robotic hand. Basic approach is to use the data captured from a cyber glove to create an actuator system for a robotic hand that can perform the captured postures. The process starts with a gathering of the postures to be performed by the robotic hand. The gathered data is transformed into actuator space since the capture data is in the joint space. This data is used to design the segmentation architecture. The design is than implemented on to the multi-axis actuator system by using a PCB board that has the electrical routing that realizes the architecture in hardware level. This board is attached to the robotic hand. As can be imagined, for different set of postures, a new design can be generated and implemented on to the same robotic hand by replacing the PCB segmentation board with a new one.

The rest of the thesis is organized as follows.

Chapter 2 discusses the coordination of many artificial muscles and introduces the concept of dimensionality reduction in actuation level. This actuation level dimensionality reduction can be used to simplify the design and control of system with large number of actuators.

Chapter 3 introduces a new control method for artificial muscle actuators, which is called Segmented Binary Control. This method simplifies the control of a nonlinear artificial actuator into an On-Off control, similar to stepper motors. With this control scheme, a multi-axis cellular actuator array can be created. The dimensionality reduction concept can be applied to this cellular actuator array. Due to more redundancies present caused by binary constraint, the dimensionality reduction will be more effective when applied to cellular actuator arrays.

Chapter 4 discusses the problem formulation and algorithms to design the segmentation

architecture. The algorithm used is called Nonnegative Matrix Factorization, which takes into account the physical limitations that need to be considered to realize the extracted features. The algorithm will produce a segmentation design that can be directly implemented to an actuator system.



Fig. 1-1 Overview of the Process

Chapter 5 discusses the application of the algorithms for designing an actuator system for a robotic hand. In this chapter, the procedure of gathering data to designing the actuator system to reproduce various grasps is shown. The hand is a very good example of large DOF robotic system with coordinated motions.

Chapter 6 discusses the hardware implementation of the segmented binary control. The segmented binary control is implemented by local activation of the shape memory alloy actuators. Local activation can be achieved either by thermoelectric devices or by joule heating. Actuator system for a robotic hand with 16 DOF is built based on the hardware implementation of the

segmented binary control. Overall robotic hand hardware is presented as well.

Chapter 7 concludes the thesis and presents future research that could be extended from the current research.

Chapter 2. Coordination using artificial muscles

2.1. Introduction

In order to build a dexterous, more human like robot, multi-axis actuation system with large degrees of freedom is required. For example, robotic hands that can create human-like movements require more than 20 degrees of freedom to be packaged in a size of a human arm. Artificial muscle actuators that are lightweight and compact will enable us to build such systems.

There are several issues in the implementation of these artificial muscle actuators for driving systems of large DOF. One issue is the complexity of controlling many DOF. As the number of DOF increases, the complexity of controlling these DOF increases exponentially. It is interesting to look into how biological systems handle the complexity of controlling many muscles to create coordinated motions.

Dimensionality reduction is a mapping of a multidimensional space into a space of lower dimensions. It is used in image analysis, data mining, pattern recognition and other similar tasks that require finding features/basis from a large amount of data. It is a data analysis process rather than a data modeling process. Although many different dimension reduction techniques have been developed and applied in different aspects of robotics, most of their applications are for vision, image analysis or speech recognition, which does not require physical implementation of each features or basis vectors found in the process of dimension reduction.

In this research, implementation of the dimension reduction technique for design of artificial muscle actuator system using C-segments is investigated. Most of the applications of dimension reduction start with a data captured from physical space, but does not realize the results back to the physical space. But for C-segment design problem, the process starts with a data captured from the physical space, and has to be realized in the physical space as well. Realization of the reduced dimensional data in the physical world requires some hard constraints on the values of the components in the reduced dimensional representation.

In this section, the problem of actuator C-segment design is formulated into a dimension reduction problem, and the constraints needed for realization are presented and discussed.

2.2. Background

A central issue in biological motor control is how the central nervous system generates the muscle activity patterns necessary to achieve a variety of behavioral goals. The many degrees of freedom of the musculoskeletal apparatus provide great flexibility but make the control problem extremely complex. In 1967, muscle synergies--coherent activations, in space or time, of a group of muscles--have been proposed as building blocks that could simplify the construction of motor behaviors [22]. Since then many researchers have worked on identifying the existence of muscle synergies. [23-32]

Bizzi et al [23][24][25] developed a new method to extract invariant spatiotemporal components from the simultaneous recordings of the activity of many muscles. They used this technique to analyze the muscle patterns of intact and unrestrained frogs during kicking. They showed that combinations of three time-varying muscle synergies underlie the variety of muscle patterns required to kick in different directions, that the recruitment of these synergies is related to movement kinematics, and that there are similarities among the synergies extracted from different behaviors.

Santello et al[29][30] did a similar research with human hands. According to their research, a wide variety of hand shapes can be characterized as a weighted combination of just two or three main patterns of covariation in joint rotations, or "postural synergies.", since humans have limited ability to independently control the many joints of the hand. They also tried to align muscle synergies with these main postural synergies and describe the form of membership of motor units in these postural/muscle synergies. Seventeen joint angles and the electromyographic (EMG) activities of several hand muscles were recorded while human subjects held the hand statically in 52 specific shapes (i.e., shaping the hand around 26 commonly grasped objects or forming the 26 letter shapes of a manual alphabet). They used principal-components analysis to reveal several patterns of muscle synergy, some of which represented either coactivation of all hand muscles, or reciprocal patterns of activity in the intrinsic index finger and thumb muscles or in the extrinsic four-tendoned extensor and flexor muscles. Although usage of principal component analysis raises question about how they can be actually implemented in biological systems due to their negative values, it is interesting to see that wide variety of hand shapes can be represented by limited number of synergies.

In robotic hand research, several researchers have built robotic hands with reduced number

of actuators [33] [34][36]. Cabas et al.[35] have built a robotic hand that has a single DC motor. With this single DC motor, the three fingered hand can grasp various different objects. This was done using a complicated mechanical device. There are other under actuated fingers and under actuated hands. These robotic hands try to reduce the number of actuators and perform different tasks that normal robotic hands can perform. But using artificial muscle actuators, the limitation in size and weight is reduced, and it becomes easier to increase the number of actuators rather than trying to create complicated mechanisms.

Coordination of multi-axis movements has been an important research issue in both robotics and biomechanics. For example, it is known in biomechanics that the five fingers of a human hand are highly coordinated and coupled in their joint movements [38]-[42]. Although 19 joints are involved in a single hand, only a limited number of distinct finger postures are needed for performing daily chores. Based on this fact, most prosthetic arms and robotic hands developed so far have much fewer degrees of freedom, yet they can perform a class of simple tasks [37]. This coupled behavior is also used for recognition of hand motion and gesture understanding. The search space of possible hand postures is significantly reduced by considering this coupled behavior as constraints [38][39]. Macroscopic coordinated behavior has been studied by many investigators. However, there is no prior work addressing coupled multi-axis behavior of actuators at the building block level.

2.3. Actuator Level Coupling

Material based actuators, such as conducting polymer actuators and shape memory alloy actuators have different characteristics compared to traditional actuators such as DC motors [5][6]. These characteristics should be carefully noted, and should be used in designing the actuator system.

The fundamental characteristic of these material based actuators is the way they create motion. Unlike DC motors, these actuators create motion by shrinking or expanding the material itself. Therefore the displacement or force is created proportional to the amount of activation of these materials. Most other actuators, like DC motors, create motion between two parts, where one part is moved against the other by creating a force between the two parts. But the material based actuators are different. These actuators create motion by heating or charging the material. The amount of the motion or the amount of the force is proportional to the amount

of the material that is being activated. Therefore by controlling the amount of the material that is being activated, we can create different amount of displacements and forces.

We have adapted the concept of synergy – coherent activations, in space or time, of groups of muscles – to simplify the design of a multi-axis cellular actuator array using artificial muscle actuators [22].

The concept of synergy is similar to a concept of transforming the original data into a new coordinate system. By intelligent choice of the coordinate system, the dimension of the coordinate system can be reduced. Fig. 2-1(a) shows a set of data on a three-dimensional space. Methods such as principal component analysis can be used to find new coordinates that can recreate the information with fewer dimensions. The output points can now be represented on a two-dimensional space as shown in the right. An effective reduction of dimensionality is possible only when the data is highly correlated or coupled.

Likewise, for a multi-axis actuator system that performs a highly correlated/coupled task, dimensionality can be reduced by grouping segments of actuators. A co-activated group of segments is referred to as a *C-segment*. *C-segment* is a group of parts of artificial muscle actuator that is activated with the same activation signal.

Fig. 2-1 (b) shows three actuator axes where the output displacement can be plotted on a 3-dimensional space. The same outputs can be generated with two C-segments. These C-segments correspond to the new coordinate system.

As seen from research, there is much coupling in the motions of the human hand [29][30]. The C-segment concept can therefore be applied to reduce the complexity of control. Using a C-segment approach, all of the postures and grasps required for daily chores can be reproduced with just a few control signals;

The simplicity of control and the simplicity of resources needed to drive and control such system are the main benefits of actuator level coupling. The benefit of the simplicity increases exponentially as the degrees of freedom increases. These benefits are similar to the benefits that are gained by compressing a data. Image processing, speech recognition or any data related processing that requires large amount of data can benefit in terms of the amount of channel resources needed to transfer and store the data [44]. Also, it is used for data analysis, such as pattern recognition and speech recognition.



Fig. 2-1 Concept of dimensionality reduction using C-segments

Unlike the biological muscle that has a very efficient method of activating the large number of muscles using chemical reaction, artificial muscle actuator requires electrical energy which in turn requires power electronic circuitry and wires, that can be burdensome. Not just in terms of the number and weight but also in terms of connection and etc. Therefore, reduction of the number of controls is a benefit in the hardware level.

2.4. Fundamental characteristics of Actuator level coupling

One big characteristic of the material based actuator is that we don't have to treat it as a single input single output system, but rather a distributed input, single output system. Even though the actuator equations are written as a single input and single output system, in actual physical system, the area that the input is covering is related to the output of the actuator. In another view point, you can think of this kind of actuator as an actuator where you can add the displacement by adding more actuators in series. Simple connection of the actuator in series and parallel, without complicated gearing or connection can be done to increase the displacement or force. This characteristic is something distinct from DC motors.

2.4.1. Coupling coefficient and coupling theory of the actuator material

Based on the two-dimensional segmentation architecture, fundamental properties of multi-axis coordination using SBC will be analyzed in this section.

Given a class of tasks, correlation of motion among multiple axes is analyzed. Based on the correlation, a multi-axis actuator is segmented in two-dimensional space.

Let y_i be the output displacement of the *i*-th actuator axis and *M* be the number of actuator axes. Consider *M* output displacements, y_1, \dots, y_M , called a "posture", that must be generated by *M* actuator axes. For the sake of simplicity let us normalize each output displacement, y_i , with its maximum stroke so that it is confined within the interval between 0 and 1: $0 \le y_i \le 1$. For performing a complete task, the actuator system must take a series of postures given by

$$0 \le y_i(t) \le 1, \quad 1 \le i \le M \text{ and } t \in T_i, \tag{2-1}$$

where variable t is an index specifying a posture, taking either a discrete or a continuous value in index set T_{J} . In this paper we do not consider dynamic behavior but consider only kinematic movements. The task goal is to take all the postures given by (2-1).

In Segmented Binary Control, the output displacement is proportional to the total length of the actuator material whose state is "ON". If the total ON-state length is normalized with its maximum length so that $0 \le x_i^{ON} \le 1$, then the total ON-state length of the actuator material is the same as the normalized output displacement y_i ; $x_i^{ON} = y_i$. On the other hand, the actuator material length of "OFF" state is the rest of the entire actuator length. Therefore,

$$x_i^{ON} + x_i^{OFF} = 1 \tag{2-2}$$

The internal state of each actuator axis is a distribution of ON and OFF states along the axis. Note that, as long as the total length of actuator material that takes ON-state remains the same, the output displacement of that axis, $x_i^{ON} = y_i$, is the same, regardless of how the states are distributed. Therefore, each axis has a considerable degree of redundancy. The key point in the two-dimensional segmentation design is to exploit this redundancy and find which segments of individual axes can be coupled together for generating all the required postures. The following theory explains the fundamental concept of coupling in the segmentation design, and derives a coupling coefficient matrix to be used in the design.



Fig. 2-2 Coupled and independent segments of m axis array

Consider a subset of actuator axes A_m , as shown in Fig. 2-2. The subset consists of m ($1 \le m \le M$) actuator axes with an equal total length:

$$A_m = \{i_j | 1 \le i_1 < i_2 < \dots < i_m \le M\}$$
(2-3)

If these m axes are to generate "similar" output displacements, some portions of the output displacements can be generated with some coupled segments. The goal of the analysis is to quantify this "similarity" of output postures and find the coupled segments of a proper length that can generate the "similar" part of the postures.

When coupled segments are used, the ON-state length x_i^{ON} consists of the one involved in the coupled segments and the one in the independent segments. As shown in Fig. 2-2, the coupled segments are divided into the ON state of length z_{Am}^{ON} and the OFF-state of length z_{Am}^{OFF} . Likewise each independent strip of segments is divided into Δx_i^{ON} and Δx_i^{OFF} .

$$\begin{aligned} x_i^{ON} &= z_{Am}^{ON} + \Delta x_i^{ON}, \\ x_i^{OFF} &= z_{Am}^{OFF} + \Delta x_i^{OFF}, \quad \forall i \in A_M \end{aligned}$$
(2-4)

The question is to find the size of the coupled segments: $z_{Am}^{ON} + z_{Am}^{OFF}$. The maximum size of the coupled segments depends on tasks. The more the outputs of a given subset of actuator axes are similar to each other, the larger the coupled segments can be. Note that such similarity must be evaluated for the entire set of postures to be generated.



Fig. 2-3 (a) Postures of m axes, (b) All On and OFF states for all the m axes, (c) The common segments

Fig. 2-3-(a) illustrates a series of postures over $T_I = [0, t_f]$. The task goal is to take all the postures $y_i(t)$, $1 \le i \le M$, for all $t \in T_I$. This requires each axis to take the total ON-state length of $x_i^{ON}(t)$ and OFF-state length of $x_i^{OFF}(t)$, as shown in the figure. Consider the *envelope* of these postures, as shown by the shaded area in Fig. 2-2-(b). All the output displacements to be generated with the *m* actuator axes for the entire set of T_I are involved in the shaded area. If this shaded area is narrow in the vertical direction, the postures are similar and thereby they can be generated with a large block of coupled segments.

As shown in Fig. 2-3-(b), the shape of the shaded area can be characterized with two variables: $\ell^{ON}(t) = \underset{i \in Am}{Min} x_i^{ON}(t)$ and $\ell^{OFF}(t) = \underset{i \in Am}{Min} x_i^{OFF}(t)$, where $x_i^{ON}(t)$ is the normalized length of the actuator material at ON-state and $x_i^{OFF}(t)$ is that of OFF-state, determined by the output posture, $x_i^{ON}(t) = y_i(t)$. The height below the shaded area implies that all the *m* actuators have at

least ℓ^{ON} length of ON-state, while the one above the shaded area indicates that all the *m* actuators have at least ℓ^{OFF} length of OFF-state. In other words, all the *m* actuators share $\ell^{ON} + \ell^{OFF}$ length of ON/OFF states. Fig. 2-3-(c) shows the profile of the sum of these lengths $\ell^{ON} + \ell^{OFF}$ and its lower bound for the entire set of T_I :

$$C_{Am} = \underset{i \in T_i}{Min} \left[\underset{i \in Am}{Min} x_i^{ON}(t) + \underset{i \in Am}{Min} x_i^{OFF}(t) \right]$$
(2-5)

The lower bound C_{Am} , shown by the dot-and-dash line in the figure, represents how much the subset of axes A_m can be controlled together for the entire set of *postures* with a block of coupled segments. The coefficient C_{Am} , referred to as a Coupling Coefficient of order *m*, plays a major role in determining how much the multi-axis segments can be lumped together. Now, the following theorem constitutes the principle of multi-axis segmentation.

Theorem Let $x_i^{ON}(t)$ and $x_i^{OFF}(t)$ be, respectively, the normalized lengths of ON-state segments and OFF-state segments that generate the *i*-th actuator displacement of a given posture at $t \in T_i$. Let A_m be a subset of *m* actuator axes and $z_{Am}^{ON}(t)$ and $z_{Am}^{OFF}(t)$ be, respectively, the lengths of ON state and OFF state of coupled segments across the *m* axes in A_m . If the total length of the coupled segments is no larger than the coupling coefficient C_{Am} ,

$$z_{Am}^{ON}(t) + z_{Am}^{OFF}(t) \le C_{Am}$$

= Min [Min $x_i^{ON}(t) + Min x_i^{OFF}(t)$], $\forall t \in T_I$
 $t \in T_I^{i} \in Am^{i} \quad i \in Am^{i}$ (2-6)

then the segment lengths, $x_i^{ON}(t)$ and $x_i^{OFF}(t)$, can be generated by a combination of the coupled segments $z_{Am}^{ON}(t)$ and $z_{Am}^{OFF}(t)$ and independent segments such that:

$$x_i^{ON}(t) = z_{Am}^{ON}(t) + \Delta x_i^{ON}(t),$$

$$x_i^{OFF}(t) = z_{Am}^{OFF}(t) + \Delta x_i^{OFF}(t), \quad \forall t \in T_I, \forall i \in A_m$$
(2-7)

where $\Delta x_i^{ON}(t)$ and $\Delta x_i^{OFF}(t)$ are, respectively, the lengths of the independent segments at ON state and OFF state. The total length of the two is no larger than 1- C_{Am} ,

$$0 \le \Delta x_i^{ON}(t) + \Delta x_i^{OFF}(t) \le 1 - C_{Am}, \quad \forall i \in A_m$$
(2-8)

This theorem gives the upper limit to the length of coupled segments common to all the m axes, and guarantees that, although the coupled segments of length less than C_{Am} are lumped together, a combination of the coupled and independent segments can generate all the postures of the m axes.

The proof of this theorem is given in the Appendix.

Remark 1 Among the independent portion of the *m* axes, $\Delta x_i^{ON} + \Delta x_i^{OFF}$, some axes (less than *m*) may further be coupled with each other. In (2-4), subtracting the length of ON-state coupled segments $z_{Am}^{ON}(t)$ from the original output displacement y_i (= x_i^{ON}) yields the *residual output displacement* to be generated by the independent segments, $\Delta x_i^{ON}(t)$:

$$\Delta y_j(t) = \Delta x_j^{ON}(t) = x_j^{ON}(t) - z_{Ami}^{ON}(t)$$
(2-9)

Treating m' $(2 \le m' \le m)$ axes of these residual output displacements $\Delta x_i^{ON}(t)$ as the original *postures*, the above theorem can be applied again. If the coupling coefficient of m' axes is non-zero: $C_{Am'} > 0$, there exist coupled segments of length $C_{Am'}$ across the m' axes. We call $\Delta y_i(t)$ the *residual posture*. This procedure can be repeated until all the residual output displacements have no coupling, i.e. $C_{Am'} = 0$.

Remark 2 Since the *m*-axis actuators are activated together at the coupled portion of the actuator material, as shown in Fig. 2-2, the actual segmentation design of this coupled portion reduces to a single-axis segmentation problem. Therefore, an efficient segmentation design can be applied to the coupled portion.

Remark 3 The residual postures are not unique. The proportion of $z_{Am}^{ON}(t)$ to $z_{Am}^{OFF}(t)$ is determined uniquely only at $\{t \in T_I | z_{Am}^{ON}(t) + z_{Am}^{OFF}(t) = C_{Am}\}$. At other t's, they are not unique and thereby the residual postures in (2-9) are not unique.

Remark 4 Let m_1 and m_2 be two integers, $M \ge m_1 \ge m_2 \ge 2$, and A_{m1} and A_{m2} be two subsets of axes: $A_{m1} \supset A_{m2} \ne \emptyset$. Then, for an arbitrary set of postures,

$$C_{Am1} \le C_{Am2} \tag{2-10}$$

Namely, the more axes one wishes to lump together, the shorter the maximum length of the coupled segments becomes.

As defined in (5), the coupling coefficient can be computed for various numbers of axes, $2 \le m \le M$. For m = 2, (5) reduces to

$$C_{ij} = \underset{t \in T_i}{Min} \left[Min(x_i^{ON}(t), x_j^{ON}(t)) + Min(x_i^{OFF}(t), x_j^{OFF}(t)) \right], \quad 1 \le i, j \le M$$
(2-11)

Note that when $C_{ij} = 0$, axis *i* and axis *j* cannot be coupled at all, whereas the two actuators are totally coupled if $C_{ij} = 1$. Arranging C_{ij} ($1 \le i, j \le M$) in matrix form yields a *M*-by-*M* Coupling Matrix of order 2.

$$\mathbf{C} = \left\{ C_{ij} \right\} \in \mathfrak{R}^{M \times M} \tag{2-12}$$

This symmetric matrix provides useful insights as to how the multiple axes can be coupled together.

2.5. Summary

This chapter described the concept of actuator level coupling using C-segments. Biological systems use a concept called synergy to coordinate the large number of muscles. Coordination and the coupling is a closely related concept. The characteristics of the coordination have been discussed. Coupling is the characteristic that allows the coordination of the motions and hence reduce the dimensionality of the actuators. In the next chapter I will discuss the cellular actuator array, which is to be used for a hardware implementation of multi-axis actuator system. The cellular actuator array uses a new simple control scheme called Segmented Binary Control.

Chapter 3. Cellular Actuator Arrays

3.1. Introduction

In this chapter, I will present a novel architecture for a multi-axis actuator system that uses artificial muscle actuator. A cellular actuator array is created using this concept of binary control. The cellular actuator array is inspired by the structure of the biological muscle. The biological muscles are segmented, beginning with the sarcomeres and forms whole muscles using this basic component [1]. Similar to this biological structure we propose a cellular actuator array that is composed of small cells of actuators that can be activated either On or Off. The On off control of each cell simplifies the control by avoiding the nonlinearities of the actuator material.

3.2. Background

Over the last few decades the materials community has been striving to develop new materials for artificial muscle actuators. Among others, conducting polypyrrole actuators have shown great promise with respect to strain and stress [2][3][4]. It is still difficult to produce reliable conducting polymer actuators with a long life span, large stress and high strain. However, polypyrrole actuators point in the future direction of artificial muscle actuators. Dielectric elastomers, on the other hand, can create an extremely large strain with excellent power efficiency. Requirements for a preloading mechanism and an extremely high applied voltage have been a bottleneck to this technology. Although these new actuator materials are promising, a number of technical issues including reliability must be overcome before they become usable for broad applications.

Most of the theoretical results to be developed in this paper will be applicable to diverse actuator materials, including conducting polymers and elastomers as well as shape memory alloy (SMA) and piezoelectric devices. For practical embodiment and demonstration of the concept, we will use SMA since it is a mature actuator material proved to be reliable. Among other artificial muscle actuators, SMA actuators have distinctive advantages in terms of stress and power density. Despite its limitations [5], SMA has the highest stress and superb power density. Many researchers have already developed various robotic devices using SMA actuators [45]-[51].

One of the major difficulties in controlling SMA actuators is prominent hysteresis and

nonlinearities. In the past various control strategies have been developed to cope with the difficulty of SMA actuators [52]-[60]. While the majority of those control methods treat a SMA actuator as a single plant with a single input, e.g., an electric current for joule heating, an *alternative* method is being developed to overcome the fundamental difficulty of SMA. Instead of driving a single input current through the entire SMA wire, the alternative method divides the SMA wire into many segments and controls the individual segments in a coordinated manner. This segmented architecture is similar to the structure of muscle actuators illustrated in Fig. 3-1, if the segmented wire elements are treated as building blocks of a muscle. We have developed this segmented control of SMA actuators for single axis applications [67]. Using Peltier effect thermo-electric devices [68], the temperature of each segment along a SMA wire is controlled locally and selectively. Furthermore, exploiting the prominent hysteresis of SMA and coordinating multiple segments, an effective control method has been developed for improving speed of response and reducing power consumption [67].

Digital mechanisms such as binary manipulators are active structures consisting of a number of simple building blocks that are controlled as ON-OFF finite state machines [69], [70]. Collectively, finite-state building blocks can generate reconfigurable structures [71], highly flexible and reliable robots [72], and more [73], [74]. In the past, digital mechanisms have been studied extensively. However, there is no prior work on actuators and actuator arrays consisting of numerous building blocks.

3.3. Segmented Architecture of SMA Actuators / Segmented Binary Control

As of today there is no artificial muscle actuator that consists of billions of tiny building blocks. However, it is feasible to build a small-scale actuator system analogous to the structure depicted in Fig. 3-1. Using shape memory alloy and the segmented control architecture one can build a multi-axis actuator array consisting of a number of segmented units that contribute to the output displacements and forces. This section summarizes the segmented control architecture described in [67], and extends the design to a multi-axis array.



Fig. 3-1 Schematic of multi-axis artificial muscle

3.3.1. Segmented Binary Control

Most SMA drive systems consist of heating the entire length of wire with electric current (Joule heating) and cooling with natural or forced convection. Fig. 3-2-(a) shows a schematic of such a system in which the entire SMA wire is controlled as a single process with a single input, i.e. the electric current. In the segmented architecture, as shown in Fig. 3-2-(b), the SMA wire is divided into many short intervals and their thermo-mechanical states are controlled segment-by-segment. The resultant output displacement is the integral of the strains at the individual segments along the wire.

This segmented system architecture entails local heating and cooling of a SMA wire, which would cause interference between adjacent segments at different temperatures. However, the thermal conductivity of SMA is so small that the heat transfer to an adjacent segment is negligibly small. The thermal conductivity of SMA is only 8 W/mK, approximately 1/30 of aluminum. Considering that the wire diameter is small, heat transfer from one segment to the adjacent segments is negligible. Therefore, local heating and cooling are feasible with small gaps between adjacent segments.

Consider simple ON-OFF controls for individual segments. The bi-stable nature of SMA's phase transition is suitable for ON-OFF control. As shown in Fig. 3-2-(c), each segment is controlled to be either at fully martensite or fully austenite state. Since each segmented unit produces a fixed amount of displacement, the resultant displacement of the entire wire is approximately proportional to the number of the "ON" segments. This control architecture is called Segmented Binary Control or SBC for short.



Fig. 3-2 (a) Traditional SMA heating method and (b) Segmented Binary Control. N is the number of segments and n is the number of austenite segments (c) Phase transition diagram of SMA. Diagram shifts with change in stress.

To control the output displacement of a SMA wire, traditional methods use some *analog* property of SMA, e.g. the ratio of austenite/martensite phase correlated with the strain-temperature characteristics. In contrast, SBC is a *digital* approach. It pushes the phase of specific segments to all austenite phase or all martensite phase, rather than keeping the phase of the entire wire somewhere between the two extremes. This would make the SMA wire insensitive to complex nonlinearities, as shown in Fig. 3-2-(c). It has been shown in [67] that, despite prominent hysteresis and stress-dependent phase transition of SMA, the repeatable and positioning accuracy of SBC are good for diverse loads.

The basic concept of Segmented Binary Control will be useful for other artificial muscle actuators, such as conducting polymer actuators, dielectric elastomers, and piezoelectric actuators. All of these actuator materials have significant hysteresis loops and other nonlinearities. Standard control methods treat the entire material as a single process, although the real system is not a uniform system where each segment of the material may take a diverse state. Rather than relying on the complex analogue characteristics for controlling the entire actuator material, one can consider SBC as an alternative. Dividing the material into small building blocks and controlling the state of each segment in a discrete manner will make the system more

robust and repeatable.

In the following sections, the focus will be placed on high-level segmentation and coordination architecture rather than low-level control of actuator materials. Now that the feasibility and usefulness of SBC have been demonstrated for single-axis SMA actuators, we will explore how those segmented units can be organized or combined to perform a given class of tasks needing multi-axis coordination.

3.3.2. Verification of segmentation through local activation

Unlike the binary manipulators which use modular binary actuators, the cellular actuator array uses local activation to create segments that are activated in a binary manner. Although each segment works as a binary actuator, each segment is physically very next to the adjacent segment.

3.4. Segmentation architecture for multi-axis system

One drawback of using SBC is increased number of control loops and drive systems. Although each control loop regulating the phase of an individual segment is rather simple, many loops are needed. SBC may not be useful nor can practically be justifiable, if it entails many feedback loops for controlling individual segments of each axis. Since the actuator material is digitized under SBC, the resolution of the output displacement depends on the number of segments. If the entire actuator material is segmented equally in length, as in the case of Fig. 3-2-(b), the length of each segment is given by $\ell = L/N$, where L is the length of the wire and N is the number of segments having an equal length. The resolution is then $\eta = \bar{\varepsilon} \cdot \ell$, where $\bar{\varepsilon}$ is target strain, typically 4%. This uniform segmentation is redundant.



Fig. 3-3 Minimum segmentation of single axis

Consider the segmentation architecture shown in Fig. 3-3. Similar architecture has been used

in a digital hydraulic actuator [70]. It consists of *b* segments with different lengths set to be 1, 2, 4, 8, ...2^{*b*-1} of the unit length ℓ . An independent control loop is assigned to each segment with a different unit length. Then the *b* control loops can generate any number of units between 0 and $N = \sum_{i=1}^{b} 2^{i-1} = 2^{b} - 1$ turned on or off. With this segmentation, the number of control loops can be minimized for a single axis SMA actuator. Yet the resolution η is the same as the one segmented equally in length if the total number of segments is 2^{b} -1.

Robotic systems are mostly multi-axis systems. If one uses the same segmentation architecture as the above single axis for multi-axis actuators, say M axes, the resultant number of control loops becomes $b \times M$, which will be too large. It is interesting to note, however, that efficient segmentation architecture exists for multi-axis systems, if coupled behavior among multiple axes is taken into account. Namely, segments in different axes can be connected to the same control loop if both axes tend to move together in a certain range.

Consider *M*-axes of actuator materials, e.g. SMA wires, shown in Fig. 3-4. Bundles of SMA wires are laid on a two-dimensional array of local heating and cooling units. Note that this twodimensional array of units is segmented not only in the longitudinal direction of each bundle of SMA wires but also in the transverse direction. Therefore adjacent SMA wires laid on the same segment are heated or cooled at the same time for *that particular portion* of the wires. This, of course, reduces independence of the adjacent SMA wires to a certain degree, but on the other hand generates a coordinated movement among them. If a given class of tasks contains a significant amount of coordinated motion among multiple axes, the coupled nature of the multiple axes can be exploited in order to reduce the number of independent control loops. The five fingers of a human hand, for example, are highly coupled for performing most daily chores. To perform such a class of given tasks, each actuator axis does not have to move independently, but can be coupled with others. The segmentation architecture illustrated in Fig. 3-4 has the potential to generate coordinated movements among multiple axes. The combination of independent segments and coupled, dependent segments will allow us to move the multiple axes independently as well as in a coupled manner in certain ranges of the workspace.



Fig. 3-4 Two-dimensional segmentation of multi-axis SMA actuator system

Cell of an actuator is a certain amount of actuator that can create certain maximum displacement and force when activated to full of it's capability. By viewing the actuator as a cellular architecture, the control of this type of actuator becomes distributed control, meaning in order to control the displacement of a single actuator; we need to activate the cells of the actuator. Coordinating this cellular architecture is another problem. Since the number of cells will increase drastically, the coordination will be a lot harder. One approach would be to coordinate the activation of different cells, thereby reducing the control of multiple axes into controlling more number of cells. The way we coordinate these cells is by forming a group of cells to be controlled together, called C-segments as described in the previous chapter.

3.5. Summary

In this chapter, a description of cellular actuator arrays that uses Segmented Binary Control has been presented. The cellular actuator array is the basic actuator architecture that is used to create a multi-axis actuator system using the artificial muscle actuator. Basic concept of reducing the dimensionality by grouping the segments together has been presented.

Chapter 4. Algorithms for Generating the C-segment Architecture

4.1. Introduction

As discussed in chapter 2, C-segmentation architecture can be used to reduce the dimensionality of multi-axis actuation system. Especially for a cellular actuator array that uses Segmented Binary Control, larger amount of reduction can be expected. The question now is how we generate this architecture. In this chapter, algorithms for designing the C-segmentation architecture will be presented and discussed.

First I will discuss the constraints and the nature of the problem of generating this design. By investigating the nature of the problem, we will show more clearly, what kind of algorithm is needed, and what kind of constraints need to be considered. I will show that the problem of segmentation design is similar to dimension reduction problem when given trajectories to follow. But there are differences. These differences are the constraints that need to be taken into account. But the basic formulation of the problem can stay the same while adding different constraints.

4.2. Problem Formulation

In order to design the C-segmentation architecture, a set of target posture data is needed. A data glove from Immersion Corporation (CyberGlove[®]) is used to capture the data. The data glove provides 22 joint-angle measurements to transform hand and finger motions into real-time digital joint-angle data. Data from three flexion sensors per each finger is used for the actuator design. In order for the data to be used for actuator design, the gathered joint angle data are first transformed into displacement data of the actuators. Therefore, a single desired posture of a robotic hand with *m* actuators can be represented as an *m*-dimensional vector, $\vec{y_x}$, where the elements of the vector are output displacements of each actuator. The objective of C-segmentation design problem is to recreate the desired outputs with an encoding that small number of elements. A single desired output is to be recreated with two components. One is the C-segment matrix *W*, which in dimension reduction techniques is called features or basis vectors, and another is the encoding matrix *H*, which is equivalent to coefficients or coordinates. Fig.

4-1 shows each components of the analysis. In this paper, the terms C-segment matrix and encoding matrix will be used to refer to the basis matrix and the coefficient matrix for C-segmentation design problem.



Fig. 4-1 Representation of single desired output using C-segment Matrix and a Encoding vector

Each C-segment consists of a certain amount of co-activated parts of each actuator axis. Therefore, a single C-segment can be represented as an *m*-dimensional vector, \vec{w} , where the *i*th element represents the amount of *i*th actuator axis that is to be included in the C-segment.

When *r* number of C-segments is used, these C-segments can be represented as an *m* by *r* matrix, *W*. This matrix is analogous to a set of new coordinate axes, and any posture can be mapped to this new coordinate system. A coordinate that represents a single posture on this coordinate system is the encoding vector, \vec{h} , which represents which C-segments are to be activated in order to reproduce the target posture. Therefore, the process of reproducing a posture using the C-segmentation architecture can be represented as a multiplication of the matrix *W* and encoding vector, \vec{h} .

If *n* desired postures are given, the data set of desired postures can be represented as an *m* by n matrix, *M*, where the column vectors are the desired actuator displacements.

The encoding vectors will now form an r by n matrix, H, where the i^{th} column vector is the encodings needed to reproduce the i^{th} desired posture. The whole process of reproducing n different postures using r C-segments is now formulated into a multiplication of two matrices W and H to reproduce matrix M. Fig. 4-2 shows the matrices and its components.



Fig. 4-2 Mathematical formulation of the C-segment design problem

The problem of finding the C-segmentation architecture design and the encodings needed to reproduce the postures can now be formulated as a matrix decomposition problem, which is to find two matrices W and H that can closely reproduce matrix M.

The problem formulation starts with a given set of outputs that needs to be produced using the actuators. This output can be displacements, or force of actuators, or something that can be directly controlled. Therefore, the segmentation design problem is a data driven problem, and depending on the characteristics of the data, the results can vary a lot. A single output point consists of outputs of *m* actuators, hence can be represented as an *m* dimensional vector, $\vec{y_x}$. When *n* different set of desired output points is given, the whole set of outputs is represented as an number of *m* dimensional vectors. Therefore the desired output can be represented as an *m* by *n* matrix, *M* whose column vector represents each output point.

$$M = \left[\overrightarrow{y_1}, \overrightarrow{y_2}, \overrightarrow{y_3} \dots \overrightarrow{y_n} \right]$$
(4-1)

The problem is to recreate this set of outputs using two parts. One is the segmentation, which shows how much of the each actuator axis will be included in the single segment, and the other is the activation signal or encoding signal, which shows activation level of each segments. By multiplying each segment with the activation signal for each output point, the actual output is created. Our objective is to find this set of segments and activation signals that will be able to reproduce the desired outputs.

The problem of segmentation design is analogous to transformation of coordinate axis. For a set of n points on m-dimensional space, we would like to find r number of vectors in the m-

dimensional space, a set of new coordinate axes that can be used instead of the m orthogonal unit vectors. Mathematically, the problem of finding the segmentation design can be formulated into problem of finding two matrices W and H, which can be multiplied to reproduce the output matrix M. W represents the segmentation design, and H represents the activation signal level. Each segment can be represented as a *m*-dimensional column vector \vec{w} where the *i*th element represents the amount of actuator material of *i*th axis that is to be included in the segment. When *r* number of segments is used, this segmentation design can be represented as *m* by *r* matrix, *W*.

$$W = \left[\overrightarrow{w_1}, \overrightarrow{w_2}, \overrightarrow{w_3}...\overrightarrow{w_r}\right]$$
(4-2)

Fig. 4-3 shows the relationship of the matrix and the actual segment.



Fig. 4-3 W matrix and corresponding design

Each activation can be represented as a *r*-dimensional column vector, \vec{h} , where the *i*th element of the vector represents the activation level of the *i*th segment, $\vec{s_i}$. With *n* number of output points, there will be *n* vectors; hence the activation for all the output points can be represented as *r* by *m* matrix, *H*.

$$H = \left[\vec{h_1}, \vec{h_2}, \vec{h_3} \dots \vec{h_n}\right]$$
(4-3)

The benefit of formulating the problem into matrix decomposition problem is that it will allow us to use the methods used in dimensionality reduction of feature extraction or other mathematical tools for matrix decomposition. Other optimization methods can also be used.

4.3. Constraints in C-segment design

As mentioned earlier, in order to implement the dimension reduction technique for design of

artificial muscle actuator system, the mathematical description should be physically realizable. Muscles are unidirectional actuators. It can apply force only by shrinking. Likewise, artificial muscle actuators are unidirectional, although the direction could be either shrinking or expanding. Therefore the values of the reduced dimensional representations, matrices W and H, should be constrained to have non-negative elements.

The scale of the matrices W and H are also constrained in a unique way for C-segmentation design problems. In general dimension reduction problems, the basis matrix or the feature matrix is a normal matrix and the coefficients or the coordinates can be of any values. But for C-segmentation design problem, the basis matrix or C-segmentation matrix represents the exact length of the actuator material that will be used in the physical world. Therefore, it is not limited to be a normal matrix, but left to represent the exact length needed. The displacement that the actuator can produce is proportional to the length of the actuator material that is activated. The elements of encoding matrix, *H*, should therefore be constrained to be between 0 and 1. It represents the percentage amount of the actuator material that will be activated.

4.3.1. Limited length of sum of rows of W matrix

W matrix represents the actual physical length of each segment. Therefore, the values of the W matrix should be constrained such that they can be implemented physically. First, the sum of each row is the actual length of a single actuator. Actual length should be limited to the maximum value of the actuator. It should be smaller than the maximum length of i^{th} actuator, $A_{i,max}$

$$\sum_{j=1}^{c} W_{ij} < A_{i,\max}$$

$$(4-4))$$

In the NMF algorithm, this constraint is enforced by normalizing the W matrix and multiplying it by the maximum length of the i^{th} actuator, $A_{i,max}$

4.3.2. Constraint on H matrix

The values of elements of H are constrained to be between 0 and 1. The values represent the activation of a C-segment for certain instance. Value of 1 in H matrix represents 100 percent activation, and the value of 0 represent 0 percent activation and any values in between represents the percentage of the activation. By multiplying these values with the W matrix, we get a
percentage of The actual displacement created by the actuator is determined by multiplying a scale As, which represents the actual displacement percentage of an actuator.

4.3.3. Non-Negativity Constraint

For many physical systems, non-negativity is a natural constraint. Algorithm that applies a nonnegative constraint on pattern recognition problem allows a part based learning of an object [75]. When applied on face recognition, it was able to identify parts of the face, such as nose, mouse and so on. But this does not mean that the non-negativity constraint should be enforced on the elements of the matrix, since they are just data representations, not a physical representation of actual face.

However, for the C-segment design problem, the C-segment matrix and the encoding matrix should be physically realized but the negative values are not physically realizable. Therefore, the negative values should be avoided or transformed to non-negative values.

In general, any negative components of the C-segment matrix W or encoding matrix H can be transformed to positive values by shifting the values and subtracting the result with a new matrix V.

$$M = (W + S_W)\Box (H + S_H) - V$$
(4-5)

 S_W is a shifting matrix for the C-segment matrix W and S_H is a shifting matrix for the encoding matrix H. New shifting matrix V is as follows

$$V = W \Box S_H + S_W \Box H + S_W \Box S_H \tag{4-6}$$

For recognition and many other applications, the shifting matrix V does not have any constraints. It is a mere subtraction of values. But for C-segment design problem, there is a hard constraint on the matrix V since this shifting matrix has to be physically realized as well. Shifting can be physically realized by changing the position of the actuator. Instead of placing the actuators such that the outputs are zero when the actuator is not activated, it can be placed such that even when the actuator is not activated, it is at a certain displacement v. The hard constraint is that since this positioning is fixed for each axis, the values of each row of matrix V should be a single value. Therefore, the shifting matrix V should be able to be represented as

$$V = \overline{s_v} \overline{\Box_N} \tag{4-7}$$

where $\overline{s_v}$ is a column vector representing the shifting of each axis, and $\overline{i_N}$ is a row vector with all the elements composed of one. For example, $\overline{i_N} = [111111]$. The shifting matrix V is basically a single vector repeated through out N different postures. The shifted value should not be changed for each posture.

4.3.4. Transformation of negative components

With this constraint in mind, negative components of in the C-segment procedure can be handled as follows:

If the elements of the encoding matrix are negative, it can be shifted with a matrix S_{H} , which is in the form of $\overline{s_H} \square_N$.

$$M = W\Box (H + \overline{s_H} \Box_N) - W\Box \overline{s_H} \overline{\Box_N}$$
(4-8)

The shifting matrix V is in the form of $V = (W \Box s_H) \Box s_N$ and hence can be physically realized. So if only the encoding matrix had negative elements, it can be transformed into physically realizable matrices by shifting.

For the case when the C-segment matrix has negative elements, it is not as straight forward.

$$M = (W + \overline{s_w} \,\overline{B_N}) \Box H - \overline{s_w} \,\overline{B_N} \Box H \tag{4-9}$$

The shifting matrix $V = \overline{s_{w}} \overline{\Box_{N}} \Box H$ is no longer guaranteed to be physically realizable.

One way to circumvent this problem is to separate the positive and negative parts of the basis vectors and form two matrices W_p and W_n . W_p consists of positive components of the W matrix and W_n consists of negative components of the W matrix. Hence $W = W_p + W_n$.

$$M = (W_n + W_n) \Box H \tag{4-10}$$

By shifting the H vector to be multiplied to the negative components to be all negative, the M matrix can be reconstructed as follows.

$$M = W_{p} \Box (H + s_{Hp} \Box_{N}) + W_{n} \Box (H - s_{Hn} \Box_{N}) - (W_{p} \Box s_{Hp} - W_{n} \Box s_{Hn}) \Box_{N}$$
(4-11)

This can be represented in a more compact form,

$$M = W_{pn}\Box(H + s_{Hpn}\Box_N) + (v - W_{pn}\Box_{Hpn})\Box_N$$
(4-12)

where W_{pn} is a concatenated matrix of W_p and W_n and s_{Hpn} is a concatenated matrix of s_{Hp}

and S_{Hn} .

Depending on where the negative components lie in the W matrix, the dimension of the final W_{PN} matrix will be between N and 2N.

If there is no negative component at all, it would be N and if there is N vectors that contain both negative and positive components, it would be 2N. Therefore, shifting the matrices requires additional dimensionality.

Although nonnegative components can be transformed into positive components by shifting, there are other problems that remain. This approach of shifting requires initial positions of the actuators to be at a negative position. When physically realized, this could mean that for the robotic hands, some joints could be at a position that is not feasible for normal human hand. Therefore, although it is possible to transform the negative components to nonnegative components, it is not recommended.

Another way to circumvent this problem is to use set of antagonistic actuators. By adding a set of antagonistic actuators that opposes the current direction, we can implement any negative values that occur in the basis vectors and coefficients. But one big problem of this method is the cancellation of the values. While cancellation of numbers means it's just a cancellation, for actual physical systems, cancellation means that the actuators are pulling each other with the same amount of displacement to be generated and results in zero displacement. Unless this cancellation is intended to increase the stiffness of the actuator, it is very undesirable. Therefore, using antagonistic actuators to accommodate the negative values of the components are highly discouraged.

In this section, I've formulated the problem of C-segmentation design into dimension reduction problem and the constraints needed to physically realize the design were addressed. Means to deal with some of the constraints were provided. Algorithms that can be used for designing the C-segmentation can be developed based on these problem formulation and the constraints.

4.4. Principal Component Analysis

The Principal Component Analysis is a method of reducing multidimensional datasets to lower dimensions. It simplifies the dataset for better analysis. PCA is basically a linear transformation that transforms the data into a new coordinate system such that the greatest variance lies on the first coordinate, the second greatest variance on the second coordinate and so on. It keeps the lower dimension coordinates, and removes the higher order terms, which significantly reduces the amount of information needed to represent the given dataset. Actual algorithm finds the eigenvectors of the covariance matrix of the dataset. The advantage of PCA is that it has a nice closed from solution from linear algebra that is easy to implement. For example, matlab has a function called "princomp" that generates the principal components and the coefficients when given a set of data.

PCA is a representative method of all dimension reduction techniques and it generates negative components. As discussed in the previous chapter, this negativity is a problem when we try to physically realize the design. Negative coordinate means activating the actuator in reverse direction. Most of the artificial muscle actuators, like SMA are unidirectional, like biological muscle, and can be activated to shrink, but not to extend. Therefore, the PCA cannot be directly applied to the design.

Muscles use antagonistic configuration to produce the displacements in both direction. Therefore, by using the antagonistic configuration it is possible to use PCA for the segmentation design. Fig. 4-4 shows an example of designing actuator segmentation from negative values. In antagonistic configuration, negative values can be represented as activation of antagonistic actuator. Therefore, any give coordinate axis can be represented on the actuator space with antagonistic configuration.

Simple Example: $PCA = [1 - 2 \ 3 \ -1]^T$

4 Actuators and 4 Antagonistic Actuator



Fig. 4-4 Example of implementing single component from PCA. Single component is equivalent to single coordinate axis.

The principal components, or the coordinate axes are implemented as C-segmentation, and the coordinates are implemented as encoding signals. The encoding signals also has negative values, hence it should be implemented with antagonistic actuators. In order to use the SBC, the actuators have to be discretized. The problem of discretization is now a one axis problem of C-segmentation design, since each axis can be programmed to create all the encoding signals necessary. Fig. 4-5 shows an example of implementing the encoding signals.

Encoding of [1 2 -3 3 -4 7 -7 4]

To implement this, the Actuator length has to be discretized. So this is now an single axis discretization problem.



Eight Different Activations can be done with five segments



Fig. 4-5 Implementation of the different encodings on the antagonistic actuator. Just like the C-segmentation design, negative values can be treated by using the antagonistic actuator.

Unlike the coordinates generated by using the NMF, the coordinates created by using PCA are orthogonal to each other.

4.5. Nonnegative Matrix Factorization

Non-negative Matrix Factorization (NMF) has been proposed as a tool to find a set of basis functions for representing non-negative data [75]. The notion is particularly applicable to image libraries made up of images showing a composite object in many articulations and poses. When used in the analysis of such data, NMF would find the intrinsic 'parts' underlying the object being pictured. NMF has also been applied to extract invariant spatiotemporal components, or synergies, from the simultaneous recordings of the activity of many muscles. The algorithm is as follows:

The cost function used to quantify the quality of the approximation can be constructed using the square of the Euclidean distance between M and WH. This is lower bounded by zero and becomes zero when M = WH.

$$\|M - WH\|^{2} = \sum_{ij} (M_{ij} - (WH)_{ij})^{2}$$
(4-13)

The following update rules are used to update the W matrix and H matrix iteratively. The Euclidean distance is non increasing under these update rules.

$$W_{ia} \leftarrow W_{ia} \frac{(MH^T)_{ia}}{(WHH^T)_{ia}} \qquad H_{aj} \leftarrow H_{aj} \frac{(W^T M)_{aj}}{(W^T WH)_{aj}}$$
(4-14)

The Euclidean distance is invariant under these updates if and only if W and H are at a stationary point of the distance. The above update law is derived by choosing the step size of the gradient descent method, such that the update law of the gradient descent method becomes a multiplicative one. The proof of convergence is also given in the literature. The update laws have been directly applied to the current problem of designing actuator segmentation.

4.6. Comparison of PCA and NMF

In this section comparison of the results of PCA and NMF for the reconstruction error as the dimension used changes is given. A dataset of 16 different postures of daily chore is used for this analysis. Absolute mean of the joint angle errors and the standard deviation are plotted in Fig.

4-6 for different number of dimensions.



Fig. 4-6 Comparison of mean joint angle error and standard deviation for NMF and PCA results

For the same number of dimensions, the reconstruction error of PCA is lower. Since the error between the NMF and PCA is not that different, NMF method can have less reconstruction error by using one more dimension in the design. Due to the nonnegativity constraint the design

generated with PCA has to be transformed and ends up using more number of dimensions that does not reduce the error. So when compared for the same number of dimensions, NMF method eventually is a better method in terms of reconstruction error. This applies to any method that will generate negative components.

Another main difference of PCA and NMF is that PCA generates orthonormal basis vectors, whereas NMF cannot generate orthogonal basis vectors. PCA assumes orthonormality condition to generate the basis vectors. The true benefit of the ortho-normality assumption is that it makes the solution amenable to linear algebra and there exist decompositions that can provide efficient, explicit algebraic solutions. The distance between two points change in non-orthogonal transformation and it could be a problem for clustering. But for reconstructing the original data in a reduced dimension space, the distance between the points has no effect as long as the reproduction error is the same. And even when the distance changes, this does not directly correspond to energy or effort used in actuator space. If the final representation is the same, than the effort used will be the same. Therefore there is no need to assume an orthonomality condition for C-segmentation design.

4.6.1. Case Study using 6 Dimensions

In order to compare the algorithms, an example of design using a given dataset is given. In this section a case study of designing C-segment design has been proposed that uses three dimensions or three number of controls. Here we assume that the control signal is a continuous signal. Fig. 4-7 shows the results of PCA and Fig. 4-8 shows the results of NMF. The upper graphs show the actual design of the actuator C-segmentation. The results of PCA was generated by using 3 C-segments, but the number of C-segments doubled to 6 due to the transformation of negative components into positive components. Therefore the comparison of errors shown in Fig. 4-9 is a comparison between NMF result that uses 6 dimension against a PCA result that uses 3 dimension. As seen in the actual design, the number of C-segments is both 6. The bottom graphs shows the encodings needed to generate all 16 postures. NMF results shows more sparse encoding, which means that less number of C-segments are activated at the same time to generate different postures when using NMF. This is coherent with the observation that the NMF produces parts based features when applied to image analysis.



Fig. 4-7 Results of using PCA



45



Fig. 4-8 Results of using NMF



Fig. 4-9 Comparison of errors for NMF and PCA

The result of this section shows that by using NMF algorithm, the error is reduced. Although in terms of mathematical dimension the PCA has better accuracy, because of doubling of dimension due to negativity factor, the resultant error is smaller for NMF. Therefore, algorithm with a nonnegative constraint is better for designing C-segments.

4.7. Binary Algorithm for Implementation of SBC

4.7.1. Binary NMF

In this section, additional constraint of segmented binary control is applied to nonnegative matrix factorization. The flow chart of the algorithm with constraints is shown in Fig. 4-12. Multiple iterations are done with different initializations of W and H matrices, since different initializations can lead to different solutions.

The additional constraint of making the H matrix a binary matrix is applied after each update of the W and the H matrix according to the update rules (2). The idea is to filter the H matrix so that the values of the matrix elements go to either one or zero. For this purpose, sigmoid function in Eq. (4-11) is used as a filter.

$$S(x) = \frac{1}{1 + e^{-k(x - 0.5)}}$$
(4-15)

where x is the element of the H matrix, and k is a variable that changes with every update, starting from 5.5 to 100. This variable change increases the slope of the sigmoid function, making it steeper as the update proceeds. At the last update, all the elements of H matrix are either zero or one. The W matrix is normalized such that the largest value of the sum of each axis equals the length of the actuator. This normalization reduces the otherwise multiple solutions that can exist. Several initializations are used to find the best solution.



Fig. 4-10 Sigmoid function for binary constraint

4.8. Improved solution using simulated annealing

Simulated Annealing is a probabilistic approach for finding solution to a global optimization problem. The name simulated annealing comes from the physical process of increasing the temperature and slowly cooling it to reduce the defects of metal. Each step of SA replaces the current solution by a random neighbor, which is chosen with a probability that depends on the parameter called temperature. Similar to the physical system, the probability of current solution being replaced by the random neighbor reduces as the temperature decreases.

Since the current algorithm using NMF converges to a local minimum, SA can improve the solution generated by NMF. In order to apply SA in our algorithm, a random process has to be defined. The random process is to change an element of binary matrix H to either 0 or 1. The element to change to change the value is chosen randomly.



Fig. 4-11 Comparison of solutions found by NMF and applying SA to the solutions

It is not feasible to use SA from the beginning, since the amount of computation required to reach a solution would be too large. Therefore, it is reasonable to use SA after the solution is found using other algorithms, to improve the solution.

Fig. 4-11 shows a graph comparing solutions found by using NMF and using SA. As shown in the graph, the solutions tend to improve by applying SA after the solutions has been found.

Fig. 4-12 shows the overall flow of the algorithm for generating the C-segmentation architecture.



Fig. 4-12 Flow chart of the binary algorithm

4.9. Summary

In this chapter, the problem of designing the C-segment architecture has been formulated as a matrix factorization problem. Since the resultant matrices from the factorization have to be physically realized, there are limitations applied to the characteristics of the matrices. For example, the elements of the matrices have to be nonnegative, since the actuator is a unidirectional actuator and the encoding signal is just a ON-OFF signal. The encoding matrix has to be a binary matrix, with either 0 or 1, since we are using a binary control. Implementing these limitations lead this problem to be NP hard problem, with convergence hard to prove. Solution using mixed-integer programming can be sought, but in this thesis, a suboptimal solution using Nonnegative Matrix Factorization was used for its simplicity to implement.

Chapter 5. Application to an Anthropomorphic Robot Hand

The hand is perhaps the most sophisticated part of a humanoid robot, with many degrees of freedom required to reproduce the diverse postures and grasps that the human hand can perform. Due to limitations caused by the currently implemented actuators, many robotic hands are either too bulky or driven by less actuators, using a coupling mechanism or simplified design[34][35]. Artificial muscle actuators will enable us to increase the number of DOF in robots. Artificial muscle actuators are lightweight, compact, and have large power to weight ratio. Many new artificial muscle actuators are currently being developed [5]

There are several issues in the implementation of these artificial muscle actuators for driving systems of numerous DOF. One of them is the inherent nonlinearity of the artificial actuator material, which makes it hard to control. We have proposed a new approach of controlling artificial muscle actuators, called *Segmented Binary Control* (SBC). Instead of controlling the actuator as a single unit, this method divides the actuator into several segments, and controls each segment using a simple ON-OFF control as discussed in chapter 3.

Another issue is the complexity of controlling many DOF. As the number of DOF increases, the complexity of controlling these DOF increases exponentially. It is interesting to look into how biological systems handle the complexity of controlling many muscles to create motions. We have adapted the concept of synergy – coherent activations, in space or time, of groups of muscles – to simplify the design of a multi-axis cellular actuator array using artificial muscle actuators as discussed in chapter 2.

A single human hand has 19 joints at the five fingers alone. Despite numerous degrees of freedom, many of them are coupled, as mentioned previously. Such coupled motion can be generated effectively with use of multi-axis segmentation architecture. The desired postures are captured using the data glove. Using the desired postures segmentation architecture is designed.

5.1. Data collection of postures

In order to design the C-segmentation architecture, a set of target posture data is needed. A data glove from Immersion Corporation (CyberGlove[®]) is used to capture the data. The data glove

provides 22 joint-angle measurements to transform hand and finger motions into real-time digital joint-angle data. Data from three flexion sensors per each finger is used for the actuator design. The hand postures to be performed by the robotic hand are chosen from the grips used in a standard hand function test, which is used to assess and compare the result of a hand surgery. The eight most common hand grips used in 20 activities of daily living is defined in the sollerman hand function test.



Fig. 5-1 Points of angle measurements in the cyberglove

Fifteen different hand postures based on the MHI's requirement has been gathered and used to generate the C-segmentation design. The overview of how this is applied is shown in Fig. 1-1. The process starts with gathering of 15 hand postures using cyberglove. Because the measured angles of cyberglove does not match one to one with the robotic hand, a transformation has to be done for the measured angles to be suitable for use in robotic hand. Fig. 5-1 shows the placement of the sensors on cyberglove. For better transformation of the angles, some reference postures can be used to get a good comparison of the measured angles and the robotic hand angles.

For MCP, PIP and DIP of the index, middle, ring and pinkie fingers, the measured angles are directly used, except for the case when the angles exceed the limit of the robotic hand. The range of motion for these joints of the robotic hand is from 0 degrees to 90 degrees. Therefore, angles that are smaller than 0 or larger than 90 degrees are projected to the limit of the robotic finger joint.

For Rotation of the thumb, the MCP of the thumb can be used with a scale of 0.8 applied.

This is verified by comparing some of the postures of the robotic hand and the actual hand postures. For Abduction of thumb, the center is set to be at 15 degrees.



Fig. 5-2 Pictures of 15 MHI Hand Grasps being taken while wearing the cyberglove. The graphics in the box shows the captured postures, while the larger pictures are the actual hand.

5.2. Transformation of joint angles to actuator displacement

In order to use the collected data for the actuator segmentation design, the joint angles are transformed into actuator displacements. Fig. 5-3 shows the cross-sectional view of the robotic finger along with the tendons of SMA actuator and bias springs that are connected to the distal phalange (the bone close to the finger tip).



Fig. 5-3 Cross-section of a index finger of the robotic hand: SMA actuator is used as a flexor to bend the finger and SMA spring is used as an extensor to provide bias force that pulls the finger back to full extended posture when the SMA actuator is not activated.

The angle of each joint is related to distance between points A_i and B_i, shown in Fig. 5-3. Given a bending angle of θ_i , the actuator displacement required is the amount of distance change between points A_i and B_i, between the case when the bending angle is 0 and when the bending angle is θ_i . The total actuator displacement is given by sum of these distance changes for all three joints. Using a rotation matrix R_i(θ), the rotated position of A_i, denoted A_i' is R_i(θ)*A_i. The distance between Ai' and Bi, is l_i . Therefore, the total actuator displacement required to produce a bending angle of (θ_1 , θ_2 , θ_3) is given by $L_i - (l_1+l_2+l_3)$, where L_i is the sum of distance between A_i and B_i for all i's. Similar calculation can be done for the actuator controlling the other joints.

There are a few things to note about the design of the robotic finger. An SMA spring is used as an extensor to provide bias force. The finger is in a fully extended posture when the SMA actuators are not activated. SMA actuators and springs run from the end of the finger tip all the way to the end of the multi-axis actuator array. The DIP (distal interphalangeal) joint and the PIP (proximal interphalangeal) joints are coupled and under-actuated, thus there is only one actuator for driving both joints. The actuator, however, drives the DIP joint only after the actuator has driven the PIP joint to its maximum bending angle or when there is something blocking the PIP joint from rotating. More detailed version of the transformation is in Appendix C.

5.3. C-segmentation Design for MHI Grasps

Using the algorithms described in the previous section, C-segmentation is designed. Total of 8 C-segment postures are used to generate 16 desired postures. One C-segment is added to move the abduction joints of four fingers. Fig. 5-4 shows the design of the actual C-segmentation of the actuator. The Y axis represents each actuator, showing total of 16 actuators, and the color coded bar graph represents C-segments. As can be seen from the figure, each actuator axis has 3 to 6 Csegments which shows that each axis is part of 3 to 4 different C-segments.



Fig. 5-4 C-segment design in the actuator

Fig. 5-5 shows the individual activation of each C-segment. Although the robotic hand is designed to perform 16 specific postures, it is capable of 256 different postures in total by using different combinations of C-segment activation. Using this C-segmentation design, all fifteen hand postures can be reproduced. Fig. 5-6 show the picture of actual grasps, the robotic hand posture using the captured data, and generated robotic hand posture using the C-segmentation design. The difference of the grasps is the errors caused by the C-segmentation design shown at the bottom of each figure. The grasp was performed five times for each posture, and averaged before being used for the design. As can be seen from each figure, the error caused by the grasps varies from 0 upto 30 degrees.



Fig. 5-5 Eight C-segment Postures generated from the dataset

Desired	Actual		Desired	Actual
Grasping a beer bottle		Writing with a pen		
Holding a brush		Holding a remote controller		
Holding a cellphone		Holding a small piece		
Grasping a cup		Grapsing a toothpick		



Fig. 5-6 Comparison of desired posture(left) and generated posture(right)

The cause of error is not just due to the actuator C-segmentation design. Because the current robotic hand design has two actuators for each finger, one for the DIP and another for the MCP of the finger, the hand postures captured using the cyberglove cannot be perfectly reproduced. MCP is the joint closest to the palm, and the DIP is the joint farthest from the palm. PIP is the joint between these two joints.

The DIP joint starts turning only after the PIP joint has turned 90 degrees, or when there is a force being applied at the PIP joint, which blocks the PIP joint form turning, thereby allowing DIP joint to turn. Therefore, the error shown in each figure should be considered a conservative value of error, not an exact value.

The error graph on the left shows the error between the captured data and the reproduced

data. The four bar graphs on the left represent Thumb DIP, thumb MCP, thumb abduction and thumb rotation respectively. Three graphs for each finger from index to pinkie represents DIP, PIP and MCP respectively. The two graphs on the right show the angle data of captured grasp and reproduced grasp respectively.

By using these graphs and the solid works models of the captured and reproduced grasps, the quality of the grasps created by using the C-segmentation can be evaluated qualitatively and quantitatively. Fig. 5-7 shows the mean of the error of each joint and the standard deviation. On average, each joint angle has about 10degrees of error. In order to use this robotic hand for actual grasping purpose, the amount of error has to be limited.



Fig. 5-7 mean of the error of each joint created by the actuator with clustering. (below) Standard deviation of the error

Principal component analysis has also been implemented to design the C-segmentation of the actuator array. shows the values of the two principal components. The principal components were generated using the matlab command princomp. The princomp command generates the principal components as well as the zscores, which is the encoding needed to reproduce the dataset. One thing to note while using PCA for the C-segmentation design is that there is a mean value, which is basically the initial position of each joints. All the joints should be initialized at a certain values, in order to use PCA. These initial values can make the assembly of the fingers quite difficult, since initializing each joint to certain angle is not easy. The method using NMF does not require initial angles, so they can all be initialized to a position where all the fingers are opened up. This allows assembly of the fingers to be much less cumbersome.

Using this C-segmentation design, all sixteen hand postures can be reproduced. Fig. 5-6 shows pictures of the actual grasps, the robotic hand posture from the captured data, and recreated robotic hand posture using the C-segmentation. The error in the grasps varies from 0 to 30 degrees. The cause of error is not just due to the actuator C-segmentation design. Since the robotic hand is underactuated, the hand postures captured using the cyberglove cannot be perfectly reproduced. Therefore, the error shown in each figure should be considered a conservative value of error, not an exact value.



Fig. 5-8 Two principal components shown as a robotic hand postures

The two principal components are shown as a robotic hand postures in Fig. 5-8. This figure shows the two hand postures activated from minimum to the maximum, which are equivalent to activating the two C-segments with minimum to maximum.



Fig. 5-9 C-segmentation design of the actuator

Fig. 5-9 shows the actual actuator C-segmentation design. For fair comparison of the errors, the design used total of 6 principal components. One major difference from the design using NMF is that now it has antagonistic actuator as part of the design, increasing the number of actuators to be activated in a controlled manner. Another noticeable difference is that the distribution of the segments is quite dense. For each axis, there are 5 or 6 C-segments. Compared to 3 to 4 C-segments when NMF was used, this complicates the design of the C-segmentation.

Therefore, PCA approach does not generate better design than the NMF approach, in the sense that more C-segments are required even when same total number of groups are used. The error does not seem to decrease that much, rather it increases when PCA is used for the design.

5.4. Summary

In this chapter, algorithm of C-segmentation design has been applied to a set of 15 grasps of daily chores. The results show that there is inherently an error of about 30 degrees in the joint angles. In order to perform actual grasping with these set of postures, the angles should be designed to be smaller than necessary, so the grasps will always be in a force-closure. The errors are due to the fact that a binary control is being used, and also because of the limited number of C-segments being used.

Chapter 6. Hardware Implementation of Cellular Actuator Array and Experimental Results

In this chapter, cellular actuator array that uses segmented binary control is implemented using shape memory alloy actuator wires. Two types of implementation are shown in this chapter. The first one utilizes thermoelectric devices to heat and cool the shape memory alloy actuator.

6.1. Shape Memory Alloy Actuator

Shape memory alloy actuators used in this thesis are Flexinol[®] from Dynalloy Inc.[76] It's a Nickel Titanium based alloy. There are three types of basic forms of shape memory alloy wires. Superelastic, shape memory and actuator wires. Superelastic Nitinol is a very flexible or elastic for of the alloy and has the ability to recover strains to around 8%. It is widely used for cell phone antennae, stents and various other medical implants and tools. It is also used for eyeglass frames that bends easily but returns to its original shape quickly. The transition temperature of superelastic Nitinol is a room temperature. The alloy is at fully austenite phase at room temperature, but when an external force is applied to the alloy, phase change occurs. This phase change is not due to the temperature change, but due to the shifting of the hysteresis curve of the temperature vs. strain curve of the alloy. As the force increases, the curve shifts towards martensite state, changing the compact austenite phase material into skewed martensite phase, creating a large displacement. On the other hand, shape memory nitinol that has a phase transition temperature higher than room temperature, has the ability to be deformed at normal temperature and returns to its original shape when heated. It is used for low cycle or "one time" use applications. Actuator wires are specifically manufactures for high cycle multiple repetitions. Flexinol wire's strain recovery provides outstanding consistency compared to other artificial muscle actuators that deteriorates after few times of use. This is the main reason that shape memory alloy was chosen for the application of cellular actuator array implementing segmented binary control. Fig. 6-1 shows the experimental results of strain temperature graph of 0.015" diameter wire. As the load increases, the graph is shifted to a higher temperature.



Fig. 6-1 Experimental results of a strain-temperature graph of 0.015" diameter SMA actuator wire

6.2. Implementation of SBC using thermoelectric modules

6.2.1. Basic Structure

Thermoelectric modules (TEM), composed of Peltier junctions, are used to rapidly heat and cool the segments of SMA actuators [67]. By changing the direction of the current flow, a TEM can either heat or cool a surface.



Fig. 6-2 Schematic of one segment of an actuator array. Thermoelectric module on a grooved substrate board drives the segment of SMA actuator wire.

Fig. 6-2 shows the basic structure of one segment of an actuator array. TEM with a heat sink on top is attached to a cast acrylic substrate board, and the SMA passes through a groove in the board. A maximum gap of 0.122mm between the wire and the TEM exists, which is sufficient for wire to move smoothly, but small enough for the heat to be transferred to the SMA wire from the TEM. A laser cutter (Epilog legend 24TT) was used to fabricate the acrylic substrate board with grooves. The temperature of the SMA wire segment is controlled by actively heating or cooling the TEM. An average current of 1.1 A is used to heat up a single TEM, and 0.5 A to cool it down. Temperatures are measured using thermocouples (5SC-TT-J-30-36) attached to the TEM and feedback control of the device is done with a simple On-off controller. Detailed experiments with this setup are published in [67]

6.2.2. Single-Axis Setup



Fig. 6-3 Setup for single axis minimum segmentation

A single-axis experimental setup has been built to verify the segmented binary control using TEM. The segmentation architecture shown in Fig. 6-3 provides the minimum number of segments for a single-axis actuator. The apparatus shown in Fig. 6-3 consists of 4 segments, each having a length of 15 mm, 30 mm, 60 mm and 120 mm respectively. These lengths are equivalent to 1, 2, 4, and 8 of the unit length of the module; b = 4 and 1 = 15 mm. TEM from TE Technology, Inc. with a size of 30mm by 5mm (TE-23-1.0-1.3) is used for segment sizes 2, 4, 8 and size of 15mm by 15mm (TE-17-1.4-1.5) is used for the segment size 1. Modules in the same segment were all electrically connected in parallel. The SMA wire used in the experiment was *Flexinol* from *Dynalloy Inc.* with a diameter of 0.305 mm and a transition temperature (A_s) of 70°C. Maximum pull force for each wire is 2kgf [75]. Both ends of SMA wire are crimped with ring terminals. Movable end of SMA wire is attached to a holder, and the holder is attached to a linear potentiometer for displacement measurement and two springs to provide bias force. The unmovable end of the wire is attached to a larger ground board.

This single axis actuator can produce 16 discrete levels of output displacement, 0 through 15; b = 4, $N = 2^{b} - 1 = 15$. Fig. 6-4 shows experimental results of positioning accuracy. The horizontal axis is the commanded number of TEM units, while the vertical axis is the normalized length of output displacement. All the 16 points were visited in 4 different sequences so that each

point was arrived at from different points. Although moved from different points, each of the final positions is in the vicinity of the straight line. This step-wise motion characteristic is similar to that of a stepping motor. Fig. 6-4 shows the experimental result of step displacement experiment.



Fig. 6-4 Result of the unit step displacement experiment

6.2.3. Ten-Axis Actuator Array

Multi-Axis actuator array can be built based on the configuration shown in the previous section. The minimum segmentation layout designed in the previous section for driving a robot hand has been implemented. Fig. 6-5 shows the prototype of 10-axis SMA actuator array consisting of 10 SMA wires, a two-dimensional TEM array, and a grooved substrate board. The SMA wires were crimped at both ends with ring terminals, placed in the grooves of the substrate board, and sandwiched with the TEM array using thermal adhesive tapes, in the same manner as the single axis setup. For this multi-axis array, a special PCB board with slots to place the TEMs was designed to simplify the electrical wiring. Fifty 30 *mm* by 5 *mm* TEMs (TE-23-1.0-1.3) were placed inside the slots and two leads from each TEM were soldered to the PCB board. Connectors designed on one side of the PCB board provided one to one connection with the leads of TEMs. Segmentation design has been implemented by electrically connecting the TEMs

that were to be grouped as a single segment, as well as by using a single heat sink for each segment. Fig. 6-5 highlights the heat sinks placed on the TE modules. Note that the layout of the heat sinks is the same as the segmentation architecture design in the previous section. The total number of control loops for this system is 21.



Fig. 6-5 Top view of ten-axis SMA actuator array prototype. Segmentation architecture is shown with the heat sink design.

6.3. Implementation using Joule Heated SMA

6.3.1. Concept

Segmented Binary Control requires that each SMA actuator wire be divided into several segments. SBC has been implemented with thermoelectric devices in the previous section. Although robust temperature control is possible with thermoelectric devices, they are slow, costly and quite inefficient. Joule heating would improve performance in each of these areas. Joule heating is accomplished by running a current directly through the SMA wire; a segmented design would entail that only a portion of the wire would experience said current. By applying a voltage difference between any two points on the wire, a current will be induced which would in turn heat up the selected segment.



Fig. 6-6 Electrical posts used for Joule heated SBC

As shown in Fig. 6-6, C-segment architecture can be implemented by grouping cells to form a C-segment. Grouping means that all the cells in the same C-segment should be activated by a single input signal.



Fig. 6-7 Forming C-segments

Grouping can be implemented in two different ways, either by activating the coupled segments in series or in parallel. Fig. 6-7 shows an example of grouping the segments in parallel. Each segment is bounded with two ground posts at the end, and Voltage is applied at the center of each segment. Segments in the same C-segment can be grouped by applying same voltage at the segments of the same C-segment. Isolation of multiple segments in each axis is much simpler in this scheme, since the ground posts naturally separates each segment from each other. But the parallel connection of the loads reduces the effective resistance. Since joule-heating already requires relatively large currents (~3 amps) in order to activate the SMA with a reasonable response time, the simultaneous activation of multiple segments in parallel would require substantial currents. Furthermore, since the segments are of different lengths, which correspond to different resistances, each segment would have to be driven by different voltages



in order to obtain equal currents, creating another level of complexity.

Fig. 6-8 Grouping segments in parallel to form a C-segment.

The other option, driving the segments in series, makes it very difficult to isolate the different segments of each individual axis. While the activation of a single C-segment would entail the current following a very specific path, shown as the dashed line in Fig. 6-9, simultaneous activation of multiple C-segments would result in multiple voltage differences within each wire. These voltage differences would serve to not only induce current in the desired paths but would also cause unwanted cross-currents through other C-segments.



The methodology used to isolate each C-segment has been coined Time Sharing PWM. It is implemented by first establishing a path between the electrical contacts of each segment of each axis of a given C-segment. The path is then connected using solid state switches. These switches are then only closed when the associated C-segment is activated. Thus the current is limited to only the desired path, resulting in the desired actuation.



Fig. 6-10 Implementation of C-segment with Joule Heated SBC

PCB board with MOSFET switches, wiring scheme and holes for post connection is designed and fabricated to implement this concept.

6.4. Experimental Results



Fig. 6-11 Experimental Setup of Single Axis Joule-Heated Segmented Binary Control of SMA actuator

In order to verify the Joule-heated segmented binary control of SMA actuator wire, an experimental setup of single-axis actuation was built. It is equipped with load cells to measure

the force being applied and potentiometer connected to a pulley to measure the displacement of the actuator. The Top and the bottom plate of the setup can also be easily switched to accommodate different spacing of the posts for the electrical connection. The smaller spacing of the posts can provide better resolution of the actuation, but it could block the airflow, thereby reducing the cooling time of the actuation. Fig. 6-11 shows the top and side view of the experimental setup. The setup is configured to have two SMA actuator wires in an antagonistic mode. One SMA actuator is connected to a spring which is connected to the load cell and the other end is connected to the pulley. The other SMA actuator is directly connected to the loadcell and the pulley on each side. The pulley is connected to a potentiometer to measure the displacement, and the pulley has another pulley with a gear ratio of 2:1, which can be connected to a load. Weight can be hand on the pulley to change the loading conditions. Electrical fan is attached on the bottom side for the forced cooling. Each fan has a cooling capacity of 3 CFM. There are two plates on top and bottom, which is used as a placement holder for the electrical posts. The electrical posts are made of 0.0625" diameter steel rods. In order to increase the contact between the post and the SMA actuator wire, the posts were not placed in line, but with a spacing of 30mil from the center line. Increasing the contact area is important; hence conductive grease (CW7100 from ITW Chemtronics) was used between the post and the SMA actuator wire. It is important especially for the part of the actuator wire that is close to the nonmoving part, since the nonmoving part of the SMA actuator tends to get hot quicker and breaks easily.



Fig. 6-12 Sensitivity Analysis of SMA actuator at various loading conditions

Fig. 6-12 shows the sensitivity analysis of the SMA actuator at various loading conditions. Due



Fig. 6-13 Experiment of heating up and cooling with various current

Experiments of heating up the wire for 1 second with different amount of currents under different loading conditions were done. Fig. 6-13 shows a sample of how the experiments were performed. Certain current was applied for 1 sec, and the after the wire has cooled down, a current larger by 0.1 Amp was applied. This experiment was done for different loading conditions. As shown in the graphs, the displacement increases until certain amount of current, and starts to saturate. The duration of current applied can be reduced or increased, which would require increase or decrease of amount of current being applied.



Fig. 6-14 Experimental result of activation under different loading conditions with different currents for 1 sec

Fig. 6-14 shows the experimental results under different loading conditions. For each loading condition, the amount of current being applied has been changed from 2.4 amps to 3.6 amps. All of these experiments were done with for 1 sec. 9 inches of shape memory alloy wire was activated. As can be seen from the graph, below activation current of 2.9 amps, the amount of maximum displacement reached were different depending on the load. But above 2.9 amps the maximum amount of displacement is close even with changing loading conditions. The wires used were 0.015" diameter, which has a maximum load of 2kgf. This experiment verifies that the use of single current amplitude for driving the SMA actuator will result in equal displacement regardless of the load on the actuator wire.

Fig. 6-15 is a plot of slope of the displacement shows the slope of the SMA actuator displacement vs. time graph at the end point of current input. This graph shows how sensitive the change of displacement is with more current input right before the current has been stopped. This is a good measure to find the amount of current input needed to reach the temperature that we want. (This will be better explained with a graph of slope) After current of 2.9Amps, the slope

reduces to less than 1mm/sec for all the different load settings. For currents lower than 2.9 Amps, the slope is steeper, which means that the SMA wire has not yet reached the saturation point. Although there is no ideal or perfect saturation point for SMA wires, this graph presents us with a means to figure out a *virtual saturation point*, which is the point where the change of displacement is much smaller.



Fig. 6-15 Slope of the SMA wire under different loading conditions

Cooling time is another important factor of performance for SMA actuator. The cooling time to martensite start temperature is shown in Fig. 6-16. The temperature of cooling to martensite is a temperature where the displacement starts to change. This temperature is meaningful because it is the temperature that starts creating a motion, and it could be important for release of the grasp. As shown in the figure, the cooling time was around 0.6secs when using 3 Amps for activation. The cooling time showed no big difference for different loading conditions.


Fig. 6-16 Cooling Temperature to martensite start temperature



Fig. 6-17 Cooling time to 10% of maximum displacement

Although the cooling time to martensite start temperature is a useful measure to find out when the release can begin, cooling time back to its initial position is also important. The experiments in the Fig. 6-17 show the cooling time to 10% of the actuator's maximum displacement, once the heating stops. The experiments were done with a forced cooling.

6.5. Hardware Implementation of Robotic Hand with Cellular Actuator Array



Fig. 6-18 Solid Works model of Robotic Hand (Exploded View)

The designed actuator system is attached to a robotic hand with five fingers. Each finger has three joints, where two joints, the PIP and the DIP are coupled. The thumb has two joints that bend toward the palm and one joint that rotates the thumb perpendicular to the palm, and one joint that rotates the thumb toward or away from the palm. Fig. 6-18 shows the solid works model of the robotic hand in an exploded view. More detailed description of the hand design is in Appendix C. Two SMA actuators are connected to each finger, one at the end of the finger tip, and another at the end of the bone that is closest to the palm. These SMA actuators are pulled to bend the finger at the PIP and the MCP(metacarpophalangeal) joint, the joint that is closest to the

palm. The PIP joint and the DIP joint is coupled, as mentioned before. As shown in Fig. 9, an SMA spring, that utilizes super-elasticity of SMA, is used to apply bias force, instead of regular springs which would have taken up more space. When the SMA actuator is deactivated, the SMA springs pull the fingers back to its neutral position, which is all opened posture. All the parts of the robotic hand are fabricated with a rapid prototyping machine (FDM) and assembled with a steel rod that connects each joints.



6.6. Implementation with TED driven SMA

Fig. 6-19 Picture of the prototype robotic hand

The actuator array uses the Segmented Binary Control (SBC), which segments an artificial muscle actuator into many independently controlled, spatially discrete volumes, and applies simple On-Off control to each segment. Thermoelectric modules (TEM) are used to rapidly heat and cool the segments of SMA actuators. By changing the direction of the current flow, a TEM can either heat or cool a surface. Fig. 6-20 shows the schematic of one segment of an actuator array. An SMA actuator is sandwiched between a thermoelectric module and a thermally insulating acrylic plate. One issue with the first prototype was that because one large heat sink was used to press down against the SMA actuator wire by applying force, the pressure applied to

each element were uneven and therefore could not guarantee a good thermal contact between the SMA actuator wire and the thermoelectric module. A new design, uses a vinyl foam rubber that is cut out into the size of a thermoelectric module, to apply pressure individually to each thermoelectric module. Acrylic plate is also cut into the size of a single thermoelectric module and they are taped together aligned with each peltier element on the peltier board. The vinly foam rubber has a minimum thickness of 0.03" when pressured down and maximum thickness of 0.12" when no pressure is applied. The elastic characteristic of the foam rubber is used to accommodate any misalignment during the assembly process of the peltier element. The peltier element that is attached to the peltier board, made of PCB board does not always stay flat, nor the height of each element cannot be guaranteed, especially with the leads of the peltier element soldered onto the PCB board. Therefore the individual pressure system is needed. Foam rubber with a acrylic plate is aligned on the aluminum support plate with same spacing between each foam as the peltier elements in the peltier board. One large water cooling heat sink is used, and the flat surface of the heat sink provides a reference plane. The heatsink is pressured towards the SMA actuator wires by set of springs placed between the cover plate and the heatsink.



Fig. 6-20 Schematic of one segment of an actuator array. Thermoelectric module on a substrate board drives a segment of SMA actuator wire.

Fig. 6-21 shows three views of the first prototype actuator system. From the front view, the bed of thermoelectric modules can be seen, which is embedded in a PCB board for easy wiring. Electric connectors are located at one side of the actuator that has one to one electrical connection with all the thermoelectric modules. 12 SMA actuators are sandwiched between transparent acrylic substrate board and thermoelectric modules. One large water-cooling heat

sink is attached at the other side of the thermoelectric modules as can be seen from the back view. SMA springs run on top of the acrylic substrate board. All of the SMA actuators and SMA springs are connected at the bottom plate of the actuator system, tied to a hole on the head of the screw. Screws are used to fine tune the tension of the SMA actuator and SMA spring. Aluminum bars are used to stiffen the acrylic substrate board, and springs between the heatsink and the outside plate apply pressure on to the thermoelectric modules, since it is important to make sure that the SMA wires are firmly in contact with the thermoelectric modules.





The first prototype did not use individual pressure system, which resulted in a slower response of the actuator due to poor contact between the SMA actuator wire and the peltier element.

£ 1000	1	55555	1111	10000 -111	1000	0002	and Sale
5182-1113		1992	SALES COMMANY		71220		and Calif
ANNI DISOLOGY		EDICI I	-EDSO	6040	1050	2040	-1050
1050	NO XONO IS	YUSU	EDSO	1050	1050	EOSO	1050
NEEDER OF COMPANY	BRANDSU BRAN	1010	\$050	1050	1050	1050	1050
10000000000000000000000000000000000000	MESS COST DEE	1050	1050	EOSO	2050	ROCO	1050
1050	1040	6050	EDSC .	EDED	£050	1050	1050
REAL COLORER	FOSO	1050	1000	1050	1050	1050	6050
STREET OF LODIE	1050 C	1050	1203	6050	EOSO	1050	EQSC.
101050 000	1050	1050	1050	EUSU	2050	\$050	EDSO
	STATISTICS.	£050	1050	EDGO	EBSE	£050	A £050
6050	ESCOLUTE.	1050	6050	1050	£050	2050	1050
NAME OF A PARTY OF A P	1050	1050	Sale Concellant	1050	\$050	1050	.* 6050
S DOCT	BARR SILL MARRIE	COLUMN STORE	SHIT TO STORE	0203	1050	6000	1050

Fig. 6-22 Peltier board showing 96 peltier elements connected using a PCB board that is wired and connectors placed on one side.

Fig. 6-22 shows the Peltier board. This board is an assembly of PCB board and 96 peltier elements. The PCB board is slotted with 96 rectangular holes for each peltier element. The PCB board is designed with an electrical connection that routes two wires from each peltier element to one side of the board, thereby allowing all the connectors to be placed on one side of the board. Each peltier element is soldered on to the board after they are put in the slots. This board allowed us to have a large number of peltier elements assembled into a small area without having to worry about the electrical connections.

Fig. 6-23 shows the cover plate for the heat sink, with 8 springs to press down on to the heat sink. This board is assembled onto the frame of the actuator system.



Fig. 6-23 Cover plate for the heatsink and the springs that apply pressure on to the heat sinks



Fig. 6-24 Connection board that routes 96 peltier elements into 24 drivers

A connector board is designed to provide the clustering of 96 elements into 24 clusters. Fig. 6-24 shows the picture of the connector board. The connector board reduces the amount of wiring that needs to come out of the actuator system to the driver board. From this board, cables are connected to the driver. More space and cable saving can be achieved by having driver board integrated with this board. For current design, 48 wires from this board are connected to the driver board.



- A: Pressure elements and SMA actuator wires
- B: Petleir Board attached
- C: Heatsink attached
- D: Spring and Support Plate attached

Fig. 6-25 Procedure of assembly of peltier board and heat sink on to the actuator system.

Fig. 6-25 shows the procedure of assembly of peltier board and heat sink on to the actuator system for the second prototype. The second prototype was built with a new feature, where each element is pressured independently, thereby providing a good thermal contact between the SMA actuator and the peltier elements. The peltier board is attached to the one side of the actuator system where SMA actuator passes by, against array of foam rubbers designed to provide individual pressure to each peltier elements. Water cooling heat sink is attached, and a cover plate with springs is attached.

Α



B



С



A: SMA spring side B: Connector board attached

C: Connector board closed view

Fig. 6-26 Other side of the robotic hand system.

Fig. 6-26 shows the front side of the actuator system. On this side you can see the SMA springs. The electrical connectors from the Peltier board passes through the slots of the support plate and they are reachable from the front side of the actuator system. Customized cables are used to connect all the wires to the connector board, which then electrically merges the 96 peltier elements into 24 clusters. The current implementation has 24 clusters, where each cluster can move each joint independently. The segmentation design presented on the previous chapter is not

yet implemented on to the actuator system. But this connector board can be redesigned and manufactured easily. Any new segmentation design can be implemented by redesigning this connector board.



B: Pressure elements close up

C: Tension adjusting mechanism

Fig. 6-27 Features of the actuator system

Fig. 6-27 shows some of the features of the actuator system. Fig. 6-27(A) shows the connection between the SMA actuator wire and the Kevlar wires. In order to maintain a strong connection and also take up small volume, the SMA actuator wires are first crimped to form a loop where Kevlar wires can pass through. The Kevlar wires are tied on to the SMA actuator wire on this loop. Fig. 6-27(B) shows the picture of array of foam rubber and acrylic plates attached on to the support plate. They are aligned with the Peltier elements on the Peltier board. Small slots on the foam rubber are for thermocouples to pass through. Thermocouples can be attached through this hole, on to the peltier elements if needed. Fig. 6-27(C) shows array of screws with a hole on the head. These screws are used to adjust the tension of the SMA actuator wires and SMA spring wires. The distance between the screws were carefully chosen to allow socket heads to go in without interference. The tension can be easily adjusted by holding on to the nuts using a socket and turning the screws to adjust the tension with a screw driver. After the tension is adjusted, the screw is held in place with a screw driver and nuts are tightened with a socket. The SMA actuators used in this system are 0.015" diameter flexinol from Dynalloy inc., and the SMA springs used are 0.013" diameter Nitinol wires from Small parts inc.

6.7. Implementation using Joule Heated SMA

The robotic hand has 20 DOF total with 4 joints per each finger. The total number of actuators is 32, with 16 for flexor and 16 for extensor. The C-segment design is implemented on the flexors and the extensors are used to just open each finger. As shown in the Fig. 6-28, the PCB board with C-segmentation design is attached at the front of the hand and the extensor actuators are all driven by the drive amplifiers at the side of the hand. A Travel limiting compliance is used to keep all the wires taught so that the contact between the SMA wires and the electrical posts are maintained at all times. Electrical fan is attached at the back of the hand to cool the SMA actuators.



Fig. 6-28 View of the final assembled robotic hand





Fig. 6-29 Robotic hand showing the electrical posts on the left, and the size of the robotic hand compared with the real human hand on the right.

As seen in Fig. 6-29, the C-segmetation design is implemented on an acrylic boards that hold the electrical posts in place before the PCB board is attached to the robotic hand. Two acrylic boards that are 6.35mm apart hold the electrical posts in place. Each post is about 15mm long and is cut from a long stock of 1.588mm diameter steel rod.

The PCB board layout of the circuit for creating a C-segment is shown in Fig. 6-30. Total of 34 IRF8910 MOSFETs are used in the board. Instead of using zener diodes at all the stages of the resistor, based on the load of each resistor, some of the loads can be grouped to be connected with a single zener diode. Cathode of the zener diode can be connected to the gate of multiple MOSFETs while the anode is connected to the load that is has the lowest voltage. Total of and 28 zener diodes are used in the board.



Fig. 6-30 PCB board layout of the C-segmentation design implemented on PCB board.

The green holes represent the contacts with the electrical posts. The contact between the electrical post and the PCB board are mostly done without extra soldering. This is because of the natural misalignment that exists between the PCB board and the acrylic boards that holds the electrical posts in place. Due to this misalignment, pressure is applied between the electrical post and the holes which create good electrical contacts. In addition, by adding conductive grease in the holes before inserting the posts, more robust electrical contacts are achieved. This allows easy assembly of the posts on to the boards.

The Video of the robotic hand in action can be seen at http://web.mit.edu/kyujin/www.

6.8. Summary

This chapter discussed the hardware implementation of the segmented binary control. The segmented binary control is implemented by local activation of the shape memory alloy actuators. Local activation can be achieved either by thermoelectric devices or by joule heating. Thermoelectric device can provide an accurate control of the temperature, but the size and the

weight of the overall system is increased. By implementing electrical rods in conjunction with a travel-limited compliance mechanism, joule heated Segmented Binary Control was realized. Actuator system for a robotic hand with 16 DOF is built based on the hardware implementation of the segmented binary control. The weight of the system is 780grams including the electrical fan and the C-Segmentation board with total of 32 SMA actuators in the system.

Chapter 7. Conclusion

In this thesis, a biologically inspired design and control of artificial muscle actuator system is presented. While the conventional approach of using the artificial muscle actuators is controlling the actuator as a whole and individually, this thesis presented a muscle-like architecture where the artificial muscle actuators are controlled by segments in a coordinated manner. This proposed approach simplifies the design and control of the artificial muscle actuator system with vast degrees of freedom. This concept was applied to an actuator system for driving a robotic hand. Data from various hand postures are collected using data glove and used in creating the C-segment design that is capable of performing the given postures. Problem of designing the C-segment was formulated as matrix factorization. Nonnegative Matrix Factorization with binary filter was used to design the C-segments. This approach does not guarantee optimal solution, but the result is produced in a reasonable amount of time with a suboptimal solution. An electrical connection scheme was developed to implement the actuator design to joule heated SMA actuators.

A lightweight robotic hand using a joule heated Segmented Binary Control has been built with shape memory alloy actuators. The robotic hand has 16 controlled DOF and 32 independent SMA axes and the total weight of the robotic hand system was less than 800 grams. 16 different grasping postures were successfully recreated by using only 8 C-segments. Due to the error in the postures, compliant fingertips are needed to perform actual grasping tasks. Theoretically, 256 different postures can be generated with 8 C-segments; this shows that a robotic hand with simple binary control can perform various tasks needed for a robotic hand.

7.1. Future Areas of Research

7.1.1. Brain Machine Interface

Recently, brain machine interface has drawn lots of attention of science community. The idea of being able to control a machine by just thinking about it is a fascinating idea. One of the problems of the current brain machine interface is that the resolution of the extracted signal is very low.



Fig. 7-1 Application of dimensionality reduction to brain machine interface

By using the concept of C-segment, gap between the reduced dimension of the actuator system and the large dimensions of the robotic hand can be reduced. Fig. 7-1 shows a schematic of a brain machine interface system used to control a robotic hand. The current state of the art can extract a 2-Dimensional signal from the brain machine interface that could be used to control a cursor up down left and right. With this 2-Dimensional signal, multi-fingered robotic hand cannot be cannot be controlled with a full DOF. Therefore the approach presented in this thesis can be considered as one of the candidate technologies to control a large DOF robotic hand with few number of control signals.

7.1.2. Application to new artificial muscle actuators

Although the this thesis applies segmented binary control has been applied to shape memory alloy actuators, a lot of current artificial muscle actuators are used as ON-OFF actuators as well, due to its nonlinearity and characteristics that makes it hard to control. By applying the Segmented Binary Control and Cellular Actuator Array architecture to other artificial actuators, more elaborate tasks can be performed with the current artificial muscle technologies that are still in its infancy.

7.1.3. Optimization of the design

A new C-segmentation design algorithm that optimizes the design based on various parameters, such as the efficiency of activation or the importance of different grasps can be developed for

various applications.

7.1.4. Trajectory following using desired postures

The current experimental results shows performance of static hands, but in order to create trajectory, more number of postures can be added, which can be used as the intermediate postures to perform a trajectory. This is similar to the finding that grasps are normally followed by a pregrasp, which is a grasp that is performed right before performing the actual grasp. Therefore, more experimental design with number of grasps for pregrasping will allow trajectory following.

REFERENCES

- [1] R. L. Lieber, "Skeletal muscle is a biological example of a linear electroactive actuator," in *Smart Structures and Materials 1999: Electroactive Polymer Actuators and Devices*, pp.19-25, 1999.
- [2] S. Hara, T. Zama, W. Takashima, and K. Kaneto, "Free-standing polypyrrole actuators with response rate of 10.8% s⁻¹," *Synthetic Metals*, vol. 149, pp. 199-201, 2005.
- [3] S. Hara, T. Zama, W. Takashima, and K. Kaneto, "Gel-like polypyrrole based artificial muscles with extremely large strain," *Polymer Journal*, vol. 36, pp. 933-936, 2004.
- [4] P. A. Anquetil, D. Rinderknecht, N. A. Vandesteeg, J. D. Madden, and I. W. Hunter, "Large strain actuation in polypyrrole actuators," in *Smart Structures and Materials 2003: Electroactive Polymers Actuators and Devices, Proceedings of the SPIE*, pp.380-87, 2004.
- [5] J. D. W. Madden, N. A. Vandesteeg, P. A. Anquetil, P. G. A. Madden, A. Takshi, R. Z. Pytel, S. R. Lafontaine, P. A. Wieringa, and I. W. Hunter, "Artificial muscle technology: Physical principles and naval prospects," *IEEE Journal of Oceanic Engineering*, vol. 29, pp. 706-728, 2004.
- [6] I. W. Hunter and S. Lafontaine, "A comparison of muscle with artificial actuators," in *Technical Digest of the IEEE Solid-State Sensor and Actuator Workshop*, pp. 178-185, 1992.
- [7] Ikuta, K., Tsukamoto, M., and Hirose, S., "Shape Memory Alloy Servo Actuator System with Electric Resistance Feedback and Application for Active Endoscope," *International Conference on Robotics and Automation*, Vol. 1, pp.427-430, 1988.
- [8] K. K. Safak and G. Adams, "Modeling and simulation of an artificial muscle and its application to biomimetic robot posture control," *Robotics and Autonomous Systems*, vol. 41, pp. 225-243, 2002.
- [9] S. Dilibal, E. Guner, and N. Akturk, "Three-finger SMA robot hand and its practical analysis," *Robotica*, vol. 20, pp. 175-180, 2002.
- [10] B.-H. Park, M. Shantz, and F. Prinz, "Scalable rotary actuators with embedded shape memory alloys," in Smart Structures and Materials 2001- Smart Structures and Integrated Systems, 2001.
- [11] Otsuka, K., Wayman, C., "Ti-Ni shape memory alloys," Shape Memory Materials, Cambridge University Press, pp 49-96, 1998
- [12] J. L. Pons, D. Reynaerts, J. Peirs, R. Ceres, and H. VanBrussel, "Comparison of different control approaches to drive SMA actuators," in *Proceedings of International Conference on Advanced Robotics*, pp.819-824, 1997.
- [13] G. V. Webb, D. C. Lagoudas, and A. J. Kurdila, "Hysteresis modeling of SMA actuators for control applications," *Journal of Intelligent Material Systems and Structures*, vol. 9, pp. 432, 1998.
- [14] D. Grant, "Accurate and Rapid Control of Shape Memory Alloy Actuator", Thesis of degree of PhD, McGill University, 1999.
- [15] M. H. Elahinia and H. Ashrafiuon, "Nonlinear control of a shape memory alloy actuated manipulator," *Trans. ASME. J. Vib. Acoust.*, vol. 124, pp. 566-75, 2002.
- [16] Song, "Precision tracking control of shape memory alloy actuators using neural networks and a sliding-mode based robust controller," *Smart Materials and Structures*, pp. 223-231, 2003.

- [17] C. E. Lang and M. H. Schieber, "Human Finger Independence: Limitations due to Passive Mechanical Coupling Versus Active Neuromuscular Control," *Journal of Neurophysiology*, vol. 92, pp. 2802-2810, 2004.
- [18] C. Sollerman and A. Ejeskar, "Sollerman hand function test. A standardised method and its use in tetraplegic patients," *Scand J Plast Reconstr Surg Hand Surg*, Vol.29, pp. 167-176, June 1995.
- [19] R. F. ff. Weir, "Design of artificial arms and hands for prosthetic applications," in *Standard Handbook of Biomedical Engineering & Design*. New York: McGraw-Hill, 2003, pp. 32.1-32.61.
- [20] T. E. Jerde, J. F. Soechting, and M. Flanders, "Biological constraints simplify the recognition of hand shapes," *IEEE Transactions on Biomedical Engineering*, vol. 50, pp. 265-269, 2003.
- [21] J. Lee and T. L. Kunii, "Model-based analysis of hand posture," *IEEE Computer Graphics and Applications*, vol. 15, pp. 77-86, 1995.
- [22] Berstein N. The Coordination and Regulation of Movements, Oxford, UK: Pergamon, 1967
- [23] A. d'Avella, P. Saltiel, and E. Bizzi, "Combinations of muscle synergies in the construction of a natural motor behavior," Nature Neuroscience, vol. 6, pp. 300, 2003.
- [24] F. Mussa-Ivaldi, S. Giszter, and E. Bizzi, "Linear Combinations of Primitives in Vertebrate Motor Control," PNAS, vol. 91, pp. 7534-38, 1994.
- [25] P. Saltiel, K. Wyler-Duda, A. D'Avella, M. C. Tresch, and E. Bizzi, "Muscle Synergies Encoded Within the Spinal Cord: Evidence From Focal Intraspinal NMDA Iontophoresis in the Frog," J Neurophysiol, vol. 85, pp. 605-19, 2001.
- [26] V. C. K. Cheung, A. d'Avella, M. C. Tresch, and E. Bizzi, "Central and Sensory Contributions to the Activation and Organization of Muscle Synergies during Natural Motor Behaviors," J. Neurosci., vol. 25, pp. 6419-34, 2005.
- [27] T. S. Buchanan, D. P. Almdale, J. L. Lewis, and W. Z. Rymer, "Characteristics of synergic relations during isometric contractions of human elbow muscles," J Neurophysiol, vol. 56, pp. 1225-41, 1986.
- [28] J.-J. E. Slotine and W. Lohmiller, "Modularity, evolution, and the binding problem: a view from stability theory," Neural Networks, vol. 14, pp. 137, 2000.
- [29] M. Santello, M. Flanders, and J. F. Soechting, "Postural Hand Synergies for Tool Use," J. Neurosci., vol. 18, pp. 10105-15, 1998.
- [30] M. Santello, M. Flanders, and J. F. Soechting, "Patterns of Hand Motion during Grasping and the Influence of Sensory Guidance," J. Neurosci., vol. 22, pp. 1426-35, 2002.
- [31]E. J. Weiss and M. Flanders, "Muscular and Postural Synergies of the Human Hand," J Neurophysiol, vol. 92, pp. 523-35, 2004.
- [32] S. A. Winges and M. Santello, "Common Input to Motor Units of Digit Flexors During Multi-Digit Grasping," J Neurophysiol, vol. 92, pp. 3210-20, 2004.
- [33] A. Bicchi, "Hands for dexterous manipulation and robust grasping: a difficult road toward simplicity," Robotics and Automation, IEEE Transactions on, vol. 16, pp. 652-662, 2000.
- [34] N. Fukaya, S. Toyama, T. Asfour, and R. Dillmann, "Design of the TUAT/Karlsruhe humanoid hand," in *IEEE International Conference on Intelligent Robots and Systems*, 2000, pp.1754-1759.
- [35] R. Cabas and C. Balaguer, "Design and development of a light weight embodied robotic hand activated with only one actuator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp.2369-2374.

- [36] T. Laliberte and C. M. Gosselin, "Simulation and design of underactuated mechanical hands," *Mechanism & Machine Theory*, vol. 33, pp. 39, 1998.
- [37] T.Iberall, G. Sukhatme, D. Beattie, and G. Bekey, "Control philosophy and simulation of a robotic hand as a model for prosthetic hands," in Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems., 1993
- [38] T. E. Jerde, J. F. Soechting, and M. Flanders, "Biological constraints simplify the recognition of hand shapes," *IEEE Transactions on Biomedical Engineering*, vol. 50, pp. 265-269, 2003.
- [39] J. Lee and T. L. Kunii, "Model-based analysis of hand posture," *IEEE Computer Graphics and Applications*, vol. 15, pp. 77-86, 1995.
- [40] C. E. Lang and M. H. Schieber, "Human Finger Independence: Limitations due to Passive Mechanical Coupling Versus Active Neuromuscular Control," *Journal of Neurophysiology*, vol. 92, pp. 2802-2810, 2004.
- [41] J. Lin, Y. Wu, and T. S. Huang, "Modeling the constraints of human hand motion," presented at Proceedings Workshop on Human Motion, 7-8 Dec. 2000, Los Alamitos, CA, USA, 2000.
- [42] Y. Wu and T. S. Huang, "Capturing articulated human hand motion: a divide-and-conquer approach," presented at Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, 1999.
- [43] C. Sollerman and A. Ejeskar, "Sollerman hand function test. A standardised method and its use in tetraplegic patients," Scand J Plast Reconstr Surg Hand Surg, Vol.29, pp. 167-176, June 1995.
- [44] D. Salomon, Data compression: The Complete Reference, 3rd ed. New York: Springer, 2004.
- [45] J. N. Sarris and N. A. Aspragathos, "Development of a simple anthropomorphic robot hand using shape memory alloys," in IEE Colloquium on Innovative Actuators for Mechatronic Systems, 1995.
- [46] K. Yang and C. L. Gu, "A novel robot hand with embedded shape memory alloy actuators," in Proc. Institution of Mechanical Engineers, Part C (Journal of Mechanical Engineering Science), vol. 216, pp. 737-45, 2002.
- [47] M. Bergamasco, F. Salsedo and P. Dario, "Shape Memory Alloy Micromotors for Direct-Drive Actuation of Dexterous Artificial Hands," Sensors and Actuators, vol. 17, pp. 115-119, 1989.
- [48] R. D. Howe, D. A. Kontarinis, and W. J. Peine, "Shape memory alloy actuator controller design for tactile displays," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 3540-3544, 1995.
- [49] K. Kuribayashi, "Millimeter size joint actuator using shape memory alloy," in *IEEE Conference on Micro Electro Mechanical Systems*, pp.139-144, 1989.
- [50] M. J. Mosley and C. Mavroidis, "Experimental Nonlinear Dynamics of a Shape Memory Alloy Wire Bundle Actuator," *Journal of Dynamic Systems, Measurement, and Control*, vol. 123, pp. 103-112, 2001
- [51] D. Reynaerts and H. Van Brussel, "Design aspects of shape memory actuators," *Mechatronics*, vol. 8, pp. 635-656, 1998.
- [52] Otsuka, K., Wayman, C., "Ti-Ni shape memory alloys," Shape Memory Materials, Cambridge University Press, pp 49-96, 1998

- [53] J. L. Pons, D. Reynaerts, J. Peirs, R. Ceres, and H. VanBrussel, "Comparison of different control approaches to drive SMA actuators," in *Proceedings of International Conference on Advanced Robotics*, pp.819-824, 1997.
- [54] G. V. Webb, D. C. Lagoudas, and A. J. Kurdila, "Hysteresis modeling of SMA actuators for control applications," *Journal of Intelligent Material Systems and Structures*, vol. 9, pp. 432, 1998.
- [55] D. R. Madill and D. Wang, "Modeling and L2-stability of a shape memory alloy position control system," *IEEE Transactions on Control Systems Technology*, vol. 6, pp. 473-481, 1998.
- [56] R. B. Gorbet, K. A. Morris, and D. W. L. Wang, "Passivity-based stability and control of hysteresis in smart actuators," *IEEE Transactions on Control Systems Technology*, vol. 9, pp. 5-16, 2001.
- [57] D Hughes, J.T. Wen, "Preisach Modeling of Piezoceramic and Shape Memory Alloy Hysteresis", *Smart Materials and Structures*, 6, June, 1997, pp. 287-300.
- [58] D. Grant, "Accurate and Rapid Control of Shape Memory Alloy Actuator", Thesis of degree of PhD, McGill University, 1999.
- [59] M. H. Elahinia and H. Ashrafiuon, "Nonlinear control of a shape memory alloy actuated manipulator," *Trans. ASME. J. Vib. Acoust.*, vol. 124, pp. 566-75, 2002.
- [60] Kumagai, "Neuro-fuzzy model based feedback controller for shape memory alloy actuators," Proceedings of SPIE, v 3984, pp. 291-9, 2000.
- [61] Carl F. Ruoff, "Memory Metal Actuator," U.S. Patent 4,553,393, November 19, 1985
- [62] K. J. De Laurentis, A. Fisch, J. Nikitczuk, and C. Mavroidis, "Optimal design of shape memory alloy wire bundle actuators," in Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2363-2368, 2002.
- [63] Lu, X., Grant, Danny, Hayward, V., "Design and Comparison of High Strain Shape Memory Actuators," in Proc. IEEE Int. Conf. on Robotics and Automation, pp. 260-267, 1997.
- [64] M. Branciforte, R. Caponetto, and L. Fortuna, "RD-CNN for driving a 2-D conveyor belt via memory shape alloys," Control Engineering Practice, vol. 11, pp. 687-695, 2002.
- [65] C. Niezrecki, D. Brei, S. Balakrishnan, and A. Moskalik, "Piezoelectric actuation: State of the art," Shock and Vibration Digest, vol. 33, pp. 269-280, 2001.
- [66] N.A. Sherwani, "Algorithms for VLSI physical design automation," Kluwer Academic Publishers, pp.1-27, 1993
- [67] B. Selden, K. J. Cho, H.H. Asada, "Segmented shape memory alloy actuators using hysteresis loop control," accepted for publication in *Smart Materials and Structures*
- [68] A. R. Shahin, P. H. Meckl, J. D. Jones, and M. A. Thrasher, "Enhanced cooling of shape memory alloy wires using semiconductor `heat pump' modules," *Journal of Intelligent Material Systems and Structures*, vol. 5, pp. 95, 1994.
- [69] G. S. Chirikjian, "Binary paradigm for robotic manipulators," in *Proc. IEEE Int. Conf. on Robotics and Automation*, pp.3063-69, 1994.
- [70] A. H. Delmege and M. J. Tremblay, "Hydraulic digital actuator converts binary inputs to linear motion," *Control Engineering*, vol. 12, pp. 69-70, 1965.
- [71] V. A. Sujan and S. Dubowsky, "Design of a lightweight hyper-redundant deployable binary manipulator," *Transactions of the ASME. Journal of Mechanical Design*, vol. 126, pp. 29-39, 2004.
- [72] M. Hafez, M. D. Lichter, and S. Dubowsky, "Optimized binary modular reconfigurable robotic devices," *IEEE/ASME Transactions on Mechatronics*, vol. 8, pp. 18, 2003.

- [73] Y. Y. Kim, G. W. Jang, and S. J. Nam, "Inverse Kinematics of Binary Manipulators by Using the Continuous-Variable-Based Optimization Method," *IEEE Transactions on Robotics*, vol. 22, pp. 33-42, 2006.
- [74]G. S. Chirikjian and D. S. Lees, "Inverse kinematics of binary manipulators with applications to service robotics," in *IEEE International Conference on Intelligent Robots and Systems*, pp.65-71, 1995.
- [75] D. Lee and S. Seung. Learning the parts of objects by non-negative matrix factorization. Nature, 401:788-791, 1999.
- [76] Flexinol technical data. Retrieved Jan. 28, 2006, from http://dynalloy.com/TechnicalData.html
- [77] T. Okada, "Analysis of Finger Motion and Hand tasks," Biomechanism 3, University of Tokyo press. pp. 133-144, 1975.
- [78] M. R. Cutkosky, "On grasp choice, grasp models, and the design of hands for manufacturing tasks," IEEE Transactions on Robotics and Automation, vol. 5, pp. 269-79, 1989.
- [79] J. K. Salisbury and J. J. Craig, "Articulated hands Force control and kinematic issues," International Journal of Robotics Research, vol. 1, pp. 4-17, 1982.
- [80] S. C. Jacobsen, J. E. Wood, D. F. Knutti, and K. B. Biggers, "UTAH/MIT DEXTROUS HAND: WORK IN PROGRESS.," in Robotics Research, the 1st International Symposium., 1984.
- [81] C. Loucks, V. Johnson, P. Boissiere, G. Starr, and J. Steele, "Modeling and control of the stanford/JPL hand," in Proc. IEEE Int. Conf. Robotics and Automation. on, 1987.
- [82] A. Caffaz and G. Cannata, "The design and development of the DIST-Hand dexterous gripper," in Proc. IEEE Int. Conf. Robotics and Automation, 1998.
- [83] H. Kawasaki, T. Komatsu, and K. Uchiyama, "Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand II," IEEE/ASME Trans. Mechatronics, vol. 7, pp. 296-303, 2002.
- [84] Y. Kuroki, K. Kato, K. Nagasaka, A. Miyamoto, K. Ueno, and J. Yamaguchi, "Motion evaluating system for a small biped entertainment robot," in Proc. IEEE Int. Conf. Robotics and Automation, 2004.
- [85] Honda Motor Co., Ltd, "Asimo's specification," Honda Motors, 2004, http://world.honda.com/ASIMO/technology/spec.html (Jan. 2005).
- [86] Semenyuk, V., Stockholm, J., Musolff, A., Seelecke, S, "The Use of Thermoelectric Cooling for Shape Memory Wire Temperature Control," Proceedings of the ETS Workshop 98, 1998.
- [87] Khan, M, Lagoudas, D., Rediniotis, O., "Thermoelectric SMA actuator: preliminary prototype testing," Proceedings of SPIE, v 5054, pp. 147-155, 2003.

APPENDIX A: Proof of the theorem 1

Consider a block of coupled segments of length C_{Am} , consisting of ON-state segments with length $z_{Am}^{ON}(t)$ and OFF-state segments with length $z_{Am}^{OFF}(t)$:

$$z_{Am}^{ON}(t) + z_{Am}^{OFF}(t) = C_{Am}, \quad t \in T_I$$
(A-1)

Let us design the coupled segments such that $z_{Am}^{ON}(t)$ is given by

$$z_{Am}^{ON}(t) = Min\left[C_{Am}, \underset{i \in A_m}{Min} x_i^{ON}(t)\right]$$
(A-2)

and that the OFF segment length is given by $z_{Am}^{OFF}(t)$

$$z_{Am}^{OFF}(t) = C_{Am} - z_{Am}^{ON}(t)$$
 (A-3)

We want to show that, although a fraction of $y_i^{ON}(t)$ is determined by the coupled segments, the remaining independent segments whose length is 1 - C_{Am} are large enough to generate the rest of $x_i^{ON}(t)$:

$$\Delta x_i^{ON}(t) = x_i^{ON}(t) - z_{Am}^{ON}(t) \le 1 - C_{Am}$$
(A-4)

Let us first assume that

$$\underset{i \in A_m}{Min \, x_i} x_i^{ON}(t) \le C_{Am} \tag{A-5}$$

Then, from (A-2),

$$\Delta x_{i}^{ON}(t) = x_{i}^{ON}(t) - \underset{j \in Am}{Min} x_{j}^{ON}$$

$$\leq 1 - \underset{j \in Am}{Min} x_{j}^{OFF}(t) - \underset{j \in Am}{Min} x_{j}^{ON}(t)$$

$$\leq 1 - \underset{i \in T_{i}}{Min} \left[\underset{j \in Am}{Min} x_{j}^{ON}(t) + \underset{j \in Am}{Min} x_{j}^{OFF}(t) \right] = 1 - C_{Am}$$
(A-6)

From the first line of the above equation it is clear that $\Delta x_i^{ON}(t)$ is non-negative, hence

$$0 \le \Delta x_i^{ON}(t) \le 1 - C_{Am} \tag{A-7}$$

If (A-5) does not hold, then from (A-2), $z_{Am}^{ON}(t) = C_{Am}$.

$$\Delta x_i^{ON}(t) = x_i^{ON}(t) - C_{Am} \le 1 - C_{Am}$$
(A-8)

Also,

$$0 \le \min_{j \in A_m} x_j^{ON}(t) - C_{Am} \le x_i^{ON}(t) - C_{Am} = \Delta x_j^{ON}(t)$$
(A-9)

Therefore, (A-4) has been derived, i.e. the independent segments are large enough to generate $x_i^{ON}(t)$, so that $x_i^{ON}(t) = z_{Am}^{ON}(t) + \Delta x_i^{ON}(t)$. Similarly, we can show that the independent segments are long enough to generate $x_i^{OFF}(t)$. From (A-3) and (13),

$$\Delta x_i^{OFF}(t) = x_i^{OFF}(t) - z_{Am}^{OFF}(t)$$

$$= (1 - x_i^{ON}(t)) - (C_{Am} - z_{Am}^{ON}(t))$$

$$= 1 - C_{Am} - (x_i^{ON}(t) - z_{Am}^{ON}(t))$$

$$= 1 - C_{Am} - \Delta y_i^{ON}(t) \le 1 - C_{Am}$$
(A-10)

Using (A-8) in the above equation yields

$$\Delta x_i^{OFF}(t) = 1 - C_{Am} - \Delta x_i^{ON}(t) \ge 0$$
(A-11)

Therefore,

$$0 \le \Delta x_i^{OFF}(t) \le 1 - C_{Am} \tag{A-12}$$

Q.E.D.

APPENDIX B: Heuristic approach of generating the C-segments and it's application to a robotic hand

Based on the theoretical analysis described in chapter 2, a heuristic design procedure for finding multi-axis segmentation with reduced control loops is presented.

Coupling coefficients play the major role in finding which axes to lump together and how long the coupled portion of segments can be.

The procedure for designing a multi-axis segmentation is three-fold: first the original M axis problem is divided into smaller uncoupled problems of fewer axes, second a gross segmentation layout is made for each uncoupled problem, and third the gross layout is fine-tuned to finalize the design.

Step 1. Decoupling and Merger

The original M axis problem can be divided into reduced axis problems using the following algorithm:

1.1 Compute the *M*-by-*M* Coupling Matrix given by (2-11) for all the combinations of two axes, that is,

$$\{y_{i}(t), y_{j}(t) | i, j \in A_{M}, t \subset T_{I}\}$$
(B-1)

- 1.2 Search for 1's in the matrix. If 1 is found at the $(i,j)^{th}$ element, actuators *i* and *j* are totally coupled. Merge axes *i* and *j* to reduce the number of axes. If there are multiple 1's in a row, merge all of them into a single axis.
- 1.3 Search for 0's in the Coupling Matrix. Swap the actuator numbers *1* through *M* for shifting 0's to off-diagonal blocks so that the Coupling Matrix may be transformed to a block-diagonal matrix. Resultant diagonal blocks, if exist, determine subgroups of axes that are mutually independent. Segmentation design may be performed for each subgroup of axes.

Step 2 Gross Segmentation

For each of the axis subgroups determined in Step 1, the following gross segmentation is performed.

2.1 Enumerate all the exclusive partitions of the axes involved in the subgroup. Let M_o be the total number of axes involved in a subgroup, and A_{Mo} be the set of the M_o axes. Each partition of the axes is represented with a set of axis subsets given by

$$P_{k} = \{A_{m1}, A_{m2}, \cdots, A_{mk} \mid A_{Mo} = A_{m1} \cup A_{m2} \cup \cdots \cup A_{mk}\},$$

$$k = 1, \cdots, K$$
(B-2)

where A_{m1}, \ldots, A_{mk} are axis subsets that are exclusive to each other and that cover the whole M_o axes. The axis subset A_{mi} may be a singleton, having only one axis as a component.

- 2.2 Eliminate partition P_k , if it includes an axis subset A_{mi} that is not a singleton and whose coupling coefficient C_{Ami} is zero. For such an axis subset, segments cannot be lumped. Hence it is subsumed by other partition having finer partitions.
- 2.3 Compute the coupling coefficient of each A_{mi} , and construct coupled segments of length C_{Ami} and independent segments of length 1- C_{Ami} .

Step 3 Fine Segmentation

The independent segments created in Step 2 may further be coupled.

- 3.1 Obtain the residual postures $\Delta y_i(t)$ given by (2-9), and compute the M_o -by- M_o Coupling Matrix.
- 3.2 Repeat Step 2 for the residual postures, and construct coupled segments within the independent segment area generated by the gross segmentation. See Remark 1.
- 3.3 Repeat Steps 3.1 and 3.2 until all the residual postures have zero coupling coefficients, or the number of segments cannot be reduced by coupling the segments.
- 3.4 Select the segmentation design that has the minimum number of segments among all the partitions evaluated above.

Example To help understand the above procedures, a simple example is presented in this section. Consider a seven-axis actuator array. Table B-1 shows a set of normalized outputs to be generated by the actuator array. Fig. B-1 shows the process of designing the multi-axis segmentation for generating the set of outputs, leading to the minimum segmentation, i.e. the minimum number of segments with independent ON-OFF controls for generating the set of outputs.

Step 1 Decoupling and Merger First, the coupling matrix is constructed based on Step 1.1.

See Table B-2. The first row and column show the actuator numbers, 1 through 7. From the coupling matrix, two sets of mutually independent axis subsets $\{1,2\}$ and $\{3, 4, 5, 6, 7\}$ are identified. The segmentation of these two groups can be performed separately as shown in Fig. B-1-(a). Note that the coupling coefficient $C_{6,7}$ is one, hence they are totally coupled.

Axis	Normalized Trajectories											
1	0	0.5	0.75	1	0	0.25	0					
2	0	0.5	0.75	1	0.75	0.75	0					
3	0	0.5	0.75	1	0	0.25	1					
4	0	0.5	0.75	1	0	0.5	1					
5	0	0.5	0.75	1	0.75	0.75	1					
6	0	0.5	0.75	1	1	1	1					
7	0	0.5	0.75	1	1	1	1					

TABLE B-1 EXAMPLE OF POSTURES FOR SEVEN-AXIS ACTUATOR ARRAY

TABLE B-2 COUPLING COEFFICIENT MATRIX OF THE EXAMPLE

	1	2	3	4	5	6	7
1	1	0.25	0	0	0	0	0
2	0.25	1	0	0	0	0	0
3	0	0	1	0.75	0.25	0	0
4	0	0	0.75	1	0.25	0	0
5	0	0	0.25	0.25	1	0.75	0.75
6	0	0	0	0	0.75	1	1
7	0	0	0	0	0.75	1	1

Step 2 Gross Segmentation C_{12} is 0.25; hence 25% of the length of the actuator axes 1 and 2 can be lumped together. Fig. B-1-(b) indicates that Axes 1 and 2 share 25% of common segments. There are eleven different ways of exclusively partitioning the actuators 3 to 7. Among these only meaningful partitions are $P_1 = \{\{3,4\}, \{5,6,7\}\}$ and $P_2 = \{\{3,4,5\}, \{6,7\}\}$. Other cases are pruned out in Step 2.2. In Fig. 7-(b), the partition P_1 , comprising $\{3,4\}$ and $\{5,6,7\}$, is shown. With $C_{3,4} = 0.75$ and $C_{5,6,7} = 0.75$, both groups can have 75% length of common segments shown by the shaded areas. A similar result can be obtained for the partition P_2 , too.



Fig. B-1 Procedures of segmentation design for the postures given in table 1. (a) Actuators divided into two decoupled groups and actuators {6, 7} are totally coupled. (b) Common segments for actuators {1,2}, {3,4} and {5, 6, 7} are designed. (c) Common segments for independent postures of actuators {3, 5} are designed (d) Minimum number segmentation found for each segments

Step 3 Fine Segmentation To further couple the independent portion of the segments, *residual postures* are obtained from (2-9), and a new coupling matrix is derived for the residual postures. As stated in Remark 3, the residual postures are in general not unique. For the gross segmentation given by Fig. B-1-(b), there are 128 different sets of residual postures. When the largest possible ON-state length is used for generating the coupled portion, $z_{Am}^{ON}(t)$ and $z_{Am}^{OFF}(t)$, the coupling matrix of Table 3 for axes 1 through 6 (now denoted *1i* to *6i*) is obtained. In this coupling matrix, $C_{3i,5i}$ is 0.25, hence the independent segments of axes 3 and 5 can be lumped together up to 25% of the total length of the actuator, which is the whole length of the independent segments of axes 3 and 5.

	1 <i>i</i>	2i	3i	4i	5i	6i
1i	0.75	0	0	0	0	0
2i	0	0.75	0	0	0	0
3i	0	0	0.25	0	0.25	0
4i	0	0	0	0.25	0	0
5i	0	0	0.25	0	0.25	0
6i	0	0	0	0	0	0.25

TABLE B-3 COUPLING COEFFICIENT MATRIX OF INDEPENDENT POSTURES

As shown in Fig. B-1-(c), axis 3 is swapped with axis 4 to make axes 3 and 5 adjacent to

each other so that they can be coupled together. Finally, Fig. B-1-(d) shows the final segmentation layout including the ones inside the common segment blocks. Segmentation design for these common segment blocks is basically a single-axis segmentation problem, as mentioned in Remark 2.

The above computation is repeated for all the possible residual postures as well as for different gross segmentations in Step 2. The design with twelve segments as shown in Fig. B-1-(d) is the minimum segmentation among these. If the segments were not coupled and all the segments were uniform, twenty-eight segments would have been used; Hence 57 % savings in the number of segments have been achieved.

The design concept of multi-axis array actuators with two-dimensional segmentation architecture is applied to an anthropomorphic robot hand. A single human hand has 19 joints at the five fingers alone. Despite numerous degrees of freedom, many of them are coupled, as mentioned previously. Such coupled motion can be generated effectively with use of multi-axis segmentation architecture.



Fig. B-2 Hand postures used in the design of the actuator

Fig. B-2 shows fourteen hand postures that span the useful range of human hand motions. For these postures, finger joint angles can be derived, and an actuator array can be designed such that the joint angles of all the postures may be generated with the minimum number of segments.

These joint angles are transformed into displacements of the actuators driving the individual joints. To implement segmented binary control, the displacements must be discretized and the resolution of actuator axes must be determined accordingly.

Actuator #	Assigned Joint	Open	Survey	Envelopel	Envelope2	Ball1	Ball2	Fistl	Fist2	Pinchl	Pinch2	Pinch3	Pinch4	Point	Write
1	Thumb(DIP)	0	0.2	0.2	0.4	0.4	0.6	0.8	1	0.4	0.4	0.4	0.4	1	0
2	Thumb(MP)	0	0.4	0.4	0.6	0.4	0.6	0.8	1	0.6	0.6	0.6	0.6	1	0.4
3	Index(DIP)	0	0.4	0.4	0.6	0.6	0.8	0.8	1	0.6	0.6	0.6	0.6	0	0.6
4	Index(MP)	0	0.2	0.4	0.6	0.2	0.4	0.8	1	0.6	0.8	0.6	0.8	0.4	0.6
5	Middle(DIP)	0	0.4	0.4	0.6	0.6	0.8	0.8	1	0.4	0.6	0.6	0.6	1	0.8
6	Middle(MP)	0	0.2	0.4	0.6	0.2	0.4	0.8	1	0.2	0.4	0.6	0.8	1	0.6
7	Ring(DIP)	0	0.4	0.4	0.6	0.6	0.8	0.8	1	0.2	0.4	1	1	1	1
8	Ring(MP)	0	0.2	0.4	0.6	0.2	0.4	0.8	1	0	0.2	1	1	1	1
9	Pinkie(DIP)	0	0.4	0.4	0.6	0.6	0.8	0.8	1	0	0.2	1	1	1	1
10	Pinkie(MP)	0	0.2	0.4	0.6	0.2	0.4	0.8	1	0	0.2	1	1	1	1

TABLE B-1 DISCRETIZED AND NORMALIZED DISPLACEMENTS USED FOR SEGMENTATION DESIGN

Let θ_i be the *i*-th finger joint angle and α_i be the maximum bending angle, i.e. the stroke, of the *i-th* finger axis. The normalized output displacement y_i of the actuator axis *i* is given by $y_i = \theta_i / \alpha_i$, $0 \le \theta_i \le \alpha_i$. The output displacement is quantized into five levels: N=5. Table B-1 shows the discretized and normalized displacements used for segmentation design, where each column represents a specific hand posture and each row represents the displacements of each actuator. Now using these data, segmentation is designed using the procedure given in the previous section.



Fig. B-3 Minimum segmentation design for 10-axis array actuators driving robotic hand

Fig. B-3 shows the final segmentation design achieving the minimum number of segments. The black lines represent the actuator wire that changes displacement as different segments are activated. Twenty-one segments are used in the final design, which is 58% savings in the number of segments compared to the original uncoupled design.

For this prototype system, experiments have been performed to verify that all of the fourteen hand postures shown can be generated with the minimum segmentation layout implemented above. Fig. B-4 shows the activation patterns that generate the different hand postures. The shaded segments represent the segments to be activated. By activating different combination of segments, all 14 postures can be generated. Fig. B-5 shows the displacements of axes 7 and 10 for all the 14 hand postures. Positioning accuracy in taking all the 14 postures in four different sequences has been tested. Although arrived from different postures, the final positions of each posture are consistent. The unit displacement created with a single segment is 1 = 1.4 mm, and the standard deviation of the positioning error is approximately 0.1 mm. These results demonstrate the feasibility of using the actuator array for driving the robot hand with open loop controls by merely varying the segment temperature.



Fig. B-4 Activation patterns of the Actuator Array. Shaded segments represent the segments to be activated to generate different postures.



Fig. B-5 Experimental result with the ten axis actuator array: Normalized displacements of (a) Axis 7 and (b) Axis 10 for all 14 hand postures.

APPENDIX C: Mechanical design of the robotic hand

The main consideration in the mechanical design was simplicity. The resulting design was produce a compact, lightweight, 16-axis, 20 degree-of-freedom, open loop humanoid hand.

Forearm

Following the human model, the apparatus was designed such that all of the actuators, which are located in the forearm, transmit the forces to the fingers via Kevlar-cable tendons. The hand was designed to be actuated exclusively by a shape memory alloy (SMA) due to the compact, high power-density characteristics of the material. Being that SMA is a unilateral actuator, an antagonistic actuation scheme was implemented.

Electrical Contact Pins

It was chosen to use an array of metal pins as electrical contacts to the SMA wires for a variety of reasons. It is important to keep the space around the SMA wires as vacant as possible since they are cooled by forced convection. Crimping a different wire to each segment of each axis would choke airflow and furthermore prove to create a very complex assembly. The metal pins, however, leave plenty of room for airflow and can be soldered directly to a circuit board.



Fig. C-1: Segments of SMA contacting perpendicular pins which can be easily soldered to a circuit board.

Usage of the pins, however, introduces new problems. Since the contact between the pins and the SMA wires is basically two perpendicular cylinders, it is very small, almost a point contact. The resistance at this contact can become greater than at the rest of the wire causing a small section of SMA to heat up to a much larger temperature than the rest. This can cause the wire life to be significantly shorter. This can be avoided by holding the SMA wires under tension to ensure a greater contact.

Travel-Limited Compliance

Since SMA's are unilateral actuators, an antagonistic actuation scheme is implemented.



Fig. C-2 Relaxed finger with unactivated actuators at natural length (l_{θ}) .

Since SMA actuators shrink in length when activated, the total length of both actuators (antagonistic pair) changes when one axis is activated (see Fig. C-2 & Fig. C-3).



Fig. C-3: Bent finger with flexor axis activated

As can be seen, any change in the total length of the actuators either eliminates the option of a fixed, anchored endpoint or causes the introduction of slack to the system. Since a moving anchor point is not practical, and constant tension is needed to ensure proper functionality of the system, compliance must be introduced.

The introduction of compliance, however, creates a new problem. If the compliance is too high, the energy from the actuators is stored in springs rather than actually used for useful work, but if the compliance is too low then the pullback actuator is not capable of exerting enough force to restore the finger to its original position.

However, since the total travel of the tendons for each axis is known, springs with a low stiffness can be limited to the exact travel of the axis. Thus the flexor axes experience little interference during flexion, but they experience the full force of the extensor actuators when the extensor actuators are activated.



Fig. C-4 Relaxed compression spring



Fig. C-5 Compression spring at travel limit

It was chosen to use compression springs since devices for mechanically limiting travel are much simpler.

Hand

Finger Design

Fingers in the human hand have four degrees of freedom, three of which are coplanar. These four degrees of freedom are accomplished in three joints, the Distal Interphalangeal joint (DIP), the Proximal Interphalangeal (PIP) joint and the Metacarpal Phalangeal joint (MCP). Rather than attempting to recreate the exact structure of the human hand with only three joints (two pin joints and a saddle joint) it was decided that it would be advantageous to focus on recreating the functionality of the human model. As such, four pin joints are used in each of the five fingers of the robotic hand. Consistent with the human model, the fingers are underactuated; each finger has only three controlled degrees of freedom.





Fig. C-6: Robotic Finger



Thumb Design

The thumb was designed to have four degrees of freedom. Its design was analogous to a finger with the PIP joint removed, but with an extra degree of freedom which allows it to swing out in front of the hand. It is to be noted that the thumb does not directly oppose the fingers in grasping motions: in such configurations it is perpendicular. Intuitively, when people think of the pinching motion, they think that the thumb is perfectly opposite to the finger. The thumb was originally designed to accomplish this configuration, however it was found very quickly that in order to replicate the motion of the thumb without introducing extra degrees of freedom, a similar joint configuration is necessary.

The robotic thumb has four segments and four joints. It should be noted that some of the labels of the joints of the thumb in Fig. C-7 are the same as the labels of the joints of the fingers in Fig. C-6. The corresponding joints are almost identical. The only difference is the MCP_2 joint of the thumb has a larger range of motion. The CMC joint of the thumb, however, has no analogous joint in the finger. This too replicates the human hand. In both Fig. C-6 and Fig.
C-7, small holes can be seen which do not correspond to pin joints. These holes locate tendon routing pins. These pins are included because the coefficient of friction of steel on Kevlar is significantly less than that of unfinished plastic on Kevlar.

Tendon Routing

The tendons were routed through the hands using steel posts. Almost all of the tendons (except those associated with the thumb) coalesce in the palm. As such, the palm has a removable cover to allow for assembly. The holes at the base of the opening in Fig. C-8 correspond to steel posts which align the tendons with the actuators in the forearm.



Fig. C-8 Removable Cover

The steel guide-posts make for a significantly simpler internal structure of the plastic. Lateral holes replaced pulley-like structures, and voids replaced unnecessary long sheaths. Fig. C-9 shows an orthogonal view of the tendon routing within the finger. Since shape memory alloys are unilateral actuators, the tendons were routed antagonistically to maintain full control, thus for each controlled degree of freedom, there are two tendons. In the figure, the circles labeled 'P' correspond to the pins in the pin joints. The circles labeled 'RP' correspond to the redirecting pins, and the markers labeled 'T' correspond to points of tendon termination. As can be noted, no tendons terminate in the "Middle Phalanx" or the second most extreme finger; as such, the PIP joint is not independently controlled. The tendon which terminates at the fingertip, however, is routed in such a way that its length is affected by the DIP, PIP and MCP₁ joints. This coupling allows for the entire finger to be actuated by a single axis. However, the termination of the tendons in the Proximal Phalanx allow for complete control of the MCP₁ joint. This makes it such that the controller can choose the level of underactuation. The final set of tendons terminates in the base piece. This piece has no analogous bone in the human hand. The three coplanar joints, the DIP, PIP and MCP₁, can all undergo a motion of 90°. The third joint, the MCP₂, can bend 20° in either direction from its neutral position in the fingers; in the thumb, it can bend a full 45° from its neutral position.





Fig. C-9: Orthogonal view of tendon routing

Fig. C-10: Tendon routing of MCP segment

Both sets of the flexion/extension tendons are routed through the center of the finger. This is so that there is no coupling associated with the movement of the MCP₂ joint. However, since these are pin joints, the tendons are located in the exact same space as the pin would be. In order to be able to allow both of them to coexist, the pin for the pin joint is separated into two pieces. As can be seen in Fig. C-12, all of the tendons occupy a void, which is represented as the white space at the center. The black regions labeled MCP₂ correspond to the pins of the MCP₂ joint. It should be noted that these pins do not interfere with the cables, which are routed through the void. Fig. C-10 shows the tendon routing through the MCP segment; the three different angles of each tendon corresponds to the tendon configuration in its neutral position and at its travel limits in both directions. The MCP segment is the only piece that uses its own geometry as a pulley for the tendons. As stated earlier, the rest of the joints rely on metal posts. This is the only part of the entire hand that contacts the tendons in way other than as an anchor.



Fig. C-11: The thumb base

As stated earlier, the first three segments of the thumb are exactly the same as three corresponding segments of the finger, but the base segment is completely different. This segment is significantly larger than the other segments and undergoes a significantly greater displacement when moving. As such, without careful design, the coupling from the movement of this joint becomes severely detrimental to the performance of the thumb. It is this movement that makes the thumb opposable; it is arguably the most important feature of the thumb. Since it is the most difficult motion to replicate, it was tempting to just discard this motion; obviously, both the functionality and the anthropomorphicity would be detrimentally affected. The design of the thumb led to a minimal coupling due to this motion. However, the coupling was still severe enough to affect the performance. The coupling only affects the performance when the joint moves away from a specific position, a neutral position. Since the thumb is always opposing the hand when performing tasks, it was determined that this would be the natural When the thumb is pulled back, it has limited functionality; once again, this perfectly position. mimics the human hand. Motion is significantly more difficult for a human when the thumb is The shaded region in the thumb (see Fig. C-11) represents a void in the material. pulled back.

The holes near the base are for steel posts which redirect the tendons from the rest of the thumb. The tendons exit the segment at the base.

Finger Kinematics

The geometry of the internal structure of the hand allows for very simple kinematics of the extensor tendons, but very complex kinematics for the flexor tendons. Almost all of the joints have the same configuration, (all but the MCP_2 joints) so the calculations only have to be done once.



Fig. C-13: Finger Kinematics

The change in length of the extensor due to a change in angle, θ , is simply given by the equation

$$\Delta l = (R + r_0 \theta) - (R) = r_0 \theta \tag{C-1}$$

where l is the length of the tendon, and R and r_0 are known dimensions of the finger. Very similarly, the equation for the flexor is

$$\Delta l = (d + r_0 \phi) - (R) \tag{C-2}$$

where $_R$ and r_0 are once again known dimensions of the finger. However, $_d$ and φ are configuration dependent. The length of d can be calculated from the equation

$$d = \sqrt{d_1^2 - r_0^2}$$
 (C-3)

where dl also needs to be calculated. The length of d_1 can be found by calculating the distance from P_1 to P_2 .

$$\vec{d}_1 = \vec{P}_2 - \vec{P}_1 \tag{C-4}$$

The position of P_I (distance from O) is described by the following equation:

$$\vec{P}_1 = \begin{bmatrix} -d_2 \\ -d_2 \end{bmatrix} \tag{C-5}$$

it can be noted that since P_1 is fixed, it does not change when θ is changed. The position of P_2 , however, does change with θ , and as such is significantly more complex:

$$\vec{P}_2 = \begin{bmatrix} -r_0 \cos\theta - R \sin\theta \\ -r_0 \sin\theta + R \cos\theta \end{bmatrix}$$
(C-6)Equations (21)-(24) fully describe

d in terms of known parameters. However, in order to solve (20), φ too must be calculated. φ is described by the equation

$$\phi = \pi - \left(\phi_1 + \phi_2\right) \tag{C-7}$$

where φ_1 and φ_2 are expressed in radians. φ_1 is described by the equation

$$\cos\phi_1 = \frac{r_0}{d_1} \tag{C-8}$$

and φ_2 is described by the equation

$$\tan\phi_2 = \frac{d_{1y}}{d_{1x}} \tag{C-9}$$

combining equations (20)-(27) we arrive at an expression for the flexion motion:

$$\Delta l = \begin{pmatrix} \sqrt{(d_2 - r_0 \cos \theta - R \sin \theta)^2 + (d_2 - r_0 \sin \theta + R \cos \theta)^2 - r_0^2} \\ + r_0 \pi - \arccos\left(\frac{r_0}{\sqrt{(d_2 - r_0 \cos \theta - R \sin \theta)^2 + (d_2 - r_0 \sin \theta + R \cos \theta)^2}}\right) \\ - \arctan\left(\frac{d_2 - r_0 \sin \theta + R \cos \theta}{d_2 - r_0 \cos \theta - R \sin \theta}\right) - (R) \end{pmatrix}$$
(C-10)

This equation is rather large and cumbersome. Finding the inverse kinematics for this would be quite difficult. However, Fig. C-14 shows that the parameters of the finger make it such that the equation can be approximated quite effectively as linear.



Fig. C-14: Tendon displacement as a function of angular displacement

APPENDIX D: Drive circuitry of the C-Segmentation board

Schematics of drive circuitry to drive each C-segment is shown in. Each segment of SMA actuator from different axis is represented as a resistor that is connected to each other in series using switches. By using a DUAL MOSFET IRF8910 that has a current capacity of 10Amps in a SO-8 pacakaging, many MOSFET switches can be put in a small footprint.



Fig. D-1 Schematic of Circuit Diagram for single C-segment

The setup shown in Fig. D-1**Error! Reference source not found.** is a unique configuration of MOSFET switches, since all the MOSFETs connecting a single C-segment should be turned on and off at the same time but the required drive voltages are all different. Since V_{GS} for all the MOSFETS should be between 5V to 20Volts, the voltage at the gate of the MOSFETS can range from 10 to 80Volts.

Although it is possible to use MOSFET drivers for each MOSFET, that would increase the number of components and complexity of the system. One solution to this problem is to use zener diodes to limit the voltages V_{GS} of all the MOSFETs and drive the MOSFETs with single high voltage.



Fig. D-1 Voltage requirement for different MOSFET gates

As shown in Fig. D-2 the gate driving voltages are limited to 18Volts using a voltage limiting zener diode. The cathode of the zener diode is connected to the gate of the MOSFET and the anode is connected to the drain, so that the voltage between the gate and the source is limited.



Fig. D-2 Voltage limiting zener diode

But there is still a problem with this schematic. Since all the C-segments are electrically connected to the neighboring segments, any voltage applied at the segment will show up at the gate of the segment next to it through the forward bias of the zener diode. This configuration is shown in Fig. D-3.



Fig. D-3 Shadow voltage showing up at the gate of the connected segments



Fig. D-4 Blocking diode paired with zener diode

Therefore, a blocking diode that will isolate the voltages applied at the adjacent segments is needed next to the zener diodes. This configuration is shown in Fig. D-5.