uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Aaron David Price

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.A.Sc. (Mechanical Engineering)

GRADE / DEGREE

Department of Mechanical Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Biologically Inspired Dexterous Robot Hand Actuated by Smart Material Based Artificial Muscles

TITRE DE LA THÈSE / TITLE OF THESIS

Dr. Hani Naguib

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

Dr. A. Jnifene (absent)

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Dr. Natalie Baddour

Dr. Fred Nitzsche

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Biologically Inspired Dexterous Robot Hand Actuated by Smart Material Based Artificial Muscles

by

Aaron David Price

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.A.Sc. degree in
Mechanical Engineering

Dept. of Mechanical Engineering
Faculty of Engineering
University of Ottawa

Your file Votre référence
ISBN: 978-0-494-25822-4
Our file Notre référence
ISBN: 978-0-494-25822-4

Abstract

Modern externally powered upper-body prostheses are conventionally actuated by electric servomotors. Although these motors achieve reasonable kinematic performance, they are voluminous and heavy. Deterring factors such as these lead to a substantial proportion of upper extremity amputees avoiding the use of their prostheses. Therefore, it is apparent that there exists a need for functional prosthetic devices that are compact and light-weight. The realization of such a device requires an alternative actuation technology, and biological inspiration suggests that tendon based systems are advantageous. Shape memory alloys are a type of smart material that exhibit an actuation mechanism resembling the biological equivalent. As such, shape memory alloy enabled devices promise to be of major importance in the future of dexterous robotics, and to prosthetics in particular. This thesis investigates the issues surrounding the practical application of shape memory alloys as artificial muscles in a three fingered robot hand. First the function of the human hand and the kinematic requirements for manipulation are reviewed. An overview of artificial hands is provided, followed by a discussion on shape memory alloys focused on the unique phenomena of the shape memory effect. Second, the forward and inverse kinematics of the artificial finger are established in order to relate the desired finger tip contact point to the required joint angles. This is followed by the design of the requisite instrumentation and control systems. Due to the highly nonlinear nature of both the SMA and the robot hand, alternative control approaches such as neural networks are reviewed. Finally, a large-strain SMA actuator is proposed and the concepts explored herein are applied to the design, manufacture, and evaluation of an SMA actuated robotic hand.

Our hands become extensions of the intellect, for by hand movements the dumb converse, with the specialized fingertips the blind read; and through the written word we learn from the past and transmit to the future.

–Sterling Bunnell
Surgery of the hand
Lippincott, Philadelphia, 1944

Acknowledgements

Thanks to my supervisors, Dr. Hani Naguib and Dr. Amor Jnifene for their guidance throughout the duration of this project. Thanks also to Dr. Florent Goldberg, formerly of the Smart Structures and Intelligent Systems Laboratory, and Dr. Edward Lemaire of the Institute for Rehabilitation Research and Development for offering their expertise in shape memory alloys and prosthetic devices respectively. Finally, thank you to my friends, family, colleagues, and the administrative and technical support staff, especially to Andrew Cameron, Andrew Edgerton, Michael Milner, and Sarah Mercer for their technical assistance and to Kelly MacDonnell for her boundless support.

This project was made possible by the financial support of NSERC, The Institute for Robotics and Intelligent Systems (IRIS), Precarn Inc., the University of Ottawa, and the Department of Mechanical Engineering.

Contents

1 Introduction **1**

1.1 Biologically inspired technology 1

1.2 Motivation and objectives 2

1.3 Contributions 4

1.4 Thesis organization 4

2 Literature Review **6**

2.1 The human hand 6

2.1.1 Biomechanics 6

2.1.2 Performance criteria 8

2.1.3 Kinematic parameter determination 9

2.2 Kinematic requirements for manipulation 10

2.2.1 Artificial hand configurations 10

2.2.2 Design requirements 11

2.3 Conventional artificial hands 14

2.3.1 Prosthetic devices 15

2.3.2 Robots 15

2.3.3 Conventional Actuator Technology 18

2.4 Smart materials 19

2.4.1 Typical smart materials and properties 19

2.5 Shape memory alloys 20

2.5.1 SMA actuators 21

2.5.2 SMA phenomenology 21

2.5.3 SMA based devices 23

2.5.4 Methods of controlling SMA 24

2.6 Summary 31

3 Kinematic Analysis and Design of a Three Fingered Robotic Hand 33

3.1 Forward kinematics 33
3.2 Inverse kinematics 35
3.3 Coordinate transformations 38
3.4 System instrumentation 40
3.5 Summary 42

4 Materials and Methods 44

4.1 SMA metallurgy 44
4.2 SMA enabled artificial finger 45
4.3 SMA enabled artificial hand 49
4.4 System instrumentation 49
4.5 Evaluation of constitutive model 53
4.6 Sensor quantization 56
4.7 High-level control system 60
4.8 Summary 64

5 Development of a high strain actuator 66

5.1 Actuator design 66
5.2 Actuator performance 67
5.3 Summary 71

6 Position control of an SMA based artificial hand 72

6.1 Sigmoid feedback control law 72
6.2 Modified sigmoid feedback control law 74
6.3 Summary 79

7 Conclusions 81

8 Recommendations and future work 83

8.1 Recommendations 83
8.2 Future work 84

List of references 85

Appendices

A Detailed design of robot phalanges **92**

B Detailed electrical system design **97**

C Software code listing **101**

C.1 Micro-controller Programs 101

C.1.1 Master micro-controller program 101

C.1.2 Slave micro-controller program 105

C.2 MATLAB Functions 108

C.2.1 Main program 108

C.2.2 Supporting functions 111

List of Tables

2.1 Sensory resolution of the human hand 9
2.2 Structural properties of selected hands 16

3.1 DH parameters for adduction-abduction finger 34

4.1 Actual DH Parameters. 46
4.2 Maximum joint range of motion 48
4.3 Model parameters for Flexinol® wire tests 54
4.4 Model Parameters for nitinol ribbon tests 55

5.1 Maximum strain achieved by four-loop actuator 70

List of Figures

1.1 Myoelectric prosthetic hand 3

2.1 The human finger 7
2.2 Contact types 12
2.3 Mechanical hand classification 14
2.4 Utah/MIT hand 17
2.5 Stanford/JPL mechanical hand 17
2.6 Belgrade mechanical hand 18
2.7 Shape memory effect for uniaxial SMA 22
2.8 Open loop hysteresis compensation using an MLP 29
2.9 Position control using RBF 30
2.10 Position control using ANFIS with PD control 31

3.1 Kinematic structure of artificial finger 34
3.2 Kinematic structure in the $xy - z$ plane 37
3.3 Coordinate transformations 39
3.4 Hand system instrumentation diagram 41
3.5 Finger IO subsystem 41
3.6 Sensor signal conditioning circuit 43
3.7 SMA drive circuit 43

4.1 Flexinol® wire in original condition 45
4.2 Flexinol® wire after annealing and quenching 45
4.3 SMA routing in linear and joint traversing arrangement 47
4.4 MCP joint range of motion 48
4.5 SMA actuated artificial hand 50
4.6 SMA actuated artificial hand (alternate view) 51

4.7 Flexinol® wire 54
4.8 Nitinol ribbon 55
4.9 Typical MCP joint sensor calibration curve 57
4.10 Typical PIP joint sensor calibration curve 58
4.11 Sigmoid control signal 62
4.12 Single finger control algorithm 64

5.1 High strain braided SMA actuator 67
5.2 Linear test frame 68
5.3 Actuator response 69
5.4 Biomimetic prosthetic arm 71

6.1 Contact point trajectory expressed in the base frame 75
6.2 Contact point trajectory expressed in the joint space 76
6.3 Joint error and normalized control signal 77
6.4 Modified control algorithm 78
6.5 Response using modified controller 79

A.1 Proximal phalanx 93
A.2 Middle phalanx 94
A.3 Distal phalanx 95
A.4 Metacarpal base 96

B.1 Digital circuit 98
B.2 Analog circuit 99
B.3 Sensor signal conditioning circuit 100

Nomenclature

α	Coefficient of linear thermal expansion
α	DH link twist angle parameter
α^{c}	Coefficient of linear thermal expansion of the constraint
φ_δ	Contact point offset angle
ε	Strain
ε^{c}	Initial macroscopic strain when constraint is applied
$\varepsilon^{\mathrm{cc}}$	Macroscopic strain constraint
$\varepsilon^{\mathrm{el}}$	Elastic strain
$\boldsymbol{e}_\theta$	Joint error vector
ε_r	Apparent residual strain
$\varepsilon^{\mathrm{th}}$	Thermal expansion strain
$\varepsilon^{\mathrm{tr}}$	Transformation strain
$\varepsilon^{\mathrm{tr,max}}$	Maximum transformation strain
φ	Sum of contact point offset angle and DIP joint angle
φ	Thermomechanical driving force on the martensite transformation
κ	Finger tip contact point error tolerance
$\boldsymbol{\mu}$	Vector of sigmoid shift parameters

$\boldsymbol{\theta}$	Joint angle vector
θ	DH joint angle parameter
θ	Joint angle coordinate
$\hat{\boldsymbol{\theta}}$	Required joint angle vector
σ	Stress
σ^{re}	Reorientation stress
$\xi(\varphi)$	Martensite evolution function
a	DH link length parameter
A_{f}	Austenite finish temperature
$\boldsymbol{A}_{\mathrm{f}}^{\mathrm{w}}$	Coordinate transformation from world frame, w, to local frame f
A_{s}	Austenite start temperature
atan2	Four-quadrant arctangent function
C	Finger contact point
c	Constitutive model fitting parameter
d	DH joint distance parameter
DOF	Degrees of freedom
$\mathrm{DOF}_{\mathrm{contact}}$	Degrees of freedom of the relative motion at the contact interface
E	Elastic modulus
E^{c}	Elastic modulus of the constraint
G	Constitutive model fitting parameter
$\boldsymbol{G}_{\mathrm{PWM}}$	PWM duty gain vector
$\boldsymbol{I}$	Unit vector corresponding to x direction in world frame

$\boldsymbol{i}$	Unit vector corresponding to x direction in local frame
$\boldsymbol{j}$	Unit vector corresponding to y direction in local frame
$\boldsymbol{k}$	Unit vector corresponding to z direction in local frame
k_1	Constitutive model fitting parameter
k_2	Constitutive model fitting parameter
K_p	Proportional gain
K_v	Velocity gain
l	Constitutive model fitting parameter
l	Link length
l_δ	Effective link length incorporating contact point offset
$\boldsymbol{m}$	Direction vector in the $xy-z$ plane
m	Constitutive model fitting parameter
M_s	Martensite start temperature
n	Constitutive model fitting parameter
n	Trajectory step index
NDOFM	Necessary finger DOF for manipulation
NNFM	Necessary number of fingers for manipulation
n_p	Number of pure rolling contact points
n_s	Number of slide rolling contact points
n_t	Number of twist rolling contact points
O	Coordinate system origin
$\boldsymbol{p}_\mathrm{base}$	Actual contact point position in the local coordinate system

$\hat{\boldsymbol{p}}_{\text{base}}$	Desired contact point position in the local coordinate system
$\hat{\boldsymbol{p}}_{\text{global}}$	Desired contact point position in the global coordinate system
PR	Pure rolling
$\boldsymbol{PWM}_{\text{max}}$	Vector of maximum PWM duty cycle limits
$\boldsymbol{PWM}_{\text{min}}$	Vector of minimum PWM duty cycle limits
$\boldsymbol{PWM}_{\text{range}}$	Vector of PWM duty cycle ranges
$\boldsymbol{r}$	Contact point location vector relative to PIP joint
$\boldsymbol{r}$	Position vector of the local frame origin in the world frame coordinate system
s	Slope of the temperature dependence of the transformation start stresses in tensile psuedoelastic tests above A_f
SR	Slide rolling
T	Temperature
t	Time
TR	Twist rolling
T_{s}	Test start temperature
T_0	Effective equilibrium temperature
u_{fb}	Feedback component of control signal
u_{ff}	Feedforward component of control signal
u_{norm}	Normalized control signal
u_{sma}	Control signal applied to SMA consisting of feedback and feedforward components
$\dot{y}(k)$	Actual velocity at current time

$\hat{\dot{y}}(k)$	Desired velocity at current time
$\hat{\dot{y}}(k+1)$	Desired velocity at next time step
x_{c}	x coordinate of the finger tip contact point
xy_{c}	xy component of contact point position
$\boldsymbol{Y}$	Unit vector corresponding to y direction in world frame
$y(k)$	Actual position at current time
$\hat{y}(k)$	Desired position at current time
$\hat{y}(k+1)$	Desired position at next time step
y_{c}	y coordinate of the finger tip contact point
$\boldsymbol{Z}$	Unit vector corresponding to z direction in world frame
z_{c}	z coordinate of the finger tip contact point

Chapter 1

Introduction

Dexterous manipulation is an aspect of robotics in which multiple manipulators, or fingers, cooperate to grasp and manipulate objects [1]. This specialized area aims to develop a robotic hand that exhibits the same versatility as humans do. Since our goal is to mimic the function of the hand, it is logical to utilize our knowledge of the human hand to gain insight pertaining to the design of the artificial equivalent.

1.1 Biologically inspired technology

Biologically inspired (or *biomimetic*) engineering is based on the philosophy of examining natural systems as a source for design principles of synthetic systems. Such techniques have gained immense popularity in recent years as they provide the designer with a wealth of insight resulting from thousands of years of evolution, and as such these biological systems are typically highly optimized.

Reports of biologically inspired materials are becoming increasingly common; one radical example is the development of adaptive textiles that change colour and texture according to their surroundings in a manner that mimics the unique ability of the octopus

to alter the appearance of its skin, despite being colour-blind [2].

This project draws inspiration from several biological concepts to implement artificial muscles in the form of shape memory alloy actuators. The resulting biologically inspired system is developed with the eventual goal of achieving a compact and light-weight dexterous robot hand which overcomes the limitations associated with modern prosthetic devices.

1.2 Motivation and objectives

The successful development of industrial robots has focused largely on serial manipulators with custom end-effector tooling that are capable of autonomously performing specific manufacturing and handling tasks. These robots have greatly enhanced productivity for these tasks; however, they lack the capacity to perform general object manipulation. A dexterous robot hand inspired by the versatility of the human hand would have immediate application beyond mass production tasks ranging from complex telerobotic devices in remote or dangerous environments to compact yet fully-functional prosthetic devices [3].

Pioneering work in the field of dexterous robot hands established the kinematic, sensing and control, and grasp planning issues resulting in the development of numerous artificial hands including the Belgrade-USC hand, the Utah-MIT hand, and the Stanford-JPL hand [4]. However, wide-spread commercial adaptation of these devices has been primarily hindered by a lack of practical actuation technology [5].

As a result of this technology, artificial upper-body limbs have evolved into externally powered prostheses which are conventionally actuated by electric servomotors [6]. Although these motors achieve reasonable kinematic performance, their major limitation is that they are voluminous and heavy [7]. As illustrated in Figure 1.1, the size of the motor limits the function of hand to a single degree of freedom grasping operation. The

Figure 1.1: Myoelectric prosthetic hand actuated by a single electric servomotor [19]

excess weight of the device is a major source of patient discontent: a recent survey on the satisfaction with prosthetic devices indicated that approximately 23% of the participants were dissatisfied with the weight of their prosthetic limb [8]. Deterring factors such as these lead to 30% to 50% of upper extremity amputees avoiding the use of their prostheses [9].

Given that a considerable portion of the population relies on prosthetic devices to overcome the difficulties associated with the loss of a limb through congenital defects, trauma, or disease (it has been reported that annually 200–500 major amputations are performed per million people [10]), there exists a clear need for functional prosthetic devices that are compact and light-weight.

In an effort to improve the functionality and reduce the weight of these prosthetic devices, the actuation function provided by the electric motor has been replaced by various actuators such as electroactive polymers and McKibben pneumatic artificial muscles with limited success [11, 12]. Alternatively, shape memory alloys (SMA) have been successfully implemented in many biomedical applications [13], and extensive research has been reported on their use as artificial muscles in both straight tensile [7, 9, 14, 15, 16] and hybrid composite [17, 18] configurations with promising results.

1.3 Contributions

This project investigates many of the issues surrounding SMA technology such as implementation obstacles (such as strain limitations and semi-passive time response) and modeling, instrumentation, and control approaches (such as constitutive models, neural networks, and feedback control). These considerations are first explored via the design and evaluation of a compact, lightweight, and high-strain SMA ribbon artificial muscle for use in biomimetic applications. Two significant results are produced: first, the strain achieved at low loads exceeds that of the previously reported maximum of SMA based actuators by over 300%. Second, the actuator is capable of attaining a previously unattainable strain under considerable loading. These results are a promising advancement in the development of SMA based robotic devices.

The second part of the project explores the implementation of SMA as artificial muscle actuators in a light-weight dexterous robot hand. Design lessons encountered are reported, and experimental results conveying performance characteristics of the hand are also provided. A novel control law is proposed to minimize the deficiencies inherent to SMA behaviour. The dexterous hand apparatus and associated instrumentation will also serve as a facility for the future characterization of artificial muscles.

1.4 Thesis organization

This thesis investigates the issues surrounding the practical application of shape memory alloys as artificial muscles in a three fingered robot hand. First we review the function of the human hand and the kinematic requirements for manipulation. An overview of artificial hands is provided, followed by a discussion on shape memory alloys focused on the unique phenomenon known as the shape memory effect.

Second, the forward and inverse kinematics of the hand itself are explored by examining the fingers as separate serial manipulators working cooperatively, and the focus of the hand lies in the position control of the finger tip contact point locations (*i.e.* the complex issues of grasp planning and tactile sensing explored in [20, 21] require extensive perceptual knowledge of the robot's environment are therefore left for future work) [22]. This is followed by the design of the requisite instrumentation and control systems. Due to the highly nonlinear nature of both the SMA and the robot hand, alternatives to traditional PID control schemes such as neural networks are reviewed.

Finally, a large-strain SMA actuator is proposed and the concepts explored herein are applied to the design, manufacture, and evaluation of an SMA actuated robotic hand.

Chapter 2

Literature Review

This chapter establishes the fundamentals of the human hand structure and the associated performance criteria are explored for comparative purposes. Kinematic requirements for manipulation and existing artificial hands are then reviewed such that a novel robot hand may be synthesized. Smart materials are then introduced, and the phenomenology of shape memory alloys is discussed. Finally, various methods of controlling SMA are summarized.

2.1 The human hand

Bearing in mind the biomimetic design practice introduced in section 1.1, we will first examine the physiology of the human hand for inspiration to develop a synthetic hand.

2.1.1 Biomechanics

The hand is an intricate system with over twenty-five degrees of freedom, and the complexity of the system is further increased by coupling between the tendons that transmit motion to the fingers [23]. To better understand the entire system, we will focus on

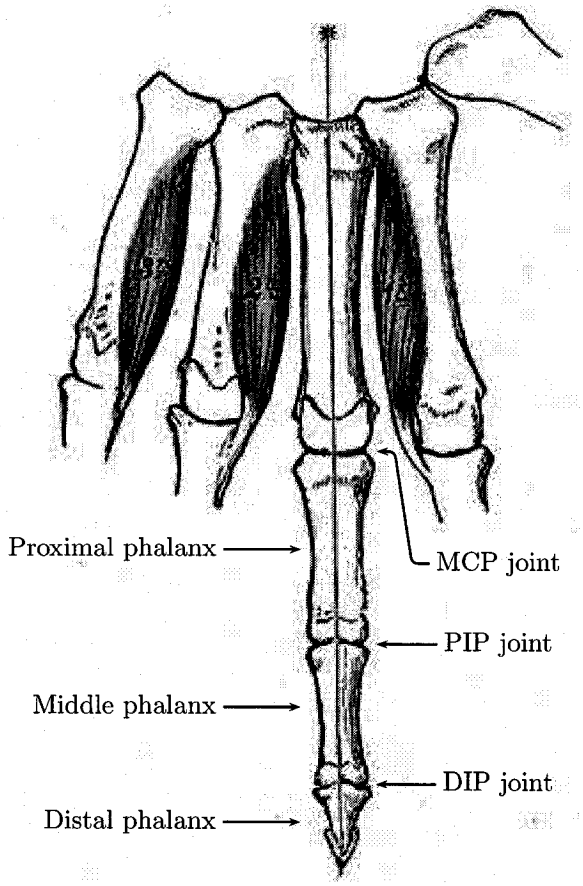


Figure 2.1: Skeletal structure of human finger [24]

understanding the mechanics of a single finger. The results of this study will give insight into the development of a functional artificial finger which when controlled cooperatively in the proximity of additional fingers will proximate the functionality of an artificial hand.

As shown in Figure 2.1, the fingers consist of 3 phalanges each (with exception of the thumb, which has only 2) which meet the metacarpals of the hand. The proximal phalanx meets the metacarpal at the metacarpophalangeal (MCP) joint, the proximal phalanx meets the middle phalanx at the proximal interphalangeal (PIP) joint, and finally the middle phalanx meets the distal phalanx at the distal interphalangeal (DIP)

joint [25, 26].

The MCP joints have two axes of motion that permit flexion-extension and abduction-adduction, while both the PIP and DIP joints have a single flexion-extension axis. These axes combine for a total of 4 degrees of freedom in each finger.

Movement in the fingers is produced by the contraction of muscles originating in the hand and forearm (the long finger is actuated by 6 muscles [27]), and this contraction is transmitted to the fingers by means of tendons. The tension in the tendons acts at a small displacement from the joint axis thereby inducing a moment that actuates the joint. The tendons are restricted by sheaths (called *retinacula*) that act as pulleys and ensures the tendons' proximity to the skeletal plane. This restriction maintains the moment arm and prevents the tendon from bowstringing across joints [28, 29].

Thus the human anatomy teaches two primary lessons:

1. Tendon based systems permit bulky actuators to be located remotely from the hand.

2. Tendon sheaths reduce interference between the transmission medium and the work envelope.

These principles will be implicitly incorporated into the design of the artificial hand.

2.1.2 Performance criteria

In addition to biomechanical insights, the human hand is the source of target performance criteria. The typical ranges of motion of the PIP and DIP joints 110°, and 65° respectively. Additionally, although certain movements involving the fingers occur quite fast, typical grasping movements only require finger joint angular velocities of 3–8 rad/s

Table 2.1: Sensory resolution of the human hand [30].

Property	Resolution	Conditions
Detection of movement of digits	0.5–1°	10–80°/sec velocity range
Reproduction of movement amplitude	3° error	0.1–65° range
Reproduction of digit position	5–9°	100–175° range
Digit position resolution	0.1°	
Detection of change in digit position	4–7°	

[30]. In terms of actuation capability, biological muscles contract between 1–100%, and can extert stresses of 0.007–0.8 MPa [2].

Jones has reported on the thermal, tactile, and motor sensory abilities of the human hand, and those of kinematic relevance are provided in Table 2.1 [30]. It follows that since the human hand is capable of functioning with these sensory resolution thresholds that an artificial hand should function subject to the same conditions. As such, the instrumentation of the system should be designed with these guidelines in mind.

Finally, although the human arm typically weighs approximately 3 kg from mid-humerous to the distal end, amputees consider prosthetic limbs weighing in excess of 1–1.4 kg unacceptable [30].

2.1.3 Kinematic parameter determination

Biggs and Horch have proposed a kinematic model of the human long finger that provides a set of kinematic parameters which approximate the mapping of finger joint angles into the fingertip location [27].

These parameters are based on measurements from a male cadaver, and estimates were statistically determined based on measurements from 1250 unique poses that the finger may normally assume. It will be shown in section 4.2 that the resulting kinematic parameters provide the geometric basis for the artificial fingers.

2.2 Kinematic requirements for manipulation

This section explores the design requirements (such as the number of fingers, and degrees of freedom per finger) and common kinematic configurations for artificial hands.

2.2.1 Artificial hand configurations

Although a variety of artificial hands have been developed, they may be classified into four categories according to kinematic function: opposition, flexion, cylindrical, and adduction-abduction [5]. The latter is of particular interest as this configuration satisfies the kinematic requirements for spatial manipulation.

Each configuration is kinematically unique despite having the same number of fingers, same number of phalanx per finger, and same number of joints per finger. Flexion-type hands are the simplest configuration, having all joint axes parallel permitting flexion only. This hand has limited grasping capability since each finger tip is constrained to plane motion. Cylindrical-type hands replace the first parallel joint axis with an orthogonal axis (parallel with the fully extended phalanges), which permits improved grasping capability. This hand configuration is, however, not found in nature. Adduction-abduction type hands are similar to flexion hands with the exception that the first joint axis is perpendicular to the remaining two. As implied by its name, this arrangement facilitates adduction and abduction of the fingers to improve the grasping capabilities of the hand. Finally, opposition-type hands most resemble human hands. This type of hand incorporates an added joint on the thumb that permits opposition, thereby increasing the possible number of grips the hand is capable of. Despite being the analog to the human hand, this design type suffers from a variety of complications hindering practical implementation. These complications mainly result from an incomplete understanding of human prehension in the context of grasp planning [25].

2.2.2 Design requirements

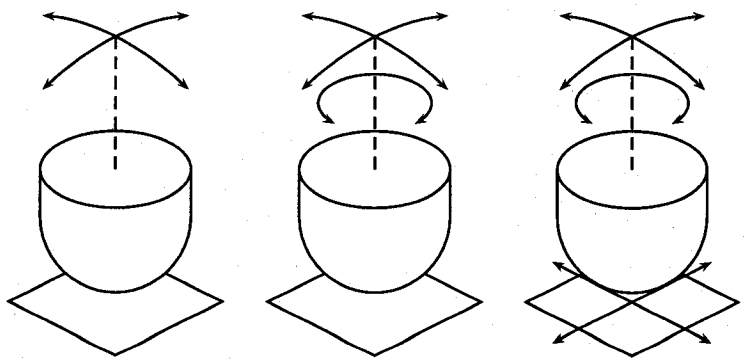
This section investigates several preliminary design requirements of potential artificial hand designs proposed by Nagashima [31]. These requirements include defining the type of relative motion constraint between the finger tips and the object, calculating the necessary finger degrees of freedom for manipulation, and finally computing the necessary number of fingers for manipulation.

Relative motion at the interface

Dexterous hands are distinguished from traditional industrial robots in that they have the capability to both grasp (restrain the motion) and manipulate (reposition and reorient) an object. Unlike manipulator motion analysis that is based on absolute position, finger-based systems manipulate an object by movement of the contact point between the finger surface and the object. In order to facilitate analysis, the object and finger are often assumed to be rigid and convex to ensure point contact. By making these assumptions (which are generally realized in the physical implementation), relative motion between the fingers and object can be categorized into three fundamental contact motion types: pure rolling (PR), twist rolling (TR), and slide rolling (SR).

As illustrated in Figure 2.2(a), pure rolling is the situation when there is a high frictional resistance to both twisting and sliding. This resistance results in the rigid bodies acting as though they are hinged about the contact point and consequently this type of motion results in two degrees of freedom (DOF). Due to the no-slip condition at the point of contact the surface velocities are equal.

Twist rolling, indicated in Figure 2.2(b), is typical at the interface between a pair of rigid bodies where there exists high resistance to relative slip, but little resistance to twisting. TR contacts have three degrees of freedom. Finally, the case of slide rolling,



(a) Pure rolling (b) Twist rolling (c) Slide rolling

Figure 2.2: Relative motion at the contact point interface [31].

illustrated in Figure 2.2(c), is analogous to a slider on a cam where there is little friction between the two contact surfaces resulting solely in a normal force.

Through experimentation or experience, the designer can specify the type of contact anticipated for a given task. Once the type of contact is established both the number of degrees per finger and the total number of fingers required can be evaluated.

Necessary Finger DOF for Manipulation

In order to impart an arbitrary position and orientation to the grasped object (realizable within the space of the finger), the sum of finger joint DOF and relative motion DOF at the interface between the finger and object being manipulated must be equal to the total number of DOF in the space in the manipulator. Therefore, the necessary finger DOF for manipulation (NDOFM) is simply the number of degrees of freedom in the space of the manipulator (6 in $\mathbb{R}^3$) less the DOF of the contact motion. For example, consider the TR type of contact motion consisting of three DOF in which the required degrees of

freedom each finger must have is:

$$\begin{aligned} \text{NDOFM} &= 6 - \text{DOF}_{\text{contact}} \\ &= 6 - 3 \\ &= 3. \end{aligned} \tag{2.1}$$

Thus any kinematic configuration considered when modeling an artificial hand must have at least 3 finger degrees of freedom when TR conditions are assumed [31].

Necessary Number of Fingers for Manipulation

The minimum necessary number of fingers to arbitrarily manipulate (NNFM) an object is given by equation 2.2, where n_{p}, n_{t}, and n_{s} are the number of pure rolling contacts, twist rolling contacts, and slide rolling contacts respectively:

$$4n_{\text{p}} + 3n_{\text{t}} + n_{\text{s}} \geq 6. \tag{2.2}$$

Considering the case of TR contacts only (presented in the previous section), n_{p} and n_{s} are both zero and the number of TR contacts required is determined to be greater than or equal to 2.

To understand the physical significance of this result, consider the case of rigid fingers grasping a rigid sphere. For the case when n_{t} is 2, the sphere will be constrained from translation, but the contacts cannot resist a twist motion and the sphere will be free to spin on the axis connecting the two points of contact. By increasing n_{t} to 3 the sphere becomes fully constrained and can no longer rotate. Thus any kinematic configuration considered for an artificial hand capable of basic object manipulation should have at least 3 fingers. Hands having more than 3 fingers introduces redundancy in grasping when

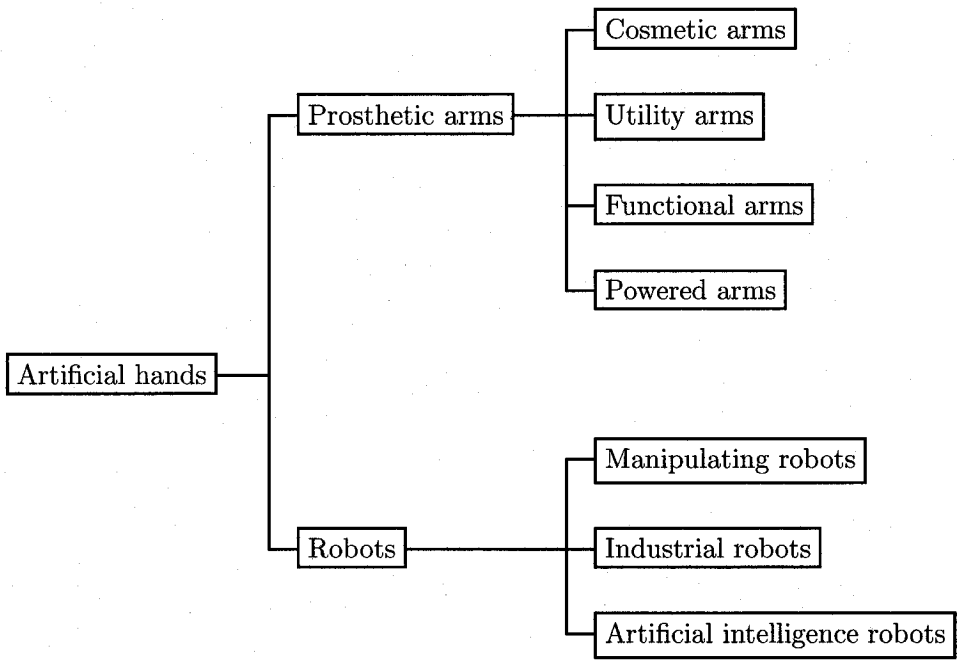


Figure 2.3: Classification of mechanical hands [32]

twist rolling conditions are present and are capable of performing more complex tasks such as object regrapsing, however these conditions are beyond the scope of this study.

2.3 Conventional artificial hands

Tanie proposed a classification of artificial hands shown in Figure 2.3 [32]. The primary classification is based on the ultimate purpose for which the artificial hand is utilized. This distinction is made since the purpose of the hand influences its functional and structural properties, stemming from the fact that robots are typically intended to replicate a specific function, while prosthetic limbs are required to have broader functionality and are governed by additional operating restrictions such as compactness, weight, and cosmetic appearance [32].

2.3.1 Prosthetic devices

As implied by their name, cosmetic arms are strictly for appearance and do not provide any functionality to the user. Utility arms are typically equipped with hooks so that some limited functionality is provided to the user. Functional arms (also known as body-powered arms) permit the user to operate a simple grasping hook mechanism by means of a cable system anchored to a harness worn by the user. Finally, powered (or externally-powered) arms permit the user to control the artificial hand by means of a myoelectric signal at the human-prosthetic interface [6, 33].

Modern commercially available myoelectric devices (such as the Otto Bock model DMC Plus Hand shown in Figure 1.1) have limited functionality due to their reliance on electric motors for actuation. The hand itself weighs 540 g (excluding the forearm attachment and battery), draws a peak current of 700 mA, and has an average power consumption of approximately 2 W when using a 7.2 V battery [19].

2.3.2 Robots

Robotic type artificial hands are categorized as either manipulating robots, industrial robots, or artificial intelligence robots. Manipulating robots are directly controlled by humans (for applications such as tele-surgery). Conversely, industrial robots are programmed to automatically perform repetitive production tasks. Finally, artificial intelligence robots are autonomous robots which are primarily used for research in machine learning.

According to Okamura, Smaby, and Cutkosky, the major chronological developments of robotic dexterous manipulation include the availability of the first myoelectric devices and the development of the Belgrade hand which had primitive pressure sensing capabilities in the 1960s [1]. These initial developments were followed by the formalizing of the

Table 2.2: Structural properties of selected dexterous robot hands [30, 34, 37, 38].

Robot hand	Number of fingers	DOF per finger	Actuation mechanism	Transmission mechanism
Utah/MIT hand	4	4	Pneumatic	Polymer tendons
Stanford/JPL hand	3	3	DC motor	Cables
Belgrade/USC hand	5	3[a]	DC motor	Linkages

[a] Coupled as a single DOF.

grasp problem and the introduction of the Grasp Jacobian which were implemented in the Stanford/JPL hand in the early 1980s [34, 35]. Later in the decade grasp and contact kinematics that tend to dominate the motion were investigated by Kerr and Roth [36]. Metrics quantifying grasp quality were introduced in the early 1990s which led to the introduction of various grasp optimization techniques. The most recent research has examined the autonomous exploration of unknown objects.

All of these developments have improved our understanding of both human and robot manipulation, however the robots themselves are not yet viable candidates for the next generation of prosthetic limbs as thus far they have been developed strictly as robotics research tools, and therefore have not been subjected to the additional requirements that apply to prosthetic devices previously discussed. Structural properties of the most well-known robotic hands are given in Table 2.2 and are illustrated in figures 2.4, 2.5, and 2.6 respectively. It is interesting to note that all the hands have their actuators remotely located from the hand. This practice mimics the biological equivalent by placing bulky actuators away from the fingers (as the bulk of our muscles are located in the forearm), thereby allowing for increased dexterity.

These prototypes represent some of the major advancements in the development of artificial hand design. In particular, the JPL and MIT hands represent some of the earliest tendon based robot power transmission systems which made it possible for their actuation systems to be located away from the hand. This arrangement is advantageous

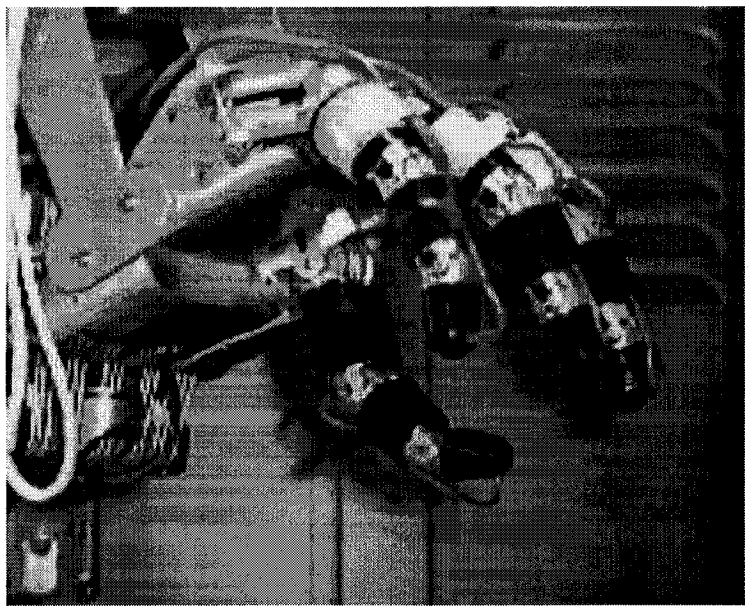


Figure 2.4: Utah/MIT hand [38]

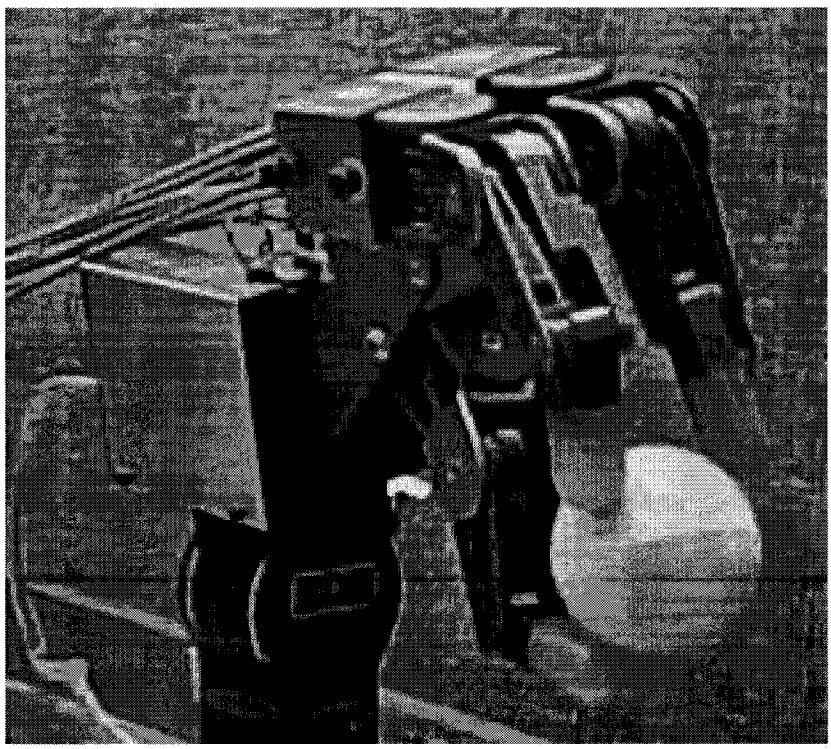


Figure 2.5: Stanford/JPL mechanical hand [38]

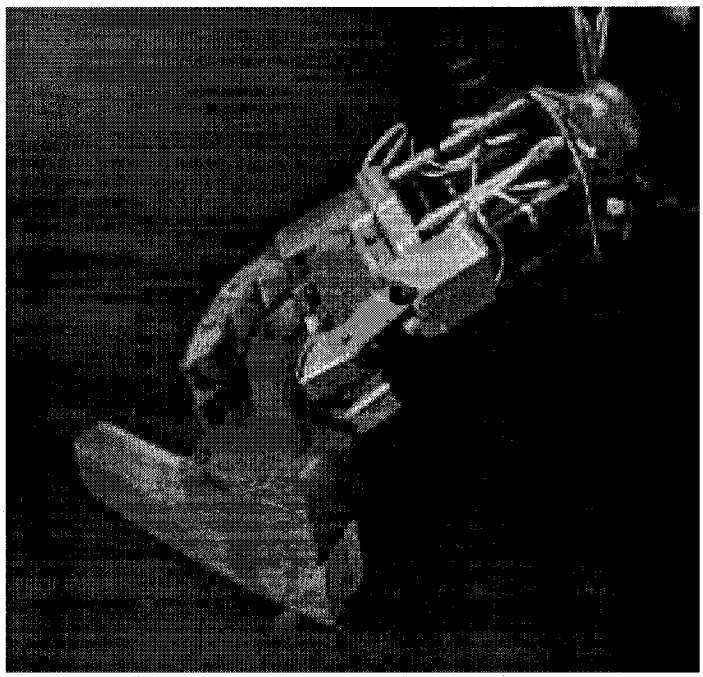


Figure 2.6: Belgrade mechanical hand [38]

for research tools in robotics, however these hands have not been successfully developed for prosthetic applications since ultimately the actuators must be carried by the patient, even if the actuators are removed from the hand itself. Because of this limitation, alternative lightweight actuation concepts must be explored.

2.3.3 Conventional Actuator Technology

Pons, Ceres, and Pfeiffer suggest that actuation technology has long been (and still remains) one of the most serious and long term impediments to the design of artificial hands [5]. In a study of the various actuator technologies for upper limb prostheses, Del Cura *et al.* suggest a classification system based on the working principle of conventional and nonconventional actuation technologies [6]. Conventional actuators include those based on electromagnetism (such as DC motors, coreless DC micromotors, brushless DC

micromotors, and servomotors) as well as those based on hydraulic and pneumatic systems (such as a standard piston and cylinder arrangement). Despite their high efficiency, these actuators are hindered by a low power density attributed to the relatively large mass associated with the actuator itself and the ancillary equipment required to operate it. Because of this deficiency, nonconventional actuators based on phenomena related to the material's atomic structure (including piezoelectric actuators, ER and MR fluids, and shape memory alloys) are being investigated as potential candidates for the next generation of prosthetic limb actuators.

2.4 Smart materials

The use of smart materials based devices is of vital importance in many fields. Smart materials act as both a sensor and actuator, and as such produce direct inherent mechanical responses to signals such as temperature, voltage, pressure, magnetic fields, and light. The ability to use a very simple mechanism to produce precise mechanical actions in response to specific stimuli or control signals can dramatically improve the overall performance of many smart material based solutions. Recent advances in the biotechnology industry have revealed enormous potential for the application of smart materials for biomedical use.

2.4.1 Typical smart materials and properties

Common examples of smart materials include piezoelectric materials, electrorheological (ER) and magnetorheological (MR) fluids, and shape memory alloys. Piezoelectric materials generate electricity when an external pressure is applied and conversely change dimensions when an external voltage is applied. Although these dimensional changes are

typically too small for macroscopic displacements (achievable strains of 0.1% are possible), operating frequencies in the kilohertz range are possible, which is ideal for vibration control applications in engineering structures [39, 40].

ER and MR fluids have viscous properties that are affected by the application of electric and magnetic fields respectively. A typical example of an ER fluid is a suspension of corn starch in silicone oil and simple MR fluids may be constructed from iron filings suspended in vegetable oil. Additional ingredients are typically added to the mixture to impede settling of the suspended particles. These fluids have been applied as clutches and variable stiffness vehicle suspension systems, but since they only offer a modifiable yield stress they are not suitable as displacement actuators [40].

Shape memory alloys are a class of metal alloys that exhibit the unique phenomena of shape memory effect and superelasticity. Early research performed by Ölander first documented the shape memory effect in AuCd alloys, however Buehler of the Naval Ordinance Laboratory is credited for the first sustained SMA research program focusing primarily on NiTi alloys in the early 1960s. Thus NiTi alloys are commonly referred to by the name Nitinol, which is an acronym for Nickel Titanium Naval Ordnance Laboratory [41]. As discussed in the following section, these alloys exhibit several beneficial properties such as appreciable linear strains and reasonable fatigue life, and as such are excellent candidates for artificial muscle actuators.

2.5 Shape memory alloys

The physical metallurgy of SMA has been extensively documented [42, 43, 44]. The process by which an SMA achieves actuation is through a reversible phase change from a low temperature phase (martensite) to a high temperature phase (austenite) triggered by introducing a temperature change or stress to the material. This research exploits the

shape memory effect (SME) of NiTi SMA to achieve actuation during this phase transformation. This section explores SMA actuators, the underlying phenomenon governing the shape memory effect, existing SMA actuated devices, and finally various methods of controlling SMA.

2.5.1 SMA actuators

The potential application of several alternative lightweight actuators as artificial muscles has been investigated including electroactive polymers and pneumatic muscles with little success [11, 12]. Alternatively, shape memory alloys have been successfully implemented in many biomedical applications [13], and a vast amount of research has been reported on their use as artificial muscles in both straight tensile [7, 9, 14, 15, 16] and hybrid composite [17, 18] configurations with promising results. The former configuration results in actuators that produce high force but are limited to low strain, while the latter configuration results in actuators that produce low force and achieve higher strains. The current maximum reported cyclic strain is limited to approximately 10% [17].

2.5.2 SMA phenomenology

The fundamental property of SMA that allows actuation to be achieved during the phase transformation is the shape memory effect. In particular, the properties of nickel-titanium alloys have been studied extensively [44]. These findings provide the basis for the ensuing discussion surrounding the properties and applications of SMA actuators.

At low temperatures, the martensite phase is present and large induced strains will result in an apparent residual strain, ε_r, of the SMA. As illustrated in Figure 2.7, the SME is exhibited when this plastically deformed alloy is heated until it transforms into the high temperature austenite phase. If, upon heating, the alloy is unconstrained and

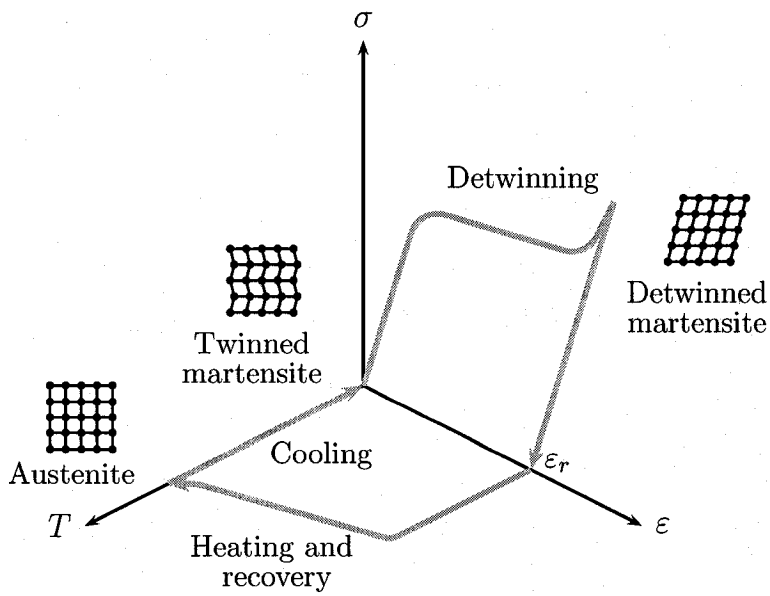


Figure 2.7: Shape memory effect for uniaxial SMA (adapted from [41])

low temperature deformations do not exceed the recoverable strain threshold, the alloy will return to its original shape once the transformation into the austenite phase is complete. If external forces are applied to the SMA during this heating process, work can be performed during the reverse transformation. Thus, the SME enables SMA to be used as an actuator [45]. The stresses generated can be as high as 700 MPa [46], which is much higher than the stresses encountered in biological muscles, which suggests that a lightweight solution may be feasible.

The actuation problem is complicated by the potential for the SMA to exhibit a one-way or two-way SME. Thermal cycling procedures can be used to impart either effect in an SMA. Neither the one-way or two-way SME can perform substantial work during the cooling phase of the cycle. In an alloy that exhibits the one-way SME, shape recovery is only possible during the heating portion of the thermal actuation cycle. In order to reintroduce strain to the actuator after the cooling phase (to return it to its original position), external work must be applied to the actuator. The two-way SME allows the

actuator to return to its original position without external work, however the induced strain that can be recovered on the heating stage of a cycle is much higher for the one-way than for the two-way effect.

SMA can be used in the design of artificial muscles in a variety of ways. In one arrangement of SMA based artificial muscles, SMA actuators are loaded in uniaxial tension and exert very large constrictive forces with small strain capability. Unfortunately there is no direct method for achieving the strain requirements of an artificial muscle using this configuration, especially when long life-cycle considerations are critical. Therefore, some form of compact mechanical advantage must be created to impart greater deformations at achievable strains.

2.5.3 SMA based devices

The concept of using SMA as actuators in artificial fingers and hands is not a new idea. A complex SMA hand prototype was proposed by Yamamoto in 1986 [47]. Although the hand has 17 degrees of freedom, it has a weight of 3.3 kg which is unacceptable for prosthetic applications according to the 1.4 kg limit imposed in section 2.1.2. The hand is also hindered by the requirement of a 70 mm diameter cylindrical forearm, which houses the SMA filaments and active cooling apparatus. The motion is transmitted from the SMA to the joints by means of stainless steel wire.

De Laurentis, Pfeiffer, and Mavroidis have proposed a preliminary finger design having 3 revolute joints that are actuated by SMA wires [7, 15]. Braided cables are utilized to transmit motion to the joint from SMA wires located on an artificial wrist, and extension springs spanning each joint provide the required recovery force. Experimental results indicate significant joint deflections of 30° and 40° for the PIP and DIP finger joints respectively, however no facility for automatic control has been developed, and

specific time response data has not been reported.

More recently, Bundhoo and Park suggest an SMA based finger that incorporates antagonistic muscles pairs [9]. The finger has 3 independent degrees of freedom, however unlike the design proposed by De Laurentis *et al.* there are 2 actuated DOF at the MCP joint, and a single actuated DOF at the PIP joint (the DIP joint is directly coupled to the PIP joint by means of a pulley system). Although the antagonistic muscle arrangement more closely resembles the function of biological systems, the required number of SMA actuators is doubled. This increase in the number of SMA wires reduces the efficiency of the system since energy is persistantly being dissipated in the form of heat in the inactive member of the pair (as opposed to the alternative arrangement of using energy conservative spring elements). This source of energy inefficiency is a major concern for prosthetic applications where the available energy is limited. No experimental data has been reported for this proposed configuration.

2.5.4 Methods of controlling SMA

Constitutive Models

Many control algorithms rely on system models to accurately predict the behaviour of the dynamic system. To this end, SMA have been the subject of many constitutive models that attempt to predict the non-linear and pseudo-elastic response the material exhibits as a function of thermomechanical conditions [48]. These models are generally formulated according to either microstructural phenomena (such as lattice cell mechanics, interface nucleation and propagation, and micromechanics) or macroscopic behavior (models without internal variables, models with internal variables, and hysteresis models) [49].

Among the more recent models, Peng, Yang, and Huang propose a model dependent

on internal variables that is based on the individual contributions of the martensite and austenite phases [50]. With this two-phase mixture model of the SMA thermomechanical behavior, the overall strain of the material is decomposed into three components: an elastic component, a plastic component, and a phase transformation component. In the work of Lu and Weng, the transformation phenomenon of SMA was modeled based on the contribution of stress (mechanical potential energy) and temperature (chemical free energy) to the Gibbs free energy [51]. This micromechanical theory demonstrates that the presence of an applied stress shifted the martensite concentration versus temperature hysteresis loop to the right during both heating and cooling, illustrating the interdependence of stress and temperature on the phase transformation of the material. Matsuzaki and Naito presented a one-dimensional model for SMA wires that accounted for thermomechanical interactions such as energy dissipation, latent heat, and heat transfer [52]. With this constitutive model, it is possible to examine the effect of strain rate on the stress-strain characteristics and the temperature of the wire.

Šittner, Stalmans, and Tokuda recently developed an extensive macroscopic one-dimensional constitutive model that dealt with the thermomechanical aspect of the SMA hysteresis [53]. This strain (ε) – stress (σ) – temperature (T) hysteresis caused by the dissipative nature of SMA is modeled based on SMA thermodynamics phenomenology and relies on seven characteristic material parameters:

- elastic modulus, E (GPa)
- coefficient of linear thermal expansion, α (K^{-1})
- maximum transformation strain, $\varepsilon^{tr,max}$
- martensite start temperature, M_s (K)
- austenite start temperature, A_s (K)

- slope of the temperature dependence of the transformation start stresses in tensile psuedoelastic tests above A_f, s (MPa K^{-1})
- reorientation stress, σ^{re} (MPa)

The model combines the notion of a martensite evolution function, $\xi(\varphi)$, and a traditional constitutive equation that accounts for elastic, thermal, and transformation strain. An internal variable φ is introduced to quantify the thermomechanical driving force on the martensite transformation and is defined as

$$\varphi = T - \frac{\sigma}{s}. \tag{2.3}$$

The differential kinetics equations proposed for the forward and reverse phase transformations are respectively given by:

$$\frac{d\xi}{dt} = \begin{cases} c\frac{G}{s}\xi^{l}\left(1-\xi\right)^{m+n} e^{-G(\varphi-T_0)\frac{d\varphi}{dt}}, & \frac{d\varphi}{dt} < 0 \\ c\frac{G}{s}\xi^{l+n}\left(1-\xi\right)^{m} e^{G(\varphi-T_0)\frac{d\varphi}{dt}}, & \frac{d\varphi}{dt} > 0. \end{cases} \tag{2.4}$$

The constants G, c, n, l, and m are fitting parameters. The variable T_0 denotes the effective equilibrium temperature and is defined as

$$T_0 = \left(M_s - \frac{\sigma^{re}}{s}\right)(1-\xi) + \left(A_s - \frac{\sigma^{re}}{s}\right)\xi. \tag{2.5}$$

The constitutive equation relating the strain to the stress and temperature is given by the sum of elastic, thermal, and transformation uniaxial strain components respectively

according to:

$$\varepsilon = \varepsilon^{\mathrm{el}} + \varepsilon^{\mathrm{th}} + \varepsilon^{\mathrm{tr}} \tag{2.6}$$

$$= \frac{\sigma}{E} + \alpha\left(T - T_{\mathrm{s}}\right) + \xi\varepsilon^{\mathrm{tr,max}} \tanh\left(k_1 \frac{\sigma}{\sigma^{\mathrm{re}} - k_2}\right), \tag{2.7}$$

where T_{s} is the test start temperature, and k_1 and k_2 are additional fitting parameters. Phase equilibrium in the stress-temperature space is defined by boundary conditions on the internal variable φ:

$$\varphi = \begin{cases} T - \frac{\sigma}{s}, & T > M_s \\ M_s + \frac{\sigma^{\mathrm{re}} - \sigma}{s}, & T \leq M_s \\ T + M_s - \frac{\sigma^{\mathrm{re}}}{s}, & \sigma \leq \sigma^{\mathrm{re}}. \end{cases} \tag{2.8}$$

These boundary conditions embody a simplification that allows the simulation of the response of SMA polycrystals to both low-temperature deformations in the martensite phase and low-stress thermal loads without introducing additional material properties.

Finally, a general macroscopic strain constraint, ε^{cc}, of the form

$$\varepsilon = \varepsilon^{cc}\left(\sigma, T\right) = \varepsilon^{c} + \frac{\sigma}{E^{c}} + \alpha^{c}\left(T - T_{s}\right) \tag{2.9}$$

may be introduced on the total strain which permits the simulation of several practical problems. Here ε^{c} is the macroscopic strain at the instant the constraint is applied, E^{c} is the elastic modulus of the constraint, and α^{c} is the coefficient of linear thermal expansion of the constraint. This model will be experimentally evaluated using both SMA wire and ribbon to investigate its potential role in an open-loop control system. Note, however, that this model requires extensive calculations using parameters that

are difficult to measure. Therefore, it will be difficult to implement and generally can not be evaluated in real-time [54]. Furthermore, the model is valid for uniaxial loading conditions, which are not strictly present as will be shown in section 4.2.

Feedback control

The constitutive models presented in the previous section provide a method of predicting what conditions must be present in order to obtain a given thermomechanical response from the SMA. Such approaches provide an open-loop model, but do not account for unforeseen errors and modeling uncertainty. A variety of feedback methods have been proposed for the control of SMA elements, however traditional PID based approaches remain the most commonly used [16, 7, 55, 56]. Due to the nonlinear nature of the system, the controller gains must be tuned empirically.

In addition to implementing a PID controller (or associated variation of PID), Ma and Song have proposed a position control method based on a control signal that is pulse width modulated (PWM), which has been shown to reduce the energy consumption of the SMA wire [55, 56]. A PWM signal appears as a train of successive pulses of constant amplitude (*i.e.* zero or full scale only), whose width is variable. The wider the pulse, the higher the duty cycle of the actuator. Such an arrangement is particularly advantageous in the case of prosthetic applications, as only a single voltage power supply is required eliminating the need for additional voltage amplification equipment.

Neural networks

A variety of approaches have been reported for the neural network control of shape memory alloys to overcome the implementation difficulties associated with constitutive models discussed in section 4.5. These techniques rely on one of three underlying network

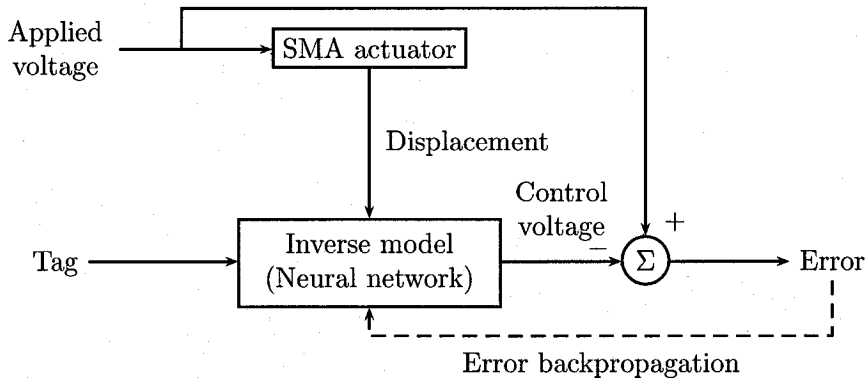


Figure 2.8: Open loop hysteresis compensation using an MLP

architectures: multilayer perceptrons (MLP), radial basis functions (RBF), and finally neuro-fuzzy systems (NF).

In their simplest form MLP networks have been implemented as hysteresis compensators in an open-loop scheme as shown in Figure 2.8. In a representative implementation, the MLP had two input nodes: the SMA displacement and an artificial tag to translate the one-to-two data correspondence of the hysteresis loop into a one-to-one relationship [57]. A single output neuron represents the required SMA activation voltage. The reported root mean square error for this system was 0.75 mm, corresponding to approximately 15% for the prescribed sinusoidal input trajectory.

This technique indicates promising results; however, its open-loop nature does not account for external disturbances to the system. Therefore, the control algorithm was modified by introducing a proportional-derivative (PD) feedback loop based on the displacement error signal. By doing so, the normalized error was reduced to approximately 7%. An interesting feature of the improved algorithm is that it uses the internal resistance of the SMA as the input signal, thereby eliminating the need for external position sensors.

Stevens and Buckner have applied networks composed of RBF to control SMA ar-

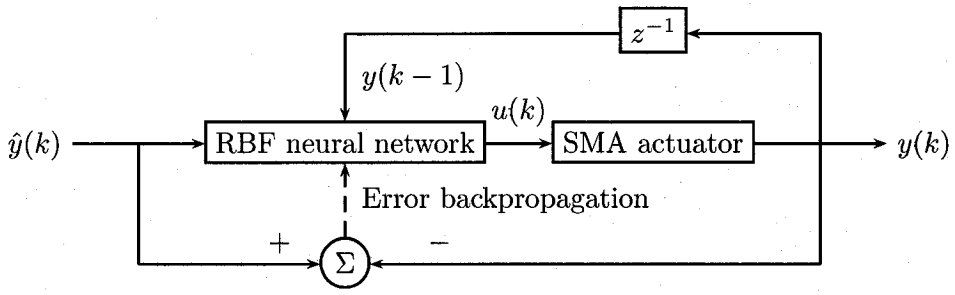


Figure 2.9: Position control scheme using radial basis function network

tificial muscles in a two degree of freedom surgical end-effector intended for minimally invasive surgery [58]. As shown in Figure 2.9, the network inputs consist of the desired displacement, $\hat{y}(k)$, and the previously measured displacement, $y(k-1)$, both of which are normalized to range between 0 and 1. The network output is the control activation voltage applied to the SMA. Despite a minimal tracking error of 0.5 mm (corresponding to approximately 0.6% of the prescribed trajectory), the results indicate that trajectory tracking is persistently erratic, which is highly undesirable for precision tasks.

In addition to MLP and RBF based implementations, NF based approaches have been proposed that combine neural network techniques with fuzzy logic methods to model SMA actuators [59]. This approach is remarkably versatile, as it can be applied to any SMA wire or actuator. As shown in Figure 2.10, the inputs to the network are the present desired position, $\hat{y}(k)$, the future desired position, $\hat{y}(k+1)$, the present desired velocity, $\dot{\hat{y}}(k)$, and the future desired velocity, $\dot{\hat{y}}(k+1)$. By including the future desired states, the technique of introducing an artificial tag variable previously discussed is encoded in the training data. The network output consists of the analog activation voltage which is added to the control signal from the PD controller and applied to the SMA. Experimental results for this type of controller indicate a more stable response and an improved response time to transient segments in the desired trajectory. These

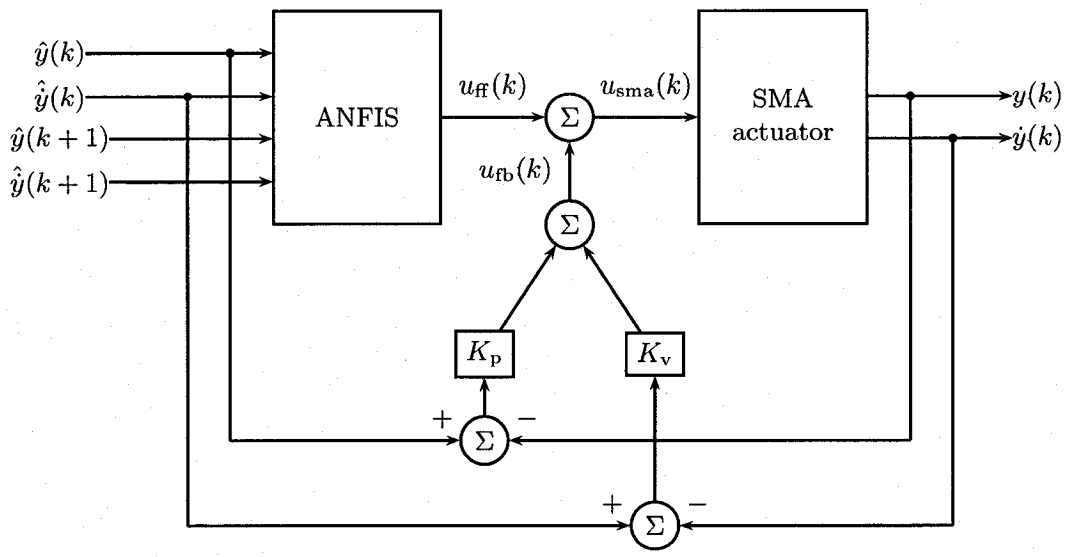


Figure 2.10: Position control scheme using an adaptive neurofuzzy inference system with PD control

characteristics signify strong trajectory tracking ability — the experimental tracking error was shown to be less than 0.5 mm, which corresponds to approximately 6% error for the prescribed input trajectory.

Of the three types of controllers considered (MLP/MLP-PD, RBF, NF-PD), the hybrid NF-PD algorithm has exhibited the most desirable tracking performance, and the combination of open- and closed-loop feedback ensures robustness to external disturbances and changes in ambient environment.

2.6 Summary

Examination of the human anatomy revealed that tendon based systems are advantageous over systems having the plant collocated at the joint, and parameter models and kinematic performance guidelines used to assist in the design and evaluation of the artificial hand were also discussed. Shape memory alloys were identified as potential artificial

muscle actuators particularly due to the unique shape memory they exhibit. Finally, various alternative control methods were presented that facilitate the position control of shape memory alloy filaments.

Chapter 3

Kinematic Analysis and Design of a Three Fingered Robotic Hand

This chapter establishes the forward and inverse kinematics of an artificial finger, and introduces coordinate transformations that permit several fingers to share a common reference frame to facilitate trajectory generation in the global frame. Finally, the requisite control and instrumentation systems are proposed.

3.1 Forward kinematics

To facilitate the trajectory control and feedback operations of the artificial hand the kinematics must first be established. The Denavit-Hartenburg (DH) convention has been used to systematically construct the forward kinematics transformation matrices [60, 61]. The proposed finger structure is illustrated in Figure 3.1 with the corresponding DH axes assigned. The manipulator joints are referenced according to their biological equivalent: links O_0–O_1, O_1–O_2, and O_2–O_3 interconnect at the metacarpophalangeal (MCP), proximal interphalangeal (PIP), and distal interphalangeal (DIP) joints respec-

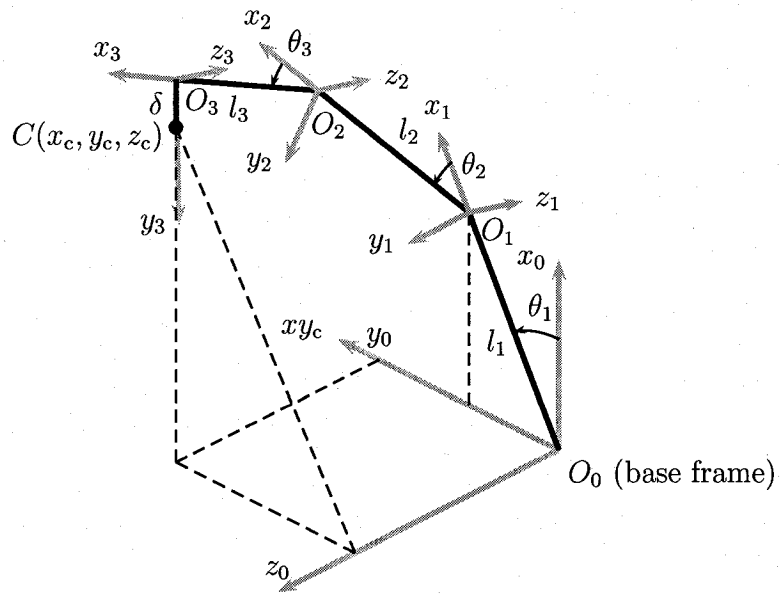


Figure 3.1: Kinematic structure of artificial finger.

Table 3.1: General Denavit-Hartenburg kinematic parameters for adduction-abduction finger.

Frame	θ_i (rad)	d_i (mm)	a_i (mm)	α_i (rad)
1	θ_1	0	l_1	$\pi/2$
2	θ_2	0	l_2	0
3	θ_3	0	l_3	0

tively. The MCP joint is constrained to move in the adduction-abduction axis, while the axes of the PIP and DIP joints permit flexion-extension. This configuration is based on the Stanford-JPL hand architecture, and the corresponding DH parameters are listed in Table 3.1 (the fingers are assumed to be of zero thickness to simplify the procedure; however, the method can readily be extended to the case of non-zero thickness). Such an architecture satisfies the kinematic constraints stipulated in section 2.2.2 when twist-rolling contact motion is assumed in the three finger system.

These parameters define the homogeneous transforms required to map the joint variables into the location of the finger tip contact point. Such a mapping constitutes the

forward kinematics of the finger, and performing the computation gives the contact point location, $C = (x_c, y_c, z_c)$, which is mapped into the base frame as point $(0, \delta, 0)$ in O_3 as:

$$x_c = \cos\theta_1(l_1 + l_2\cos\theta_2 + l_3\cos(\theta_2+\theta_3) - \delta\sin(\theta_2+\theta_3)) \quad (3.1)$$

$$y_c = \sin\theta_1(l_1 + l_2\cos\theta_2 + l_3\cos(\theta_2+\theta_3) - \delta\sin(\theta_2+\theta_3)) \quad (3.2)$$

$$z_c = l_2\sin\theta_2 + l_3\sin(\theta_2+\theta_3) + \delta\cos(\theta_2+\theta_3). \quad (3.3)$$

3.2 Inverse kinematics

The inverse kinematics determines the required joint variables to locate the manipulator contact point. Due to the unique configuration of the adducting-abducting finger, the ratio of the contact point position in the x and y directions are affected solely by θ_1. Analytically, this relationship is realized by dividing equation 3.2 by equation 3.1 which gives

$$\frac{y_c}{x_c} = \frac{\sin\theta_1\,(l_1 + l_2\cos\theta_2 + l_3\cos(\theta_2+\theta_3) - \delta\sin(\theta_2+\theta_3))}{\cos\theta_1\,(l_1 + l_2\cos\theta_2 + l_3\cos(\theta_2+\theta_3) - \delta\sin(\theta_2+\theta_3))} = \tan\theta_1. \quad (3.4)$$

It follows that the variable θ_1 can simply be evaluated using the four-quadrant arctangent function (atan2) in terms of the desired finger tip location as

$$\theta_1 = \text{atan2}\,(y_c, x_c)\,. \quad (3.5)$$

The determination of θ_2 and θ_3 is slightly more rigorous, but can be attained with the geometric projection shown in Figure 3.2. From Figure 3.1, the point xy_c can be introduced, which when combined with z_c, defines the location of the contact point in

the plane $xy - z$. Hence xy_c can be calculated from the desired finger tip position using Pythagorean Theorem as

$$xy_c = \sqrt{x_c^2 + y_c^2}. \tag{3.6}$$

We also introduce the variables φ_δ, φ, and l_δ which are defined as:

$$\varphi_\delta = \arctan(\delta / l_3) \tag{3.7}$$

$$\varphi = \theta_3 + \varphi_\delta \tag{3.8}$$

$$l_\delta = \sqrt{l_3^2 + \delta^2}. \tag{3.9}$$

As shown in Figure 3.1 the point C can be reached by traversing two alternative vector paths, which produces two expressions for the magnitude of the vector $\boldsymbol{r}$ which can be used to solve for $\cos\varphi$:

$$\cos\varphi = \frac{(xy_c - l_1)^2 + z_c^2 - l_2^2 - l_\delta^2}{2 l_2 l_\delta}. \tag{3.10}$$

Several possible values of φ satisfy this equation, so a second relation must be introduced to further constrain the angle. From trigonometry, the sine of the angle and the cosine of the angle are related by

$$\sin\varphi = +\sqrt{1 - \cos^2\varphi}, \tag{3.11}$$

from which φ can be determined using

$$\varphi = \operatorname{atan2}(\sin\varphi, \cos\varphi). \tag{3.12}$$

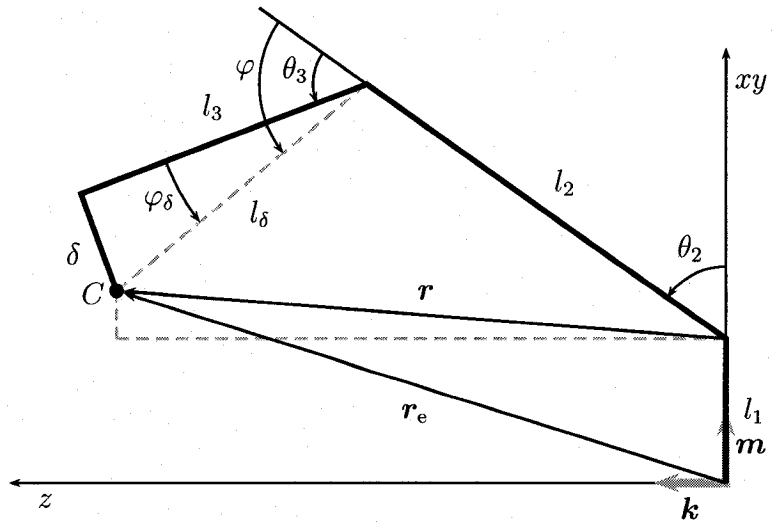


Figure 3.2: Kinematic structure of artificial finger in the $xy - z$ plane.

The joint variable θ_3 can then be determined by rearranging equation 3.8:

$$\theta_3 = \varphi - \varphi_\delta. \tag{3.13}$$

Finally, in order to solve for the remaining joint variable, θ_2, consider the vector $\boldsymbol{r}_e$ in Figure 3.2. In terms of the direction vectors $\boldsymbol{k}$ and $\boldsymbol{m}$, $\boldsymbol{r}_c$ can be stated as

$$\begin{aligned} \boldsymbol{r}_c &= xy_c\boldsymbol{m} + z_c\boldsymbol{k} && (3.14) \\ &= (l_1 + l_2\cos\theta_2 + l_\delta\cos(\theta_2+\varphi))\boldsymbol{m} + (l_2\sin\theta_2 + l_\delta\sin(\theta_2+\varphi))\boldsymbol{k}. && (3.15) \end{aligned}$$

Equating the vector components, then introducing the sum of angles trigonometric relations and expanding the terms yields a system of two equations which can be used to solve for $\cos\theta_2$:

$$\cos\theta_2 = \frac{xy_e - l_1 + \frac{z_e l_\delta \sin\varphi}{l_2 + l_\delta\cos\varphi}}{l_2 + l_\delta\cos\varphi + \frac{(l_\delta\sin\varphi)^2}{l_2 + l_\delta\cos\varphi}}. \tag{3.16}$$

Similarly to the formulation for φ, several possible values of θ_2 satisfy this equation,

and an additional relationship must be introduced to constrain the angle. In this case the equation is

$$\sin\theta_2 = +\sqrt{1-\cos^2\theta_2}, \tag{3.17}$$

where the positive root is indicative of the knuckle-up posture inherent to the finger. Finally, it follows that θ_2 can be solved according to

$$\theta_2 = \text{atan2}\left(\sin\theta_2, \cos\theta_2\right). \tag{3.18}$$

These equations constitute the complete inverse kinematics solution of the finger, and as such each of the joint variables can been determined to achieve a desired contact point position. Having developed both the forward and inverse kinematics of a single finger, the kinematics of the entire hand can readily be formulated. These equations can be used to calculate the required joint angles given a desired fingertip trajectory.

3.3 Coordinate transformations

The kinematics of a single finger can be extended to a hand by locating several fingers within a common coordinate reference frame called the world frame. The goal of this section is to develop a series of homogeneous transforms that will map desired finger contact points from the world frame into the base frame of the corresponding finger, at which point the inverse kinematics formulated for an individual finger remain valid.

The homogeneous transform that maps coordinates from the world frame into the base frame of a given finger, f, is given by:

$$\left[\boldsymbol{p}_f(x_\text{f}, y_\text{f}, z_\text{f})\ 1\right]^\text{T} = \boldsymbol{A}_\text{f}^\text{w}\left[\boldsymbol{p}_\text{w}(x_\text{w}, y_\text{w}, z_\text{w})\ 1\right]^\text{T}, \tag{3.19}$$

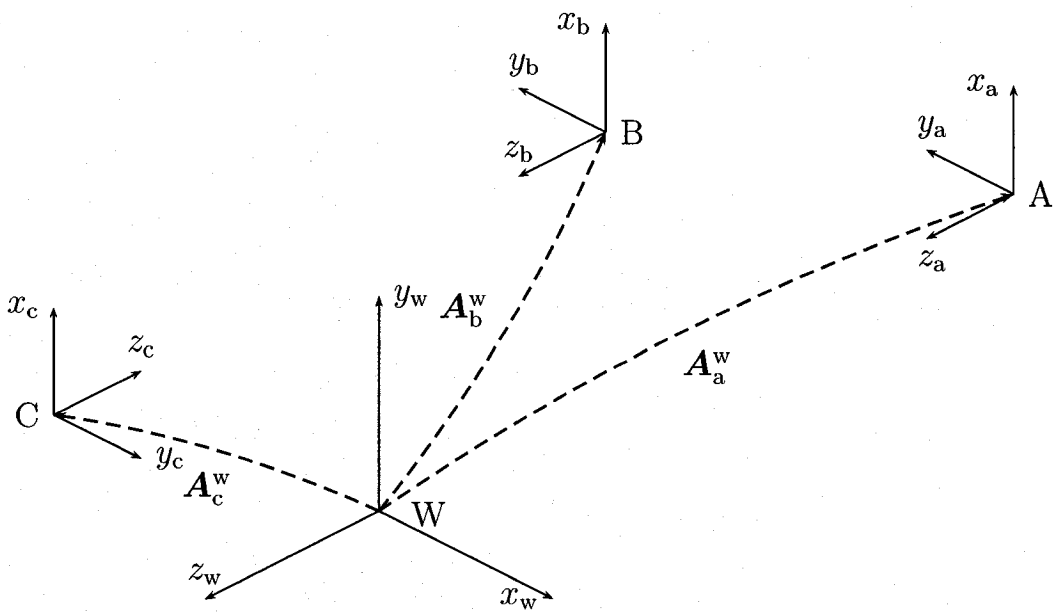


Figure 3.3: Coordinate transformations map coordinates from the world frame into each finger's local base frame

where:

$$\boldsymbol{A}_{\mathrm{f}}^{\mathrm{w}} = \begin{bmatrix} \boldsymbol{i}\cdot\boldsymbol{I} & \boldsymbol{i}\cdot\boldsymbol{J} & \boldsymbol{i}\cdot\boldsymbol{K} & r_x \\ \boldsymbol{j}\cdot\boldsymbol{I} & \boldsymbol{j}\cdot\boldsymbol{J} & \boldsymbol{j}\cdot\boldsymbol{K} & r_y \\ \boldsymbol{k}\cdot\boldsymbol{I} & \boldsymbol{k}\cdot\boldsymbol{J} & \boldsymbol{k}\cdot\boldsymbol{K} & r_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.20}$$

In the case of the opposition-type hand configuration shown in Figure 3.3, a transform can be derived for each finger to map each contact point from the world frame into the corresponding finger's local base frame:

$$\boldsymbol{A}_a^w = \begin{bmatrix} 0 & 1 & 0 & r_{x,a} \\ -1 & 0 & 0 & r_{y,a} \\ 0 & 0 & 1 & r_{z,a} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.21}$$

$$\boldsymbol{A}_b^w = \begin{bmatrix} 0 & 1 & 0 & r_{x,b} \\ -1 & 0 & 0 & r_{y,b} \\ 0 & 0 & 1 & r_{z,b} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.22}$$

$$\boldsymbol{A}_c^w = \begin{bmatrix} 0 & 1 & 0 & r_{x,c} \\ 1 & 0 & 0 & r_{y,c} \\ 0 & 0 & -1 & r_{z,c} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.23}$$

where $\boldsymbol{r} = (r_x, r_y, r_y)$ denotes the position vector of the local frame origin in the world frame coordinate system.

These transformations permit individual specification of finger contact point positions in the world frame, and therefore can be used to map contact point trajectories from the world frame into the finger's local coordinate system such that the inverse kinematics can be applied to obtain the required joint angles.

3.4 System instrumentation

An overview of the instrumentation subsystem is shown in Figure 3.4. The instrumentation of the hand is comprised of a high-level controller executed on a PC. The high

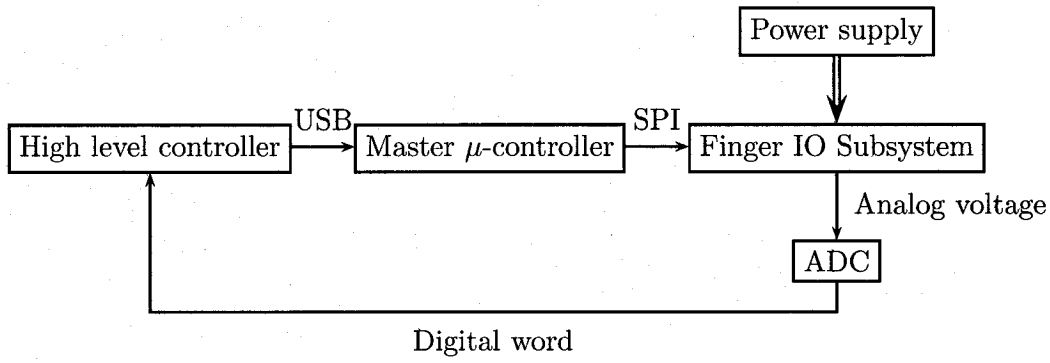


Figure 3.4: Hand system instrumentation diagram

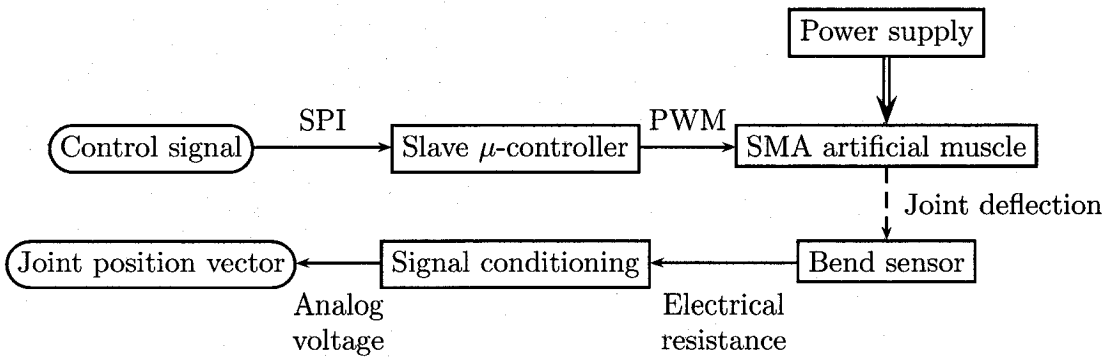


Figure 3.5: Finger IO subsystem

level controller calculates the required control signal which is then output to a master micro-controller. The master micro-controller facilitates communication between the PC and the hand via a USB bus, and relays the control signal to the appropriate finger IO subsystem via an SPI bus.

As shown in Figure 3.5, the finger IO subsystem is comprised of a slave micro-controller which accepts the incoming control signal, and converts the signal into an equivalent PWM output command. Power MOSFET drivers activate the SMA according to the PWM signal, thereby inducing a joint deflection.

Flexible bend sensors installed at each joint will be utilized to sense the joint de-

flections. These sensors provide a promising alternative to traditional options such as optical encoders, potentiometers, and vision systems in that they are very compact and portable, light-weight, add negligible resistance to joint motion, and their low profile does not interfere with the work envelope. Finally, the analog joint deflection signals are processed by an ADC data acquisition card and used as feedback by the high-level controller.

The bend sensors consist of a thin layer of electrically conductive ink deposited on a thin polymer substrate. As the substrate is bent, microcracks form in the ink which increases the electrical resistance of the sensor. After attempting several alternative approaches, the signal conditioning circuit illustrated in Figure 3.6 permits the variable resistance of the sensor to be converted into an analog voltage most compatible with the input range of the data acquisition system. The variable resistances R2 and R3 are tunable such that the resolution and dynamic range of the sensor can be maximized by eliminating the baseline voltage and rescaling the gain. The variable resistance S1 denotes the bend sensor. A schematic depicting the interconnections between the signal conditioning units is provided in appendix B.

Each SMA is driven by the circuit shown in Figure 3.7, and operational details are provided in section 4.4. A schematic depicting the interconnections between each SMA drive unit and the input specified by the slave microcontrollers are provided in appendix B.

3.5 Summary

The forward and inverse kinematics of the artificial finger have been developed. Coordinate transformations that map coordinates from each finger's local reference frame to a global frame were established. Finally, the proposed control hardware and instrumentation systems have been outlined to interface the mechanical system with the PC.

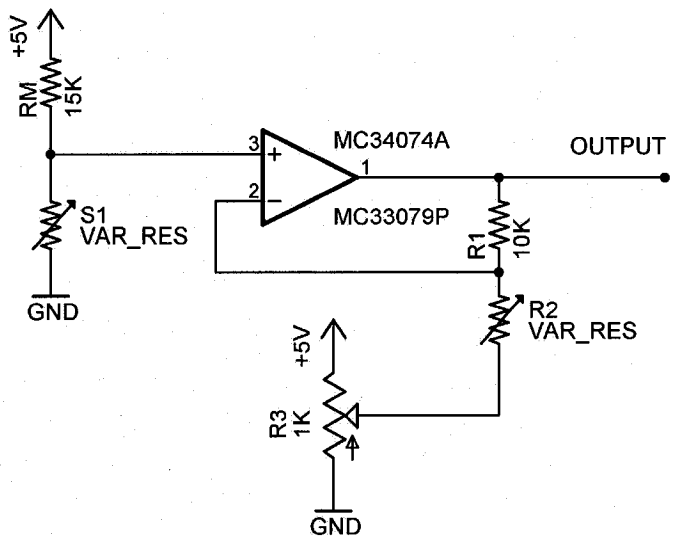


Figure 3.6: Sensor signal conditioning circuit

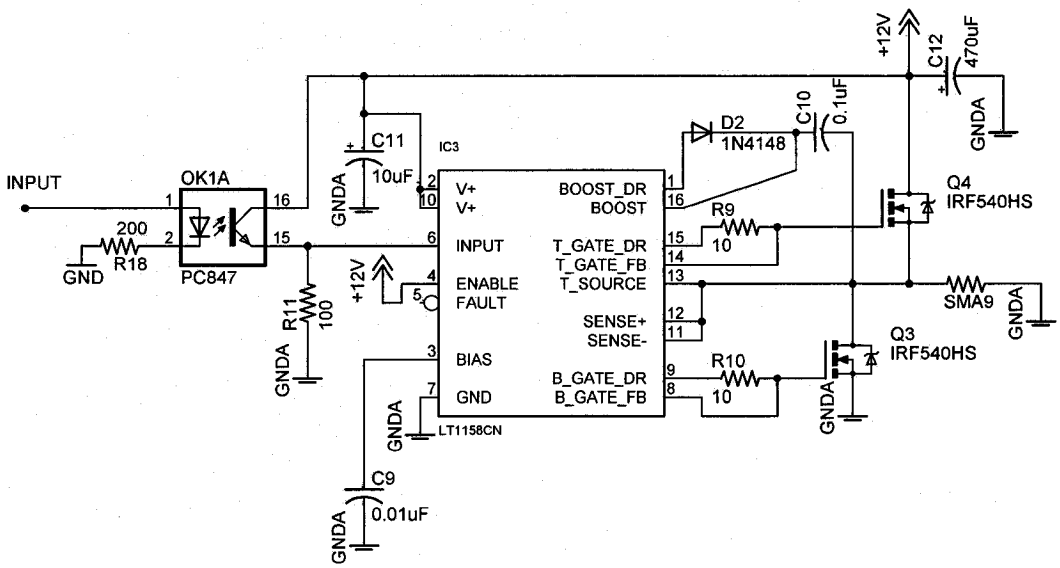


Figure 3.7: SMA drive circuit

Chapter 4

Materials and Methods

This chapter conveys the materials and methods used to manufacture the physical embodiment of the artificial hand. The SMA heat treatment is first discussed, followed by the development of a high strain SMA actuator. The kinematic properties, mechanical structure, and ranges of motion of the realized artificial finger are provided and three such fingers are then assembled to form a hand. Next, a constitutive model is evaluated to investigate its applicability for real-time control. Sensor digitization issues are then explored, followed by an explanation of the proposed high-level position control algorithm.

4.1 SMA metallurgy

Flexinol® wire of 0.015 in. diameter having a transition temperature of 70°C (Dynalloy, California) provides the necessary SMA actuation. In order to relieve residual stress and erase any existing memory in the material, the wire was installed in a fixture to maintain its linearity and annealed in a furnace for 2 hours at 600°C. Upon removal from the furnace, the filaments are quenched to minimize the formation of precipitates that are

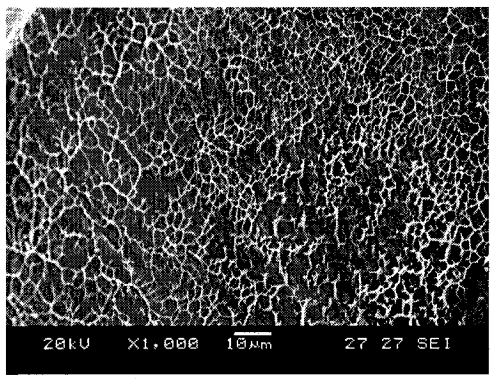


Figure 4.1: Flexinol® wire in original condition

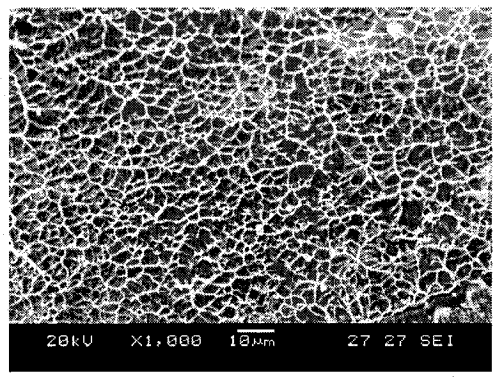


Figure 4.2: Flexinol® wire after annealing and quenching

detrimental to the extent of shape recovery. Figures 4.1 and 4.2 illustrate SEM images of the wire in the original and post-heat treatment conditions respectively. The wires are then elongated by 5% using an Intron tensile load frame and installed on the hand such that they contract to their original length when heated.

4.2 SMA enabled artificial finger

As discussed in section 2.1.3, artificial fingers were fabricated using a subset of the kinematic parameters determined in a model of the human long finger proposed by Biggs and Horch [27], and the associated DH parameters are given in Table 4.1.

Table 4.1: Actual DH Parameters.

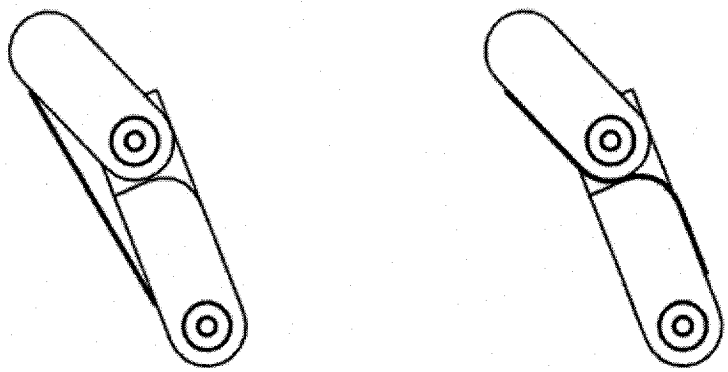
Frame	θ_i (rad)	d_i (mm)	a_i (mm)	α_i (rad)
1	θ_1	0	49.5	$\pi/2$
2	θ_2	0	32.0	0
3	θ_3	0	20.0	0

Contact point offset: $\delta = 6.35$ mm

The robot links are constructed from Teflon® (McMaster Carr, Ohio: 8735K42 and 8735K641) due to its high melting point and electrical insulation properties. Low friction revolute joints are constructed using precision steel shafts (Stock Drive Products/Sterling Instrument, New York: S40PX0-HG2-008) and miniature ball bearings (Boca Bearing, Florida: 99FR2-5ZZ) between successive links (refer to appendix A for detailed design drawings).

Socket set screws fitted with nylon tipped points (McMaster Carr, Ohio: 94115A107) are installed at the MCP joint of each finger and act as a locking mechanism to facilitate the sensor installation and calibration tasks. Cup point socket set screws (McMaster Carr, Ohio: 91375A112) are installed within the distal end of the proximal and middle phalanges and act as adjustable stops to precisely limit the range of motion of the PIP and DIP joints in their fully extended position.

The fingers are activated by SMA wires that are fixed at one end near the anterior distal end of each link, then pass along the link surface (toward the base of the finger) and across the interphalangeal joint where they are fixed to the posterior proximal end of the next link. The wire ends are clamped between the link surface and a stainless steel washer (McMaster Carr, Ohio: 90945A710) by means of cup point socket set screws (McMaster Carr, Ohio: 91375A112) and nylon hex nuts (McMaster Carr, Ohio: 94812A112) installed at the extreme ends of the link (and the base of the finger in the case of the MCP joint). As shown in Figure 4.3, this arrangement results in the activation of the SMA inducing



(a) **(b)**

Figure 4.3: SMA routing in a (a) traditional linear arrangement and (b) the implemented joint traversing arrangement

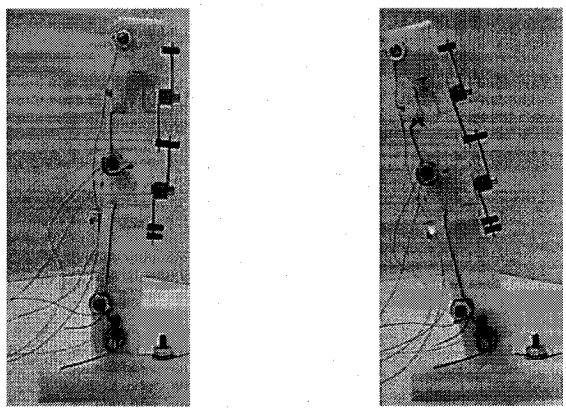
a joint flexion (by means of a length contraction of up to 5%), while minimizing the artificial muscles' interference within the work envelope when compared to a simple linear route. An additional advantage of this arrangement is that the length of the SMA wire used is increased, thereby increasing the maximum joint deflection for a given strain. Torsion springs of various stiffness (McMaster Carr, Ohio: 9271K84, 9271K75, 9271K79, and 9271K94) are affixed at each joint and provide the required bias force for the SMA, returning the joint to its un-actuated position when the SMA is deactivated.

The total mass of each finger including all shafts, bearings, sensors, springs, fasteners, and SMA wire is approximately 91 g. This indicates a considerable reduction in the total hand mass when compared to the conventional artificial hands reviewed in section 2.3.

Bend sensors are mounted at the MCP joint (Flexpoint Sensor Systems, Utah: PN2001: 2 in. base sensor, no overlaminate, clincher style connector), and the PIP and DIP joints (Flexpoint Sensor Systems, Utah: PN1001: 1 in. base sensor, no overlaminate, clincher style connector) using temporary adhesive, which readily permits repositioning of the sensors if necessary.

Table 4.2: Maximum joint range of motion in response to activation current of 2 A.

Joint	Coordinate	Displacement
MCP	θ_1	18.5°
PIP	θ_2	9.0°
DIP	θ_3	7.5°



(a) (b)

Figure 4.4: MCP joint range of motion

The maximum range of motion of the finger joints has been obtained by applying a current of 2 A to each SMA wire and the resulting data is provided in Table 4.2 [62]. The achievable range of motion of the SMA actuated MCP joint is illustrated in Figure 4.4.

The results in Table 4.2 indicate that the joint displacements do not meet the biological performance guidelines stipulated in section 2.1.2. As expected, the length of the wire is proportional to the maximum achievable joint displacement. Another limiting phenomenon is that although the activation of the SMA is relatively fast, the relaxation phase is governed by the cooling rate of the wire and as such is a passive process dependent on external heat transfer parameters.

4.3 SMA enabled artificial hand

An artificial hand was assembled by mounting 3 artificial fingers on a common base constructed of extruded aluminum channel. Each of the finger's base positions are adjustable in the horizontal plane such that the work envelope may be modified to accommodate a wide range of manipulation tasks. Figures 4.5 and 4.6 show the physical embodiment of the hand from alternate perspectives such that the details of the SMA routing, mechanical assembly, and sensor placement are evident for each joint.

In the configuration shown, the position vectors $\boldsymbol{r}$ that describe the translation between the world frame origin and each finger's respective local frame are:

$$\boldsymbol{r}_{\mathrm{a}} = (0, 0, 0) \tag{4.1}$$

$$\boldsymbol{r}_{\mathrm{b}} = (-5.65 \text{ cm}, 0, 0) \tag{4.2}$$

$$\boldsymbol{r}_{\mathrm{c}} = (-3.15 \text{ cm}, 0, 6.65 \text{ cm}) \tag{4.3}$$

4.4 System instrumentation

As described in section 3.4, the high-level control is executed by a PC (AMD Athlon™ XP 2100+ 1.73 GHz processor, 512 MB RAM). Communication between the PC's USB bus and the UART bus of the master micro-controller (Digi-Key, Minnesota: Atmel AVR-8 bit RISC microprocessor, ATMEGA32-16PC-ND) is facilitated by a specialized high speed communication bridge (FTDI Chip, Oregon: FT2232C). The slave micro-controllers (Digi-Key, Minnesota: Atmel AVR 8-bit RISC microprocessor, ATMEGA88-20PI-ND) were selected such that each had 3 available PWM channels in addition to the required SPI communication interface. Power MOSFET drivers (Digi-Key, Minnesota: LT1158CN-ND) synchronously control two MOSFET transistors (Digi-Key, Minnesota:

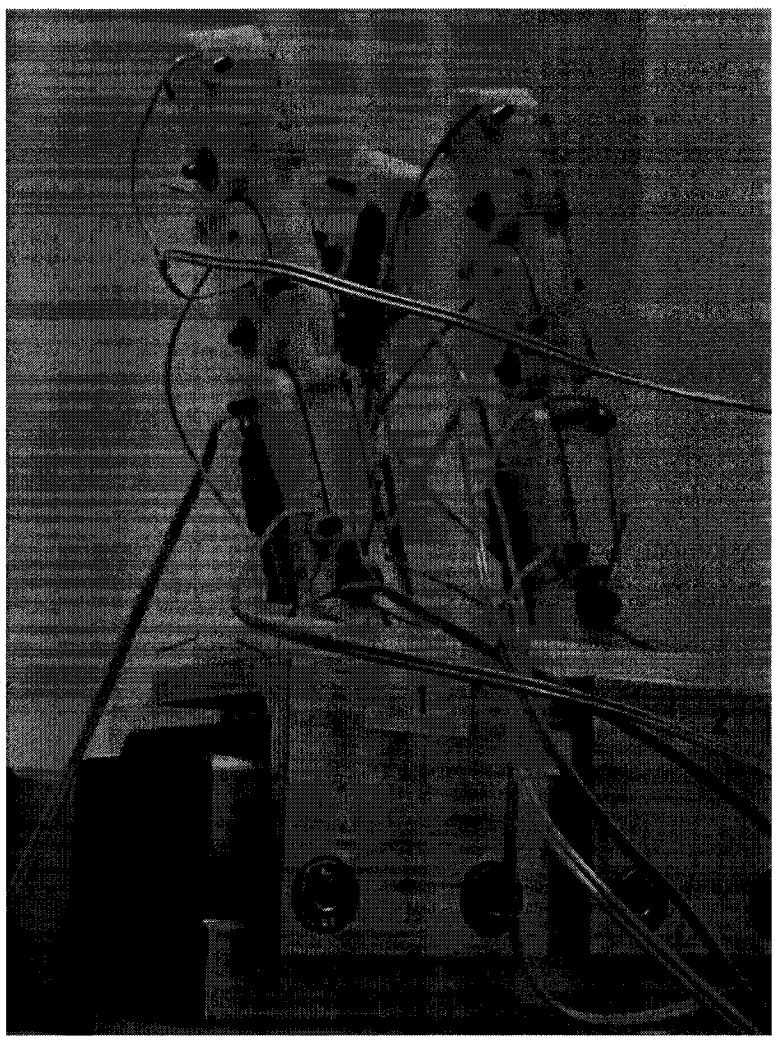


Figure 4.5: SMA actuated artificial hand

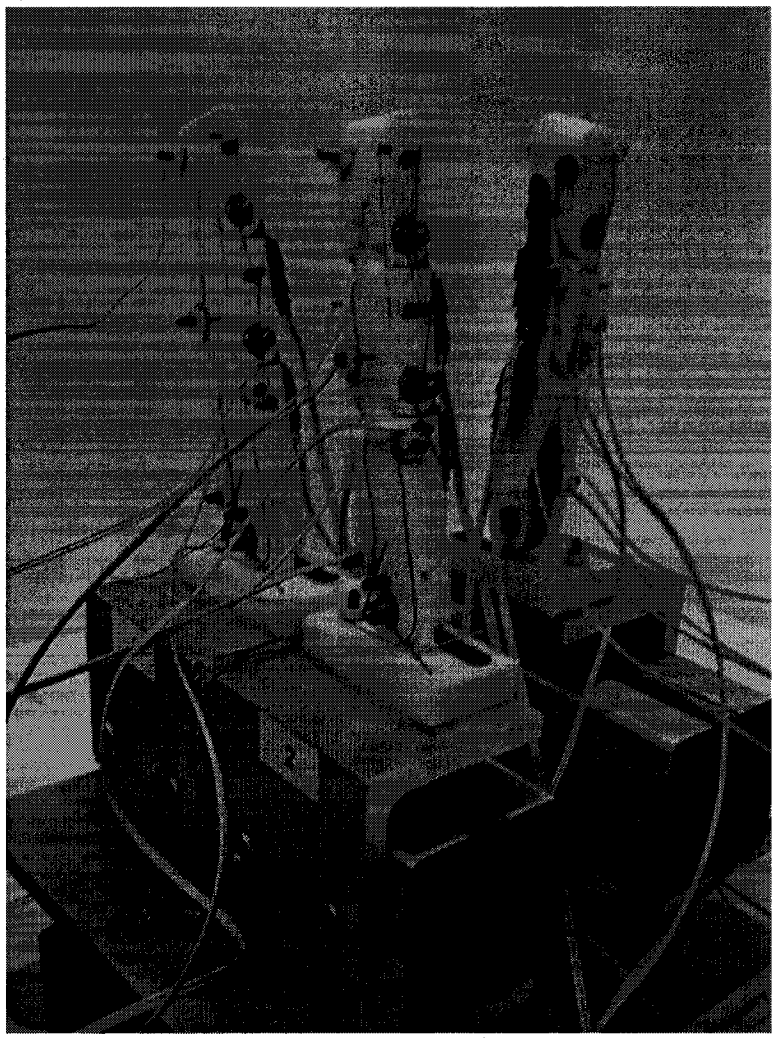


Figure 4.6: SMA actuated artificial hand (alternate view)

IRF3708L-ND) which in turn drive the SMA wire.

The micro-controllers are programmed using AVR Libc version 1.2.3 via a custom made parallel port DAPA interface. High-level control software has been written in MATLAB (version 7.1 R14SP3) in conjunction with the Data Acquisition Toolbox (version 2.7 R14SP3). Detailed code listings are provided in appendix C.

The digital and analog circuits are physically isolated from each other using optoisolators (Digi-Key, Minnesota: Sharp PC817) to protect the more expensive components should a fault occur. The digital circuits are powered directly from the USB bus, and therefore do not require an external power supply. Power is supplied to the analog drive circuits via a standard 5 V DC power supply. At this supply voltage, a balance is achieved in which reasonable response time of the SMA is obtained while minimizing the risk of overheating the SMA. For safety reasons the total maximum current delivered to the hand is limited to 3 A. In the worst case, the maximum power consumption of the 9 DOFs of the hand is limited to 15 W, which is approximately three times greater than the maximum power consumption of the single DOF Otto Bock hand.

Conversion of the bend sensor resistance into a voltage compatible with the DAQ board (National Instruments: PCI-6023E and SCB-68) is accomplished using operation amplifiers (Digi-Key, Minnesota: MC34074APOS-ND). All additional standard passive components indicated in appendix B were supplied from Digi-Key.

The electronic components have a small contribution to the overall mass: the largest components being the four micro-controllers which have a collective mass of approximately 13 g. With this in mind, the total mass of the electronic components is estimated to be less than 100 g. For prosthetic limb applications, the DAQ hardware and USB interface hardware are not required, and the cumulative mass of the three fingers and all remaining electronic components is under 400 g, which is a mass savings of over 26% com-

pared to the Otto Bock hand. This lightweight design is therefore very promising for the development of light-weight prosthetic devices that are subject to the mass restrictions stipulated in section 2.1.2.

4.5 Evaluation of constitutive model

The validity of Šittner's model was first evaluated by comparing the model output with the experimentally determined thermomechanical response of Flexinol® wire at various temperatures. Samples were prepared from 0.381 mm (0.015 in.) diameter Flexinol® wire from (Dynalloy Inc., California), and 0.635 mm x 0.953 mm (0.025 in. x 0.0375 in.) nitinol ribbon (Johnson Matthey, San Jose, CA). The experiments were performed using an Instron tensile testing machine outfitted with an environmental furnace. The strain and stress response of the trials were calculated from the crosshead extension and load cell output respectively by means of a data acquisition computer. Figure 4.7 indicates that the model correlates with the experimental hysteretic results for temperatures below and above the phase transformation start temperatures (the corresponding modeling parameters are listed in Table 4.3.

The small signal-to-noise ratio indicated in Figure 4.7 is attributed to the low forces required to induce large stresses in the specimen. These forces were minute in relation to the capacity of the load cell, and as such were highly susceptible to electrical interference. A running-average filter was employed to reduce the effects of the noise.

Having determined the model to be accurate for wire specimens, the model was employed to predict the thermomechanical response of an SMA ribbon (Johnson-Matthey, California: 0.025 in. thick, 0.038 in. wide). As shown in Figure 4.8, the model predicts the shape of the hysteretic loading segment, however the unloading segment of each curve is not accurately modeled (the corresponding modeling parameters are provided

Table 4.3: Model parameters for Flexinol® wire tests

Parameter	Low Temp	High Temp
Effective martensite start temperature, M_s (K)	328	328
Effective austenite start temperature, A_s (K)	351	351
Transformation entropy coefficient, s (MPa K^{-1})	2.78	3.72
Transformation strain, $\varepsilon^{tr,max}$	0.06	0.07
Reorientation stress, σ^{re} (MPa)	162.3	398.8
Activation energy kinetics parameter, G (K^{-1})	0.458	0.196
Friction kinetics parameter, c (MPa K^{-1})	1.115	0.016
Transformation flow kinetics parameter, n	6.543	7.235
Reverse-MT interaction kinetics parameter, m	2.062	1
Forward-MT interaction kinetics parameter, l	1	1

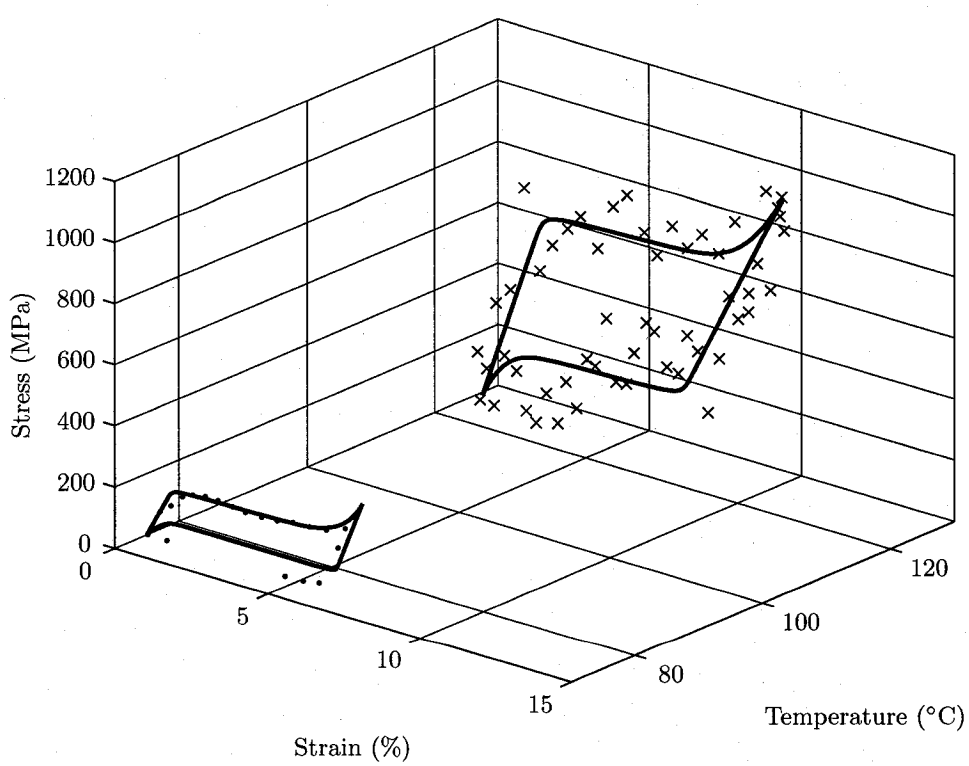


Figure 4.7: Flexinol® wire

Table 4.4: Model Parameters for nitinol ribbon tests

Parameter	Low Temp	High Temp
Effective martensite start temperature, M_s (K)	328	328
Effective austenite start temperature, A_s (K)	351	351
Transformation entropy coefficient, s (MPa K^{-1})	2.78	5.5
Transformation strain, $\varepsilon^{tr,max}$	0.05	0.09
Reorientation stress, σ^{re} (MPa)	162.3	272.9
Activation energy kinetics parameter, G (K^{-1})	0.456	0.179
Friction kinetics parameter, c (MPa K^{-1})	1.239	2.331
Transformation flow kinetics parameter, n	8	3.803
Reverse-MT interaction kinetics parameter, m	2.062	1
Forward-MT interaction kinetics parameter, l	1	1

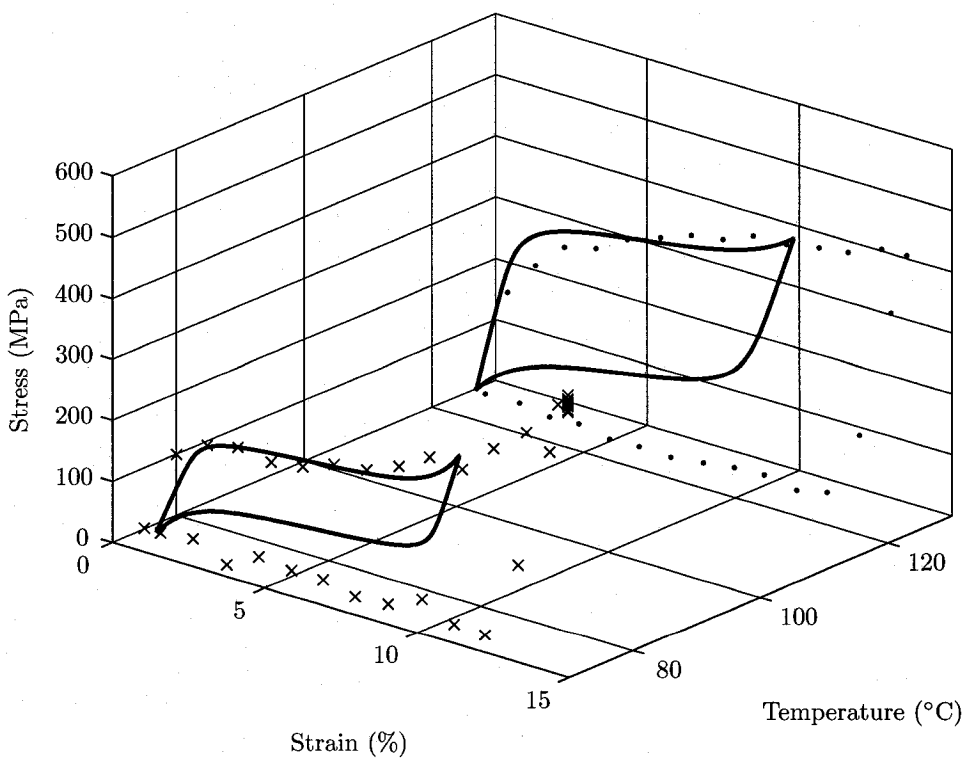


Figure 4.8: Nitinol ribbon

in table Table 4.4). The source of this disagreement requires further investigation, and a potential source of error includes exceeding the threshold of pseudoelasticity resulting in unrecoverable plastic deformation.

Although the experimentally determined thermomechanical response of the ribbon does not correlate with the model output, the former does coincide with the results reported by Otsuka and Ren that investigate the effects of thermomechanical treatment on shape memory characteristics [43]. These results indicate that the discrepancy may be attributed to simplifications inherent to the model. As indicated in development of the model, the effects of certain material characteristics such as precipitation effects and grain size are not explicitly captured, and as such the fitting procedure may not account for the effects of the thermomechanical treatment imposed on the ribbon.

In addition to these discrepancies, the use of this model as a feedforward control component is further limited by two major factors: first, the model is restricted to simple uniaxial loading conditions, which are not present in the proposed artificial finger configuration. Second, the model requires extensive computation, and is therefore not well-suited for real-time control. Due to these deficiencies, alternative control approaches must be explored.

4.6 Sensor quantization

The bend sensors located at each joint were calibrated by mounting custom-made transparent angle gauges graduated in degrees coincident with the joint axis. Typical calibration curves for the MCP joint and PIP joint are shown in Figure 4.9 and Figure 4.10 respectively (the DIP joint sensor output resembles the latter).

As proposed in section 2.1.2, the resolution of these sensors should be comparable to that of the human hand if we expect to obtain comparable performance. Consider

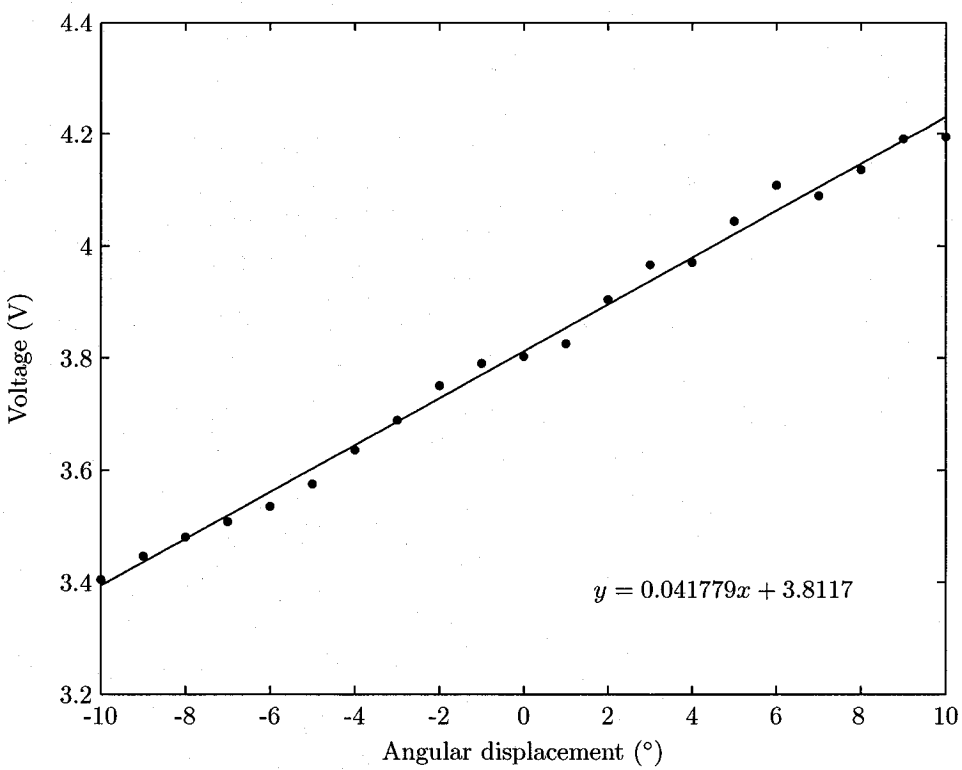


Figure 4.9: Typical MCP joint sensor calibration curve

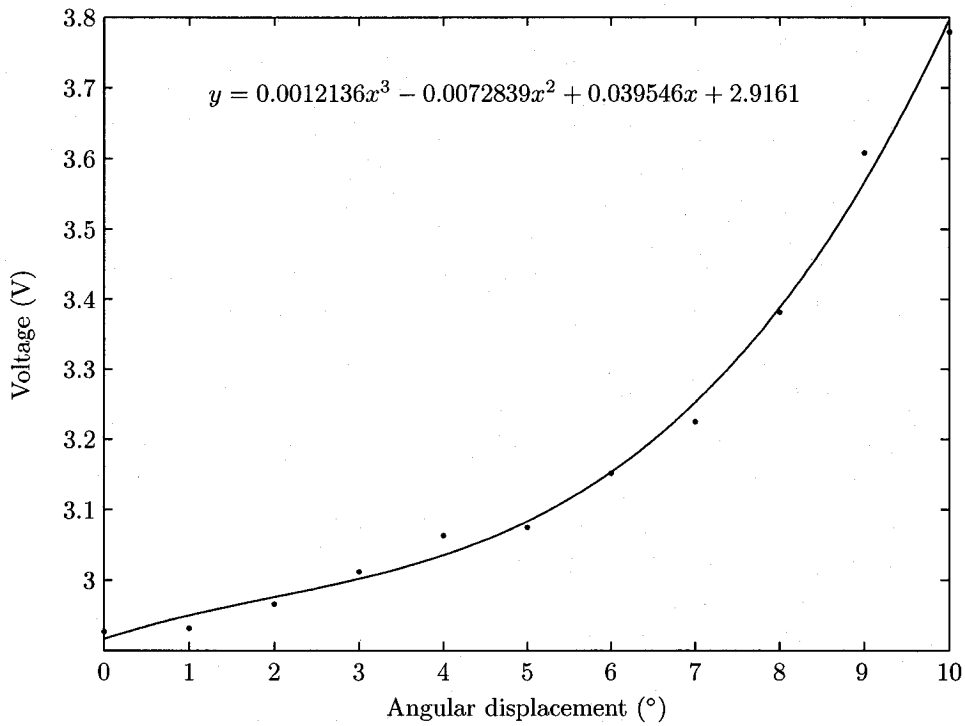


Figure 4.10: Typical PIP joint sensor calibration curve

first the MCP joint sensor output illustrated in Figure 4.9. Basic least-squares fitting indicates that a linear relationship adequately describes the relationship between the sensor output voltage (y) as a function of joint displacement (x). The slope of this line indicates the sensor gain, and in particular since

$$y = 0.041779x + 3.8117 \text{ V}, \tag{4.4}$$

then

$$\frac{dy}{dx} = 0.041779 \text{ V}/^\circ. \tag{4.5}$$

A 12 bit ADC is utilized to quantize the signal over a 10 V range (-5 V to +5 V), so the digital resolution can be computed as

$$\frac{10 \text{ V}}{2^{12} \text{ distinct digital values}} = 2.44 \text{ mV/distinct digital value}. \tag{4.6}$$

Combining the result of equation 4.5 and equation 4.6 gives the overall sensor resolution as

$$\frac{2.44 \text{ mV/distinct digital value}}{0.041779 \text{ V}/^\circ} = 0.06^\circ/\text{distinct digital value}, \tag{4.7}$$

which is slightly improved in comparison with the human digit position resolution of 0.1° suggested in section 2.1.2. An alternative configuration was considered wherein the integrated 10 bit ADC functionality of the slave micro-controllers would be used in place of the external data acquisition board, however the reduced digital accuracy resulted in an overall resolution of approximately 0.23°/distinct digital value, which is worse that the resolution of the human hand.

Both the PIP and DIP joint sensor output are best fit by a cubic polynomial and a

typicaly calibration curve is shown in Figure 4.10:

$$y = 0.0012136x^3 - 0.0072839x^2 + 0.039546x + 2.9161 \text{ V}. \tag{4.8}$$

Again, the slope indicates the sensor gain, however in this case the gain is also a function of joint position:

$$\frac{dy}{dx} = 0.0036408x^2 - 0.0145678x + 0.039546 \text{ V}/^\circ. \tag{4.9}$$

The minimum slope of the curve can be determined using simple calculus optimization, and indicates the region of minimum sensor resolution. In this case, the minimum gain was determined to be

$$\left(\frac{dy}{dx}\right)_{\text{min}} = 0.025 \text{ V}/^\circ. \tag{4.10}$$

Using the same digital resolution calculated in equation 4.6, the overall sensor resolution is given by

$$\frac{2.44 \text{ mV/distinct digital value}}{0.025 \text{ V}/^\circ} = 0.0976^\circ/\text{distinct digital value}, \tag{4.11}$$

which is 3% better than the reported human resolution (in the worst case), and is therefore sufficient for the artificial hand.

4.7 High-level control system

The neural network approaches reviewed in section 2.5.4 will not be implemented here as frequent network retraining to accommodate changes in external environmental conditions would not be permissible in the case of prosthetic applications. The sigmoid

function commonly used in neural network applications will however be used in the context of a nonlinear feedback controller.

Two major observations of SMA actuated systems contribute to the selection of the control algorithm: first, unlike motor actuated systems controlled by traditional linear PID methods (which negate the output signal when the incoming error signal is negative), a negative error signal should produce a zero output signal to permit the SMA element to cool and recover. Second, since the SMA wire contraction (which is governed by an active heating process) occurs much faster than the SMA extension (which is governed by a passive cooling process), the control law should minimize the amount of overshoot on the heating phase, even at the expense of time response upon heating.

In order to incorporate these two aspects into the control law, a nonlinear sigmoid function is used to map the joint error into a normalized control signal, which is then scaled via a PWM duty cycle gain vector to obtain the actual control signal. As shown in Figure 4.11, the sigmoid function has been shifted such that the output at zero error falls at the SMA activation threshold. Note that the threshold arises from data typecasting from a real number to an integer for use by the micro-controllers, such that a small real valued output signal will be typecast into an null activation signal. The sigmoid shift parameter vector, $\boldsymbol{\mu}$, should be tuned according to $\boldsymbol{\mu} = \ln(\boldsymbol{PWM}_{\max}-1)$ in order to match the minimum activation threshold with the zero error value. In addition, unlike traditional proportional control laws, the slope of the sigmoid function is not constant. This has the effect of producing a full scale output signal when the error is appreciable, and rapidly transitions to a small output signal when the error is small in order to avoid overshoot.

The sigmoid control law is then integrated into the high-level control algorithm shown in Figure 4.12. The main control loop initiates with the production of a trajectory

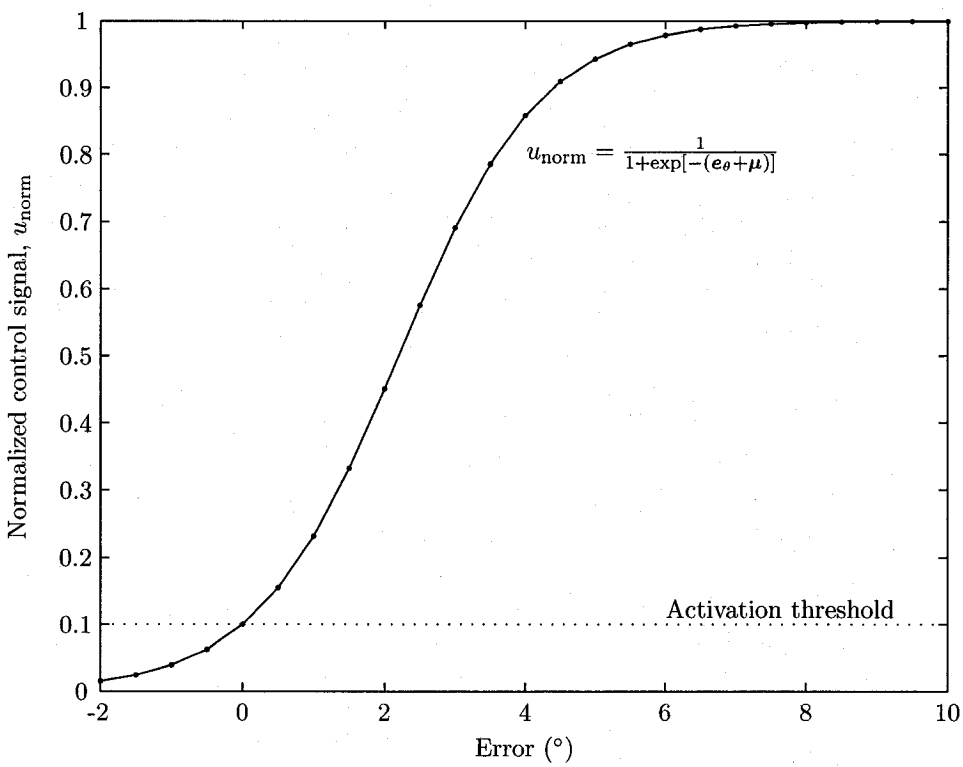


Figure 4.11: Sigmoid control signal

step n. The trajectory generator itself outputs a series of desired intermediate points, $\hat{\boldsymbol{p}}_{\text{global}}(n) = (\hat{x}, \hat{y}, \hat{z})$, whose successive proximity determines the degree of coordination of the fingers, as the fingers must be sufficiently close to the desired intermediate point before the trajectory generator will output the next desired location in the series. The error tolerance governing this step is given by κ (*i.e.* the finger tip contact point must be within κ of the desired contact point location before the trajectory generator will advance to the next desired trajectory point).

The output of the trajectory generator is defined with respect to the world frame, and so the coordinate transformations developed in section 3.3 are used to map these coordinates into the finger's local coordinate system, $\hat{\boldsymbol{p}}_{\text{base}}(n) = (\hat{x}, \hat{y}, \hat{z})$. The inverse kinematics developed in section 3.2 are then applied to translate the desired finger tip location into the required joint angles, $\hat{\boldsymbol{\theta}}(t)$. Next, the DAQ system acquires the bend sensor signals and decodes the information into the actual joint deflections, $\boldsymbol{\theta}(t)$. The actual joint angle vector is input to the forward kinematics derived in section 3.1 to obtain the actual finger tip end point location. From the actual and desired end point location an error vector can be computed, and the Euclidean norm of this vector is compared with the tolerance term κ to check is suitable convergence has been achieved. If so, the trajectory step n is incremented, but if not the joint error vector, $\boldsymbol{e}_{\theta}$, is input to the control law in order to obtain the normalized control signal, u_{norm}. The signal is then rescaled by the PWM duty gain vector, $\boldsymbol{G}_{\text{PWM}}$, and used to activate the SMA wires.

The DAQ system persists to acquire the actual joint deflections and the control loop iterates until the desired tolerance is achieved. Software safeguards are implemented to validate that desired contact point locations are in fact within the work envelope of the finger, and a maximum number of iterations is imposed should the point lie within the

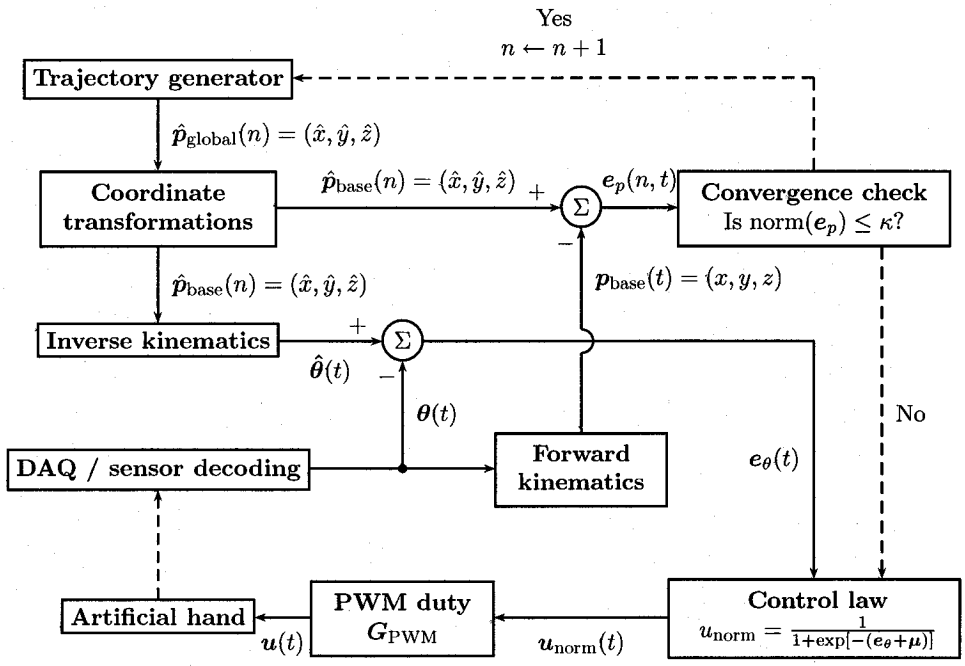


Figure 4.12: Single finger control algorithm

work envelope, but is unattainable due to restrictions on joint range of motion (attributed to the limited strain capacity of the SMA or mechanical stops at the joint).

The diagram shown in Figure 4.12 applies to the case of a single finger, however the process is readily extended to the entire hand by performing convergence checking on each finger before incrementing the trajectory step n to ensure coordinated movement of all fingers.

4.8 Summary

The manufacture of the artificial hand and the SMA heat treatment have been discussed. The hand's kinematic properties, mechanical structure, and ranges of motion have been discussed, and the fingers were arranged to form a hand. Constitutive models were shown to be insufficient for real-time control. Finally, the resolution of the bend sensors was

shown to be sufficient, and a sigmoid function based position control algorithm has been proposed.

Chapter 5

Development of a high strain actuator

5.1 Actuator design

Due to the relatively low strains exhibited by SMA with reference to biological muscles, the general need for a lightweight SMA based actuator with high strain capability was identified. An actuator was constructed by loosely braiding four strands of nitinol ribbon (Johnson-Matthey, California: 0.025 in. thick, 0.038 in. wide) as shown in Figure 5.1. Several twists were then induced into the braid to ensure uniform tension in each of the constituent elements.

A fixture was used to form the actuator into a compact figure-eight pattern by winding the braid around staggered pegs, forming alternating loops in the actuator. Once the actuator was installed into the fixture, the entire assembly was annealed at 600°C for one hour to eliminate residual stresses and to reset the memory of the material. Finally, the actuator was extended to full length by applying a small tensile load.

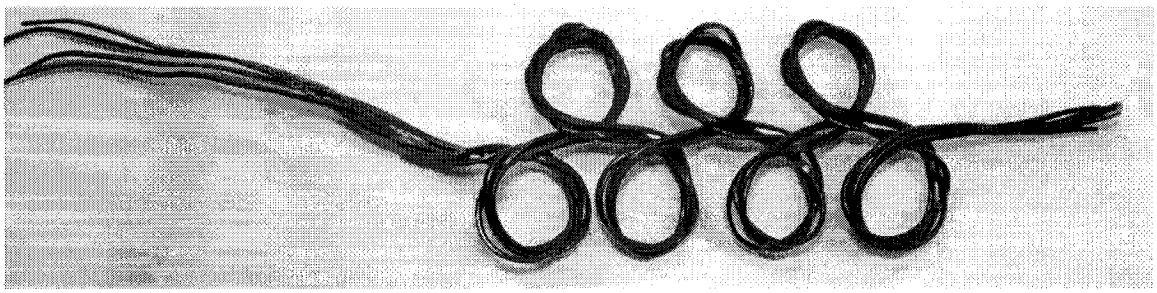


Figure 5.1: High strain braided SMA actuator

5.2 Actuator performance

Actuators can be fabricated with any number of loops depending on the scale of the mechanism in which they are to be installed. Increasing the number of loops increases the achievable strain for a given initial actuator length. A low-friction linear test frame was fabricated to evaluate the strain response of the actuator (Figure 5.2). Strain data was recorded via an integrated LVDT, and the SMA was activated by means of electric resistance heating.

The strain response of an actuator consisting of four loops has been recorded for several applied loads in order to demonstrate the typical performance characteristics. As indicated in Figure 5.3, each trial behaved similarly upon activation: an initial period of heating precluded a rapid contraction that asymptotically approached a maximum contractile strain. The jagged shape of the curves is attributed to the geometry of the actuator: reorientation of the loops upon heating resulted in a nonlinear strain response.

The maximum strain achieved under each load is tabulated in Table 5.1. The results indicated are significant, as the strain achieved at low loads exceeded that of the previously reported maximum of SMA based actuators by over 300%. A second important result is that the actuator was capable of attaining a previously unattainable strain under considerable loading. This result is a promising advancement in the development of

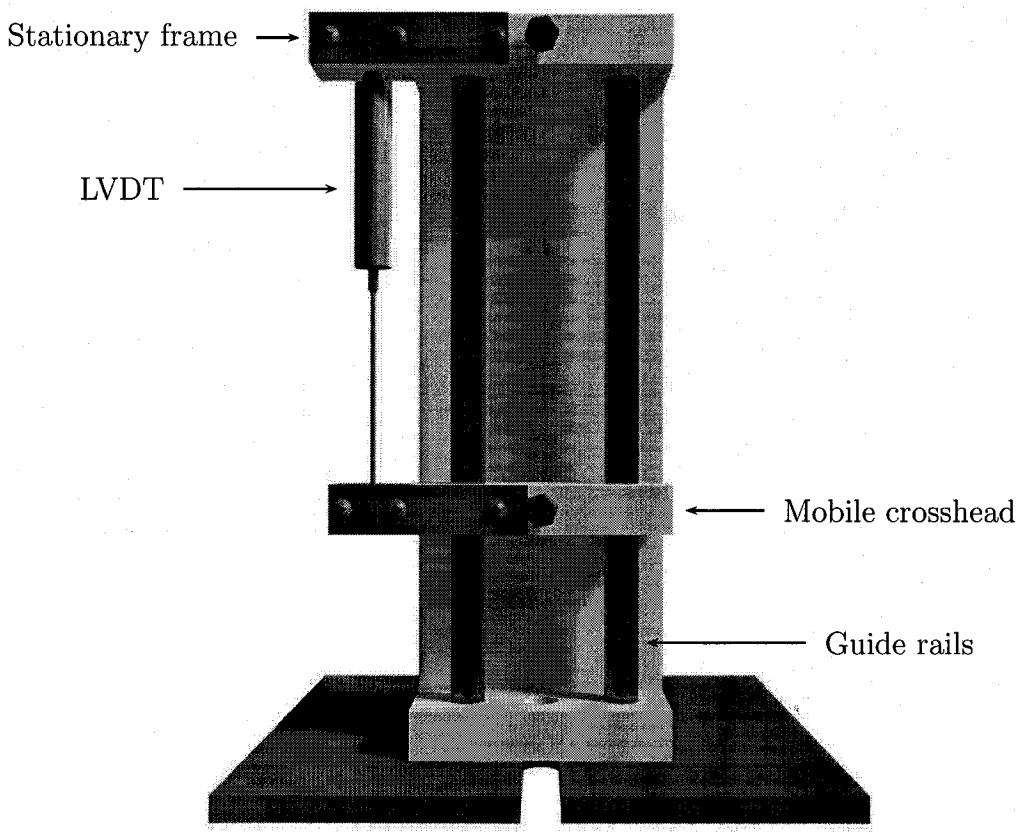


Figure 5.2: Linear test frame

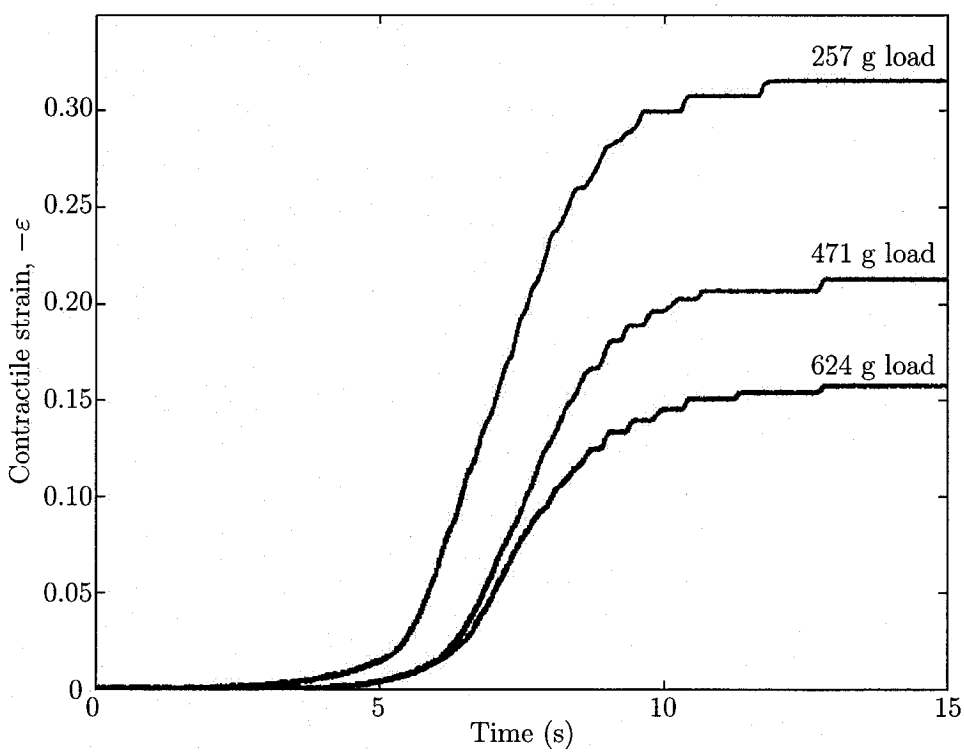


Figure 5.3: Strain response at various loads for a four-loop actuator

Table 5.1: Maximum strain achieved by four-loop actuator

Load (g)	Maximum Contractile Strain (%)
257	31.6
471	21.3
624	15.9

SMA based actuators.

In addition to the tests performed in the linear test frame, a biologically inspired prosthetic arm was designed and manufactured to assess the potential of using the SMA based actuator as an artificial muscle (Figure 5.4). The links of the arm are constructed from electrically non-conductive acrylic tubing. Three independent revolute joints permit elbow extension, wrist pronation, and wrist extension. Potentiometers measure the angular position of each joint, and provide a feedback signal to the position controller. Artificial muscles are fixed at their ends to the arm by means of ring clamps such that their locations are adjustable. The muscles are activated by pulse width modulated power amplifiers controlled via a data acquisition computer. Since the experiments discussed herein are primarily intended to demonstrate the feasibility of SMA based artificial muscles, the wrist pronation and wrist extension joints have been fixed to reduce the system complexity and a pair of artificial muscles equivalent to the human brachioradialis and biceps were installed on the prosthesis to actuate the elbow joint Figure 5.4(a). An angular displacement of 27° was attained, therefore such an actuator has great potential in fields such as robotics.

The large cross-section of the braided actuator results in a small electric resistance, therefore a current of approximately 18 A is required from a 5 V power supply to increase the temperature of the muscle beyond the transition temperature. Therefore, due to this large power requirement, these actuators are not suitable for portable applications such as artificial limbs.

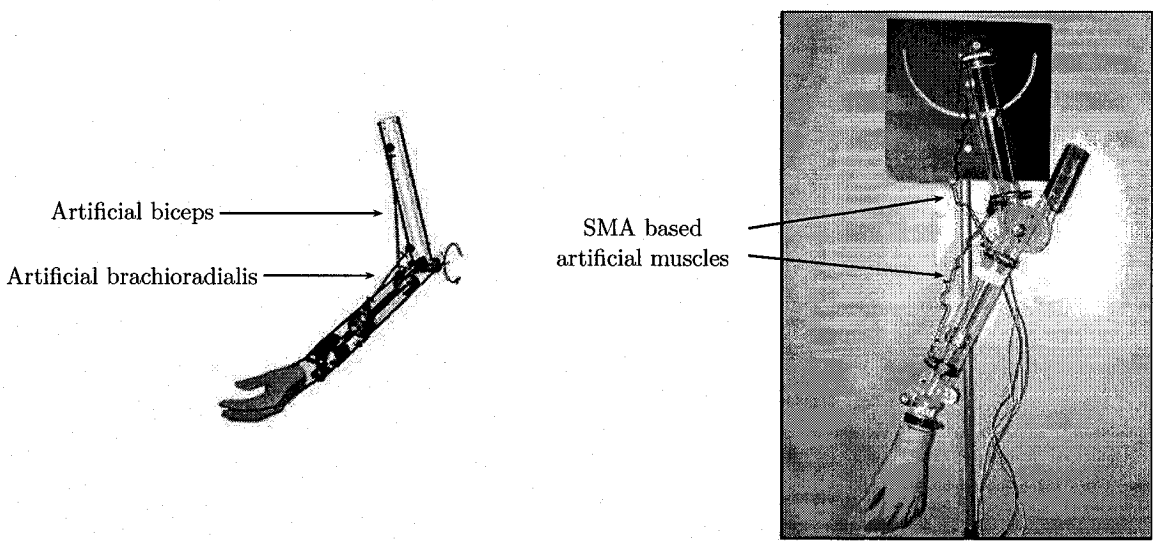


Figure 5.4: Biomimetic prosthetic arm (a) schematic and (b) equivalent experimental embodiment

5.3 Summary

The phenomenon of SME was utilized to develop a lightweight SMA based actuator capable of achieving strains of 31.6%, over 300% greater than the previously reported maximum. The actuator also achieves 15.9% strain under a considerable load of 624 g. This type of novel actuator would result in immediate social benefit in the biomedical field when employed as an artificial muscle in robotic devices.

Chapter 6

Position control of an SMA based artificial hand

This chapter conveys experimental results that demonstrate the performance characteristics of the hand, and insight pertaining to SMA behaviour is utilized to propose a modified control-algorithm to improve the response of the hand.

6.1 Sigmoid feedback control law

A simple two step trajectory was defined to illustrate the typical performance of the SMA actuated finger. The first step corresponds to a desired finger tip contact point location in the world frame that corresponds to required joint variables of $\hat{\boldsymbol{\theta}} = (-5°, 6.5°, -3°)$, which induced muscle contraction at each joint. Note that sensor drift was evident, such as in the case of the DIP joint above, the angle of $-3°$ is outside of the actual range of motion of the joint, however it was within the range of motion as sensed by the bend sensor using its calibration curve. This issue will be readdressed in section 8.1. The second trajectory step was programmed to illustrate the cooling response of the SMA

at each joint using $\hat{\boldsymbol{\theta}} = (-10°, 6°, -3.5°)$. The results of the experiment are shown in figures 6.1, 6.2, and 6.3.

Figure 6.1 illustrates the desired and actual contact point trajectories expressed in the base frame. These trajectories are particularly important as it is the position of the contact point in the world frame that is vital for manipulation. In relation to this, the total error shown is sufficiently small at $t = 7.7$ s to trigger the issue of the second trajectory step. This transition is apparent throughout the results, and is responsible for the abrupt change in desired position in both the Cartesian and joint space, as well as the corresponding error signals. For this experiment, the error tolerance was set to $\kappa = 1.5$ mm and the maximum duty cycle for the MCP joint was set to only 4%, and the PIP and DIP joints were set to approximately 12%.

Figure 6.2 indicates the trajectories of the contact point in the finger joint space and the corresponding joint error signals. These results depict several response characteristics attributed to the sigmoid control law implemented. As shown in the time response of θ_1, the initial active heating response occurs fast: a 12° change in the joint angle is accomplished in approximately 2 s. As Figure 6.3 illustrates, this initial high speed activation period is followed by a period of slower response which is attributed to the smaller control signal issued to the SMA as the joint error diminishes.

The limiting factor governing the performance of the SMA actuators is the passive cooling phase. As shown in the post-transition response of the MCP joint in Figure 6.2, the cooling of the SMA requires much more time: the plot shows that the desired position is not obtained even after 4 s, twice as much time as the heating phase for less than half the joint deflection.

This behaviour confirms one of the initial reasons for using a sigmoidal activation function as overshoot was successfully avoided. Although the small magnitude of the

control signal for small joint errors eliminates the response overshoot, it also results in steady state errors. These residual errors are particularly noticeable in the response of the PIP and DIP joints, and as shown in Figure 6.3 the former has a normalized control signal of less than 8%, which is insufficient for heating the SMA to the required temperature.

Finally, experimental observation indicates that the control loop is typically executed for all joints at a minimum frequency of 10 Hz, which is reasonably fast when compared to the response time of the SMA.

6.2 Modified sigmoid feedback control law

In view of the results provided in the previous section, the original control algorithm proposed in section 4.7 has been modified to compensate for the residual error attributed to the small magnitude of the control signal when the joint error is small. As shown in Figure 6.4, the control law has been modified to include an addition normalized offset component, $\boldsymbol{\rho}$, which is only active for joints having positive joint errors and has the effect of rescaling the lower bound of the control signal from zero to the minimum PWM duty cycle required to activate the SMA. The elements of $\boldsymbol{\rho}$ are individually set for each SMA wire to accommodate the electrical resistance inherent to each artificial muscle.

In this case, the sigmoid shift parameter $\boldsymbol{\mu}$ is redefined according to the range of PWM duty (previously the maximum PWM duty was used since the lower bound of the range was identically zero) according to $\boldsymbol{\mu} = \ln(\boldsymbol{PWM}_{\text{range}}-1)$. The PWM gain vector is now also redefined as the PWM duty range, such that each element, i, of the normalized offset component vector can be determined as:

$$\boldsymbol{\rho}_i = \frac{\boldsymbol{PWM}_{\text{min},i}}{\boldsymbol{PWM}_{\text{range},i}}. \tag{6.1}$$

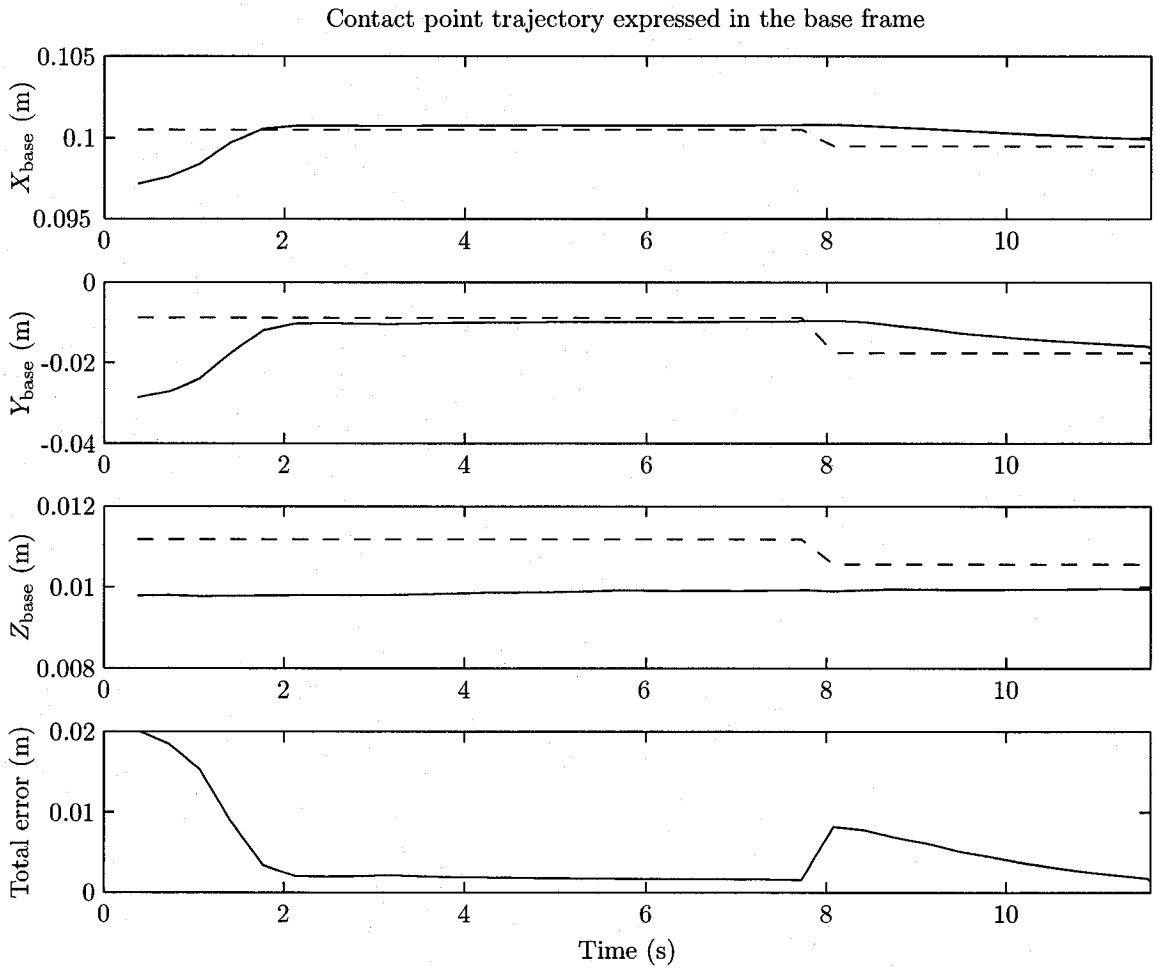


Figure 6.1: Contact point trajectory expressed in the base frame. Desired trajectory is indicated by a dashed line and actual trajectory is indicated by a solid line.

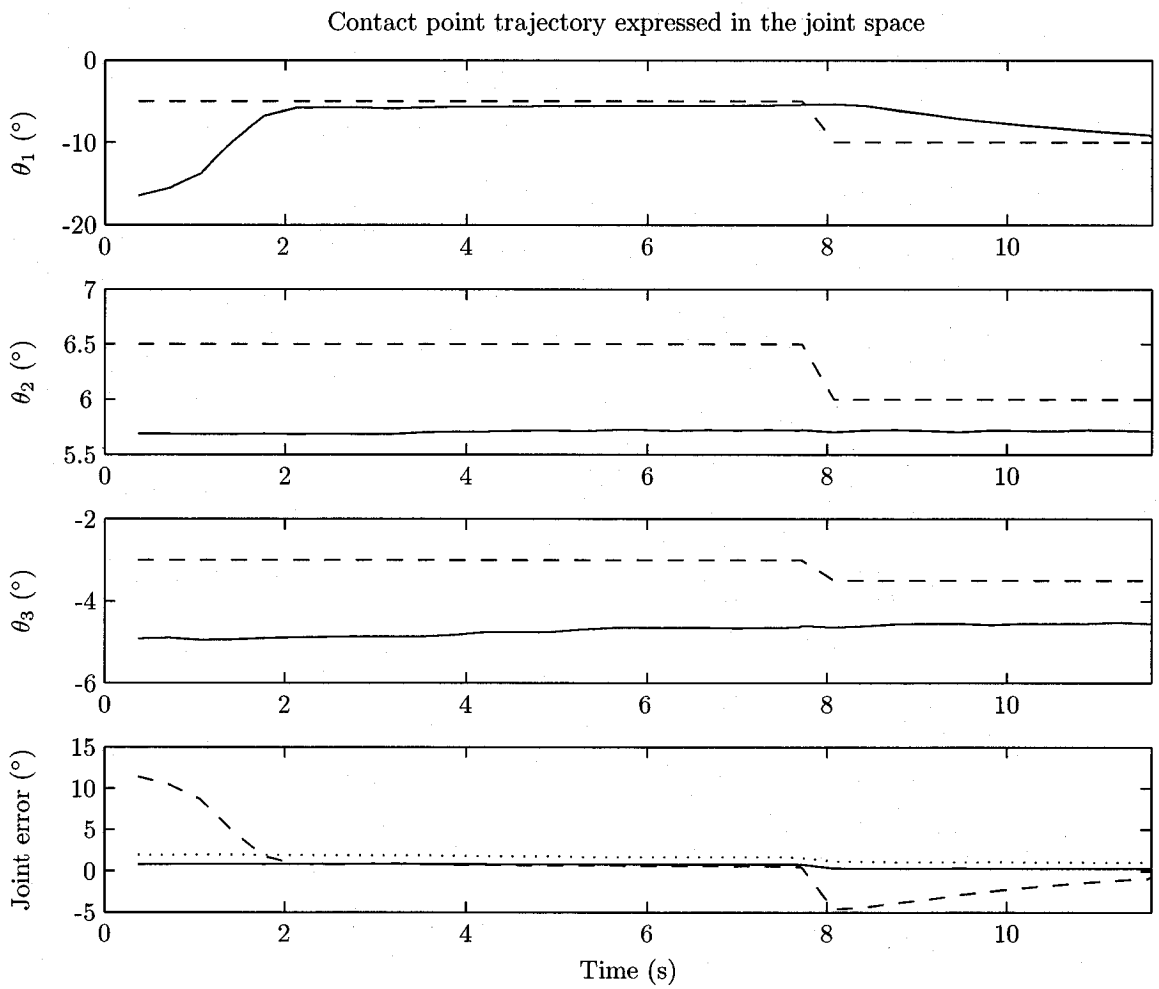


Figure 6.2: Contact point trajectory expressed in the joint space. Desired trajectory is indicated by a dashed line and actual trajectory is indicated by a solid line, with exception of the last plot where the MCP joint error is depicted by a dashed line, the PIP joint error is depicted by a dotted line, and the DIP joint error is depicted by a solid line.

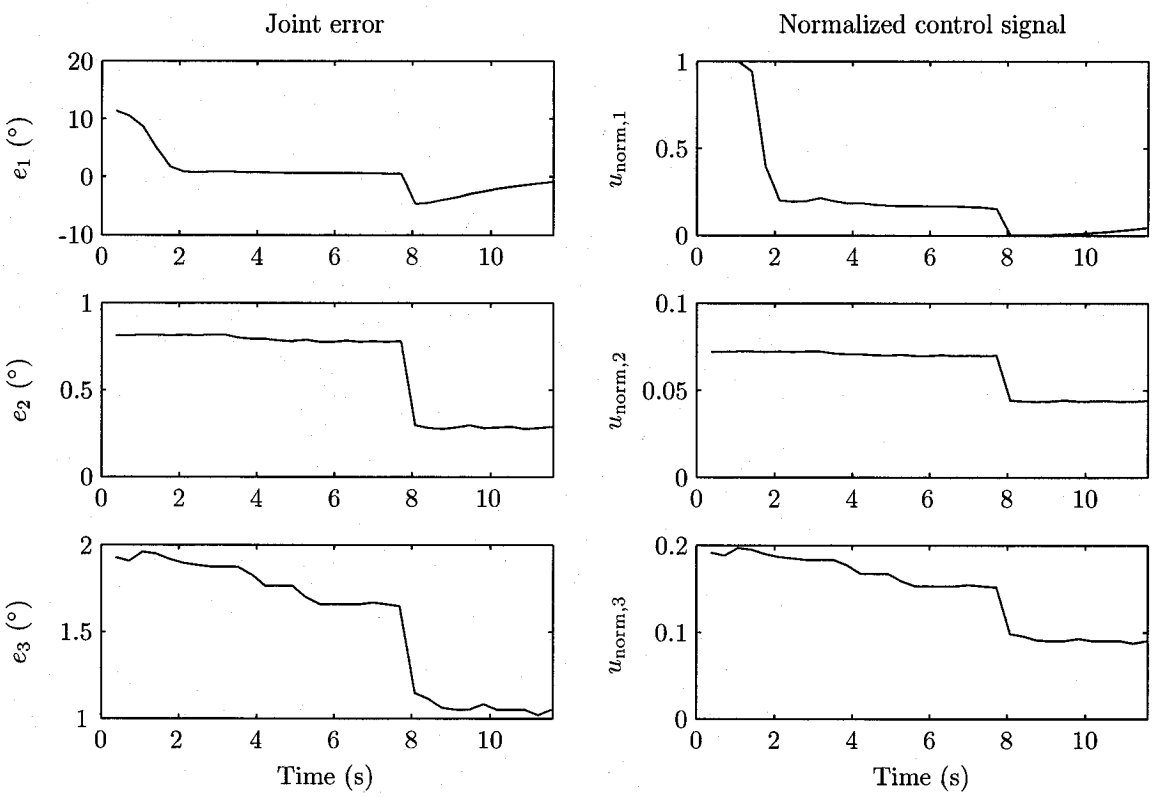


Figure 6.3: Joint error and normalized control signal

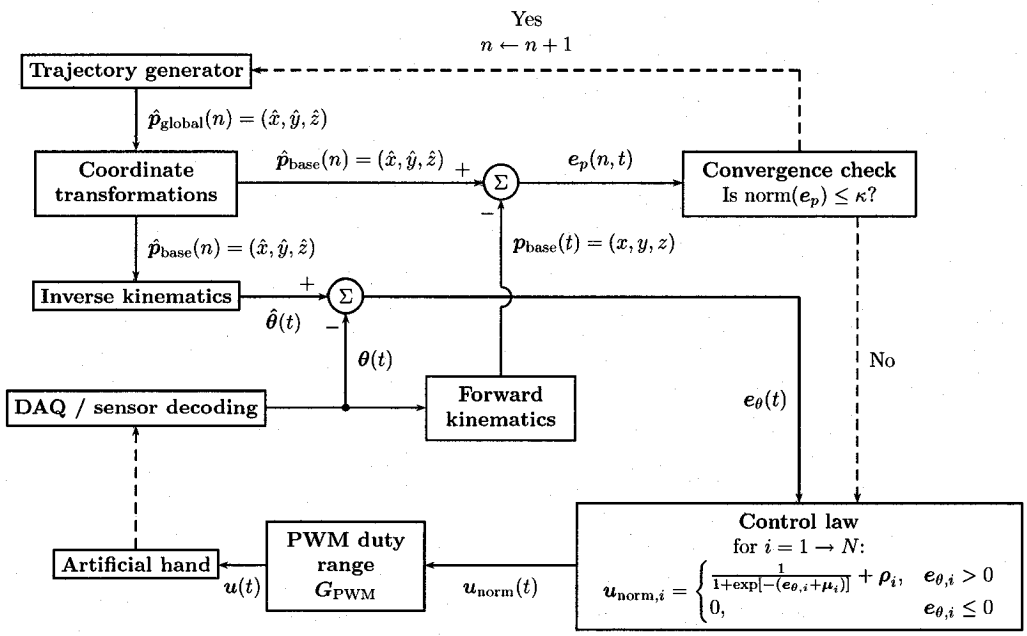


Figure 6.4: Control algorithm modified to include a PWM duty offset

In this manner, saturation of the output signal is guaranteed, thereby protecting the SMA from overheating due to sustained full power activation arising in the event of sensor failure.

Tuning of each joint's PWM duty range and offset must be empirically performed in order to establish a balance between reasonable time response and avoidance of overshoot. Representative experimental joint responses using the modified controller are shown in Figure 6.5 for an MCP and PIP joint (the DIP joint response exhibits the same characteristics of the latter). The inclusion of the constant PWM offset component in the control signal eliminates the steady state error previously encountered for small joint errors. The offset component also slightly increases the effective PWM duty cycle (when the duty range is strictly set to the duty maximum of the initially proposed control algorithm). This increase in the duty cycle logically improves the general time response of the joints as the SMA elements are more rapidly heated. Moreover, improved

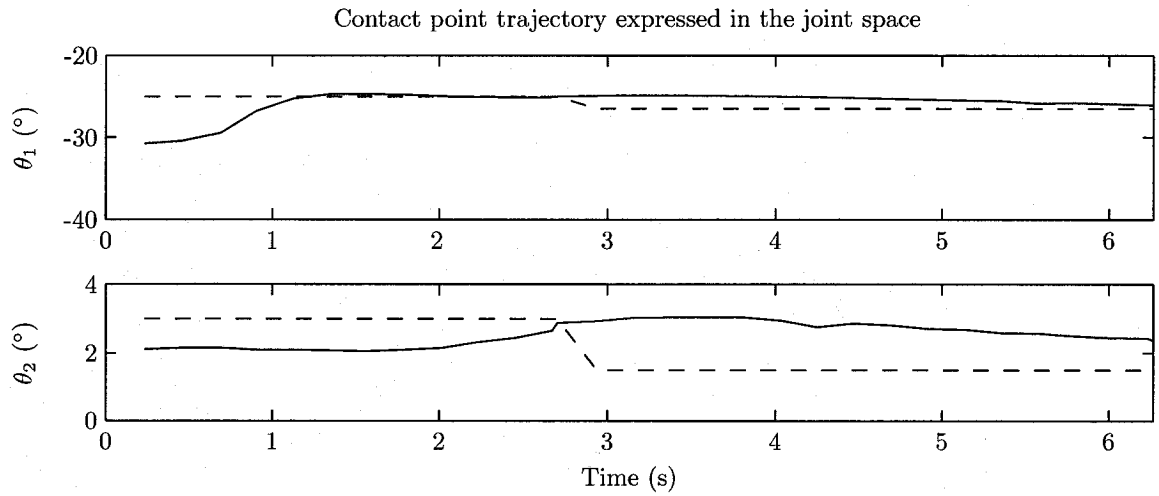


Figure 6.5: The MCP and PIP response using modified controller indicate a reduced steady state error and improved time response. Desired trajectories are indicated by a dashed line and actual trajectories are indicated by a solid line.

response characteristics may be obtained from the SMA by imposing some initial cyclic thermomechanical loading to stabilize the hysteretic response.

6.3 Summary

This chapter explored the experimental evaluation of the SMA actuated robot hand. The proposed sigmoid control law was implemented and the resulting behaviour was examined. In order to reduce the steady state error, a modified version of the control law is proposed that includes a nonzero minimum activation signal for positive joint error.

The experimental results indicate that although SMA wires are promising actuators for prosthetic devices, they suffer from the slow time response inherent to the passive cooling necessary for SMA elongation. This drawback may be improved through the implementation of an active cooling system, however such a system will add mass to the

device and further increase its power consumption.

Chapter 7

Conclusions

This thesis investigates the issues surrounding the practical application of shape memory alloys as artificial muscles in a three fingered robot hand intended for prosthetic applications. The experience gained through this study has led to the following conclusions:

1. The function of the human hand offered valuable insight on the design of an artificial hand from a biomimetic standpoint.

2. An investigation of the kinematic requirements for manipulation determined both the necessary finger DOF and the number of fingers required for manipulation assuming twist rolling type contact.

3. The forward and inverse kinematics of the artificial finger have been established.

4. Shape memory alloys have been shown to provide a feasible alternative actuation technology for light-weight robot hand applications by means of the shape memory effect.

5. A high strain SMA actuator has been introduced having a strain capacity of 31.6%, an improvement of over 300% over the previously reported maximum. The actuator

also achieves 15.9% strain under a considerable load of 624 g.

6. A new prosthetic hand has been designed and manufactured, and has a mass of less than 400 g, which represents an appreciable mass savings of 26% when compared to a typical commercially available prosthetic hand.

7. The requisite instrumentation and computer interfacing systems have been implemented to facilitate the operation and experimental performance evaluation of the hand.

8. A new sigmoid based control algorithm has been proposed and evaluated for the position control of SMA elements that minimizes overshoot to avoid the slow time response inherent to the passive cooling necessary for SMA elongation. This approach was taken after a hysteresis type constitutive model was shown to be insufficient for real-time control purposes.

9. A modified sigmoid control algorithm having a nonzero offset is proposed to reduce the steady state error observed in the robot joint space response.

10. Finally, the dexterous hand apparatus and associated instrumentation remains as a facility for the future characterization of artificial muscles and exploration of other research areas of dexterous robotics.

Chapter 8

Recommendations and future work

8.1 Recommendations

The following recommendations are suggested for improving the research project:

1. Shallow guide channels machined on the surface of the finger links will restrict lateral motion of the SMA wires.

2. The current method for fixing the ends of the SMA wire entails clamping the wire end between the link surface and a washer using a nut, however this method is insufficient for long-term use as repeated extension and contraction of the SMA results in the eventual failure of the clamping.

3. The bend sensors are quite sensitive such that they require frequent calibration. Their long-term stability could be improved by mounting the sensor on a stiff (but still flexible) substrate such that long-term sensor repeatability is improved, and the original fully extended shape of the sensor is preserved.

4. The current apparatus relies solely on the bend sensors for position information such

that if a sensor is functioning incorrectly the controller can not discern a problem. In order to avoid this problem, an external coordinate measuring machine could be implemented to objectively evaluate the operation of the robot.

8.2 Future work

The artificial hand apparatus and associated computer interfacing hardware provides the foundation for a vast field of research. Since this project has established the fundamental framework of the robot, future work from a robotics standpoint may focus on extending the high-level functionality of the robot. An example of this functionality may include the autonomous manipulation of objects. Such a scenario requires *a priori* knowledge of the environment and the object being manipulated, which is typically permissible for industrial robot applications where the environment is generally known or machine vision based registration techniques may be employed. Grasp planning and tactile sensing issues are other potential avenues of research. From a materials science standpoint, the apparatus serves as a valuable facility for the practical evaluation of emerging types of artificial muscles such as shape memory polymers that do not suffer from the same performance drawbacks as SMA.

List of references

[1] Allison M. Okamura, Niels Smaby, and Mark R. Cutkosky. Overview of dexterous manipulation. *Proceedings - IEEE International Conference on Robotics and Automation*, 1:255–262, 2000.

[2] Yoseph Bar-Cohen and Cynthia L. Breazeal. *Biologically-Inspired Intelligent Robots.* SPIE - International Society for Optical Engineering, 2003.

[3] D. C. Lagoudas, P. B. Entchev, P. Popov, E. Patoor, L. C. Brinson, and X. Gao. Shape memory alloys, Part II: Modeling of polycrystals. *Mechanics of Materials*, 38(5-6):430–462, 2006.

[4] Zexiang Li and Shankar Sastry. *Issues in Dextrous Robot Hands*, pages 154–186. Dextrous Robot Hands. Springer-Verlag, New York, 1989.

[5] J. L. Pons, R. Ceres, and F. Pfeiffer. Multifingered dextrous robotics hand design and control: A review. *Robotica*, 17(6):661–674, 1999.

[6] V. O. Del Cura, F. L. Cunha, M. L. Aguiar, and A. Cliquet Jr. Study of the different types of actuators and mechanisms for upper limb prostheses. *Artificial Organs*, 27(6):507–516, 2003.

[7] C. Pfeiffer, K. DeLaurentis, and C. Mavroidis. Shape memory alloy actuated robot prostheses: initial experiments. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2385–2391, 1999.

[8] L. E. Pezzin, T. R. Dillingham, E. J. MacKenzie, P. Ephraim, and P. Rossbach. Use and satisfaction with prosthetic limb devices and related services. *Archives of Physical Medicine and Rehabilitation*, 85(5):723–729, 2004.

[9] V. Bundhoo and E. J. Park. Design of an artificial muscle actuated finger towards biomimetic prosthetic hands. In *12th International Conference on Advanced Robotics*, pages 368–375, 2005.

[10] J. Dormandy, L. Heeck, and S. Vig. Major amputations: Clinical patterns and predictors. *Seminars in Vascular Surgery*, 12(2):154–161, 1999.

[11] Y. Bar-Cohen, T. Xue, M. Shahinpoor, J. Simpson, and J. Smith. Flexible, low-mass robotic arm actuated by electroactive polymers and operated equivalently to human arm and hand. In *Proceedings of the ASCE Specialty Conference on Robotics for Challenging Environments*, pages 15–21, 1998.

[12] C. P. Chou. Measurement and modeling of McKibben pneumatic artificial muscles. *IEEE transactions on robotics and automation*, 12(1):90–103, 1996.

[13] F. El Feninat, G. Laroche, M. Fiset, and D. Mantovani. Shape memory materials for biomedical applications. *Advanced Engineering Materials*, 4(3):91–104, 2002.

[14] J. B. Makaran, D. K. Dittmer, R. O. Buchal, and D. B. MacArthur. The smart wrist-hand orthosis (WHO) for quadriplegic patients. *JPO*, 5(3):73–76, 1993.

[15] K. J. De Laurentis and C. Mavroidis. Mechanical design of a shape memory alloy actuated prosthetic hand. *Technology and Health Care*, 10(2):91–106, 2002.

[16] R. B. Gorbet and R. A. Russell. Novel differential shape memory alloy actuator for position control. *Robotica*, 13(4):423–430, 1995.

[17] R. Barrett and R. S. Gross. Super-active shape-memory alloy composites. *Smart Materials and Structures*, 5(3):255–260, 1996.

[18] U. Icardi. Large bending actuator made with SMA contractile wires: Theory, numerical simulation and experiments. *Composites Part B: Engineering*, 32(3):259–267, 2001.

[19] Otto Bock Health Care. DMC Plus Hand. `http://www.ottobockus.com`, July 2006.

[20] Keree M. B. Bennett and Umberto Castiello. *Insights into the reach to grasp movement.* North-Holland, New York, 1994.

[21] J. W. Morley. *Neural aspects in tactile sensation.* Elsevier, New York, 1998.

[22] M. Gentilucci, L. Caselli, and C. Secchi. Finger control in the tripod grasp. *Experimental Brain Research*, 149(3):351–360, 2003.

[23] Thea Iberall and Christine L. MacKenzie. *Opposition space and human prehension*, pages 32–54. Dextrous Robot Hands. Springer-Verlag, New York, 1989.

[24] Henry Gray. Anatomy of the human body. `http://www.bartleby.com/107/`, July 2006.

[25] Christine L. MacKenzie. *The grasping hand.* North-Holland, New York, 1994.

[26] I. MacKay Murray. *Human Anatomy Made Simple: A Comprehensive Course for Self-Study and Review.* Doubleday Publishing, May 1969.

[27] J. Biggs and K. Horch. A three-dimensional kinematic model of the human long finger and the muscles that actuate it. *Medical Engineering and Physics*, 21(9):625–639, 1999.

[28] Paul W. Brand. *Clinical mechanics of the hand.* Mosby, St. Louis, Mo., 1999.

[29] Sarah Goodfellow and Sheila Jennett. *Hands.* The Oxford Companion to the Body. Oxford University Press, New York, 2001.

[30] L. Jones. Dextrous hands: Human, prosthetic, and robotic. *Presence: Teleoperators and Virtual Environments*, 6(1):29–56, 1997.

[31] T. Nagashima, H. Seki, and M. Takano. Analysis and simulation of grasping/manipulation by multi-fingersurface. *Mechanism and Machine Theory*, 32(2):175–191, 1997.

[32] Kazuo Tanie. *Mechanism of Artificial Hands*, pages 7–13. Mechanical Hands Illustrated. Taylor & Francis Group, 1986.

[33] Dejan B. Popović. *Externally Powered and Controlled Orthotics and Prosthetics*, pages 142–9–142–11. The Biomedical Engineering Handbook. Springer, 2000.

[34] Matthew T. Mason and J. Kenneth Salisbury. *Robot Hands and the Mechanics of Manipulation.* MIT Press, January 1985.

[35] J. Kenneth Salisbury and John J. Craig. Articulated hands - force control and kinematic issues. *International Journal of Robotics Research*, 1(1):4–17, 1982.

[36] Jeffrey Kerr and Bernard Roth. Analysis of multifingered hands. *International Journal of Robotics Research*, 4(4):3–17, 1986.

[37] Ian D. McCammon and Steve C. Jacobson. *Tactile sensing and Control for the Utah/MIT Hand*, pages 239–266. Dextrous Robot Hands. Springer-Verlag, New York, 1989.

[38] L. Biagiotti, F. Lotti, C. Melchiorri, and G. Vassura. How far is the human hand? a review on anthropomorphic robotic end-effectors, November 2003.

[39] A. V. Srinivasan and D. Michael McFarland. *Smart Structures: Analysis and Design.* Cambridge University Press, November 2000.

[40] Brian Culshaw. *Smart Structures and Materials.* Artech House, Incorporated, December 1995.

[41] R. C. Smith. *Smart Material Systems: Model Development.* Society for Industrial & Applied Mathematics, Philadelphia, 2005.

[42] K. Otsuka and X. Ren. Recent developments in the research of shape memory alloys. *Intermetallics*, 7(5):511–528, 1999.

[43] K. Otsuka and X. Ren. Physical metallurgy of Ti-Ni-based shape memory alloys. *Progress in Materials Science*, 50(5):511–678, 2005.

[44] J. Van Humbeeck, K. U. Leuven, and R. Stalmans. *Shape Memory Alloys, Types and Functionalities*, volume 2 of *Encyclopedia of Smart Materials*, pages 951–964. J. Wiley, New York, 2002.

[45] J. Van Humbeeck. Shape memory alloys: A material and a technology. *Advanced engineering materials*, 3(11):837–850, 2001.

[46] Yoseph Bar-Cohen. *Electroactive Polymer (EAP) Actuators as Artificial Muscles: Reality, Potential, and Challenges.* SPIE - International Society for Optical Engineering, 2001.

[47] Yoshio Yamamoto. Development of robot arm, an application of shape memory alloys to robotics. *Furukawa Denko Jiho/ Furukawa Electric Review*, (75):109–113, 1985.

[48] H. Prahlad and I. Chopra. Comparative evaluation of shape memory alloy constitutive models with experimental data. *Journal of Intelligent Material Systems and Structures*, 12(6):383–395, 2001.

[49] D. Bernardini and T. J. Pence. *Memory Materials, Modeling*, volume 2 of *Encyclopedia of Smart Materials*, pages 964–980. J. Wiley, New York, 2002.

[50] X. Peng, Y. Yang, and S. Huang. Two-phase mixture model for SMAs and application to the analysis for pseudoelasticity of a SMA polycrystal. *Smart Materials and Structures*, 9(5):604–612, 2000.

[51] Z. K. Lu and G. J. Weng. Micromechanical theory for the thermally induced phase transformation in shape memory alloys. *Smart Materials and Structures*, 9(5):582–591, 2000.

[52] Y. Matsuzaki and H. Naito. Macroscopic and microscopic constitutive models of shape memory alloys based on phase interaction energy function: A review. *Journal of Intelligent Material Systems and Structures*, 15(2):141–155, 2004.

[53] P. Sittner, R. Stalmans, and M. Tokuda. Algorithm for prediction of the hysteretic responses of shape memory alloys. *Smart Materials and Structures*, 9(4):452–465, 2000.

[54] A. Price, A. Edgerton, C. Cocaud, H. Naguib, and A. Jnifene. A study on the thermomechanical properties of shape memory alloys based actuators used in artificial muscles. *Journal of Intelligent Material Systems and Structures*, pages JIMSS–05–012, in press.

[55] N. Ma and G. Song. Control of shape memory alloy actuator using pulse width modulation. *Smart Materials and Structures*, 12(5):712–719, 2003.

[56] N. Ma. Control of shape memory alloy actuator using pulse width (pw) modulation. *Proceedings of SPIE*, 4693:348, 2002.

[57] G. Song, V. Chaudhry, and C. Batur. A neural network inverse model for a shape memory alloy wire actuator. *Journal of Intelligent Material Systems and Structures*, 14(6):371–377, 2003.

[58] J. M. Stevens and G. D. Buckner. Intelligent control of a micro-manipulator actuated with shape memory alloy tendons. In *Proceedings of SPIE*, volume 5049, pages 56–64, 2003.

[59] A. Kumagai. Neuro-fuzzy model based feedback controller for shape memory alloy actuators. In *Proceedings of SPIE*, volume 3984, pages 291–299, 2000.

[60] J. Denavit and R. S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *ASME J. Appl. Mech.*, 22:215–221, 1955.

[61] Robert J. Schilling. *Fundamentals of Robotics: Analysis and Control.* Prentice Hall, New Jersey, 1990.

[62] A. Price, A. Jnifene, and H. E. Naguib. Biologically inspired anthropomorphic arm and dextrous robot hand actuated by smart-material-based artificial muscles. In Yuji Matsuzaki, editor, *Proc. SPIE Smart Structures and Materials 2006: Smart Structures and Integrated Systems*, volume 6173, page 61730X. SPIE, March 16 2006.

Appendices

Appendix A

Detailed design of robot phalanges

This appendix provides CAD drawings of the structural elements comprising the artificial fingers including the proximal phalanx, middle phalanx, distal phalanx, and base. All dimensions are in inches, and the title block has been intentionally removed to facilitate the drawings' inclusion.

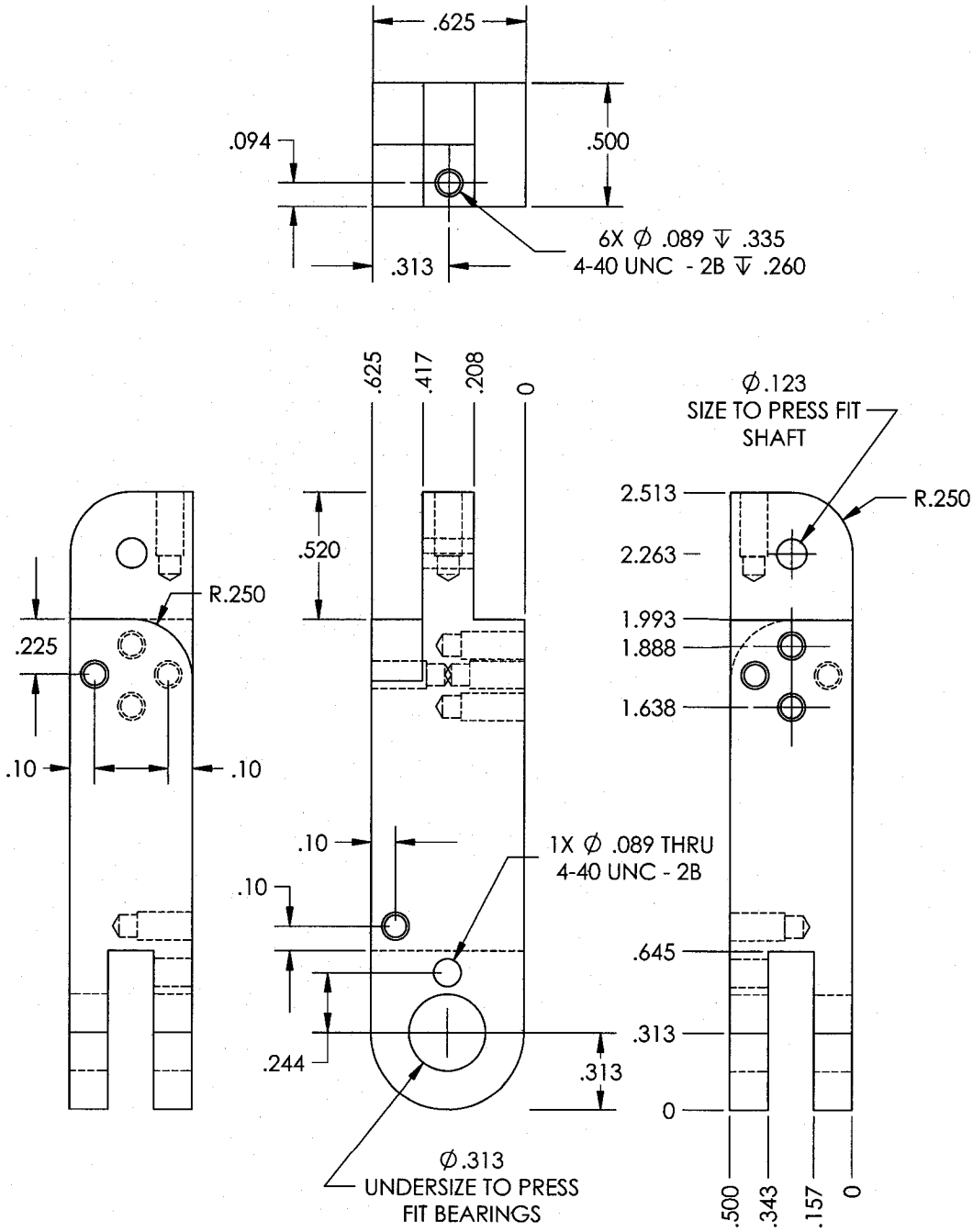


Figure A.1: Proximal phalanx

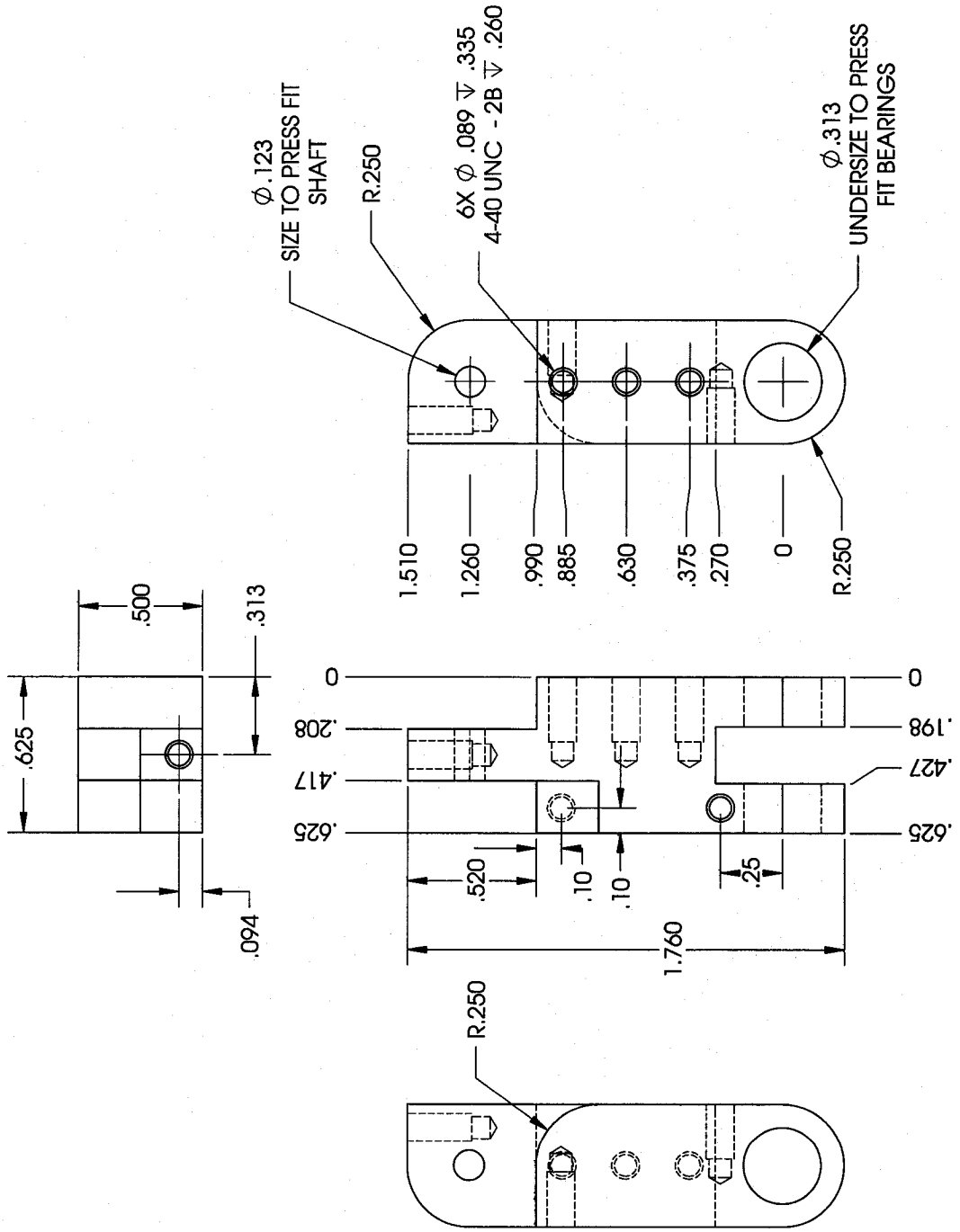


Figure A.2: Middle phalanx

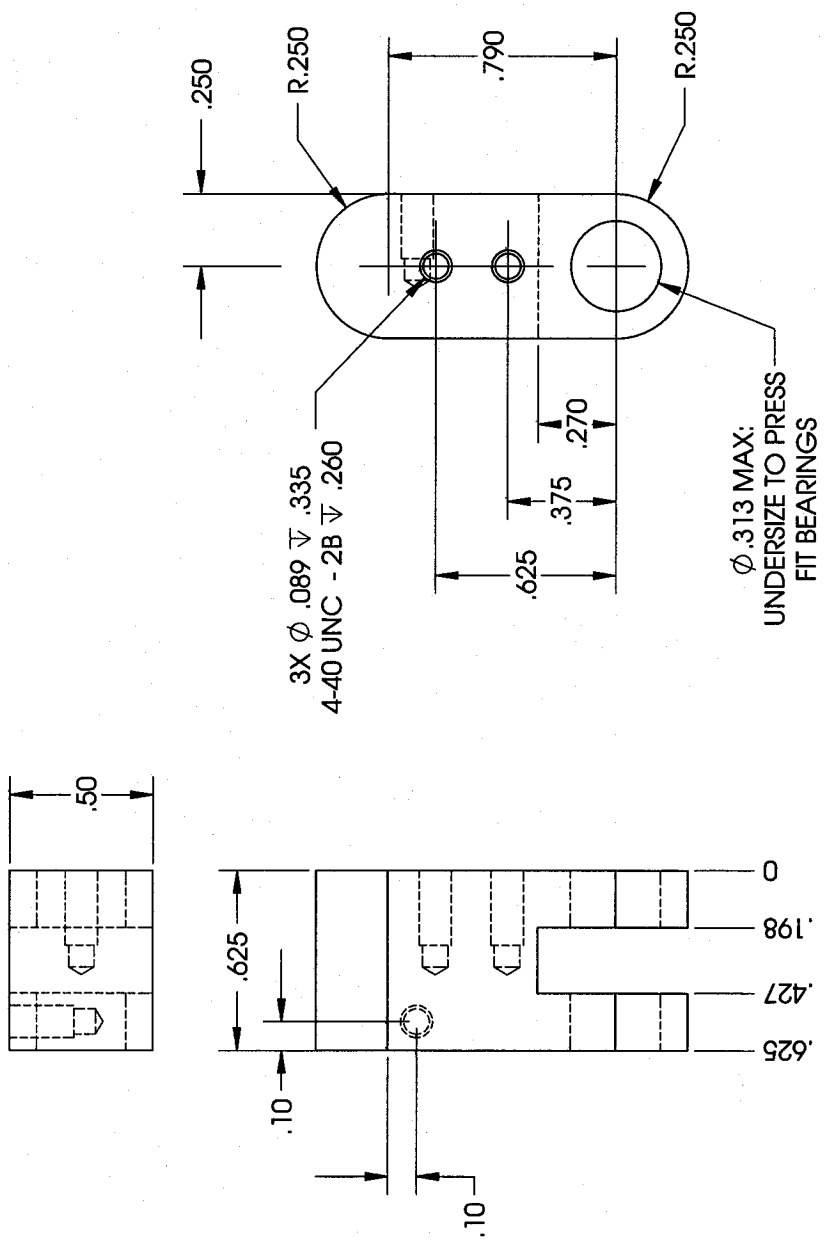


Figure A.3: Distal phalanx

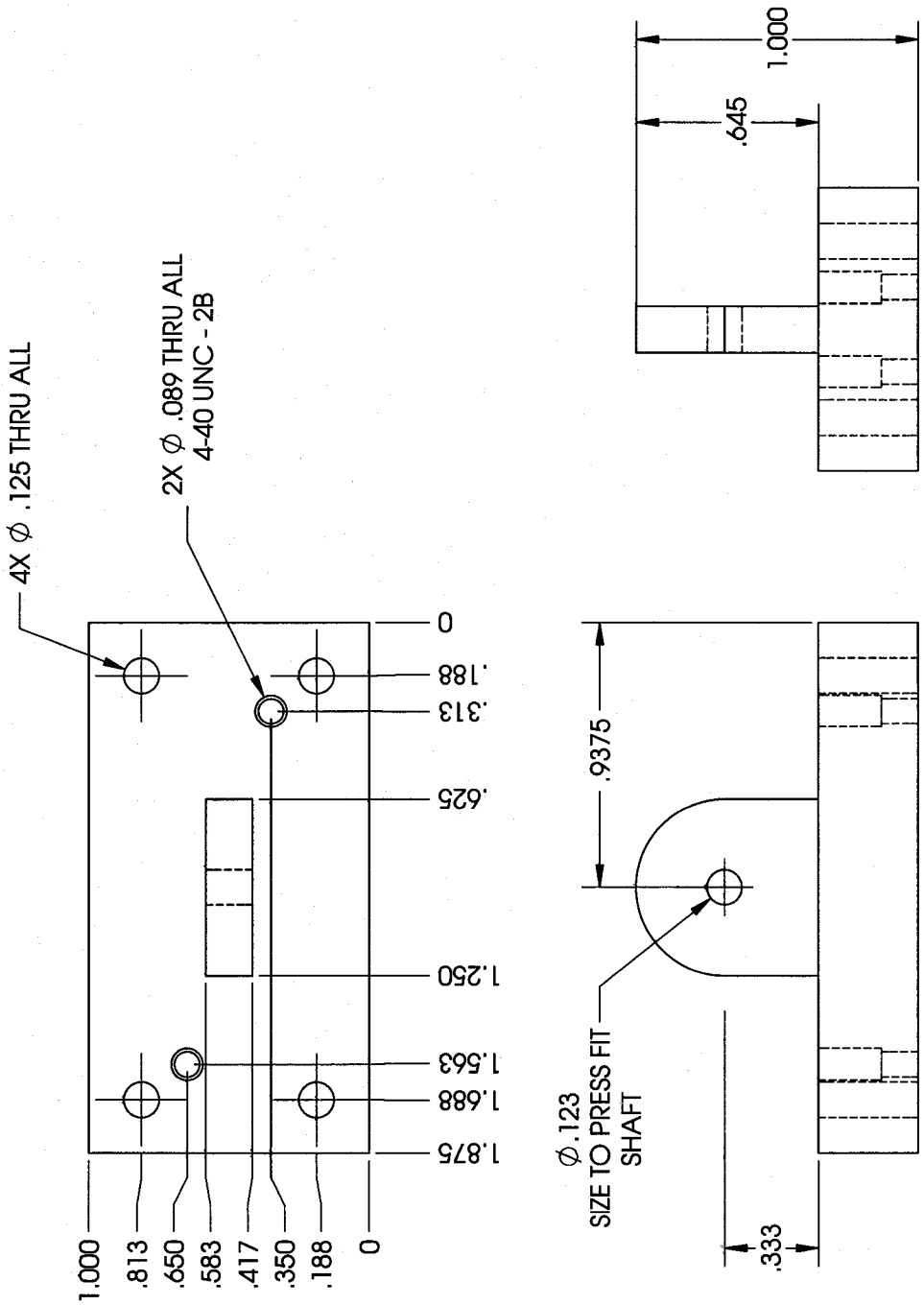


Figure A.4: Metacarpal base

Appendix B

Detailed electrical system design

Electrical schematics are presented in this appendix that provide complete details pertaining to the electrical system design for the artificial hand. The complete system is comprised of a digital subsystem (Figure B.1) that utilizes micro-controllers to facilitate communication and drives second subsystem. The second subsystem contains the analog SMA power circuits (Figure B.2), which are optically isolated from the digital components to minimize the risk of damage to the (more expensive) digital devices and potential injury to the the patient should any electrical problems occur. Finally, the bend sensor signal conditioning circuit used to maximize the resolution and dynamic range of the sensor is outlined in Figure B.3.

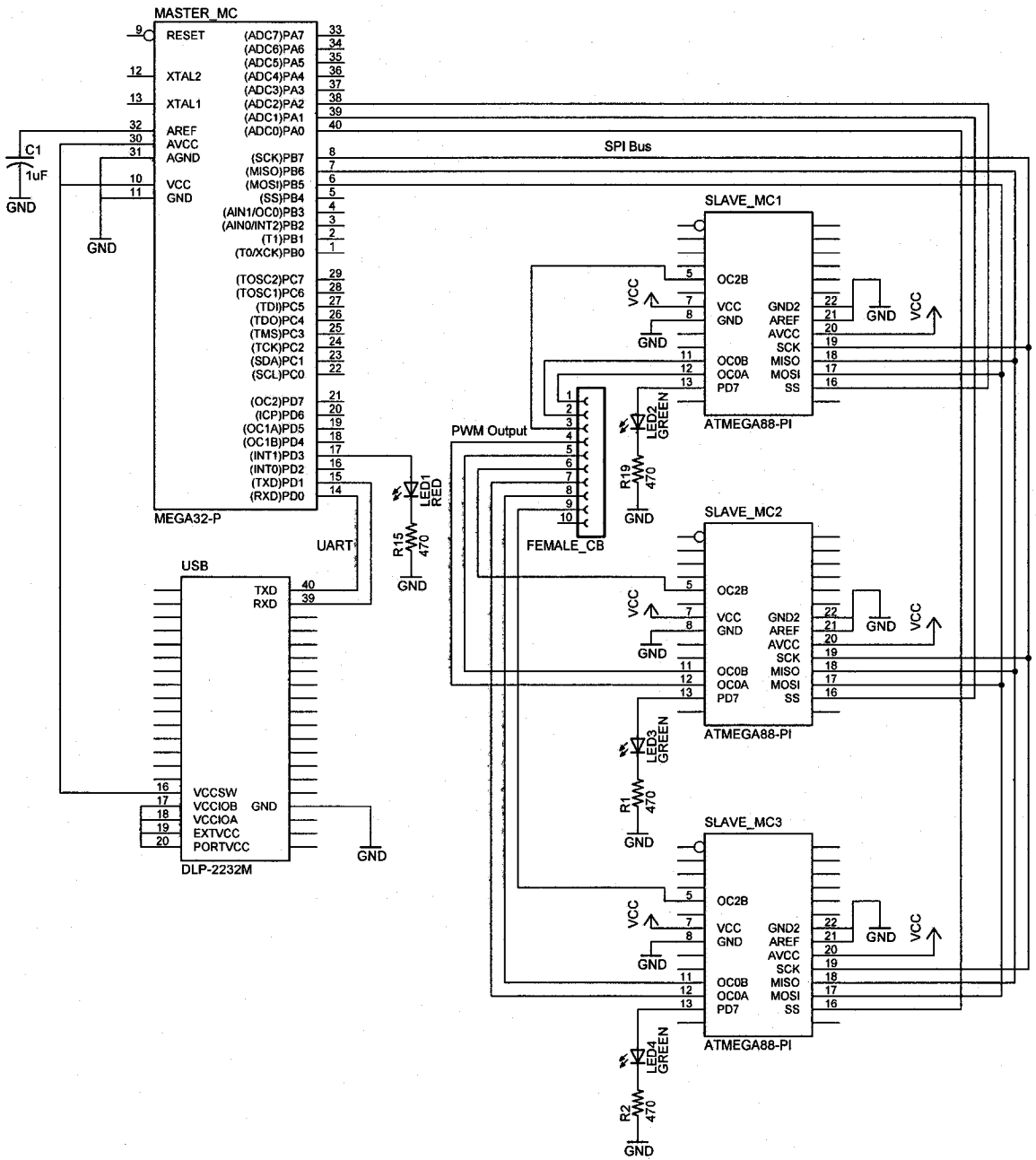


Figure B.1: Digital circuit

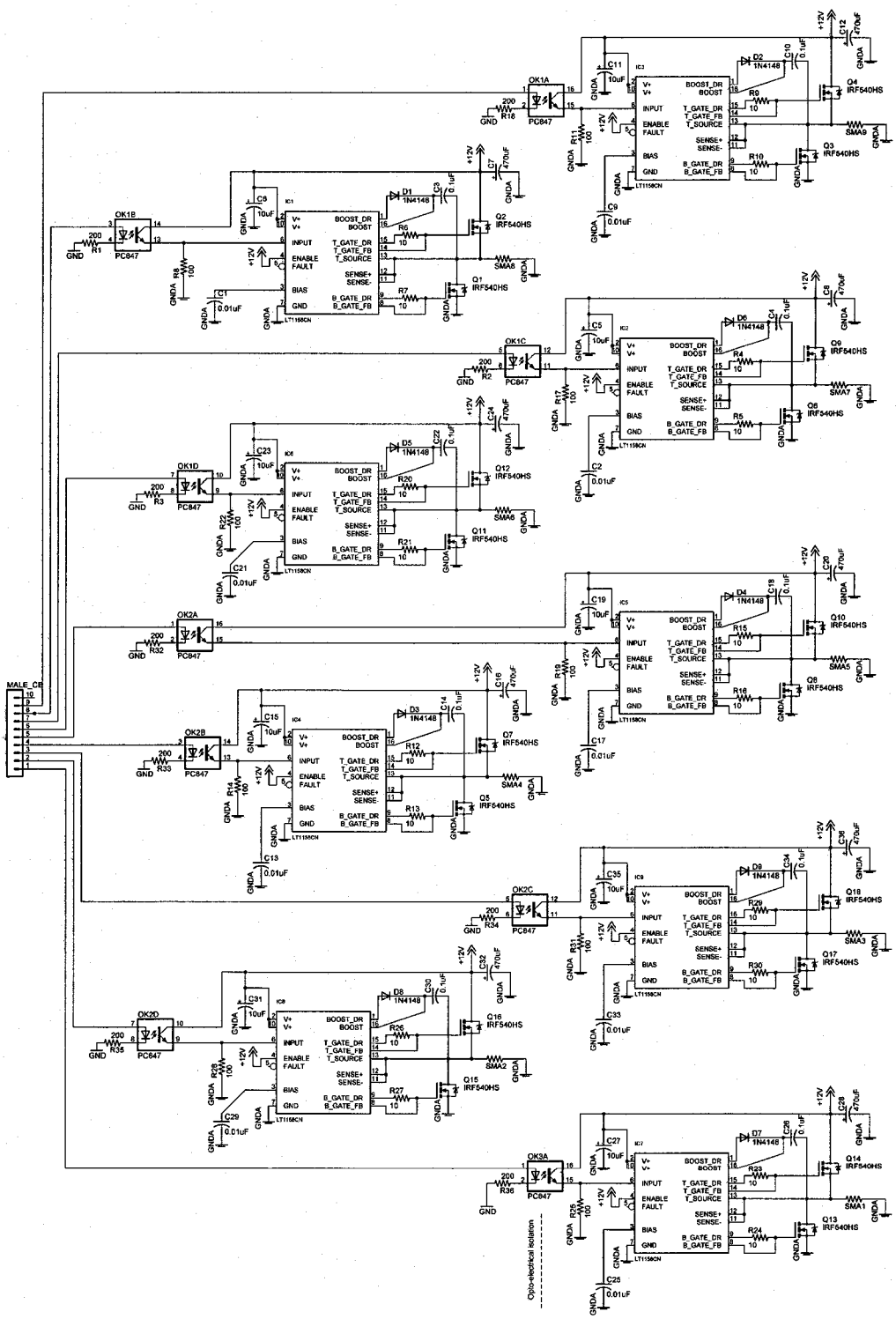


Figure B.2: Analog circuit

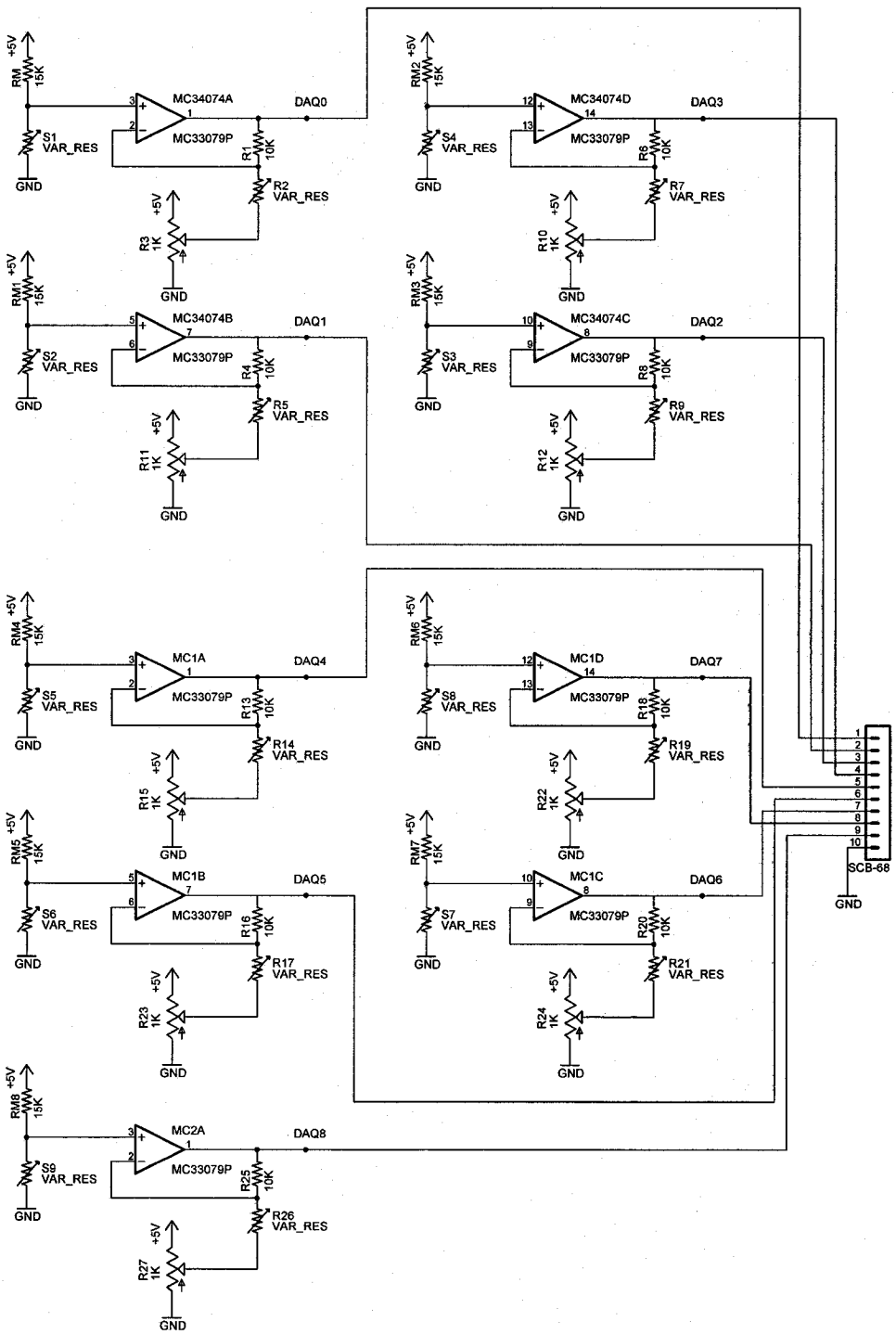


Figure B.3: Sensor signal conditioning circuit

Appendix C

Software code listing

C.1 Micro-controller Programs

C.1.1 Master micro-controller program

Listing C.1: *spi_master.c* Master micro-controller program

```
#include <inttypes.h>
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/signal.h>
#include <avr/delay.h>

#  define OC1 PD5
#  define DDROC DDRD
#  define OCR OCR1A
#  define PWM10 WGM10
#  define PWM11 WGM11

#define BAUD_RATE 4800
#define UART_HIGH_SPEED 0

#define F_CPU 1000000   //1 MHz

//Handle different settings for the "high-speed" UBR
#ifdef UART_HIGH_SPEED
#       define UART_DIV 8L
#else
```

```
#       define UART_DIV 16L
#endif

#define UART_BAUD_SELECT(baudRate,xtalCpu) ((xtalCpu)/((baudRate)*
    UART_DIV)-1)

#define BAUD_VALUE UART_BAUD_SELECT(BAUD_RATE, F_CPU);

void ubusInit()
{
        UBRRL = BAUD_VALUE;

        //Set or clear the double-speed bit as required
        #ifdef UART_HIGH_SPEED
                UCSRA |= _BV(U2X);
        #else
                UCSRA &= ~_BV(U2X);
        #endif
        UCSRB |= _BV(RXCIE) | _BV(RXEN) | _BV(TXEN);

        UCSRC |= _BV(URSEL) | _BV(UCSZ1) | _BV(UCSZ0);
}

void ubusPutChar(uint8_t data)
{
        //Send Byte
        UDR = data;

        asm("nop");

        //Wait for byte to be sent
        loop_until_bit_is_set(UCSRA, TXC);

        asm("nop");

        //reset flag
        UCSRA |= _BV(TXC);
}

void SPI_MasterInit(void)
{
```

```
        // Add a delay to allow the slaves to come online
        _delay_ms(500);

        volatile char IOReg;
        /* enable PA0-PA2 as SS lines */
    DDRA = (1<<PA0)|(1<<PA1)|(1<<PA2);

        PORTA |= _BV(PA0);
        PORTA |= _BV(PA1);
        PORTA |= _BV(PA2);

        /* Set MOSI and SCK output, all others input */
        DDRB = (1<<PB4)|(1<<PB5)|(1<<PB7);
        /* Enable SPI, Master, set clock rate fck/16 */
        SPCR = (1<<SPE)|(1<<MSTR);
        IOReg = SPSR;
        IOReg = SPDR;
}

void SPI_MasterTransmit(char cData)
{
        /* Start transmission */
        SPDR = cData;
        /* Wait for transmission complete */
        while(!(SPSR & (1<<SPIF)));
}

void ioinit (void)
{
        DDRD |= _BV(PD3);
}

unsigned char USART_Receive( void )
{
        /* Wait for data to be received */
        while ( !(UCSRA & (1<<RXC)) );
        /* Get and return received data from buffer */
        return UDR;
}

void relay_pwm_1(uint8_t rc)
```

```
{
                // Forward the byte to the first finger
                PORTA &= ~_BV(PA0);
                // Send the activation
                SPI_MasterTransmit(rc);
                // Turn off the SS line
                PORTA |= _BV(PA0);
}

void relay_pwm_2(uint8_t rc)
{
                // Forward the byte to the first finger
                PORTA &= ~_BV(PA1);
                // Send the activation
                SPI_MasterTransmit(rc);
                // Turn off the SS line
                PORTA |= _BV(PA1);
}

void relay_pwm_3(uint8_t rc)
{
                // Forward the byte to the first finger
                PORTA &= ~_BV(PA2);
                // Send the activation
                SPI_MasterTransmit(rc);
                // Turn off the SS line
                PORTA |= _BV(PA2);
}

void toggleIndicator(void) {
        // if the light is on turn it off
        if (PORTD & _BV(PD3)) {
                        PORTD &= ~_BV(PD3);
        }
        // else turn it on
        else {
                PORTD |= _BV(PD3);
        }
}

int main (void)
```

```
{
        ubusInit();
        ioinit();
        SPI_MasterInit();

        uint8_t i = 0;
        uint8_t relay_char;

        toggleIndicator();

        for (;;) {

                for (i=0;i<3;i++) {
                        toggleIndicator();
                        relay_char = USART_Receive();
                        relay_pwm_1(relay_char);
                }

                for (i=0;i<3;i++) {
                        toggleIndicator();
                        relay_char = USART_Receive();
                        relay_pwm_2(relay_char);
                }

                for (i=0;i<3;i++) {
                        toggleIndicator();
                        relay_char = USART_Receive();
                        relay_pwm_3(relay_char);
                }
        }
    return (0);
}
```

C.1.2 Slave micro-controller program

Listing C.2: *spi_slave.c* Slave micro-controller program

```
#include <inttypes.h>
#include <avr/io.h>
#include <avr/interrupt.h>
```

```
#include <avr/signal.h>

#  define OC1 PB1
#  define DDROC DDRB
#  define OCR OCR1A
#  define PWM10 WGM10
#  define PWM11 WGM11

#define F_CPU 1000000   //1 MHz

void SPI_SlaveInit(void) {
        volatile char IOReg;
        /* Set MISO output, all others input */
        DDRB = (1<<PB4);
        /* Enable SPI */
        SPCR = (1<<SPE);
        IOReg = SPSR;
        IOReg = SPDR;
}

char SPI_SlaveReceive(void) {
        /* Wait for reception complete */
        while(!(SPSR & (1<<SPIF)));
        /* Return data register */
        return SPDR;
}

void ioinit (void) {
        DDRD |= _BV(PD7);

        /* tmr0 is 8-bit PWM */
    TCCR0A = _BV (WGM00) | _BV (WGM01) | _BV (COM0A1) | _BV (COM0B1);
    TCCR0B = _BV (CS00);
        timer_enable_int (_BV (TOIE0));

        /* tmr2 is 8-bit PWM */
    TCCR2A = _BV (WGM20) | _BV (WGM21) | _BV (COM2B1);
    TCCR2B = _BV (CS20);
        timer_enable_int (_BV (TOIE2));

    /* set initial PWM values */
```

```
    OCR0A = 0x00;
        OCR0B = 0x00;
        OCR2B = 0x00;

        /* enable PWM outputs */
    DDRD |= _BV(PD3);
        DDRD |= _BV(PD5);
        DDRD |= _BV(PD6);
}

void toggleIndicator(void) {
        // if the light is on turn it off
        if (PORTD & _BV(PD7)) {
                        PORTD &= ~_BV(PD7);
        }
        // else turn it on
        else {
                PORTD |= _BV(PD7);
        }
}

int main (void) {

        PORTD |= _BV(PD7);

        ioinit();
        SPI_SlaveInit();
        sei();

        for (;;) {
        OCR0A = SPI_SlaveReceive();
                OCR0B = SPI_SlaveReceive();
                OCR2B = SPI_SlaveReceive();
                toggleIndicator();
        }

    return (0);
}
```

C.2 MATLAB Functions

C.2.1 Main program

Listing C.3: *main.m* Main program loop

```
% Initialize the external communication hardware
initSerial;
daq_init;

% Initialize the global variables
data;

% Initialize the local variables and allocate memory to improve
   performance
tf = 2;
global data_log1 data_log2 data_log3
max_iterations = 300;
data_log1 = zeros(max_iterations,27);
data_log2 = zeros(max_iterations,27);
data_log3 = zeros(max_iterations,27);
p_base = zeros(9,1);
logindex = 1;
step = 1;
terminate = false;
u_norm = zeros(9,1);
u = zeros(9,1);

% start the clock
initial_time = now;

% Main control loop
for t = 1:tf,

    converged = false;

    while (~converged),

    % Generate trajectory point for current time step
    p_global_hat = traj_gen(t);
```

```
    % Homogeneous transforms
    p_base_hat = homogeneous_xform(p_global_hat);

    % Inverse kinematics
    thetaA_hat = inversekinematics(p_base_hat(1:3));
    thetaB_hat = inversekinematics(p_base_hat(4:6));
    thetaC_hat = inversekinematics(p_base_hat(7:9));

    % Concatenate desired joint variables at current time step
    theta_hat = [thetaA_hat; thetaB_hat; thetaC_hat];

    % Acquire sensor data sample from DAQ
    sample = daq_sample;
    % Map sensor voltage into radians
    theta = sensor_decoding(sample);

    % Calculate the error signal
    e = theta_hat  - theta;

    % Check for convergence at each finger tip
    p_base(1:3) = forwardkinematics(theta(1:3));
    p_base(4:6) = forwardkinematics(theta(4:6));
    p_base(7:9) = forwardkinematics(theta(7:9));
    e_xyz1 = norm(p_base_hat(1:3)-p_base(1:3));
    e_xyz2 = norm(p_base_hat(4:6)-p_base(4:6));
    e_xyz3 = norm(p_base_hat(7:9)-p_base(7:9));

    if (e_xyz1<kappa),
            if (e_xyz2<kappa),
                    if (e_xyz3<kappa),
                        % All 3 finger tips within acceptable tolerance
                        converged = true;
                        disp(['Trajectory step ' num2str(t) ' reached.
                           (elapsed time: ' num2str(dt) ')']);
                    end
            end
    end
    if (~converged),
        % Compute normalized control signal
        u_norm = normalized_cs(e);
        % Rescale control signal to maximum allowable duty cycle
```

```
        % (element-wise multiplication)
        u = duty_range.*u_norm;
        % Transmit control signal to hand controller
        ioxmit(u);
        pause(0.1);
    end

    % Calculate the elapsed time in seconds
    dt = (now-initial_time)*24*60*60;

    data_log1(logindex,:) = [t dt p_global_hat(1:3)'  p_base_hat(1:3)'
       p_base(1:3)' e(1:3)' theta(1:3)' e_xyz1 u_norm(1:3)' u(1:3)'
       thetaA_hat'];
    data_log2(logindex,:) = [t dt p_global_hat(4:6)'  p_base_hat(4:6)'
       p_base(4:6)' e(4:6)' theta(4:6)' e_xyz2 u_norm(4:6)' u(4:6)'
       thetaB_hat'];
    data_log3(logindex,:) = [t dt p_global_hat(7:9)'  p_base_hat(7:9)'
       p_base(7:9)' e(7:9)' theta(7:9)' e_xyz3 u_norm(7:9)' u(7:9)'
       thetaC_hat'];

    % Terminate the loop after the log is exhausted
    if (logindex == max_iterations),
        converged = true;
        terminate = true;
        disp(['Maximum number of iterations reached. (elapsed time: '
           num2str(dt) ')']);
        break;
    end
    logindex = logindex + 1;

    end % end of convergance checking interior loop

    % If the logs are full, stop the program
    if (terminate),
        break;
    end

end % end of main control loop
disp('Main control loop exited');

% Turn off SMA
```

```
disp('Closing comm. links and powering down SMA');
u = zeros(9,1);
ioxmit(u);
% Close the external hardware communication links
tidySerial;
daq_tidy;

disp('Trimming unused data log');
% Trim unused data
data_log1 = data_log1(1:logindex-1,:);
data_log2 = data_log2(1:logindex-1,:);
data_log3 = data_log3(1:logindex-1,:);

% Clean up the CPU memory to improve plotting performance
pack;

% Plot the data
plot_results(data_log1,1);
plot_results(data_log2,2);
plot_results(data_log3,3);
```

C.2.2 Supporting functions

Listing C.4: *cal_sensor.m* Function to obtain joint sensor input-output correlation

```
function C = cal_sensor(S,Range,init)

% number of measurements to take, i.e. delta+1
max_rows = Range+1;


C = zeros(max_rows,2);

daq_init;

P = init;
for(row=1:max_rows)
    C(row,1) =P;
    P=P+1;
    pause;
```

```
    data = daq_sample;
    C(row,2) =data(S);
end

figure
plot(C(:,1),C(:,2),'o:')
title(['Calibration curve for sensor ',num2str(S)])
ylabel('Voltage (V)')
xlabel('Angular Displacement (\circ)')

figure
plot(C(:,2),C(:,1),'o:')
title(['Calibration curve for sensor ',num2str(S)])
xlabel('Voltage (V)')
ylabel('Angular Displacement (\circ)')

daq_tidy;
```

Listing C.5: *daq_init.m* DAQ system initialization

```
global ai;

ai = analoginput('nidaq',1);
set(ai,'InputType','SingleEnded');
chans = addchannel(ai,0:8);
set(ai,'SampleRate',10000)
set(ai,'SamplesPerTrigger',10)
```

Listing C.6: *daq_sample.m* DAQ sample acquisition

```
function data = daq_sample()

global ai
%global data

start(ai);
data = mean(getdata(ai));
%data(1)

% plot(data)
```

Listing C.7: *daq_tidy.m* DAQ shutdown process

```
delete(ai)
clear ai
```

Listing C.8: *data.m* Physical parameter initialization

```
% data.m
% Aaron Price, 1993953
% Definitions of physical parameters

global L1 L2 L3 rxA ryA rzA rxB ryB rzB rxC ryC rzC delta
global xA1 yA1 zA1 xB1 yB1 zB1 xC1 yC1 zC1
global ax ay az bx by bz cx cy cz
global kappa normalized_duty_offset duty_range mu

% link geometry [m]
L1 = 49.5e-3;
L2 = 32e-3;
L3 = 20e-3;
delta = 6.35e-3;

% Base frame translation offsets from World frame [m]
% Finger 1 (A)
rxA = 0.0;
ryA = 0.0;
rzA = 0.0;

% Finger 2 (B)
rxB = -5.65e-2;
ryB =  0.0;
rzB =  0.0;

% Finger 3 (C)
rxC =  -3.15e-2;
ryC =  0.0;
rzC =  6.65e-2;

% Cubic Trajectory Points (World frame initial and final)
% Currently Set to touch at a common point
xA1 = rxA;
yA1 = ryA+L1+L2+L3;
```

```
zA1 = rzA+delta;

xB1 = rxB;
yB1 = ryB+L1+L2+L3;
zB1 = rzB+delta;

xC1 = rxC;
yC1 = ryC+L1+L2+L3;
zC1 = rzC-delta;

% Object contact point mapping variables (Object Frame)
ax = -0.02;
ay = 0.0;
az = -0.005;

bx = 0.025;
by = 0.0;
bz = -0.005;

cx = 0.001;
cy = 0.0;
cz = 0.005;

% Cartesian error tolerance [m]
kappa = 5e-3;

% Maximum allowable duty cycle
min_duty(1) = 5;
min_duty(2) = 5;
min_duty(3) = 5;

min_duty(4) = 5;
min_duty(5) = 5;
min_duty(6) = 5;

min_duty(7) = 5;
min_duty(8) = 5;
min_duty(9) = 5;

duty_range = zeros(9,1);
```

```
duty_range(1) = 20;
duty_range(2) = 25;
duty_range(3) = 25;

duty_range(4) = 30;
duty_range(5) = 25;
duty_range(6) = 25;

duty_range(7) = 20;
duty_range(8) = 25;
duty_range(9) = 25;

normalized_duty_offset = zeros(9,1);
for i = 1:3
    normalized_duty_offset(i) = min_duty(i)/duty_range(i);
end

% Sigmoid function horiztonal shift (sensitivity)
mu = log(duty_range-1);
```

Listing C.9: *forwardkinematics.m* Finger forward kinematics

```
% forwardkinematics.m
% Aaron Price, 1993953
% Function for calculating the forward kinematics
% Arguments theta1: Variable of the first joint
%           theta2: Variable of the second joint
%           theta3: Variable of the third joint
%
% Output: XYZ_T: vector giving the position of the end of the tool
    frame

function [XYZ_T] = ForwardKinematics(q)

global L1 L2 L3 delta

theta1=q(1);
theta2=q(2);
theta3=q(3);

c1 = cos(theta1);
```

```
c2 = cos(theta2);
c3 = cos(theta3);
c23 = cos(theta2+theta3);
s1 = sin(theta1);
s2 = sin(theta2);
s3 = sin(theta3);
s23 = sin(theta2+theta3);

x = c1*(L1 + L2*c2 + L3*c23 - delta*s23);
y = s1*(L1 + L2*c2 + L3*c23 - delta*s23);
z = L2*s2 + L3*s23 + delta*c23;

% x = cos(theta1)*(-delta*cos(theta2)*sin(theta3)-delta*sin(theta2)*cos
    (theta3)+cos(theta2)*cos(theta3)*L3-sin(theta2)*sin(theta3)*L3+cos(
    theta2)*L2+L1);
% y = sin(theta1)*(-delta*cos(theta2)*sin(theta3)-delta*sin(theta2)*cos
    (theta3)+cos(theta2)*cos(theta3)*L3-sin(theta2)*sin(theta3)*L3+cos(
    theta2)*L2+L1);
% z = cos(theta2+theta3)*delta+L3*sin(theta2+theta3)+sin(theta2)*L2;

[XYZ_T] = [x; y; z];
```

Listing C.10: *homogeneous_xform.m* Coordinate transformations for the hand

```
% homogeneous_xform.m
% Aaron Price, 1993953
% Maps coordinates from world frame into base frame

function p_base =homogeneous_xform(p_global)

global rxA ryA rzA rxB ryB rzB rxC ryC rzC

A = [0 1 0 -ryA; -1 0 0 rxA; 0 0 1 -rzA; 0 0 0 1]*[p_global(1);
    p_global(2); p_global(3); 1];
B = [0 1 0 -ryB; -1 0 0 rxB; 0 0 1 -rzB; 0 0 0 1]*[p_global(4);
    p_global(5); p_global(6); 1];
C = [0 1 0 -ryC; 1 0 0 -rxC; 0 0 -1 rzC; 0 0 0 1]*[p_global(7);
    p_global(8); p_global(9); 1];

p_base = [A(1); A(2); A(3); B(1); B(2); B(3); C(1); C(2); C(3)];
```

Listing C.11: *initSerial.m* Serial bus initialization

```
global s

s = serial('COM3','BaudRate',4800,'DataBits',8,'StopBits',1);
set(s,'TimeOut',1);
fopen(s)
```

Listing C.12: *inversekinematics.m* Finger inverse kinematics

```
% inversekinematics.m
% Aaron Price, 1993953
% input:
%       end-effector positions = [x; y; z]
%
% output:
%       joint positions = [theta1; theta2; theta3]

function positions=inversekinematics(q)

Xc = q(1);
Yc = q(2);
Zc = q(3);

% Declare the global variables:
global L1 L2 L3 delta

theta1 = atan2(Yc,Xc);

phid = atan(delta/L3);
Ld = sqrt(L3^2+delta^2);

XYc = (Xc^2 + Yc^2)^0.5;

cphi = ((XYc - L1)^2 + Zc^2 - L2^2 - Ld^2)/(2*L2*Ld);
if (abs(cphi)) > 1
    error('Desired position is outside the work envelope');
end
sphi = sqrt(1-cphi^2);
phi = atan2(sphi,cphi);

theta3 = phi - phid;
```

```
c2 = (XYc - L1 + ((Ld*sphi*Zc)/(L2 + Ld*cphi)))/(L2 + Ld*cphi + (((Ld*
    sphi)^2)/(L2 + Ld*cphi)));
if (abs(c2)) > 1
    error('Desired position is outside the work envelope');
end
s2 = sqrt(1-c2^2);%* -1; % Note: the negative root for knuckle-up
    posture
theta2 = atan2(s2,c2);

positions = [theta1; theta2; theta3];
```

Listing C.13: *inv_homogeneous_xform.m* Inverse coordinate transformations for the hand

```
% homogeneous_xform.m
% Aaron Price, 1993953
% Maps coordinates from base frame into global frame

function p_global = inv_homogeneous_xform(p_base)

global rxA ryA rzA rxB ryB rzB rxC ryC rzC

A = [0 -1 0 rxA; 1 0 0 ryA; 0 0 1 rzA; 0 0 0 1]*[p_base(1); p_base(2);
    p_base(3); 1];
B = [0 -1 0 rxB; 1 0 0 ryB; 0 0 1 rzB; 0 0 0 1]*[p_base(4); p_base(5);
    p_base(6); 1];
C = [0 1 0 rxC; 1 0 0 ryC; 0 0 -1 rzC; 0 0 0 1]*[p_base(7); p_base(8);
    p_base(9); 1];

p_global = [A(1); A(2); A(3); B(1); B(2); B(3); C(1); C(2); C(3)];
```

ioxmit

Listing C.14: *normalized_cs.m* Compute the normalized control signal

```
function u_norm = normalized_cs(e)

global mu normalized_duty_offset

u_norm = zeros(9,1);

for i=1:9,
```

```
    if e(i) > 0,
        u_norm(i) = (1+exp(-180*e(i)/pi+mu(i)))^-1 +
           normalized_duty_offset(i);
    else
        u_norm(i) = 0;
    end
end

u_norm = real(u_norm);
```

Listing C.15: *plot_results.m* Visualize the experimental results

```
function plot_results(log_file, finger_num)

% Field order:
%
% step log_file(1)
% dt log_file(2)
% p_global_hat(1:3)' log_file(3) log_file(4) log_file(5)
% p_base_hat(1:3)' log_file(6) log_file(7) log_file(8)
% p_base(1:3)' log_file(9) log_file(10) log_file(11)
% e(1:3)' log_file(12) log_file(13) log_file(14)
% theta(1:3)' log_file(15) log_file(16) log_file(17)
% e_xyz1 log_file(18)
% u_norm(1:3)' log_file(19) log_file(20) log_file(21)
% u(1:3)' log_file(22) log_file(23) log_file(24)
% theta_hat log_file(25) log_file(26) log_file(27)

% Set LaPrint interpreter compatability
set(0,'defaulttextinterpreter','none');
L = length(log_file(:,1));

% Plot XYZ - base frame trajectories
figure;
% Plot x coords first
subplot(4,1,1);
plot(log_file(:,2),log_file(:,6),'-k',log_file(:,2),log_file(:,9),':k')
title(['Finger ' num2str(finger_num) ' base frame trajectories'])
ylabel('X_{base} (m)')
axis([0 log_file(L,2) 0 1])
axis 'auto y'
```

```
% Plot y coords
subplot(4,1,2);
plot(log_file(:,2),log_file(:,7),'-k',log_file(:,2),log_file(:,10),':k'
    )
ylabel('Y_{base} (m)')
axis([0 log_file(L,2) 0 1])
axis 'auto y'
% Plot z coords
subplot(4,1,3);
plot(log_file(:,2),log_file(:,8),'-k',log_file(:,2),log_file(:,11),':k'
    )
ylabel('Z_{base} (m)')
axis([0 log_file(L,2) 0 1])
axis 'auto y'
% Plot error
subplot(4,1,4)
plot(log_file(:,2),log_file(:,18),'-k')
ylabel('Total error (m)')
xlabel('Time (s)')
axis([0 log_file(L,2) 0 1])
axis 'auto y'

% Plot joint space trajectories next
figure;
% Plot theta1 first
subplot(4,1,1);
plot(log_file(:,2),log_file(:,25)*180/pi,'-k',log_file(:,2),log_file
    (:,15)*180/pi,':k')
title(['Finger ' num2str(finger_num) ' joint space trajectories'])
ylabel('\theta_1 (\circ)')
axis([0 log_file(L,2) 0 1])
axis 'auto y'
% Plot theta2
subplot(4,1,2);
plot(log_file(:,2),log_file(:,26)*180/pi,'-k',log_file(:,2),log_file
    (:,16)*180/pi,':k')
ylabel('\theta_2 (\circ)')
axis([0 log_file(L,2) 0 1])
axis 'auto y'
% Plot theta3
subplot(4,1,3);
```

```
plot(log_file(:,2),log_file(:,27)*180/pi,'-k',log_file(:,2),log_file
    (:,17)*180/pi,':k')
ylabel('\theta_3 (\circ)')
axis([0 log_file(L,2) 0 1])
axis 'auto y'
% Plot error
subplot(4,1,4)
plot(log_file(:,2),log_file(:,12)*180/pi,'k-',log_file(:,2),log_file
    (:,13)*180/pi,'k--',log_file(:,2),log_file(:,14)*180/pi,'k:')
ylabel('Joint error (\circ)')
xlabel('Time (s)')
axis([0 log_file(L,2) 0 1])
axis 'auto y'

% Plot control signal and error vs. time
figure;
% Error first
% Plot theta1 first
subplot(3,2,1);
plot(log_file(:,2),log_file(:,12)*180/pi,'-k')
title(['Finger ' num2str(finger_num) ' joint error'])
ylabel('\theta_1 error (\circ)')
axis([0 log_file(L,2) 0 1])
axis 'auto y'
% Plot theta2
subplot(3,2,3);
plot(log_file(:,2),log_file(:,13)*180/pi,'-k')
ylabel('\theta_2 error (\circ)')
axis([0 log_file(L,2) 0 1])
axis 'auto y'
% Plot theta3
subplot(3,2,5);
plot(log_file(:,2),log_file(:,14)*180/pi,'-k')
ylabel('\theta_3 error (\circ)')
xlabel('Time (s)')
axis([0 log_file(L,2) 0 1])
axis 'auto y'

% CS in right column
% Plot theta1 first
subplot(3,2,2);
```

```
plot(log_file(:,2),log_file(:,19),'-k')
title(['Finger ' num2str(finger_num) ' joint error'])
ylabel('\theta_1 normalized control signal')
title(['Finger ' num2str(finger_num) ' normalized control signal'])
axis([0 log_file(L,2) -0.2 1.2])
% Plot theta2
subplot(3,2,4);
plot(log_file(:,2),log_file(:,20),'-k')
ylabel('\theta_2 normalized control signal')
axis([0 log_file(L,2) -0.2 1.2])
% Plot theta3
subplot(3,2,6);
plot(log_file(:,2),log_file(:,21),'-k')
ylabel('\theta_3 normalized control signal')
xlabel('Time (s)')
axis([0 log_file(L,2) -0.2 1.2])
```

Listing C.16: *recover.m* Reinitialize hardware in case of transmission timeout

```
fclose(instrfind);
initSerial;
u = zeros(9,1);
ioxmit(u);
tidySerial;
```

Listing C.17: *sensor_decoding.m* Translate sensor voltage into joint deflection

```
% m-file to convert sensor voltage to joint angles based on calibration
% curves

function y = sensor_decoding(v)

y = zeros(9,1);

x = v(1);
y(1) = 5.0208*x^3 - 53.198*x^2 + 199.41*x - 261.74;
x = v(2);
y(2) = -15.559*x^2 + 122.25*x - 229.5;
x = v(3);
y(3) = 891.3*x^3 - 4089.3*x^2 + 6270.5*x - 3204.6;
x = v(4);
```

```
y(4) = -1.204*x^2 + 23.5*x - 70.08;
x = v(5);
y(5) = -135.56*x^2 + 792.19*x - 1147.4;
x = v(6);
y(6) = 505.48*x^3 - 2168.6*x^2 + 3116.2*x - 1490.3;
x = v(7);
y(7) = -3061*x^3 + 37256*x^2 - 1.5114e+005*x + 2.0439e+005;
x = v(8);
y(8) = 212.39*x^3 - 1333.7*x^2 + 2805.3*x - 1968.3;
x = v(9);
y(9) = 10185*x^3 - 26123*x^2 + 22375*x - 6391.7;
y = (pi/180)*y;
```

Listing C.18: *tidySerial.m* Shut down serial bus upon experiment completion

```
fclose(s);
```

Listing C.19: *traj_gen.m* Generate a trajectory for the hand to follow

```
function p = traj_gen(t)

global xB1 yB1 zB1 xC1 yC1 zC1

% Static step input:
if (t==1),
    F1 = forwardkinematics([0 12 10]*pi/180);
    F2 = forwardkinematics([-6 -190 52]*pi/180);
    F3 = forwardkinematics([14 25 755]*pi/180);
    p = inv_homogeneous_xform([F1; F2; F3]);
% Relax:
elseif (t==2),
    F1 = forwardkinematics([-5 10 8.5]*pi/180);
    F2 = forwardkinematics([-7 -192 51]*pi/180);
    F3 = forwardkinematics([13.5 23 753]*pi/180);
    p = inv_homogeneous_xform([F1; F2; F3]);
end
```