

**REINFORCEMENT LEARNING FOR ACTIVE LENGTH CONTROL AND
HYSTERESIS CHARACTERIZATION OF SHAPE MEMORY ALLOYS**

A Thesis

by

KENTON CONRAD KIRKPATRICK

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2009

Major Subject: Aerospace Engineering

**REINFORCEMENT LEARNING FOR ACTIVE LENGTH CONTROL AND
HYSTERESIS CHARACTERIZATION OF SHAPE MEMORY ALLOYS**

A Thesis

by

KENTON CONRAD KIRKPATRICK

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	John Valasek
Committee Members,	Dimitris Lagoudas
	Thomas Ioerger
Head of Department,	Dimitris Lagoudas

May 2009

Major Subject: Aerospace Engineering

ABSTRACT

Reinforcement Learning for Active Length Control and Hysteresis

Characterization of Shape Memory Alloys. (May 2009)

Kenton Conrad Kirkpatrick, B.S., Texas A&M University

Chair of Advisory Committee: Dr. John Valasek

Shape Memory Alloy actuators can be used for morphing, or shape change, by controlling their temperature, which is effectively done by applying a voltage difference across their length. Control of these actuators requires determination of the relationship between voltage and strain so that an input-output map can be developed. In this research, a computer simulation uses a hyperbolic tangent curve to simulate the hysteresis behavior of a virtual Shape Memory Alloy wire in temperature-strain space, and uses a Reinforcement Learning algorithm called Sarsa to learn a near-optimal control policy and map the hysteretic region. The algorithm developed in simulation is then applied to an experimental apparatus where a Shape Memory Alloy wire is characterized in temperature-strain space. This algorithm is then modified so that the learning is done in voltage-strain space. This allows for the learning of a control policy that can provide a direct input-output mapping of voltage to position for a real wire.

This research was successful in achieving its objectives. In the simulation phase, the Reinforcement Learning algorithm proved to be capable of controlling a virtual Shape Memory Alloy wire by determining an accurate input-output map of temperature

to strain. The virtual model used was also shown to be accurate for characterizing Shape Memory Alloy hysteresis by validating it through comparison to the commonly used modified Preisach model. The validated algorithm was successfully applied to an experimental apparatus, in which both major and minor hysteresis loops were learned in temperature-strain space. Finally, the modified algorithm was able to learn the control policy in voltage-strain space with the capability of achieving all learned goal states within a tolerance of $\pm 0.5\%$ strain, or $\pm 0.65\text{mm}$. This policy provides the capability of achieving any learned goal when starting from any initial strain state. This research has validated that Reinforcement Learning is capable of determining a control policy for Shape Memory Alloy crystal phase transformations, and will open the door for research into the development of length controllable Shape Memory Alloy actuators.

DEDICATION

This thesis is dedicated to my wife, Lindsay Kirkpatrick. She has been by my side through the entirety of this research and has driven me to excel with love and devotion.

ACKNOWLEDGEMENTS

I would like to thank my advisor and committee chair, Dr. Valasek, for his continued support during the duration of this research. Without his knowledge and guidance this research could not have been completed. I would also like to thank Dr. Lagoudas for both providing an experimental testing facility and serving on my committee. My thanks are also extended to Dr. Ioerger for providing invaluable direction with machine learning and for serving on my committee.

I also wish to thank all of my colleagues at Texas A&M University for helping me with specifics whenever I needed their aid, and for always perpetuating a learning atmosphere that is conducive to great research. My thanks also go out to the National Science Foundation for funding my graduate career and making it possible for myself and many others to participate in academic research.

Finally, my thanks are extended to my parents for supporting me through the years and always encouraging me to excel, as well as my wife, Lindsay, for never giving up on me and continually pushing me to be better.

NOMENCLATURE

SMA	Shape Memory Alloy
RL	Reinforcement Learning
M_s	Martensitic Starting Temperature
M_f	Martensitic Finishing Temperature
A_s	Austenitic Starting Temperature
A_f	Austenitic Finishing Temperature
ρ	Density of SMA Wire
c	Specific Heat of SMA wire
V_w	Volume of SMA Wire
T	Temperature of SMA Wire
V	Voltage Difference Across SMA Wire
t	Time
R	Electrical Resistance of SMA Wire
h	Convective Heat Coefficient of SMA Wire
A	SMA Wire Surface Area
T_∞	Ambient Temperature
ϵ	Exploration Probability
Q	Control Policy Matrix
s	Current State
a	Current Action

g	Current Goal
q	Current Step Subscript
η	Repetition Constant
δ	Control Policy Update Term
s'	Future State
a'	Future Action
γ	Future Policy Weight
r	Reward
ΔL	Change in SMA Wire Length
L	SMA Wire Martensitic Length
ϵ_w	Tensile Strain in SMA Wire
Pr	Probability
k	Number of Nearest Neighbors
n	Number of Attributes
d	Euclidean Distance
x	Instance
i,j	Instance Indices
a_r	Attribute
M_l	Major Loop Left Side
M_r	Major Loop Right Side
H	Hyperbolic Tangent Shape Parameter
ct_r	Hyperbolic Tangent Shape Parameter

c_{t_l}	Hyperbolic Tangent Shape Parameter
a_h	Hyperbolic Tangent Shape Parameter
s_h	Hyperbolic Tangent Shape Parameter
c_s	Hyperbolic Tangent Shape Parameter
α	Preisach Plane Independent Variable
β	Preisach Plane Dependent Variable
k_{BH}	Thermoelastic Constant – Bottom Heating
k_{TH}	Thermoelastic Constant – Top Heating
k_{BC}	Thermoelastic Constant – Bottom Cooling
k_{TC}	Thermoelastic Constant – Top Cooling
T_{0H}	Heating Phase Change Average Temperature
T_{0C}	Cooling Phase Change Average Temperature

TABLE OF CONTENTS

	Page
ABSTRACT	iii
DEDICATION.....	v
ACKNOWLEDGEMENTS.....	vi
NOMENCLATURE	vii
TABLE OF CONTENTS	x
LIST OF FIGURES	xi
LIST OF TABLES.....	xiii
 CHAPTER	
I INTRODUCTION	1
II REINFORCEMENT LEARNING	9
III CONTROL POLICY FUNCTION APPROXIMATION.....	21
IV EXPERIMENTAL APPARATUS DEVELOPMENT.....	24
V VALIDATION OF SIMULATION MODEL	31
VI TEMPERATURE-STRAIN CHARACTERIZATION	40
VII VOLTAGE-STRAIN LEARNING	47
VIII CONCLUSIONS	52
IX RECOMMENDATIONS.....	54
REFERENCES	56
VITA.....	62

LIST OF FIGURES

FIGURE	Page
1a Thermally Induced Phase Transformations for a Shape Memory Alloy....	3
1b Temperature-Strain Hysteresis for a Typical Shape Memory Alloy.....	3
2 Results of Shape Memory Alloy Hysteresis Simulation	6
3 Non-Markovian Travel in SMA Hysteresis.....	18
4 Markovian Travel in SMA Hysteresis.....	19
5 Experimental Apparatus	25
6 Experimental Hardware Setup.....	26
7 NiTi Major Hysteresis in Water-Filled Apparatus	27
8 Major Hysteresis in Antifreeze for NiTi SMA	29
9 Hardware / Software Connectivity for the Experimental Apparatus.....	30
10a Preisach Plane Positive Motion	32
10b Preisach Plane Negative Motion.....	32
11 Preisach Model of General Hysteresis.....	33
12 Preisach Simulation of SMA without Thermoelastic Boundary	35
13 Modified Preisach Model Simulation of SMA Hysteresis	38
14 Actions Required to Find Goal in Temperature-Strain Space	41
15 Hysteresis in Temperature-Strain Space after 37 Episodes.....	42
16a Path Taken by Learner after 12 Episodes	43
16b Path Taken by Learner after 23 Episodes	43

FIGURE	Page
16c Path Taken by Learner after 30 Episodes	44
17 Minor Hysteresis Loops Produced During Learning Episodes	45
18 Policy Test for Goal = 2.7%	48
19 Policy Test for Goal = 0.1%	50

LIST OF TABLES

TABLE	Page
1 Episodic ϵ -Greedy Values	15
2 Material Parameters Input to Preisach Model	34
3 Thermoelastic Parameters.....	37

CHAPTER I

INTRODUCTION

Advancement of aerospace structures has led to an era where researchers now look to nature for ideas that will increase performance in aerospace vehicles, particularly by advancing the research and development of bio- and nano-technology.¹ Birds have the natural ability to move their wings to adjust to different configurations of optimal performance. The ability for an aircraft to change its shape during flight for the purpose of optimizing its performance under different flight conditions and maneuvers would be revolutionary to the aerospace industry. To achieve the ability to morph an aircraft, exploration in the materials field has led to the idea of using Shape Memory Alloys (SMAs) as actuators to drive the shape change of a wing. The idea of using active materials for nonlinear structural morphing is being explored in a variety of ways with different types of smart materials are used, and SMAs are one field that shows promise.^{2,3,4,5,6} The field of SMA research has already begun branching into conceptualized morphing aircraft, with considerations to structure and aeroelasticity being considered.^{7,8,9} There are many types of SMAs which have different compositions, but the most commonly used SMAs are either a composition of nickel and titanium or the combination of nickel, titanium, and copper.

SMAs have a unique ability known as the Shape Memory Effect.^{10,11} This material can be put under a stress that leads to a plastic deformation and then fully

This thesis follows the style of the AIAA *Journal of Aerospace Computing, Information, and Communication*.

recover to its original shape after heating it to a high temperature. This would make SMAs useful for structures that undergo large deformations, such as morphing aircraft.¹² At room temperature, SMAs begin in a crystalline structure of martensite and undergo a phase change to austenite as the alloy is heated. This phase transformation realigns the molecules so that the alloy returns to its original austenitic shape. The original martensitic shape is re-obtained when the SMA is cooled back to a martensitic state, recovering the SMA from the strain that it had endured.

When a SMA wire undergoes a crystal phase transformation, it changes its length. The phase transformation from martensite to austenite (heating) causes a decrease in length while the reverse process extends it back to its original length. Control of this transformation is needed in order for morphing actuation to be possible, but it is difficult because the relationship between temperature and strain is highly nonlinear. The SMA wire exhibits a hysteresis behavior in its relationship between temperature and strain due to non-uniformity in the phase transformations.¹² This occurs because the phase transformation from martensite to austenite begins and ends at different temperatures than the reverse process, and the relationship is highly nonlinear. Figures 1a and 1b demonstrate this behavior, where in Figure 1a M_s is the martensitic starting temperature, M_f is the martensitic finishing temperature, A_s is the austenitic starting temperature, and A_f is the austenitic finishing temperature.

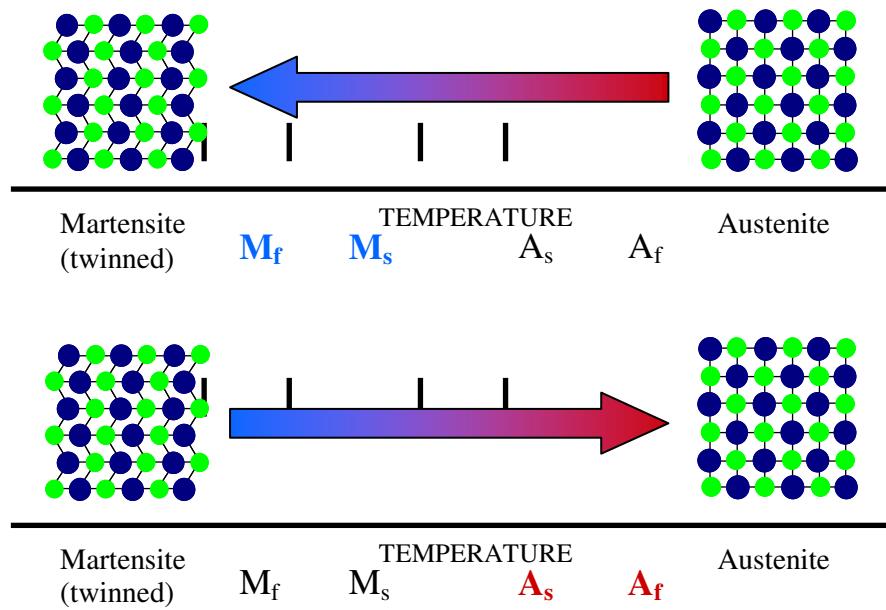


Fig. 1a Thermally Induced Phase Transformations for a Shape Memory Alloy

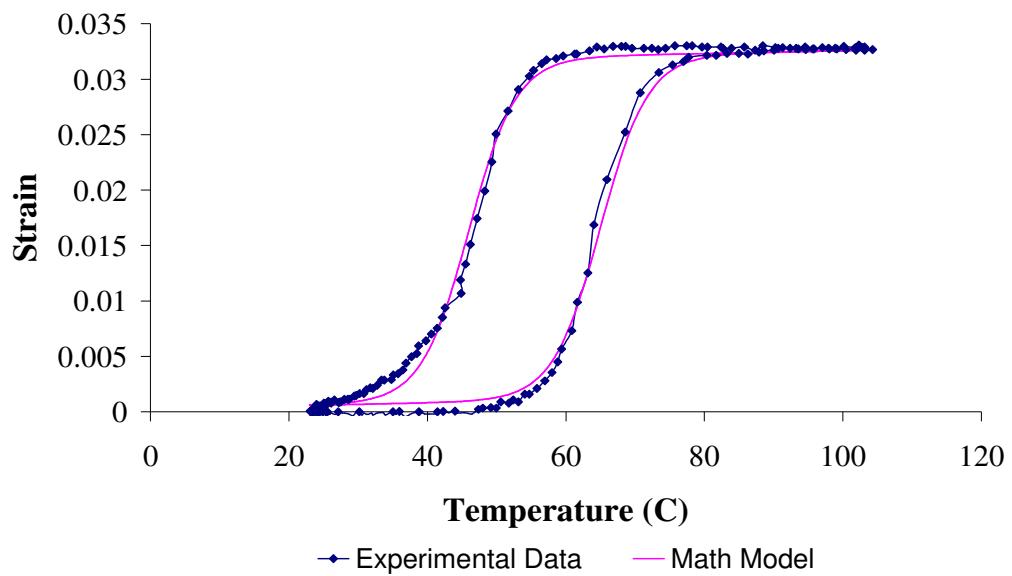


Fig. 1b Temperature-Strain Hysteresis for a Typical Shape Memory Alloy

One of the most common methods of controlling temperature in a SMA wire for inducing actuation is the use of heating by electrical resistance. The rate at which the wire changes temperature depends on the physical properties of the wire, the rate at which heat is lost to the environment, and the rate at which the wire heats due to electrical current. This can be modeled by a differential equation based on these parameters.

$$\rho c V_w \frac{dT}{dt} = \frac{V^2(t)}{R} - hA[T(t) - T_\infty] \quad (1)$$

In Equation (1), ρ is the SMA wire density, c is the specific heat of the wire, V_w is the SMA wire volume, T is the SMA wire temperature, V is the voltage difference in the SMA wire, t is time, R is the SMA wire electrical resistance, h is the convective heat transfer coefficient, A is the SMA wire surface area, and T_∞ is the ambient temperature of the coolant surrounding the SMA wire. The required voltage can be determined when temperature and its time derivative are known, but it can not be easily used in the case of SMAs. This is because the specific heat and convective coefficient change dynamically during crystal phase transformation. For this conversion between temperature and voltage to be useful for the learning agent, these two coefficients would also have to be learned, increasing the complexity of the state-space by two more dimensions. Therefore, it is a simpler process from a machine learning standpoint to learn the policy for voltage-strain directly rather than try to convert between temperature and voltage.

The hysteresis behavior of SMAs in temperature-strain space is most often characterized through the use of constitutive models that are based on material parameters or by models resulting from system identification.¹³ This is a time and labor intensive process that requires external supervision and does not actively discover the hysteresis in real-time, both of which are considerations that are undesirable for online learning of a control policy. Other methods that characterize this behavior are phenomenological models,^{14,15,16} micromechanical models,^{17,18} and empirical models based on system identification.^{19,20} These models are quite accurate, but some only work for particular types of SMAs and most require complex computations. Many of them are also unable to be used in dynamic loading conditions, making them unusable in the case of morphing. A drawback to using any of these methods is that the minor hysteresis loops within an SMA that is not fully actuated are not characterized and must be determined within analytical models. A simulated model of the major and minor hysteresis loops for a SMA wire is shown in Figure 2.

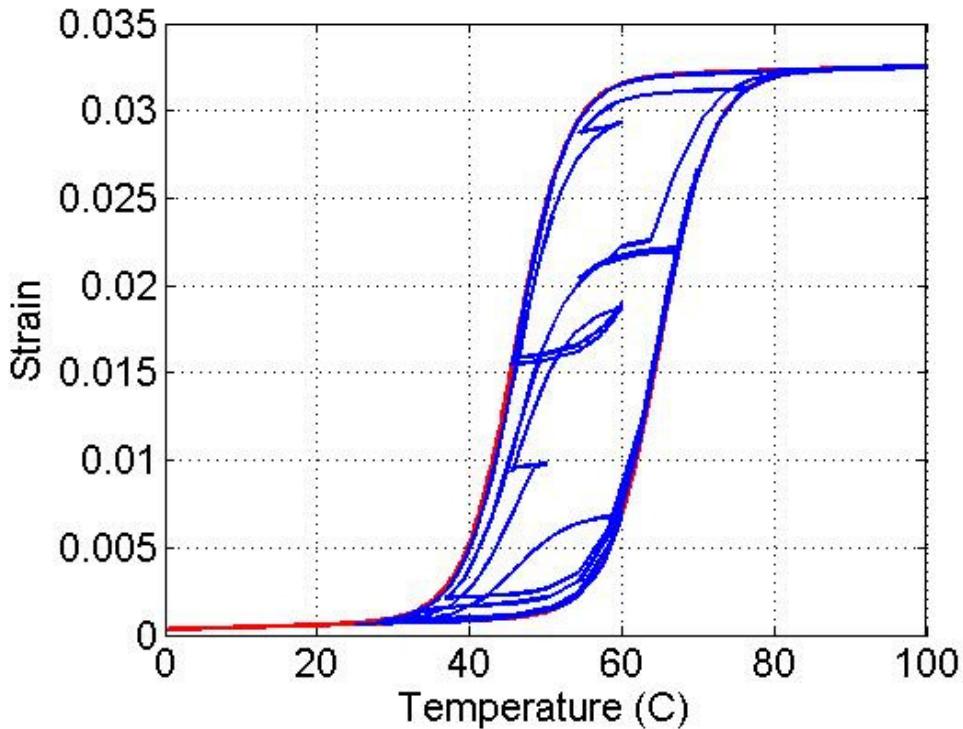


Fig. 2 Results of Shape Memory Alloy Hysteresis Simulation

This research investigates a technique for determining input-output policies for controlling SMA wires. Since there is not a parametric model available to use for this policy, this research uses a machine learning algorithm known as Reinforcement Learning (RL) to discover the black-box control policy for an SMA wire.²¹ RL is a form of machine learning that utilizes the interaction with multiple situations many times in order to discover the optimal path that must be taken to reach the pre-determined goal. By learning the behavior in real-time, both major and minor hysteresis loops can be experimentally determined through this method, while simultaneously learning the policy required for control.

RL is useful for morphing technology because it allows for a machine to learn the optimal control policy in real-time with no external supervision. By creating and updating a control policy based on states, actions, and goals, RL can begin by exploration and move to exploitation once it finds the optimal actions to take for each individual state. RL uses a process of rewards and consequences that allow the program to remember which actions are good at reaching the goal and which are poor.

RL is a tool that has already been demonstrated to be useful for morphing control architecture learning, and can be further implemented to achieve SMA shape control.^{22,23,24} The SMA phase transformation is not a thermodynamically reversible process. This leaves uncertainty in the model due to the highly non-linear behavior of the SMA. Since SMA phase change control depends on the voltage applied to the material, and there are currently not useful real-time parametric models of this relationship, the model needed to achieve characterizing the SMA morphing is unknown. This model needs to be determined by use of the RL algorithm in conjunction with the experimental setup so that a black box control policy can be determined for the use of SMA length control. Since RL does not require any prior knowledge of the control policy to discover it, exploiting RL for SMA length control is ideal.²⁵

The objective of the research presented in this thesis is to discover a method to control SMA wire length by characterizing a specimen and its hysteresis response using Reinforcement Learning methods. This was first accomplished in simulation, where a mathematical model of SMA hysteresis was used to provide states to the Reinforcement Learning agent. Once this was demonstrated to work in simulation, an experimental

apparatus was constructed and utilized to demonstrate that this method works in practice. The final stage seeks to obtain experimental results that verify that Reinforcement Learning was successful in characterizing SMA major and minor hysteresis loops, as well as learning how to control SMA wire length.

This thesis is organized as follows. In Chapter II, the basics of Reinforcement Learning are explained and extended to the specifics of this research. The details involving the Sarsa method, the ε -Greedy approach and function approximations are all discussed. Chapter III discusses the k -Nearest Neighbor algorithm, which was used for function approximation in this work. Chapter IV explains how the experimental apparatus was constructed for testing this method, and it includes details about problems which arose in active cooling and how it was solved. The details of how the RL agent is integrated with the experimental apparatus and is able to interact with the SMA wire in real-time are also explained. In Chapter V, the Preisach model is introduced and used to validate the hyperbolic tangent model used in the simulation phase of this research. The next chapter, Chapter VI, provides a detailed explanation of the characterization of SMA hysteresis behavior in temperature-strain space. Both simulation and experimentation results are presented and discussed. Chapter VII provides the results of the voltage-strain space learning and demonstrates the ability of RL to learn how to control a SMA wire. This is then followed by conclusions in Chapter VIII and recommendations in Chapter IX.

CHAPTER II

REINFORCEMENT LEARNING

Reinforcement Learning is a process of learning through interaction in which a program uses previous knowledge of the results of its actions in each situation to make an informed decision when it later returns to the same situation. It is a method that has been used for many diverse situations ranging from board games to behavior-based robotics.^{26,27,28} The purpose of the learning agent used in RL is to maximize the long-term cumulative reward, not just the immediate reward.²⁶ However, in this research there is only one dimension that yields any reward: strain. Since any goal strain is attainable within a certain error range based on knowledge of both current strain and current temperature, this implies that the agent maximizes rewards by minimizing the actions required to reach the goal strain, making the action associated with the maximum immediate reward also the action associated with the maximum cumulative award. The agent uses the knowledge gained by reward maximization to update a control policy that is a function of the states and actions. This control policy is essentially a large matrix that is composed of every possible state for the rows, and every possible action for the columns. In this research, a third dimension is included in the control policy that is composed of every possible goal state.

The three most commonly used classes of RL algorithms are Dynamic Programming, Monte Carlo, and Temporal Difference.²⁶ The majority of Dynamic Programming methods require an environmental model, making the use of them

impractical in problems with complex models. Monte Carlo only allows learning to occur at the end of each episode, causing problems that have long episodes to have a slow learning rate. Temporal Difference methods have the advantage of being able to learn at every time step without requiring the input of an environmental model. The most commonly used method of Temporal Difference is known as Q-Learning. Q-Learning is an off-policy form of Temporal Difference that utilizes an action-value function update rule based on the equation:

$$Q_q(s, a) \leftarrow Q_q(s, a) + \eta \delta_q \quad (2)$$

where s is the current state, a is the current action, Q is the control policy, and the q subscript signifies the current policy. The constant η is a parameter that is used to “punish” the RL algorithm when it repeats itself within each episode. The term δ_q is defined as:

$$\delta_q = r_{q+1}(s', a') + \gamma \max_a Q_{q+1}(s', a') - Q_q(s, a) \quad (3)$$

The term s' refers to the future state, a' is the future action, $q+1$ corresponds to the future policy, and γ represents a constant that is used to optimize the rate of convergence by weighting the future policy. Equations (2) and (3) can be combined to form the detailed Q-learning action-value function update rule:²⁶

$$Q_q(s, a) \leftarrow Q_q(s, a) + \eta[r_{q+1}(s', a') + \gamma \max_a Q_{q+1}(s', a') - Q_q(s, a)] \quad (4)$$

Q-Learning uses a simple algorithm involving the update rule in Equation (4) to reevaluate the Q matrix at each step. The Q-Learning algorithm is outlined as follow:²⁶

Q-Learning Method

- Initialize $Q(s, a)$ arbitrarily
- Repeat for each episode:
 - Initialize s
 - Choose a from s using policy derived from $Q(s, a)$ (e.g., ε -Greedy)
 - Repeat for each time step:
 - ❖ Take action a , observe r, s'
 - ❖ Choose a' from s' using policy derived from $Q(s, a)$ (e.g., ε -Greedy)
 - ❖ $Q(s, a) \leftarrow Q(s, a) + \eta \left[r + \gamma \max_a Q(s', a') - Q(s, a) \right]$
 - ❖ $s \leftarrow s'$
 - Until s is terminal

This research utilizes a method of Temporal Difference known as Sarsa. Sarsa is an on-policy form of Temporal Difference, meaning that at every time interval the control policy is evaluated and improved. An on-policy method is preferred here because the learning will occur in real-time, and the Q matrix needs to be updated in real-time as this learning progresses. Sarsa updates the control policy by using the current state, current action, future reward, future state, and future action to dictate the

transition from one state/action pair to the next.²⁶ The action value function used to update this control policy is:

$$Q_q(s, a) \leftarrow Q_q(s, a) + \eta[r_{q+1}(s', a') + \gamma Q_{q+1}(s', a') - Q_q(s, a)] \quad (5)$$

where it can be seen that Equation (5) differs from Equation (4) by using the actual future value rather than the value predicted by the maximum action value from the Q matrix.

The reward given for each state/action pair is defined by r , and the reward that is given for each situation is a user-defined parameter. For this research, a reward of 1 is given when a goal state is achieved, while a reward of 0 is given for any other state within the boundary. If the boundaries of the problem are exceeded, a reward of -1 is given to discourage following that path again. In this research, the control policy was modified to a three-dimensional matrix that includes the goal as the third dimension. By adding this third dimension, the control policy created by the RL algorithm can be represented as a set of tables that can be used to look up the correct voltage values needed when the current state and goal state are known. With g representing the goal state, the action value function now becomes:

$$Q_q(s, a, g) = Q_q(s, a, g) + \eta[r_{q+1}(s', a', g) + \gamma Q_{q+1}(s', a', g) - Q_q(s, a, g)] \quad (6)$$

This action-value function creates the policy that can be used to learn the parameters of the system being explored through RL. The Sarsa method uses a simple algorithm to update the policy using the action value function provided in Equation (6), and differs from the Q-Learning algorithm by only two steps. In Q-Learning, the Q matrix update rule follows Equation (5) while the Sarsa method uses Equation (6). The other difference is the fact that the Q-Learning algorithm updates the current state at every step while the Sarsa algorithm updates both the current state and current action according to the future state and action. This algorithm is outlined as follows:²⁶

Sarsa Method

- Initialize $Q(s,a,g)$ arbitrarily
- Repeat for each episode:
 - Initialize s
 - Choose a from s using policy derived from $Q(s,a,g)$ (e.g., ε -Greedy)
 - Repeat for each time step:
 - ❖ Take action a , observe r, s'
 - ❖ Choose a' from s' using policy derived from $Q(s,a,g)$ (e.g., ε -Greedy)
 - ❖ $Q(s,a,g) \leftarrow Q(s,a,g) + \eta [r + \gamma Q(s',a',g) - Q(s,a,g)]$
 - ❖ $s \leftarrow s', a \leftarrow a'$
 - Until s is terminal

When approaching the point in the algorithm where the action must be determined from Q , the problem of which method would be best for choosing this action must be solved. The dilemma lies in the fact that the policy does not have any

information about the system in the beginning, and must explore so that it can learn the system. The point of using RL is to learn the system when no prior knowledge of the system is known by the algorithm, so it can not exploit previous knowledge in the beginning stages. However, in future episodes the policy will have more information about the system, and exploitation of knowledge becomes more favorable. The key to optimizing the convergence of the RL module upon the best control policy is to balance the use of exploration and exploitation.

The ε -Greedy method of choosing an action is used in this research, which means that for some percentage of the time that an action is chosen, the RL module will choose to randomly explore rather than choose the action that the action-value function declares is the best.²⁹ This is because the RL agent might not have already explored every possible option, and a better path may exist than the one that is presently thought to yield the greatest reward. A fully greedy method chooses only the optimal path without ever choosing to explore new paths, which corresponds to an ε -Greedy method where $\varepsilon = 0$. The ε -Greedy action-value method can be implemented by the following algorithm:

ε -Greedy Action-Value Method

- Repeat for each action value:
 - Choose ε between 0 and 1
 - Generate random value β between 0 and 1
 - If $\beta \geq 1 - \varepsilon$
 - $a \leftarrow$ random
 - If $\beta < 1 - \varepsilon$
 - $a \leftarrow$ RL control policy exploitation

To converge on the optimal control policy in the shortest amount of time, this research used an episodically changing ϵ -Greedy method by altering the exploration constant, ϵ , depending upon the current episode. ϵ is a number between 0 and 1 that determines the percent chance that exploration will be used instead of exploitation. In the first episodes, little to no information has been learned by the policy, so a greater degree of exploration is required. Conversely, in future episodes less exploration is desired so that the RL module can exploit the knowledge of the system that it has learned.

To achieve an episodically changing ϵ -Greedy method, a simple algorithm was constructed that determines what value would be used for ϵ at each individual episode. The values of ϵ ranged from 70% in the first several episodes to 5% in the final episodes, and were chosen during simulation by experimenting with the values and episode numbers until the best convergence time was found. Even during later episodes, the algorithm still never exhibits a fully greedy method of choosing actions. A small chance of performing exploratory actions is still used because it allows the system to check for better paths in case the path it converged upon is not actually the most optimal choice. The ranges of episodes for each value of ϵ are as follows (Table 1):

Table 1: Episodic ϵ -Greedy Values

Episodes	1- 29	30 - 59	60 - 79	80 - 99	100 - 139	140+
ϵ	0.7	0.6	0.5	0.3	0.2	0.05

In this research, the states are defined by the current strain and temperature, while the actions are defined by the desired voltage that is applied to the SMA wire. The purpose of the RL agent is to converge to the optimal voltage needed to produce the desired strain based on the current strain in the wire. Conversion between strain and position is a trivial process, so strain was used as the state choice so that it can be global to specimens of different length. The conversion from strain to change in length is as follows:

$$\Delta L = \varepsilon_w L \quad (7)$$

In Equation (7), ΔL is the change in length, ε_w is the strain, and L is the original martensitic length of the SMA wire. The goal that the system is attempting to reach is the desired strain of the SMA wire.

For RL to be used to learn an input-output relationship, the environment needs to have the Markov property. In an environment with the Markov property, learning how to move from one state to another depends only on the current state, and not state history.^{26,30} In a general environment, the probability of achieving a specific goal and thereby obtaining a specific reward depends on the current and past states, actions, and rewards. This is demonstrated in Equation (8).²⁶

$$\Pr\{s_{t+1} = s', r_{t+1} = r | s_t, a_t, r_t, s_{t-1}, a_{t-1}, \dots, r_1, s_0, a_0\} \quad (8)$$

In an environment that has the Markov property, the probability distribution described by Equation (8) can be simplified to only depend on the current state and action. The dynamics of the system can be fully described by only using the probability of achieving a certain state and obtaining the associated reward given the current state. The Markov property probability distribution is represented by Equation (9).²⁶

$$\Pr\{s_{t+1} = s', r_{t+1} = r | s_t, a_t\} \quad (9)$$

Hysteresis is non-Markovian in nature because moving from one state to another requires knowing not only the current state but also the state history. In this research, this would imply that due to the hysteresis, both the current strain and past strains would be needed to know how to reach the goal strain. This is a problem when using RL because the control policy learned by RL is a function of only the current state, action, and goal. However, this problem was overcome by the specific formulation of this learning environment. Hysteresis is non-Markovian in the case of attempting to move from one strain in the hysteretic space to another, as shown in Figure 3.

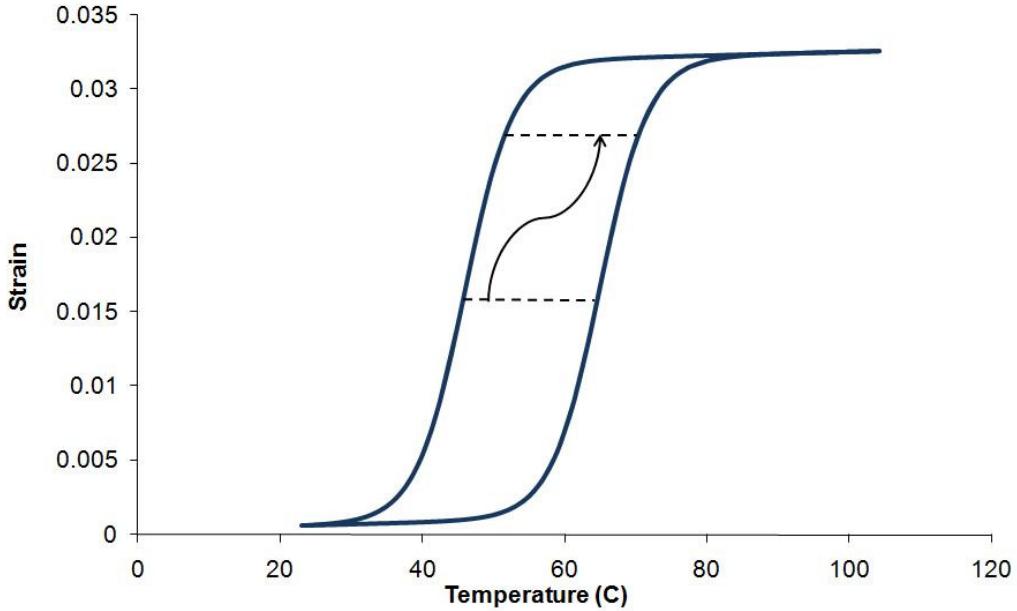


Fig. 3 Non-Markovian Travel in SMA Hysteresis

Attempting to learn this motion using RL would be a challenge since moving from one strain to another in a hysteretic environment requires strain time history to be known. However, in this research the current state of the system is not simply the current strain, but both current strain and temperature. The goal in this research is to move from one specific point in temperature-strain space to any point along the goal strain line in one action, without any restrictions on goal temperature. This type of learning environment is represented in Figure 4.

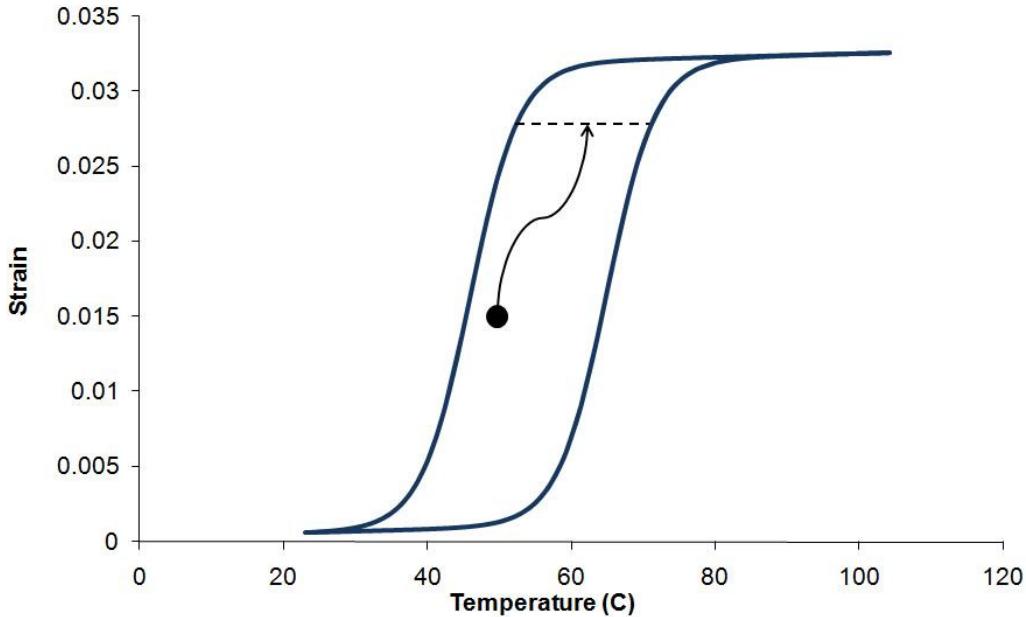


Fig. 4 Markovian Travel in SMA Hysteresis

The learning environment described by Figure 4 allows travel from any one point in the hysteresis space to anywhere along the horizontal goal line in the hysteresis space without knowing state history, indicating that it is Markovian. The only reason history of strain would be needed would be in the event that temperature was not measured, in which case the agent would need to know where along the horizontal line of current strain it lies. In the learning environment used in this research, temperature is directly measured by means of a thermocouple. The need to know strain state history is eliminated by the inclusion of a temperature dimension, indicating that the environment is Markovian. Using this construct of the learning environment, RL can be used for learning the optimal control policy.

Once the RL algorithm learns the optimal voltage required to achieve each goal strain from each initial strain, it can then be used to control the length of a SMA wire in real-time. The learned policy's ability to control the SMA wire's length can then be demonstrated and plotted for validation.

CHAPTER III

CONTROL POLICY FUNCTION APPROXIMATION

The problem that arises from using Reinforcement Learning for a continuous state-space is that the learned control policy is a discrete table of values. When RL is applied to a system with a continuous state-space, the state-space is not entirely represented. Since SMA wire length is continuous, the policy must be approximated for a continuous system.³¹ The use of function approximation is beneficial to this research because it allows the agent to explore a state-space that has fewer discrete values, resulting in shorter learning times. Without function approximation, the state-space would have to be finely discretized into strain states representing every possible state that could be desired. The resulting state-space would be too large to feasibly explore. By using function approximations, a coarser state-space discretization can be used by the agent, decreasing the overall learning time.

However, problems can arise with function approximation because the approximation can cause higher inaccuracies in the discrete points. The values within the control policy that are associated with each state-action pair are determined to be the optimal choice by the learning agent. When a function approximation is used to smooth the discrete matrix into a continuous system, these values can be changed. Some algorithms commonly used for function approximations are least mean squares, artificial neural networks, and self-organizing maps. While these methods would be capable of

handling the approximation of the policy learned in this research, a simpler but still accurate method is more desirable due to the low dimensionality of the state-space.

This research uses k -Nearest Neighbor for function approximation because it is a simple method that is fast, accurate, and stable. The k -Nearest Neighbor method is an instance-based machine learning algorithm that learns an approximation of a target function by means of assigning values to attributes associated with the k -nearest points in Euclidean distance to the target instance.³² The Euclidean distance is the geometric distance between instances in n -dimensional space, where n is the total number of attributes, and is denoted by Equation (10).³²

$$d(x_i, x_j) \equiv \sqrt{\sum_{r=1}^n (a_r(x_i) - a_r(x_j))^2} \quad (10)$$

In Equation (10), d is the Euclidean distance, r is the attribute index, n is the total number of attributes, a is the attribute, and x_i and x_j are the instances between which the distance is being measured.

The idea that is assumed when the k -Nearest Neighbor algorithm is used is that the properties of a point in the state-space are likely to be very similar to the properties of the points that have similar attribute values.³³ In this research, the state-space is 1-dimensional in both formulations of the problem that are explored. For the initial characterization approach, the two input attributes were strain and temperature, but these were combined into one attribute in order to overcome the non-Markovian behavior of

hysteresis. As a result, the state-space is 1-dimensional in the control policy. During the final phase of the experiment, the only input attribute considered was the 1-D strain. Since the state-space is 1-dimensional and the requirement is to achieve a direct input-output policy, the most logical choice for the k -Nearest Neighbor algorithm is to set $k = 1$.

For a 1-Nearest Neighbor approach the algorithm becomes simplified: an instance becomes classified by setting it equal to the value of the instance closest to it. By using this approach, the discrete control policy can be approximated for the continuous system. At any instance where the current or goal strain lie between two strains represented in the control policy, the assigned value is equal to the value associated with the closest strain.

CHAPTER IV

EXPERIMENTAL APPARATUS DEVELOPMENT

For an SMA wire to be used for experimental verification of this approach, a physical experimental apparatus was first constructed. The SMA is mounted in an apparatus that is constructed of Plexiglas and aluminum supports. The apparatus is sealed so that no fluid can leak out as the experimentation is proceeding. The wire is attached to the walls by Kevlar chords and is set in series with a free-weight that is attached by Kevlar over a dual pulley system. The mass of the free-weight changes depending on the diameter of the wire being tested, and is selected so that the wire experiences a stress of approximately 120MPa in its initial martensitic state at zero strain and zero voltage.

A Linear Position Transducer (LPT) is supported above the fluid by an aluminum beam, and the probe end is connected to the Kevlar chords for position measurement without receiving current from the SMA wire. The LPT sends a voltage to the Data Acquisition (DAQ) board which changes depending on the position of the probe. A variable voltage supply is used to provide a voltage difference across the wire for heating and is connected to the SMA wire via alligator clips that are positioned carefully along the wire so that every specimen tested maintains the same effective initial length. The voltage supply receives its commands from the DAQ board with an input/output voltage ratio of 3.6 and outputs voltages in the range of 0.00V-2.80V. For the input to the RL algorithm, the temperature and strain are required measurements to describe the

state of the system. Voltage is the control used to affect change in temperature, thereby changing the strain. A thermocouple is attached to the SMA wire for temperature measurements and sends small voltages to the DAQ board that are converted to temperature. Figure 5 shows the complete experimental apparatus.

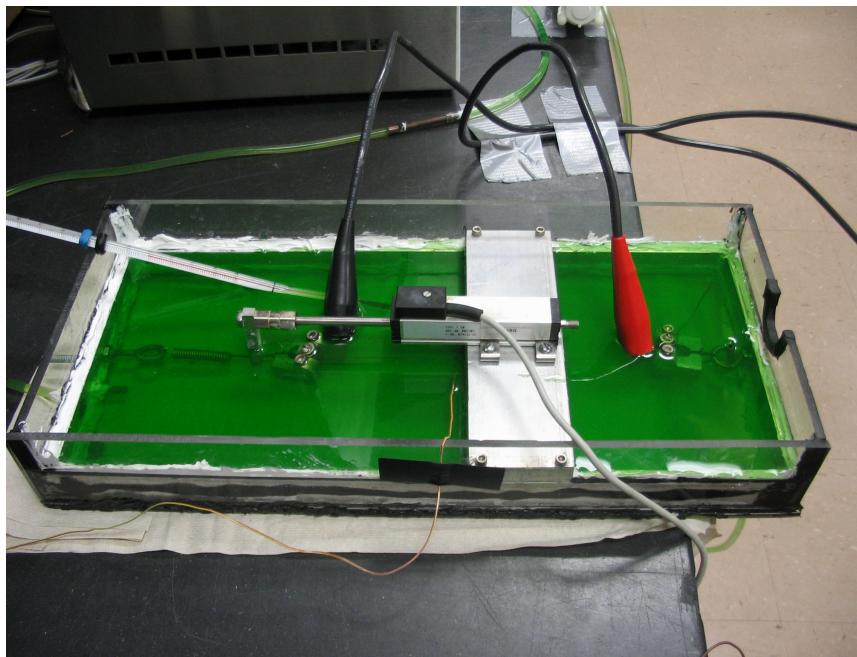


Fig. 5 Experimental Apparatus

The apparatus contains a pool of antifreeze which completely submerges the SMA wire and the alligator clips to allow sufficient cooling of the wire for prevention of overheating and to decrease the time required for the reverse phase transformation from austenite to martensite. The antifreeze is drawn out of the apparatus by a pump that sends it into a pool for temperature regulation. The external pool contains both heating and cooling coils that allow it to keep the antifreeze at a specified ambient temperature.

In this research, the ambient temperature is kept at 21°C. The cooled antifreeze is then drawn back out of the temperature regulation pool by another pump and is sent into the apparatus to continue fluid circulation and keep the coolant at a constant room temperature. Figure 6 shows the complete experimental hardware setup.



Fig. 6 Experimental Hardware Setup

Antifreeze was used as the coolant in this experiment because it was concluded that it was the best coolant choice available. Water was originally assumed to be a good fluid to use as it was readily available and has low electrical conductivity. Temperature regulation for water is also very easy, making it an obvious choice for the coolant.

However water transfers heat too easily, leading to poor temperature measurements by the thermocouple. This occurs due to the fact that the thermocouple experiences large temperature differences between the water touching the wire and the water at ambient temperature. In addition, water cannot exceed 100°C while in its liquid state so measurements at high temperatures become highly inaccurate. The water also causes some current loss due to impurities in the water so that high voltages (10-12V) are required to achieve full actuation. This unfortunately causes not only a greater need for power, but some of the extra current that is lost to the water occasionally interferes with thermocouple signals. The characterization of the major hysteresis loop in temperature-strain space using direct user input for a water-filled apparatus is shown in Figure 7.

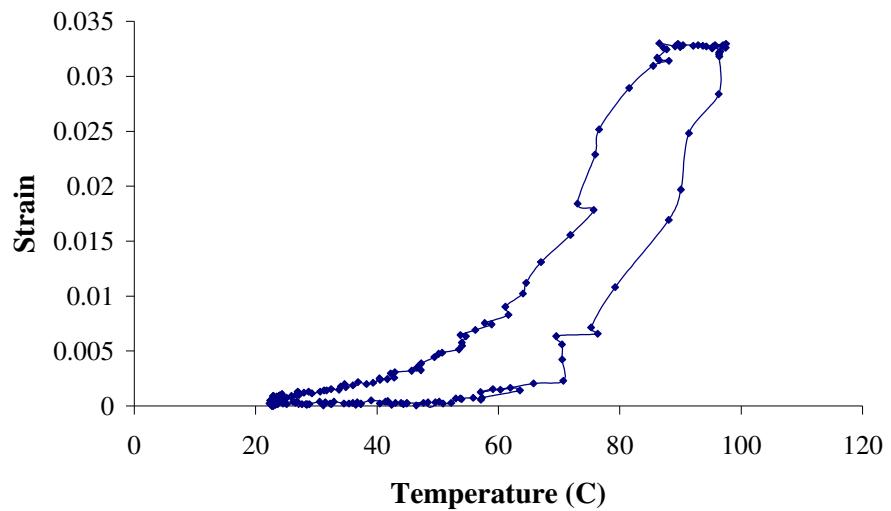


Fig. 7 NiTi Major Hysteresis in Water-Filled Apparatus

By using ethylene glycol (antifreeze) as a coolant instead of water, these problems can be overcome. Antifreeze does not transfer heat as easily as water so the ambient temperature in the apparatus does not affect the antifreeze that touches the SMA wire as quickly. This allows for much smoother temperature measurements throughout the experiment, although slower temperature propagation does cause a more delayed phase transformation back to martensite. This fact helps the accuracy of temperature measurements, but makes the process take as much as 10-15 seconds longer. Since these current experiments are static, the slower transformations are only a slight nuisance. When this research is later extended to dynamic control involving successive, immediate shape changes, this issue will need to be addressed.

Antifreeze also has the ability to greatly exceed the previous limit of 100°C without boiling, thereby eliminating the turbulence effects caused by water at high temperatures and allowing for better measurements. Antifreeze is a very good electrical insulator, and by using antifreeze, full actuation can occur with 2.8V instead of the 12V required in water. The characterization of the major hysteresis behavior using direct user input in an antifreeze-filled apparatus is shown in Figure 8.

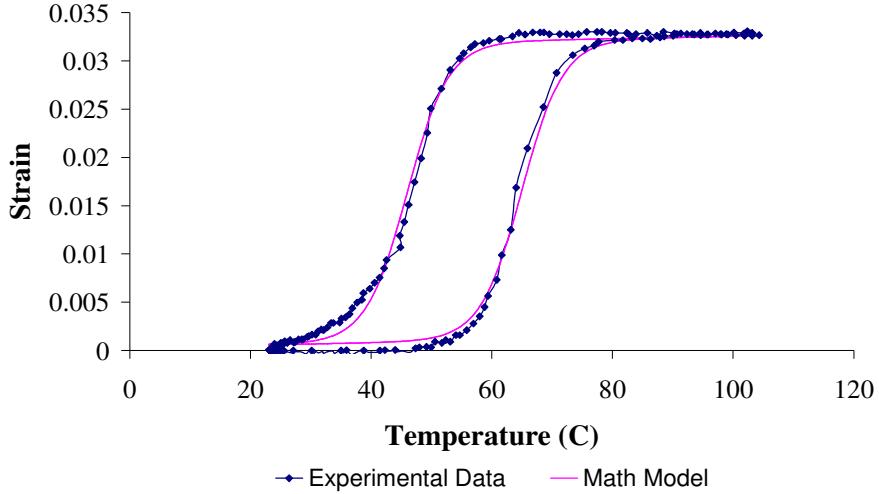


Fig. 8 Major Hysteresis in Antifreeze for NiTi SMA

In Figure 8, the experimental results are compared to the hyperbolic tangent model that was used in the simulation portion of SMA characterization. This model is based on a hyperbolic tangent curve that is represented by Equations (11) and (12):

$$M_l = \frac{H}{2} \tanh\left(\left(T - ct_l\right)a_h\right) + s_h \left(T - \frac{(ct_l + ct_r)}{2} \right) + \frac{H}{2} + c_s \quad (11)$$

$$M_r = \frac{H}{2} \tanh\left(\left(T - ct_r\right)a_h\right) + s_h \left(T - \frac{(ct_l + ct_r)}{2} \right) + \frac{H}{2} + c_s \quad (12)$$

In these equations, H , ct_r , a_h , s_h , ct_l , and c_s are constants that determine the shape of the hyperbolic tangent model. M_r and M_l are the strain values that correspond to the

temperature input into the equations. The constants were selected by creating a curve that best fit a known hysteresis behavior for a SMA wire.

For the RL MATLAB script to converse with the experimental setup, an interface was created using the software program LabVIEW. This program uses graphical functions to create a program capable of communicating with external hardware. The DAQ board relays the input voltages from the thermocouple and the LPT to the computer via a DAQ card installed in the computer. The constructed LabVIEW program takes these voltages and converts them into the current temperature and strain readings. These inputs are sent to MATLAB for use by Reinforcement Learning and then MATLAB sends LabVIEW the value of the voltage that needs to be applied to the wire in order for the desired strain to be reached. LabVIEW then transfers this voltage to the DAQ board, which sends the signal to the variable voltage supply, telling it to output the required voltage to the SMA wire. In this manner, the RL script is able to learn the required control policy of a real, physical SMA wire in an experimental setup. The block diagram shown in Figure 9 reveals the structure of this setup.

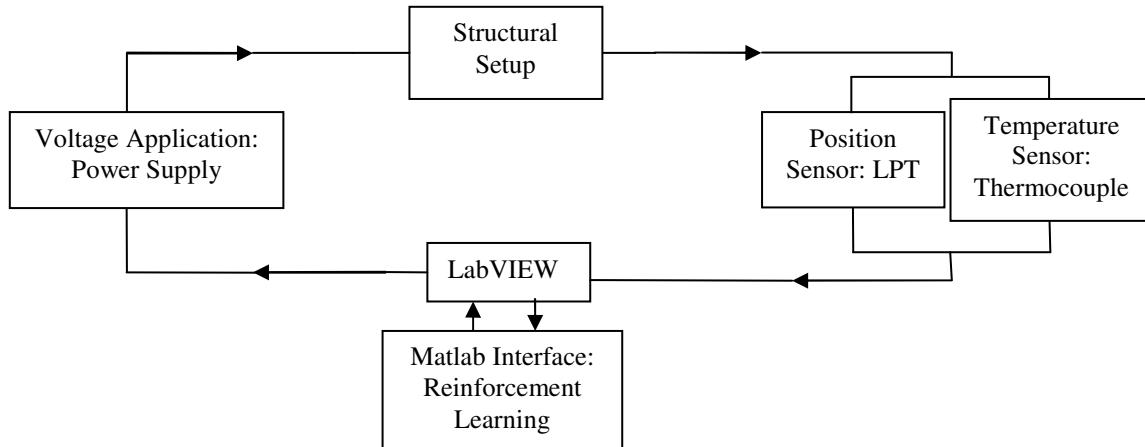


Fig. 9 Hardware / Software Connectivity for the Experimental Apparatus

CHAPTER V

VALIDATION OF SIMULATION MODEL

Simulation of SMA hysteresis behavior requires finding an accurate method of modeling this hysteresis. As was discussed in the previous chapter, the method used to model SMA hysteresis for this research was a hyperbolic tangent function that was curve fit to closely approximate experimental measurements of SMA hysteresis. This method was chosen because it provided both accurate results and a simple function that could be called during the RL simulation. However, for validation of this approach it is necessary to compare it to a more commonly used method of SMA hysteresis simulation.

One of the most widely used methods of approximating hysteresis behavior among SMA researchers is the Preisach Model.^{34,35} The Preisach Model is a general method of mapping hysteresis behavior that uses system parameters, and can be used for a wide variety of hysteretic environments, not only SMA hysteresis.³⁶ This is accomplished by mapping the direction-dependent curve area from the Preisach Plane to the hysteresis space. The Preisach Plane is a triangular region that retains state memory and uses this to map the area to a new function. Figures 10a and 10b represent travel in the Preisach Plane.

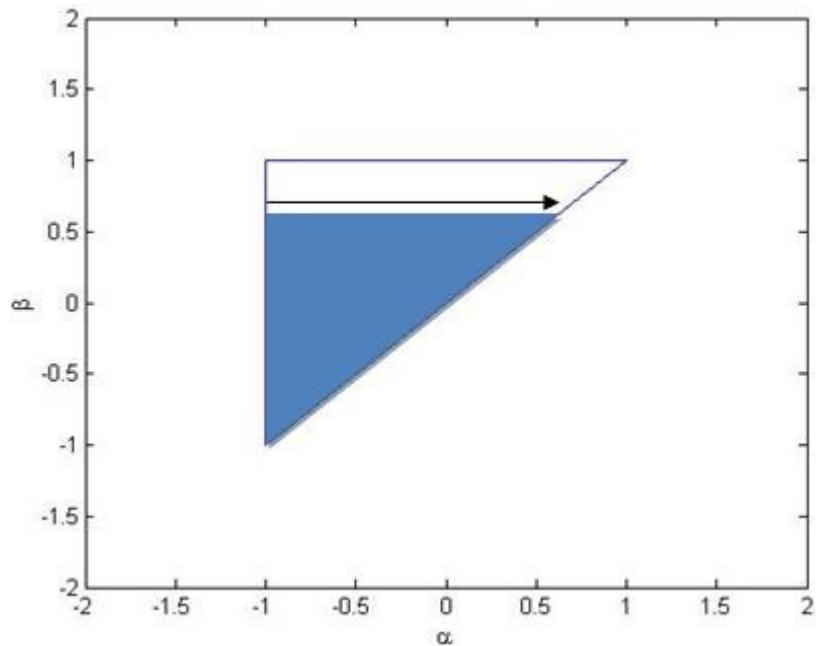


Fig. 10a Preisach Plane Positive Motion

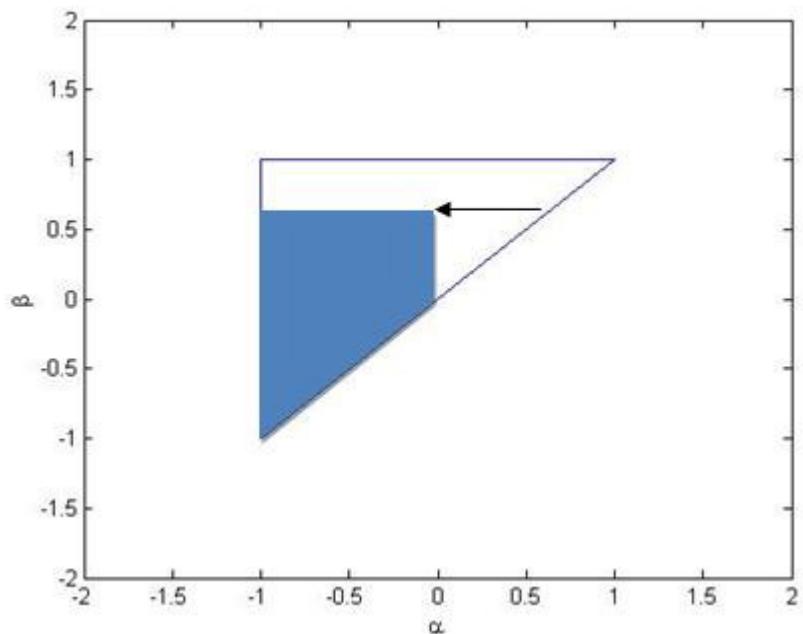


Fig. 10b Preisach Plane Negative Motion

As Figure 10a shows, when α is increased the effective area becomes the area of the Preisach plane that lies below the horizontal line associated with $\beta = \alpha$. However, Figure 10b shows that the effective area used for mapping is different when the value of α is decreased because the area subtracted from it is taken from the vertical line associated with $\alpha = \beta$. The effective area in the Preisach Plane is plotted as a function of α , and this new function is hysteretic. Figure 11 reveals the Preisach function corresponding to α traveling along the path $\alpha_{\min} \rightarrow \alpha_{\max} \rightarrow \alpha_{\min}$.

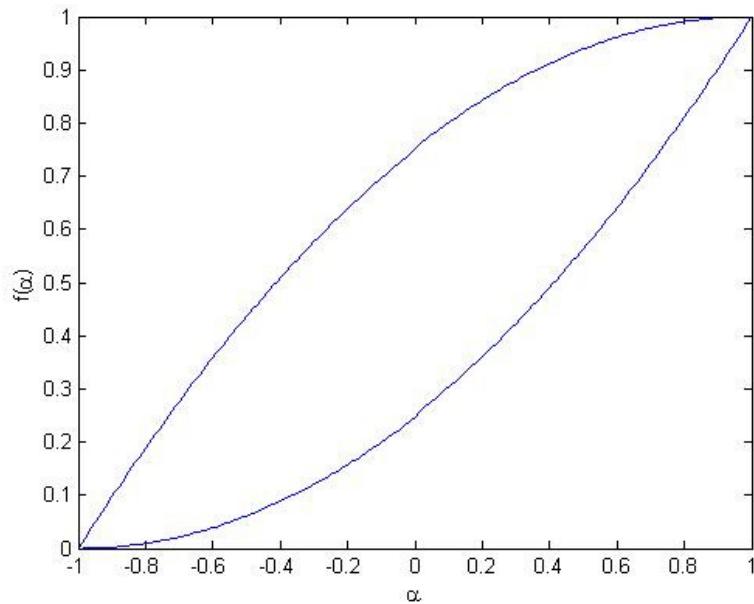


Fig. 11 Preisach Model of General Hysteresis

Figure 11 is an example of a general hysteresis loop that was mapped using the Preisach model. By adjusting the parameters of this model using the parameters associated with the SMA material properties, this loop can be adjusted to correspond

with SMA hysteresis behavior. The SMA wire that is used in this research has the following properties associated with the crystal phase transformation:

Table 2: Material Parameters Input to Preisach Model

M _s	M _f	A _s	A _f	ε _{w_min}	ε _{w_max}
60°C	35°C	45°C	75°C	0	0.033

By using the Table 2 values for temperature to define the α -axis and the strain values to define the β -axis of the Preisach plane, the hysteresis mapping can now approximate the major hysteresis behavior of the SMA wire being simulated according to these parameters. Likewise, using interior values for the minimum and maximum temperatures and strains can allow for approximate mapping of the minor hysteresis loops. These values are based on the strains achieved by the SMA used in the experimental apparatus, because it is important for the simulation to match the experimental parameters. SMA phase transformation is highly dependent on variations in mechanical loading and temperature changes, so the precise parameters must be used in the model to make sure the Preisach model matches correctly.³⁷ After applying these parameters and approximating the major and minor hysteresis loops, the Preisach model of this SMA wire can be simulated, as is shown in Figure 12.

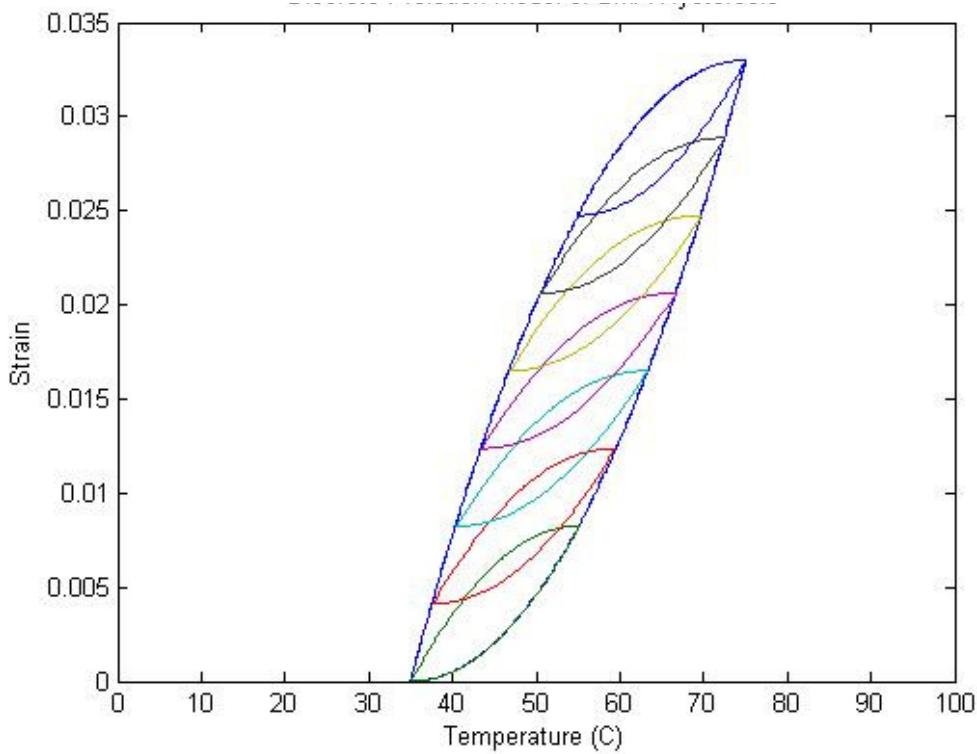


Fig. 12 Preisach Simulation of SMA without Thermoelastic Boundary

Figure 12 reveals that this simulation provides a good estimation of the hysteresis behavior in the interior of the transformation, but the minimum and maximum strain regions do not accurately reflect the thermoelastic effects that occur. To compensate for this, the Preisach model can be modified to include thermoelastic effects for these two regions.³⁸ Simulating the thermoelastic effects of the minimum and maximum strain regions can be accomplished by calculating these strain values according to Equation (13).

$$\varepsilon_w(T) = \frac{1}{1 + \exp[k_{--}(T - T_0)]} \quad (13)$$

In Equation (13), the values for $k_{..}$ and T_0 are dependent upon whether it is a heating or cooling process, as well as the material parameters being used. The values for $k_{..}$ and T_0 used to best approximate the SMA hysteresis behavior of the wire used in this research are calculated according to Equations (14)-(19). In these equations, the subscript H corresponds to heating while subscript C corresponds to cooling, and the subscript B corresponds to the bottom of the loop while the subscript T corresponds to the top of the loop.

$$T_{0H} = \frac{A_s + A_f}{2} \quad (14)$$

$$T_{0C} = \frac{M_s + M_f}{2} \quad (15)$$

$$k_{BH} = \frac{5.6}{A_s - A_f} \quad (16)$$

$$k_{TH} = \frac{4.3}{A_s - A_f} \quad (17)$$

$$k_{BC} = \frac{3.6}{M_f - M_s} \quad (18)$$

$$k_{TC} = \frac{4.5}{M_f - M_s} \quad (19)$$

Using Equations (14)-(19), the values for the thermoelastic parameters were calculated and can be seen in Table 3.

Table 3: Thermoelastic Parameters

T_{0H}	T_{0C}	k_{BH}	k_{TH}	k_{BC}	k_{TC}
60°C	47.5°C	-0.1867 /°C	-0.1433 /°C	-0.1440 /°C	-0.1800 /°C

Using the parameters in Table 3 with Equation (13), the Preisach model approximation of SMA hysteresis behavior can be modified to include thermoelastic effects at the minimum and maximum strain regions. These modifications can be seen in Figure 13.

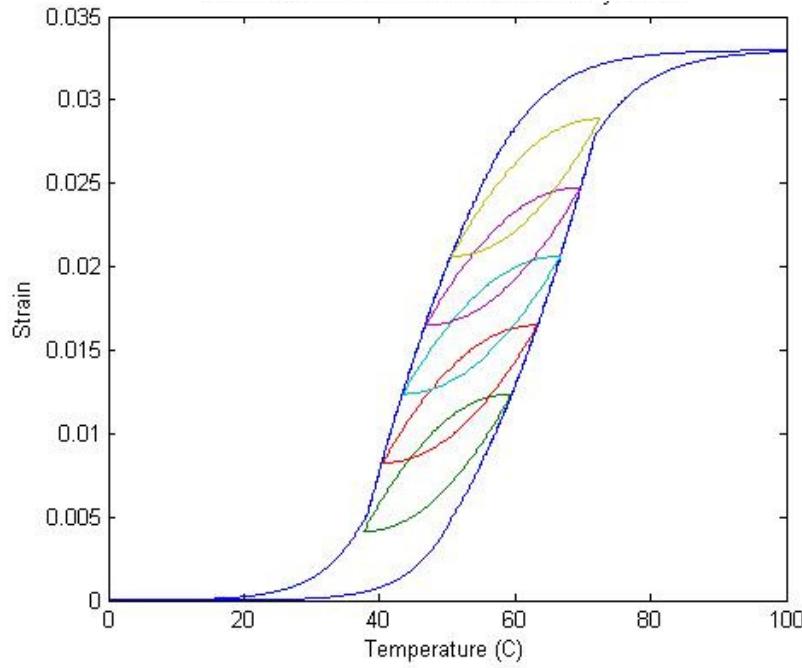


Fig. 13 Modified Preisach Model Simulation of SMA Hysteresis

The modifications to the Preisach model reflected in Figure 13 create a more accurate simulation of SMA hysteresis behavior. This simulation matches the experimental data with a maximum normalized error of 0.14. This model is often used to simulate SMA hysteresis behavior, but is limited by time complexity. For the simulation to be viable for use in online RL, the feedback from the simulation must be nearly immediate due to the real-time nature of the learning. Using this modified Preisach model, the time required to compute the major hysteresis loop in MATLAB is 144 seconds. This time is unacceptable for a real-time learning system. Due to this problem, a faster simulation is required for SMA hysteresis. As discussed in the previous chapter, the simulation chosen for this research was a curve fit using a set of

hyperbolic tangent functions. The hyperbolic tangent approximation was chosen over the Preisach model because the time required to compute a simple one-line function is nearly instantaneous. The average time required for MATLAB to process the hyperbolic tangent function and temperature propagation is 0.5 milliseconds. This speed allows for the real-time feedback needed to accomplish online learning of SMA hysteresis. The hyperbolic tangent model also has the benefit of making a closer curve-fit to the experimental data, with a maximum normalized error of 0.03.

The drawback to using the hyperbolic tangent model rather than the modified Preisach model is that the hyperbolic tangent function is not parameterized by the material properties. The constants used in Equations (11) and (12) are numerical in nature and are chosen purely for obtaining the closest fit possible to experimental data. The Preisach model has the benefit of being parameterized according to the crystal phase transformation temperatures and the minimum and maximum strains. However, this benefit is outweighed by the harsh time constraints. The hyperbolic tangent model is more useful for online RL because the simple function design allows for real-time feedback of the simulation.

CHAPTER VI

TEMPERATURE-STRAIN CHARACTERIZATION

Testing is conducted in temperature-strain space over many episodes at several different goal states corresponding to individual strain states, where an episode is defined as the achievement of a goal. With the current configuration, 3.3% strain is the maximum strain possible due to a completed crystal phase transformation to austenite. To demonstrate the convergence of the RL program, a goal state of 2.7% was investigated in detail. This goal was chosen because it requires nearly complete actuation of the SMA wire, but does not reach a fully actuated state. This forced the RL program to find the correct temperature exactly. When the maximum goal state of 3.3% is chosen the state is achieved more easily since any temperature exceeding the austenite finish temperature will yield a fully actuated strain state. This makes observing an intermediate strain state much more useful.

Figure 14 shows the relationship between the episodes completed and the total Reinforcement Learning actions attempted for reaching a goal of 2.7% strain. Every episode presented in this data begins at a fully un-actuated strain of 0%. As this graph shows, the RL algorithm takes fewer actions to achieve the desired goal state as it experiences more episodes. This proves that the RL becomes more successful in completing its objective of finding the optimal temperature required to achieve this goal state as it continues to learn.

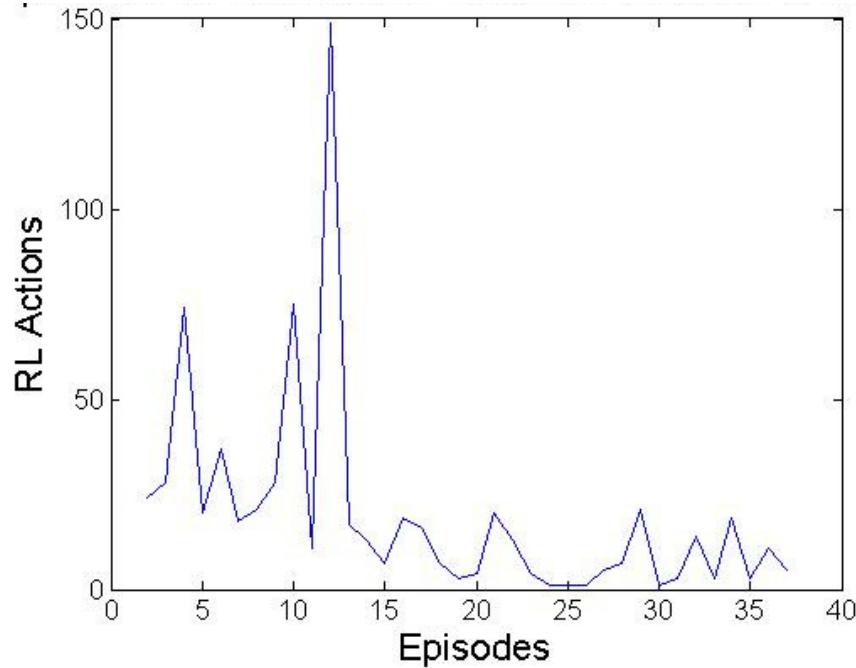


Fig. 14 Actions Required to Find Goal in Temperature-Strain Space

Figure 14 reveals that the control policy begins learning enough about the system to obtain the desired strain with only a few actions by the time it has reached 20 to 25 episodes. However, it can also be seen that even after this point there are a few episodes that required a larger number of actions to find the goal. This happens for 2 main reasons. Since the RL algorithm being used incorporates the logic of the ϵ -Greedy method, even after the algorithm begins converging on the optimal policy exploration is still encouraged to allow the system to find a better path to goal state achievement. The other reason that it still does not exhibit perfect control is because the measurements of the thermocouple are inaccurate during the intermediate phase changes, and can

sometimes be off by as much as 10°C. This can cause problems with the learning process that require many more episodes to achieve an optimal policy.

Over the course of 37 episodes to a goal state of 2.7% strain and back to a goal state of 0% strain, the major hysteresis behavior becomes visible. Figure 15 shows that the major hysteresis behavior is experimentally attainable from Reinforcement Learning.

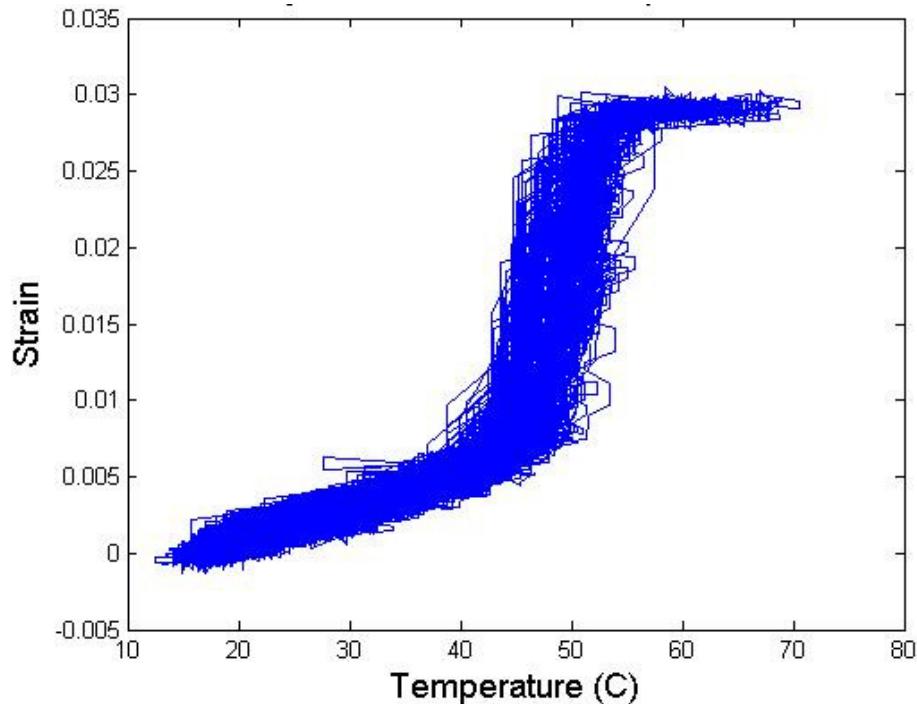


Fig. 15 Hysteresis in Temperature-Strain Space after 37 Episodes

The progression of the control policy's ability to obtain the hysteresis behavior was also of interest from this experiment. This information shows how well the experiment was able to utilize the learning capabilities of a RL algorithm. Figures 16a-

16c show the paths that are taken to obtain the final goal state for three different episodes that are represented in the convergence behavior shown in Figure 14.

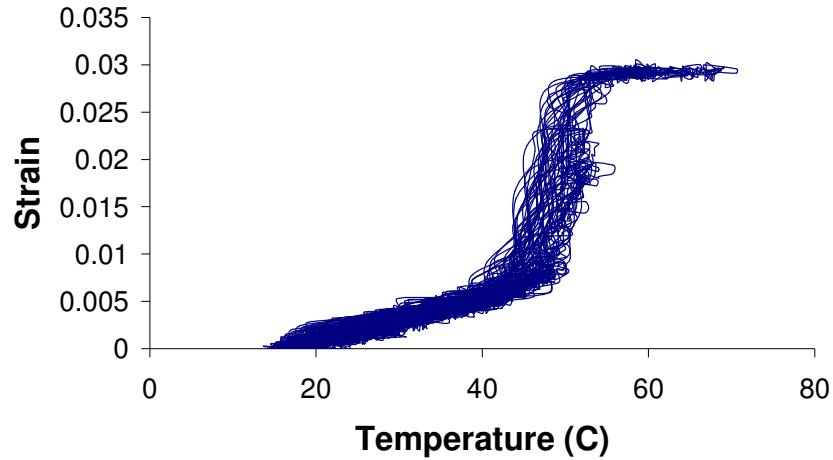


Fig. 16a Path Taken by Learner after 12 Episodes

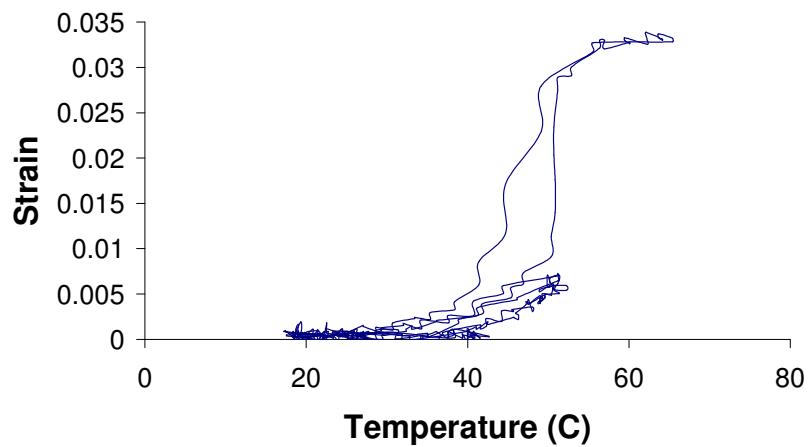


Fig. 16b Path Taken by Learner after 23 Episodes

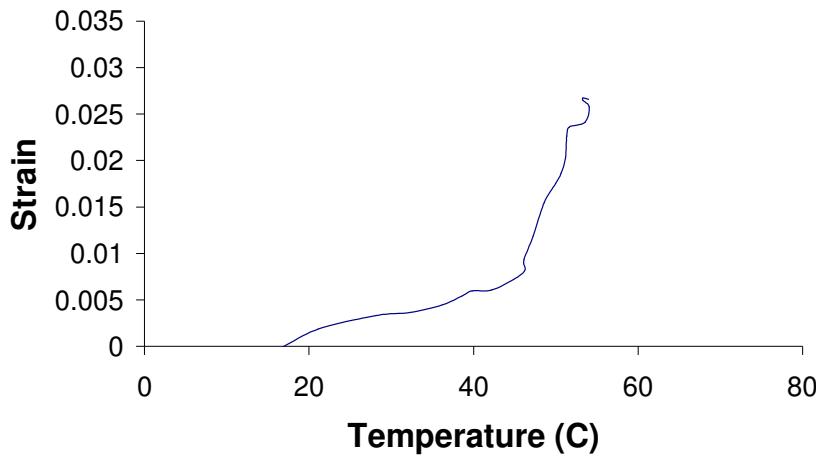


Fig. 16c Path Taken by Learner after 30 Episodes

During episode 12, the experimental system required 147 actions to achieve the goal strain of 2.7%. As a result, the system wandered between many different temperatures before it was finally able to find the temperature that would yield the correct goal strain. After running more similar episodes, the control policy learned how to achieve the goal state while taking fewer actions. By episode 23, only 4 actions were required to achieve the goal of 2.7% strain. Episode 30 demonstrates the control policy's ability to find the correct goal state in only 1 action. Figure 16c shows the affects of the RL algorithm's convergence upon an optimal control policy.

Reinforcement Learning's ability to find a control policy that learns the minor hysteresis behavior of a Shape Memory Alloy was of special interest because minor hysteresis loops are difficult to obtain by other methods. By using RL to characterize the hysteresis behavior, the minor loops are obtained just as easily as the major loops.

The minor hysteresis behavior can be extracted from individual episodes, as is demonstrated in Figure 17.

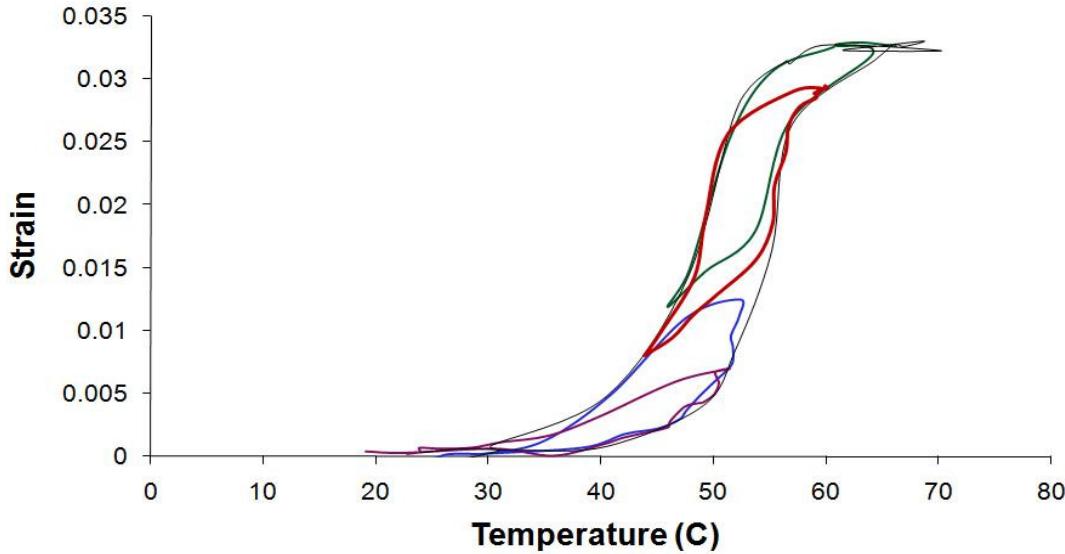


Fig. 17 Minor Hysteresis Loops Produced During Learning Episodes

Figure 17 represents the extraction of the major hysteresis loop and 3 minor hysteresis loops from episode 12 of the 2.7% goal experimentation. Normally these minor loops must be obtained by using mathematical models based on the major hysteresis behavior, but this shows that the minor hysteresis loops can be experimentally obtained through the RL method. The real-time data collection as the RL algorithm experimentally determines how to achieve each goal state allows both major and minor hysteresis loops to be mapped precisely. This is of particular importance for extension to voltage-strain space control because it shows that the control policy learned by RL

can achieve a goal state starting from any initial state, not just the fully un-actuated or actuated states.

CHAPTER VII

VOLTAGE-STRAIN LEARNING

The control policy developed for the particular SMA specimen tested provided the ability to control the length of a NiTi SMA wire for 2 specific goal strains within an error range of ± 0.005 strain. The wire used for this experiment had an initial effective length of 13cm, so with a maximum strain possible of 3.3%, the total operating range of motion was 4.29mm. Since the control policy learned was able to reach its goal within a range of $\pm 0.5\%$, the error range allowed was ± 0.65 mm. Under these specified conditions, the RL module was executed for 100 episodes using specified alternating goal strains of 2.7% and 0.1%, providing 50 episodes per goal. Each episode in this experiment consists of 450 seconds worth of seeking a single goal, where the RL module is called every 15 seconds. This provides 30 new actions per episode for the learning module.

The first goal presented is 2.7% strain. This goal was chosen for experimentation because it represents a partially actuated state for which the maximum strain of 3.3% falls outside of the allowed tolerance range of $\pm 0.5\%$. This ensures that it can not achieve the goal by simply applying the maximum voltage available. This goal is also of particular interest since it was previously used for temperature-strain space validation. Under these conditions, the final control policy was tested and the results can be seen in Figure 18.

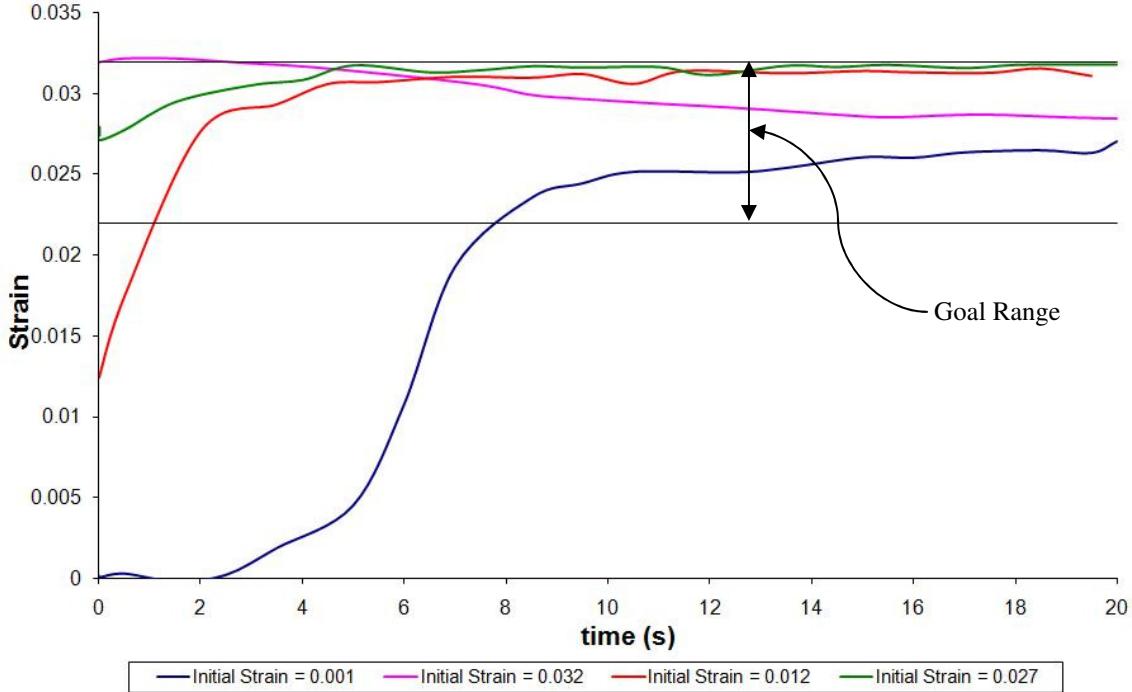


Fig. 18 Policy Test for Goal = 2.7%

Figure 18 reveals that the control policy developed by the RL agent is capable of bringing an SMA wire to the desired goal from multiple initial positions. This ability makes the development of morphing actuators possible. In Figure 18, the initial strains chosen for testing here were 0.1%, 3.2%, 1.2%, and 2.7%. The two horizontal lines represent the goal range of $2.7\% \pm 0.5\%$ strain. The initial strains of 0.1% and 3.2% were chosen so that the control policy could be tested from initial strains corresponding to fully un-actuated and fully actuated states, respectively. The initial strain of 1.2% was selected in order to test from an initially intermediate strain, and the goal strain of 2.7% was also chosen as an initial strain to show that the agent can learn how to stay within

the specified range when the specimen is there initially. As Figure 18 shows, the control policy was successful in achieving its goal of $2.7\% \pm 0.5\%$ in all 4 test cases.

Using RL to learn a control policy capable of achieving a strain that rests within the interior of the transformation curve is important because it greatly increases the range of functionality of SMA actuators. If the only values learned by the agent are those that correspond to maximum and minimum strains, a SMA actuator would be limited to only two possible positions. Learning these interior goals is also far more complicated than learning the extreme values because all that would be required for the latter would be to apply the maximum and minimum voltages every time. By showing that this RL approach can learn how to reach 2.7% strain, this research has proven that using a RL agent to learn a SMA control policy makes it possible to create a SMA actuator capable of achieving multiple position changes.

The second goal that was chosen for experimental learning was 0.1%. This goal was chosen because it represents a state that is not quite on the boundary of the system, but effectively is on the boundary because the lower bound is encompassed by the tolerance range. While it could achieve its goal by applying 0 volts, it is not limited to this action. Figure 19 shows the results of testing the control policy for a goal of 0.1% strain.

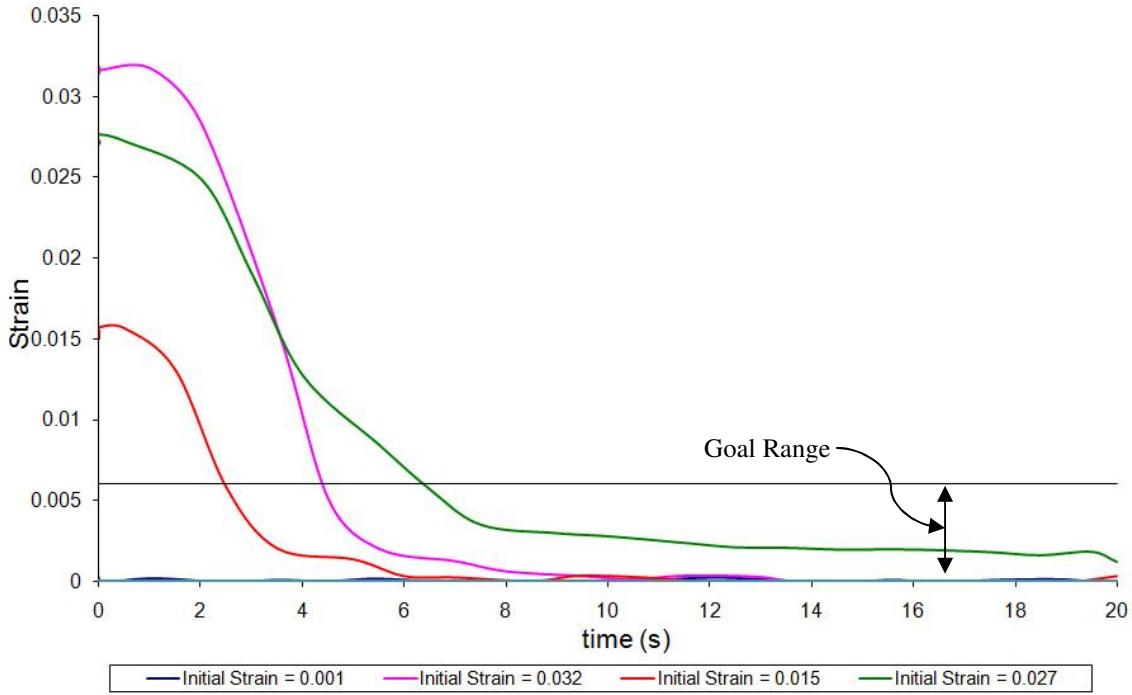


Fig. 19 Policy Test for Goal = 0.1%

The horizontal black line represents the upper bound of the tolerance range while the lower bound corresponds to a strain of 0%. The initial strains chosen for Figure 19 were 0.1%, 3.2%, 1.5%, and 2.7%, which are nearly identical to the initial strains chosen in Figure 18. The 0.1% strain was chosen because it demonstrates the ability of the system to remain at the goal strain when already there, and 3.2% was selected because that it is the other system boundary. The other strains were chosen because they nearly match the initial strains used in the previous test. Figure 19 shows that for each of these initial strains, the control policy is able to achieve its specified goal, but here it was accomplished for the goal of $0.1\% \pm 0.5\%$ strain.

Just as it was important to show that this approach allows for the ability to control SMAs in the interior of the transformation process, it was also important to reveal that RL is not limited to the interior. By demonstrating that the control policy is able to also learn how to move the SMA wire back to its initial position, this research has proven that using a RL approach provides the ability to learn both the extreme positions and the interior positions. It follows from these tests that creating SMA actuators for the purpose of developing morphing aircraft is feasible.

CHAPTER VIII

CONCLUSIONS

Based upon the analysis and results presented in this thesis, the following conclusions are made:

- 1) It was determined that using a hyperbolic tangent model for simulation is more feasible than a thermoelastic Preisach model for a real-time learning procedure like Reinforcement Learning. The hyperbolic tangent model provided a more accurate fit to the experimental data with maximum error of 0.03 versus the Thermoelastic Preisach model's maximum error of 0.14. The hyperbolic tangent was also a faster method than the Preisach model. Using MATLAB, the Thermoelastic Preisach model takes 144 seconds to characterize the major hysteresis while the hyperbolic tangent model takes 0.5 milliseconds.
- 2) Although hysteresis space is classically considered to be non-Markovian, the Shape Memory Alloy's temperature-strain space can be made Markovian by measuring the temperature and using it to increase the dimensionality of the state-space. Measuring strain state history is only needed to know what the current temperature is, so measuring temperature directly eliminates the need to know strain history. This is not mathematically proven, but is validated for this case by the learning results.
- 3) The results of the experimental stage established the ability to learn a control policy in an online experiment without human external supervision, and validated

the approach experimentally. With the tolerances chosen for goal achievements, the Reinforcement Learning agent was able to converge to a near-optimal policy within 100 episodes. The feasibility of achieving goals on both the boundary and interior of the system was also demonstrated based on the time histories shown in Figures 18 and 19.

- 4) Learning the control policy in voltage-strain space was straight-forward and more accurate than approach of learning in temperature-strain space. The accuracy of the voltage measurements permits a reduction in required episodes since it allows for less error than the thermocouple measurements. This allows the agent to learn the policy in less time. By learning in voltage-strain space, a direct input-output mapping of strain to voltage is provided in less time than temperature-strain, and the user of the policy can skip the step of converting temperature to voltage.

CHAPTER IX

RECOMMENDATIONS

Recommendations and Potential extensions of this research include, but are not limited to:

- 1) Apply the control policy learned in voltage-strain space to a feedback control law that uses Shape Memory Alloy actuators. The simple input/output function that is produced by the agent would be useful for application to any control law that uses NiTi Shape Memory Alloy wire actuation.
- 2) Use alternative machine learning methods to learn the control policy and compare the accuracy and time to the Reinforcement Learning results. Although Reinforcement Learning has the benefit of learning without initial training data, it is a slow process. A useful alternative for comparison would be to generate training data experimentally and then use it to train an Artificial Neural Network.
- 3) Use simulation of the Shape Memory Alloy wire to learn the policy rather than learning with the experimental model. Learning in simulation using the hyperbolic tangent model can be accomplished much faster because the simulated response is immediate while an actual Shape Memory Alloy wire takes several seconds to reach steady-state. With an accurate hyperbolic tangent curve fit, the Reinforcement Learning agent can determine the optimal control policy quickly, and then the policy can be validated by testing on the experimental model.

- 4) Learn how to control a Shape Memory Alloy wire such that both number of actions and energy is minimized. By learning to control the system to a single point in voltage-strain space rather than the horizontal line corresponding to goal strain, the energy required for morphing could be minimized.
- 5) Adapt this Reinforcement Learning method to learning the control policy for arrays composed of Shape Memory Alloys. This would allow the development and use of actuators driven by Shape Memory Alloy wires, which has direct applicability to morphing aircraft. The dimension of the state-space can be increased to allow Reinforcement Learning to discover how to control the coupled motion of individual Shape Memory Alloy wire specimens for the achievement of a two-dimensional shape.
- 6) The ability to learn the control policy for Shape Memory Alloy materials of multi-dimensions, e.g. plates and solids as opposed to one-dimensional wires, is necessary for development of morphing structural elements. By learning to control morphing plates and solids, research can be opened to application of these structures.

REFERENCES

¹Texas Institute for Intelligent Bio-Nano Materials and Structures for Aerospace Vehicles Home Page. URL: <http://tiims.tamu.edu> [cited 15 May 2004].

²Agarwal, S., “Structural Morphing using Piezoelectric Modulation of Joint Friction,” *Journal of Intelligent Material Systems and Structures*, vol. 18, 2007, pp. 389-407.

³Barbarino, S., Ameduri, S., Lecce, L., and Concilio, A., “Wing Shape Control though an SMA-Based Device,” *Journal of Intelligent Material Systems and Structures*, vol. 20, no. 3, 2009, pp. 283-296.

⁴Kudva, J. N., “Overview of the DARPA Smart Wing Project,” *Journal of Intelligent Material Systems and Structures*, vol. 15, 2004, pp. 261-267.

⁵Yoo, I. K. and Desu, S. B., “Fatigue and Hysteresis Modeling of Ferroelectric Materials,” *Journal of Intelligent Material Systems and Structures*, vol. 4, 1993, pp. 490-495.

⁶Johnson, T., Frecker, M. I., Joo, J., Abdalla, M. M., Gurdal, Z., and Lindner, D. K., “Nonlinear Analysis and Optimization of Diamond Cell Morphing Wings,” *ASME-Publications-AD*, vol. 71, 2006, pp. 163-178.

⁷Bae, J-S, Kyong, N-H, Seigler, T. M., and Inman, D. J., “Aeroelastic Considerations on Shape Control of an Adaptive Wing,” *Journal of Intelligent Material Systems and Structures*, vol. 16, 2005, pp. 1051-1056.

⁸Matsuzaki, Y., "Recent Research on Adaptive Structures and Materials: Shape Memory Alloys and Aeroelastic Stability Prediction," *Journal of Intelligent Material Systems and Structures*, vol. 16, 2005, pp. 907-917.

⁹Strelc, J. K., Lagoudas, D. C., Khan, M. A., and Yen, J., "Design and Implementation of a Shape Memory Alloy Actuated Reconfigurable Airfoil," *Journal of Intelligent Material Systems and Structures*, vol. 14, 2003, pp. 257-273.

¹⁰Waram, T., *Actuator Design Using Shape Memory Alloys*. Hamilton, Ontario: T.C. Waram, 1993.

¹¹Sofla, A.Y.N., Elzey, D.M., and Wadley, H.N.G., "Two-way Antagonistic Shape Actuation Based on the One-way Shape Memory Effect," *Journal of Intelligent Material Systems and Structures*, vol. 19, 2008, pp. 1017-1027.

¹²Mavroidis, C., Pfeiffer, C. and Mosley, M., "Conventional Actuators, Shape Memory Alloys, and Electrorheological Fluids." *Invited Chapter in Automation, Miniature Robotics and Sensors for Non-Destructive Testing and Evaluation*, vol. 1, 1999, p. 10-21.

¹³Lagoudas, D., Mayes, J., Khan, M., "Simplified Shape Memory Alloy (SMA) Material Model for Vibration Isolation," *Smart Structures and Materials Conference*, Newport Beach, CA, 5-8 March 2001, pp. 452-461.

¹⁴Lagoudas, D. C., Bo, Z., and Qidwai, M. A., "A Unified Thermodynamic Constitutive Model for SMA and Finite Element Analysis of Active Metal Matrix Composites" *Mechanics of Composite Materials and Structures*, vol. 3, 1996, pp. 153-179.

¹⁵Bo, Z. and Lagoudas, D. C., "Thermomechanical Modeling of Polycrystalline SMAs Under Cyclic Loading, Part I-IV" *International Journal of Engineering Science*, vol. 37, 1999, pp. 1205-1249.

¹⁶Malovrh, B. and Gandhi, F., "Mechanism-Based Phenomenological Models for the Pseudoelastic Hysteresis Behavior of Shape Memory Alloys," *Journal of Intelligent Material Systems and Structures*, vol. 12, 2001, pp. 21-30.

¹⁷Patoor, E., Eberhardt, A., and Berveiller, M., "Potential pseudoelastic et plasticité de transformation martensitique dans les mono-et polycristaux métalliques." *Acta Metall*, vol. 35(12), 1987, pp. 2779.

¹⁸Falk, F., "Pseudoelastic Stress Strain Curves of Polycrystalline Shape Memory Alloys Calculated from Single Crystal Data" *International Journal of Engineering Science*, vol. 27, 1989, pp. 277.

¹⁹Banks, H., Kurdila, A. and Webb, G., "Modeling and Identification of Hysteresis in Active Material Actuators, Part (ii): Convergent Approximations," *Journal of Intelligent Material Systems and Structures*, vol. 8(6), 1997, pp. 536-550.

²⁰Webb, G., Kurdila, A. and Lagoudas, D., "Hysteresis Modeling of SMA Actuators for Control Applications," *Journal of Intelligent Material Systems and Structures*, vol. 9, no. 6, 1998, pp.432-447.

²¹Kirkpatrick, K. and Valasek, J., "Reinforcement Learning for Characterizing Hysteresis Behavior of Shape Memory Alloys," AIAA-2007-2932, *Proceedings of the AIAA Infotech@Aerospace Conference*, Rohnert Park, CA, 7-10 May 2007.

²²Haag, C., Tandale, M., Valasek, J., "Characterization of Shape Memory Alloy Behavior and Position Control Using Reinforcement Learning," AIAA-2005-7160, *Proceedings of the AIAA Infotech@Aerospace Conference*, Arlington, VA, 26-29 September 2005.

²³Valasek, J., Tandale, M., and Rong, J., "A Reinforcement Learning - Adaptive Control Architecture for Morphing," *Journal of Aerospace Computing, Information, and Communication*, Vol. 2, No.5, 2005, pp. 174-195.

²⁴Valasek, J., Doeblner, J., Tandale, M. D., and Meade, A. J., "Improved Adaptive-Reinforcement Learning Control for Morphing Unmanned Air Vehicles," *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, vol. 38(4), 2008, pp. 1014-1020.

²⁵Sun, R. and Sessions, C., "Learning Plans without a priori Knowledge," *Adaptive Behavior*, vol. 8, 2000, pp. 225-253.

²⁶Sutton, R. and Barto, A., *Reinforcement Learning: An Introduction*. Cambridge, Massachusetts: The MIT Press, 1998.

²⁷Konidaris, G. D. and Hayes, G.M., "An Architecture for Behavior-Based Reinforcement Learning," *Adaptive Behavior*, vol. 13, 2005, pp. 5-32.

²⁸Varshavskaya, P., Kaelbling, L. P., and Rus, D., "Automated Design of Adaptive Controllers for Modular Robots using Reinforcement Learning," *The International Journal of Robotics Research*, vol. 27, 2008, pp. 505-526.

²⁹Whiteson, S., Taylor, M. E., and Stone, P., "Empirical Studies in Action Selection with Reinforcement Learning," *Adaptive Behavior*, vol. 15, 2007, pp. 33-50.

³⁰Bhatnagar, S. and Abdulla, M. S., "Simulation-Based Optimization Algorithms for Finite-Horizon Markov Decision Processes" *SIMULATION*, vol. 84, 2008, pp. 577-600.

³¹Santamaria, J. C., Sutton, R. S., and Ram, A., "Experiments with Reinforcement Learning in Problems with Continuous State and Action Spaces," *Adaptive Behavior*, vol. 6, 1997, pp. 163-217.

³²Mitchell, T. M., *Machine Learning*. Singapore: The McGraw-Hill Companies, Inc., 1997, pp. 231-232.

³³Russell, S. and Norvig, P., *Artificial Intelligence: A Modern Approach*. Upper Saddle River, New Jersey: Pearson Education, Inc., 2003, p. 733.

³⁴Khan, M. M., Lagoudas, D. C., Mayes, J. J., and Henderson, B. K., "Pseudoelastic SMA Spring Elements for Passive Vibration Isolation: Part I – Modeling," *Journal of Intelligent Material Systems and Structures*, vol. 15, 2004, pp. 415-441.

³⁵Gorbet, R.B., and Morris, K. A., "Closed-Loop Position Control of Preisach Hystereses," *Journal of Intelligent Material Systems and Structures*, vol. 14, 2003, pp. 483-495.

³⁶Han, Y-M, Choi, S-B, and Werely, N. M., "Hysteretic Behavior of Magnetorheological Fluid and Identification Using Preisach Model," *Journal of Intelligent Material Systems and Structures*, vol. 18, 2007, pp. 973-981.

³⁷Lagoudas, D.C., and Bhattacharyya, A., "On the Correspondence Between Micromechanical Models for Isothermal Pseudoelastic Response of Shape Memory

Alloys and the Preisach Model of Hysteresis,” *Mathematics and Mechanics of Solids*, vol. 2, 1997, pp. 405-440.

³⁸Ikuta, K., Tsukamoto, M., and Hirose, S., “Mathematical Model and Experimental Verification of Shape Memory Alloy for Designing Micro Actuator,” *Proceedings of the IEEE Micro Electro Mechanical Systems Conference*, Nara, Japan, 30 January – 2 February 1991, pp. 103-108.

VITA

Name: Kenton Conrad Kirkpatrick

Address: Texas A&M University
Department of Aerospace Engineering
3141 TAMU
College Station, TX 77843-3141

Email Address: kentonkirk@gmail.com

Education: B.S., Aerospace Engineering, Texas A&M University, 2007
M.S., Aerospace Engineering, Texas A&M University, 2009