Reinforcement Learning for Active Length Control of Shape Memory Alloys

Kenton Kirkpatrick* and John Valasek†
Texas A&M University, College Station, Texas 77843-3141

The ability to actively control the shape of aerospace structures has initiated research regarding the use of Shape Memory Alloy actuators. These 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. The ability to characterize this temperature-strain relationship using Reinforcement Learning has been previously accomplished, but in order to control Shape Memory Alloy wires it is more beneficial to learn the voltage-position relationship. Numerical simulation using Reinforcement Learning has been used for determining the temperature-strain relationship for characterizing the major and minor hysteresis loops, and determining a limited control policy relating applied temperature to desired strain. Since Reinforcement Learning creates a non-parametric control policy, and there is not currently a general parametric model for this control policy, determining the voltage-position relationship for a Shape Memory Alloy is done separately. This paper extends earlier numerical simulation results and experimental results in temperature-strain space by applying a similar Reinforcement Learning algorithm to voltage-position space using an experimental hardware apparatus. Results presented in the paper show the ability to converge on a near-optimal control policy for Shape Memory Alloy length control by means of an improved Reinforcement Learning algorithm. These results demonstrate the power of Reinforcement Learning as a method of constructing a policy capable of controlling Shape Memory Alloy wire length.

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. The main focus of the Texas Institute for Intelligent Bio-Nano Materials and Structures for Aerospace Vehicles is to revolutionize aircraft and space systems 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

*Graduate Research Assistant, Vehicle Systems & Control Laboratory, Aerospace Engineering Department. Student Member AIAA. kentonkirk@gmail.com

†Associate Professor and Director, Vehicle Systems & Control Laboratory, Aerospace Engineering Department. Associate Fellow AIAA. valasek@tamu.edu. Website: http://jungfrau.tamu.edu/valasek
idea of using Shape Memory Alloys (SMA) as actuators to drive the shape change of a wing. 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. In this research, NiTi SMAs were used.

SMAs have a unique ability known as the Shape Memory Effect (SME). This material can be put under a stress that leads to a plastic deformation and then fully recover to its original shape after heating it to a high temperature. This makes SMAs ideal for structures that undergo large amounts of stress, such as aircraft. 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 shape. This original shape is retained 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. The SMA wire exhibits a hysteresis behavior in its relationship between temperature and strain due to non-uniformity in the phase transformations. 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 martensitic start, $M_f$ is martensitic finish, $A_s$ is austenitic start, and $A_f$ is austenitic finish.

![Thermally Induced Phase Transformations](image-url)
The most common method of affecting temperature in a SMA 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, as shown by Equation (1).

$$\rho c V \frac{dT}{dt} = \frac{V^2(t)}{R} - hA(T(t) - T_\infty)$$  \hspace{1cm} (1)

In Equation (1), $\rho$ is the wire density, $c$ is the specific heat of the wire, $V$ is the wire volume, $T$ is the wire temperature, $V$ is the voltage difference in the wire, $t$ is time, $R$ is the wire electrical resistance, $h$ is the convective heat transfer coefficient, $A$ is the wire surface area, and $T_\infty$ is the ambient temperature of the coolant surrounding the wire. With Equation (1), 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 because the specific heat and convective coefficient change dynamically during crystal phase transformation. Therefore, it is a simpler process 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.\(^4\) This is a time and labor intensive process that requires external supervision and does not actively discover the hysteresis in real-time. Other methods that characterize this behavior are phenomenological models,\(^5,6\) micromechanical models,\(^7,8\) and empirical models based on system identification.\(^9,10\) 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 major drawback to using any of these methods is that the minor hysteresis loops that correspond to a SMA that is not fully actuated are unattainable and must be determined through mathematical models. A simulated model of the major and minor hysteresis loops for an SMA wire is shown in Figure 2.

![Fig. 1b Temperature-Strain Hysteresis](#)
To achieve mapping this SMA hysteresis autonomously and in real-time, previous research was done to characterize SMAs in temperature-strain space using the approach of an artificial intelligence algorithm known as Reinforcement Learning (RL). This 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 mapping 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 ideal for the morphing technology because it allows for a machine to learn the optimal control 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. Reinforcement Learning is the driving force of SMA hysteresis learning that can be further implemented to achieve SMA shape control. 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 parametric models of this relationship, the model needed to achieve characterizing the SMA morphing in voltage-position space 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.

In a previous, similar work, the development of an RL algorithm that can actively learn the hysteresis behavior of a SMA wire in temperature-strain space was followed. RL was used to determine the major and minor hysteresis behavior in a SMA wire, and the algorithm was validated using an experimental hardware apparatus for the training, testing, and experimentation of specimen SMA wires.
This paper is organized as follows. In Section 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 the function approximations are all discussed. Section III explains how the experimental apparatus was constructed for testing this method, and it includes details about the problems that arose in deciding how to choose a proper coolant. This is followed by Section IV, which explains how the RL agent connects to the experimental apparatus and is able to interact with the SMA wire in real-time. The next section, Section V, provides a detailed explanation of the characterization of SMA hysteresis behavior in temperature/strain space. Both simulation and experimentation results are discussed. Section VI provides the results of this research and demonstrates the ability of RL to learn how to control a SMA wire. This is then followed by conclusions and future work in Section VII.

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. RL uses 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 algorithms of RL are Dynamic Programming, Monte Carlo, and Temporal Difference.\textsuperscript{15} 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. 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. 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.\textsuperscript{15} The action value function used to update this control policy is:

\[ Q_k(s,a) = Q_k(s,a) + \alpha \delta_k \] \hspace{1cm} (2)

where \( s \) is the current state, \( a \) is the current action, \( Q \) is the control policy, and the \( k \) term signifies the current policy. The \( \alpha \) term is a parameter that is used to “punish” the RL algorithm when it repeats itself within each episode. The term \( \delta_k \) is defined as:

\[ \delta_k = r_{k+1}(s',a') + \gamma Q_{k+1}(s',a') - Q_k(s,a) \] \hspace{1cm} (3)

The term \( s' \) refers to the future state, \( a' \) is the future action, \( k+1 \) corresponds to the future policy, and \( \gamma \) 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 action value function:

\[ Q_k(s,a) = Q_k(s,a) + \alpha [r_{k+1}(s',a') + \gamma Q_{k+1}(s',a') - Q_k(s,a)] \] \hspace{1cm} (4)
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 range. 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:

\[
Q_k(s, a, g) = Q_k(s, a, g) + \alpha [r_{k+1}(s', a', g) + \gamma Q_{k+1}(s', a', g) - Q_k(s, a, g)]
\]

(5)

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 (5). This algorithm is outlined as follows:

**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., \( \varepsilon \)-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., \( \varepsilon \)-Greedy)
    - \( Q(s, a, g) \leftarrow Q(s, a, g) + \alpha [r + \gamma Q(s', a', g) - Q(s, a, g)] \)
    - \( s \leftarrow s', a \leftarrow a' \)
  - 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 known 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 \( \varepsilon \)-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. This is because the RL 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 ε = 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 β ≥ 1 - ε
    - a ← random
  - If β < 1 - ε
    - a ← RL control policy exploitation

In order 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 a 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. 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:

- Episode 1 – Episode 29
  - ε = 0.7
- Episode 30 – Episode 59
  - ε = 0.6
- Episode 60 – Episode 79
  - ε = 0.5
- Episode 80 – Episode 99
  - ε = 0.3
- Episode 100 – Episode 139
  - ε = 0.2
In this research, the states are defined by the current strain, while the actions are defined by the desired voltage that is applied to the SMA wire. Conversion between strain and position is a trivial process, so strain was kept as the state choice so that it can be global to specimens of different length. The goal that the system is attempting to reach is the desired strain of the SMA wire. The purpose of the RL agent is to converge on the optimal voltage needed to produce the desired strain based on the current strain in the wire.

The RL method described up to this point will solve a problem where discreet values of strain states are needed, but the SMA wire used in this experimentation is a physical specimen that has a continuous state-space. Its action-value function \( Q(s,a) \) and action-preference function \( p_a(s) \) are both defined on continuous domains. One approach to solving this type of Reinforcement Learning problem with a continuous state-space is function approximation methods. In this instance, the approximate action-value function \( Q(s,a) \) is represented using a parameterized functional form with parameter vector \( \theta_v \):

\[
Q(s,a) = \sum_{j=1}^{N} \theta_y \Phi_j(s), \quad \forall s \in s
\]

where \( \Phi_j(s) \) is predetermined and satisfies

\[
\Phi_i(s) \cdot \Phi_j(s) = \begin{cases} 
1 & \text{if } i = j \\
0 & \text{if } i \neq j 
\end{cases}
\]

Thus the basis vector

\[
H = \begin{bmatrix} \Phi_1(s) & \Phi_2(s) & \ldots & \Phi_N(s) \end{bmatrix}^T
\]

is an orthogonal vector that satisfies

\[
HH^T = I_{N \times N}
\]

Using Equation (8), Equation (6) becomes

\[
Q(s,a) = H^T \theta_v, \quad \forall s \in s
\]

As \( H^T \) is fixed, \( Q(s,a) \) depends totally on \( \theta_v \), and varies from time step to time step only when \( \theta_v \) changes. Updating \( \theta_v \) is related to the Temporal-Difference error \( \delta_i \), as defined in Equation (3). The process of deriving the updating formula for \( \theta_v \) is shown as follows: First, we selected the gradient descent method for updating \( Q(s,a) \), upon which Equation (2) becomes

\[
Q(s_t,a) \leftarrow Q(s_t,a) + \alpha \delta_i
\]

Using Equation (10), Equation (12) becomes

\[
H_i^T \theta_v \leftarrow H_i^T \theta_v + \alpha \delta_i
\]

where \( H_i = [\Phi_1(s_t) \quad \Phi_2(s_t) \quad \ldots \quad \Phi_N(s_t)] \), and \( H_i \) is the known coefficient matrix. Using Equation (9), the formula for updating \( \theta_v \) becomes
\[ \theta_t \leftarrow \theta_t + \alpha \delta_t H_t \]

Note that the purpose of the Reinforcement Learning module is not to compute the exact \( Q(s,a) \), but to learn the optimal policy with the optimal action-value function. In the process of policy iteration, an intermediate policy can be improved to a better policy even if its action-value function \( Q(s,a) \) is not exactly computed. Similarly, the preference function for each action is represented using a parameterized functional form with parameter vector \( \theta_{pa} \).

\[ p_a(s) = H^T \theta_{pa} \]

Since \( Q(s,a) \) and \( p_a(s) \) are both defined on the same state-space, the basis vector \( H \) can also be used for \( Q(s,a) \). Following a similar process from Eq. 25 to Eq. 27, the formula for updating \( \theta_{pa} \) is

\[ \theta_{pa} \leftarrow \theta_{pa} + \alpha \beta \delta H^T \]

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.

### III. Experimental Apparatus

For the SMA wire to be tested, a physical experimental setup 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, only position and voltage are needed, but since the actuation of the SMA wire is temperature-based, the experimental setup includes temperature measurements for the sake of reference to make sure that the wire follows the hysteresis path correctly. 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 3 shows the complete 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 4 shows the complete experimental hardware setup.

Antifreeze was used as the coolant in this experiment because it was concluded in the previous characterization phase of research 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 strain-temperature space using direct user input for a water-filled apparatus is shown in Figure 5.

![SMA Major Hysteresis](image)

Fig. 5 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 6.
In Figure 6, the experimental results are compared to the mathematical 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 (16) and (17):

\[
M_l = H / 2 \tanh\left(\left(T - ct_i\right)a + s\left(T - (ct_i + ct_r) / 2\right) + H / 2 + cs\right)
\]

\[
M_r = H / 2 \tanh\left(\left(T - ct_r\right)a + s\left(T - (ct_i + ct_r) / 2\right) + H / 2 + cs\right)
\]

In these equations, \(H\), \(ct_i\), \(a\), \(s\), \(ct_r\), and \(cs\) are constants that determine the shape of the hyperbolic tangent model. \(M_l\) and \(M_r\) 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.

**IV. Hardware/Software Interface**

In order 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.
V. Temperature-Strain Learning

This experiment has been tested 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 that corresponds to complete actuation. In order 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 state 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 9 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.
Figure 9 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 10 shows that the major hysteresis behavior is experimentally attainable from Reinforcement Learning.

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. Figure 12 shows 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 9.

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 state. 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 12 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 12.

![Minor Hysteresis Loops](image)

**Fig. 12 Minor Hysteresis Loops Produced During Learning Episodes**

Figure 12 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-position 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.

### VI. Results

The control policy developed for this SMA specimen 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 ±0.5%, the error range allowed was ±0.65mm.

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 ±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 13.

![Figure 13 Policy Test for Goal = 2.7%](image)

Figure 13 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 13, initial voltages were applied at time $t = 0$ seconds so that the control policy could be tested at several different initial strains. The initial strains chosen for testing here were 0.1%, 3.2%, 1.2%, and 2.7%. The actual exploitation of the control policy began at $t = 21$ seconds in each case, and 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 13 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 14 shows the results of testing the control policy for a goal of 0.1% strain.

Like the previous case, initial voltages were applied at time $t = 0$ seconds, and the control policy exploitation began at $t = 21$ seconds. 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 14 were 0.1%, 3.2%, 1.5%, and 2.7%, which are nearly identical to the initial strains chosen in Figure 13. 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 is the other system boundary. The other strains were chosen because they nearly match the initial strains used in the previous test. Figure 14 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.

![Fig. 14 Policy Test for Goal = 0.1%](image-url)
VII. Conclusions and Future Research

This paper used the Sarsa Reinforcement Learning algorithm to experimentally develop a voltage/strain space active length control policy for a NiTi Shape Memory Alloy wire. The results established the ability to learn a control policy in an online experiment without human external supervision, and validated the approach experimentally. The feasibility of achieving goals on both the boundary and interior of the system was also demonstrated. The approach used here of the agent learning the control policy in voltage/strain space was easier and more accurate than the previous approach of learning in the temperature/strain space because accurate voltage measurements replace the inaccurate thermocouple measurements of the latter. The control policy in voltage/strain space is also directly useful for use in a feedback control law.

The results presented here encourage extension of this research toward practical shape change control of Shape Memory Alloy devices. The ability to learn the control policy for Shape Memory Alloy materials of multi-dimensions, e.g. plates and solids as opposed to single wires, is required. Another investigation is Shape Memory Alloy actuators, as well as the application of this Reinforcement Learning method to learning the control policy for arrays composed of Shape Memory Alloys. This latter topic has direct correlation to the experiment presented and discussed here, and will be the focus of the next phase of this research.

Acknowledgment

This work was sponsored (in part) by the National Science Foundation Graduate Research Fellowship Program and the Air Force Office of Scientific Research, USAF, under grant/contract number FA9550-08-1-0038. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the National Science Foundation, Air Force Office of Scientific Research, or the U.S. Government.

VIII. References


