JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 6, March 2009

Reinforcement Learning for Characterizing Hysteresis Behavior of Shape Memory Alloys

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

DOI: 10.2514/1.36217

The ability to actively control the shape of aerospace structures has spawned the use of shape memory alloy actuators. These actuators can be used for morphing or shape control by modulating their temperature, which is effectively done by applying a voltage difference across their length. Characterization of this temperature–strain relationship is currently done using constitutive models, which is time and labor intensive. Shape memory alloys also contain both major and minor hysteresis loops. Understanding the hysteresis is crucial for practical applications, and characterization of the minor hysteresis loops, which map the behavior of a wire that is not fully actuated, is not possible using the constitutive method. Numerical simulation using reinforcement learning has been used for determining the temperature–strain relationship and characterizing the major and minor hysteresis loops, and determining a control policy relating applied voltage to desired strain. This paper extends and improves upon the numerical simulation results, using an experimental hardware apparatus and improved reinforcement learning algorithms. Results presented in the paper verify the numerical simulation results for determining the temperature–strain major hysteresis loop behavior, and also determine the relationships of the minor hysteresis loops.

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 biological 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 to optimize 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 (SMA) as actuators to drive the shape change of a wing. The most commonly used SMAs are composed of either nickel and titanium, or the combination of nickel, titanium, and copper. The benefits of using each of these alloys have been explored in the present work.

SMAs have a unique ability known as the shape memory effect (SME) [2]. 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 [3]. SMAs begin in

Received 14 December 2007; revision received 14 October 2008; accepted for publication 24 November 2008.

* Undergraduate Research Assistant, Flight Simulation Laboratory, Aerospace Engineering Department. Student Member AIAA. k-man@neo.tamu.edu

† Professor and Director, Flight Simulation Laboratory, Aerospace Engineering Department. Associate Fellow AIAA. valasek@tamu.edu. Website: http://jungfrau.tamu.edu/valasek

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. Morphing wings will undergo considerable stresses in flight, so the actuators' ability to recover from plastic deformation while simultaneously accomplishing wing morphing during the crystal phase change makes the characterization and control of SMAs valuable. This desired ability to both characterize and control SMAs through a crystal phase transformation is the goal of the present work.

When a SMA wire has a phase transformation, it changes its length. The phase transformation from twinned martensite to austenite causes a decrease in length while the reverse process extends it back to its original length. Control of this transformation is needed for morphing actuation to be possible. The SMA wire exhibits a hysteresis behavior in its relationship between temperature and strain owing to non-uniformity in the phase transformations [3]. This occurs because the phase transformation from martensite to austenite begins and ends at different temperatures than the reverse process. Figure 1 demonstrates this behavior, where M_s is martensitic start, M_f is martensitic finish, A_s is austenitic start, and A_f is austenitic finish. The top section of the figure shows the cooling stage from austenite back to martensite, while the bottom shows the reverse process. M_s and M_f are both shown to the left of A_s and A_f because the martensitic transformation temperatures are actually lower than the austenitic transformation temperatures. This property causes the hysteresis in the temperature–strain curve.

The most common method of affecting temperature in a SMA to induce actuation is the use of resistive heating. 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 owing to electrical current. This can be modeled by a differential equation based on these parameters, as shown by Eq. (1).

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

In Eq. (1), ρ is the wire density, c is the specific heat of the wire, V_w is the wire volume, T is the wire temperature, V is the voltage difference in the wire, R is the wire electrical resistance, h is the convective heat transfer coefficient, A is the wire surface area, and T_∞ is the ambient temperature of the coolant surrounding the wire.

This hysteresis behavior 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

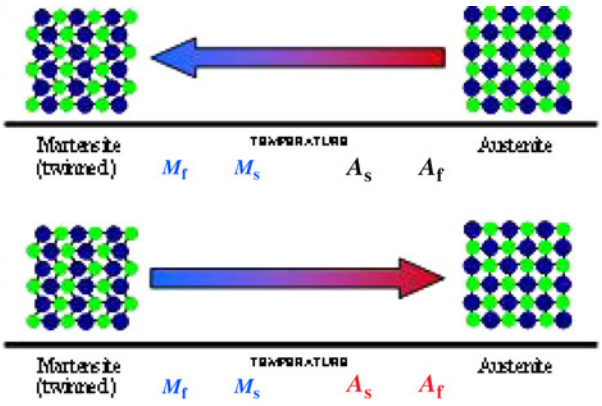

Fig. 1 Thermally induced phase transformations.

determined through mathematical models. These minor hysteresis loops correspond to the path taken by the SMA when it changes direction before becoming fully actuated or fully unactuated. A simulated model of the major and minor hysteresis loops for an SMA wire is shown in Fig. 2. In this figure, the red border shows the major hysteresis boundary while the blue curves represent the path taken during the simulation. The curves that cross into the interior of the major loop are tracing minor hysteresis loops.

The SMA phase transformation is not a thermodynamically reversible process, so there is uncertainty in the model owing to the highly nonlinear behavior of the SMA. To map SMA hysteresis autonomously and in real-time, we use a reinforcement learning (RL) approach. In the RL approach a learning "agent" repeatedly interacts with the system to discover the optimal sequence of actions which lead to a predetermined goal. This optimal sequence or path is learned by providing the agent with a process of rewards and consequences that allow the program to remember which actions are good for achieving specific states, and which are poor. By mapping this behavior in real time, both major and minor hysteresis loops can be experimentally determined. This model is unknown and can be determined by the RL in conjunction with the experimental setup. The control policy is initially unknown so RL is exploited because it does not require a predefined control policy.

Reinforcement learning (RL) is a promising approach for SMA hysteresis learning that can eventually lead to practical SMA shape control, as has been discussed in multiple works involving other aspects of morphing aircraft research [11–15]. This paper develops and evaluates an RL algorithm that can actively learn the hysteresis behavior in a SMA wire. RL is used to determine the major and minor hysteresis behavior in an SMA wire, and the algorithm is validated using an experimental hardware apparatus for the training, testing, and experimentation of specimen SMA wires. Details of the hardware/software interface for real-time experimentation are provided, and results are verified by comparison to constitutive and mathematical models.

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 work, 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 [16]. 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 work uses 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. In this work, an episode is defined to be the process between beginning a new goal at some initial strain and achieving that goal. Sarsa updates the control policy

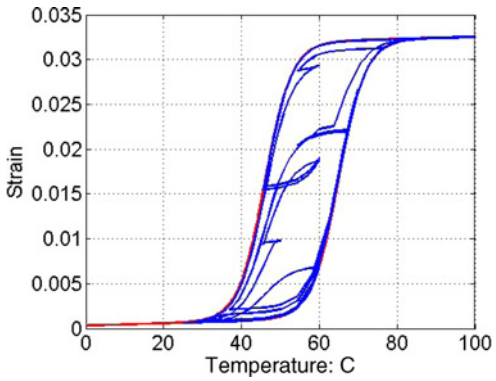

Fig. 2 Results of SMA hysteresis simulation.

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 [16]. The action value function used to update this control policy is

$$Q_k(s, a) = Q_k(s, a) + \alpha\delta_k \tag{2}$$

where s is the current state, a is the current action, Q is the action value function (which becomes the control policy after the final time step), and the k term signifies the current step. The α term is a parameter that is used to keep the RL from repeating itself within each episode. The term δ_k is defined as

$$\delta_k = r_{k+1}(s', a') + \gamma Q_{k+1}(s', a') - Q_k(s, a) \tag{3}$$

The term s' refers to the future state, a' is the future action, $k + 1$ corresponds to the next time step, and γ is the constant that is used to affect the rate of convergence by weighting the future policy. Equations (2) and (3) can be combined to form the detailed action value function

$$Q_k(s, a) = Q_k(s, a) + \alpha[r_{k+1}(s', a') + \gamma Q_{k+1}(s', a') - Q_k(s, a)] \tag{4}$$

The reward given for each state/action pair is defined by r. The reward that is given for each situation is a user-defined parameter. For this work, when a goal state is achieved, a reward of 1 is given, 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. As mentioned above, the control policy was modified to a three-dimensional matrix that includes the goal as the third dimension. With g representing the goal state, the action value function now becomes

$$Q_k(s, a, g) = Q_k(s, a, g) + \alpha[r_{k+1}(s', a', g) + \gamma Q_{k+1}(s', a', g) - Q_k(s, a, g)] \tag{5}$$

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 Eq. (5). This algorithm is outlined as follows [16]

Sarsa method:

1) Initialize $Q(s, a, g)$ arbitrarily
2) Repeat for each g:
   i) 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)
        - $Q(s, a, g) \leftarrow Q(s, a, g) + \alpha[r + \gamma Q(s', a', g) - Q(s, a, g)]$
        - $s \leftarrow s', a \leftarrow a'$
      - 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 to 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 information 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 work, which means that for some percentage of the time that an action is chosen, the RL will choose to randomly explore rather than choose the action that the control policy 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 $\epsilon = 0$. The ϵ-Greedy action-value method can be implemented by the following algorithm:

ϵ-Greedy action-value method

Repeat for each action value:

i) Choose ϵ between 0 and 1

ii) Generate random value β between 0 and 1

iii) If $\beta \geqslant 1 - \epsilon$

- $a \leftarrow$ random

iv) If $a < 1 - \epsilon$

- $a \leftarrow a^*$ (Action that maximizes $Q(s, a, g)$)

To converge on the optimal control policy in the shortest amount of time, this work used a progressively 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 progressive ϵ-Greedy method, a simple algorithm was constructed to determine 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 were determined during the simulation phase by trial and error to find the fastest convergence rates, and they are as follows:

1) Episode 1–Episode 29
   - $\epsilon = 0.7$
2) Episode 30–Episode 59
   - $\epsilon = 0.6$
3) Episode 60–Episode 79
   - $\epsilon = 0.5$
4) Episode 80–Episode 99
   - $\epsilon = 0.3$
5) Episode 100–Episode 139
   - $\epsilon = 0.2$
6) Episode 140+
   - $\epsilon = 0.05$

In this work, the states are defined by the current strain and temperature, while the actions are defined by the desired temperature. The desired temperature is immediately converted to voltage that is applied to the SMA wire. 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 temperature needed to produce the desired strain based on the current strain in the wire and the current temperature of the wire.

The RL method described up to this point will solve a problem where discrete values of strain states and temperature states are needed, but the SMA wire used in this experimentation is a physical specimen that has a continuous state-space. One approach to solving this type of reinforcement learning problem with a continuous state-space is function approximation methods. The function approximation method used here is the k-nearest neighbor algorithm. This algorithm chooses the value associated with the average of the k-nearest values to the current state. More precisely, this work exploited a 1-nearest neighbor approach. This means that whenever a state lies between two discrete states, the action is chosen by the value of the state that is closest to it. As a result of using this method, the action-preference function that was developed in generality above, $p_{\mathrm{a}}(s)$, becomes simplified to a step function. Between every discrete point in the state space a step function determines which value will be assigned. This is a simple but highly effective method of approximating the discrete action-value function over continuous state space.

The characterization of the hysteresis behavior can be accomplished in two different ways by using RL. During the exploration phase, the paths that are followed plot the major and minor hysteresis behavior while real-time data are recorded. This is what was used for characterization here. Once the RL learns the optimal temperature required to

achieve each goal strain from each initial strain, it can then be used to map the hysteresis behavior of the SMA wire in real-time. By allowing RL to run through each of its learned situations and recording the strain and temperature data at each time interval, the characterized hysteresis loop can be easily plotted to graphically show that it has learned the SMA phase transformation strain/temperature behavior. This method requires that the policy be learned for every goal state in the discrete state-space. Because this paper only shows learning of limited goals, it is not demonstrated here. RL is powerful in this context because it can characterize the hysteresis in real-time while learning, develops a functional control policy, and can be used after learning to characterize the hysteresis as well.

III. Experimental Apparatus

For the SMA wire to be tested, a physical experimental setup was first constructed. The SMA wire is contained within an apparatus that is constructed of Plexiglas and aluminum supports. The apparatus is sealed so that no coolant can leak out as the experimentation is proceeding. The wire was originally attached to the walls by Kevlar chords and is set in series with a spring with constant $k = 4.34$ N/mm. It was later determined during experimentation that using a dead weight providing a constant tensile stress of 105-MPa in the initial martensitic state of zero strain was more practical, so the spring was replaced. In this context, zero strain is defined to be the reference point where the wire is under constant tensile stress from the weight but is at room temperature where the crystal phase is completely martensite.

A thermocouple is connected to the wire, which measures the temperature of the wire and sends small voltages to the data acquisition (DAQ) board. A linear voltage differential transducer (LVDT) 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. This is a position sensor that is used in this experiment to determine the tensile strain by measuring the change in length of the SMA wire. The LVDT sends a voltage to the DAQ board which changes depending on the position of the probe. A variable voltage supply is used to provide a voltage difference across the SMA wire for heating it and is connected to the SMA wire via alligator clips. 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 0.00 V−2.50 V. Figures 3 and 4 show 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 pool contains both heating and cooling coils that allow it to keep the

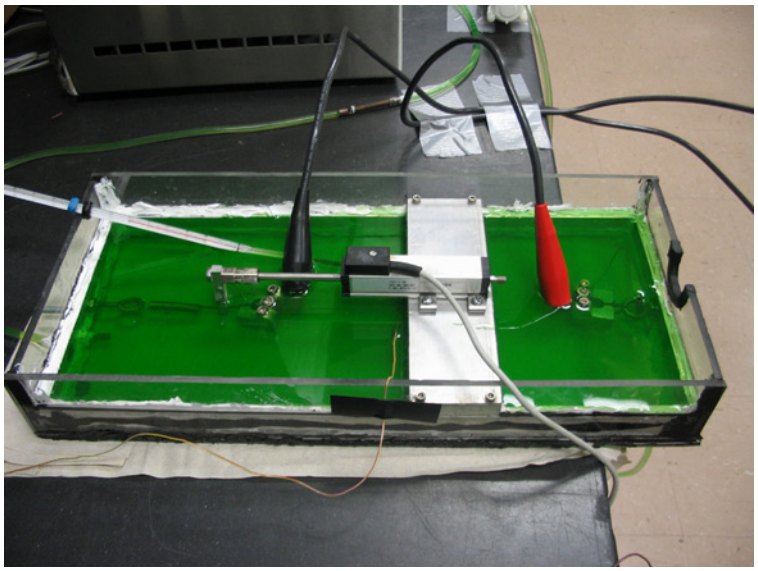

Fig. 3 Experimental apparatus.

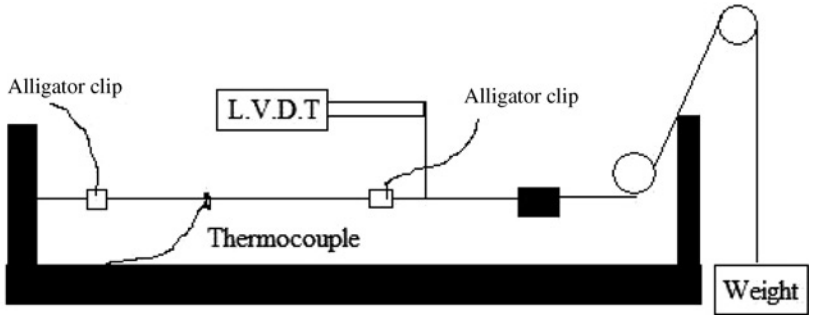

Fig. 4 Diagram of experimental apparatus.

antifreeze at a specified ambient temperature. In this work, 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.

The setup of this hardware led to many technical issues, some of which revealed key conclusions about the ability to characterize a SMA wire using this real-time method. The coolant originally used for decreasing the time required to achieve martensite was water. Water was 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, this work has revealed that water was not an ideal coolant for this particular case. Water transfers heat too easily, leading to poor temperature measurements by the thermocouple. The thermocouple does not perfectly touch the SMA wire owing to electrical problems, so poor readings occur because 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 temperature measurements at high temperatures become highly inaccurate and useless for application in this experiment. The water also causes some current loss owing to impurities in the water so high voltages (10–12 V) are required to achieve high temperatures. The characterization of the major hysteresis loop using forced voltage inputs for a water-filled apparatus is shown in Fig. 5.

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 phase transformations, although it does cause a slower phase transformation back to martensite. Antifreeze also has the ability to greatly exceed the previous limit of 100 °C without boiling, thereby eliminating the boiling effects caused by water at high temperatures and allowing for better measurements. Antifreeze has low conductivity, and by using antifreeze, full actuation can occur with 2.5 V instead of the 12 V required in water. The characterization of the major

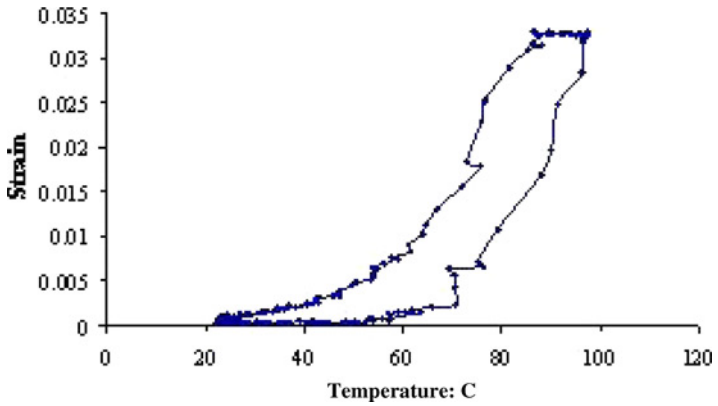

Fig. 5 Major hysteresis in water for NiTi SMA.

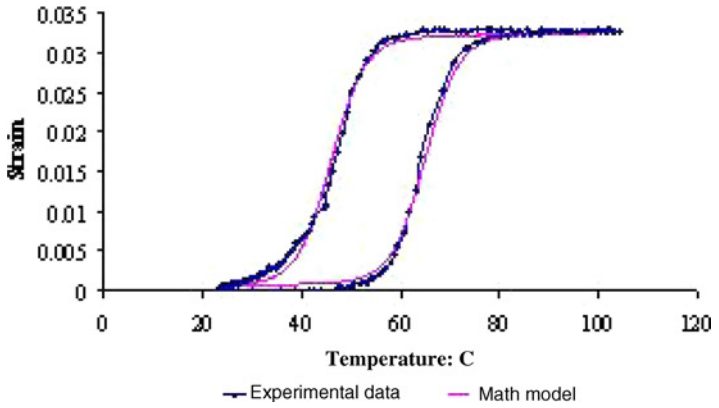

Fig. 6 Major hysteresis in antifreeze for NiTi SMA.

hysteresis behavior using forced voltage inputs in an antifreeze-filled apparatus is shown in Fig. 6. This was done to form a baseline to compare the hysteresis results from RL to what is known to be physically accurate.

In Fig. 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 Eqs (6) and (7)

$$M_{\mathrm{l}} = H/2 \tanh((T - ct_{\mathrm{l}})a) + s(T - (ct_{\mathrm{l}} + ct_{\mathrm{r}})/2) + H/2 + cs \tag{6}$$

$$M_{\mathrm{r}} = H/2 \tanh((T - ct_{\mathrm{r}})a) + s(T - (ct_{\mathrm{l}} + ct_{\mathrm{r}})/2) + H/2 + cs \tag{7}$$

In these equations, H, ct_{r}, a, s, ct_{l}, and cs 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.

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 LVDT 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 was determined by either exploration or exploitation, depending on the ϵ-Greedy choice. 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 hysteresis of a real, physical SMA wire in an experimental setup as diagrammed in Fig. 7.

V. Results

This work initially used a CuNiTi wire for testing, which has the favorable property of taking much more stress to fail than a NiTi SMA wire allows. However, this work has uncovered issues with using this type of SMA because of poor hysteresis characterization over a period of a few episodes. Figure 8 shows a plot of the hysteresis behavior of a CuNiTi wire as obtained over a course of three episodes.

As can be seen in Fig. 8, the hysteresis behavior does not appear nearly as clearly as it does with the NiTi wire shown in Fig. 6. Owing to this fact, the CuNiTi wire was replaced by a NiTi wire for the remainder of the tests. By using the NiTi wire, the lower tensile strength became a problem. The spring that was used to keep the wire under constant stress was replaced with a dead weight providing a tensile stress of 105 MPa. The dead weight is a superior method of providing stress in this case because it provides a constant stress that does not increase with SMA strain. Owing to the use of dead weight, the NiTi wire does not break as easily as before, allowing data to be recorded using the same sample for a larger period of time.

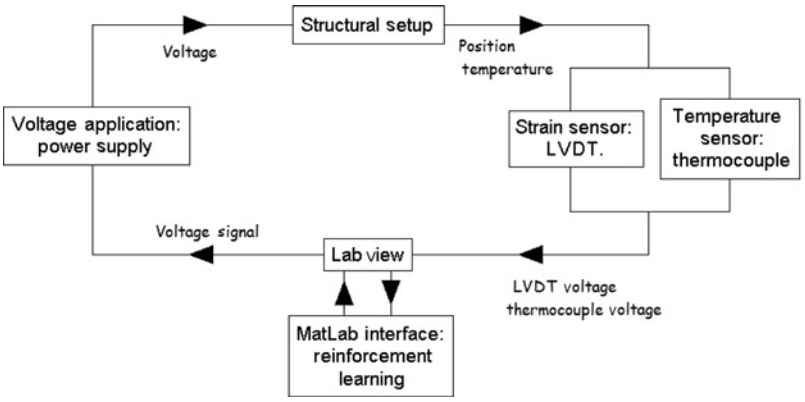

Fig. 7 Hardware/software connectivity and interfaces of the experimental apparatus.

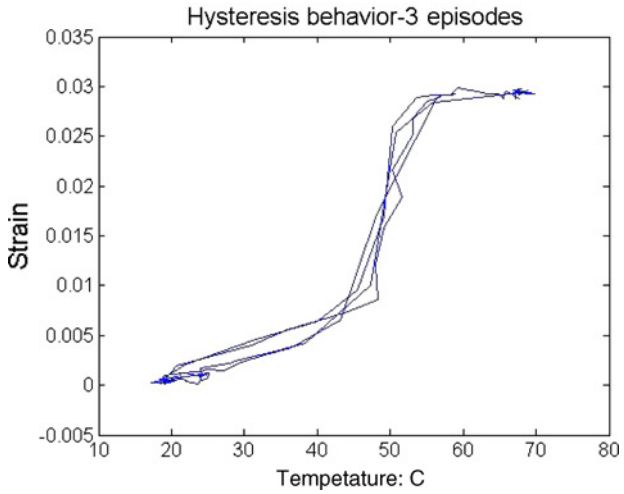

Fig. 8 Behavior of CuNiTi SMA.

This experiment has been tested over many episodes at several different goal states corresponding to individual strain states, where the end of 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. 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 forces 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 because 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 to reach a goal of 2.7% strain. Every episode presented in this data begins at a fully unactuated strain of 0%. As this graph shows, the RL algorithm takes fewer actions per episode 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 because the RL

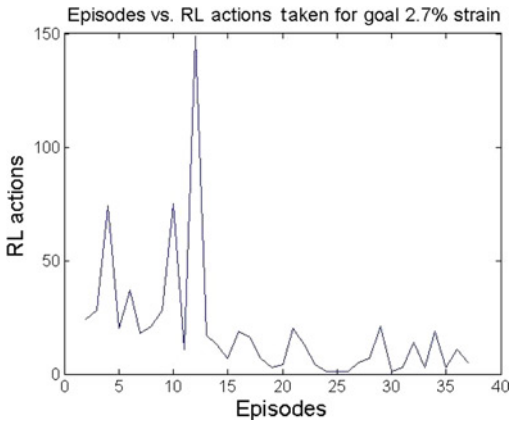

Fig. 9 Episodes vs. actions.

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.

Over the course of 37 episodes to a goal state of 2.7% strain, the major hysteresis behavior becomes visible. Figure 10 shows that the major hysteresis behavior is experimentally attainable from reinforcement learning. The blue lines indicate the path that was taken throughout the 37 episodes and demonstrates that the major hysteresis loop is shown by the boundary of this loop.

The progression of the control policy's ability to obtain the hysteresis behavior is also of interest from this experiment. This information shows how well the experiment is able to use the learning capabilities of a RL algorithm. Figure 11 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 Fig. 9.

In this figure, the blue lines trace the path taken to reach the goal of 2.7% strain for three different episodes: episode 12, episode 23, and episode 30. These episodes were chosen simply because they demonstrate the progression of the policy the best by showing how the path to the goal is shortened considerably as the agent learns. 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

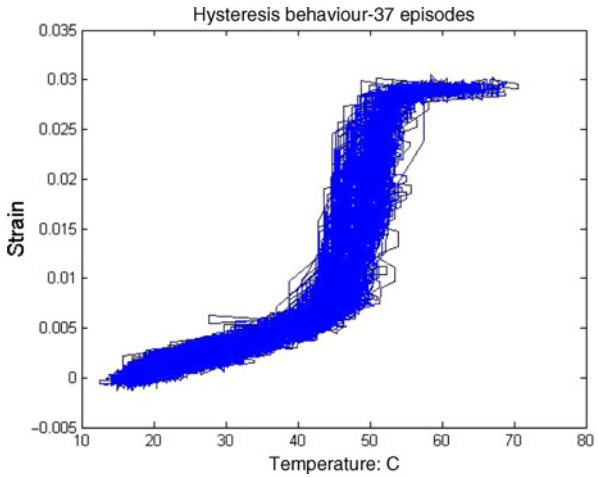

Fig. 10 Hysteresis behavior from RL.

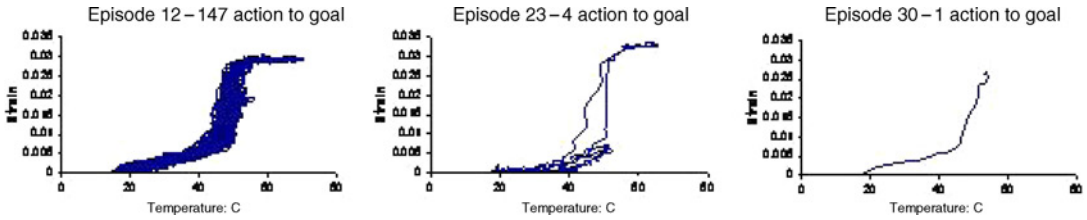

Fig. 11 Result of control policy learning.

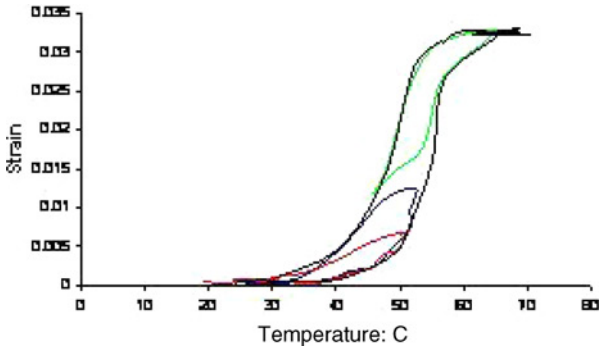

Fig. 12 Minor hysteresis loops.

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 11 shows the effects 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 SMA is 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 exploration that occurred while learning the control policy provided data that can be used to extract that major and minor hysteresis loops of the SMA wire. The minor hysteresis behavior can be extracted from individual episodes, as is demonstrated in Fig. 12.

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.

VI. Conclusions and Future Research

This research has made several conclusions about the characterization of SMA wires using Reinforcement Learning. It has been determined that water is a poor coolant for this approach, while antifreeze provides a remedy to the problems presented by water. The experimentation using both a spring and a dead weight as stress methods has revealed that a dead weight is much more useful because it keeps the restorative tensile stress constant. The dead weight provides a system with fewer variables and allows experimentation with a NiTi SMA specimen. This experiment also concluded that NiTi wires are superior to CuNiTi wires owing to the fact that the hysteresis behavior is less extreme and more difficult to model in a copper-based wire. It was also concluded that the reinforcement learning approach does indeed accomplish its goal of converging on the optimal temperature for achieving a particular goal state, which allows the program to learn the control policy and simultaneously record the temperature and strain data that maps the hysteresis. This work furthers morphing aircraft research by making it possible to use SMAs for actuators that drive the morphing process.

Acknowledgment

The material is based upon work supported by NASA under award no. NCC-1-02038. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Aeronautics and Space Administration.

References

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

[2] Waram, T., *Actuator Design Using Shape Memory Alloys*, T. C. Waram, Hamilton, Ontario, 1993.

[3] Mavroidis, C., Pfeiffer, C., and Mosley, M., "Conventional Actuators, Shape Memory Alloys, and Electrorheological Fluids," *Automation, Miniature Robotics and Sensors for Non-Destructive Testing and Evaluation*, Invited Chapter, Vol. 5.1, April 1999, pp. 10–21.

[4] Lagoudas, D., Mayes, J., and Khan, M., "Simplified Shape Memory Alloy (SMA) Material Model for Vibration Isolation," *Proceedings of Smart Structures and Materials Conference*, Newport Beach, CA, 2001.

[5] Lagoudas, 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, No. 153, 1996, pp. 153–179.

[6] Bo, 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. 1089–1249.

[7] Patoor, E., Eberhardt, A., and Berveiller, M., "Potential pseudoelastic et plasticite de transformation martensitique dans les mono-et polycristaux metalliques," *Acta Metall*, Vol. 35, No. 11, 1987, p. 2779.

[8] Falk, F., "Pseudoelastic Stress Strain Curves of Polycrystalline Shape Memory Alloys Calculated From Single Crystal Data," *International Journal of Engineering Science*, Vol. 27, 1989, p. 277.
doi: 10.1016/0020-7225(89)90115-8

[9] Banks, H., Kurdila, A., and Webb, G., "Identification of Hysteretic Control Influence Operators Representing Smart Actuators, Part II: Convergent Approximations," *Journal of Intelligent Material Systems and Structures*, Vol. 8, No. 6, 1997, pp. 536–550.

[10] Webb, 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.

[11] Tandale, M. D., Rong, J., and Valasek, J., "Preliminary Results of Adaptive-Reinforcement Learning Control for Morphing Aircraft," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA, Reston, VA, 2004, AIAA Paper 2004-5358.

[12] Haag, C., Tandale, M., and Valasek, J., "Characterization of Shape Memory Alloy Behavior and Position Control Using Reinforcement Learning," *Proceedings of the AIAA Infotech@Aerospace Conference,* AIAA, Reston, VA, 2005, AIAA Paper 2005-7160.

[13] Valasek, 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.

[14] Tandale, M. D., Valasek, J., Doebbler, J., and Meade, A. J., "Improved Adaptive-Reinforcement Learning Control for Morphing Unmanned Air Vehicles," *Proceedings of the AIAA Infotech@Aerospace Conference*, AIAA Reston, VA, 2005, AIAA Paper 2005-7159.

[15] Valasek, J., Tandale, M. D., and Rong, J., "A Reinforcement Learning–Adaptive Control Architecture for Morphing," *Proceedings of the AIAA 1st Intelligent Systems Technical Conference*, AIAA Reston, VA, 2004, AIAA Paper 2004-6220.

[16] Sutton, R., and Barto, A., *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

K. Cohen
Associate Editor