JOURNAL OF AEROSPACE INFORMATION SYSTEMS
Vol. 10, No. 6, June 2013

Characterization and Control of Hysteretic Dynamics Using Online Reinforcement Learning

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

DOI: 10.2514/1.49261

Hysteretic dynamical systems are challenging to control due to their hard nonlinearity and difficulty in modeling. One type of system with hysteretic dynamics that is gaining use in aerospace systems is the shape-memory alloy-based actuator. These actuators provide aircraft and spacecraft systems with the ability to achieve component-level or vehicle-level geometry or shape changes. Characterization of the material dynamics and properties of these actuators is usually accomplished with empirical testing of physical specimens, in which the hysteresis dynamics are often abstracted to very simplified models or ignored entirely. Machine learning techniques have the potential to learn hysteretic dynamics, but they routinely encounter difficulties that make them unsuitable. This paper proposes and develops a reinforcement learning-based approach that directly learns an input–output mapping characterization of hysteretic dynamics, which is then used as a control policy. A hyperbolic tangent-based model is used to develop a simulation of a shape-memory alloy, which is then validated experimentally using the Sarsa algorithm. The simulation model produces the temperature-versus-strain behavior and characterizes both the major and minor hysteresis loops. The learning results produce a near-optimal control policy for modulating a shape-memory alloy wire to a specified length. Results presented in the paper show that casting the shape-memory alloy control problem as a reinforcement learning problem shows promise for characterizing and controlling shape-memory alloy hysteresis behavior.

I. Introduction

RESEARCHERS have historically looked to flight in nature as inspiration for optimizing the flight performance of air vehicles. Mimicking the ability of birds to change the shape of their wings in flight to optimize flight maneuvers is seen as an achievement that would be revolutionary. Research being conducted, ranging from materials and structures to controls, is now being investigated for the potential to contribute to the development of morphing aerospace vehicles [1]. In some cases, advanced control architectures are being specifically designed for the morphing aircraft problem.

Problems often arise during the development of control policies for morphing aircraft due to the lack of knowledge of complicated actuator dynamics and displacement control. For aircraft morphing to be feasible, lightweight actuators are needed that allow shape changes into multiple configurations. This has lead to the investigation of smart materials as the basis for morphing actuators. Unfortunately, many of the most useful types of smart materials have unknown complex dynamics, and some have largely varying dynamics from sample to sample. For instance, shape-memory alloy (SMA) wires are believed to be good candidates for morphing actuators. If an SMA wire is used as the actuator to control, uncertainty is present in the model due to nonlinear hysteretic behavior and the SMA phase transformation being a thermodynamically irreversible process. The dynamics of an SMA wire also differ depending upon the specific composition and heat treatment history of the material, and the dynamics are known to be highly nonlinear and hysteretic in most cases.

This paper proposes and develops a novel computational approach for learning the input–output characteristics and control policies of systems with hysteretic behavior, such as SMA specimens, by posing the control synthesis effort as an online reinforcement learning (RL) problem. By learning the input–output behavior online, in real time, both major and minor hysteresis loops can be characterized while simultaneously learning a near-optimal control policy. An online RL approach has already been demonstrated to be useful for control of morphing, and it can be extended to hysteretic SMA characterization and control [2,3]. Since RL does not require any prior knowledge of the nonlinear behavior or the control policy, exploiting RL for morphing actuator control is advantageous [4]. The key contribution is the ability to accurately characterize SMA hysteresis behavior and provide a near-optimal control policy using an online learning agent instead of the standard empirical or constitutive modeling approaches currently used. This is done using a simulated SMA wire that is based on a hyperbolic tangent curve fit to experimentally obtained data. By casting the hysteresis behavior as a reinforcement learning problem, it is also possible to maintain lifelong learning, allowing small changes in the specimen behavior over time to be compensated by continued learning.

The paper is organized as follows. In Sec. II, the basics of reinforcement learning are explained and extended to the specifics of this paper. Details involving the Sarsa method, and how it compares to the commonly used Q-learning method, are discussed. Section III explains the mathematical model used for the SMA hysteresis simulation and provides a comparison to a commonly used hysteresis modeling method known as the Preisach model. Section IV explains how the simulated hysteretic SMA dynamics were cast as a reinforcement learning problem. In Sec. V, the results of this simulation are presented, followed by conclusions in Sec. VI.

II. Reinforcement Learning

RL is a process of learning from experience to achieve a goal [4,5]. This learning technique can be used to learn control policies without the need of a model, which makes it ideal for use in problems with either no model or very complex models [6]. RL involves the interaction of an agent with

Presented as Paper 2005-7160 at the Infotech@Aerospace Conference, Arlington, VA, 26–29 September 2005; received 8 February 2010; revision received 27 January 2013; accepted for publication 10 February 2013; published online 26 June 2013.

*Graduate Research Assistant, Vehicle Systems and Control Laboratory, Aerospace Engineering Department; kentonkirk@gmail.com. Student Member AIAA.
†Professor and Director, Vehicle Systems and Control Laboratory, Aerospace Engineering Department; valasek@tamu.edu. Associate Fellow AIAA.
‡Undergraduate Research Assistant, Vehicle Systems and Control Laboratory, Aerospace Engineering Department; chaag@tamu.edu.

an environment based on a constantly updated mapping procedure. The agent is defined as the unit making the decisions and learning from the consequences of those decisions, and the environment is defined as all external conditions outside the agent that are influenced by the agent's decisions. The agent carries out actions in its environment in a sequence of discrete time steps, $t = 0, 1, 2, 3 \ldots$ At each time step t, the agent receives a representation of the current state of the environment, denoted by $s_t \in S$, and uses this information to choose an action, denoted by $a_t \in A(s_t)$. Here, the state space S is the set of all possible states, and the action space $A(s_t)$ is the set of all actions available when in state s_t. At the next time step, $t + 1$, the agent is located in a new state, s_{t+1}, and receives a numerical reward, $r_{t+1} = R$, based on the previous action and resulting state. The objective is for the agent to learn a mapping or policy, $\pi\colon S \to A$, from the state space S to the action space A that maximizes some scalar reinforcement signal, $r\colon S \times A \to R$, over a specified period of time. An agent improves its behavior through experience by a constant updating of this policy π, which represents a mapping of states to probabilities of selecting each possible action at that state to achieve the long-term goal. Thus, $\pi_t(s, a)$ indicates the probability that action a will be selected when in state s at time t.

The various RL methods chiefly differ in the process by which the agent changes its policy due to interaction with the environment. Since the agent desires to maximize the total amount of reward that it receives over the long run, a state-value function $V^{\pi}(s)$ is defined as the expected total reward return starting from state s and continuing on with the use of policy π. Policy improvement is the process by which $V^{\pi}(s)$ is used to improve the policy π to π'. The objective of most RL methods is to find the optimal policy π^* that has the associated optimal state-value function $V^*(s)$, defined as $V^*(s) = \max_{\pi} V^{\pi}(s)$. Policy iteration is defined as the means of finding the optimal policy π^*, and policy evaluation is defined as the method of computing the current $V^{\pi}(s)$. The action-value function $Q^{\pi}(s, a)$ is also used in some applications in finding the optimal policy. The value $Q(s, a)$ is defined as the expected return acquired from taking action a when in state s and subsequently following the optimal policy. As these values are being learned, the best action from any state at its current time step is defined to be the one with the highest Q value.

The three most commonly used classes of algorithms for the solving of RL problems are dynamic programming, Monte Carlo, and temporal difference [4]. The majority of the 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. While there are temporal difference implementations that do use models, it is not necessary to do so. The most commonly used method of temporal difference is known as Q learning. Q learning is an off-policy form of temporal difference that uses an action-value function update rule based on the equation

$$Q_t(s, a) \leftarrow Q_t(s, a) + \alpha\delta_t \tag{1}$$

where s is the current state, a is the current action, Q is the action-value function to be used as a control policy, and the t subscript signifies the current time step. The constant α is the learning rate parameter that is used to "punish" the Q learning algorithm when it repeats itself within each episode. The term δ_t is defined as

$$\delta_t = r_{t+1} + \gamma \max_{a'} Q_t(s', a') - Q_t(s, a) \tag{2}$$

The term s' refers to the future state, a' is the future action, and γ represents a discount rate that is used to affect the rate of convergence by weighting the future policy. Equations (1) and (2) can be combined to form the detailed Q-learning action-value function update rule [4]:

$$Q_t(s, a) \leftarrow Q_t(s, a) + \alpha[r_{t+1} + \gamma \max_{a'} Q_t(s', a') - Q_t(s, a)] \tag{3}$$

In this paper, the reinforcement learning agent employed is the algorithm known as Sarsa, with action selection choices based on the ϵ-greedy method that will be explained in Sec. IV. The Sarsa action-value update rule is similar to Q learning, and it is as follows:

$$Q_t(s, a) \leftarrow Q_t(s, a) + \alpha[r_{t+1} + \gamma Q_t(s', a') - Q_t(s, a)] \tag{4}$$

As can be seen by comparing Eq. (3) to Eq. (4), the update rule for Sarsa is very similar to that of Q learning. The main difference between the algorithms lies in the fact that Sarsa updates the currently used policy online. The Q-learning algorithm is an off-policy method, meaning that it uses a policy for update that differs from the current approximation of the optimal policy. Sarsa is an on-policy method, meaning that it updates the action-value function using the current approximation of the optimal function. Sarsa is chosen here instead of Q learning because it allows real-time updating of the action-value function used as the control policy without waiting for the end of an episode. Therefore, if further learning episodes are needed to accommodate changes in actuator properties over time, they can be executed and the results are implemented immediately.

The development of temporal difference learning methods like Q learning or Sarsa assumes that the system is a Markov decision process (MDP). An MDP is a process by which the probability of reaching a particular state given a particular action is conditional only on the current state information and current action, and not on any past state or action information. To have guaranteed convergence to a feasible control policy, the system must be an MDP, so most implementations of RL methods assume an MDP. Hysteretic functions are commonly known to be non-MDP due to dependence on direction. While this means there is no longer a guarantee of convergence, it does not mean that Sarsa cannot converge to a feasible policy. The implementation of this algorithm to this hysteretic system is done without knowing in advance if it is possible to learn the policy, but as the results of this paper will show, it is able to do so successfully. This is an indication that a hysteretic process does not negate the possibility of using RL for successful control but simply negates the guarantee of convergence to a useful control policy.

III. Hysteretic Dynamics

What makes SMAs both useful and challenging as actuation devices is the shape-memory effect [7]. These materials can be put under a stress that leads to a seemingly plastic deformation, yet they fully recover to their original shape after heating to a high temperature. When an 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 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 nonuniformity in the phase transformations [8]. This occurs because the phase transformation from martensite to austenite begins and ends at different temperatures than the reverse process. Figures 1 and 2 demonstrate this behavior, where in Fig. 1, 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.

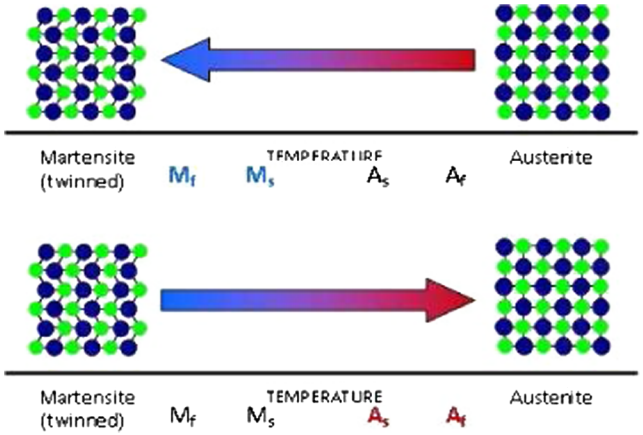

Fig. 1 Thermally induced phase transformations for a shape-memory alloy.

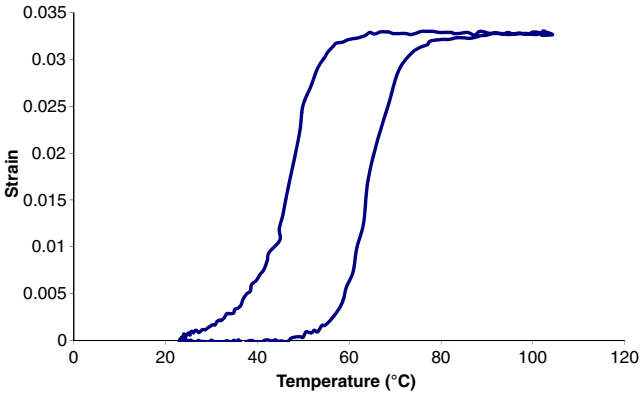

Fig. 2 Temperature–strain hysteresis for a typical shape-memory alloy.

It is important to learn the hysteresis behavior of SMAs in temperature–strain space, and this characterization is usually done through the use of constitutive models that are based on material parameters, or by models resulting from system identification [9]. 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 [10], micromechanical models [11], and empirical models based on system identification [12]. 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 they must be determined within analytical models. Generic models for modeling hysteresis include the Preisach model and the Duhem model, but these do not take into account the thermodynamic effects of the SMA transformation [13,14]. Model-based control methods are beneficial in systems where an accurate general model is available, but in the system described in this paper, the model is a curve fit of one particular material at a particular stress level.

The shape-memory effect occurs due to a temperature- and stress-dependent crystal phase transformation in the material between the martensite and austenite phases. The change that occurs within the SMA crystalline structure results in temperature hysteresis due to energy dissipation from internal friction and the creation of microstructural defects [8]. This temperature hysteresis translates directly into hysteresis in the temperature–strain relationship, and this hysteresis behavior makes it challenging to develop accurate models and control schemes for an SMA actuator.

Here, a hysteresis model based on the hyperbolic tangent function is developed. Note that this model is not used to design control laws but only to model the hysteresis behavior. The temperature–strain relationship not only has a major hysteresis loop, but it exhibits minor hysteresis loops if the direction of the change in temperature is reversed in between the minimum and maximum temperatures (T_l and T_r, respectively) spanned by the major hysteresis loop. This behavior will be explained in detail next in the modeling of the respective hysteresis loops.

The major hysteresis loop is modeled as a combination of two hyperbolic tangent functions M_r and M_l. The system follows path M_r when the temperature increases, and it follows path M_l when the temperature decreases. The subscript l refers to the lowering, or left, side of the curve. The subscript r refers to the rising, or right, side of the curve:

$$M_l = \frac{H}{2}\tanh((T - c_{tl})a) + s\left(T - \frac{c_{tl} + c_{tr}}{2}\right) + \frac{H}{2} + c_s \tag{5}$$

$$M_r = \frac{H}{2}\tanh((T - c_{tr})a) + s\left(T - \frac{c_{tl} + c_{tr}}{2}\right) + \frac{H}{2} + c_s \tag{6}$$

In Eqs. (5) and (6), H, c_{tl}, c_{tr}, a, s, and c_s are constants that dictate the shape-determining parameters of the major hysteresis loop, such as width, height, location, and slope of the lines. These constants are not based on material parameters; they were simply chosen to fit the shape of the curve numerically. By appropriately selecting these constants so that the curves match experimentally determined data, this model of the major

hysteresis loop can represent a wide range of hysteresis behaviors. It is assumed here that the minor hysteresis loops follow generally similar shapes as the major hysteresis loops, so the minor loops are modeled with similar equations but with a different height constant h. Also, all of the minor loops converge with the major loop lines beyond the temperatures T_l and T_r. The equation for a rising minor loop is

$$m_r = \frac{h}{2}\tanh((T - c_{tr})a) + s\left(T - \frac{c_{tl} + c_{tr}}{2}\right) + H - \frac{h}{2} + c_s \tag{7}$$

where h is calculated by considering that the current state is the intersection of the previous curve and the current minor loop, so that

$$h = \frac{h_{\text{prev}}(\tanh((T - c_{tl})a) + 1) - 2H}{\tanh((T - c_{tr})a) - 1} \tag{8}$$

and h_{prev} is the height parameter for the previous curve. Similarly, the equation for a lowering minor loop is

$$m_l = \frac{h}{2}\tanh((T - c_{tl})a) + s\left(T - \frac{c_{tl} + c_{tr}}{2}\right) + \frac{h}{2} + c_s \tag{9}$$

with h calculated as

$$h = \frac{h_{prev}(\tanh((T - c_{tr})a) - 1) + 2H}{\tanh((T - c_{tl})a) + 1} \tag{10}$$

For the temperature–strain relation, the hyperbolic tangent model employs constants H, c_{tl}, c_{tr}, a, s and c_s, which are manually tuned to represent any range of hysteresis loops that can exist in the domain of SMA behavior. The hyperbolic tangent-based model used in this work simulates the temperature–strain behavior of a one-dimensional NiTi SMA wire. Figure 3 shows a validation of the simulation employing the hyperbolic tangent model by comparing the major loop to an experimentally determined major hysteresis behavior for a NiTi SMA wire. The experimental data for the SMA specimen were obtained via the direct application of electrical current to a NiTi wire. The voltage was increased until the upper-limit hysteresis temperature was reached, and then the voltage was decreased until the initial length was attained. Notice that Fig. 3 only shows the major hysteresis loop of the SMA wire for both cases, but based on the assumed dependence of minor loops on the major loop and the percent composition of each crystalline phase (which can be determined from the strain and the upper-limit hysteresis temperature), the minor loops are

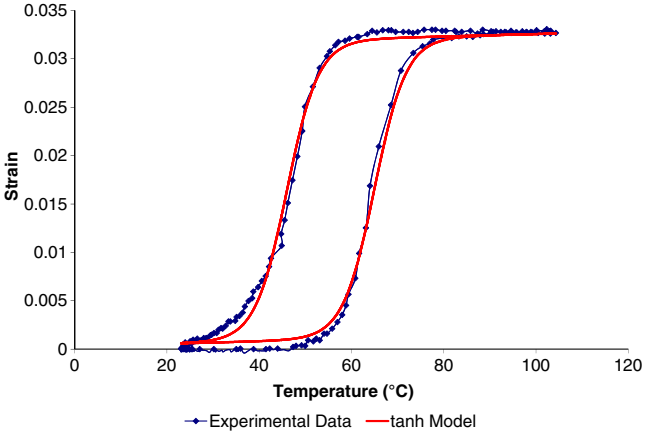

Fig. 3 Validation of modeled SMA hysteresis.

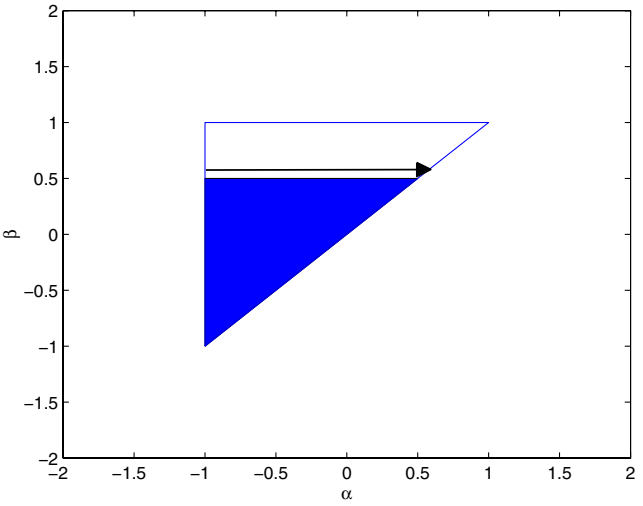

Fig. 4 Preisach plane positive motion.

also approximately known. The results validate that the hyperbolic tangent SMA model can be used to approximate the temperature–strain relationship of an experimental SMA wire.

For further validation, it is necessary to compare it to a more commonly used method of SMA hysteresis simulation. A widely accepted method of approximating hysteresis behavior among SMA researchers is the Preisach model [13,15]. The Preisach model is a general method of mapping hysteresis behavior that uses system parameters, and it can be used for a wide variety of hysteretic environments, not only SMA hysteresis [16]. 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 4 and 5 represent travel in the Preisach plane.

As shown in Fig. 4, when α is increased, the effective area becomes the area of the Preisach plane that lies below the horizontal line extending left from $\beta = \alpha$. However, Fig. 5 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 extending up from $\alpha = \beta$. The effective area in the Preisach plane is plotted as a function of α, and this new function is hysteretic. Figure 6 reveals the Preisach function corresponding to α traveling along the path $\alpha_{\min} \rightarrow \alpha_{\max} \rightarrow \alpha_{\min}$.

Figure 6 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 used here has properties associated with the crystal phase transformation, as shown in Table 1.

By using the Table 1 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 are experimentally determined values, 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 [17]. Modeling the thermoelastic effects of the minimum and maximum strain regions can be accomplished by calculating these strain values according to Eq. (11):

$$\varepsilon(T) = \frac{1}{1 + \exp[k(T - T_0)]} \tag{11}$$

In Eq. (11), 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. For the major hysteresis behavior being modeled, the values for the thermoelastic parameters were determined and can be seen in Table 2.

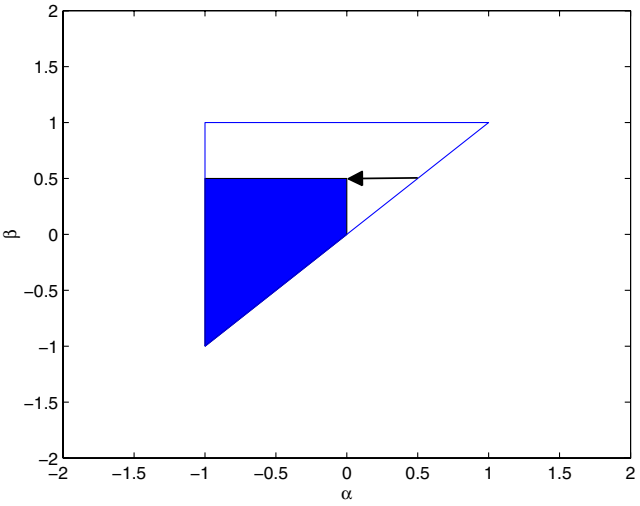

Fig. 5 Preisach plane negative motion.

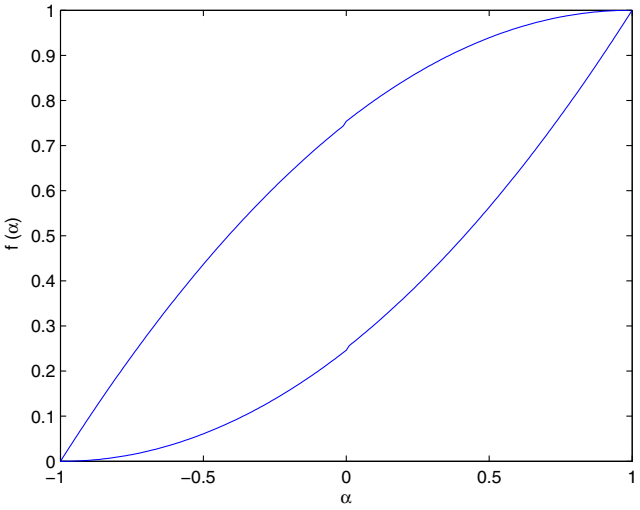

Fig. 6 Preisach model of general hysteresis.

Table 1 Material parameters input to Preisach model

Parameter	Value
M_s	60°C
M_f	35°C
A_s	45°C
A_f	75°C
$\varepsilon_{\min}$	0
$\varepsilon_{\max}$	0.033

Table 2 Thermoelastic parameters

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

Using the parameters in Table 2 with Eq. (11), 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 Fig. 7.

The modifications to the Preisach model reflected in Fig. 7 create a more accurate model of SMA hysteresis behavior. This model matches experimental data with a maximum normalized error of 0.14. This model is often used to simulate SMA hysteresis behavior, but it is limited by the time required to obtain the model outputs. For the simulation to be viable for online RL, the time required for feedback must be small. Using this modified Preisach model, the CPU time required to determine the entire major hysteresis loop in MATLAB is 144 s, so the simulation was simplified to 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 acceptably small. The average time required for MATLAB to process the hyperbolic tangent function and temperature propagation is 0.5 ms CPU time. 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 than this version of the Preisach model, with a maximum normalized error of 0.03.

The major 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. While the Preisach model has the benefit of being parameterized according to the crystal phase transformation temperatures and the minimum and maximum strains, the constants used in Eqs. (5) and (6) are chosen purely for obtaining the closest fit possible to experimental data. However, the hyperbolic tangent model is more useful for online RL because the simple function design allows for real-time feedback of the simulation.

IV. Implementation of Sarsa Algorithm

To perform a dynamic task, the SMA must experience a cycle of heating and cooling, which induces cyclic deformation. This can be accomplished with any type of applied heating, but it is most often done with resistive heating. The rate at which the temperature of the SMA component changes is dictated by the balance between heat produced in the wire by the electrical current and heat lost from the wire to the surrounding fluid via convection [18]. To simplify the state space for the RL agent, the states that can adequately describe the environment are considered to be the temperature and strain of the SMA wire. Both states are required for properly describing the reinforcement learning problem since the space being explored is hysteretic. The goal state is defined by the desired strain ε_d, and since the voltage–temperature relation is known, the action that will be used to attain this goal state is the desired temperature T_d. Hence, when the agent is at state s_t defined by a unique strain and

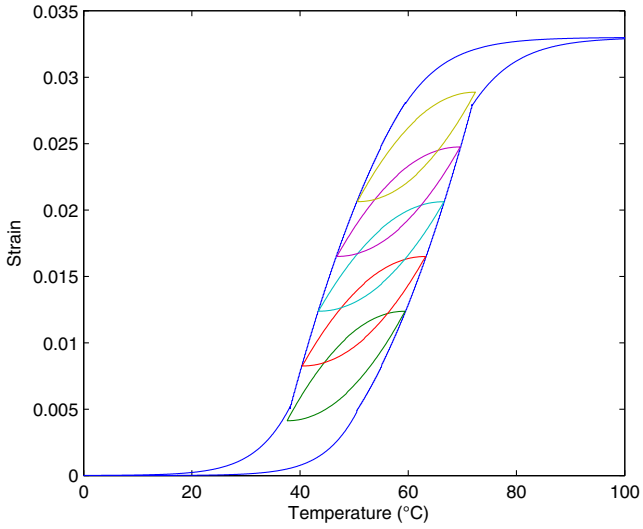

Fig. 7 Modified Preisach model simulation of SMA hysteresis.

temperature (ε_t and T_t, respectively), action a_t will be chosen that will correspond to temperature T_d, which will attempt to drive the strain to the goal state, or ε_d.

Since this implementation of Sarsa is a model-free learning method, the agent has no advance information concerning the temperature–strain relationship. Also, no knowledge of the optimal value functions or optimal desired temperatures are known. What the agent is attempting to learn is the correct T_d that will yield ε_d when at state s_t. The agent also knows all possible actions that can be taken and has accurate real-time information of the current state (ε_t and T_t) of the SMA wire. The state space and action space are discretized so that the action-value function can be created as a table of values. This discretization requirements must be fine enough to be able to reach ranges of strain desired by the user, but they must remain coarse enough to ensure learning in finite time. Only the discrete goals determined by the user will be learned. To reach the goal states, the agent seeks to learn the optimal action-value function that, given the current environmental state, commands the temperature to extend or contract the SMA wire to the desired strain.

The learning process includes an exploration–exploitation feature that helps drive the action selection process as time progresses. Some degree of exploration is needed to find actions that allow achievement of the goal since the policy is initially naïve. Choosing random actions can conceivably drive the system to instability, but the reward structure causes bad actions to be penalized while good actions are rewarded. This allows the learned policy to determine good and bad actions when exploited. The particular policy used here is known as ϵ greedy because the policy chooses to explore with probability ϵ, and it chooses greedy actions otherwise. The agent is exploiting when it chooses an action that is currently known to have the highest probability of attaining the goal state, and the agent is exploring when it chooses a random action to improve its knowledge of all the actions' Q values. At every new command selection, a uniformly distributed random variable is compared to the exploration probability ϵ to determine which choice to make. If exploration is chosen, a random command is given to the SMA that was chosen from within the set of all possible actions. If exploitation is chosen, the current Q matrix is exploited to determine the action with the highest value.

In the early stages of learning, the agent uses a uniform probability policy, which results in equal probability of all actions being selected. As time increases, the actions with a higher Q value have a higher probability of being selected based on the value of ϵ. The number of greedy actions selected increases with time until it reaches a saturation value of $\epsilon = 0.05$. This is done to allow for the possibility that a better action can be taken in case the Sarsa algorithm has not yet converged to the best policy.

V. Numerical Examples

The purpose of the numerical simulation examples is to validate and verify the learning and control performance of the online reinforcement learning agent. Validation of the agent's ability to learn hysteresis input–output characteristics online is conducted on a simulated SMA wire modeled as a hyperbolic tangent function. The agent learns and updates the action-value function over an arbitrarily selected number of 24,000 episodes, where each episode consists of 10 goal states. Within each episode, a current goal is held constant until 225 actions have been attempted, and then the agent moves on to a new goal. The agent receives a $+1$ reward for successfully attaining each goal. Refinement in learning the temperature–strain behavior is for a control policy consisting of no initial knowledge and for multiple desired strain states. The SMA wires simulated here are based upon NiTi samples.

For verification, the RL agent demonstrates the capability of learning the hysteresis behavior of two different simulated SMA wire specimens. Case 1 consists of a fully greedy policy, which the agent uses to command a temperature (action), which achieves a commanded strain (state). Case 2 consists of the characterization and control of a new SMA specimen with different physical properties. For case 2, a characterization similar to the validation case is conducted with 25,000 learning episodes, and then the learned control policy is used to command a temperature (action), which achieves a commanded strain (state). The desired strain trajectory includes multiple randomly generated goal states. New actions are commanded every 15 s to allow for damping to the correct temperature, and 15 actions are allowed before moving on to the next commanded goal. The state space S is a two-dimensional discrete Cartesian space with $S = [10, 35]$ for the first simulation and $S = [10, 50]$ in the second, where the columns represent temperatures ranging from 30–130°C with 10°C increments, and the rows represent strains ranging from 0–3.5% in the first case and 0–5% in the second, with 0.1% increments in each. Thus, with one state representing all conditions outside these acceptable boundaries, there are a total of 351 states in case 1 and 501 states in case 2. For both cases, the action space A consists of desired temperatures ranging from 30–130°C with 10°C increments, thus resulting in 10 actions per state. Rewards used to update the control policy at time t are the following: $+1$ for moving to the goal state, 0 for moving to any other permissible states, and -1 for moving to any impermissible states. The state space, action space, and reward structure are also shown in Eqs. (12–14). The constants used in Eqs. (5) and (6) were determined so that the major hysteresis best fit the experimental data referenced, and they are reported in Table 3:

$$S_{\text{rows}}: \begin{cases} \varepsilon = 0.0 \text{ to } 3.5\% \text{ in increments of } 0.01\% & \text{Case1} \\ \varepsilon = 0.0 \text{ to } 5.0\% \text{ in increments of } 0.01\% & \text{Case2} \end{cases} \tag{12}$$

$$S_{\text{cols}}: \; T = 30 \text{ to } 130^{\circ}\text{in increments of } 10^{\circ}, \qquad \text{Both Cases} \tag{13}$$

$$r = \begin{cases} 1 & \text{if } \varepsilon = \varepsilon_d \\ -1 & \text{if } \varepsilon \notin S \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Figure 8 demonstrates the learning refinement and how the agent's knowledge of hysteresis behavior evolves over time. The time that this learning through interaction encompasses is a function of the size of the action-value function $Q_t(s, a)$. Thus, as the size of the temperature–strain mesh is enlarged, the number of states and actions increases, and the required learning time increases. For episode 1, the agent initially experiences most of the possible actions at each state, regardless of the outcome. Thus, several major and minor hysteresis loop paths that do not lead to the goal are experienced. As time progresses, the agent reduces the number of exploration actions and exploits the knowledge it has obtained. This later learning shown in Fig. 8 (24,000 episodes) demonstrates that, when the agent learns how to find the goals, the paths followed are very different

Table 3 Constant values for tanh curve fit

Case	H	c_{tl}	c_{tr}	a	s	c_s
1	0.031	46	65	0.147	$1.25e-5$	0.001
2	0.033	70	90	0.067	$1.25e-4$	0.01

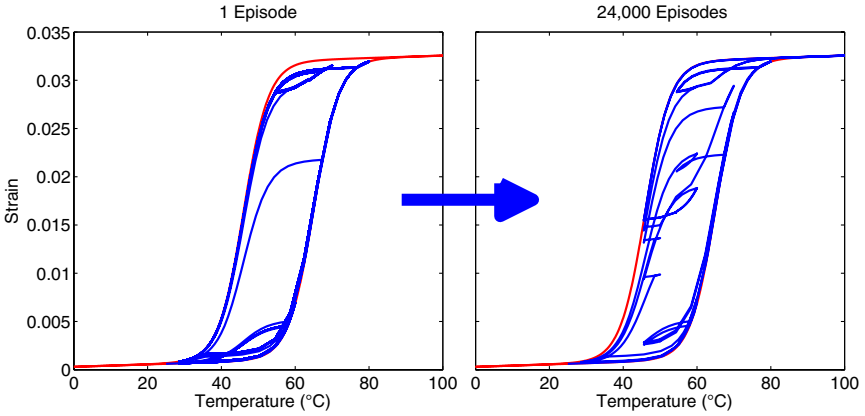

Fig. 8 Learning refinement for a simulated tanh SMA model.

than they were initially. After 24,000 learning episodes are completed by the agent, it has learned most of the SMA temperature–strain behavior and yet remained in the acceptable strain range in all exploiting actions thereafter.

For case 1, Fig. 9 shows a time history of a fully exploiting RL agent that has experienced 24,000 episodes. The action is desired temperature, and the states are current strain and current temperature. The allowed error for each goal strain was chosen to be $\pm 0.2\%$ strain. Due to the hysteresis, some states are unattainable with only one action, so they result in requiring one or two extra actions to achieve the goal. Also, there can exist some small changes within the goal range that occur as a result of all states within the range being equally rewarding for the learning agent. The transition from goal 4 to goal 5 shows that the agent leaves the goal range early but then returns permanently. This behavior was the result of early reinforcement that has not been corrected after these 24,000 episodes, but it may be overcome by more learning episodes. By comparing the behavior around the first and second goals versus the penultimate and ultimate goals in Fig. 9, it is seen that the agent makes the same choices. This

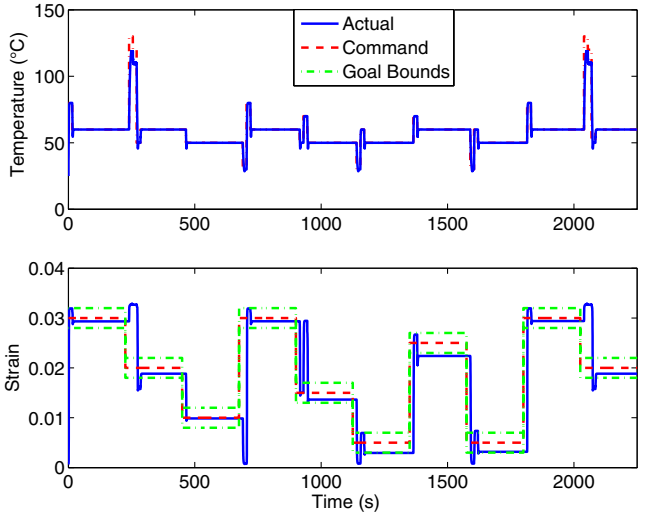

Fig. 9 Time histories of temperature actions and strain state responses: case 1.

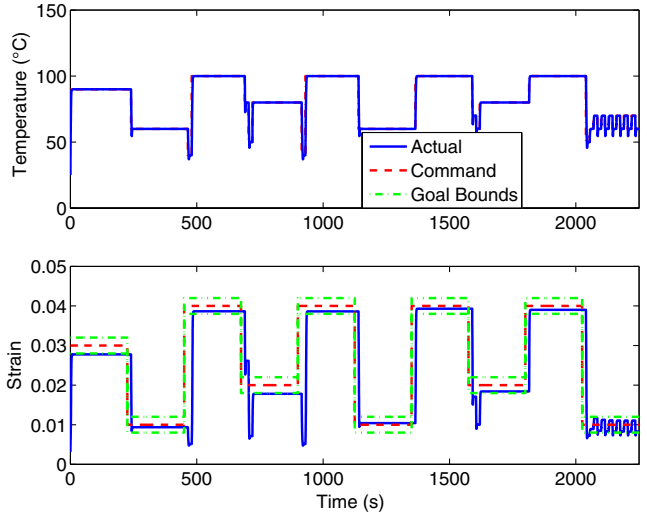

Fig. 10 Time histories of temperature actions and strain state responses: case 2.

demonstrates that, when encountering an identical current state to goal state transition, the agent chooses the same actions. Overall, the results from case 1 demonstrate that the agent is capable of achieving the goal strains quickly, and then holding position at the commanded strain.

For case 2, Fig. 10 shows the time history of a fully exploiting RL agent after experiencing 24,000 episodes for a different SMA hysteresis space. Like case 1, the RL agent was able to converge to a near-optimal control policy, and in this case, the maximum allowable error for each goal is also $\pm 0.2\%$ strain. The agent is seen to achieve the goal strains quickly, and then hold position at the commanded strain. Once again, some state-to-state transitions need more than one action to achieve the goal state. Also, oscillatory motion can be observed in the behavior around the ultimate goal because the rewards are identical within the goal range, and the agent reinforced that behavior early. This behavior may not occur if the agent were to start over.

VI. Conclusions

This paper proposed and developed an online reinforcement learning approach for directly learning an input–output mapping to control nonlinear hysteretic actuators. The method was demonstrated using a simulated shape-memory alloy wire, modeled using hyperbolic tangent functions. An online agent based upon the Sarsa algorithm was developed to learn near-optimal control policies, and the approach was applied to both the characterization and length control of simulated shape-memory alloy wires. Results presented in the paper demonstrate that the agent directly learned a control policy for the expansion and contraction control of a shape-memory alloy wire to a particular length, within a goal range of $\pm 0.2\%$ strain. While there were some state-to-state transitions that cannot occur in only one action, the agent learned to reach each goal state with as few actions as possible. The results from two different cases demonstrated that a reinforcement learning agent will make the same action choice when faced with an identical state-to-goal transition. It is concluded that the proposed online episodic learning of the temperature–strain relationship, when employing the hyperbolic tangent model, adequately approximates hysteresis characteristics and can be used as a characterization tool for systems with nonlinear hysteresis behavior such as shape-memory alloy materials. It is also concluded that casting the control of actuators with hysteretic dynamics as a reinforcement learning problem allows for the determination of a control policy without the need of a model.

Acknowledgments

This work was sponsored in part by the National Science Foundation Graduate Research Fellowship Program, by the U.S. Air Force Office of Scientific Research under contract FA9550-08-1-0038, with technical monitor Fariba Fahroo, and by Texas A&M University from the Undergraduate Summer Research Grant Program. This support is gratefully acknowledged by the authors. Any opinions, findings, conclusions. or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Air Force.

References

[1] Valasek, J. (ed.), *Morphing Aerospace Vehicles and Structures*, Wiley, Chichester, England, U.K., 2012, pp. 207–260.

[2] 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, April 2005, pp. 174–195. doi:10.2514/1.11388

[3] Valasek, J., Doebbler, 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, No. 4, Aug. 2008, pp. 1014–1020. doi:10.1109/TSMCB.2008.922018

[4] Sutton, R., and Barto, A., *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998, p. 133.

[5] Kaelbling, L. P., Littman, M., and Moore, A., "Reinforcement Learning: A Survey," *Journal of Artificial Intelligence Research*, Vol. 4, 1996, pp. 237–285. doi:10.1613/jair.301

[6] Syafiie, A., Tadeo, F., and Martinez, E., "Model-Free Intelligent Control Using Reinforcement Learning And Temporal Abstraction-Applied to pH Control," *IFAC Workshop Congress*, Vol. 16, edited by Zitek, P., July 2005.

[7] Waram, T., *Actuator Design Using Shape Memory Alloys*, T. C. Waram, Hamilton, ON, Canada, 1993.

[8] 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*, American Society for Nondestructive Testing, April 1999, pp. 10–21.

[9] Lagoudas, D., Mayes, J., and Khan, M., "Simplified Shape Memory Alloy (SMA) Material Model for Vibration Isolation," *Smart Structures and Materials Conference*, 2001, pp. 452–461.

[10] Lagoudas, D. C., Bo, A., 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. doi:10.1080/10759419608945861

[11] Patoor, E., Eberhardt, A., and Berveiller, M., "Potential Pseudoelastic et Plasticite de Transformation Martensitique dans les Mono-Et Polycristaux Metalliques," *Acta Metallurgica*, Vol. 35, No. 11, 1987, pp. 2779–2789. doi:10.1016/0001-6160(87)90276-8

[12] Banks, H., Kurdila, A., and Webb, G., "Modeling and Identification of Hysteresis in Active Material Actuators, Part 2: Convergent Approximations," *Journal of Intelligent Material Systems and Structures*, Vol. 8, No. 6, 1997, pp. 536–550. doi:10.1177/1045389X9700800606

[13] Gorbet, R. B., and Morris, K. A., "Closed-Loop Position Control of Preisach Hysteresis," *Journal of Intelligent Material Systems and Structures*, Vol. 14, No. 8, Aug. 2003, pp. 483–495. doi:10.1177/104538903035391

[14] Feng, Y., Rabbath, C. A., Chai, T., and Su, C.-Y., "Robust Adaptive Control of Systems with Hysteretic Nonlinearities: A Duhem Hysteresis Modelling Approach," *IEEE Africon*, Vol. 1, IEEE, Nairobi, Kenya, 2009, pp. 130–135.

[15] Khan, M. M., Lagoudas, D. C., Mayes, J. J., and Henderson, B. K., "Pseudoelastic SMA Spring Elements for Passive Vibration Isolation, Part 1: Modeling," *Journal of Intelligent Material Systems and Structures*, Vol. 15, No. 6, June 2004, pp. 415–441. doi:10.1177/1045389X04041529

[16] Han, Y.-M., Choi, S.-B., and Wereley, N. M., "Hysteretic Behavior of Magnetorheological Fluid and Identification Using Preisach Model," *Journal of Intelligent Material Systems and Structures*, Vol. 18, No. 9, Sept. 2007, pp. 973–981. doi:10.1177/1045389X06071647

[17] Lagoudas, 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, No. 4, Dec. 1997, pp. 405–440. doi:10.1177/108128659700200403

[18] Incropera, F., and DeWitt, D., *Fundamentals of Heat Transfer*, Wiley, New York, NY, 1981, pp. 1–33.

A. Banerjee
Associate Editor