A Hybrid-Adaptive Dynamic Programming Approach for the Model-free Control of Nonlinear Switched Systems

Wenjie Lu, *Student Member IEEE*, Pingping Zhu, *Member IEEE*, and Silvia Ferrari, *Senior Member IEEE*

***Abstract*—This paper presents a hybrid adaptive dynamic programming ADP approach (hybrid-ADP) for determining the optimal continuous and discrete control laws of a switched system online, solely from state observations. The new hybrid-ADP recurrence relationships presented in this paper are applicable to model-free control of switched hybrid systems that are possibly nonlinear. The computational complexity and convergence of the hybrid-ADP algorithm are analyzed, and the method is validated numerically showing that the optimal controller and value function can be learned iteratively from state observations.**

***Index Terms*—Adaptive Dynamic Programming, Hybrid Systems, Switched Systems, Model-free Control, Learning.**

I. Introduction

HYBRID systems consist of time-driven and event-driven kinematics. Event-driven dynamics are described by discrete state and control variables that can be represented by finite alphabets. Time-driven dynamics are described by differential or difference equations, in terms of continuous state and control vectors in Euclidean space. An important example of hybrid system that consists of a collection of subsystems comprised of time-driven dynamics and selected according to an event-driven switching rule [1], [2]. The discrete control law determines when to switch between subsystems, while the continuous control law regulates the subsystem selected by the switching rule [3].

Several approaches have been proposed to obtain the optimal discrete and continuous control laws for linear switched systems with quadratic objective functions [2]–[4]. The optimality conditions for the optimal control of switched linear-quadratic (LQ) systems were first derived in [5] using Pontryagin Minimum Principle [6]. A relaxation framework was proposed in [7] to simplify the computation of the value function for infinite-horizon switched linear quadratic regulator (LQR) problems. In [8], a relaxed dynamic programming (DP) approach was applied to the optimal control and scheduling of switched systems, by relaxing optimality within pre-specified bounds.

Adaptive dynamic programming (ADP) is an iterative approach for determining model-free optimal control laws in the presence of nonlinearities, unmodeled dynamics, control failures, or parameter variations [9], [10]. This paper presents a new hybrid-ADP approach that learns the optimal continuous and discrete control laws for a hybrid switched system online. The approach is based on novel ADP recurrence relationships that are derived from the switched system optimality conditions presented in [5]. Based on these relationships and state observations, the hybrid-ADP approach solves Bellman's equation iteratively over time, thereby adapting and optimizing the continuous and discrete control laws subject to actual system dynamics.

Novel hybrid-ADP recurrence relationships, transversality conditions, and learning algorithm are presented in Section III. The computational complexity of the hybrid-ADP algorithm is analyzed in Section III, and a proof of convergence is presented in Section IV. The hybrid-ADP approach is validated numerically using a switched LQ hybrid system for which an optimal solution can be obtained by solving a switched differential Riccati equation presented in [5]. The simulation results in Section V demonstrate that the hybrid-ADP approach converges to the optimal solution by learning the continuous and discrete control laws online from simulated system dynamics.

II. Problem Formulation and Assumptions

Switched hybrid systems are commonly used to model processes in which both discrete and continuous control inputs are crucial to system performance. The possible values of the discrete state or *mode* ξ are taken from a discrete and finite index set $\mathcal{E} = \{1, \dots, E\}$, where E typically is a small integer. The discrete control ν selects the next system mode, such that $\xi, \nu \in \mathcal{E}$. This paper considers a discrete-time switched dynamical system,

$$\mathbf{x}(k+1) = \mathbf{f}_\xi[\mathbf{x}(k), \mathbf{u}_\xi(k)], \quad \xi(k+1) = \nu(k) \tag{1}$$

where $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$ is the continuous state, $\mathcal{X}$ is the state space, $\mathbf{u}_\xi \in \mathcal{U}_\xi \subset \mathbb{R}^m$ is the continuous control input, and $\mathcal{U}_\xi$ is the space of admissible control inputs for mode ξ. For every possible value of ξ, the system dynamics, described by the function $\mathbf{f}_\xi : \mathcal{X} \times \mathcal{U}_\xi \to \mathcal{X}$, are possibly nonlinear. The initial state $\mathbf{x}(0) = \mathbf{x}_0$ and mode $\xi(0) = \xi_0$ are assumed given, and the final time N is known and finite. The switched system is also assumed to obey the following assumptions:

Assumption 1. *Mode switching can occur at any time step k and is determined solely by ν with zero cost. k^+ is the time after the switch occurs, such that $\xi(k^+) = \xi(k+1)$ for any k.*

Assumption 2. *The continuous state $\mathbf{x}$ is fully observable and the measurement error is negligible.*

The system performance is represented by the cost function,

$$J \triangleq \phi[\mathbf{x}(N)] + \sum_{j=0}^{N-1} \mathscr{L}_\xi[\mathbf{x}(j), \mathbf{u}_\xi(j), \nu(j)] \tag{2}$$

to be minimized with respect to the continuous and discrete control laws,

$$\mathbf{u}_\xi(k) = \mathbf{c}_\xi[\mathbf{x}(k), k], \;\; \nu(k) = a[\mathbf{x}(k), \xi(k), k] \tag{3}$$

respectively, where $\xi = 1, \dots, E$. Then, the goal of the hybrid-ADP algorithm is to determine the switched system *policy*, defined as the tuple $\pi = \{a, \mathbf{c}_\xi : \; \xi \in \mathcal{E}\}$.

III. Hybrid ADP Approach

ADP seeks to approximate the policy of an optimal control problem by using a recurrence relationship to improve the approximations of the optimal value function and control law over time. The two function approximations can be obtained through aggregation functions [11], [12], such as support vector regression, or neural networks. The value function approximation, commonly referred to as a critic network, and the controller approximation, referred to as an actor network, are both optimized based on observations of the state obtained from the real system or its simulation, assuming the state is fully observable. In order to accelerate converge to the optimal

Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, USA.

This work was supported by the National Science Foundation under grant ECCS 1028506.

solution, the gradient of the value function with respect to the state can be used as the critic in lieu of the scalar value function.

This section presents new ADP recurrence relationships and transversality conditions for solving the switched optimal control problem presented in Section II iteratively over time. From Bellman's principle of optimality [13], the optimization of the objective function (2) can be embedded in the optimization of a switched system value function or *cost-to-go* which, at any time k, is defined as

$$V[\mathbf{x}(k), \xi(k), \pi, k] \triangleq \phi[\mathbf{x}(N)] + \sum_{j=k}^{N-1} \mathscr{L}_\xi[\mathbf{x}(j), \mathbf{u}_\xi(j)] \tag{4}$$

From the above definition, the value function obeys the recurrence relationship,

$$\begin{aligned} V[\mathbf{x}(k), \xi(k), \pi, k] =& \mathscr{L}_\xi[\mathbf{x}(k), \mathbf{u}_\xi(k)] \\ &+ V[\mathbf{x}(k+1), \xi(k+1), \pi, k+1] \end{aligned} \tag{5}$$

and, thus, from [14], a necessary optimality condition for an extremal of (5), denoted by $\mathbf{u}_\xi^*$, is

$$\left[\frac{\partial \mathscr{L}_\xi[\mathbf{x}(k), \mathbf{u}_\xi(k)]}{\partial \mathbf{u}_\xi(k)} + \left(\frac{\partial \mathbf{f}_\xi\, [\mathbf{x}(k), \mathbf{u}_\xi(k)]}{\partial \mathbf{u}_\xi(k)} \right)^T \times \frac{\partial V[\mathbf{x}(k+1), \xi(k+1), k+1]}{\partial \mathbf{x}(k+1)} \right] \Bigg|_{\mathbf{u}_\xi(k) = \mathbf{u}_\xi^*} = \mathbf{0} \tag{6}$$

A sufficient condition for the extremal $\mathbf{u}_\xi^*$ to be a minimum of the value function is that the Hessian be positive definite [14]. Since, in practice, the sufficient condition is easily verified once an extremal is found [15], hereon $(\cdot)^*$ will denote optimality.

In order to accelerate evaluation and converge in (6), the critic network is used to approximate the gradient $\boldsymbol{\lambda} \triangleq \partial V / \partial \mathbf{x}$, also known as costate or adjoint vector [15]. Noting that $\mathbf{u}_\xi$ is a function of $\mathbf{x}$, (5) is differentiated with respect to $\mathbf{x}$ to obtain a recurrence relationship for the critic shown below in (7) and with boundary condition

$$\boldsymbol{\lambda}(N) = \partial \phi[\mathbf{x}(N)] / \partial \mathbf{x} \tag{8}$$

From Assumption 1, before and after a mode switch the Lagrangian remains constant or,

$$\mathscr{L}_\xi[\mathbf{x}(k), \mathbf{u}_\xi(k)] = \mathscr{L}_\xi[\mathbf{x}(k^+), \mathbf{u}_\xi(k^+)] \tag{9}$$

and, thus, from (4) the following holds,

$$\begin{aligned} V[\mathbf{x}(k), \xi(k), \pi, k] &= V[\mathbf{x}(k^+), \xi(k^+), \pi, k^+] \\ &= V[\mathbf{x}(k), \xi(k+1), \pi, k] \end{aligned} \tag{10}$$

because $\mathbf{x}(k^+) = \mathbf{x}(k)$ and $\xi(k^+) = \xi(k+1)$. Now, differentiating (10) with respect to $\mathbf{x}$, the recurrence relationship,

$$\begin{aligned} \boldsymbol{\lambda}[\mathbf{x}(k), \xi(k), \pi, k] &= \boldsymbol{\lambda}[\mathbf{x}(k^+), \xi(k^+), \pi, k^+] \\ &= \boldsymbol{\lambda}[\mathbf{x}(k), \xi(k+1), \pi, k] \end{aligned} \tag{11}$$

is obtained for the costate vector during a mode switch. Since $\boldsymbol{\lambda}$ is an implicit function of π, the argument will be omitted hereon for simplicity.

The optimality condition for the discrete control input is found by introducing the Hamiltonian

$$\begin{aligned} H &\triangleq \mathscr{L}_\xi[\mathbf{x}(k), \mathbf{u}_\xi(k)] + \boldsymbol{\lambda}[\mathbf{x}(k+1), \nu(k), k+1] \mathbf{f}_\xi[\mathbf{x}(k), \mathbf{u}_\xi(k)] \\ &= H[\mathbf{x}, \mathbf{u}_\xi, \boldsymbol{\lambda}, \nu, k] \end{aligned} \tag{12}$$

Then, given $\boldsymbol{\lambda}$ and $\mathbf{u}_\xi$, the discrete control law can be optimized using the discrete-time minimum principle [13], or

$$\nu^* = \underset{\nu}{\operatorname{argmin}}\, H[\mathbf{x}, \mathbf{u}_\xi, \boldsymbol{\lambda}, \nu, k] \tag{13}$$

The optimality conditions (6) and (13), and the recurrence relationships (7) and (11) are used in the next subsection to obtain approximations to the optimal control and costate approximations known as actor and critic networks.

A. Actor and Critic Network Approximations

The optimality conditions and recurrence relationships obtained in the previous subsection are used to iteratively improve upon approximations of the continuous and discrete control laws and value function gradient, until convergence to their optimal counterparts is obtained, as shown in Section IV. Artificial neural networks (NNs) are chosen here based on their universal function approximation properties [16]. Let the NN approximations of the continuous control law, costate vector, and discrete control law be denoted by $\tilde{\mathbf{c}}_\xi[\mathbf{x}(k), k; \mathbf{w}_\xi]$, $\tilde{\boldsymbol{\lambda}}[\mathbf{x}(k), \xi(k), k; \mathbf{v}]$, and $\tilde{a}[\mathbf{x}(k), \xi(k), k; \boldsymbol{\omega}]$, where $\mathbf{w}_\xi$, $\mathbf{v}$, and $\boldsymbol{\omega}$ denote vectors of adjustable parameters, and the remaining arguments denote NN inputs.

At every cycle, l, of the hybrid-ADP algorithm a new improved NN approximation is obtained by holding the others fixed, such that the policy (actor) $\tilde{\pi}^l = \{\tilde{a}^l, \tilde{\mathbf{c}}_\xi^l : \xi \in \mathcal{E}\}$ and the critic $\tilde{\boldsymbol{\lambda}}^l$, obtained during the lth cycle, are closer to optimal than those obtained during previous cycles. As a first step of cycle l, the continuous actor network is updated to satisfy the optimality condition (6) while holding the critic from the previous cycle, $\tilde{\boldsymbol{\lambda}}^{(l-1)}$, fixed. By introducing the vector function,

$$\Gamma_A = \frac{\partial \mathscr{L}_\xi}{\partial \mathbf{u}_\xi} \bigg|_{\mathbf{u}_\xi = \tilde{\mathbf{c}}_\xi^l(k)} + \left(\frac{\partial \mathbf{f}_\xi}{\partial \mathbf{u}_\xi} \right)^T \bigg|_{\mathbf{u}_\xi = \tilde{\mathbf{c}}_\xi^l(k)} \tilde{\boldsymbol{\lambda}}^{(l-1)}(k+1) \tag{14}$$

the continuous actor parameters $\mathbf{w}_\xi$ can be updated to minimize the squared L_2-norm, $\|\Gamma_A^T\ \Gamma_A\|^2$, by means of the learning rule,

$$\Delta \mathbf{w}_\xi = -\epsilon \left(2 \Gamma_A^T\ \frac{\partial \Gamma_A}{\partial \mathbf{u}_\xi}\ \frac{\partial \tilde{\mathbf{c}}_\xi^l}{\partial \mathbf{w}_\xi} \right)^T \tag{15}$$

where ϵ is a positive learning rate, and $\tilde{\mathbf{c}}_\xi^l(k)$ and $\tilde{\boldsymbol{\lambda}}^{(l-1)}(k+1)$ are short-hand notations for the continuous actor and the critic approximations evaluated at k and $k+1$, respectively.

As in classical ADP [10], the critic network is updated by holding the actor network $\tilde{\mathbf{c}}_\xi^l(k)$ fixed and by using the previous critic network $\tilde{\boldsymbol{\lambda}}^{(l-1)}(\cdot)$ to approximate the derivative of the cost-to-go in the recurrence relationship (7). Therefore, as a second step of cycle l

$$\begin{aligned} \boldsymbol{\lambda}[\mathbf{x}(k), \xi(k), \pi, k] = & \frac{\partial \mathscr{L}_\xi[\mathbf{x}(k), \mathbf{u}_\xi(k)]}{\partial \mathbf{x}(k)} + \left(\frac{\partial \mathbf{c}_\xi[\mathbf{x}(k), k]}{\partial \mathbf{x}(k)} \right)^T \frac{\partial \mathscr{L}_\xi[\mathbf{x}(k), \mathbf{u}_\xi(k)]}{\partial \mathbf{u}_\xi(k)} \\ & + \left(\frac{\partial \mathbf{f}_\xi\, [\mathbf{x}(k), \mathbf{u}_\xi(k)]}{\partial \mathbf{x}(k)} + \frac{\partial \mathbf{f}_\xi\, [\mathbf{x}(k), \mathbf{u}_\xi(k)]}{\partial \mathbf{u}_\xi(k)} \frac{\partial \mathbf{c}_\xi[\mathbf{x}(k), k]}{\partial \mathbf{x}(k)} \right)^T \boldsymbol{\lambda}[\mathbf{x}(k+1), \xi(k+1), \pi, k+1] \end{aligned} \tag{7}$$

in the hybrid-ADP algorithm, a target vector function,

$$\Gamma_C = \frac{\partial \mathscr{L}_\xi}{\partial \mathbf{x}}\bigg|_{\mathbf{u}_\xi = \tilde{\mathbf{c}}^l_\xi(k)} + \left(\frac{\partial \tilde{\mathbf{c}}_\xi}{\partial \mathbf{x}}\right)^T \frac{\partial \mathscr{L}_\xi}{\partial \mathbf{u}_\xi}\bigg|_{\mathbf{u}_\xi = \tilde{\mathbf{c}}^l_\xi(k)} + \left(\frac{\partial \mathbf{f}_\xi}{\partial \mathbf{x}} + \frac{\partial \mathbf{f}_\xi}{\partial \mathbf{u}_\xi}\frac{\partial \tilde{\mathbf{c}}^l_\xi}{\partial \mathbf{x}}\right)^T\Bigg|_{\mathbf{u}_\xi = \tilde{\mathbf{c}}^l_\xi(k)} \tilde{\boldsymbol{\lambda}}^{(l-1)}(k+1) \tag{16}$$

is obtained from (7) and (11), such that the critic parameters $\mathbf{v}$ can be updated to minimize the squared L_2-norm, $\|(\Gamma_C - \tilde{\boldsymbol{\lambda}}^l(k)\|^2$, by means of the learning rule,

$$\Delta \mathbf{v} = 2\ \eta \left(\frac{\partial \tilde{\boldsymbol{\lambda}}^l}{\partial \mathbf{v}}\right)^T\Bigg|_k \left[\Gamma_C - \tilde{\boldsymbol{\lambda}}^l(k)\right] \tag{17}$$

where η is a positive learning rate. Finally, as a third step of cycle l in the hybrid-ADP algorithm, the discrete actor network is updated according to the optimality condition in (13) by updating the parameters $\boldsymbol{\omega}$, while holding $\tilde{\mathbf{c}}^l_\xi$ and $\tilde{\boldsymbol{\lambda}}^l$ fixed.

IV. Hybrid-ADP Analysis and Proof of Convergence

The computational complexity and convergence properties of the hybrid ADP approach presented in the previous section are analyzed in this section under the assumption that the value function and NN approximations have Lipschitz continuous gradient. For continuous state values $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$, V has a Lipschitz continuous gradient if the inequality $\|\partial V/\partial \mathbf{x}|_{\mathbf{x}=\mathbf{x}_1} - \partial V/\partial \mathbf{x}|_{\mathbf{x}=\mathbf{x}_2}\| \leq L\|\mathbf{x}_1 - \mathbf{x}_2\|$ holds for a constant modulus L [17]. This and other regularity conditions are common assumptions in the optimization literature that can be satisfied by a suitable choice of Lagrangian function, provided the design objectives are not highly nonlinear with respect to the state and the control [17]–[19].

Assumption 3. *Assume gradients $\partial V/\partial \mathbf{x}$, $\partial \mathbf{x}/\partial \mathbf{u}_\xi$, $\partial \mathscr{L}_\xi/\partial \mathbf{u}_\xi$, $\partial \tilde{\mathbf{c}}_\xi/\partial \mathbf{w}_\xi$, and $\partial \tilde{\boldsymbol{\lambda}}/\partial \mathbf{v}$ are Lipschitz continuous with modulus L_1, L_2, L_3, L_4, and L_5 respectively.*

Let N_H and N_S denote the number of hidden neurons and training samples for a critic/actor network. From [20], an approximation error ε can be guaranteed by choosing $N_S = \text{vol}(\mathcal{X})L/(\varepsilon)^n$, where $\text{vol}(\mathcal{X})$ is the volume of the state space $\mathcal{X} \subset \mathbb{R}^n$. Because NNs are universal function approximators, on a compact space $\mathcal{X}$, ε can be made arbitrarily small by increasing N_H [16], [21]. While an choice is $N_H = (n + \sqrt{N_S})/N_L$, where N_L is the number of hidden layers [22], a zero approximation error for gradient and output training samples can be guaranteed when $N_H = N_S$ for $N_L = 1$ [23]. In this paper, one critic and one actor are used to approximate $\boldsymbol{\lambda}$ and $\mathbf{c}_\xi$ for each mode and, thus, a total of $2E$ networks are implemented. Then, the computational complexity of each epoch (15) or (17) is $O(N^L_H + nN_H)$, and thus the total complexity of each hybrid-ADP cycle is $O[E(N^L_H + nN_H)TN]$, where T is the number of epochs. Based on these results, it can be assumed that ε can be made negligibly small by a suitable choice of N_H and N_L, with reasonable computational requirements.

Four lemmas are presented and then used to obtain the hybrid-ADP proof of convergence. The first two lemmas build connections among the recurrence relationship, the value function, and updating rules of the actor and critic networks. The last two lemmas establish the progression of the policy and value function updates at consecutive iterations, as schematized in Fig. 1. Then, the hybrid-ADP algorithm presented in Section III can be guaranteed to converge to a globally or locally optimal solution.

$$\begin{tikzcd}
\cdots & \tilde{\pi}^l \arrow[d] & \tilde{\pi}^{l+1} \arrow[d] & \cdots & \tilde{\pi}^\infty \arrow[d] & = & \tilde{\pi}^{\infty+1} \arrow[d] \\
\cdots & \tilde{\boldsymbol{\lambda}}^l \arrow[d] \arrow[ur] & \tilde{\boldsymbol{\lambda}}^{l+1} \arrow[d] \arrow[ur] & \cdots & \tilde{\boldsymbol{\lambda}}^\infty \arrow[d] & = & \tilde{\boldsymbol{\lambda}}^{\infty+1} \arrow[d] \\
\cdots & V^l(\mathbf{x},\xi,k) \geq & V^{l+1}(\mathbf{x},\xi,k) \geq & \cdots & V^\infty(\mathbf{x},\xi,k) & = & V^{\infty+1}(\mathbf{x},\xi,k)
\end{tikzcd}$$

Fig. 1. Actor and critic network hybrid-ADP updates.

Lemma 1. *Let $\tilde{\pi}^l$ denote the policy obtained from the lth cycle of the hybrid-ADP algorithm. Then, for any $(TN) \gg 1$, the critic $\tilde{\boldsymbol{\lambda}}^l$, obtained from (17) while holding $\tilde{\pi}^l$ fixed, satisfies,*

$$\tilde{\boldsymbol{\lambda}}^l[\mathbf{x}(k), \xi(k), k] = \frac{\partial \mathscr{L}_\xi[\mathbf{x}(k), \mathbf{u}_\xi(k)]}{\partial \mathbf{x}(k)}\bigg|_{\mathbf{u}_\xi = \tilde{\mathbf{c}}^l_\xi(k)} + \left(\frac{\partial \tilde{\mathbf{c}}^l_\xi[\mathbf{x}(k), k]}{\partial \mathbf{x}(k)}\right)^T \times \frac{\partial \mathscr{L}_\xi[\mathbf{x}(k), \mathbf{u}_\xi(k)]}{\partial \mathbf{u}_\xi(k)}\bigg|_{\mathbf{u}_\xi = \tilde{\mathbf{c}}^l_\xi(k)} + \left[\frac{\partial \mathbf{x}(k+1)}{\partial \mathbf{u}_\xi(k)}\frac{\partial \tilde{\mathbf{c}}^l_\xi[\mathbf{x}(k), k]}{\partial \mathbf{x}(k)} + \frac{\partial \mathbf{x}(k+1)}{\partial \mathbf{x}(k)}\right] \tilde{\boldsymbol{\lambda}}^l[\mathbf{x}(k+1), \xi(k+1), k+1] \tag{18}$$

where $\mathbf{x}(k+1)$ and $\xi(k+1)$ are the state values obtained by implementing the control policy $\tilde{\pi}^l$ in (1).

Proof of Lemma 1. For any $\mathbf{x}_0 \in \mathcal{X}$, the trajectory of $\mathbf{x}(k)$, $k = 1, \ldots, N$, obtained by policy $\tilde{\pi}^l$, is fixed. Then, at time step k, the coefficient of $\tilde{\boldsymbol{\lambda}}^l(k+1)$ and the remaining term in (17) are all constant matrices or vectors evaluated at $\mathbf{x}(k)$, and can be denoted as $\mathbf{A}(k)$ and $\mathbf{b}(k)$, respectively. Then, the recurrence relationship (7) can be written as,

$$\tilde{\boldsymbol{\lambda}}^l(k) = \mathbf{A}(k)\tilde{\boldsymbol{\lambda}}^l(k+1) + \mathbf{b}(k) \tag{19}$$

which is the kth equation in a linear system of equations, where Nth equation is $\tilde{\boldsymbol{\lambda}}^l(N) = \partial_{\mathbf{x}}\phi[\mathbf{x}(N)]$. Thus, (17) follows the successive over-relaxation (SOR) method with relaxation factor η [24]. The eigenvalue of the iteration matrix for this linear system is $1 - \eta$, and thus has an absolute value less than one. Then, when $(TN) \gg 1$, the solution of (19) can be computed by SOR [25], and the resulting approximation $\tilde{\boldsymbol{\lambda}}^l$ satisfies (18). □

Remark 1. *Any critic $\tilde{\boldsymbol{\lambda}}^l[\mathbf{x}, \xi, k]$ obtained from (17) also satisfies the boundary condition $\tilde{\boldsymbol{\lambda}}^l[\mathbf{x}(N), \xi(N), N] = \partial_{\mathbf{x}}\phi[\mathbf{x}(N)]$.*

Lemma 2. *When holding $\tilde{\pi}^l$ fixed, the critic network $\tilde{\boldsymbol{\lambda}}^l[\mathbf{x}, \xi, k]$ and its corresponding value function $V^l(\mathbf{x}, \xi, k)$ obey the relationships,*

$$V^l[\mathbf{x}(k), \xi(k), k] = \mathscr{L}_\xi[\mathbf{x}(k), \mathbf{u}_\xi(k)] + V^l[\mathbf{x}(k+1), \xi(k+1), k+1] \tag{20}$$

$$V^l[\mathbf{x}(N), \xi(N), N] = \phi[\mathbf{x}(N)] \tag{21}$$

$$\partial V^l[\mathbf{x}(k), \xi(k), k]/\partial \mathbf{x}(k) = \tilde{\boldsymbol{\lambda}}^l[\mathbf{x}(k), \xi(k), k] \tag{22}$$

for all $\mathbf{x}(k)$ and $\xi(k)$, and at any time step k.

Proof of Lemma 2. Equations (20)-(21) hold from (4), and (22) is proven by induction for any $k \leq N$ as follows.

Base case: At $k = N$, $\partial_{\mathbf{x}} V^l|_N = \partial_{\mathbf{x}}\phi[\mathbf{x}(N)]$. Therefore, from Remark 1,

$$\partial_{\mathbf{x}} V^l|_N = \tilde{\boldsymbol{\lambda}}^l[\mathbf{x}(N), \xi(N), N] \tag{23}$$

and thus (22) holds for $k = N$.

Induction step: Let $k < N$ be given and suppose (22) is true at time instant $k + 1$, such that

$$\partial_{\mathbf{x}} V^l|_{k+1} = \tilde{\boldsymbol{\lambda}}^l[\mathbf{x}(k+1), \xi(k+1), k+1]$$

Substituting the above into the Right-Hand Side (RHS) of (18) it can be easily shown that the RHS of (18) is equal to $\partial_{\mathbf{x}} V^l|_k$, while its

Left-Hand Side (LHS) is equal to $\tilde{\boldsymbol{\lambda}}^l(k)$. It follows that

$$\partial_{\mathbf{x}} V^l|_k = \tilde{\boldsymbol{\lambda}}^l[\mathbf{x}(k), \xi(k), k] \tag{24}$$

and (22) holds for any $k \leq N$.

Conclusion: By the principle of induction, (22) is true for any $k \leq N$. □

Lemma 3. *When holding the critic $\tilde{\boldsymbol{\lambda}}^l$ fixed, the control law $\tilde{\mathbf{c}}_\xi^{l+1}$ obtained from the learning rule (15) using the learning rate $\epsilon = 1/(L_1 L_2 + L_3)L_4$, has a value function,*

$$\begin{aligned} &V^l[\mathbf{x}(k+1), \xi(k+1), k+1]|_{\xi(k+1)=\nu(k)} + \mathscr{L}_\xi[\mathbf{x}(k), \tilde{\mathbf{c}}_\xi^{l+1}(k)] \\ &\leq V^l[\mathbf{x}(k+1), \xi(k+1), k+1]|_{\xi(k+1)=\nu(k)} + \mathscr{L}_\xi[\mathbf{x}(k), \tilde{\mathbf{c}}_\xi^{l}(k)] \end{aligned} \tag{25}$$

for all $\nu(k) \in \mathcal{E}$.

Proof of Lemma 3. When $\mathbf{x}$ and ν are given, the control input $\mathbf{u}_\xi$ is a function of the actor weights obtained at the end of the lth cycle, denoted by $\mathbf{w}_\xi^l$. Thus, the next state $\mathbf{x}(k+1)$ and value function $V^l[\mathbf{x}(k+1), \xi(k), k+1]$ are also functions of $\mathbf{w}_\xi^l$. Let $G \triangleq \{V^l[\mathbf{x}(k+1), \xi(k), k+1] + \mathscr{L}_\xi[\mathbf{x}(k), \tilde{\mathbf{c}}_\xi^l(k)] = G(\mathbf{w}_\xi^l)$. Then, from Assumption (3), G has a Lipschitz continuous gradient with modulus $(L_1 L_2 + L_3)L_4$, because

$$\begin{aligned} \|G(\mathbf{w}_\xi^l) - G(\mathbf{w}_\xi^{(l+1)})\| &= \|V^l[\mathbf{x}(k+1), \xi(k), k] + \mathscr{L}_\xi[\mathbf{x}(k), \tilde{\mathbf{c}}_\xi^l(k)] \\ &\quad - V^{(l+1)}[\mathbf{x}(k+1), \xi(k), k] - \mathscr{L}_\xi[\mathbf{x}(k), \tilde{\mathbf{c}}_\xi^{(l+1)}(k)]\| \\ &\leq \|V^l[\mathbf{x}(k+1), \xi(k), k] - V^{(l+1)}[\mathbf{x}(k+1), \xi(k), k]\| \\ &\quad + \|\mathscr{L}_\xi[\mathbf{x}(k), \tilde{\mathbf{c}}_\xi^l(k)] - \mathscr{L}_\xi[\mathbf{x}(k), \tilde{\mathbf{c}}_\xi^{(l+1)}(k)]\| \\ &\leq (L_1 L_2 + L_3)L_4 \;\|(\mathbf{w}_\xi^{(l+1)} - \mathbf{w}_\xi^l)\| \end{aligned} \tag{26}$$

From the actor learning rule (15) and the properties of functions with Lipschitz continuous gradient [26], it also follows that,

$$\begin{aligned} G(\mathbf{w}_\xi^{(l+1)}) &\leq G(\mathbf{w}_\xi^l) + < \partial_{\mathbf{w}_\xi} G|_{\mathbf{w}_\xi = \mathbf{w}_\xi^l}, (\mathbf{w}_\xi^{(l+1)} - \mathbf{w}_\xi^l) > \\ &\quad + \frac{(L_1 L_2 + L_3)L_4}{2} \|(\mathbf{w}_\xi^{(l+1)} - \mathbf{w}_\xi^l)\|^2 \\ &= G(\mathbf{w}_\xi^l) + [(L_1 L_2 + L_3)L_4 \epsilon^2/2 - \epsilon] \|\nabla G(\mathbf{w}_\xi^l)\|^2. \end{aligned}$$

where $< \cdot, \cdot >$ denotes the inner product, and, when $\epsilon \leq 2/(L_1 L_2 + L_3)L_4$, $G(\mathbf{w}_\xi^{(l+1)}) \leq G(\mathbf{w}_\xi^l)$, and, thus, (25) holds. □

Remark 2. *The equality in (25) holds iff $\|\partial_{\mathbf{w}_\xi} G|_{\mathbf{w}_\xi = \mathbf{w}_\xi^l}\| = 0$, i.e., iff the optimality condition (6) is satisfied, and the maximum convergence rate is achieved by setting $\epsilon = 1/(L_1 L_2 + L_3)L_4$.*

Lemma 4. *Let $\tilde{\pi}^{l+1}$ denote the policy obtained in the $(l+1)$th cycle of the hubrid-ADP algorithm, while holding $\tilde{\boldsymbol{\lambda}}^l$ fixed. From (13) and (15) it follows that the value function obeys the inequality,*

$$\begin{aligned} &V^{(l+1)}[\mathbf{x}(k+1), \xi(k+1), k+1] + \mathscr{L}_\xi[\mathbf{x}(k), \tilde{\mathbf{c}}_\xi^{(l+1)}(k)] \\ &\leq V^l[\mathbf{x}(k+1), \xi(k+1), k+1] + \mathscr{L}_\xi[\mathbf{x}(k), \tilde{\mathbf{c}}_\xi^l(k)], \end{aligned}$$

during subsequent cycles l and $(l+1)$ of the algorithm.

Proof of Lemma 4. From the minimum principle for discrete-time problems [13], it can be shown that,

$$\begin{aligned} V^l[\mathbf{x}(k), \xi(k), k] &= \mathbf{x}(k)\boldsymbol{\lambda}(k) + \phi[\mathbf{x}(N)] - \mathbf{x}(N)\boldsymbol{\lambda}(N) \\ &\quad + \sum_{j=k}^{N-1} \left\{ H_\nu[\mathbf{x}, \tilde{\mathbf{c}}_\xi^{(l+1)}, \boldsymbol{\lambda}, \nu, j] - \mathbf{x}(j)\boldsymbol{\lambda}(j) \right\} \end{aligned}$$

where only the Hamiltonian $H_\nu[\cdot]$ is a function of ν. Therefore, $\nu(k)$ obtained from (13) minimizes $V^l[\mathbf{x}(k), \xi(k), k]$. Furthermore, from Lemma 2, for any $\nu \in \mathcal{E}$ the control approximation $\tilde{\mathbf{c}}_\xi^{(l+1)}$ has a value function,

$$\begin{aligned} &V^{(l+1)}[\mathbf{x}(k+1), \xi(k+1), k+1] + \mathscr{L}_\xi[\mathbf{x}(k), \tilde{\mathbf{c}}_\xi^{(l+1)}] \\ &\leq V^l[\mathbf{x}(k+1), \xi(k+1), k+1] + \mathscr{L}_\xi[\mathbf{x}(k), \tilde{\mathbf{c}}_\xi^l] \end{aligned} \tag{27}$$

and, thus, $\tilde{\pi}^{(l+1)}$ results in a lower cost. □

Theorem 1 (Convergence)**.** *At every cycle l of the hybrid-ADP algorithm, the critic and actor networks obtained from (13)-(17) are characterized by an improved value function, such that $V^{(l+1)}[\mathbf{x}(k), \xi(k), k] \leq V^l[\mathbf{x}(k), \xi(k), k]$, for any $(TN) \gg 1$, $\mathbf{x} \in \mathcal{X}$, $\xi \in \mathcal{E}$, and $k = 1, \ldots, N$. Furthermore, as $l \to \infty$, the actor networks converge to an extremal policy $\pi^\infty = \{a^\infty, \mathbf{c}_\xi^\infty : \xi \in \mathcal{E}\}$, and the critic converges to a value function V^∞ that is stationary with respect to the policy. Then, when the Hamiltonian (12) is a convex function of $\mathbf{x}$ and $\mathbf{u}_\xi$, π^∞ is an optimal policy of the switched optimal control problem (1)-(2).*

Proof of Convergence. From Lemma 2, the value function at the lth cycle can be written as,

$$\begin{aligned} V^l[\mathbf{x}(k), \xi(k), k] &= \mathscr{L}_\xi[\mathbf{x}(k), \tilde{\mathbf{c}}_\xi^l(k)] \\ &\quad + V^l[\mathbf{x}(k+1), \xi(k+1), k+1] \end{aligned} \tag{28}$$

and the value function at the $(l+1)$th cycle can be written as

$$\begin{aligned} V^{(l+1)}[\mathbf{x}(k), \xi(k), k] &= \mathscr{L}_\xi[\mathbf{x}(k), \tilde{\mathbf{c}}_\xi^{(l+1)}(k)] \\ &\quad + V^{(l+1)}[\mathbf{x}(k+1), \xi(k+1), k+1] \end{aligned} \tag{29}$$

Then, subtracting (28) from (29), the change in value function during one cycle is

$$\begin{aligned} &V^{(l+1)}[\mathbf{x}(k), \xi(k), k] - V^l[\mathbf{x}(k), \xi(k), k] \\ &= \mathscr{L}_\xi[\mathbf{x}(k), \tilde{\mathbf{c}}_\xi^{(l+1)}(k)] - V^l[\mathbf{x}(k+1), \xi(k+1), k+1] \\ &\quad + V^{(l+1)}[\mathbf{x}(k+1), \xi(k+1), k+1] - \mathscr{L}_\xi[\mathbf{x}(k), \tilde{\mathbf{c}}_\xi^l(k)] \end{aligned}$$

From Lemma 4, the change in value function during one cycle obeys the following inequality,

$$\begin{aligned} &V^{(l+1)}[\mathbf{x}(k), \xi(k), k] - V^l[\mathbf{x}(k), \xi(k), k] \\ &\leq \mathscr{L}_\xi[\mathbf{x}(k), \tilde{\mathbf{c}}_\xi^{(l+1)}(k)] - V^l[\mathbf{x}(k+1), \xi(k+1), k+1] \\ &+ V^{(l+1)}[\mathbf{x}(k+1), \xi(k+1), k+1] - \mathscr{L}_\xi[\mathbf{x}(k), \tilde{\mathbf{c}}_\xi^{(l+1)}(k)] \\ &= V^{(l+1)}[\mathbf{x}(k+1), \xi(k+1), k+1] - V^l[\mathbf{x}(k+1), \xi(k+1), k+1] \end{aligned} \tag{30}$$

and from the boundary condition (8) it follows that,

$$V^l[\mathbf{x}(N), \xi(N), N] = V^{l+1}[\mathbf{x}(N), \xi(N), N]$$

and, thus, it can be concluded that,

$$\begin{aligned} &V^{(l+1)}[\mathbf{x}(k), \xi(k), k] - V^l[\mathbf{x}(k), \xi(k), k] \\ &\leq V^{(l+1)}[\mathbf{x}(N), \xi(N), N] - V^l[\mathbf{x}(N), \xi(N), N] = 0 \end{aligned} \tag{31}$$

for any $\mathbf{x} \in \mathcal{X}$, $\xi \in \mathcal{E}$, and $k = 1, \ldots, N$.

Let $\inf\{V^l\}$ denote the lower bound of V^l. Since $V^l[\mathbf{x}(k), \xi(k), k]$ is non-negative, for any $\sigma > 0$, there exists a positive integer s such that $V^s < \inf\{V^l\} + \sigma$. From Lemma 4, it follows that $\|\inf\{V^l\} - V^l\| \leq \|\inf\{V^l\} - V^s\| < \sigma$ for all $l > s$, and, by definition $\lim_{l\to\infty}\{V^l\} = \inf\{V^l\}$, such that as $l \to \infty$ an extremal policy π^∞ is obtained. If $H[\cdot]$ is convex in $\mathbf{u}_\xi$ and $\mathbf{x}$, for every discrete action $\mathbf{c}_\xi^\infty$ minimizes $H[\cdot]$. Then, from Remark 2, when $\pi^l \approx \pi^{l+1}$, $\pi^{l+1} \to \pi^\infty$ and π^∞ satisfies the optimality condition (6). Thus, π^∞ is a globally optimal solution according to the discrete-time minimum principle [13]. If $H[\cdot]$ is not convex, if the Hessian is positive definite in a neighborhood of the extremal

$\mathbf{c}_\xi^\infty$, π^∞ is a locally optimal solution [27]. Thus, convergence to an optimal policy is guaranteed for $\epsilon \leq 2/(L_1L_2 + L_3)L_4$ and, from Remark 2, maximum convergence rate is achieved for $\epsilon = 1/(L_1L_2 + L_3)L_4$. □

As example of the above argument, if the Lagrangian is a quadratic function of $\mathbf{x}$ and $\mathbf{u}$ for every mode of the switched system, and the vector function $\mathbf{f}_\xi$ in (1) is an affine function of $\mathbf{x}$ and $\mathbf{u}$, then the Hamiltonian for every mode is convex in $\mathbf{x}$ and $\mathbf{u}$. In this case, π^∞ is a globally optimal solution. In general, π^∞ can only be guaranteed to be a locally optimal solution, and multiple proper initializations may be used to search for a better local optimum.

V. Numerical Simulations and Results

The hybrid-ADP algorithm presented in Section III is demonstrated on a switched linear-quadratic (LQ) optimal control problem that can be solved numerically using the switched differential Riccati equation (SDRE) derived in [5]. The switched LQ system considered in this paper consists of a power system with a gasoline-driven mode and an electric-driven mode that can each be represented by linear time-invariant (LTI) dynamics with a continuous state vector $\mathbf{x} = [x \ \dot{x}]^T$, where $x \in \mathbb{R}$, that is fully observable and error free. The mode of the power system is represented by a discrete binary state variable $\xi \in \mathcal{E}$, where $\mathcal{E} = \{1, 2\}$, $\xi = 1$ denotes the gasoline-driven model, and $\xi = 2$ denotes the electric-driven mode. The system can switch to any of the two modes at any time, and the two power systems are independent and supplied with sufficient fuel. The system dynamics under each mode are modeled by an LTI system,

$$\mathbf{x}(k+1) = \begin{cases} \mathbf{A}_1\mathbf{x}(k) + \mathbf{B}_1 u(k), & \text{for } \nu(k) = 1 \\ \mathbf{A}_2\mathbf{x}(k) + \mathbf{B}_2 u(k), & \text{for } \nu(k) = 2 \end{cases} \tag{32}$$

where $u \in \mathbb{R}$ is the continuous control input, and the initial continuous state, $\mathbf{x}(0) = \mathbf{x}_0$, is given. In gasoline-driven mode the state-space matrices are,

$$\mathbf{A}_1 = \begin{pmatrix} 1 & 0.05 \\ -0.05 & 0.95 \end{pmatrix}, \quad \text{and} \quad \mathbf{B}_1 = \begin{pmatrix} 0 \\ 0.05 \end{pmatrix}, \tag{33}$$

and in electric-driven mode they are

$$\mathbf{A}_2 = \begin{pmatrix} 1 & 0.05 \\ -0.05 & 0.975 \end{pmatrix}, \quad \text{and} \quad \mathbf{B}_2 = \begin{pmatrix} 0 \\ 0.04 \end{pmatrix}. \tag{34}$$

At any time $k \in \{0, \ldots, (N-1)\}$, the system mode ξ can be fully controlled at no cost by a switching signal $\nu \in \mathcal{E}$ provided by the discrete controller. Unlike gain-scheduled designs, the system performance depends on both the discrete and continuous state and control histories, and is defined differently between modes. Thus, the cost function to be minimized is represented by,

$$J = \mathbf{x}^T(N)\mathbf{P}_f\mathbf{x}(N) + \sum_{j=0}^{N-1} \mathbf{x}^T(j)\mathbf{Q}_\xi\mathbf{x}(j) + u_\xi^T(j)R_\xi u_\xi(j)$$

where $N = 100$, and the weighting matrices of the gasoline-driven mode are,

$$\mathbf{Q}_1 = \begin{pmatrix} 100 & 0 \\ 0 & 200 \end{pmatrix}, \quad \text{and} \quad R_1 = 400, \tag{35}$$

while those of the electric-driven mode are

$$\mathbf{Q}_2 = \begin{pmatrix} 250 & 0 \\ 0 & 200 \end{pmatrix}, \quad \text{and} \quad R_2 = 50. \tag{36}$$

The terminal cost is defined by the matrix,

$$\mathbf{P}_f = \begin{pmatrix} 1500 & -1500 \\ -1500 & 3000 \end{pmatrix} \tag{37}$$

and the initial conditions are $\mathbf{x}(0) = [0.5596 \ -0.6387]^T$ and $\xi(0) = 1$. From [5], the switched differential Riccati Equation is given by,

$$\mathbf{P}(k-1) - \mathbf{Q}_\xi = \mathbf{A}_\xi^T\Big(\mathbf{P}(k) - \mathbf{P}(k)\mathbf{B}_\xi(R_\xi + \mathbf{B}_\xi^T\mathbf{P}(k)\mathbf{B}_\xi)^{-1}\mathbf{B}_\xi^T\mathbf{P}(k)\Big)\mathbf{A}_\xi$$

where the discrete controller is obtained by minimizing the Hamiltonian function [13], such that

$$\nu(k) = \underset{\nu}{\operatorname{argmin}}\{H[\mathbf{P}(k), \mathbf{x}(k), \xi(k), u(k)]\} \tag{38}$$

The solution obtained by solving the SDRE numerically, using the approach in [5], is plotted in Fig. 2 where the gasoline-driven mode is shown by red dashed lines with square markers, and the electric-driven mode is shown by blue dashed lines with dot markers. The switching mode and time instants can be identified by the change in color and curve style, as well as by the symbol "+" on the trajectory.

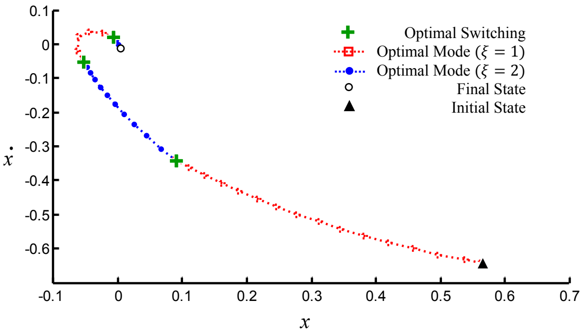

Fig. 2. Optimal state trajectory obtained from SDRE solution.

In the proposed hybrid-ADP solution, the critic network is initialized to satisfy the terminal condition on the costate vector,

$$\boldsymbol{\lambda}(N) = P_f\mathbf{x}(N) = [0 \ 0]^T, \tag{39}$$

while the actor network is initialized to satisfy,

$$u_\xi(k) = -\Big(R_\xi + B_\xi^T B_\xi\Big)^{-1}\Big[B_\xi^T(I + A_\xi)\mathbf{x}(k)\Big] \tag{40}$$

such that (6) holds, given (39). Subsequently, the hybrid-ADP recurrence relationships presented in Section III are used to adapt the critic and the actor networks online, while the actor networks are used to control the power system. Unlike the SDRE approach, the hybrid-ADP only uses online evaluations of the state and immediate cost, as could be obtained from a simulation or the real system. Thus, as shown in Figs. 3-4, the hybrid-ADP updates conducted over several cycles to learn the optimal policy and critic networks, and terminates when the recurrence relationships are satisfied within a desired tolerance.

In this example, the learning rates η and ϵ are both chosen equal to 5×10^{-2}, and the learning steps are $T = 400$ and $M = 100$. The critic (actor) neural networks have two hidden layer with 30 (10) hyperbolic tangent (or sigmoidal) functions. The value of the cost function is evaluated at every cycle, and plotted in Fig. 3, where it is shown to converge to the optimal cost known from the SDRE solution (dashed line). In this simulation, the learning rates η and ϵ are deliberatively chosen greater than $1/(L_1L_2 + L_3)L_4$ in order to accelerate convergence, therefore the cost function does not decrease at every cycle of the algorithm. However, the simulations also show that when this limit is satisfied, the cost function is improved at every cycle of the hybrid-ADP algorithm, and thus the approach could also be applied using state observations obtained from the real system (e.g., during operation), but with a lower convergence rate. The state trajectories obtained by the hybrid-ADP algorithm are shown in a solid line in Fig. 4 for five cycles. When $l = 5$, the state trajectories converge to the optimal state trajectory obtained from the SDRE solution in Fig. 2, and also shown in Fig. 4 for comparison. For the trajectory obtained by ADP, the gasoline-driven mode is shown in red solid lines with diamond markers, the electric-driven mode is shown in blue solid lines with circle markers, and the switching mode and instants can be identified by the change in color and curve style, and "×", along the trajectory.

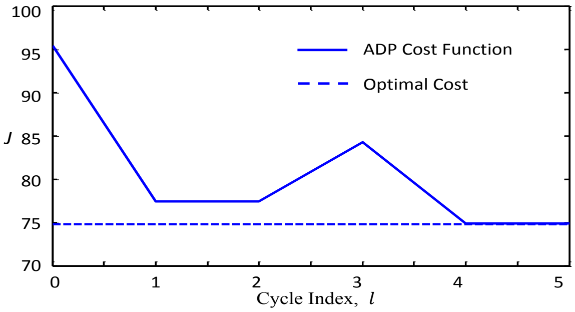

Fig. 3. Hybrid-ADP cost function convergence to the optimal solution obtained by SDRE.

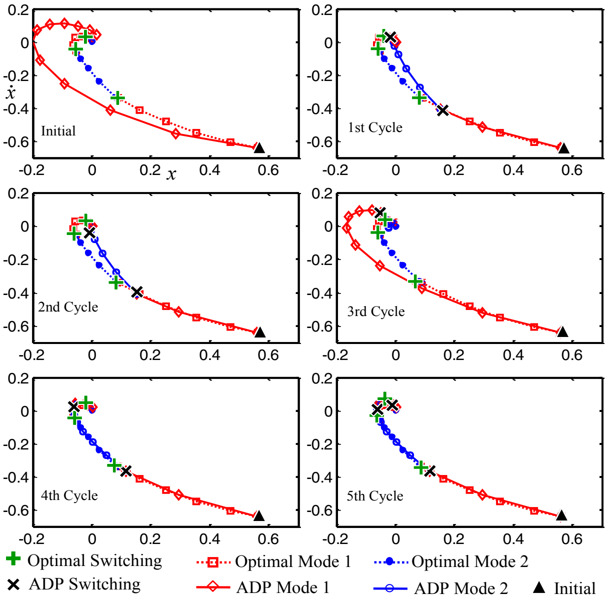

Fig. 4. State trajectory optimization for five cycles of the hybrid-ADP algorithm, and convergence to optimal solution obtained by SDRE.

VI. Summary and Conclusions

This paper presents new recurrence relationships, proof of convergence, and computational complexity for a hybrid-ADP approach applicable to switched hybrid systems that are possibly nonlinear. The results show that the hybrid-ADP algorithm is capable of learning the optimal controller and value function for a switched LQ problem online, using state observations obtained over time from a simulation of the system. The approach is demonstrated on a switched LQ optimal control problem that can be solved numerically using an SDRE off line. Because the algorithm does not rely on the LQ structure of the system dynamics and cost function, hybrid-ADP can be similarly applied to nonlinear (and/or time varying) switched systems, for which SDER solutions are not typically available.

References

[1] Z. Sun and S. Ge, *Switched Linear Systems: Control and Design*, ser. Communications and Control Engineering. Springer, 2005.

[2] C. Seatzu, D. Corona, A. Giua, and A. Bemporad, "Optimal control of continuous-time switched affine systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 726–741, may 2006.

[3] M. Branicky, V. Borkar, and S. Mitter, "A unified framework for hybrid control: model and optimal control theory," *IEEE Transactions on Automatic Control*, vol. 43, no. 1, pp. 31–45, jan 1998.

[4] X. Xu and P. Antsaklis, "Optimal control of switched systems based on parameterization of the switching instants," *Automatic Control, IEEE Transactions on*, vol. 49, no. 1, pp. 2–16, jan. 2004.

[5] P. Riedinger, F. Kratz, C. Iung, and C. Zane, "Linear quadratic optimization for hybrid systems," in *Proceedings of the Conference on Decision and Control*, vol. 3, Phoenix, AZ, 1999, pp. 3059–3064.

[6] L. Rozonoer, "Ls pontryagin maximum principle in the theory of optimum systems. i, ii, iii," *Automatic Remote Control*, vol. 20, pp. 1288–1302, 1959.

[7] W. Zhang, J. Hu, and A. Abate, "Infinite-horizon switched lqr problems in discrete time: A suboptimal algorithm with performance analysis," *IEEE Transactions on Automatic Control*, vol. 57, no. 7, pp. 1815–1821, 2012.

[8] D. Gorges, M. Izak, and S. Liu, "Optimal control and scheduling of switched systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 135–140, 2011.

[9] S. Ferrari and R. Stengel, "On-line adaptive critic flight control," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 5, pp. 777–786, 2004.

[10] ——, "Model-based adaptive critic designs," in *Handbook of Learning and Approximate Dynamic Programming*, J. Si, A. Barto, and W. Powell, Eds. John Wiley and Sons, 2004, vol. 2, p. 65.

[11] M. Grabisch, J. Marichal, R. Mesiar, and E. Pap, *Aggregation Functions (Encyclopedia of Mathematics and its Applications)*, 1st ed. New York, NY, USA: Cambridge University Press, 2009.

[12] T. Kollar and N. Roy, "Trajectory optimization using reinforcement learning for map exploration," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 175–196, 2008.

[13] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vols. I and II*. Belmont, MA: Athena Scientific, 1995.

[14] C. Fox, *An Introduction to the Calculus of Variations*. New York: Dover Publications, Inc., 1987.

[15] R. F. Stengel, *Optimal Control and Estimation*. Dover Publications, Inc., 1986.

[16] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[17] Y. Nesterov, *ntroductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004.

[18] A. E. Niclas Andreasson and M. Patriksson, *An Introduction to Optimization: Foundations and Fundamental Algorithms*. Kluwer Academic Publishers, 2005.

[19] N. Gould, *An introduction to algorithms for continuous optimization*. Oxford University Computing Laboratory Notes, 2006.

[20] J. Boissonnat and S. Oudot, "Provably good sampling and meshing of surfaces," *Graphical Models*, vol. 67, no. 5, pp. 405–451, 2005.

[21] G. Cybenko, "Approximation by superposition of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 359–366, 1989.

[22] J. Ke and X. Liu, "Empirical analysis of optimal hidden neurons in neural network modeling for stock prediction," in *Computational Intelligence and Industrial Application, 2008. PACIIA '08. Pacific-Asia Workshop on*, vol. 2, Dec 2008, pp. 828–832.

[23] S. Ferrari and R. Stengel, "Smooth function approximation using neural networks," *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 24–38, 2005.

[24] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2003.

[25] R. Varga, *Matrix Iterative Analysis*, ser. Springer Series in Computational Mathematics. Springer, 2009.

[26] D. P. Bertsekas, *Convex Analysis and Optimization*. Belmont, MA: Athena Scientific, 2003.

[27] S. Dempe, "A necessary and a sufficient optimality condition for bilevel programming problems," *Optimization*, vol. 25, no. 4, pp. 341–354, 1992.