Neurocomputing 420 (2021) 197–209

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Sub-optimal tracking in switched systems with fixed final time and fixed mode sequence using reinforcement learning

Tohid Sardarmehni [a], Xingyong Song [b,*]

[a] Assistant Professor of Mechanical Engineering, University of Texas Rio Grande Valley, Endinburg, TX 78539, USA
[b] Department of Engineering Technology and Industrial Distribution, Department of Mechanical Engineering, College of Engineering, Texas A&M University, College Station, TX 77843, USA

ARTICLE INFO

Article history:
Received 27 February 2020
Revised 18 July 2020
Accepted 1 September 2020
Available online 12 September 2020
Communicated by Ding Wang

Keywords:
Optimal control
Tracking
Switched systems
Fixed mode sequence

ABSTRACT

Approximate dynamic programming is used to solve optimal tracking problems in switched systems with controlled subsystems and fixed mode sequence. Two feedback control solutions are generated such that the system tracks a desired reference signal, and the optimal switching instants are sought. Simulation results are provided to illustrate the effectiveness of the solutions.

1. Introduction

In many engineering applications, a dynamical system is modeled as a switched system [1–6]. In this study, dynamical systems comprised of a finite number of subsystems/modes are studied. Furthermore, it is assumed that at each time instant, only one subsystem is active [7]. Also, the solutions provided in this study are meant to solve the problems in which the final time is fixed and it is not free. In case the dynamics of the subsystems include a continuously varying control, the subsystems are called *controlled* subsystems. Otherwise, the subsystems are called autonomous.

In switched systems, the sequence of active subsystems is called the *mode sequence*. The mode sequence can be fixed or free. In a fixed mode sequence, the system should activate the subsystems according to a prespecified mode sequence [7]. For instance, consider a manual gearbox in an automobile. Each gear ratio is a mode in this system. For acceleration from rest, the mode sequence is gear 1, gear 2, gear 3, and then gear 4. The role of the driver in this system is assigning the switching times from one mode to another.[1] On the other hand, in switched systems with free mode sequence, the system can activate any arbitrary mode at any time.

The control problem in switched systems can be categorized based on the type of the subsystems and the type of the mode sequence. In switched systems with fixed mode sequence and controlled subsystems, the controller does not decide about the mode sequence. Therefore, the controller only assigns the switching times and the continuous control in the subsystems [7–9]. On the other hand, in switched systems with free mode sequence, at each time instant, the controller assigns the active mode and the continuous control in that active mode [10–16]. Control of switched systems with autonomous subsystems follows the same logic except that lack of continuous control signal in the subsystems eliminates the need to find the continuous control in the active subsystems. In general, control of switched systems is a challenging task due to discontinuous nature of the problem [13].

Optimal control generates control signals that minimize a cost function subjected to state and input constraints. Solving optimal control problems is in fact solving the underlying Hamilton–Jacobi-Bellman (HJB) equation which provides the necessary and sufficient condition for optimality [17]. However, solving the HJB equation explicitly is difficult and in most cases impossible. As a solution, Dynamic Programming (DP) was proposed to solve the optimal control problems through recursive application of the Bellman Principle of Optimality. However, as the order of the system increases, rapid access to memory becomes prohibitive which is known as the *curse of dimensionality* in DP [17]. To remedy the problems in DP, Approximate Dynamic Programming (ADP) was

* Corresponding author.
E-mail addresses: tohid.sardarmehni@utrgv.edu (T. Sardarmehni), songxy@tamu.edu (X. Song).

[1] In other words, driver cannot change the mode sequence. For instance, the driver cannot jump from gear 1 to gear 3 without visiting gear 2.

https://doi.org/10.1016/j.neucom.2020.09.011
0925-2312/

introduced. In summary, ADP methods use function approximators to approximate the optimal cost-to-go, namely critic, and sometimes the optimal policy, namely actor. Then, the ADP methods use iterative techniques to tune the parameters of those function approximators. In other words, ADP methods find the near or approximate optimal control solution instead of the exact optimal control solution. This results in sub-optimality.

Reinforcement Learning (RL) is a common learning strategy in humans and most mammals. In simple words, RL is a learning strategy in which a cognitive agent learns from interacting with the environment [18]. The agent learns to repeat the actions leading to rewards and avoids the actions leading to punishments. The relationship between the ADP and RL can be described as follows. ADP methods use iterative schemes from RL to solve optimal control problems.

In this paper, two ADP solutions are introduced to solve the optimal tracking problem in switched systems with controlled subsystems, fixed mode sequence, and fixed final time. Hence, this study combines some ADP solutions for optimal tracking and optimal switching control to develop new solutions.

In systems with conventional dynamics, i.e., control affine and non-switching dynamics, optimal tracking with ADP was investigated in [19] by using a Single Network Adaptive Critic (SNAC) to approximate the costates, and in [20] by approximating the value functions and augmenting the reference signal and the tracking error in the state vector. Also, in [21,22] model free tracking was investigated.

ADP solutions for optimal control of switched systems with free mode sequence was studied in [10,23,13,12,11,24,16] for optimal regulation, and in [15,3,25,13] for optimal tracking. Optimal control of switched systems with fixed mode sequence was studied in [7] through introducing a transformation to incorporate the switching instants as parameters in the optimal control problem. This transformation was used in [8] to develop an ADP solution to solve the optimal regulation problem. Therefore, [8] provided a Heuristic Dynamic Programming (HDP) solution which trained two networks, namely actor and critic. Also, Non-ADP methods to control switched systems with fixed mode sequence were studied in [9,26,27].

This study explores two possible solutions for optimal tracking in switched systems with fixed mode sequence and fixed final time. The backbone of this study is using the transformation introduced in [7] to incorporate the switching instants as parameters in the optimal control problem. As a result, in both discussed solutions of the present study, there are two levels of control. In the upper level, optimal switching instants are sought through constraint optimization. In the lower level, a feedback optimal control policy is sought through ADP methods. Hence, the contributions of this paper are mainly focused on the lower level control. The key contributions of the paper are as follows[2].

- A SNAC solution is introduced for optimal tracking in the switched systems.
- Based on the SNAC solution, a new method is introduced to reduce the computational burden.

The first solution uses the SNAC structure introduced in [19] to formulate and solve the optimal control problem. As the second solution, to improve the speed of training and reduce the computational burden of the controller, a new method is introduced which uses the immature costates at each time instant to find the optimal policies.

[2] The preliminary results of this research were presented in ASME 2019 Dynamic Systems and Control Conference [28].

Compared to [15,3,25,13], where optimal tracking in switched systems with free mode sequence is performed with ADP, the mode sequence in this paper is fixed. This necessitates a different solution for control. Both of the solutions discussed in this paper are inspired by [19]. However, the methods discussed in this paper deal with switching dynamics which is a more challenging problem than the one discussed in [19]. Compared to [8], the methods discussed in this paper deal with a tracking problem rather than a regulation one. Also, the methods discussed in this paper try to eliminate the dependency of the convergence on the sampling time, as discussed in [8,19]. At last, the second method in this paper works faster than the methods discussed in [8,19] by eliminating the inner loop for training the actor or the costate approximators. In [29], an event triggering system is designed and a thorough stability analysis is conducted. The system in [29] activates piecewise constant controls at each time instant. This is unlike the present study that the control signals in each mode are designed. In [30], a decentralized optimal tacking solution is presented which deals with interconnected fuzzy systems with partially unknown dynamics. Compared to [30], the present work deals with finite horizon problem and the lack of the convergence of the feedback control to zero will not affect our solution.

The rest of the manuscript is organized as follows. In Section 2, optimal control problem formulation and some assumptions are presented. The SNAC solution is presented in Section 3 and the single loop SNAC solution is presented in Section 4. Simulation results are discussed in Section 5. At last, Section 6 concludes the paper.

Remark 1. The practical motivation behind this research is applying the developed method in this paper for optimal control of autonomous systems including wheel loaders in a construction site. The movement of a wheel loader to pick up a load from one point and drop it in another point can be modeled as a switched system with a fixed mode sequence. The interested readers are referred to [31,32] for more information.

2. Problem formulation

Let the dynamics of a switched system be

$$\dot{x}(t) = \bar{f}_v(x(t)) + \bar{g}_v(x(t))u(t),\quad v \in \mathscr{V} = \{1,2,\dots,M\},\ x(0) = x_0 \tag{1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, and t denote the state vector, the input control, and the time, respectively. In (1), the dynamics of the subsystems are shown with smooth functions $\bar{f}_v : \mathbb{R}^n \to \mathbb{R}^n$ and $\bar{g}_v : \mathbb{R}^n \to \mathbb{R}^{n\times m}$. The active mode is shown by sub-index v and the set of all available modes is shown by $\mathscr{V}$. It is further assumed that $\bar{f}_v(0) = 0$, for all modes $v \in \mathscr{V}$. The inclusion of continuous control, i.e., $u(.)$, in (1) shows that the subsystems are controlled subsystems.

To formulate the optimal control problem, consider the cost function as

$$\begin{aligned} J\ (x_0, r_0) &= \tfrac{1}{2}\big(x(t_f) - r(t_f)\big)^T S\big(x(t_f) - r(t_f)\big) \\ &+ \int_{t_0}^{t_f} \tfrac{1}{2}\Big((x(t) - r(t))^T \overline{Q}(x(t) - r(t)) + u(t)^T \overline{R} u(t)\Big)dt \end{aligned} \tag{2}$$

where the initial time and the final time are denoted by t_0 and t_f, respectively. In (2), $S \in \mathbb{R}^{n\times n}$ is a positive semi-definite matrix for penalizing the terminal cost, $\overline{Q} \in \mathbb{R}^{n\times n}$ is the state penalizing matrix which is positive semi-definite, and $\overline{R} \in \mathbb{R}^{m\times m}$ is a positive definite control penalizing matrix. Also, the reference signal is shown by $r \in \mathbb{R}^n$ and the dynamics of the reference signal can be depicted as

$$\dot{r}(t) = \bar{f}_{r_v}(r(t)) \tag{3}$$

where $\bar{f}_{r_v} : \mathbb{R}^n \to \mathbb{R}^n$ is a smooth function.

To include the mode sequence in the dynamics introduced in (1) and the cost function introduced in (2), consider the following transformation [7]

$$t=\begin{cases} t_0+(t_1-t_0)\hat{t} & \text{if } 0\leqslant \hat{t}<1\\ t_1+(t_2-t_1)(\hat{t}-1) & \text{if } 1\leqslant \hat{t}<2\\ \vdots & \vdots\\ t_p+(t_f-t_p)(\hat{t}-p) & \text{if } p\leqslant \hat{t}\leqslant p+1 \end{cases} \tag{4}$$

where $t_1,t_2,\dots t_p$ are the switching times and p is the number of switchings. In (4), t is the actual time domain, and $\hat{t}$ denotes the transformed time domain. Considering the set of switching times as $\Gamma=\{t_0=0,t_1,t_2,\dots t_p\}$, one can consider the mode sequence as $\{v_{t_0},v_{t_1},v_{t_2},\dots,v_{t_p}\}$. Hence, when $t_0\leqslant t<t_1$, mode v_{t_0} is active. Similarly, when $t_1\leqslant t<t_2$, mode v_{t_1} is active. With a similar procedure, one can identify the active modes at each time. The merit of the transformation introduced in (4) is that the switching times $t_1,\dots,t_p$ can be any point in $[t_0,t_f]$. However, in the transformed time domain, switching happens only at $\hat{t}=1,2,\dots,p$. Using (4), one can transform (1) as

$$x'(\hat{t})=\frac{dx}{d\hat{t}}=\frac{dx}{dt}\frac{dt}{d\hat{t}} \tag{5}$$

Considering the known sequence of active modes as $\{v_{t_0},v_{t_1},v_{t_2},\dots,v_{t_p}\}$, (1) transforms as

$$x'(\hat{t})=\begin{cases} \left(\bar{f}_{v_{t_0}}(x)+\bar{g}_{v_{t_0}}(x)u\right)(t_1-t_0) & \text{if } 0\leqslant \hat{t}<1\\ \left(\bar{f}_{v_{t_1}}(x)+\bar{g}_{v_{t_1}}(x)u\right)(t_2-t_1) & \text{if } 1\leqslant \hat{t}<2\\ \vdots & \vdots\\ \left(\bar{f}_{v_{t_p}}(x)+\bar{g}_{v_{t_p}}(x)u\right)(t_f-t_p) & \text{if } p\leqslant \hat{t}\leqslant p+1 \end{cases} \tag{6}$$

where $x\equiv x(\hat{t})$ and $u\equiv u(\hat{t})$ for notational simplicity. With the same procedure, one can transform the cost function in (2) as

$$\begin{aligned} J(x_0,r_0)&=\tfrac{1}{2}(x(p+1)-r(p+1))^TS(x(p+1)-r(p+1))\\ &+\tfrac{1}{2}\int_0^1\left((x-r)^T\overline{Q}(t_1-t_0)(x-r)+u^T\overline{R}(t_1-t_0)u\right)d\hat{t}\\ &+\tfrac{1}{2}\int_1^2\left((x-r)^T\overline{Q}(t_2-t_1)(x-r)+u^T\overline{R}(t_2-t_1)u\right)d\hat{t}\\ &\quad\vdots\qquad\qquad\vdots\\ &+\tfrac{1}{2}\int_p^{p+1}\left((x-r)^T\overline{Q}(t_f-t_p)(x-r)+u^T\overline{R}(t_f-t_p)u\right)d\hat{t} \end{aligned} \tag{7}$$

where $x\equiv x(\hat{t}),r\equiv r(\hat{t})$, and $u\equiv u(\hat{t})$ for notational simplicity.

Remark 2. As one can see from (7), the transformed cost function is a function of x_0,r_0, and the switching times, i.e., Γ. Hence, $J(.,.,.)=J(\Gamma,x_0,r_0)$. This is an important observation which will be used in the subsequent sections to solve the optimal control problem.

Using the Euler integration method, by choosing a small sample time $\delta\hat{t}$, one can discretize (6) as

$$x_{\hat{k}+1}=\begin{cases} f_{v_{t_0}}(x_{\hat{k}})+g_{v_{t_0}}(x_{\hat{k}})u_{\hat{k}} & \text{if } 0\leqslant \hat{k}\delta\hat{t}<1\\ f_{v_{t_1}}(x_{\hat{k}})+g_{v_{t_1}}(x_{\hat{k}})u_{\hat{k}} & \text{if } 1\leqslant \hat{k}\delta\hat{t}<2\\ \vdots & \\ f_{v_{t_p}}(x_{\hat{k}})+g_{v_{t_p}}(x_{\hat{k}})u_{\hat{k}} & \text{if } p\leqslant \hat{k}\delta\hat{t}\leqslant p+1 \end{cases} \tag{8}$$

where

$$\begin{aligned} f_{v_{t_0}}(x_{\hat{k}})&=x(\hat{t})+\bar{f}_{v_{t_0}}(x(\hat{t}))(t_1-t_0)\delta\hat{t}\\ g_{v_{t_0}}(x_{\hat{k}})&=\bar{g}_{v_{t_0}}(x(\hat{t}))(t_1-t_0)\delta\hat{t}\\ f_{v_{t_1}}(x_{\hat{k}})&=x(\hat{t})+\bar{f}_{v_{t_1}}(x(\hat{t}))(t_2-t_1)\delta\hat{t}\\ g_{v_{t_1}}(x_{\hat{k}})&=\bar{g}_{v_{t_1}}(x(\hat{t}))(t_2-t_1)\delta\hat{t} \end{aligned}$$

Similarly, one can define $f_{v_{t_i}}(.)$ and $g_{v_{t_i}}(.)$ for the rest of $i\leqslant p$. In (8), $\hat{k}\in[0,N']$ is the discrete time index where $N'=\frac{p+1}{\delta\hat{t}}$ [8]. Considering Remark 2, one can discretize (7) as

$$\begin{aligned} J(\Gamma,x_0,r_0)&=\tfrac{1}{2}(x_{N'}-r_{N'})^TS(x_{N'}-r_{N'})\\ &+\sum_{\hat{k}=0}^{\hat{k}=1/\delta\hat{t}}\tfrac{1}{2}(x_{\hat{k}}-r_{\hat{k}})^TQ_1(x_{\hat{k}}-r_{\hat{k}})+\tfrac{1}{2}u_{\hat{k}}^TR_1u_{\hat{k}}\\ &+\sum_{\hat{k}=1/\delta\hat{t}}^{\hat{k}=2/\delta\hat{t}}\tfrac{1}{2}(x_{\hat{k}}-r_{\hat{k}})^TQ_2(x_{\hat{k}}-r_{\hat{k}})+\tfrac{1}{2}u_{\hat{k}}^TR_2u_{\hat{k}}\\ &\quad\vdots\qquad\qquad\vdots\\ &+\sum_{\hat{k}=p/\delta\hat{t}}^{\hat{k}=N'}\tfrac{1}{2}(x_{\hat{k}}-r_{\hat{k}})^TQ_p(x_{\hat{k}}-r_{\hat{k}})+\tfrac{1}{2}u_{\hat{k}}^TR_pu_{\hat{k}} \end{aligned} \tag{9}$$

where

$$\begin{aligned} Q_1&=\overline{Q}(t_1-t_0)\delta\hat{t},\quad R_1=\overline{R}(t_1-t_0)\delta\hat{t}\\ Q_2&=\overline{Q}(t_2-t_1)\delta\hat{t},\quad R_2=\overline{R}(t_2-t_1)\delta\hat{t}\\ Q_p&=\overline{Q}(t_f-t_p)\delta\hat{t},\quad R_p=\overline{R}(t_f-t_p)\delta\hat{t} \end{aligned}$$

Considering (9), one can define the cost-to-go at each time instant $\hat{k}$ as

$$\begin{aligned} J(\Gamma,x_{\hat{k}},r_{\hat{k}})&=\tfrac{1}{2}(x_{N'}-r_{N'})^TS(x_{N'}-r_{N'})\\ &+\tfrac{1}{2}\sum_{\bar{k}=\hat{k}}^{\frac{\lfloor\hat{k}\delta\hat{t}\rfloor+1}{\delta\hat{t}}}(x_{\bar{k}}-r_{\bar{k}})^TQ_{\lfloor\hat{k}\delta\hat{t}\rfloor+1}(x_{\bar{k}}-r_{\bar{k}})+u_{\bar{k}}^TR_{\lfloor\hat{k}\delta\hat{t}\rfloor+1}u_{\bar{k}}\\ &+\tfrac{1}{2}\sum_{j=\lfloor\hat{k}\delta\hat{t}\rfloor+2}^{p}\sum_{\bar{k}=j/\delta\hat{t}}^{(j+1)/\delta\hat{t}}(x_{\bar{k}}-r_{\bar{k}})^TQ_j(x_{\bar{k}}-r_{\bar{k}})+u_{\bar{k}}^TR_ju_{\bar{k}} \end{aligned} \tag{10}$$

where $\lfloor.\rfloor$ denotes the floor function. Using (10), one can define the minimum cost-to-go, i.e., value function, as

$$\begin{aligned} &V\left(\Gamma,x_{\hat{k}},r_{\hat{k}},\hat{k}\right)\equiv V_{\hat{k}}(\Gamma,x_{\hat{k}},r_{\hat{k}})\\ &=\min_{u(.)}\Bigg(\tfrac{1}{2}(x_{N'}-r_{N'})^TS(x_{N'}-r_{N'})\\ &+\tfrac{1}{2}\sum_{\bar{k}=\hat{k}}^{\frac{\lfloor\hat{k}\delta\hat{t}\rfloor+1}{\delta\hat{t}}}(x_{\bar{k}}-r_{\bar{k}})^TQ_{\lfloor\hat{k}\delta\hat{t}\rfloor+1}(x_{\bar{k}}-r_{\bar{k}})+u_{\bar{k}}^TR_{\lfloor\hat{k}\delta\hat{t}\rfloor+1}u_{\bar{k}}\\ &+\tfrac{1}{2}\sum_{j=\lfloor\hat{k}\delta\hat{t}\rfloor+2}^{p}\sum_{\bar{k}=j/\delta\hat{t}}^{(j+1)/\delta\hat{t}}(x_{\bar{k}}-r_{\bar{k}})^TQ_j(x_{\bar{k}}-r_{\bar{k}})+u_{\bar{k}}^TR_ju_{\bar{k}}\Bigg) \end{aligned} \tag{11}$$

Considering the time step $\hat{k}$ to $\hat{k}+1$, one has

$$\begin{aligned} V_{\hat{k}}(\Gamma,x_{\hat{k}},r_{\hat{k}})&=\min_{u(.)}\Bigg(\tfrac{1}{2}(x_{N'}-r_{N'})^TS(x_{N'}-r_{N'})\\ &+\tfrac{1}{2}(x_{\hat{k}}-r_{\hat{k}})^TQ_{\lfloor\hat{k}\delta\hat{t}\rfloor+1}(x_{\hat{k}}-r_{\hat{k}})+\tfrac{1}{2}u_{\hat{k}}^TR_{\lfloor\hat{k}\delta\hat{t}\rfloor+1}u_{\hat{k}}\\ &+\tfrac{1}{2}\sum_{\bar{k}=\hat{k}+1}^{\frac{\lfloor\hat{k}\delta\hat{t}\rfloor+1}{\delta\hat{t}}}(x_{\bar{k}}-r_{\bar{k}})^TQ_{\lfloor\hat{k}\delta\hat{t}\rfloor+1}(x_{\bar{k}}-r_{\bar{k}})+u_{\bar{k}}^TR_{\lfloor\hat{k}\delta\hat{t}\rfloor+1}u_{\bar{k}}\\ &+\tfrac{1}{2}\sum_{j=\lfloor\hat{k}\delta\hat{t}\rfloor+2}^{p}\sum_{\bar{k}=j/\delta\hat{t}}^{(j+1)/\delta\hat{t}}(x_{\bar{k}}-r_{\bar{k}})^TQ_j(x_{\bar{k}}-r_{\bar{k}})+u_{\bar{k}}^TR_ju_{\bar{k}}\Bigg) \end{aligned} \tag{12}$$

Letting $\nu_{\hat{k}} = \lfloor \hat{k}\delta\hat{t} \rfloor + 1$, through some algebraic manipulations one has

$$V_{\hat{k}}(\Gamma, x_{\hat{k}}, r_{\hat{k}}) = \min_{u(.)} \Big(\tfrac{1}{2}(x_{\hat{k}} - r_{\hat{k}})^T Q_{\nu_{\hat{k}}}(x_{\hat{k}} - r_{\hat{k}}) + \tfrac{1}{2} u_{\hat{k}}^T R_{\nu_{\hat{k}}} u_{\hat{k}} + V_{\hat{k}+1}(\Gamma, x_{\hat{k}+1}, r_{\hat{k}+1}) \Big) \tag{13}$$

Eq. (13) is the representation of the Bellman principle of optimality which simply states that the minimum cost of going from time $\hat{k} \to N'$ is the sum of minimum cost of going from $\hat{k} \to \hat{k}+1$ and the minimum cost of going from $\hat{k}+1 \to N'$. This equation is the backbone of the formulations which is also known as the Hamilton–Jacobi-Bellman (HJB) equation. As mentioned before, solutions of the HJB equation provide the necessary and sufficient condition for optimality [17]. Given (13), one can define the optimal policy as

$$u_{\hat{k}}^*(\Gamma, x_{\hat{k}}, r_{\hat{k}}) = \operatorname*{argmin}_{u(.)} \Big(\tfrac{1}{2}(x_{\hat{k}} - r_{\hat{k}})^T Q_{\nu_{\hat{k}}}(x_{\hat{k}} - r_{\hat{k}}) + \tfrac{1}{2} u_{\hat{k}}^T R_{\nu_{\hat{k}}} u_{\hat{k}} + V_{\hat{k}+1}(\Gamma, x_{\hat{k}+1}, r_{\hat{k}+1}) \Big) \tag{14}$$

In order to continue the solution, one assumes that the value functions are smooth. In the case of quadratic cost functions as in (2), it is straightforward to see that

$$u_{\hat{k}}^*(\Gamma, x_{\hat{k}}, r_{\hat{k}}) = -R_{\nu_{\hat{k}}}^{-1} g_{\nu_{\hat{k}}}^T(x_{\hat{k}}) \frac{\partial V_{\hat{k}+1}(\Gamma, x_{\hat{k}+1}, r_{\hat{k}+1})}{\partial x_{\hat{k}}}|_{x_{\hat{k}+1}} \tag{15}$$

In (15), if $V_{\hat{k}+1}(\Gamma, x_{\hat{k}+1}, r_{\hat{k}+1})$ is known, one can easily find the optimal policy.

By defining the costates as $\lambda_{\hat{k}}(\Gamma, x_{\hat{k}}, r_{\hat{k}}) = \frac{\partial V_{\hat{k}}(\Gamma, x_{\hat{k}}, r_{\hat{k}})}{\partial x_{\hat{k}}}$, one can see that the otpimal policy can be defined as

$$u_{\hat{k}}^*(\Gamma, x_{\hat{k}}, r_{\hat{k}}) = -R_{\nu_{\hat{k}}}^{-1} g_{\nu_{\hat{k}}}^T(x_{\hat{k}}) \lambda_{\hat{k}+1}(\Gamma, x_{\hat{k}+1}, r_{\hat{k}+1}) \tag{16}$$

Similar to (15), in case $\lambda_{\hat{k}+1}$ is known, one can find the optimal policy from (16). Considering (13) and time step $\hat{k} = N'$, by taking the gradient with respect to $x_{N'}$ from both sides one has

$$\lambda_{N'}(\Gamma, x_{N'}, r_{N'}) = S(x_{N'} - r_{N'}) \tag{17}$$

Also, considering (13) for $\hat{k} < N'$, by taking the gradient with respect to $x_{\hat{k}}$ one has

$$\lambda_{\hat{k}}(\Gamma, x_{\hat{k}}, r_{\hat{k}}) = Q_{\nu_{\hat{k}}}(x_{\hat{k}} - r_{\hat{k}}) + \left(\frac{\partial x_{\hat{k}+1}}{\partial x_{\hat{k}}} \right)^T \lambda_{\hat{k}+1}\left(\Gamma, x_{\hat{k}+1}^*, r_{\hat{k}+1}\right) \tag{18}$$

where $x_{\hat{k}+1}^*$ means x is propagated from $x_{\hat{k}}$ to $x_{\hat{k}+1}$ along policy $u_{\hat{k}}^*$. Evaluating (18) at $\hat{k}+1$ leads [19]

$$\lambda_{\hat{k}+1}\left(\Gamma, x_{\hat{k}+1}^*, r_{\hat{k}+1}\right) = Q_{\nu_{\hat{k}+1}}\left(x_{\hat{k}+1}^* - r_{\hat{k}+1}\right) + \left(\frac{\partial x_{\hat{k}+2}}{\partial x_{\hat{k}+1}} \right)^T \lambda_{\hat{k}+2}\left(\Gamma, x_{\hat{k}+2}^*, r_{\hat{k}+2}\right) \tag{19}$$

Using (17) and (19), one can go backward in time and find $\lambda_{\hat{k}+1}$ for all times and then find the optimal policies from (16).

As seen in the optimal control problem formulation, by using the transformation introduced in (6), one treated the switching times as parameters. This parametrization of the switching times necessitates two levels of optimization. In the upper level, the switching times are sought. In the lower level, one finds the optimal policies as (16).

3. SNAC solution

In this section, a SNAC solution introduced in [19] is adapted to solve the optimal tracking in switched systems. The main idea is using function approximators to approximate/predict the costates. Due to smoothness assumption of the value functions, one can use neural networks to uniformly approximate the costates [33]. Also, considering the Weierstrass Approximation Theorem [34], one can use linear in parameter neural networks with polynomial basis functions to approximate the costates to any degree of precision.

For approximation, neural networks can be used to approximate $\lambda_{\hat{k}+1}(\Gamma, x_{\hat{k}+1}, r_{\hat{k}+1})$ from $(\Gamma, x_{\hat{k}}, r_{\hat{k}})$. Consider the exact costate at discrete time index $\hat{k}+1$ as

$$\lambda_{\hat{k}+1}(\Gamma, x_{\hat{k}+1}, r_{\hat{k}+1}) = W_{\hat{k}}^{*T} \phi(\Gamma, x_{\hat{k}}, r_{\hat{k}}) + \varepsilon_{\hat{k}}^*(\Gamma, x_{\hat{k}}, r_{\hat{k}}) \tag{20}$$

where $W_{\hat{k}}^* \in \mathbb{R}^{m_\lambda \times n}$ is a weight vector and $\phi : \mathbb{R}^p \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{m_\lambda}$ is a vector of m_λ number of linearly independent polynomial basis functions (neurons). Similar to the previous sections, the dependence of the parameters/functions to discrete time index is shown with a sub-index $\hat{k}$ in (20). Let the approximate costates be

$$\widehat{\lambda}_{\hat{k}+1}(\Gamma, x_{\hat{k}+1}, r_{\hat{k}+1}) = \widehat{W}_{\hat{k}}^T \phi(\Gamma, x_{\hat{k}}, r_{\hat{k}}) \tag{21}$$

where $\widehat{W}_{\hat{k}} \in \mathbb{R}^{m_\lambda \times n}$ is a tunable weight vector adjusted through training. Considering (16), it is straightforward to find the approximate optimal policy with the approximate optimal costates as

$$\begin{aligned} \widehat{u}_{\hat{k}}(\Gamma, x_{\hat{k}}, r_{\hat{k}}) &= -R_{\nu_{\hat{k}}}^{-1} g_{\nu_{\hat{k}}}^T(x_{\hat{k}}) \widehat{\lambda}_{\hat{k}+1}(\Gamma, x_{\hat{k}+1}, r_{\hat{k}+1}) \\ &= -R_{\nu_{\hat{k}}}^{-1} g_{\nu_{\hat{k}}}^T(x_{\hat{k}}) \widehat{W}_{\hat{k}}^T \phi(\Gamma, x_{\hat{k}}, r_{\hat{k}}) \end{aligned} \tag{22}$$

For finding the optimal costates, one can go backward in time as follows. Considering $\hat{k} = N'$, by substituting for $\lambda_{N'}$ from (21) in (17) one has

$$\widehat{W}_{N'-1}^T \phi(\Gamma, x_{N'-1}, r_{N'-1}) = S(x_{N'} - r_{N'}) \tag{23}$$

Also, by substituting $\lambda_{\hat{k}}$ from (21) in (19) for the rest of $\hat{k}$, one has

$$\widehat{W}_{\hat{k}}^T \phi(\Gamma, x_{\hat{k}}, r_{\hat{k}}) = Q_{\nu_{\hat{k}+1}}\left(x_{\hat{k}+1}^* - r_{\hat{k}+1}\right) + \left(\frac{\partial x_{\hat{k}+2}}{\partial x_{\hat{k}+1}} \right)^T \widehat{W}_{\hat{k}+1}^T \phi\left(\Gamma, x_{\hat{k}+1}^*, r_{\hat{k}+1}\right) \tag{24}$$

In (23), one notes that the states on the right-hand side are propagated along policy $\widehat{u}_{N'-1}$ from $x_{N'-1}$ to $x_{N'}$. As one can see from (22), policy $\widehat{u}_{N'-1}$ requires $\widehat{\lambda}_{N'}$ which is unknown. Hence, $\widehat{\lambda}_{N'}$ is present on both sides of (23). Similarly, in (24), one notices a propagation from $x_{\hat{k}}$ to $x_{\hat{k}+1}$ and the requirement of $\widehat{\lambda}_{\hat{k}+1}$. With the same discussion about the presence of $\widehat{\lambda}_{N'}$ on both sides of (23), one can see the presence of $\widehat{\lambda}_{\hat{k}+1}$ on both sides of (24).

In order to solve this problem, an iterative solution was introduced in [19] as follows. Let the iteration index be denoted by i. Considering the state propagated from $x_{\hat{k}}$ to $x_{\hat{k}+1}$ using policy $\widehat{u}_{\hat{k}}^i(.)$ as $x_{\hat{k}+1}^i$, one has

$$\lambda_{N'}^{i+1}(\Gamma, x_{N'}, r_{N'}) = S(x_{N'}^i - r_{N'}) \tag{25}$$

Substituting from (21) in (25) leads

$$\widehat{W}_{N'-1}^{i+1^T} \phi(\Gamma, x_{N'-1}, r_{N'-1}) = S(x_{N'}^i - r_{N'}) \tag{26}$$

Considering (19), for $\hat{k} < N'$ one has

$$\lambda_{\hat{k}+1}^{i+1}(\Gamma, x_{\hat{k}+1}, r_{\hat{k}+1}) = Q_{\nu_{\hat{k}+1}}\left(x_{\hat{k}+1}^i - r_{\hat{k}+1}\right) + \left(\frac{\partial x_{\hat{k}+2}}{\partial x_{\hat{k}+1}} \right)^T \lambda_{\hat{k}+2}\left(\Gamma, x_{\hat{k}+2}^i, r_{\hat{k}+2}\right) \tag{27}$$

Similar to (26), substituting from (21) leads

$$\widehat{W}_{\hat{k}}^{i+1^T}\phi(\Gamma, x_{\hat{k}}, r_{\hat{k}}) = Q_{v_{\hat{k}+1}}\left(x_{\hat{k}+1}^{i} - r_{\hat{k}+1}\right) + \left(\frac{\partial x_{\hat{k}+2}}{\partial x_{\hat{k}+1}}\right)^T \widehat{W}_{\hat{k}+1}^{T}\phi\left(\Gamma, x_{\hat{k}+1}^{i}, r_{\hat{k}+1}\right) \tag{28}$$

In (26) and (28), the inner loop starts with a random initial guess for $\lambda_{\hat{k}+1}^{0}$, i.e., $\widehat{W}_{\hat{k}}^{0}$. With $\widehat{W}_{\hat{k}}^{0}$, one finds u^0 and uses this policy to propagate the states. Using (26) or (28), one finds $\widehat{W}_{\hat{k}}^{1}$. This process continues until the weights converge. After calculating $\widehat{W}_{\hat{k}}$, one can go backward in time to find the rest of the costates. When the training is concluded, the stored optimal costates are used for online control without re-training. This training process is summarized in Algorithm 1.

Algorithm 1: **SNAC Solution**

step 1: Let $N' =$ (no. of switching +1) $/\delta\hat{t}$ where $\delta\hat{t}$ is a small selected sampling time. Initialize $\widehat{W}_{\hat{k}}^{0}$ for all positive integer $\hat{k} \le N'$. Select η random training samples $x^{[l]} \in \Omega$, $r^{[l]} \in \Omega$, and $\Gamma^{[l]}$ where $l \in \{1, 2, ..., \eta\}$, and $\Gamma^{[l]} = \{t_1^{[l]} \le t_2^{[l]} \le \cdots \le t_p^{[l]}\}$. Set $\hat{k} = N'$, and select a small positive number γ as the convergence tolerance.

while $\hat{k} \ge 1$ **do**

  step 2: Set $\hat{k} = \hat{k} - 1$ and $i = 0$

  while $\|\widehat{W}_{\hat{k}}^{i+1} - \widehat{W}_{\hat{k}}^{i}\| > \gamma$ **do**

    step 3: Calculate $\hat{\lambda}_{\hat{k}+1}^{i} = \widehat{W}_{\hat{k}}^{i^T}\phi(\Gamma^{[l]}, x^{[l]}, r^{[l]})$, $\forall l < \eta$.

    step 4: With $\hat{\lambda}_{\hat{k}+1}^{i}$ find $u_{\hat{k}}^{i}$ from (22).

    step 5: With $u_{\hat{k}}^{i}$ find $x_{\hat{k}+1}^{i}$. Also find $r_{\hat{k}+1}$.

    step 6: Find $\widehat{W}_{\hat{k}}^{i+1}$ from (26) or (28) via least squares.

    step 7: Set $i = i + 1$.

  end

  step 8: Set $\widehat{W}_{\hat{k}} = \widehat{W}_{\hat{k}}^{i+1}$.

end

Remark 3. The convergence of the inner loop in steps 3–7 of Algorithm 1 was studied in Theorem 1 of [19] for systems with conventional dynamics. The convergence of inner loop in Algorithm 1 is studied in Theorem 1.

Theorem 1. *Considering the iterative solution illustrated by (25) and (27), there exists a control penalizing matrix, i.e., R^*, such that for any control penalizing matrix R that $\|R\| \geqslant \|R^*\|$, the iterations shown in (25) and (27) converge to the optimal solution.*

Proof. Please see Appendix A.

Remark 4. The presented control in this paper deals with finite horizon. As mentioned in (11) and (18), the value functions and the costates are functions of the time, $\hat{k}$. The time dependence of the costates (or the value function) results in different weights at different times. Therefore, the history the weights of the neural network does not need to show a convergent behavior. This can also be verified from the presentation of Algorithm 1 where the convergence is required in the inner loop and not the outer loop. In Algorithm 1, the inner loop deals with solving Eqs. (26) and (28) iteratively. However, the outer loop only goes backward in the time to perform the recursions.

Remark 5. Once the training is concluded, one needs to find the optimal switching times from the costates for a selected initial condition $x_0 \in \Omega$. Three methods are suggested below to find the optimal switching times from the optimal costates.

- **– Method 1:** propagating the states analytically along the optimal policy by treating switching time as a parameter and finding the optimal cost-to-go from the cost function. Once done, one can use constrained minimization methods to find switching times.
- **– Method 2:** integrating the costate analytically to find the value function. Similar to finding the velocity field from potential flow in fluid mechanics, one can integrate the costates analytically to find the value functions. The convenient feature of this method is that the analytical solutions provide the optimal value function $\forall x_0 \in \Omega$. In order words, one does not need to integrate the costates again when the initial condition is changed. However, this method becomes very complicated as the order of system increases. Therefore, Method 2 is more suitable for systems with low order dynamics.
- **– Method 3:** propagating the states along all possible switching times and find the optimal cost to go for all possible switching times. Once done, choose the switching times which lead to the minimum value function. Method 3 is similar to the forward dynamic programming method and when the number of switching increases, performing this method might become time-consuming.

4. Single loop SNAC solution

In this section, a new algorithm is introduced which uses immature policies to derive the optimal control solution. In Algorithm 1, an inner loop was introduced in step 3 to step 7. The purpose of this inner loop is finding the parameters of the function approximator to predict the costates. As mentioned in Algorithm 1, steps 3 to 7 are repeated until the iterations converge, i.e., the $\|\widehat{W}_{\hat{k}}^{i+1} - \widehat{W}_{\hat{k}}^{i}\|$ becomes smaller than a selected convergence tolerance γ. Therefore, neglecting the inner loop results in generation of immature policies. In the Algorithm 2, the inner loop in step 3 to step 7 of Algorithm 1 is eliminated. By eliminating the inner loop, the overall training process can be performed faster. In what follows, we discuss the effect of eliminating the inner loop in Algorithm 1.

Theorem 2. *The error between the states propagated along policies generated by Algorithm 1 and Algorithm 2 is bounded.*

Proof. Please see Appendix B.

Algorithm 2: **Single Loop SNAC Solution**

step 1: Let $N' =$ (no. of switching +1) $/\delta\hat{t}$ where $\delta\hat{t}$ is a small selected sampling time. Initialize $\widehat{W}_{\hat{k}}^{0}$ for all positive integer $\hat{k} \le N'$. Select η random training samples $x^{[l]} \in \Omega$, $r^{[l]} \in \Omega$, and $\Gamma^{[l]}$ where $l \in \{1, 2, ..., \eta\}$ and $\Gamma^{[l]} = \{t_1^{[l]} \le t_2^{[l]} \le \cdots \le t_p^{[l]}\}$. Set $\hat{k} = N'$.

while $\hat{k} \ge 1$ **do**

  step 2: Set $\hat{k} = \hat{k} - 1$ and $i = 0$

  step 3: Calculate $\hat{\lambda}_{\hat{k}+1}^{i} = \widehat{W}_{\hat{k}}^{i^T}\phi(\Gamma^{[l]}, x^{[l]}, r^{[l]})$, $\forall l < \eta$.

  step 4: With $\hat{\lambda}_{\hat{k}+1}^{i}$ find $u_{\hat{k}}^{i}$ from (22).

  step 5: With $u_{\hat{k}}^{i}$ find $x_{\hat{k}+1}^{i}$. Also find $r_{\hat{k}+1}$.

  step 6: Find $\widehat{W}_{\hat{k}}^{i+1}$ from (26) or (28) via least squares.

  step 7: Set $\widehat{W}_{\hat{k}} = \widehat{W}_{\hat{k}}^{i+1}$.

  step 8: Set $i = i + 1$.

end

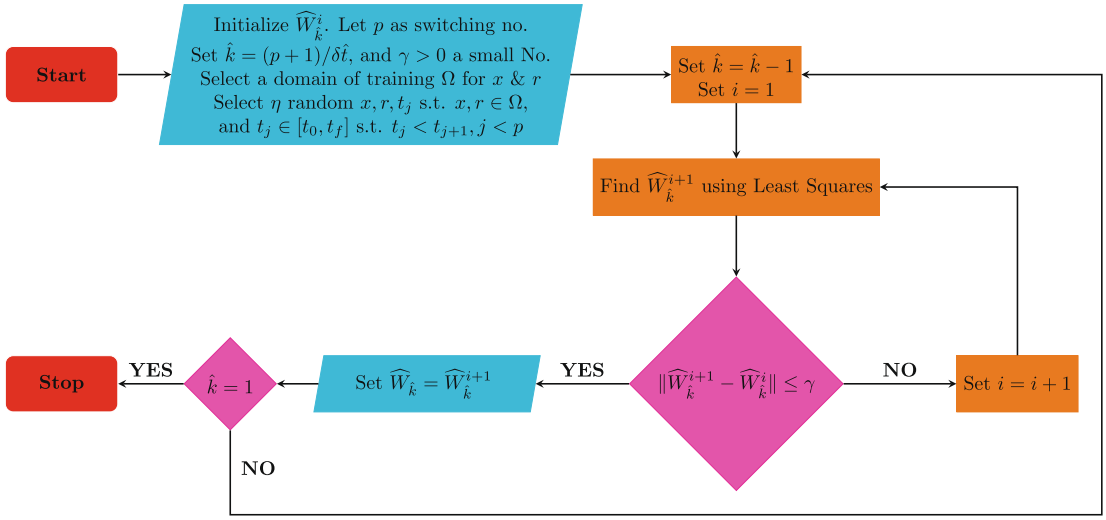


Fig. 1. Flowchart of Algorithm 1.

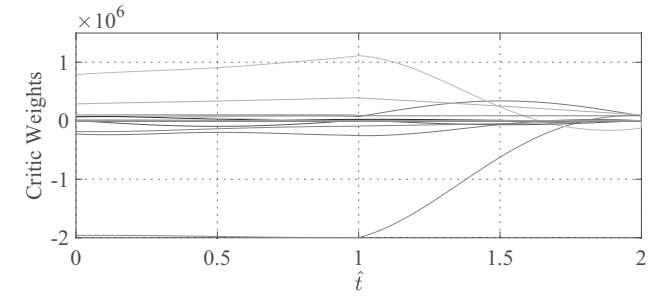


Fig. 2. The history of the weights of the neural network to approximate costates. Training was conducted by Algorithm 1. The switching time is denoted at $\hat{t}=1$.

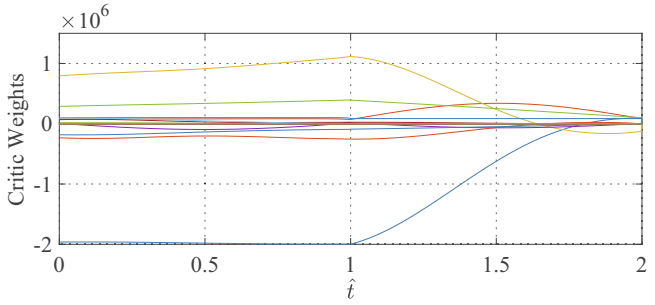


Fig. 3. The history of the weights of the neural network trained with Algorithm 2. Similar to Fig. 2, the jump at $\hat{t}=1$ shows the switching at this time.

5. Numerical simulations

Consider a switched system comprised of 3 modes as follows.

$$\dot{x}(t)=\begin{cases} x_2(t) \\ (1-x_1^2(t))x_2(t)-x_1(t)+u(t) \end{cases} \tag{29}$$

$$\dot{x}(t)=\begin{cases} x_2(t) \\ 2x_1-x_2+u(t) \end{cases} \tag{30}$$

$$\dot{x}(t)=\begin{cases} x_2(t) \\ x_1^2-x_2^2+u(t) \end{cases} \tag{31}$$

Eqs. (29)–(31) indicate mode 1, mode 2, and mode 3, respectively. Also, consider two different mode sequences as $\Gamma^1=\{mode\ 1,\ mode\ 2\}$, and $\Gamma^2=\{mode\ 1,\ mode\ 2,\ mode\ 3\}$. It can be seen that Γ^1has only one switching, and Γ^2 has two switchings. At last, the dynamics of the reference signal can be represented as a function of time or a function of states. The dynamics of the time-based reference signal was selected as

$$\begin{aligned} \dot{r}_1(t) &= \sin(\pi t) \\ \dot{r}_2(t) &= \pi\cos(\pi t) \end{aligned} \tag{32}$$

Also, the state-based reference signal was selected as

$$\begin{aligned} \dot{r}_1(t) &= -r_1(t) \\ \dot{r}_2(t) &= -r_2(t) \end{aligned} \tag{33}$$

5.1. One switching

5.1.1. Time-based reference signal

Considering the mode sequence as Γ^1, it is desired to find the optimal switching time t_1, and the optimal policies such that the system tracks (32). Selecting $S=\mathrm{diag}\left(10^5,10^5\right)$, $\overline{Q}=\mathrm{diag}\left(10^5,10^7\right), \overline{R}=10^3$, the discretization sample time $\delta\hat{t}=0.001$, one starts the simulations.

For training, 1000 random training patterns were generated in $\Omega=\{(t_1,x_1,x_2)|\ 0\leqslant t_1\leqslant 3,\ |x_1|\leqslant 4,\ |x_2|\leqslant 4\}$. To approximate the costates, a linear-in-parameter neural network with basis functions comprised of polynomials with all possible combinations of t_1, x_1, and x_2 up to the power of 3 without repetition was selected. The training for finding the optimal costates was concluded in 22.62 (s) using Algorithm 1 and only 7.52 (s) using Algorithm 2. By analytically integrating the optimal costates at $\hat{t}=0$, the value functions were found. Afterward, the value functions were evaluated at $x_0=[1,-0.5]^T$ to get the value functions with only one variable as t_1. Then, the value functions were minimized with respect to the switching time, i.e., t_1. Using the optimal costates trained by Algorithms 1 and 2, the optimal switching time was found at $t_1=2.655$ (s) and $t_1=2.656$ (s), respectively (see Fig. 1).

The histories of the weights of the neural networks used to approximate the costates with Algorithms 1 and 2 are illustrated in Figs. 2 and 3. As one can see from Figs. 2 and 3, the behavior of the weights before and after the switching time, i.e., $\hat{t}=1$, is different. In fact a jumping behavior can be detected at the switching

time in both Figs. 2 and 3. Such behavior is similar to what was reported in [8].

At last, the performance of the controllers trained by Algorithms 1 and 2 are compared in Fig. 4. As one can see the performance of the controllers are very similar which shows the effectiveness of the methods discussed in this paper.

5.1.2. State-based reference signal

Similar to the previous section, consider the mode sequence as Γ^1. It is desired to find the optimal switching time t_1, and the optimal policies such that the system tracks (33). Selecting the state and control penalizing matrices, and the discretization sample time the same as the previous example, one starts the solution.

In this example, since the reference signal is a function of the the states, the reference signal should be included in the costates. The domain of training was chosen as $\Omega = \{(t_1, x_1, x_2, r_1, r_2) | 0 < t_1 < 3, |x_1| \leqslant 4, |x_2| \leqslant 4, |r_1| \leqslant 4, |r_2| \leqslant 4\}$. In order to approximate the costates, a linear in parameter neural network with polynomial basis functions comprised of t_1, x_1, x_2, r_1, and r_2 up to the power of 4 without repetition was selected. Using Algorithms 1 and 2, the training process concluded in 91.53 (s) and 30.94 (s), respectively. The history of the weights of the neural networks used to approximate the costates are shown in Figs. 5 and 6.

Once the training concluded, the optimal costates were integrated analytically to find the optimal value functions. Evaluating the value functions at $x_0 = [1, -0.5]^T$ and $r_0 = [1, -1]^T$, the optimal switching time was sought as $t_1 = 2.50$ for the controller trained by Algorithm 1. The optimal switching time for the controller trained by Algorithm 2 was $t_1 = 2.433$.

The history of the state trajectories using controllers trained by Algorithms 1 and 2 are compared in Fig. 7. As one can see, both controllers could effectively track the reference signal and the performance of the controllers are close to each other.

5.2. Two switching

5.2.1. Time-based reference signal

Let the mode sequence be as Γ^2. It is desired to find the optimal switching times t_1 and t_2 such that the cost function in (2) is minimized and the system tracks the reference signal denoted in (32). The penalizing parameters of the cost function are selected as $S = \text{diag}\left(10^5, 10^5\right), \overline{Q} = \text{diag}\left(10^5, 10^7\right), \overline{R} = 10^3$. Also, the discretization sample time was selected as $\delta\hat{t} = 0.001$.

For training, 1000 random training patterns were generated in $\Omega = \{(t_1, t_2, x_1, x_2) | 0 \leqslant t_1 \leqslant t_2 \leqslant 3, |x_1| \leqslant 4, |x_2| \leqslant 4\}$. For approximating the costates, a linear in parameter neural network with polynomial basis functions up to the power of 4 without repetition was selected. The training process concluded in 85.59 (s) using Algorithm 1 and only 18.5 (s) using Algorithm 2. The histories of weights of the neural networks for approximating the costates using Algorithms 1 and 2 are shown in Figs. 8 and 9, respectively.

Once the training concluded, the optimal costates were used to find the optimal switching times with a similar procedure that was used in the previous example. Using Algorithm 1, the optimal

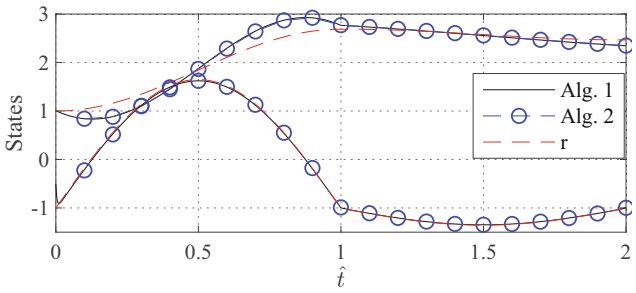


Fig. 4. Comparison among the state trajectories generated by controllers trained by Algorithms 1 and 2.

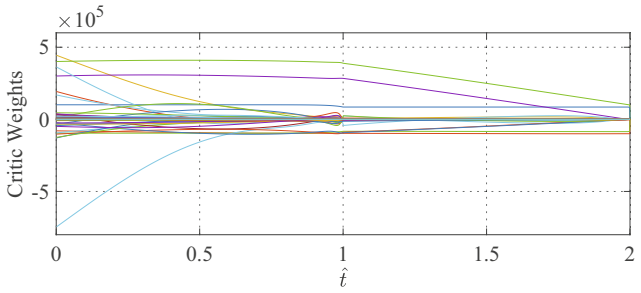


Fig. 5. The history of the weights of the neural network to approximate costates trained by Algorithm 1 and the state-based reference signal.

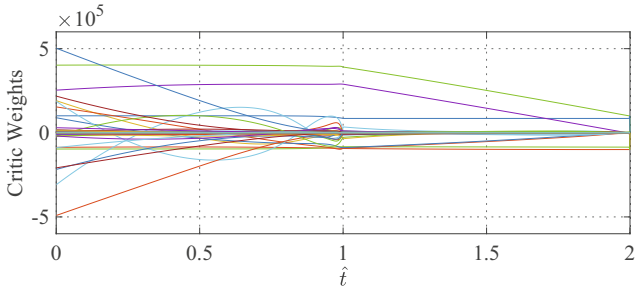


Fig. 6. The history of the weights of the neural network to approximate costates trained by Algorithm 2 and the state-based reference signal.

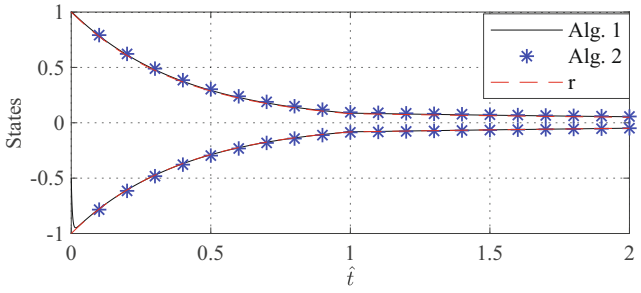


Fig. 7. Comparison among the state trajectories generated by controllers trained by Algorithms 1 and 2.

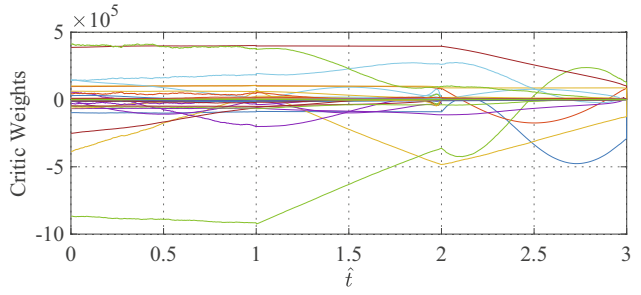


Fig. 8. The history of the weights of the neural network to approximate costates using Algorithm 1. The jumps at $\hat{t} = 1$ and $\hat{t} = 2$ show the switching at these times.

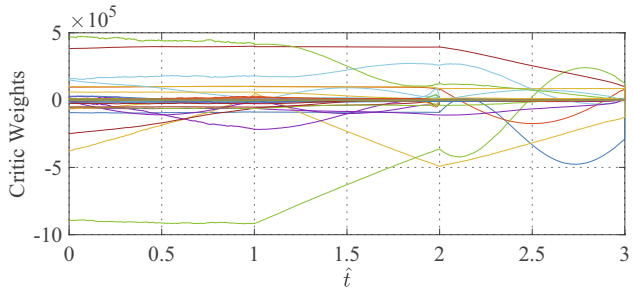


Fig 9: The history of the weights of the neural net

Fig. 9. The history of the weights of the neural network to approximate costates using Algorithm 2. The jumps at $\hat{t} = 1$ and $\hat{t} = 2$ show the switching at these times.

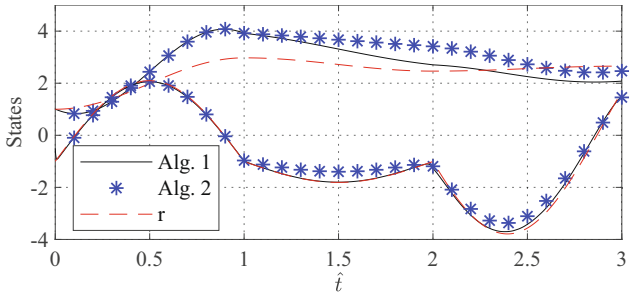


Fig. 10. Comparison among the state trajectories propagated along the controls generated by the ADP methods discussed in this paper.

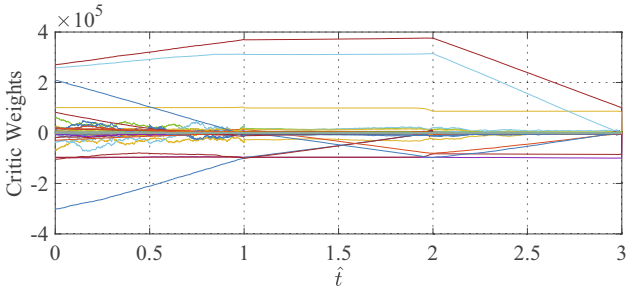


Fig. 11. The history of the weights of the neural network to approximate costates trained by Algorithm 1 and the state-based reference signal.

switching time instants were sought as $t_1 = 2.72$ (s) and $t_2 = 2.78$ (s). Also, using Algorithm 2, the optimal switching times were sought as $t_1 = 2.72$ (s) and $t_2 = 2.82$ (s) which are close to the results of using Algorithm 1.

At last, the controllers trained by Algorithms 1 and 2 were used online to propagate the states. The performance of these controllers are compared in Fig. 10. As one can see from Fig. 10, both controllers have a very good performance in tracking the reference signal and the performance of the controllers are close to each other.

5.2.2. State-based reference signal

Considering the mode sequence as Γ^2, the cost function as (2), and the reference signal as (33), it is desired to find the optimal switching times t_1 and t_2, and the optimal policy such that (2) is minimized and the system tracks the reference signal.

Selecting the state/control penalizing matrices, and the discretization sample time the same as the previous example, one starts the simulations. The domain of training was selected

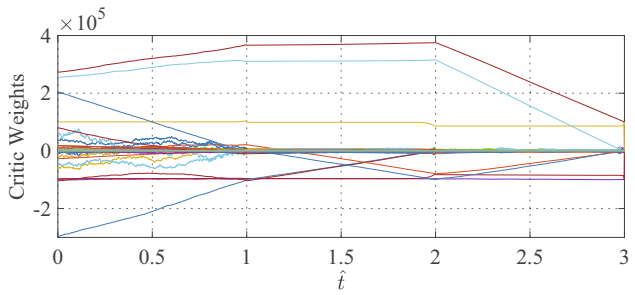


Fig. 12. The history of the weights of the neural network to approximate costates trained by Algorithm 2 and the state-based reference signal.

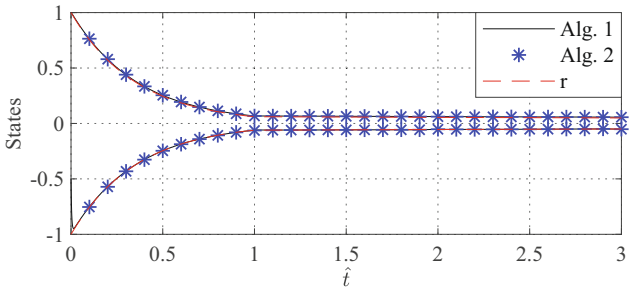


Fig. 13. Comparison among the state trajectories generated by controllers trained by Algorithms 1 and 2.

as $\Omega = \{(t_1, t_2, x_1, x_2, r_1, r_2) | 0 < t_1 < t_2 < 3, |x_1| \leqslant 4, |x_2| \leqslant 4, |r_1| \leqslant 4, |r_2| \leqslant 4\}$. Using Algorithms 1 and 2, the training process concluded in 92.8 (s) and 31.6 (s), respectively. The histories of the weights of the neural networks trained by Algorithms 1 and 2 are shown in Figs. 11 and 12, respectively. Like the previous examples, one can detect jumps at the switching instants.

Once the training concluded, the controllers trained by Algorithms 1 and 2 were used for online control without re-training. The optimal costates were first integrated at an initial condition to find the value function as a function of switching times. Then, the resulted value function was minimized with respect to the switching times to find the optimal switching times. Using the costates trained by Algorithm 1, the switching times were sought as $t_1 = 2.8$ (s) and $t_2 = 2.9$ (s). The same switching times were sought using the costates trained by Algorithm 2.

At last, the performance of the controllers trained by Algorithms 1 and 2 are compared in Figs. 13. As one can see from 13, both controllers could track the desired trajectory and the performance of the controllers are close to each other.

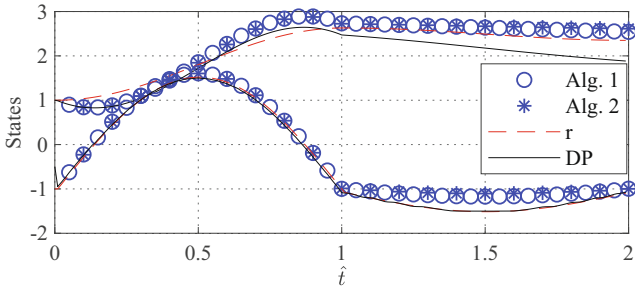


Fig. 14. Comparison of the performance of the controllers trained in Section 5.1.1 with DP.

5.3. Comparing with dynamic programming

In order to further evaluate the performance of the introduced ADP methods, the performance of the controllers trained in Section 5.1.1 is compared with that of a DP controller. To perform DP, the quantizaion values were selected as $\delta x_1 = 0.1, \delta x_2 = 0.1, \delta t_1 = 0.05$, and $\delta u = 0.5$. The discretization sample time was selected as $\delta \hat{t} = 0.01$ and it took 1286.8 (s) for the DP to complete the solution. Once the solution concluded, the optimal switching time was sought at $t_1 = 2.56$ (s) for the same initial condition as used in the previous examples, i.e., $x_0 = [1, -0.5]^T$. The history of the states are compared in Fig. 14. As one can see from Fig. 14, the performance of the controllers trained in this paper is very close to that of the DP one.

6. Conclusion

Optimal tracking in switched systems with fixed mode sequence was studied in this paper. The main feature of the proposed controller was using a transformation [7] to parametrize the switching times and then incorporating the parametrization in a Single Network Adaptive Critic (SNAC) [19] to find the optimal costates. Afterward, the optimal switching times were sought through constrained minimization for each initial condition once the training was concluded. Also, to improve the speed of training, a single loop SNAC controller was introduced, and the performance of this controller was studied. At last, the performance of the proposed controllers was evaluated through numerical simulations.

Some potential directions for future research are as follows. As the first direction, the proposed method in this paper can be used for optimal control of autonomous systems such as off-road vehicles. Also, one can consider the application of fault tolerancing such as the one reported in [35] to improve the performance of the system in the occurrence of faults. At last, one can study the effect of input constraints as studied in [36].

CRediT authorship contribution statement

Tohid Sardarmehni: Conceptualization, Methodology, Software, Data curation, Writing - original draft, Visualization, Investigation. **Xingyong Song:** Conceptualization, Methodology, Supervision, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research was partially supported by the National Science Foundation under Grant No. 1826410.

Appendix A. Proof of Theorem 1

Proof: The proof is inspired by [19]. In order to start the proof, one first considers the time step $N'-1 \to N'$. Considering (25), one has

$$\begin{aligned}&\lambda^{i+1}_{N'}\left(x^{i+1}_{N'}\right) = S\left(x^{i}_{N'} - r_{N'}\right)\\&= S\Big(f_{v_{N'-1}}(x_{N'-1}) + g_{v_{N'-1}}(x_{N'-1})\left(-R_{v_{N'-1}}\right)^{-1}\\&\times g^T_{v_{N'-1}}(x_{N'-1})\lambda^{i}_{N'}\left(x^{i}_{N'}\right) - r_{N'}\Big)\end{aligned} \tag{34}$$

Also, evaluating (25) at iteration i gives

$$\begin{aligned}&\lambda^{i}_{N'}\left(x^{i}_{N'}\right) = S\left(x^{i-1}_{N'} - r_{N'}\right)\\&= S\Big(f_{v_{N'-1}}(x_{N'-1}) + g_{v_{N'-1}}(x_{N'-1})\left(-R_{v_{N'-1}}\right)^{-1}\\&\times g^T_{v_{N'-1}}(x_{N'-1})\lambda^{i-1}_{N'}\left(x^{i-1}_{N'}\right) - r_{N'}\Big)\end{aligned} \tag{35}$$

Subtracting (35) from (34), one can find $\lambda^{i+1} - \lambda^i$ as

$$\lambda^{i+1} - \lambda^i = Sg(x_{N'-1})(-R)^{-1}g^T(x_{N'-1})\left(\lambda^i - \lambda^{i-1}\right) \tag{36}$$

In (36), $\lambda^i = \lambda^i_{N'}\left(x^i_{N'}\right), g(x_{N'-1}) = g_{v_{N'-1}}(x_{N'-1})$, and $R = R_{v_{N'-1}}$ for notational simplicity. Letting $\varepsilon^{i+1} = \lambda^{i+1} - \lambda^i$ and $\varepsilon^i = \lambda^i - \lambda^{i-1}$ in (36), one has

$$\varepsilon^{i+1} = \underbrace{Sg(x_{N'-1})(-R)^{-1}g^T(x_{N'-1})}_{\alpha}\,\varepsilon^i \tag{37}$$

Applying norms leads

$$\|\varepsilon^{i+1}\| \leqslant \|\alpha\|\|\varepsilon^i\| \tag{38}$$

In (38), one notes that $\|\alpha\| = \rho_1^2\|R^{-1}\|$ where ρ_1 is an upper bound for the smooth function $\|g(.)\|$ in a compact set Ω. It can be seen that the value of $\|\alpha\|$ can be adjusted through choice of R. Hence, there exists a control penalizing matrix R^* such that for any R that $\|R\| > \|R^*\|$ one has $\|\alpha\| < 1$. Therefore, one has $\|\varepsilon^{i+1}\| < \|\varepsilon^i\|$. As a result, the sequence of $\|\varepsilon^i\|$ resulted from the proposed iterations forms a monotonically decreasing sequence which is bounded below by 0 as

$$\|\varepsilon^0\| > \|\varepsilon^1\| > \|\varepsilon^2\| \cdots > \|\varepsilon^\infty\| \geqslant 0 \tag{39}$$

The convergence of the sequence of $\|\varepsilon^i\|$ can be concluded since any lower bounded monotonically decreasing sequence converges [34]. Hence, $\|\varepsilon^i\| \to 0$ which means $\lambda^{i+1} \to \lambda^i$.

Depending on the discrete time index and the active mode, the iteration $i+1$ for $\hat{k} < N'$ can be denoted by

$$\begin{aligned}&\lambda^{i+1}_{\hat{k}+1}\left(\Gamma, x^{i+1}_{\hat{k}+1}, r_{\hat{k}+1}\right) =\\&Q_{v_{\hat{k}+1}}\left(x^i_{\hat{k}+1} - r_{\hat{k}+1}\right) + \left(\frac{\partial x_{\hat{k}+2}}{\partial x_{\hat{k}+1}}\right)^T \lambda_{\hat{k}+2}\left(\Gamma, x^i_{\hat{k}+2}, r_{\hat{k}+2}\right)\end{aligned} \tag{40}$$

For the sake of notational simplicity, hereafter in this proof, $\lambda^i_{\hat{k}+1}\left(\Gamma, x^i_{\hat{k}+1}, r_{\hat{k}+1}\right) \equiv \lambda^i_{\hat{k}+1}$, and the sub-index for active mode, i.e., $v_{\hat{k}}$, is dropped in showing Q, R, $f(.)$, and $g(.)$ unless otherwise stated. Therefore, one has

$$\lambda^{i+1}_{\hat{k}+1} = Q\left(x^i_{\hat{k}+1} - r_{\hat{k}+1}\right) + \left(\frac{\partial x_{\hat{k}+2}}{\partial x_{\hat{k}+1}}\right)^T \lambda_{\hat{k}+2}\left(x^i_{\hat{k}+2}\right) \tag{41}$$

In (41),

$$\frac{\partial x_{\hat{k}+2}}{\partial x_{\hat{k}+1}} = \frac{\partial}{\partial x}\left(f(x) + g(x)u(t)\right)|_{x^i_{\hat{k}+1}}$$

where $x^i_{\hat{k}+1} = f(x_{\hat{k}}) + g(x_{\hat{k}})(-R)^{-1}g^T(x_{\hat{k}})\lambda^i_{\hat{k}+1}$. Considering the gradient as a column vector, one can easily find $\frac{\partial f(.)}{\partial x}$. Also, one has

$$\frac{\partial\left(g(x)u^i(t)\right)}{\partial x} = \begin{bmatrix} \frac{\partial g_{1j}}{\partial x_1}u^i_j & \frac{\partial g_{2j}}{\partial x_1}u^i_j & \cdots & \frac{\partial g_{nj}}{\partial x_1}u^i_j \\ \frac{\partial g_{1j}}{\partial x_2}u^i_j & \frac{\partial g_{2j}}{\partial x_2}u^i_j & \cdots & \frac{\partial g_{nj}}{\partial x_2}u^i_j \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial g_{1j}}{\partial x_n}u^i_j & \frac{\partial g_{2j}}{\partial x_n}u^i_j & \cdots & \frac{\partial g_{nj}}{\partial x_n}u^i_j \end{bmatrix} \tag{42}$$

In presenting (42), one used tensor notations to show the summations. Hence,

$$\frac{\partial g_{*j}}{\partial x_{**}} u_j^i = \sum_{j=1}^{m} \frac{\partial g_{*j}}{\partial x_{**}} u_j^i$$

where $*and\ \ **\in\{1,2,\dots,n\}$. For notational simplicity, let us denote all the elements in a row or a column of a matrix with :. For instance, $g_{(1,:)}$ means all the elements in the first row of matrix $g(.)$. Similarly, $g_{(:,1)}$denotes all the elements in the first column of matrix $g(.)$. Using the above mentioned notation, one can rewrite (42) as

$$\frac{\partial\left(g\left(x_{k+1}^i\right)u_{k+1}^i\right)}{\partial x_{k+1}^i} = \begin{bmatrix} \left(\nabla g_{(1,:)}\left(x_{k+1}^i\right)u_{k+1}^i\right)^T \\ \left(\nabla g_{(2,:)}\left(x_{k+1}^i\right)u_{k+1}^i\right)^T \\ \vdots \\ \left(\nabla g_{(n,:)}\left(x_{k+1}^i\right)u_{k+1}^i\right)^T \end{bmatrix} \tag{43}$$

With the notations developed in (43), one can find λ_{k+1}^i and λ_{k+1}^{i+1}. From (41), one has

$$\lambda_{k+1}^{i+1} = Q\left(x_{k+1}^i - r_{k+1}\right) + \left(\nabla f\left(x_{k+1}^i\right) + \nabla\left(g\left(x_{k+1}^i\right)u_{k+1}^i\right)\right)^T \lambda_{k+2}\left(x_{k+2}^i\right) = Q\left(x_{k+1}^i - r_{k+1}\right) + \nabla^T f\left(x_{k+1}^i\right)\lambda_{k+2}\left(x_{k+2}^i\right) + \underbrace{\begin{bmatrix} {u_{k+1}^i}^T\nabla^T g_{(1,:)}\left(x_{k+1}^i\right) \\ {u_{k+1}^i}^T\nabla^T g_{(2,:)}\left(x_{k+1}^i\right) \\ \vdots \\ {u_{k+1}^i}^T\nabla^T g_{(n,:)}\left(x_{k+1}^i\right) \end{bmatrix}}_{\Upsilon_{k+1}^i}\lambda_{k+2}\left(x_{k+2}^i\right) \tag{44}$$

Considering (44), substituting for ${u_{k+1}^i}^T$ one has

$$\Upsilon_{k+1}^i = \begin{bmatrix} \lambda_{k+2}^T\left(x_{k+2}^i\right)g\left(x_{k+1}^i\right)\left(-R^{-1}\right)\nabla^T g_{(1,:)}\left(x_{k+1}^i\right) \\ \lambda_{k+2}^T\left(x_{k+2}^i\right)g\left(x_{k+1}^i\right)\left(-R^{-1}\right)\nabla^T g_{(2,:)}\left(x_{k+1}^i\right) \\ \vdots \\ \lambda_{k+2}^T\left(x_{k+2}^i\right)g\left(x_{k+1}^i\right)\left(-R^{-1}\right)\nabla^T g_{(n,:)}\left(x_{k+1}^i\right) \end{bmatrix}\lambda_{k+2}\left(x_{k+2}^i\right) \tag{45}$$

In order to further simplify the notations, consider $\Lambda_{1_{k+1}}^i$as

$$\Lambda_{1_{k+1}}^i = \begin{bmatrix} \lambda_{k+2}^{iT}g_{k+1}^i & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \lambda_{k+2}^{iT}g_{k+1}^i & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & & & & \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \lambda_{k+2}^{iT}g_{k+1}^i \end{bmatrix} \tag{46}$$

where $\lambda_{k+2}^{iT}g_{k+1}^i = \lambda_{k+2}^{iT}\left(x_{k+2}^i\right)g\left(x_{k+1}^i\right)$ and $\mathbf{0}\in\mathbb{R}^{1\times m}$ is a matrix of all elements zeros. Similarly, consider $\Lambda_{2_{k+1}}^i$

$$\Lambda_{2_{k+1}}^i = \begin{bmatrix} -R^{-1} & [\mathbf{0}] & [\mathbf{0}] & \cdots & [\mathbf{0}] \\ [\mathbf{0}] & -R^{-1} & [\mathbf{0}] & \cdots & [\mathbf{0}] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ [\mathbf{0}] & [\mathbf{0}] & \cdots & [\mathbf{0}] & -R^{-1} \end{bmatrix} \tag{47}$$

where $[\mathbf{0}]$ is a $m\times m$ matrix with all elements as zeros. At last, consider $\Lambda_{3_{k+1}}^i$ as

$$\Lambda_{3_{k+1}}^i = \begin{bmatrix} \nabla^T g_{(1,:)}\left(x_{k+1}^i\right) \\ \nabla^T g_{(2,:)}\left(x_{k+1}^i\right) \\ \vdots \\ \nabla^T g_{(n,:)}\left(x_{k+1}^i\right) \end{bmatrix}\lambda_{k+2}\left(x_{k+2}^i\right) \tag{48}$$

With (46)–(48), one can rewrite (44) as

$$\lambda_{k+1}^{i+1} = Q\left(x_{k+1}^i - r_{k+1}\right) + \nabla^T f\left(x_{k+1}^i\right)\lambda_{k+2}\left(x_{k+2}^i\right) + \Lambda_{1_{k+1}}^i\Lambda_{2_{k+1}}^i\Lambda_{3_{k+1}}^i \tag{49}$$

Similar to (49), one can perform the same calculations for λ_{k+1}^i as

$$\lambda_{k+1}^i = Q\left(x_{k+1}^{i-1} - r_{k+1}\right) + \nabla^T f\left(x_{k+1}^{i-1}\right)\lambda_{k+2}\left(x_{k+2}^{i-1}\right) + \Lambda_{1_{k+1}}^{i-1}\Lambda_{2_{k+1}}^{i-1}\Lambda_{3_{k+1}}^{i-1} \tag{50}$$

Subtracting (50) from (49), one has

$$\begin{aligned} \lambda_{k+1}^{i+1} - \lambda_{k+1}^i &= Q\left(x_{k+1}^i - x_{k+1}^{i-1}\right) \\ &+\left(\nabla^T f\left(x_{k+1}^i\right)\lambda_{k+2}\left(x_{k+2}^i\right) - \nabla^T f\left(x_{k+1}^{i-1}\right)\lambda_{k+2}\left(x_{k+2}^{i-1}\right)\right) \\ &+\left(\Lambda_{1_{k+1}}^i\Lambda_{2_{k+1}}^i\Lambda_{3_{k+1}}^i - \Lambda_{1_{k+1}}^{i-1}\Lambda_{2_{k+1}}^{i-1}\Lambda_{3_{k+1}}^{i-1}\right) \end{aligned} \tag{51}$$

In (50), one notes that $\Lambda_{2_{k+1}}^i = \Lambda_{2_{k+1}}^{i-1}$ which is not dependent to the iterations. Hence, one can consider $\Lambda_{2_{k+1}}^i = \Lambda_{2_{k+1}}^{i-1} \equiv \Lambda_{2_{k+1}}$. To continue the proof, consider the following algebraic equation with A_1, B, C_1, A_2, B_2 being matrices of the same dimensions as $\Lambda_{1_{k+1}}^i, \Lambda_{2_{k+1}}, \Lambda_{3_{k+1}}^i, \Lambda_{1_{k+1}}^{i-1}, \Lambda_{3_{k+1}}^{i-1}$

$$A_1BC_1 - A_2BC_2 = (A_1 - A_2)BC_1 + A_2B(C_1 - C_2) \tag{52}$$

Using (52) in (50), one has

$$\begin{aligned}&\lambda_{\hat{k}+1}^{i+1}-\lambda_{\hat{k}+1}^{i}=Q\left(x_{\hat{k}+1}^{i}-x_{\hat{k}+1}^{i-1}\right)\\&+\left(\nabla^{T}f\left(x_{\hat{k}+1}^{i}\right)\lambda_{\hat{k}+2}\left(x_{\hat{k}+2}^{i}\right)-\nabla^{T}f\left(x_{\hat{k}+1}^{i-1}\right)\lambda_{\hat{k}+2}\left(x_{\hat{k}+2}^{i-1}\right)\right)\\&+\left(\Lambda_{1_{\hat{k}+1}}^{i}-\Lambda_{1_{\hat{k}+1}}^{i-1}\right)\Lambda_{2_{\hat{k}+1}}\Lambda_{3_{\hat{k}+1}}^{i}\\&+\Lambda_{1_{\hat{k}+1}}^{i-1}\Lambda_{2_{\hat{k}+1}}\left(\Lambda_{3_{\hat{k}+1}}^{i}-\Lambda_{3_{\hat{k}+1}}^{i-1}\right)\end{aligned}\tag{53}$$

Similar to (38), one is interested to find a monotonic behavior in the iterations. Therefore, By taking the norm of (53), through some algebraic manipulations one has

$$\begin{aligned}&\|\lambda_{\hat{k}+1}^{i+1}-\lambda_{\hat{k}+1}^{i}\|\leqslant\|Q\|\|\left(x_{\hat{k}+1}^{i}-x_{\hat{k}+1}^{i-1}\right)\|\\&+\|\nabla^{T}f\left(x_{\hat{k}+1}^{i}\right)\lambda_{\hat{k}+2}\left(x_{\hat{k}+2}^{i}\right)-\nabla^{T}f\left(x_{\hat{k}+1}^{i-1}\right)\lambda_{\hat{k}+2}\left(x_{\hat{k}+2}^{i-1}\right)\|\\&+\|\left(\Lambda_{1_{\hat{k}+1}}^{i}-\Lambda_{1_{\hat{k}+1}}^{i-1}\right)\|\|\Lambda_{2_{\hat{k}+1}}\|\|\Lambda_{3_{\hat{k}+1}}^{i}\|\\&+\|\Lambda_{1_{\hat{k}+1}}^{i-1}\|\|\Lambda_{2_{\hat{k}+1}}\|\|\left(\Lambda_{3_{\hat{k}+1}}^{i}-\Lambda_{3_{\hat{k}+1}}^{i-1}\right)\|\end{aligned}\tag{54}$$

By Assumption, the value functions are smooth. This results in smoothness of the costates, i.e., $\lambda_{\hat{k}+2}(.)$. Also, smoothness of $f(.)$ leads to smoothness of $\nabla f(.)$. Furthermore, smoothness of $g(.)$ along with smoothness of $f(.)$ and the value functions lead to Lipschitz continuity of $\nabla f(.)\lambda_{\hat{k}+2}(.)$ with Lipschitz constant of β_1, Lipschitz continuity of $\Lambda_{1_{\hat{k}+1}}^{i}$ with Lipschitz constant of β_2, and Lipschitz continuity of $\Lambda_{3_{\hat{k}+1}}^{i}$ with Lipschitz constant of β_3. In addition, one notes that the smoothness of $g(.),\Lambda_{1_{\hat{k}+1}}^{i}$ and $\Lambda_{3_{\hat{k}+1}}^{i}$ leads to boundedness in the compact region Ω. Therefore, one has $\|g(.)\|\leqslant\rho_1,\|\Lambda_{1_{\hat{k}+1}}^{i}\|\leqslant\rho_2$, and $\|\Lambda_{3_{\hat{k}+1}}^{i}\|\leqslant\rho_3$. Also, it is straightforward to see that $\|\Lambda_{2_{\hat{k}+1}}^{i}\|\leqslant n\|R^{-1}\|$. Therefore, one has

$$\begin{aligned}&\|\lambda_{\hat{k}+1}^{i+1}-\lambda_{\hat{k}+1}^{i}\|\leqslant\|Q\|\|\left(x_{\hat{k}+1}^{i}-x_{\hat{k}+1}^{i-1}\right)\|\\&+\beta_1\|\left(x_{\hat{k}+1}^{i}-x_{\hat{k}+1}^{i-1}\right)\|\\&+n\rho_3\beta_2\|R^{-1}\|\|\left(x_{\hat{k}+1}^{i}-x_{\hat{k}+1}^{i-1}\right)\|\\&+n\rho_2\beta_3\|R^{-1}\|\|\left(x_{\hat{k}+1}^{i}-x_{\hat{k}+1}^{i-1}\right)\|\end{aligned}\tag{55}$$

Considering (55), one notes that

$$\begin{aligned}\|\left(x_{\hat{k}+1}^{i}-x_{\hat{k}+1}^{i-1}\right)\|\leqslant\;&\|g(x_{\hat{k}})\|^2\|R^{-1}\|\|\lambda_{\hat{k}+1}^{i}-\lambda_{\hat{k}+1}^{i-1}\|\\&\leqslant\rho_1^2\|R^{-1}\|\|\lambda_{\hat{k}+1}^{i}-\lambda_{\hat{k}+1}^{i-1}\|\end{aligned}\tag{56}$$

Using (56) in (55), one has

$$\begin{aligned}&\|\lambda_{\hat{k}+1}^{i+1}-\lambda_{\hat{k}+1}^{i}\|\leqslant(\|Q\|+\beta_1\\&+(n\rho_3\beta_2+n\rho_2\beta_3)\|R^{-1}\|\big)\rho_1^2\|R^{-1}\|\|\lambda_{\hat{k}+1}^{i}-\lambda_{\hat{k}+1}^{i-1}\|\\&\leqslant\alpha_1\|\lambda_{\hat{k}+1}^{i}-\lambda_{\hat{k}+1}^{i-1}\|\end{aligned}\tag{57}$$

Letting $\varepsilon^{i+1}=\lambda_{\hat{k}+1}^{i+1}-\lambda_{\hat{k}+1}^{i}$ and $\varepsilon^{i}=\lambda_{\hat{k}+1}^{i}-\lambda_{\hat{k}+1}^{i-1}$, one has

$$\|\varepsilon^{i+1}\|\leqslant\alpha_1\|\varepsilon^{i}\|\tag{58}$$

Considering the boundedness of α_1, it is easy to see that $\|\varepsilon^{i}\|$ in (58) forms a monotonically decreasing sequence by the correct choice of R. Therefore, using the same discussion in the first part of the proof, the convergence of $\|\varepsilon^{i}\|$s to zero can be concluded.

Consider $\lambda_{\hat{k}}^{\infty}$ as $i\to\infty$. We can find the policy $u_{\hat{k}}^{\infty}$ using $\lambda_{\hat{k}}^{\infty}$ and (16). Also, let $V_{\hat{k}}^{\infty}$ as the integral of $\lambda_{\hat{k}}^{\infty}$ along trajectory resulted from using policy $u_{\hat{k}}^{\infty}$. One can see that the pair $V_{\hat{k}}^{\infty}$ and $u_{\hat{k}}^{\infty}$ solve (14). Due to uniqueness of the HJB equation [37], it can be deducted that $V_{\hat{k}}^{\infty}$ and $u_{\hat{k}}^{\infty}$ are the optimal value function and the optimal policy. Hence, $\lambda_{\hat{k}}^{\infty}$ is the optimal costate. □.

Appendix B. Proof of Theorem 2

Proof: Let the costates generated by Algorithm 1 be denoted by $\lambda_{\hat{k}+1}^{\infty}(.)$ and the costates generated by Algorithm 2 be represented as $\lambda_{\hat{k}+1}^{1}$. Considering $x_{\hat{k}}$, one can denote the states propagated along $\lambda_{\hat{k}+1}^{\infty}(.)$ and $\lambda_{\hat{k}+1}^{1}$ as

$$x_{\hat{k}+1}^{\infty}=f(x_{\hat{k}})+g(x_{\hat{k}})(-R)^{-1}g^{T}(x_{\hat{k}})\lambda_{\hat{k}+1}^{\infty}\tag{59}$$

$$x_{\hat{k}+1}^{1}=f(x_{\hat{k}})+g(x_{\hat{k}})(-R)^{-1}g^{T}(x_{\hat{k}})\lambda_{\hat{k}+1}^{1}\tag{60}$$

Subtracting (60) from (59), at each time instant $\hat{k}$ one has

$$x_{\hat{k}+1}^{\infty}-x_{\hat{k}+1}^{1}=g(x_{\hat{k}})(-R)^{-1}g^{T}(x_{\hat{k}})\left(\lambda_{\hat{k}+1}^{\infty}-\lambda_{\hat{k}+1}^{1}\right)\tag{61}$$

Considering time instant $\hat{k}=N'-1$, one has

$$x_{N'}^{\infty}-x_{N'}^{1}=g(x_{N'-1})(-R)^{-1}g^{T}(x_{N'-1})(\lambda_{N'}^{\infty}-\lambda_{N'}^{1})\tag{62}$$

Substituting for $\lambda_{N'}^{\infty}$ and $\lambda_{N'}^{1}$, one has

$$\begin{aligned}&x_{N'}^{\infty}-x_{N'}^{1}=g(x_{N'-1})(-R)^{-1}g^{T}(x_{N'-1})\\&*\Big(S\Big(f(x_{N'-1})+g(x_{N'-1})(-R)^{-1}g^{T}(x_{N'-1})\lambda_{N'}^{\infty}-r_{N'}\Big)\\&-S\Big(f(x_{N'-1})+g(x_{N'-1})(-R)^{-1}g^{T}(x_{N'-1})\lambda_{N'}^{0}-r_{N'}\Big)\Big)\end{aligned}\tag{63}$$

After some algebraic manipulations, one has

$$\begin{aligned}&x_{N'}^{\infty}-x_{N'}^{1}=\underbrace{g(x_{N'-1})(-R)^{-1}g^{T}(x_{N'-1})}_{\Psi}\\&*S\Big(g(x_{N'-1})(-R)^{-1}g^{T}(x_{N'-1})(\lambda_{N'}^{\infty}-\lambda_{N'}^{0})=\Psi S\Psi(\lambda_{N'}^{\infty}-\lambda_{N'}^{0})\end{aligned}\tag{64}$$

Considering the right-hand side of (64), one has

$$\begin{aligned}&x_{N'}^{\infty}-x_{N'}^{1}=\Psi S\Psi(\lambda_{N'}^{\infty}-\lambda_{N'}^{0})\\&=\Psi S\Psi(\lambda_{N'}^{\infty}-\lambda_{N'}^{\infty-1}+\lambda_{N'}^{\infty-1}-\cdots-\lambda_{N'}^{2}+\lambda_{N'}^{1}-\lambda_{N'}^{0})\end{aligned}\tag{65}$$

Letting $\varepsilon_{N'}^{i+1}=\lambda_{N'}^{i+1}-\lambda_{N'}^{i}$, one can re-write (65) as

$$x_{N'}^{\infty}-x_{N'}^{1}=\Psi S\Psi(\varepsilon_{N'}^{\infty-1}+\varepsilon_{N'}^{\infty-2}+\cdots+\varepsilon_{N'}^{2}+\varepsilon_{N'}^{1})\tag{66}$$

Applying norms on (66), after some algebraic manipulations one has

$$\begin{aligned}&\|x_{N'}^{\infty}-x_{N'}^{1}\|\leqslant\|\Psi\|^2\|S\|\\&\times(\|\varepsilon_{N'}^{\infty-1}\|+\|\varepsilon_{N'}^{\infty-2}\|+\cdots+\|\varepsilon_{N'}^{2}\|+\|\varepsilon_{N'}^{1}\|)\end{aligned}\tag{67}$$

As shown in the proof of Theorem 1, the sequence of $\|\varepsilon_{N'}^{i}\|$ is monotonically decreasing with $\|\varepsilon_{N'}^{i+1}\|\leqslant\|\alpha\|\|\varepsilon_{N'}^{i}\|$ where $\|\alpha\|<1$. Therefore, (67) leads

$$\begin{aligned}&\|x_{N'}^{\infty}-x_{N'}^{1}\|\leqslant\\&\|\Psi\|^2\|S\|(\|\varepsilon_{N'}^{1}\|+\alpha\|\varepsilon_{N'}^{1}\|+\alpha^2\|\varepsilon_{N'}^{1}\|+\cdots+\alpha^{\infty}\|\varepsilon_{N'}^{1}\|)\\&\leqslant\|\Psi\|^2\|S\|\|\varepsilon_{N'}^{1}\|(1+\alpha+\alpha^2+\cdots+\alpha^{\infty})\end{aligned}\tag{68}$$

In (68), since $\|\alpha\|<1$, the series form a geometric series which converges to $\frac{1}{1-\|\alpha\|}$. As a result, one has

$$\begin{aligned}\|x_{N'}^{\infty}-x_{N'}^{1}\|\;&\leqslant\|\Psi\|^2\|S\|\|\varepsilon_{N'}^{1}\|\tfrac{1}{1-\|\alpha\|}\\&\leqslant\rho_1^4\|R^{-1}\|^2\|S\|\|\varepsilon_{N'}^{1}\|\tfrac{1}{1-\|\alpha\|}\end{aligned}\tag{69}$$

In (69), ρ_1 is an upper bound for $g(.)$ in a compact set of training Ω. It is straightforward to see that $\|\varepsilon^1_{N'-1}\|$ is bounded. Hence, the magnitude of the error can become small as $\|R\|$ increases.

For the rest of times $\hat{k}$, consider (59), (60) and (62). For $\hat{k} < N' - 1$, one has

$$\begin{aligned} x^{\infty}_{\hat{k}+1} - x^1_{\hat{k}+1} &= \underbrace{g(x_{\hat{k}})(-R)^{-1}g^T(x_{\hat{k}})}_{\Psi}\left(\lambda^{\infty}_{\hat{k}+1} - \lambda^1_{\hat{k}+1}\right) \\ &= \Psi\Big(Q\left(f(x_{\hat{k}}) + g(x_{\hat{k}})(-R)^{-1}g^T(x_{\hat{k}})\lambda^{\infty}_{\hat{k}+1} - r_{\hat{k}+1}\right) \\ &+ \tfrac{\partial x_{\hat{k}+2}}{\partial x_{\hat{k}+1}}|_{x^{\infty}_{\hat{k}+1}}\lambda_{\hat{k}+2}\left(x^{\infty}_{\hat{k}+2}\right) \\ &- Q\left(f(x_{\hat{k}}) + g(x_{\hat{k}})(-R)^{-1}g^T(x_{\hat{k}})\lambda^0_{\hat{k}+1} - r_{\hat{k}+1}\right) \\ &- \tfrac{\partial x_{\hat{k}+2}}{\partial x_{\hat{k}+1}}|_{x^0_{\hat{k}+1}}\lambda_{\hat{k}+2}\left(x^0_{\hat{k}+2}\right)\Big) \end{aligned} \tag{70}$$

After some algebraic manipulations, one has

$$\begin{aligned} x^{\infty}_{\hat{k}+1} - x^1_{\hat{k}+1} &= \Psi Q\Psi\left(\lambda^{\infty}_{\hat{k}+1} - \lambda^0_{\hat{k}+1}\right) \\ &+ \Psi\left(\nabla f\left(x^{\infty}_{\hat{k}+1}\right)\lambda_{\hat{k}+2}\left(x^{\infty}_{\hat{k}+2}\right) - \nabla f\left(x^0_{\hat{k}+1}\right)\lambda_{\hat{k}+2}\left(x^0_{\hat{k}+2}\right)\right) \\ &+ \Psi\left(\Lambda^{\infty}_{1_{\hat{k}+1}}\Lambda^{\infty}_{2_{\hat{k}+1}}\Lambda^{\infty}_{3_{\hat{k}+1}} - \Lambda^0_{1_{\hat{k}+1}}\Lambda^0_{2_{\hat{k}+1}}\Lambda^0_{3_{\hat{k}+1}}\right) \end{aligned} \tag{71}$$

In (71), $\Lambda^*_{1_{\hat{k}+1}}, \Lambda^*_{2_{\hat{k}+1}}$ and $\Lambda^*_{3_{\hat{k}+1}}$ were introduced in (46)–(48), respectively. Applying norms on (71) leads

$$\begin{aligned} \|x^{\infty}_{\hat{k}+1} - x^1_{\hat{k}+1}\| &\leqslant \|\Psi Q\Psi\|\|\lambda^{\infty}_{\hat{k}+1} - \lambda^0_{\hat{k}+1}\| \\ &+ \|\Psi\|\|\nabla f\left(x^{\infty}_{\hat{k}+1}\right)\lambda_{\hat{k}+2}\left(x^{\infty}_{\hat{k}+1}\right) - \nabla f\left(x^0_{\hat{k}+1}\right)\lambda_{\hat{k}+2}\left(x^0_{\hat{k}+1}\right)\| \\ &+ \|\Psi\|\|\Lambda^{\infty}_{1_{\hat{k}+1}}\Lambda^{\infty}_{2_{\hat{k}+1}}\Lambda^{\infty}_{3_{\hat{k}+1}} - \Lambda^0_{1_{\hat{k}+1}}\Lambda^0_{2_{\hat{k}+1}}\Lambda^0_{3_{\hat{k}+1}}\| \end{aligned} \tag{72}$$

Following the same procedure as explained in Eqs. (52)–(56), (58), (53)–(56), (58), (58), through some algebraic manipulations one has

$$\begin{aligned} \|x^{\infty}_{\hat{k}+1} - x^1_{\hat{k}+1}\| &\leqslant \left(\rho_1^4\|Q\|\|R_{-1}\|^2 + \beta_1\rho_1^2\|R^{-1}\|\right. \\ &\left.+ \rho_1^4 n\|R^{-1}\|^3(\rho_3\beta_2 + \rho_2\beta_3)\right)\|\lambda^{\infty}_{\hat{k}+1} - \lambda^0_{\hat{k}+1}\| \\ &\leqslant \alpha_2\|\lambda^{\infty}_{\hat{k}+1} - \lambda^0_{\hat{k}+1}\| \end{aligned} \tag{73}$$

Letting $\varepsilon^{i+1}_{\hat{k}+1} = \lambda^{i+1}_{\hat{k}+1} - \lambda^i_{\hat{k}+1}$, by similar procedure used in (65)–(67), one can use (58) to derive

$$\begin{aligned} \|x^{\infty}_{\hat{k}+1} - x^1_{\hat{k}+1}\| &\leqslant \alpha_2\|\lambda^{\infty}_{\hat{k}+1} - \lambda^0_{\hat{k}+1}\| \\ &\leqslant \alpha_2\|\varepsilon^1_{\hat{k}+1}\|\tfrac{1}{1-\|\alpha_1\|} \end{aligned} \tag{74}$$

In (74), α_2 becomes arbitrary small by a correct choice of R.

Eqs. (69) and (74) show that the error between the states propagated for one time step along controllers trained by Algorithm 1 and Algorithm 2 is bounded. To derive such relations, it was assumed that the states propagated from time $\hat{k}$ to time $\hat{k}+1$ are initiated from the same values at time $\hat{k}$. To further continue the proof, consider time step $\hat{k} = 0$ to $\hat{k} = 1$. Considering (74), one can deduct the boundedness of $\delta = x^{\infty}_1 - x^1_1$which can become arbitrary small by the choice of R. Also, one notes that $x^{\infty}_1 = x^1_1 + \delta$. Now, consider time step $\hat{k} = 1$ to $\hat{k} = 2$. It follows

$$x^{\infty}_2 = f(x^{\infty}_1) + g(x^{\infty}_1)\left(-R^{-1}g^T(x^{\infty}_1)\lambda^{\infty}_2(x^{\infty}_2)\right) \tag{75}$$

$$x^1_2 = f(x^1_1) + g(x^1_1)\left(-R^{-1}g^T(x^1_1)\lambda^1_2(x^1_2)\right) \tag{76}$$

Through some algebraic manipulations, one can write (75) as

$$x^{\infty}_2 = f(x^1_1 + \delta) + g(x^1_1 + \delta)\left(-R^{-1}g^T(x^1_1 + \delta)\lambda^{\infty}_2(x^{\infty}_2)\right) \tag{77}$$

Subtracting (76) from (77) one has

$$\begin{aligned} x^{\infty}_2 - x^1_2 &= f(x^1_1 + \delta) - f(x^1_1) \\ &+ g(x^1_1 + \delta)\left(-R^{-1}g^T(x^1_1 + \delta)\lambda^{\infty}_2(x^{\infty}_2)\right) \\ &- g(x^1_1)\left(-R^{-1}g^T(x^1_1)\lambda^1_2(x^1_2)\right) \end{aligned} \tag{78}$$

Applying norms on both sides of (78), after some algebraic manipulations one has

$$\begin{aligned} \|x^{\infty}_2 - x^1_2\| &\leqslant \|f(x^1_1 + \delta) - f(x^1_1)\| \\ &+ \|g(x^1_1 + \delta)R^{-1}g^T(x^1_1 + \delta)\lambda^{\infty}_2(x^{\infty}_2) \\ &- g(x^1_1)R^{-1}g^T(x^1_1)\lambda^1_2(x^1_2)\| \end{aligned} \tag{79}$$

Considering $\psi^{\infty} = g(x^1_1 + \delta)R^{-1}g^T(x^1_1 + \delta)$ and $\psi^1 = g(x^1_1)R^{-1}g^T(x^1_1)$, one has

$$\begin{aligned} &\psi^{\infty}\lambda^{\infty}_2(x^{\infty}_2) - \psi^1\lambda^1_2(x^1_2) = \\ &(\psi^{\infty} - \psi^1)\lambda^{\infty}_2(x^{\infty}_2) + \psi^1(\lambda^{\infty}_2(x^{\infty}_2) - \lambda^1_2(x^1_2)) \end{aligned} \tag{80}$$

Applying norms on (80), one has

$$\begin{aligned} &\|\psi^{\infty}\lambda^{\infty}_2(x^{\infty}_2) - \psi^1\lambda^1_2(x^1_2)\| \leqslant \\ &\|\psi^{\infty} - \psi^1\|\|\lambda^{\infty}_2(x^{\infty}_2)\| + \|\psi^1\|\|\lambda^{\infty}_2(x^{\infty}_2) - \lambda^1_2(x^1_2)\| \end{aligned} \tag{81}$$

Through some algebraic manipulations, one has

$$\begin{aligned} \psi^{\infty} - \psi^1 = \quad &(g(x^1_1 + \delta) - g(x^1_1))R^{-1}g^T(x^1_1 + \delta) \\ &+ g(x^1_1)R^{-1}(g(x^1_1 + \delta) - g(x^1_1))^T \end{aligned} \tag{82}$$

Applying norms on (82), through further algebraic manipulations one has

$$\|\psi^{\infty} - \psi^1\| = 2\beta_4\rho_1^2\|R^{-1}\|\|\delta\| \tag{83}$$

where β_4 is selected as Lipschitz constant for $g(.)$ in Ω. Considering $\|\lambda^{\infty}_2(x^{\infty}_2)\| \leqslant \rho_4$, it is easy to see that $\|\psi^{\infty} - \psi^1\|\|\lambda^{\infty}_2(x^{\infty}_2)\| \leqslant 2\beta_4\rho_1^2\rho_4\|R^{-1}\|\|\delta\|$. Also, considering the second term on the right-hand side of (81) one has

$$\begin{aligned} &\|\psi^1\|\|\lambda^{\infty}_2(x^{\infty}_2) - \lambda^1_2(x^1_2)\| \leqslant \\ &\rho_1^2\|R^{-1}\|\|\lambda^{\infty}_2(x^{\infty}_2) - \lambda^1_2(x^1_2)\| \leqslant \\ &\rho_1^2\|R^{-1}\|\|\lambda^{\infty}_2(x^{\infty}_2) - \lambda^{\infty-1}_2(x^{\infty-1}_2) + \lambda^{\infty-1}_2(x^{\infty-1}_2) - \cdots \\ &+ \lambda^2_2(x^2_2) - \lambda^1_2(x^1_2)\| \leqslant \rho_1^2\|R^{-1}\|(\|\lambda^{\infty}_2(x^{\infty}_2) - \lambda^{\infty-1}_2(x^{\infty-1}_2)\| \\ &+ \|\lambda^{\infty-1}_2(x^{\infty-1}_2) - \lambda^{\infty-1}_2(x^{\infty-1}_2)\| + \cdots + \|\lambda^2_2(x^2_2) - \lambda^1_2(x^1_2)\|) \end{aligned} \tag{84}$$

Considering (58), with a similar procedure used to derive (74), one can re-write (84) as

$$\begin{aligned} &\|\psi^1\|\|\lambda^{\infty}_2(x^{\infty}_2) - \lambda^1_2(x^1_2)\| \leqslant \\ &\rho_1^2\|R^{-1}\|\|\lambda^2_2(x^2_2) - \lambda^1_2(x^1_2)\|\tfrac{1}{1-\|\alpha_1\|} \end{aligned} \tag{85}$$

Using (83) and (85) in (79) along Lipschitz assumption of $f(.)$ leads

$$\begin{aligned} \|x^{\infty}_2 - x^1_2\| &\leqslant \beta_1\|\delta\| + 2\beta_4\rho_1^2\rho_4\|R^{-1}\|\|\delta\| \\ &+ \rho_1^2\|R^{-1}\|\|\lambda^2_2(x^2_2) - \lambda^1_2(x^1_2)\|\tfrac{1}{1-\|\alpha_1\|} \end{aligned} \tag{86}$$

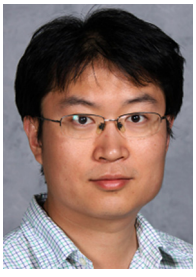
It can be seen that all the terms on the right hand side of (86) can become arbitrary small by the choice of R. Also, $\|\delta\|$ in the first term on the right-hand side of (86) can be calculated from (74) in which α_2 can become small through the choice of R.

With similar procedure, one can find the error for the rest of times. Since the time horizon is finite, such bounds for the error complete the proof. □.

References

[1] T. Sardarmehni, A. Heydari, Sub-optimal switching in anti-lock brake systems using approximate dynamic programming, IET Control Theory Appl. 13 (9) (2019) 1413–1424.
[2] M. Rinehart, M. Dahleh, D. Reed, I. Kolmanovsky, Suboptimal control of switched systems with an application to the DISC engine, IEEE Trans. Control Syst. Technol. 16 (2) (2008) 189–201.
[3] K.G. Vamvoudakis, J. Hespanha, Online optimal switching of single phase DC/AC inverters using partial information, American Control Conference (ACC) (2014) 2624–2630.
[4] Z. Gong, C. Liu, E. Feng, L. Wang, Y. Yu, Modelling and optimization for a switched system in microbial fed-batch culture, Appl. Math. Model. 35 (7) (2011) 3276–3284.
[5] M.-B. Radac, R.-E. Precup, Data-driven model-free slip control of anti-lock braking systems using reinforcement Q-learning, Neurocomputing 275 (2018) 317–329.
[6] A.G. Khiabani, A. Heydari, Design and implementation of an optimal switching controller for uninterruptible power supply inverters using adaptive dynamic programming, IET Power Electron. 12 (12) (2019) 3068–3076.
[7] X. Xu, P.J. Antsaklis, Optimal control of switched systems based on parameterization of the switching instants, IEEE Trans. Autom. Control 49 (1) (2004) 2–16.
[8] A. Heydari, S.N. Balakrishnan, Optimal switching and control of nonlinear switching systems using approximate dynamic programming, IEEE Trans. Neural Networks Learn. Syst. 25 (6) (2014) 1106–1117.
[9] M. Kamgarpour, C. Tomlin, On optimal control of non-autonomous switched systems with a fixed mode sequence, Automatica 48 (6) (2012) 1177–1181.
[10] A. Heydari, S. Balakrishnan, Optimal switching between controlled subsystems with free mode sequence, Neurocomputing 149 (2015) 1620–1630.
[11] W. Lu, P. Zhu, S. Ferrari, A hybrid-adaptive dynamic programming approach for the model-free control of nonlinear switched systems, IEEE Trans. Autom. Control 61 (10) (2016) 3203–3208.
[12] T. Sardarmehni, A. Heydari, Suboptimal scheduling in switched systems with continuous-time dynamics: a least squares approach, IEEE Trans. Neural Networks Learn. Syst. 29 (6) (2018) 2167–2178.
[13] T. Sardarmehni, A. Heydari, Sub-optimal scheduling in switched systems with continuous-time dynamics: a gradient descent approach, Neurocomputing 285 (2018) 10–22.
[14] M. Gan, C. Zhang, J. Zhao, Data-driven optimal switching of switched systems, J. Franklin Inst. 356 (10) (2019) 5193–5221.
[15] K. Zhang, H. Zhang, Y. Liang, Y. Wen, A new robust output tracking control for discrete-time switched constrained-input systems with uncertainty via a critic-only iteration learning method, Neurocomputing 396 (2019) 162–171.
[16] C. Zhang, M. Gan, J. Zhao, Data-driven optimal control of switched linear autonomous systems, Int. J. Syst. Sci. 50 (6) (2019) 1275–1289.
[17] D. Kirk, Optimal Control Theory: An Introduction, Dover Publications, Mineola, NY, USA, 2004.
[18] F. Lewis, D. Vrabie, Reinforcement learning and adaptive dynamic programming for feedback control, Circ. Syst. Mag., IEEE 9 (3) (2009) 32–50.
[19] A. Heydari, S. Balakrishnan, Fixed-final-time optimal tracking control of input-affine nonlinear systems, Neurocomputing 129 (2014) 528–539.
[20] B. Kiumarsi, F.L. Lewis, Actor-critic-based optimal tracking for partially unknown nonlinear discrete-time systems, IEEE Trans. Neural Networks Learn. Syst. 26 (1) (2015) 140–151.
[21] B. Luo, D. Liu, T. Huang, D. Wang, Model-free optimal tracking control via critic-only Q-learning, IEEE Trans. Neural Networks Learn. Syst. 27 (10) (2016) 2134–2144.
[22] B. Kiumarsi, F.L. Lewis, M. Naghibi-Sistani, A. Karimpour, Optimal tracking control of unknown discrete-time linear systems using input-output measured data, IEEE Trans. Cybern. 45 (12) (2015) 2770–2779.
[23] A. Heydari, S. Balakrishnan, Optimal switching between autonomous subsystems, J. Franklin Inst. 351 (5) (2014) 2675–2690.
[24] M. Rinehart, M. Dahleh, I. Kolmanovsky, Value iteration for (switched) homogeneous systems, IEEE Trans. Autom. Control 54 (6) (2009) 1290–1294.
[25] A. Heydari, Optimal scheduling for reference tracking or state regulation using reinforcement learning, J. Franklin Inst. 352 (8) (2015) 3285–3303.
[26] C. Seatzu, D. Corona, A. Giua, A. Bemporad, Optimal control of continuous-time switched affine systems, IEEE Trans. Autom. Control 51 (5) (2006) 726–741.
[27] M. Sakly, A. Sakly, N. Majdoub, M. Benrejeb, Optimization of switching instants for optimal control of linear switched systems based on genetic algorithms, IFAC Proceedings Volumes 42 (19) (2009) 249–253.
[28] T. Sardarmehni, X. Song, Sub-optimal tracking in switched systems with controlled subsystems and fixed-mode sequence using approximate dynamic programming, in: ASME 2019 Dynamic Systems and Control Conference (DSCC 2019), 2019, pp. V003T19A011–V003T19A017.
[29] D. Wang, M. Ha, J. Qiao, Self-learning optimal regulation for discrete-time nonlinear systems under event-driven formulation, IEEE Trans. Autom. Control 65 (3) (2020) 1272–1279.
[30] K. Zhang, H. Zhang, Y. Mu, C. Liu, Decentralized tracking optimization control for partially unknown fuzzy interconnected systems via reinforcement learning method, IEEE Trans. Fuzzy Syst. (2020) 1–11.
[31] Sub-Optimal Control of Autonomous Wheel Loader With Approximate Dynamic Programming, ASME 2019 Dynamic Systems and Control Conference (DSCC 2019), Volume 3, Park City, Utah, USA., October 8–11 2019.
[32] V. Nezhadali, B. Frank, L. Eriksson, Wheel loader operation–optimal control compared to real drive experience, Control Eng. Pract. 48 (2016) 1–9.
[33] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks 2 (5) (1989) 359–366.
[34] W. Rudin, Principles of Mathematical Analysis, third ed., McGraw-Hill, 1976.
[35] K. Sun, L. Liu, J. Qiu, G. Feng, Fuzzy adaptive finite-time fault-tolerant control for strict-feedback nonlinear systems, IEEE Trans. Fuzzy Syst. (2020) 1–11.
[36] K. Sun, J. Qiu, H.R. Karimi, H. Gao, A novel finite-time control for nonstrict feedback saturated nonlinear systems with tracking error constraint, IEEE Trans. Syst. Man Cybern. Syst. (2019) 1–12.
[37] R.W. Beard, G.N. Saridis, J.T. Wen, Galerkin approximations of the generalized hamilton-jacobi-bellman equation, Automatica 33 (12) (1997) 2159–2177.

Tohid Sardarmehni received his Ph.D. degree from Southern Methodist University in 2018. He was a Post-doc Research Scholar from 2018 to 2020 at Texas A&M University. He is currently an Assistant Professor at the University of Texas Rio Grande Valley. His research interests include optimal control, reinforcement learning, and autonomous systems.



Xingyong Song received the Ph.D. degree at the Mechanical Engineering Department, University of Minnesota, Twin Cities Campus, in 2011. He was a principal research engineer (2013–2015) at Innovation Group at Halliburton Energy Services Company in Houston, TX, and a research engineer (2011–2013) at General Motors Research and Development Center in Warren, MI, respectively. He is currently an Assistant Professor at Texas A&M University, College Station, TX. His research interests include time-varying internal model based control, optimal control, robust LPV control, nonlinear control systems, approximate dynamics programming, barrier avoidance control and their applications in autonomous vehicles, renewable energy systems and novel technologies in oil/gas industry.