10-4-2021

Optimal Tracking in Switched Systems With Free Final Time and Fixed Mode Sequence Using Approximate Dynamic Programming

Tohid Sardarmehni
The University of Texas Rio Grande Valley

Xingyong Song
Texas A&M University

Recommended Citation

T. Sardarmehni and X. Song, "Optimal Tracking in Switched Systems With Free Final Time and Fixed Mode Sequence Using Approximate Dynamic Programming," in IEEE Transactions on Neural Networks and Learning Systems, doi: 10.1109/TNNLS.2021.3113801.

Optimal Tracking in Switched Systems with Free Final Time and Fixed Mode Sequence using Approximate Dynamic Programming

Tohid Sardarmehni[1] and Xingyong Song[2,*]

***Abstract*—Optimal tracking in switched systems with fixed mode sequence and free final time is studied in this paper. In the optimal control problem formulation, the switching times and the final time are treated as parameters. For solving the optimal control problem, approximate dynamic programming is used. The approximate dynamic programming solution uses an inner loop to converge to the optimal policy at each time step. In order to decrease the computational burden of the solution, a new method is introduced which uses evolving suboptimal policies (not the optimal policies), to learn the optimal solution. The effectiveness of the proposed solutions is evaluated through numerical simulations.**

Index Terms- optimal control, switched systems, fixed mode sequence, free final time.

I. INTRODUCTION

In this study, optimal tracking in a class of hybrid systems comprised of a finite number of subsystems/modes is studied. It is assumed that at each time instant, only one subsystem is active. Furthermore, it is assumed that the sequence of active modes is fixed and known. Therefore, the role of control is assigning the switching times and the input control for the active subsystem such that the system tracks the desired trajectory.

In optimal control, control signals are generated through minimization of a cost function subject to (possible) input or state constraints. As a closed loop and feedback solution for the optimal control problem, Dynamic Programming (DP) was introduced [1]. In general, DP provides a systematic solution for optimal control problems. However, as the order of the system increases, the required memory and time to preform DP grow exponentially, which is called the *curse of dimensionality* in DP [1].

In order to remedy the curse of dimensionality in DP, Approximate Dynamic Programming (ADP) was later introduced. In summary, ADP methods use function approximators to approximate the optimal value function, namely critic, and sometimes the optimal policy, namely actor. ADP methods then use iterative schemes to tune the parameters of these function approximators through training [2]. On top of the ability to handle the curse of dimensionality, ADP methods can solve the optimal control problems forward in time, which is suitable for online training [2].

Several ADP methods are investigated for the optimal control of switched systems with free mode sequence [3]–[13]. The optimal control of switched systems with fixed mode sequence using ADP was studied in [14] by introducing a transformation to include the switching instants as parameters. This transformation was used in [15], and the mode sequence was included in the value function approximation to develop an ADP solution. In another approach, the gradient of the value function with respect to the switching time was used in [5] to find the optimal switching times. Also, the same transformation was used in [16] and the set of switching times was considered as parameters in the costates to design a controller to solve the optimal tracking problem.

Some non-ADP theoretical developments were conducted in [17], which introduced a method to solve the optimal switching problem in switched systems with fixed mode sequences. Also, [18] discretized the control space and considered piecewise constant control signals to solve for optimal switching times and the optimal controls. A novel structure for optimal control of switched systems with free mode sequence was studied in [19]. In [19], an embedded system is first formulated by introducing convex combinations of the subsystems and the constraints. The necessary and sufficient conditions for optimality of the corresponding embedded system and the original system are discussed. In [20], optimal control of switched systems was discussed with state jump in switching. Lastly, free final time in the optimal control of switched systems was considered in [21]–[23]. In [21], an embedded system is first introduced which includes the switching time and the optimization problem is formulated afterward. It is shown that the method can be extended to higher number of subsystems. In [22], a good review of advances in optimal control of switched systems is provided. In [23], optimal control of switched systems was studied. The authors considered the infinite horizon cost function along the worst and the expected costs to derive the solutions.

The above-mentioned studies related to ADP solutions dealt with optimal control of switched systems with fixed mode sequence and fixed final time. To the best of our knowledge, ADP-based solutions for optimal tracking of switched systems with fixed mode sequence and free final time has not been studied before. Hence, the goal in this paper is finding the optimal switching times, optimal final time, and optimal control such that a cost function is minimized and the system tracks the reference signal. The backbone of the solution is using the transformation introduced in [14] to include both

[1]Assistant Professor of Mechanical Engineering, University of Texas Rio Grande Valley, Edinburg, TX 78539, USA tohid.sardarmehni@utrgv.edu

[2]Department of Engineering Technology and Industrial Distribution; Department of Mechanical Engineering; College of Engineering, Texas A&M University, College Station, TX, 77843, USA. songxy@ tamu.edu (Corresponding Author)

the switching times and the final time as parameters in the optimal control problem formulation. Then, a Single Network Adaptive Critic (SNAC) method [24] is used to develop an ADP solution for optimal tracking. It is important to note that the methods discussed in [24] does not extend naturally to include switching dynamics. In general, solving a control problem with switching dynamics is a more challenging task than a control problem in systems with conventional dynamics such as [24]. In order to reduce the computational burden and speed up the calculations, the effect of using evolving suboptimal policies in the system is investigated. Hence, a new algorithm is introduced, which uses evolving suboptimal policies to learn the optimal control solution. To summarize, the contributions of this paper are as follows[1].

- An ADP solution for optimal tracking in switched systems with fixed mode sequence and free final time is introduced.
- A new algorithm is introduced, which uses evolving suboptimal policies to learn the optimal control solution.

The proposed solutions in this paper lead to a two-level optimization similar to [14], [16], [24]. In the upper level, the switching times are sought, and in the lower level, the optimal policy is sought. As mentioned before, compared to [5], [14]–[16] which the fixed final time problem was considered, in the current study free final time problem is investigated. Also, in this paper tracking problem is investigated whereas in [5], [14], [15] stabilization problem was investigated.

The rest of the manuscript is organized as follows. In Section II, the optimal control problem formulation and some assumptions are presented. In Section III, the main solution is presented. The effect of evolving suboptimal policies is investigated in Section IV. Numerical simulations are discussed in Section V, and Section VI concludes the paper.

II. Problem Formulation

The dynamics of a switched system can be shown as

$$\begin{aligned} \dot{x}(t) &= \bar{f}_v\big(x(t)\big) + \bar{g}_v\big(x(t)\big)u(t), \\ v &\in \mathscr{V} = \{1,2,\ldots,M\},\ x(t_0) = x_0 \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the input, and t denotes the time. The smooth functions $\bar{f}_v : \mathbb{R}^n \to \mathbb{R}^n$ and $\bar{g}_v : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ denote the dynamics of the subsystems. The sub-index v portrays the active mode which can be selected from the set of all available modes, $\mathscr{V}$, in the system. It is further assumed that $\bar{f}_v(0) = 0$, for all modes $v \in \mathscr{V}$. Assuming the sequence of active modes is known, it is desired to find the continuous control $u(.)$, and the switching times, and the final time such that a performance index presented as

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

is minimized. In (2), t_0 is the initial time, t_f is the unknown final time, and $r \in \mathbb{R}^n$ is the reference signal. $S \in \mathbb{R}^{n\times n}$ is a positive semi-definite matrix for penalizing the terminal cost, $\bar{Q} \in \mathbb{R}^{n\times n}$ is the state penalizing matrix which is assumed to be positive semi-definite, and $\bar{R} \in \mathbb{R}^{m\times m}$ is a positive definite control penalizing matrix. The dynamics of the reference signal can be presented as

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

where $\bar{f}_{r_v} : \mathbb{R}^n \to \mathbb{R}^n$ denotes the dynamics of the desired trajectory for the active mode v. Using the Euler integration method, by choosing a small sample time $\delta t > 0$, one can discretize the dynamics presented in (1) as

$$\begin{aligned} x_{k+1} &= f_v(x_k) + g_v(x_k)u_k \\ v &\in \mathscr{V} = \{1,2,\ldots,M\},\ x(0) = x_0 \end{aligned} \tag{4}$$

where the non-negative integer k is the discrete time index. For notational simplicity, the discrete time index is shown as a sub-index, i.e., $x_k \equiv x(k)$. Also, $f_v(x_k) = x_k + \bar{f}_v(x_k)\delta t$, and $g_v(x_k) = \bar{g}_v(x_k)\delta t$. With a similar procedure, one can discretize the cost function (2) as

$$\begin{aligned} J(x_0, r_0) &= \frac{1}{2}(x_N - r_N)^T S(x_N - r_N) \\ &+ \sum_{k=0}^{N-1} \frac{1}{2}\Big(\big(x_k - r_k\big)^T Q\big(x_k - r_k\big) + u_k^T R u_k\Big) \end{aligned} \tag{5}$$

In (5), $N = \frac{t_f - t_0}{\delta t}$, $Q = \bar{Q}\delta t$, and $R = \bar{R}\delta t$. Based on (5), one can define the cost-to-go as the cost of going from discrete time index k to N as

$$\begin{aligned} J(x_k, r_k) &= \frac{1}{2}\big(x_N - r_N\big)^T S\big(x_N - r_N\big) \\ &+ \sum_{\bar{k}=k}^{N-1} \frac{1}{2}\Big((x_{\bar{k}} - r_{\bar{k}})^T Q(x_{\bar{k}} - r_{\bar{k}}) + u_{\bar{k}}^T R u_{\bar{k}}\Big) \end{aligned} \tag{6}$$

Before going forward, the following definition and assumption are required.

Definition 1: A control policy is admissible if it stabilizes the system presented in (4) in a selected compact region of interest $\Omega \subset \mathbb{R}^n$, which includes the origin. Also, $\forall x_0 \in \Omega$ and $\forall r_0 \in \Omega$, $J(x_0, r_0)$ is finite if the state is controlled using that policy.

Remark 1: In this study, we consider the optimal control solution for discrete-time dynamics. Studying the optimal control solutions in systems with continuous-time dynamics follows a different path and is out of the scope of this paper. Interested readers are referred to [1], [2] for a brief comparison between continuous-time and discrete-time optimal control solutions.

Assumption 1: Given the mode sequence, there is at least one admissible policy for the system.

Assumption 1 is a controllability-like assumption which ensures the existence of at least one admissible policy. Considering Assumption 1, one can define the value function as

$$\begin{aligned} V\big(x_k, r_k, k\big) \equiv V_k(x_k, r_k) = \min_{u(.)} \Bigg(&\frac{1}{2}(x_N - r_N)^T S(x_N - r_N) \\ &+ \frac{1}{2}\sum_{\bar{k}=k}^{N-1} \Big((x_{\bar{k}} - r_{\bar{k}})^T Q(x_{\bar{k}} - r_{\bar{k}}) + u_{\bar{k}}^T R u_{\bar{k}}\Big)\Bigg) \end{aligned} \tag{7}$$

Considering time step k to $k+1$, after some algebraic manip-

[1]The preliminary results of this paper were presented in ASME 2019 Dynamic System and Control Conference [25].

ulations one has

$$V_k(x_k,r_k)=\min_{u_k(.)}\Big(\frac{1}{2}(x_k-r_k)^TQ(x_k-r_k)+\frac{1}{2}u_k^TRu_k + V_{k+1}(x_{k+1},r_{k+1})\Big) \tag{8}$$

Equation (8) simply means the minimum cost of going from time k to N equals to the minimum cost of going from time k to $k+1$ plus the minimum cost of going from time $k+1$ to N. This is in fact the Bellman equation of optimality [1]. Based on (8), one can define the optimal policy as

$$u_k(x_k)=\arg\min_{u_k(.)}\Big(\frac{1}{2}(x_k-r_k)^TQ(x_k-r_k)+\frac{1}{2}u_k^TRu_k + V_{k+1}(x_{k+1},r_{k+1})\Big) \tag{9}$$

At this point, consider the Hamiltonian as

$$\mathscr{H}(x_k,r_k,u_k,V_{k+1})= \frac{1}{2}(x_k-r_k)^TQ(x_k-r_k)+\frac{1}{2}u_k^TRu_k+V_{k+1}(x_{k+1},r_{k+1}) \tag{10}$$

In the optimal control problems, the minimizer of the Hamiltonian, solves the optimal control problem [1]. Using the necessary condition for optimality, i.e., $\frac{\partial\mathscr{H}(.)}{\partial u_k}=0$, one can find the optimal policy as

$$u_k^*(x_k)=-R^{-1}g^T(x_k)\frac{\partial V_{k+1}}{\partial x}|_{x=x_{k+1}} \tag{11}$$

As one can see from (11), in case the optimal value function $V(.,.)$ is known, one can easily find the optimal policy.

A. Including the Mode Sequence

For the sake of simplicity in presenting the main solution, a switched system with two modes and one switching is considered. It is assumed that the switching happens at $t=t_1$ and the mode sequence is $\{mode\ 1,\ mode\ 2\}$. Hence, one has

$$\dot{x}(t)=\begin{cases}\bar{f}_1(x)+\bar{g}_1(x)u(t) & \text{if } t_0\le t<t_1\\ \bar{f}_2(x)+\bar{g}_2(x)u(t) & \text{if } t_1\le t\le t_f\end{cases} \tag{12}$$

To include the switching times as parameters in the optimal control formulation, the following transformation is used [14].

$$t=\begin{cases}t_0+(t_1-t_0)\hat{t} & \text{if } 0\le\hat{t}<1\\ t_1+(t_f-t_1)(\hat{t}-1) & \text{if } 1\le\hat{t}\le 2\end{cases} \tag{13}$$

From the transformation introduced in (13), one notices that $t\in[t_0,t_f]$ and $\hat{t}\in[0,2]$. The merit of the transformation is that the switching time t_1 can be any point in $t\in[t_0,t_f]$. However, in the transformed time, i.e., $\hat{t}\in[0,2]$, switching only happens at $\hat{t}=1$. For $0\le\hat{t}<1$, the first mode is active, and for $1\le\hat{t}\le 2$ the second mode is active. This procedure can be easily extended to systems with more modes and switchings. Based on the introduced transformation, using chain rule leads

$$x'(\hat{t})=\frac{dx}{d\hat{t}}=\frac{dx}{dt}\frac{dt}{d\hat{t}} \tag{14}$$

Since the mode sequence is known, (14) becomes

$$x'(\hat{t})=\begin{cases}(\bar{f}_1(x)+\bar{g}_1(x)u)(t_1-t_0) & \text{if } 0\le\hat{t}<1\\ (\bar{f}_2(x)+\bar{g}_2(x)u)(t_f-t_1) & \text{if } 1\le\hat{t}\le 2\end{cases} \tag{15}$$

In (15), $x\equiv x(\hat{t})$ and $u\equiv u(\hat{t})$ for notational simplicity. Also, the cost function in (2) can be written as [15][1]

$$\begin{aligned}J(x_0,r_0)&=\frac{1}{2}\big(x(2)-r(2)\big)^TS\big(x(2)-r(2)\big)\\&+\frac{1}{2}\int_0^1\big((x-r)^T\bar{Q}(t_1-t_0)(x-r)+u^T\bar{R}(t_1-t_0)u\big)d\hat{t}\\&+\frac{1}{2}\int_1^2\big((x-r)^T\bar{Q}(t_f-t_1)(x-r)+u^T\bar{R}(t_f-t_1)u\big)d\hat{t}\end{aligned} \tag{16}$$

In (16), the dependency of x, r, and u to $\hat{t}$ is dropped for notational simplicity. An important observation in (16) is that the transformed cost function is not only a function of x_0 and r_0, but also it is a function of the switching times, i.e., $\Gamma=\{t_1\}$, and the final time t_f. Hence, $J(.,.,.,.)=J(\Gamma,t_f,x_0,r_0)$. With a similar procedure used before, by choosing a small sampling time $\delta\hat{t}$ one can discretize (15) and (16) as

$$x_{\hat{k}+1}=\begin{cases}f_1(x_{\hat{k}})+g_1(x_{\hat{k}})u_{\hat{k}} & \text{if } 0\le\hat{k}\delta\hat{t}<1\\ f_2(x_{\hat{k}})+g_2(x_{\hat{k}})u_{\hat{k}} & \text{if } 1\le\hat{k}\delta\hat{t}\le 2\end{cases} \tag{17}$$

$$\begin{aligned}J(\Gamma,t_f,x_0,r_0)&=\frac{1}{2}\big(x_{N'}-r_{N'}\big)^TS\big(x_{N'}-r_{N'}\big)\\&+\frac{1}{2}\sum_{\bar{k}=1}^{1/\delta\hat{t}}\big((x_{\bar{k}}-r_{\bar{k}})^TQ_1(x_{\bar{k}}-r_{\bar{k}})+u_{\bar{k}}^TR_1u_{\bar{k}}\big)\\&+\frac{1}{2}\sum_{\bar{k}=1/\delta\hat{t}}^{N'-1}\big((x_{\bar{k}}-r_{\bar{k}})^TQ_2(x_{\bar{k}}-r_{\bar{k}})+u_{\bar{k}}^TR_2u_{\bar{k}}\big)\end{aligned} \tag{18}$$

In (17),

$$\begin{aligned}f_1(x_{\hat{k}})&=x(\hat{t})+\bar{f}_1\big(x(\hat{t})\big)(t_1-t_0)\delta\hat{t}\\g_1(x_{\hat{k}})&=\bar{g}_1\big(x(\hat{t})\big)(t_1-t_0)\delta\hat{t}\\f_2(x_{\hat{k}})&=x(\hat{t})+\bar{f}_2\big(x(\hat{t})\big)(t_f-t_1)\delta\hat{t}\\g_2(x_{\hat{k}})&=\bar{g}_2\big(x(\hat{t})\big)(t_f-t_1)\delta\hat{t}\end{aligned}$$

Similarly, in (18)

$$\begin{aligned}Q_1&=\bar{Q}(t_1-t_0)\delta\hat{t}\\Q_2&=\bar{Q}(t_f-t_1)\delta\hat{t}\\R_1&=\bar{R}(t_1-t_0)\delta\hat{t}\\R_2&=\bar{R}(t_f-t_1)\delta\hat{t}\end{aligned}$$

In both (17) and (18), $\hat{k}\in[0,N']$ is the discrete time index, and $N'=\frac{p+1}{\delta\hat{t}}$ where p is the number of switchings [15]. Considering (17) and (18), through a similar procedure used in the previous section one can define the value function as

$$V_{\hat{k}}(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}})= \begin{cases}\mathbf{Q}_1+\mathbf{R}_1+V_{\hat{k}+1}(\Gamma,t_f,x_{\hat{k}+1},r_{\hat{k}+1}) & \text{if } 0\le\hat{k}\delta\hat{t}<1\\ \mathbf{Q}_2+\mathbf{R}_2+V_{\hat{k}+1}(\Gamma,t_f,x_{\hat{k}+1},r_{\hat{k}+1}) & \text{if } 1\le\hat{k}\delta\hat{t}\le 2\end{cases} \tag{19}$$

where

$$\begin{aligned}\mathbf{Q}_1&=\frac{1}{2}\big(x_{\hat{k}}-r_{\hat{k}}\big)^T\bar{Q}(t_1-t_0)\delta\hat{t}\big(x_{\hat{k}}-r_{\hat{k}}\big)\\\mathbf{R}_1&=\frac{1}{2}u_{\hat{k}}^T\bar{R}(t_1-t_0)\delta\hat{t}u_{\hat{k}}\\\mathbf{Q}_2&=\frac{1}{2}\big(x_{\hat{k}}-r_{\hat{k}}\big)^T\bar{Q}(t_f-t_1)\delta\hat{t}\big(x_{\hat{k}}-r_{\hat{k}}\big)\\\mathbf{R}_2&=\frac{1}{2}u_{\hat{k}}^T\bar{R}(t_f-t_1)\delta\hat{t}u_{\hat{k}}\end{aligned}$$

As one can see, the value function in (19) is a function of

[1]Since the mode sequence is known, one can consider the integral from t_0 to t_1 with the first mode, and from t_1 to t_f with the second mode.

current time $\hat{k}$, current state $x_{\hat{k}}$, the current reference signal $r_{\hat{k}}$, the switching times Γ, and the final time t_f. To define the costates as the gradient of the value functions, the following assumption is required.

Assumption 2: The value functions are smooth.

Assumption 2 not only helps with formulations of the optimal control problem, but also helps with the possibility of using function approximators to approximate the value functions/costates in the proceeding sections. Interested readers are referred to Remark 1 in [7] for more discussions.

Through Assumption 2, one can define the costate as

$$\lambda_{\hat{k}}(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}}) = \begin{cases} \mathscr{Q}_{1_{\hat{k}}} + (\frac{\partial x_{\hat{k}+1}}{\partial x_{\hat{k}}})^T \lambda_{\hat{k}+1}(\Gamma,t_f,x_{\hat{k}+1},r_{\hat{k}+1}) & \text{if } 0 \le \hat{k}\delta\hat{t} < 1 \\ \mathscr{Q}_{2_{\hat{k}}} + (\frac{\partial x_{\hat{k}+1}}{\partial x_{\hat{k}}})^T \lambda_{\hat{k}+1}(\Gamma,t_f,x_{\hat{k}+1},r_{\hat{k}+1}) & \text{if } 1 \le \hat{k}\delta\hat{t} \le 2 \end{cases} \tag{20}$$

where

$$\mathscr{Q}_{1_{\hat{k}}} = \bar{Q}(t_1 - t_0)\delta\hat{t}\big(x_{\hat{k}} - r_{\hat{k}}\big)$$
$$\mathscr{Q}_{2_{\hat{k}}} = \bar{Q}(t_f - t_1)\delta\hat{t}\big(x_{\hat{k}} - r_{\hat{k}}\big)$$

Considering (11), it is straightforward to see that the optimal policy can be formulated with $\lambda_{\hat{k}+1}$ as

$$u^*_{\hat{k}}(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}}) = \begin{cases} -R_1^{-1} g_1^T\big(x_{\hat{k}}\big)\lambda_{\hat{k}+1}(\Gamma,t_f,x_{\hat{k}+1},r_{\hat{k}+1}) & \text{if } 0 \le \hat{k}\delta\hat{t} < 1 \\ -R_2^{-1} g_2^T\big(x_{\hat{k}}\big)\lambda_{\hat{k}+1}(\Gamma,t_f,x_{\hat{k}+1},r_{\hat{k}+1}) & \text{if } 1 \le \hat{k}\delta\hat{t} \le 2 \end{cases} \tag{21}$$

where $R_1 = \bar{R}\delta\hat{t}(t_1 - t_0)$ and $R_2 = \bar{R}\delta\hat{t}(t_f - t_1)$. Similar to (20), one can define $\lambda_{\hat{k}+1}$ as

$$\lambda_{\hat{k}+1}(\Gamma,t_f,x_{\hat{k}+1},r_{\hat{k}+1}) = \begin{cases} \mathscr{Q}_{1_{\hat{k}+1}} + (\frac{\partial x_{\hat{k}+2}}{\partial x_{\hat{k}+1}})^T \lambda_{\hat{k}+2}(\Gamma,t_f,x_{\hat{k}+2},r_{\hat{k}+2}) & \text{if } 0 \le \hat{k}\delta\hat{t} < 1 \\ \mathscr{Q}_{2_{\hat{k}+1}} + (\frac{\partial x_{\hat{k}+2}}{\partial x_{\hat{k}+1}})^T \lambda_{\hat{k}+2}(\Gamma,t_f,x_{\hat{k}+2},r_{\hat{k}+2}) & \text{if } 1 \le \hat{k}\delta\hat{t} \le 2 \end{cases} \tag{22}$$

For solving a regulation optimal control problem, [14] suggests a non-ADP solution, which includes two levels of optimization. In the upper level, switching times are sought, and in the lower level, control policies are sought. Hence, [14] suggests to form the optimal value function with the unknown switching time as the parameter and then using nonlinear programming to find the optimal switching time. In [15], an ADP solution is developed for the fixed final time problem, which includes the switching time instant in the critic neural network along with the state vector and the time. Then, [15] introduces a Dual Heuristic Dynamic Programming (DHP) solution, including critic networks to find value functions and actor networks to capture optimal policies. When the training is concluded, [15] suggests finding the unknown switching instants with constrained optimization methods for each initial condition x_0. Based on these two ideas, a solution is proposed in the next section for solving the tracking problem.

The merits of the proposed solution in this paper are as follows. Firstly, the proposed solution in Section III trains only one network for predicting the costates $\lambda_{\hat{k}+1}$ from the available information at the present time. This can potentially lead to an improvement in the speed of calculations since less number of networks needs to be trained compared to DHP methods used in [15]. Meanwhile, the proposed solution in this paper tries to alleviate the dependency of the training algorithms on the magnitude of the discretization sampling time. Lastly, the proposed solution in Section III is the backbone of the new solution presented in Section IV, which aims to speed up the derivation of the optimal control solution significantly.

III. Main Solution

The application of Single Network Adaptive Critic (SNAC) for tracking was introduced in [24] for systems with conventional dynamics. This idea is adapted in this section to perform tracking in switched systems. To introduce the proposed solution, consider the costate as in (22). The idea here is solving (22) backward in time and storing the optimal costates at each time instant. For this purpose, one can train neural networks to approximate $\lambda_{\hat{k}+1}(\Gamma,t_f,x_{\hat{k}+1},r_{\hat{k}+1})$ from $(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}})$ and solve (22) backward in time. Considering Assumption 2 and Weierstrass Approximation Theorem [26], one can use linear-in-parameter neural networks with polynomial basis functions to approximate the costates. Let the exact costate at the discrete-time index $\hat{k}+1$ be presented as

$$\lambda_{\hat{k}+1}(\Gamma,t_f,x_{\hat{k}+1},r_{\hat{k}+1}) = {W^*_{\hat{k}}}^T \phi(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}}) + \varepsilon^*_{\hat{k}}(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}}) \tag{23}$$

where $W^*_k \in \mathbb{R}^{m_\lambda \times n}$ is a weight vector and $\phi : \mathbb{R}^p \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{m_\lambda}$ is a vector of linearly independent polynomial basis functions (neurons). The number of neurons is denoted by positive integer m_λ. Also, $\varepsilon^*_{\hat{k}} : \mathbb{R}^p \times \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is the error for approximating the exact costate at the discrete time instant $\hat{k}$. In (23), the dependence of the parameters/functions to the discrete-time index is shown with a sub-index $\hat{k}$. One notes that the neural network used in (23) is *simply* the polynomial expansion of costates. Also, let the approximate costates be

$$\widehat{\lambda}_{\hat{k}+1}(\Gamma,t_f,x_{\hat{k}+1},r_{\hat{k}+1}) = \widehat{W}^T_{\hat{k}} \phi(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}}) \tag{24}$$

where $\widehat{W}_{\hat{k}} \in \mathbb{R}^{m_\lambda \times n}$ is a tunable weight vector. Once the approximate costates are known, one finds the optimal policy as

$$\widehat{u}_{\hat{k}}(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}}) = \begin{cases} -R_1^{-1} g_1^T\big(x_{\hat{k}}\big)\widehat{\lambda}_{\hat{k}+1}(\Gamma,t_f,x_{\hat{k}+1},r_{\hat{k}+1}) & \text{if } 0 \le \hat{k}\delta\hat{t} < 1 \\ -R_2^{-1} g_2^T\big(x_{\hat{k}}\big)\widehat{\lambda}_{\hat{k}+1}(\Gamma,t_f,x_{\hat{k}+1},r_{\hat{k}+1}) & \text{if } 1 \le \hat{k}\delta\hat{t} \le 2 \end{cases} \tag{25}$$

In (25), $R_1 = \bar{R}\delta\hat{t}(t_1 - t_0)$ and $R_2 = \bar{R}\delta\hat{t}(t_f - t_1)$. From (24), it is straightforward to see that $\widehat{\lambda}_{\hat{k}}(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}}) = \widehat{W}^T_{\hat{k}-1}\phi(\Gamma,t_f,x_{\hat{k}-1},r_{\hat{k}-1})$. Therefore, considering time steps $\hat{k}-1$ and $\hat{k}$, by substituting (24) in (20) one has

$$\widehat{W}^T_{\hat{k}-1}\phi(\Gamma,t_f,x_{\hat{k}-1},r_{\hat{k}-1}) = \begin{cases} \mathscr{Q}_{1_{\hat{k}}} + (\frac{\partial x_{\hat{k}+1}}{\partial x_{\hat{k}}})^T \widehat{W}^T_{\hat{k}} \phi(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}}) & \text{if } 0 \le \hat{k}\delta\hat{t} < 1 \\ \mathscr{Q}_{2_{\hat{k}}} + (\frac{\partial x_{\hat{k}+1}}{\partial x_{\hat{k}}})^T \widehat{W}^T_{\hat{k}} \phi(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}}) & \text{if } 1 \le \hat{k}\delta\hat{t} \le 2 \end{cases} \tag{26}$$

Also, one notes that since the mode sequence is known, one can find the costates at $\hat{k} = N'$ as

$$\lambda_{N'}(\Gamma,t_f,x_{N'},r_{N'}) = S(x_{N'} - r_{N'}) \tag{27}$$

Substituting (24) in (27), one has

$$\widehat{W}^T_{N'-1}\phi(\Gamma,t_f,x_{N'-1},r_{N'-1}) = S(x_{N'} - r_{N'}) \tag{28}$$

Lastly, by substituting the approximate costates as (24) in (25), one can find the approximate optimal policy as

$$\widehat{u}_{\hat{k}}(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}}) = \begin{cases} -R_1^{-1}g_1^T(x_{\hat{k}})\widehat{W}_{\hat{k}}^T\phi(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}}) & \text{if } 0\le\hat{k}\delta\hat{t}<1 \\ -R_2^{-1}g_2^T(x_{\hat{k}})\widehat{W}_{\hat{k}}^T\phi(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}}) & \text{if } 1\le\hat{k}\delta\hat{t}\le 2 \end{cases} \tag{29}$$

Remark 2: For simplicity in presentation, a system with two modes and only one switching was considered. However, extension of the results from two modes and one switching to multiple modes and multiple switching is straightforward. For instance, consider a system with p number of switching. The set of switching times are $\{t_0,t_1,\cdots,t_p\}$. The transformation in (13) can be rewritten as

$$t = \begin{cases} t_0+(t_1-t_0)\hat{t} & \text{if } 0\le\hat{t}<1 \\ t_1+(t_2-t_1)(\hat{t}-1) & \text{if } 1\le\hat{t}<2 \\ \vdots & \vdots \\ t_p+(t_f-t_p)(\hat{t}-p) & \text{if } p\le\hat{t}\le p+1 \end{cases} \tag{30}$$

Similarly, by using (30), and considering the known sequence of active modes as $\{v_{t_0},v_{t_1},v_{t_2},\cdots,v_{t_p}\}$, one can rewrite the dynamics in (17) as

$$x_{\hat{k}+1} = \begin{cases} f_{v_{t_0}}(x_{\hat{k}})+g_{v_{t_0}}(x_{\hat{k}})u_{\hat{k}} & \text{if } 0\le\hat{k}\delta\hat{t}<1 \\ f_{v_{t_1}}(x_{\hat{k}})+g_{v_{t_1}}(x_{\hat{k}})u_{\hat{k}} & \text{if } 1\le\hat{k}\delta\hat{t}<2 \\ \vdots & \vdots \\ f_{v_{t_p}}(x_{\hat{k}})+g_{v_{t_p}}(x_{\hat{k}})u_{\hat{k}} & \text{if } p\le\hat{k}\delta\hat{t}\le p+1 \end{cases} \tag{31}$$

In (31), $f_{v_{t_i}}$ and $g_{v_{t_i}}$ define the dynamics of the active subsystem at time $t\in[t_i,t_{i+1}]$ which are

$$f_{v_{t_i}}(x_{\hat{k}}) = x(\hat{t})+\bar{f}_{v_{t_i}}(x(\hat{t}))(t_{i+1}-t_i)\delta\hat{t}$$
$$g_{v_{t_i}}(x_{\hat{k}}) = \bar{g}_{v_{t_i}}(x(\hat{t}))(t_{i+1}-t_i)\delta\hat{t}$$

With a similar procedure, one can find the value functions and the costates for higher number of switchings.

A. Training Algorithm

For training, one can go backward in time and find the costates, i.e., $\widehat{W}_{\hat{k}}$s, and save them for online control. The costates can be calculated through solving (26) and (28). However, by looking at the right-hand side of (26) and (28) one notices that in using such solution, at each time instant $\hat{k}$, one needs to control the states using policy $u_{\hat{k}}(.)$ which is unknown.

For solving such problem in optimal tracking of systems with conventional dynamics, an inner loop is introduced in [24]. This idea can be adapted for switched systems with fixed mode sequences.

Let the iteration index as i. Considering (27), one has

$$\lambda_{N'}^{i+1}(\Gamma,t_f,x_{N'},r_{N'}) = S(x_{N'}^i-r_{N'}) \tag{32}$$

In (33), $x_{N'}^i$ is the state controlled with policy u^i from time instant $\hat{k}=N'-1\rightarrow N'$. Substituting from (24) in (32), at time instant $\hat{k}=N'-1$ the inner loop can be defined as

$$\widehat{W}_{N'-1}^{i+1^T}\phi(\Gamma,t_f,x_{N'-1},r_{N'-1}) = S(x_{N'}^i-r_{N'}) \tag{33}$$

Hence, the inner loop starts with a random initial guess for $\lambda_{N'}^{i=0}$, i.e., $\widehat{W}_{N'-1}^0$. With $\widehat{W}_{N'-1}^0$, one finds u^0 and uses this policy to control the states and find $x_{N'}^0$. Using (33), one finds $\widehat{W}_{N'-1}^1$. This process continues until the weights converges. After calculating $\widehat{W}_{N'-1}$, one can go backward in time to find the rest of the costates. The inner loop for finding the rest of the costates can be presented as

$$\lambda_{\hat{k}+1}^{i+1}(\Gamma,t_f,x_{\hat{k}+1},r_{\hat{k}+1}) = \begin{cases} \mathscr{Q}_{1_{\hat{k}+1}}^i+(\frac{\partial x_{\hat{k}+2}}{\partial x_{\hat{k}+1}})^T\lambda_{\hat{k}+2}(\Gamma,t_f,x_{\hat{k}+2}^i,r_{\hat{k}+2}) & \text{if } 0\le\hat{k}\delta\hat{t}<1 \\ \mathscr{Q}_{2_{\hat{k}+1}}^i+(\frac{\partial x_{\hat{k}+2}}{\partial x_{\hat{k}+1}})^T\lambda_{\hat{k}+2}(\Gamma,t_f,x_{\hat{k}+2}^i,r_{\hat{k}+2}) & \text{if } 1\le\hat{k}\delta\hat{t}\le 2 \end{cases} \tag{34}$$

Similar to (33), substituting from (24) one has

$$\widehat{W}_{\hat{k}}^{i+1^T}\phi(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}}) = \begin{cases} \mathscr{Q}_{1_{\hat{k}+1}}^i+(\frac{\partial x_{\hat{k}+2}}{\partial x_{\hat{k}+1}})^T\widehat{W}_{\hat{k}+1}^T\phi(\Gamma,t_f,x_{\hat{k}+1}^i,r_{\hat{k}+1}) & \text{if } 0\le\hat{k}\delta\hat{t}<1 \\ \mathscr{Q}_{2_{\hat{k}+1}}^i+(\frac{\partial x_{\hat{k}+2}}{\partial x_{\hat{k}+1}})^T\widehat{W}_{\hat{k}+1}^T\phi(\Gamma,t_f,x_{\hat{k}+1}^i,r_{\hat{k}+1}) & \text{if } 1\le\hat{k}\delta\hat{t}\le 2 \end{cases} \tag{35}$$

Once all the costates are calculated, one can use them forward in time for online control without re-training.

The training process discussed in this section is summarized in Algorithm 1.

Algorithm 1 : *Main Solution- Finding Costates*

step 1: Set $\hat{k}=N'$. Initialize the neural network weights, $\widehat{W}_{N'-1}^0$. Also select a small positive number γ as a convergence tolerance. Select η random training samples for $x\in\Omega_x$, and η random training samples for $r\in\Omega_x$. Also, select η random switching times and final time $\{t_1\le t_2\le\cdots\le t_p\le t_f\}\in\Omega_t$ where p is the number of switchings.

step 2: Set $\hat{k}=\hat{k}-1$ and start the outer loop.

step 2-1: If $\hat{k}\neq N'-1$, set $\widehat{W}_{\hat{k}}^0=\widehat{W}_{\hat{k}+1}$. Set $i=0$ and repeat the following inner loop:

step 2-1-1: Select η random training samples for states, reference signal, switching times, and the final time with the conditions explained in step 1. Substitute all the training samples in $\phi(.,.,.,.)$ and find $\lambda_{\hat{k}+1}^i(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}})$. With $\lambda_{\hat{k}+1}^i(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}})$ find $u_{\hat{k}}^i$ and control the states along it to find $x_{\hat{k}+1}^i$. Also, find $r_{\hat{k}+1}$.

step 2-1-2: Find $\widehat{W}_{\hat{k}}^{i+1}$ from (33) or (35) using least squares on the entire set of training samples.

step 2-1-3: If $\|\widehat{W}_{\hat{k}}^{i+1}-\widehat{W}_{\hat{k}}^i\|\le\gamma$, go to step 2-2. Otherwise, set $i=i+1$ and go back to step 2-1-1.

step 2-2: If $\hat{k}=1$, stop the training. Otherwise, set $\widehat{W}_{\hat{k}}=\widehat{W}_{\hat{k}}^{i+1}$, and go to step 2.

Remark 3: The convergence of an inner loop using SNAC in systems with non-switching and control affine dynamics was studied in Theorem 1 of [24] for optimal tracking in fixed final time. Also, such convergence for switched systems with fixed mode sequence and fixed final time was studied in Theorem 1 of [16].

Remark 4: Once the training is concluded, one needs to find the optimal switching times and the optimal final time from the costates for a selected initial condition $x_0\in\Omega$. The following methods are suggested to find the optimal switching

times from the optimal costates.

- **Method 1:** 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 again when the initial condition is changed. However, as the order of the system increases, this method becomes very complicated. Therefore, this method is only suitable for systems with low order dynamics.
- **Method 2:** propagating the states along all possible switching times and final times to find the optimal cost-to-go for all possible switching/final times. Once done, choose the switching/final times, which lead to the minimum cost-to-go. This method is very straightforward, and it is similar to forward dynamic programming. However, performing such calculations might be time-consuming as the order of the system increases. For such cases, one suggestion is starting the simulations with larger steps for switching times/ final time and then narrow down your search to smaller regions with more promising results.

The convergence of the iterative solution illustrated by equations (32) and (34) is studied in Theorem 1.

Theorem 1: Considering the iterative solution illustrated by (32) and (34), there exists a control penalizing matrix, i.e., R^*, such that for any control penalizing matrix R that $\|R\| \geq \|R^*\|$, the iterations shown in (32) and (34) converge.

Proof: The proof is inspired by [16], [24]. The proof first shows that the iterations in (32) and (34) form monotonically decreasing sequences. Afterward, the established monotonicity along the boundedness of the iterations will be used to prove the convergence to an unknown limit function. At last, the uniqueness of the solutions to the Bellman equation of optimality is used to prove the convergence to the optimal control solutions. Considering the time step $N'-1 \rightarrow N'$ and (32), one can form $\lambda^{i+1}_{N'} - \lambda^{i}_{N'}$ as

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

In (36), $\lambda^i = \lambda^i_{N'}(\Gamma, t_f, x^i_{N'}, r_{N'})$, $g(x_{N'-1}) = g_{v_{N'-1}}(x_{N'-1})$, and $R = R_{v_{N'-1}}$. Considering $\varepsilon^{i+1} = \lambda^{i+1} - \lambda^i$ and $\varepsilon^i = \lambda^i - \lambda^{i-1}$ in (36), one has

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

Taking norm of (37) leads

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

In (38), one notes that $\|\alpha\| = \rho_1^2 \|S\|\|R^{-1}\|$ where $\|g(x)\| \leq \rho_1, \forall x \in \Omega$. Therefore, $\|\alpha\| < 1$ can be achieved by the correct choice of R which leads to a lower bounded monotonically decreasing sequence of $\|\varepsilon^i\|$ as

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

In (39), as $i \rightarrow \infty$, $\|\varepsilon^i\| \rightarrow 0$, which indicates $\lambda^{i+1} \rightarrow \lambda^i$.

Considering $\hat{k} < N'$, on has

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

Letting $\lambda^i_{\hat{k}+1}(\Gamma, t_f, x^i_{\hat{k}+1}, r_{\hat{k}+1}) \equiv \lambda^i_{\hat{k}+1}$, and dropping the sub-index $v_{\hat{k}}$ in showing Q, R, $f(.)$, and $g(.)$ leads

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

In (41),

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

where $x^i_{\hat{k}+1} = f(x_{\hat{k}}) + g(x_{\hat{k}})(-R)^{-1} g^T(x_{\hat{k}})\lambda^i_{\hat{k}+1}$. From (41), one has

$$\begin{aligned}\lambda^{i+1}_{\hat{k}+1} &= Q(x^i_{\hat{k}+1} - r_{\hat{k}+1}) \\ &\quad + \big(\nabla f(x^i_{\hat{k}+1}) + \nabla\big(g(x^i_{\hat{k}+1})u^i_{\hat{k}+1}\big)\big)^T \lambda_{\hat{k}+2}(x^i_{\hat{k}+2}) \\ &= Q(x^i_{\hat{k}+1} - r_{\hat{k}+1}) + \nabla^T f(x^i_{\hat{k}+1})\lambda_{\hat{k}+2}(x^i_{\hat{k}+2}) \\ &\quad + {u^i_{\hat{k}+1}}^T \nabla^T g(x^i_{\hat{k}+1})\lambda_{\hat{k}+2}(x^i_{\hat{k}+2})\end{aligned} \tag{42}$$

Considering (42), substituting for ${u^i_{\hat{k}+1}}^T = \lambda^T_{\hat{k}+2}(x^i_{\hat{k}+2})g(x^i_{\hat{k}+1})(-R^{-1})$ one has $\Upsilon^i_{\hat{k}+1} = \lambda^T_{\hat{k}+2}(x^i_{\hat{k}+2})g(x^i_{\hat{k}+1})(-R^{-1})\nabla^T g(x^i_{\hat{k}+1})$. To further simplify the notations, consider $\Lambda^i_{1_{\hat{k}+1}}$ as $\mathrm{diag}([{\lambda^i_{\hat{k}+2}}^T(x^i_{\hat{k}+2})g(x^i_{\hat{k}+1})])$ where $\mathrm{diag}([x])$ is a diagonal matrix with $[x]$ on the main diagonal and $[\mathbf{0}] \in \mathbb{R}^{1\times m}$, a matrix of all elements zero, elsewhere. Also, ${\lambda^i_{\hat{k}+2}}^T g^i_{\hat{k}+1} = {\lambda^i_{\hat{k}+2}}^T(x^i_{\hat{k}+2})g(x^i_{\hat{k}+1})$. Similarly, consider $\Lambda^i_{2_{\hat{k}+1}} = \mathrm{diag}([(-R)^{-1}])$ with $(-R)^{-1}$ on the main diagonal and $[\mathbf{0}] \in \mathbb{R}^{m\times m}$ elsewhere. At last, consider $\Lambda^i_{3_{\hat{k}+1}} = \nabla^T g(x^i_{\hat{k}+1})\lambda_{\hat{k}+2}(x^i_{\hat{k}+2})$. Therefore, one can rewrite (42) as

$$\begin{aligned}\lambda^{i+1}_{\hat{k}+1} =& Q(x^i_{\hat{k}+1} - r_{\hat{k}+1}) + \nabla^T f(x^i_{\hat{k}+1})\lambda_{\hat{k}+2}(x^i_{\hat{k}+2}) \\ &+ \Lambda^i_{1_{\hat{k}+1}}\Lambda^i_{2_{\hat{k}+1}}\Lambda^i_{3_{\hat{k}+1}}\end{aligned} \tag{43}$$

It is straightforward to derive $\lambda^i_{\hat{k}+1}$ using (43). Hence, one has

$$\begin{aligned}\lambda^{i+1}_{\hat{k}+1} - \lambda^i_{\hat{k}+1} &= Q(x^i_{\hat{k}+1} - x^{i-1}_{\hat{k}+1}) \\ &\quad + \big(\nabla^T f(x^i_{\hat{k}+1})\lambda_{\hat{k}+2}(x^i_{\hat{k}+2}) - \nabla^T f(x^{i-1}_{\hat{k}+1})\lambda_{\hat{k}+2}(x^{i-1}_{\hat{k}+2})\big) \\ &\quad + \big(\Lambda^i_{1_{\hat{k}+1}}\Lambda^i_{2_{\hat{k}+1}}\Lambda^i_{3_{\hat{k}+1}} - \Lambda^{i-1}_{1_{\hat{k}+1}}\Lambda^{i-1}_{2_{\hat{k}+1}}\Lambda^{i-1}_{3_{\hat{k}+1}}\big)\end{aligned} \tag{44}$$

In (44), $\Lambda^i_{2_{\hat{k}+1}} = \Lambda^{i-1}_{2_{\hat{k}+1}} \equiv \Lambda_{2_{\hat{k}+1}}$ since it is not dependent to the iterations. To continue the proof, consider the following algebraic equation

$$\begin{aligned}&\Lambda^i_{1_{\hat{k}+1}}\Lambda_{2_{\hat{k}+1}}\Lambda^i_{3_{\hat{k}+1}} - \Lambda^{i-1}_{1_{\hat{k}+1}}\Lambda_{2_{\hat{k}+1}}\Lambda^{i-1}_{3_{\hat{k}+1}} = \\ &(\Lambda^i_{1_{\hat{k}+1}} - \Lambda^{i-1}_{1_{\hat{k}+1}})\Lambda_{2_{\hat{k}+1}}\Lambda^i_{3_{\hat{k}+1}} + \Lambda^{i-1}_{1_{\hat{k}+1}}\Lambda_{2_{\hat{k}+1}}(\Lambda^i_{3_{\hat{k}+1}} - \Lambda^{i-1}_{3_{\hat{k}+1}})\end{aligned} \tag{45}$$

Using (45) in (44), one has

$$\begin{aligned}\lambda^{i+1}_{\hat{k}+1} - \lambda^i_{\hat{k}+1} &= Q(x^i_{\hat{k}+1} - x^{i-1}_{\hat{k}+1}) \\ &+ \big(\nabla^T f(x^i_{\hat{k}+1})\lambda_{\hat{k}+2}(x^i_{\hat{k}+2}) - \nabla^T f(x^{i-1}_{\hat{k}+1})\lambda_{\hat{k}+2}(x^{i-1}_{\hat{k}+2})\big) \\ &+ (\Lambda^i_{1_{\hat{k}+1}} - \Lambda^{i-1}_{1_{\hat{k}+1}})\Lambda_{2_{\hat{k}+1}}\Lambda^i_{3_{\hat{k}+1}} \\ &+ \Lambda^{i-1}_{1_{\hat{k}+1}}\Lambda_{2_{\hat{k}+1}}(\Lambda^i_{3_{\hat{k}+1}} - \Lambda^{i-1}_{3_{\hat{k}+1}})\end{aligned} \tag{46}$$

Similar to (38), one is interested to find a monotonic behavior

in the iterations. By taking the norm of (46), through some algebraic manipulations one has

$$\begin{aligned}&\|\lambda^{i+1}_{\hat{k}+1}-\lambda^{i}_{\hat{k}+1}\|\leq\|Q\|\|(x^{i}_{\hat{k}+1}-x^{i-1}_{\hat{k}+1})\|\\&+\|\nabla^T f(x^{i}_{\hat{k}+1})\lambda_{\hat{k}+2}(x^{i}_{\hat{k}+2})-\nabla^T f(x^{i-1}_{\hat{k}+1})\lambda_{\hat{k}+2}(x^{i-1}_{\hat{k}+2})\|\\&+\|(\Lambda^{i}_{1_{\hat{k}+1}}-\Lambda^{i-1}_{1_{\hat{k}+1}})\|\|\Lambda_{2_{\hat{k}+1}}\|\|\Lambda^{i}_{3_{\hat{k}+1}}\|\\&+\|\Lambda^{i-1}_{1_{\hat{k}+1}}\|\|\Lambda_{2_{\hat{k}+1}}\|\|(\Lambda^{i}_{3_{\hat{k}+1}}-\Lambda^{i-1}_{3_{\hat{k}+1}})\|\end{aligned}\tag{47}$$

The smoothness assumption of the value functions, $f(.)$, and $g(.)$ leads to smoothness of $\lambda_{\hat{k}+2}(.)$, $\nabla f(.)$, and $\nabla g(.)$, respectively. Also, one can deduct Lipschitz continuity of $\nabla f(.)\lambda_{\hat{k}+2}(.)$, $\Lambda^{i}_{1_{\hat{k}+1}}$, and $\Lambda^{i}_{3_{\hat{k}+1}}$ with Lipschitz constants of β_1, β_2, and β_3, respectively. In addition, one notes that the smoothness of $g(.)$, $\Lambda^{i}_{1_{\hat{k}+1}}$ and $\Lambda^{i}_{3_{\hat{k}+1}}$ leads to boundedness in the compact region Ω. Therefore, one has $\|g(.)\|\leq\rho_1$, $\|\Lambda^{i}_{1_{\hat{k}+1}}\|\leq\rho_2$, and $\|\Lambda^{i}_{3_{\hat{k}+1}}\|\leq\rho_3$. Also, it is straightforward to see that $\|\Lambda^{i}_{2_{\hat{k}+1}}\|\leq n\|R^{-1}\|$. Therefore, one has

$$\begin{aligned}\|\lambda^{i+1}_{\hat{k}+1}-\lambda^{i}_{\hat{k}+1}\|&\leq\|Q\|\|(x^{i}_{\hat{k}+1}-x^{i-1}_{\hat{k}+1})\|\\&+\beta_1\|(x^{i}_{\hat{k}+1}-x^{i-1}_{\hat{k}+1})\|\\&+n\rho_3\beta_2\|R^{-1}\|\|(x^{i}_{\hat{k}+1}-x^{i-1}_{\hat{k}+1})\|\\&+n\rho_2\beta_3\|R^{-1}\|\|(x^{i}_{\hat{k}+1}-x^{i-1}_{\hat{k}+1})\|\end{aligned}\tag{48}$$

Considering (48), one notes that

$$\begin{aligned}\|(x^{i}_{\hat{k}+1}-x^{i-1}_{\hat{k}+1})\|&\leq\|g(x_{\hat{k}})\|^2\|R^{-1}\|\|\lambda^{i}_{\hat{k}+1}-\lambda^{i-1}_{\hat{k}+1}\|\\&\leq\rho_1^2\|R^{-1}\|\|\lambda^{i}_{\hat{k}+1}-\lambda^{i-1}_{\hat{k}+1}\|\end{aligned}\tag{49}$$

Using (49) in (48), one has

$$\begin{aligned}&\|\lambda^{i+1}_{\hat{k}+1}-\lambda^{i}_{\hat{k}+1}\|\leq\big(\|Q\|+\beta_1\\&+(n\rho_3\beta_2+n\rho_2\beta_3)\|R^{-1}\|\big)\rho_1^2\|R^{-1}\|\|\lambda^{i}_{\hat{k}+1}-\lambda^{i-1}_{\hat{k}+1}\|\\&\leq\alpha_1\|\lambda^{i}_{\hat{k}+1}-\lambda^{i-1}_{\hat{k}+1}\|\end{aligned}\tag{50}$$

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

$$\|\varepsilon^{i+1}\|\leq\alpha_1\|\varepsilon^{i}\|\tag{51}$$

Considering the boundedness of α_1, it is easy to see that $\|\varepsilon^{i}\|$ in (51) 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.

In [16], it was shown for the case of fixed final time, the costates result in optimal policy which solve the Bellman equation of optimality. Since, the solutions to the Bellman equation of optimality are unique, this results in the optimality of the costates. This result is true for the free final time problem as well. □

IV. Single Loop Training Algorithm

In this section, the effect of eliminating the inner loop in Algorithm 1 is investigated. Eliminating the inner loop can reduce the required time for training. However, it raises serious concerns about the performance of the optimal controller. Note that the purpose of the inner loop is finding the optimal costates, which will be used to find the optimal policies at each time step. Hence, by eliminating the inner loop, one uses the policies that are not optimal, and they need further iterations to converge. In other words, eliminating the inner loop leads to using evolving suboptimal policies to find the optimal control solution. The training process with the single loop solution is detailed in Algorithm 2.

Algorithm 2 : *Single Loop Solution*

step 1: Set $\hat{k}=N'$. Initialize the neural network weights, $\widehat{W}^0_{N'-1}$. Select η random training samples for x and r in Ω_x. Also, select η random switching times and final time $\{t_1\leq t_2\leq\cdots\leq t_p\leq t_f\}\in\Omega_t$ where p is the number of switching.

step 2: Set $\hat{k}=\hat{k}-1$. If $\hat{k}\neq N'-1$, set $\widehat{W}_{\hat{k}}=\widehat{W}_{\hat{k}+1}$. Repeat the following loop.

- **step 2-1:** Select η random training samples for states, switching times, and the final time with the conditions explained in step 1. Substitute all the training samples in $\phi(.,.,.,.)$ and find a $\lambda_{\hat{k}+1}(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}})$. With $\lambda_{\hat{k}+1}(\Gamma,t_f,x_{\hat{k}},r_{\hat{k}})$ find $u_{\hat{k}}$ and control the states with it to find $x_{\hat{k}+1}$. Also, find $r_{\hat{k}+1}$.
- **step 2-2:** Find $\widehat{W}_{\hat{k}}$ through (20) or (28) using least squares on the entire set of training samples.
- **step 2-3:** If $\hat{k}=1$, stop the training.

The effect of evolving suboptimal policies in Algorithm 2 is studied in Theorem 2.

Theorem 2: The error between the states controlled with policies generated by Algorithms 1 and 2 is bounded and can be adjusted by choice of R.

Proof: For proving Theorem 2, we first show that the error between the states controlled from the same initial conditions along algorithms 1 and 2 is bounded for each time. Also, we show that this bound can become arbitrary small by the correct choice of control penalizing matrix R. Using this result, we then establish the boundedness of the errors for the case that the initial conditions for the states are not the same. Let the costates generated by Algorithm 1 be denoted by $\lambda^{\infty}_{\hat{k}+1}(.)$ and the costates generated by Algorithm 2 be represented as $\lambda^{1}_{\hat{k}+1}$. Considering $x_{\hat{k}}$, one can denote the states controlled with $\lambda^{\infty}_{\hat{k}+1}(.)$ and $\lambda^{1}_{\hat{k}+1}$ as $x^{\infty}_{\hat{k}+1}$ and $x^{1}_{\hat{k}+1}$, respectively. Therefore, one has

$$x^{\infty}_{\hat{k}+1}-x^{1}_{\hat{k}+1}=g(x_{\hat{k}})(-R)^{-1}g^T(x_{\hat{k}})(\lambda^{\infty}_{\hat{k}+1}-\lambda^{1}_{\hat{k}+1})\tag{52}$$

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

$$x^{\infty}_{N'}-x^{1}_{N'}=g(x_{N'-1})(-R)^{-1}g^T(x_{N'-1})(\lambda^{\infty}_{N'}-\lambda^{1}_{N'})\tag{53}$$

Substituting for $\lambda^{\infty}_{N'}$ and $\lambda^{1}_{N'}$, after some algebraic manipulations, one has

$$x^{\infty}_{N'}-x^{1}_{N'}=\Psi S\Psi\big(\lambda^{\infty}_{N'}-\lambda^{0}_{N'}\big)\tag{54}$$

where $\Psi=g(x_{N'-1})(-R)^{-1}g^T(x_{N'-1})$. By adding and subtracting $\lambda^{i}_{N'}$ to the right-hand side of (54), one can expand $\lambda^{\infty}_{N'}-\lambda^{0}_{N'}$ as $\lambda^{\infty}_{N'}-\lambda^{\infty-1}_{N'}+\lambda^{\infty-1}_{N'}-\cdots-\lambda^{1}_{N'}+\lambda^{1}_{N'}-\lambda^{0}_{N'}$. Letting $\varepsilon^{i+1}_{N'}=\lambda^{i+1}_{N'}-\lambda^{i}_{N'}$, one can rewrite (54) as

$$x^{\infty}_{N'}-x^{1}_{N'}=\Psi S\Psi\sum_{i=1}^{\infty}\varepsilon^{i}_{N'}\tag{55}$$

Applying norms on (55), one has

$$\|x^{\infty}_{N'} - x^{1}_{N'}\| \leq \|\Psi\|^2 \|S\| \sum_{i=1}^{\infty} \|\varepsilon^{i}_{N'}\| \tag{56}$$

As shown in the proof of Theorem 1, the sequence of $\|\varepsilon^{i}_{N'}\|$ is monotonically decreasing with $\|\varepsilon^{i+1}_{N'}\| \leq \|\alpha\| \|\varepsilon^{i}_{N'}\|$ where $\|\alpha\| < 1$. Therefore, (56) leads

$$\begin{aligned} \|x^{\infty}_{N'} - x^{1}_{N'}\| &\leq \|\Psi\|^2 \|S\| \sum_{i=0}^{\infty} \|\alpha\|^i \|\varepsilon^{1}_{N'}\| \\ &\leq \|\Psi\|^2 \|S\| \|\varepsilon^{1}_{N'}\| \sum_{i=0}^{\infty} \|\alpha\|^i \end{aligned} \tag{57}$$

In (57), since $\|\alpha\| < 1$, the series form a geometric series converging to $\frac{1}{1-\|\alpha\|}$. Also, since $\|\Psi\| \leq \rho_1^2 \|R^{-1}\|$, one has

$$\|x^{\infty}_{N'} - x^{1}_{N'}\| \leq \rho_1^4 \|R^{-1}\|^2 \|S\| \|\varepsilon^{1}_{N'}\| \frac{1}{1-\|\alpha\|} \tag{58}$$

In (58), $\|g(x)\| \leq \rho_1$, $\forall x \in \Omega$. Due to boundedness of $\|\varepsilon^{1}_{N'}\|$, the magnitude of the error can become small as $\|R\|$ increases.

For $\hat{k} < N' - 1$, one can find $x^{\infty}_{\hat{k}+1} - x^{1}_{\hat{k}+1}$ by using the respective controls as

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

Letting $\Psi = g(x_{\hat{k}})(-R)^{-1} g^T(x_{\hat{k}})$, substituting for $\lambda^{\infty}_{\hat{k}+1}$ and $\lambda^{1}_{\hat{k}+1}$ leads

$$\begin{aligned} x^{\infty}_{\hat{k}+1} - x^{1}_{\hat{k}+1} &= \Psi Q \Psi (\lambda^{\infty}_{\hat{k}+1} - \lambda^{0}_{\hat{k}+1}) \\ &+ \Psi\big(\nabla f(x^{\infty}_{\hat{k}+1}) \lambda_{\hat{k}+2}(x^{\infty}_{\hat{k}+2}) - \nabla f(x^{0}_{\hat{k}+1}) \lambda_{\hat{k}+2}(x^{0}_{\hat{k}+2})\big) \\ &+ \Psi\big(\Lambda^{\infty}_{1_{\hat{k}+1}} \Lambda^{\infty}_{2_{\hat{k}+1}} \Lambda^{\infty}_{3_{\hat{k}+1}} - \Lambda^{0}_{1_{\hat{k}+1}} \Lambda^{0}_{2_{\hat{k}+1}} \Lambda^{0}_{3_{\hat{k}+1}}\big) \end{aligned} \tag{60}$$

In (60), $\Lambda^{*}_{1_{\hat{k}+1}}$, $\Lambda^{*}_{2_{\hat{k}+1}}$ and $\Lambda^{*}_{3_{\hat{k}+1}}$ were introduced in the proof of Theorem 1. Applying norms on (60) leads

$$\begin{aligned} \|x^{\infty}_{\hat{k}+1} - x^{1}_{\hat{k}+1}\| &\leq \|\Psi Q \Psi\| \|\lambda^{\infty}_{\hat{k}+1} - \lambda^{0}_{\hat{k}+1}\| \\ &+ \|\Psi\| \|\nabla f(x^{\infty}_{\hat{k}+1}) \lambda_{\hat{k}+2}(x^{\infty}_{\hat{k}+1}) - \nabla f(x^{0}_{\hat{k}+1}) \lambda_{\hat{k}+2}(x^{0}_{\hat{k}+1})\| \\ &+ \|\Psi\| \|\Lambda^{\infty}_{1_{\hat{k}+1}} \Lambda^{\infty}_{2_{\hat{k}+1}} \Lambda^{\infty}_{3_{\hat{k}+1}} - \Lambda^{0}_{1_{\hat{k}+1}} \Lambda^{0}_{2_{\hat{k}+1}} \Lambda^{0}_{3_{\hat{k}+1}}\| \end{aligned} \tag{61}$$

Following the same procedure as explained in equations (45) to (51), through some algebraic manipulations one has

$$\|x^{\infty}_{\hat{k}+1} - x^{1}_{\hat{k}+1}\| \leq \alpha_2 \|\lambda^{\infty}_{\hat{k}+1} - \lambda^{0}_{\hat{k}+1}\| \tag{62}$$

where $\alpha_2 = \big(\rho_1^4 \|Q\| \|R^{-1}\| + \beta_1 \rho_1^2 + \rho_1^4 n \|R^{-1}\|^2 (\rho_3 \beta_2 + \rho_2 \beta_3)\big) \|R^{-1}\|$. By adding and subtracting $\lambda^{i}_{\hat{k}+1}$ between $\lambda^{\infty}_{\hat{k}+1} - \lambda^{0}_{\hat{k}+1}$ and letting $\varepsilon^{i+1}_{\hat{k}+1} = \lambda^{i+1}_{\hat{k}+1} - \lambda^{i}_{\hat{k}+1}$, one has

$$\begin{aligned} \|x^{\infty}_{\hat{k}+1} - x^{1}_{\hat{k}+1}\| &\leq \alpha_2 \sum_{i=0}^{\infty} \|\alpha_1\|^i \|\varepsilon^{1}_{\hat{k}+1}\| \\ &\leq \alpha_2 \|\varepsilon^{1}_{\hat{k}+1}\| \frac{1}{1-\|\alpha_1\|} \end{aligned} \tag{63}$$

In (63), α_2 becomes arbitrary small by a correct choice of R.

So far, it was shown that the error between states controlled from the same initial conditions using the controllers trained by Algorithms 1 and 2 is bounded for each time. The next step in the proof is using this result to establish the boundedness of the errors for the case that the initial conditions for the states are not the same. Therefore, consider time step $\hat{k} = 0$ to $\hat{k} = 1$. Considering (63), one can deduct the boundedness of $\delta = x^{\infty}_{1} - x^{1}_{1}$ which can become arbitrary small by the choice of R. Considering the time step $\hat{k} = 1$ to $\hat{k} = 2$, using the initial conditions as $x^{\infty}_{1} = x^{1}_{1} + \delta$ and x^{1}_{1}, one has

$$\begin{aligned} x^{\infty}_{2} &= f(x^{\infty}_{1}) + g(x^{\infty}_{1})\big(-R^{-1} g^T(x^{\infty}_{1}) \lambda^{\infty}_{2}(x^{\infty}_{2})\big) \\ &= f(x^{1}_{1} + \delta) + g(x^{1}_{1} + \delta)\big(-R^{-1} g^T(x^{1}_{1} + \delta) \lambda^{\infty}_{2}(x^{\infty}_{2})\big) \end{aligned} \tag{64}$$

$$x^{1}_{2} = f(x^{1}_{1}) + g(x^{1}_{1})\big(-R^{-1} g^T(x^{1}_{1}) \lambda^{1}_{2}(x^{1}_{2})\big) \tag{65}$$

Subtracting (65) from (64) and applying norms lead

$$\begin{aligned} \|x^{\infty}_{2} - x^{1}_{2}\| &\leq \|f(x^{1}_{1} + \delta) - f(x^{1}_{1})\| \\ &+ \|g(x^{1}_{1} + \delta) R^{-1} g^T(x^{1}_{1} + \delta) \lambda^{\infty}_{2}(x^{\infty}_{2}) \\ &- g(x^{1}_{1}) R^{-1} g^T(x^{1}_{1}) \lambda^{1}_{2}(x^{1}_{2})\| \end{aligned} \tag{66}$$

In what follows, we are going to find an upper bound for the right-hand side of (66) that can be adjusted by choice of control penalizing matrix, i.e., R. 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

$$\begin{aligned} &\psi^{\infty} \lambda^{\infty}_{2}(x^{\infty}_{2}) - \psi^{1} \lambda^{1}_{2}(x^{1}_{2}) = \\ &(\psi^{\infty} - \psi^{1}) \lambda^{\infty}_{2}(x^{\infty}_{2}) + \psi^{1}\big(\lambda^{\infty}_{2}(x^{\infty}_{2}) - \lambda^{1}_{2}(x^{1}_{2})\big) \end{aligned} \tag{67}$$

Applying norms on (67), one has

$$\begin{aligned} &\|\psi^{\infty} \lambda^{\infty}_{2}(x^{\infty}_{2}) - \psi^{1} \lambda^{1}_{2}(x^{1}_{2})\| \leq \\ &\|\psi^{\infty} - \psi^{1}\| \|\lambda^{\infty}_{2}(x^{\infty}_{2})\| + \|\psi^{1}\| \|\lambda^{\infty}_{2}(x^{\infty}_{2}) - \lambda^{1}_{2}(x^{1}_{2})\| \end{aligned} \tag{68}$$

Through some algebraic manipulations, one has

$$\begin{aligned} \psi^{\infty} - \psi^{1} =& \big(g(x^{1}_{1} + \delta) - g(x^{1}_{1})\big) R^{-1} g^T(x^{1}_{1} + \delta) \\ &+ g(x^{1}_{1}) R^{-1} \big(g(x^{1}_{1} + \delta) - g(x^{1}_{1})\big)^T \end{aligned} \tag{69}$$

Applying norms on (69), through further algebraic manipulations one has

$$\|\psi^{\infty} - \psi^{1}\| = 2\beta_4 \rho_1^2 \|R^{-1}\| \|\delta\| \tag{70}$$

where β_4 is selected as Lipschitz constant for $g(.)$ in Ω. Considering $\|\lambda^{\infty}_{2}(x^{\infty}_{2})\| \leq \rho_4$, it is easy to see that $\|\psi^{\infty} - \psi^{1}\| \|\lambda^{\infty}_{2}(x^{\infty}_{2})\| \leq 2\beta_4 \rho_1^2 \rho_4 \|R^{-1}\| \|\delta\|$. Also, considering the second term on the right-hand side of (68) with a similar procedure used to derive (63) one has

$$\begin{aligned} &\|\psi^{1}\| \|\lambda^{\infty}_{2}(x^{\infty}_{2}) - \lambda^{1}_{2}(x^{1}_{2})\| \leq \\ &\rho_1^2 \|R^{-1}\| \|\lambda^{\infty}_{2}(x^{\infty}_{2}) - \lambda^{1}_{2}(x^{1}_{2})\| \leq \\ &\leq \rho_1^2 \|R^{-1}\| \|\lambda^{2}_{2}(x^{2}_{2}) - \lambda^{1}_{2}(x^{1}_{2})\| \frac{1}{1-\|\alpha_1\|} \end{aligned} \tag{71}$$

Using (70) and (71) in (66) along Lipschitz assumption of $f(.)$ leads

$$\begin{aligned} \|x^{\infty}_{2} - x^{1}_{2}\| &\leq \beta_1 \|\delta\| + 2\beta_4 \rho_1^2 \rho_4 \|R^{-1}\| \|\delta\| \\ &+ \rho_1^2 \|R^{-1}\| \|\lambda^{2}_{2}(x^{2}_{2}) - \lambda^{1}_{2}(x^{1}_{2})\| \frac{1}{1-\|\alpha_1\|} \end{aligned} \tag{72}$$

It can be seen that all the terms on the right hand side of (72) can become arbitrary small by the choice of R. Also, $\|\delta\|$ in the first term on the right-hand side of (72) can be calculated from (63) 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 completes the proof. □

Remark 5: Based on the discussions provided in the proof of Theorems 1 and 2, it can be seen that the magnitude of the error signal can be regulated by the choice of the control penalizing matrix, i.e., R. It is straightforward to see the upper bound for the magnitude of the error signal in Theorem 1.

However, in Theorem 2, the error can be accumulated as the states are propagated. Yet, since the problem is a discrete time problem and the final time is finite, the summation of all errors is bounded, and hence, can become arbitrarily small by the choice of the control penalizing matrix R.

Remark 6: Algorithms 1 and 2 can slightly be modified to include simpler cases when the dynamics of the reference signal is a function of time as

$$\dot{r}(t) = f_{r_v}(t) \tag{73}$$

In such cases, the reference signal is not required to be considered in the structure of the costates [24], i.e., $\lambda_{\hat{k}} = \lambda_{\hat{k}}(\Gamma, t_f, x_{\hat{k}})$ and not $\lambda_{\hat{k}}(\Gamma, t_f, x_{\hat{k}}, r_{\hat{k}})$. Hence, the basis functions for approximating the costates do not include the reference signal, i.e., $\phi = \phi(\Gamma, t_f, x_{\hat{k}})$. Lastly, in performing Algorithms 1 and 2, one does not need to generate random training samples for the reference signal in steps 1 & 2-1-1 of Algorithm 1, and steps 1 & 2-1 of Algorithm 2.

V. Numerical Simulations

In all simulations, the following second order modes are used to provide the mode sequences [24]. The first mode is selected as the Van der Pol oscillator as

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= \big(1 - x_1^2(t)\big)x_2(t) - x_1(t) + u(t) \end{aligned} \tag{74}$$

For the second mode, a linear subsystem is selected as

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= 2x_1 - x_2 + u(t) \end{aligned} \tag{75}$$

Lastly, the third mode is selected as the following nonlinear second order system.

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= x_1^2 - x_2^2 + u(t) \end{aligned} \tag{76}$$

The mode sequence Γ can be selected from the modes introduced in (74)-(76). In this paper, the simulation results are conducted for both one and two switchings. In all simulations related to one switching, the mode sequence was selected as $\Gamma = \{mode\,1, mode\,2\}$. Also, in all simulations with two switchings, the mode sequence was selected as $\Gamma = \{mode\,1, mode\,2, mode\,3\}$. Following the same procedure as used for two switchings, the solutions can be easily extended to include more switchings.

In the simulations, it is desired to find the optimal switching time(s), optimal final time, and the optimal control such that the cost function given in (2) is minimized and the system tracks a desired reference signal. Throughout the simulation results, the parameters of the cost function in (2) were selected as[1] $S = \text{diag}(10^5, 10^5)$, $\bar{Q} = \text{diag}(10^5, 10^7)$, and $\bar{R} = 10^3$. Also, the discretization sample time for all simulations was selected as $\delta\hat{t} = 0.001$. In general, $\bar{R}$, S, and $\bar{Q}$ are design parameters. As stated in the proof of Theorem 1, the convergence of the inner loop is linked on the magnitude of the norm of R. Therefore, we selected $\bar{R}$ large enough that the inner loop converges and then selected the matrices S and $\bar{Q}$ to have better results.

[1] $\text{diag}(a,b)$ represents a 2×2 diagonal matrix with a and b on the main diagonal and zero elsewhere.

Based of the dynamics of the desired trajectory, the simulation results are divided into two groups. In the first group, the desired trajectory is a known function of time. For this group, the desired trajectory is selected as

$$\begin{aligned} \dot{r}_1(t) &= \sin(\pi t) \\ \dot{r}_2(t) &= \pi\cos(\pi t) \end{aligned} \tag{77}$$

The initial condition for the desired trajectory in (77) was selected as $r_0 = [1, -1]^T$ throughout the paper. For the second group of simulations, the desired trajectory is a function of the desired states as

$$\begin{aligned} \dot{r}_1(t) &= -r_1(t) \\ \dot{r}_2(t) &= -r_2(t) \end{aligned} \tag{78}$$

In the second group of simulations, one does not need to specify the initial condition for the desired trajectory as the controller would be trained for all initial conditions in a domain of training.

In all ADP solutions related to group 1 where the desired trajectory is a function of time, the domain of training included the switching time(s) (t_1 in one switching scenario and t_1, & t_2 in two switchings scenario), the final time t_f, and the states (x_1 and x_2). In the second group of simulations in which the desired trajectory was a function of the desired signal, the domain of training should additionally include the desired signals as r_1 and r_2. Therefore, in the most general scenario, the domain of training in all simulations was confined to $\Omega = \{(t_1, t_2, t_f, x_1, x_2, r_1, r_2) |\ t_0 = 0 < t_1 < t_2 < t_f < 5,\ |x_1| \leq 4,\ |x_2| \leq 4,\ |r_1| \leq 4,\ |r_2| \leq 4\}$.

At last, all the ADP solutions were coded in Matlab 2017a and performed on an office desktop computer with 16 GB of RAM and Intel(R) Core(TM) i7-3770 Central Processing Unit (CPU) @ 3.4 GHz.

A. Group 1: Time-Based Reference Signal

Two simulation examples are provided which include one switching and two switchings scenarios.

1) One Switching: Consider the mode sequence as $\Gamma = \{mode\,1, mode\,2\}$ with dynamics illustrated in (74) and (75). It is desired to find the control signals, the optimal switching time t_1, and the final time such that the cost function introduced in (2) is minimized.

In order to perform Algorithms 1 and 2, a linear-in-parameter neural network with basis functions comprised of polynomials with all possible combinations of t_1, t_f, x_1, and x_2 up to the power of 3 without repetition was selected. For training, 1000 random training patterns (t_1, t_f, x_1, x_2) were generated in the domain of training. The training for finding the optimal costates was concluded in 17.66 (sec) using Algorithm 1 and only 8.18 (sec) using Algorithm 2.

Once training concluded, the optimal costates were used to find the optimal switching time and the final time. Hence, the optimal costates at $\hat{t} = 0$ were integrated analytically to find the optimal value functions. Once done, an initial condition, i.e., $x_0 = [1, -0.5]^T$ was selected and the value functions were evaluated at the selected x_0. The results were the optimal value functions with only two variables as the switching time t_1 and the final time t_f. To find the optimal switching time t_1 and the

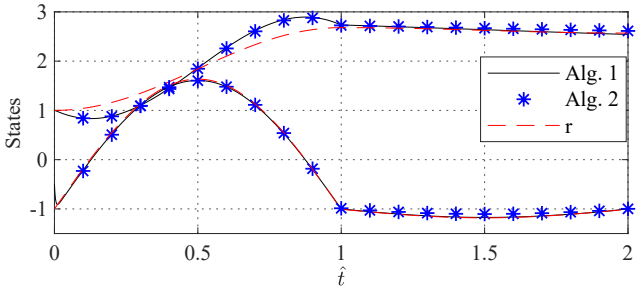

Fig. 1: Comparison among the state trajectories controlled with controls generated by the ADP methods discussed in this paper.

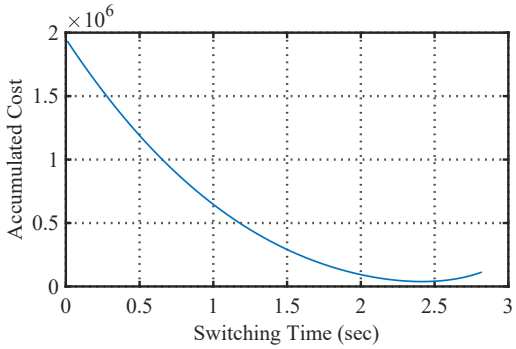

Fig. 2: Accumulated cost vs switching time with a fixed final time.

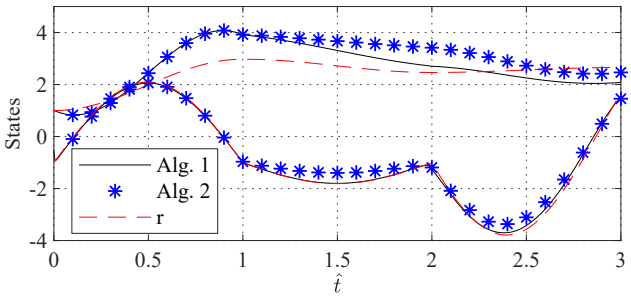

Fig. 3: Comparison among the state trajectories generated by the controllers trained with Algorithms 1 and 2.

final time t_f, the optimal value functions were evaluated at all $t_1, t_f \in (t_0, 5)$ and the values which led to the minimum of the optimal value functions were selected. This process took about 2.35 (sec). Using the optimal costates trained by Algorithm 1, the optimal switching time and the optimal final time were found at $t_1 = 2.639$ (sec) and $t_f = 2.812$ (sec), respectively. Using Algorithm 2, the optimal values for the switching time and the final time were found as $t_1 = 2.635$ (sec) and $t_f = 2.74$ (sec), respectively. The history of the states using controllers trained by Algorithms 1 and 2 are compared in Fig. 1. The results prove the effectiveness of the discussed algorithms in this paper.

In order to further investigate the performance of the controller trained by Algorithm 1, consider the same final time as the one assigned by the controller, i.e., $t_f = 2.812$. Also, imagine that we want to do a grid search for the switching time $t_1 \in [0, t_f]$, and control the states from the same initial condition as the one used in Fig. 1, along the policies dictated by the controller trained by Algorithm 1. The accumulated cost was calculated for different switching times and is depicted in Fig. 2. As one can see from Fig. 2, the minimum cost happens around $t_1 = 2.5$ seconds which is very close to the switching time assigned by the controller, i.e., $t_1 = 2.639$.

Compared to the base-line ADP solutions for non-switching dynamics [24], since the sequence of active modes is known, one can go backward in time from t_f to the unknown switching time t_1 using the second mode and then from t_1 to t_0 using the first mode. We notice that for performing such solution, for each switching time $t_1 \in \{t_0, \delta t, 2\delta t, \cdots, t_f\}$, one needs to do the training separately and then compare the costs to find the best switching time. On the other hand, by parametrization of the switching time, the training will be conducted *only* once and then we will find the optimal switching time when the training is concluded.

2) Two Switching: Let the system has two switchings at t_1, and t_2, and the mode sequence be $\Gamma = \{mode\ 1, mode\ 2, mode\ 3\}$. It is desired to find the optimal switching time instants, the final time, and the optimal policy such that the cost function represented in (2) is minimized and the system tracks the reference signal presented in (77). To start the solutions, select all the design parameters the same as the previous example.

For training with Algorithms 1 and 2, a linear in parameter neural network with basis functions comprised of polynomials with all possible combinations of t_1, t_2, t_f, x_1, and x_2 up to the power of 5 without repetition was selected. 1000 random training patterns were generated in the domain of training. The training process concluded in 158 (sec) using Algorithm 1 and only 32.28 (sec) using Algorithm 2.

Once the training concluded, the optimal costates were used to find the optimal switching times and the final time with a similar procedure that was used in the previous example. Using Algorithm 1, the optimal switching time instants and the final time were sought as $t_1 = 3.1$ (sec), $t_2 = 3.9$ (sec), and $t_f = 4.2$ (sec). Also, using Algorithm 2, these values were sought as $t_1 = 3.1$ (sec), $t_2 = 3.5$ (sec), and $t_f = 4$ (sec) which are very close to the results of training with Algorithm 1.

At last, the performance of the controllers trained by methods discussed in this paper are compared in Fig. 3. As one can see form Fig. 3, the controllers have a very similar performance which ensures the effectiveness of the solutions.

B. Group 2: State Based Reference Signal

Similar to the previous examples, the simulation results in this section also include one switching and two switchings scenarios. Since the controllers in this section are trained for a family of the desired trajectory, unlike the time-based desired trajectories, there is no need to re-train the controller for different initial conditions of the desired trajectory. In other words, the controller in this section can function effectively for all the initial conditions for the desired trajectory in the domain of training.

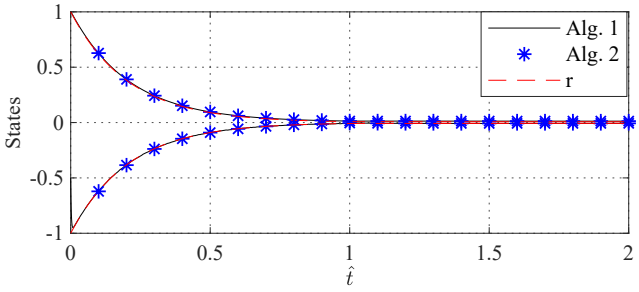

Fig. 4: Comparison of the performance of the controllers trained by Algorithms 1 and 2. As one can see, the controllers have a very similar performance.

1) One Switching: Let the mode sequence be $\Gamma = \{mode\,1, mode\,2\}$ with dynamics presented in (74) and (75). It is desired to find the optimal switching time t_1, the final time t_f, and the control signals such that the cost function in (2) is minimized where the dynamics of desired trajectories is given by (78). As mentioned before, the parameters of the cost function and the domain of training are kept the same throughout the numerical simulations.

For training with Algorithms 1 and 2, the basis functions of the neural network were selected as polynomials with all possible combination of t_1, t_f, x_1, x_2, r_1, and r_2 up to the power of 3 without repetition. Using Algorithm 1, the training process concluded in 35.23 (sec). Also, the training process concluded in 11.81 (sec) using Algorithm 2.

Once the training concluded, the optimal costates were used to find the optimal switching time t_1 and the final time t_f. Similar to the previous examples, the optimal costates were integrated analytically to find the optimal value functions at the initial states as $x_0 = [1, -0.5]^T$ and $r_0 = [1, -1]^T$. Then, the optimal value function was minimized with respect to the switching time and the final time. In both algorithms, the switching time and the final time were sought as $t_1 = 4.801$ (sec) and $t_f = 4.902$ (sec), respectively. The history of the states controlled with policies generated from controllers trained by Algorithms 1 and 2 is compared in Fig. 4. As one can see, the controllers have a very similar performance and the overall performance of the controllers in tracking the desired trajectory is good.

2) Two switchings: Let the mode sequence be $\Gamma = \{mode\,1, mode\,2, mode\,3\}$. It is desired to find the optimal switching times t_1 and t_2, the final time t_f, and the control signals such that the overall system tracks the reference trajectory depicted in (78) and the cost function in (2) is minimized. By choosing all the design parameters the same as the previous examples, one starts the solution.

For training by Algorithms 1 and 2, the basis functions of the neural network were selected as polynomials with all possible combination of t_1, t_2, t_f, x_1, x_2, r_1, and r_2 up to the power of 4 without repetition. For training, 1000 random training patterns were generated in the same domain of training used in the previous examples. Using Algorithm 1, the training concluded in 91 (sec). The training concluded in only 20.26 (sec) using Algorithm 2.

Once the training concluded, the optimal costates were used to find the optimal switching times and the final time. Again, the optimal costates at $\hat{k} = 0$ were integrated analytically to find the optimal value function. The initial values for the states and the reference signal were selected as $x_0 = [1, -0.5]$ and $r_0 = [1, -1]$, respectively. Using Algorithms 1 and 2, the optimal switching times and the final time were sought as $t_1 = 4.97$ (sec), $t_2 = 4.98$ (sec), and $t_f = 4.993$. The performance of these two controllers are compared in Fig. 5. As one can see, both controllers have a very good performance in tracking the reference signal and the performance of the controller trained by Algorithm 2 is very close to the one trained by Algorithm 1.

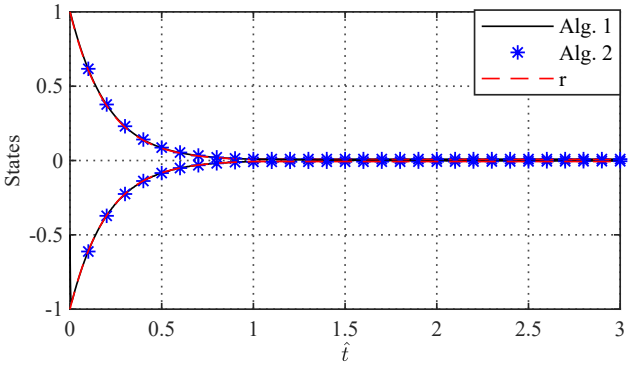

Fig. 5: Comparison of the performance of the controllers trained by Algorithms 1 and 2. As one can see, the controllers have a very similar performance.

VI. Conclusion

An approximate dynamic programming solution was introduced to solve the optimal tracking problem in switched systems with fixed mode sequences and free final time. The backbone of the solution was including the switching times and the final time as parameters in the optimal control problem formulations. A single network adaptive critic structure was used to approximate the optimal costates. Two algorithms were introduced to perform the proposed solution. In the convergence analysis of the first algorithm, the convergence of the training algorithm was linked to the magnitude of the control penalizing matrix, which is a design parameter. Meanwhile, a new solution was introduced which could be trained much faster than the first controller. Also, the performance of the new controller was analyzed and compared to that of the first controller. At last, the effectiveness of the proposed solutions was confirmed through numerical simulations.

Acknowledgment

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

References

[1] D. Kirk, *Optimal Control Theory: An Introduction*. Dover Publications, 2004.

[2] F. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *Circuits and Systems Magazine, IEEE*, vol. 9, no. 3, pp. 32–50, 2009.

[3] A. G. Khiabani, "Design and implementation of an optimal switching controller for uninterruptible power supply inverters using adaptive dynamic programming," *IET Power Electronics*, vol. 12, pp. 3068–3076, 2019.

[4] C. Zhang, M. Gan, and J. Zhao, "Data-driven optimal control of switched linear autonomous systems," *International Journal of Systems Science*, vol. 50, no. 6, pp. 1275–1289, 2019.

[5] M. Gan, C. Zhang, and J. Zhao, "Data-driven optimal switching of switched systems," *Journal of the Franklin Institute*, 2019.

[6] A. Heydari and S. Balakrishnan, "Optimal switching between controlled subsystems with free mode sequence," *Neurocomputing*, vol. 149, pp. 1620 – 1630, 2015.

[7] T. Sardarmehni and A. Heydari, "Sub-optimal scheduling in switched systems with continuous-time dynamics: A gradient descent approach," *Neurocomputing*, vol. 285, pp. 10 – 22, 2018.

[8] T. Sardarmehni and A. Heydari, "Suboptimal scheduling in switched systems with continuous-time dynamics: A least squares approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2167–2178, June 2018.

[9] A. Heydari, "Optimal switching of DC–DC power converters using approximate dynamic programming," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 3, pp. 586–596, March 2018.

[10] M. Rinehart, M. Dahleh, D. Reed, and I. Kolmanovsky, "Suboptimal control of switched systems with an application to the DISC engine," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 189–201, 2008.

[11] M. Rinehart, M. Dahleh, and I. Kolmanovsky, "Value iteration for (switched) homogeneous systems," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1290–1294, 2009.

[12] K. G. Vamvoudakis and J. Hespanha, "Online optimal switching of single phase DC/AC inverters using partial information," in *American Control Conference (ACC), 2014*, 2014, pp. 2624–2630.

[13] A. Heydari and S. Balakrishnan, "Optimal switching between autonomous subsystems," *Journal of the Franklin Institute*, vol. 351, 2014.

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

[15] A. Heydari and S. N. Balakrishnan, "Optimal switching and control of nonlinear switching systems using approximate dynamic programming," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 6, pp. 1106–1117, June 2014.

[16] T. Sardarmehni and X. Song, "Sub-optimal tracking in switched systems with fixed final time and fixed mode sequence using reinforcement learning," *Neurocomputing*, vol. 420, pp. 197 – 209, 2021. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231220314168

[17] M. Kamgarpour and C. Tomlin, "On optimal control of non-autonomous switched systems with a fixed mode sequence," *Automatica*, vol. 48, no. 6, pp. 1177–1181, 2012.

[18] R. Li, K. Teo, K. Wong, and G. Duan, "Control parameterization enhancing transform for optimal control of switched systems," *Mathematical and Computer Modelling*, vol. 43, no. 11, pp. 1393 – 1403, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0895717705004966

[19] J. Zhai, T. Niu, J. Ye, and E. Feng, "Optimal control of nonlinear switched system with mixed constraints and its parallel optimization algorithm," *Nonlinear Analysis: Hybrid Systems*, vol. 25, pp. 21 – 40, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1751570X17300080

[20] R. C. Loxton, K. L. Teo, and V. Rehbock, "Computational method for a class of switched system optimal control problems," *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2455–2460, 2009.

[21] S. C. Bengea and R. A. DeCarlo, "Optimal control of switching systems," *Automatica*, vol. 41, no. 1, pp. 11 – 27, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109804002237

[22] Q. Lin, R. C. Loxton, and K. L. Teo, "Optimal control of nonlinear switched systems: Computational methods and applications," *Journal of the Operations Research Society of China*, vol. 1, p. 275–311, 2013.

[23] L. Buşoniu, J. Daafouz, M. C. Bragagnolo, and I.-C. Morărescu, "Planning for optimal control and performance certification in nonlinear systems with controlled or uncontrolled switches," *Automatica*, vol. 78, pp. 297 – 308, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109816305350

[24] A. Heydari and S. Balakrishnan, "Fixed-final-time optimal tracking control of input-affine nonlinear systems," *Neurocomputing*, vol. 129, pp. 528 – 539, 2014.

[25] T. Sardarmehni and 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.

[26] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed. McGraw-Hill, 1976.