Proceedings of the ASME 2019
Dynamic Systems and Control Conference
DSCC2019
October 8-11, 2019, Park City, Utah, USA

DSCC2019-9216

SUB-OPTIMAL TRACKING IN SWITCHED SYSTEMS WITH CONTROLLED SUBSYSTEMS AND FIXED-MODE SEQUENCE USING APPROXIMATE DYNAMIC PROGRAMMING

Tohid Sardarmehni
Postdoctoral Research Fellow
Department of Engineering Technology
and Industrial Distribution
Texas A&M University
College Station, TX, 77843, USA
Email: tsardarmehni@tamu.edu

Xingyong Song *
Department of Engineering Technology
and Industrial Distribution;
Department of Mechanical Engineering;
College of Engineering;
Texas A&M University
College Station, TX, 77843, USA
Email: songxy@tamu.edu

ABSTRACT

Optimal tracking in switched systems with controlled subsystem and Discrete-time (DT) dynamics is investigated. A feedback control policy is generated such that a) the system tracks the desired reference signal, and b) the optimal switching time instants are sought. For finding the optimal solution, approximate dynamic programming is used. Simulation results are provided to illustrate the effectiveness of the solution.

1 Introduction

In this study, switched systems are referred to as systems comprised of several subsystems/modes such that at each time instant only one subsystem is active. If the subsystems include continuous control signals, the subsystems are called *controlled* subsystems. In case no control input exists in the subsystems, the subsystems are called *autonomous* subsystems. Control of switched systems with controlled subsystem is a challenging problem as the controller needs to define the switching time and the continuous control in the subsystems. Also, if the sequence of active subsystems. i.e., mode sequence, is known a priori, the mode sequence is called a *fixed mode sequence*. In a fixed mode sequence, the controller assigns the switching times and does not assign the active modes. On the other hand, if the mode sequence is free, then the controller needs to decide which mode should be activated at each time instant. Due to the discontinuous nature of the switched systems, control of these systems are challenging. Control of switched systems addresses many critical problems in automotive engineering [1], power engineering [2, 3], and many other engineering fields.

From the mathematical point of view, solutions of the underlying Hamilton-Jacobi-Bellman (HJB) equation provide the necessary and sufficient condition for optimality in optimal control problems [4]. However, solving the HJB equation explicitly is difficult and most cases impossible. Dynamic Programming (DP) can solve optimal control problems through discretizing the state and control domain and finding the optimal value functions backward in time. However, as the order of the system increases, rapid access to memory becomes prohibitive in DP which is known as *curse of dimensionality*. To remedy the curse of dimensionality, one solution is to seek a near-optimal control solution instead of the exact optimal control which is provided by Approximate Dynamic Programming (ADP). In general, ADP methods use function approximators to approximate the value functions and they use iterative methods to tune the parameters of these function approximators.

ADP solutions for optimal control of switched systems with free mode sequence were investigated in [3, 5–10]. As for the fixed mode sequence, a transformation was introduced in [11] to

*Address all correspondence to this author.

treat the switching times as parameters. This transformation was adapted in [12] and the mode sequence was incorporated in the value functions to solve a regularization problem.

To solve the optimal tracking problem in systems with conventional dynamics, some ADP solutions were introduced in [13, 14]. In [13], an ADP solution based on Single Network Adaptive Critic (SNAC) structure was introduced which approximated the optimal costates and used them to provide the optimal policy. Also, another ADP solution was developed in [14] which approximated the value functions. For this method, a change of variables was performed in the system and a new state vector was introduced which included the tracking error dynamics and the dynamics of the reference signal. Furthermore, in [15,16] model free tracking methods were investigated. As for the switching dynamics, optimal tracking was studied in [2, 6, 7, 17] in switched systems with autonomous subsystems and free mode sequence, and in [18] in switched systems with controlled subsystems and free mode sequence.

In this paper, optimal tracking in switched systems with controlled subsystems and fixed mode sequence is studied. The basic idea in this study is using the transformation introduced in [11] to parametrize the switching instants and then use the idea introduced in [12] to include the mode sequence in the value functions and costates. Hence, the present study combines the results of [11–14] to develop a solution for optimal tracking in switched systems with fixed mode sequence. To solve the optimal tracking problem, a single network is used to capture the optimal costates which are parametrized with respect to the switching instants. Once the optimal costates are known, some recommendations are given to find the optimal switching times from the costates at an initial condition.

The rest of the manuscript is organized as follows. In section 2, optimal control problem formulation and some assumptions are presented. In section 3, the proposed solution based on the SNAC is introduced. Simulation results are discussed in section 4, and section 5 concludes the paper.

2 Problem Formulation

The dynamics of the switched systems can be shown as

$$\begin{aligned} \dot{x}(t) &= \bar{f}_v\big(x(t)\big) + \bar{g}_v\big(x(t)\big)u(t), \\ v \in \mathscr{V} &= \{1,2,\ldots,M\}, x(0) = x_0 \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the input, and t denotes the time. The Lipschitz continuous 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 $v \in \mathscr{V}$. The inclusion of continuous control, i.e., $u(.)$, in (1) shows that the subsystems are controlled subsystems.

Assuming the sequence of active modes is known, it is desired to find the continuous control policy $u(.)$, and the switching times, such that a performance index presented as

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

is minimized. In (2), t_0 is the initial time, and t_f is the final time. $r \in \mathbb{R}^n$ is the reference signal where $\dot{r}(t) = \bar{f}_{r_v}(t)$ is the dynamics of the reference signal and the sub-index r_v shows the active reference dynamics. $S \in \mathbb{R}^{n\times n}$ 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.

Using Euler integration method, by choosing a small sample time $\delta t > 0$, one can discretize the dynamics (1) as

$$x_{k+1} = f_v(x_k) + g_v(x_k)u_k \tag{3}$$

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

$$\begin{aligned} J(x_0) &= (x_N - r_N)^T S (x_N - r_N) \\ &+ \sum_{k=0}^{N} \frac{1}{2}\Big(\big(x_k - r_k\big)^T Q\big(x_k - r_k\big) + u_k^T R u_k\Big) \end{aligned} \tag{4}$$

In (4), $N = \frac{t_f - t_0}{\delta t}$, $Q = \bar{Q}\delta t$, and $R = \bar{R}\delta t$. Based on (4), one can define the cost-to-go as the cost of going from discrete time index k to N as

$$\begin{aligned} J(x_k) &= \big(x_N - r_N\big)^T S\big(x_N - r_N\big) \\ &+ \sum_{\bar{k}=k}^{N} \frac{1}{2}\Big((x_{\bar{k}} - r_{\bar{k}})^T Q (x_{\bar{k}} - r_{\bar{k}}) + u_{\bar{k}}^T R u_{\bar{k}}\Big) \end{aligned} \tag{5}$$

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

Definition 1. *A control policy is called admissible if it stabilizes the system presented in (3) in a selected compact region of interest $\Omega \subset \mathbb{R}^n$, which includes the origin. Also, $\forall x_0 \in \Omega$, $J(x_0)$ should be finite if the state is propagated along that policy.*

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

Considering Assumption 1, one can define the value function as

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

Considering time step k to $k+1$, one has

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

After some algebraic manipulations, one can rewrite (7) as

$$\begin{aligned} V_k(x_k) = \min_{u(.)} \Big(& \frac{1}{2}(x_k - r_k)^T Q (x_k - r_k) + \frac{1}{2} u_k^T R u_k \\ & + V_{k+1}(x_{k+1}) \Big) \end{aligned} \tag{8}$$

Equation (8) simply means minimum cost of going from time k to N equals to cost of going from time k to $k+1$ plus minimum cost of going from time $k+1$ to N. This is in fact the Bellman equation of optimality [4]. Based on (8), one can define the optimal policy as

$$\begin{aligned} u_k(x_k) = \arg\min_{u(.)} \Big(& \frac{1}{2}(x_k - r_k)^T Q (x_k - r_k) + \frac{1}{2} u_k^T R u_k \\ & + V_{k+1}(x_{k+1}) \Big) \end{aligned} \tag{9}$$

2.1 Including the Mode Sequence

In [11], a transformation is introduced to include the switching times as parameters in the optimal control formulation. For ease of presentation, a case with two subsystems and only one switching is considered. Let the switching happen at $t = t_1 \in (t_0, t_f)$. Also, let the mode sequence to be $\{mode\ 1, mode\ 2\}$[1]. To make the switching time instant an independent parameter, let [11]

$$t = \begin{cases} t_0 + (t_1 - t_0)\hat{t} & \text{if } 0 \le \hat{t} < 1 \\ t_1 + (t_f - t_1)(\hat{t} - 1) & \text{if } 1 \le \hat{t} \le 2 \end{cases} \tag{10}$$

From the transformation introduced in (10), 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 $\hat{t} \ge 1$, mode 2 is active and for $\hat{t} < 1$, mode 1 is active. Based on the introduced transformation in (10), one has

$$x'(\hat{t}) = \frac{dx}{d\hat{t}} = \frac{dx}{dt}\frac{dt}{d\hat{t}} \tag{11}$$

Since the mode sequence is known, (11) becomes

$$x'(\hat{t}) = \begin{cases} \Big(\bar{f}_1\big(x(\hat{t})\big) + \bar{g}_1\big(x(\hat{t})\big) u(\hat{t}) \Big)(t_1 - t_0) & \text{if } 0 \le \hat{t} < 1 \\ \Big(\bar{f}_2\big(x(\hat{t})\big) + \bar{g}_2\big(x(\hat{t})\big) u(\hat{t}) \Big)(t_f - t_1) & \text{if } 1 \le \hat{t} \le 2 \end{cases} \tag{12}$$

[1]When $t < t_1$ mode 1 is active and when $t \ge t_1$ mode 2 is active.

Also, the cost function in (2) can be written as[2]

$$\begin{aligned} J(x_0) = & \big(x(2) - r(2)\big)^T S \big(x(2) - r(2)\big) \\ & + \int_0^1 \frac{1}{2} \Big(\big(x(\hat{t}) - r(\hat{t})\big)^T \bar{Q}(t_1 - t_0) \big(x(\hat{t}) - r(\hat{t})\big) \\ & \quad + u(\hat{t})^T \bar{R}(t_1 - t_0) u(\hat{t}) \Big) d\hat{t} \\ & + \int_1^2 \frac{1}{2} \Big(\big(x(\hat{t}) - r(\hat{t})\big)^T \bar{Q}(t_f - t_1) \big(x(\hat{t}) - r(\hat{t})\big) \\ & \quad + u(\hat{t})^T \bar{R}(t_f - t_1) u(\hat{t}) \Big) d\hat{t} \end{aligned} \tag{13}$$

An important observation in (13) is that the transformed cost function is not only a function of x_0, but also it is a function of the switching time, i.e., t_1. Hence, $J(x_0) = J(t_1, x_0)$. With a similar procedure used before, by choosing a small sampling time $\delta\hat{t}$ one can discretize (12) and (13) as

$$x_{\hat{k}+1} = \begin{cases} f_1\big(x_{\hat{k}}\big) + g_1\big(x_{\hat{k}}\big) u_{\hat{k}} & \text{if } 0 \le \hat{k} < \frac{N'}{2} \\ f_2\big(x_{\hat{k}}\big) + g_2\big(x_{\hat{k}}\big) u_{\hat{k}} & \text{if } \frac{N'}{2} \le \hat{k} \le N' \end{cases} \tag{14}$$

where $f_1(x_{\hat{k}}) = x_{\hat{k}} + \bar{f}_1(x_{\hat{k}})(t_1 - t_0)\delta\hat{t}$, $g_1(x_{\hat{k}}) = \bar{g}_1(x_{\hat{k}})(t_1 - t_0)\delta\hat{t}$, $f_2(x_{\hat{k}}) = x_{\hat{k}} + \bar{f}_2(x_{\hat{k}})(t_f - t_1)\delta\hat{t}$, and $g_2(x_{\hat{k}}) = \bar{g}_2(x_{\hat{k}})(t_f - t_1)\delta\hat{t}$. In (14), $\hat{k} \in [0, N']$ is the discrete time index where $N' = \frac{number\ of\ switching + 1}{\delta\hat{t}}$ [12]. With the transformed dynamics and the cost function, with similar procedure used in the previous section one can define cost-to-go as

$$V_{\hat{k}}(t_1, x_{\hat{k}}) = \begin{cases} \mathbf{Q}_1 + \mathbf{R}_1 + V_{\hat{k}+1}(t_1, x_{\hat{k}+1}) & \text{if } 0 \le \hat{k} < \frac{N'}{2} \\ \mathbf{Q}_2 + \mathbf{R}_2 + V_{\hat{k}+1}(t_1, x_{\hat{k}+1}) & \text{if } \frac{N'}{2} \le \hat{k} \le N' \end{cases} \tag{15}$$

where

$$\begin{aligned} \mathbf{Q}_1 &= \frac{1}{2}(x_{\hat{k}} - r_{\hat{k}})^T \bar{Q}(t_1 - t_0)\delta\hat{t}(x_{\hat{k}} - r_{\hat{k}}) \\ \mathbf{R}_1 &= \frac{1}{2} u_{\hat{k}}^T \bar{R}(t_1 - t_0)\delta\hat{t} u_{\hat{k}} \\ \mathbf{Q}_2 &= \frac{1}{2}(x_{\hat{k}} - r_{\hat{k}})^T \bar{Q}(t_f - t_1)\delta\hat{t}(x_{\hat{k}} - r_{\hat{k}}) \\ \mathbf{R}_2 &= \frac{1}{2} u_{\hat{k}}^T \bar{R}(t_f - t_1)\delta\hat{t} u_{\hat{k}} \end{aligned} \tag{16}$$

As one can see, the cost-to-go in (15) is a function of current time $\hat{k}$, current state $x_{\hat{k}}$, and the switching time t_1. Similarly, one can define the costate as[3]

$$\lambda_{\hat{k}}(t_1, x_{\hat{k}}) = \begin{cases} \bar{\mathscr{Q}}_1 + \frac{\partial x_{\hat{k}+1}}{\partial x_{\hat{k}}} \lambda_{\hat{k}+1}(t_1, x_{\hat{k}+1}) & \text{if } 0 \le \hat{k} < \frac{N'}{2} \\ \bar{\mathscr{Q}}_2 + \frac{\partial x_{\hat{k}+1}}{\partial x_{\hat{k}}} \lambda_{\hat{k}+1}(t_1, x_{\hat{k}+1}) & \text{if } \frac{N'}{2} \le \hat{k} \le N' \end{cases} \tag{17}$$

[2]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.

[3]By definition, costate is the gradient of value function, i.e., $\lambda(x) = \frac{\partial V(x)}{\partial x}$. Taking the gradient of (15) leads (17).

where

$$\begin{aligned}\bar{\mathscr{Q}}_1 &= \bar{Q}(t_1 - t_0)\delta\hat{t}\left(x_{\hat{k}} - r_{\hat{k}}\right)\\ \bar{\mathscr{Q}}_2 &= \bar{Q}(t_f - t_1)\delta\hat{t}\left(x_{\hat{k}} - r_{\hat{k}}\right)\end{aligned} \tag{18}$$

3 Single Network Adaptive Critic (SNAC)

The application of SNAC was introduced for tracking in systems with conventional dynamics [13]. This idea is adapted in this section to perform tracking in a switched system. To introduce the concept, consider the costate as in (17). The idea here is training neural networks to approximate $\lambda_{\hat{k}+1}(t_1, x_{\hat{k}+1})$ from $(t_1, x_{\hat{k}})$. Based on Weierstrass Approximation Theorem [19], linear-in-parameter neural networks with polynomial basis functions can uniformly [20] approximate continuous functions to a desired degree of precision in a compact set. In order to use Weierstrass Approximation Theorem, the following assumption is required.

Assumption 2. *The value functions are continuously differentiable.*

Through Assumption 2, one can use linear-in-parameter neural networks to approximate the value functions, and the costates. Consider the exact costate at discrete time index $\hat{k}$ as

$$\lambda_{\hat{k}+1}(t_1, x_{\hat{k}+1}) = W_{\hat{k}}^{*T}\phi(t_1, x_{\hat{k}}) + \varepsilon_{\hat{k}}^{*}(t_1, x_{\hat{k}}) \tag{19}$$

where $W_{\hat{k}}^{*} \in \mathbb{R}^{m_\lambda \times n}$ is a weight vector and $\phi : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^{m_\lambda}$ is a vector of linearly independent polynomial basis functions (neurons). The number of neurons is denoted by positive integer m_λ. In (19), the dependence of the parameters/functions to discrete time index is shown with a sub-index $\hat{k}$. Hence the approximate costates can be calculated as

$$\widehat{\lambda}_{\hat{k}+1}(t_1, x_{\hat{k}+1}) = \widehat{W}_{\hat{k}}^{T}\phi(t_1, x_{\hat{k}}) \tag{20}$$

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

$$\widehat{u}_{\hat{k}}(t_1, x_{\hat{k}}) = \begin{cases} -R_1^{-1} g_1^T\left(x_{\hat{k}}\right)\widehat{\lambda}_{\hat{k}+1}(t_1, x_{\hat{k}+1}) & \text{if } 0 \le \hat{k} < \frac{N'}{2} \\ -R_2^{-1} g_2^T\left(x_{\hat{k}}\right)\widehat{\lambda}_{\hat{k}+1}(t_1, x_{\hat{k}+1}) & \text{if } \frac{N'}{2} \le \hat{k} \le N' \end{cases} \tag{21}$$

In (21), $R_1 = R\delta\hat{t}(t_1 - t_0)$ and $R_2 = R\delta\hat{t}(t_f - t_1)$. For training, one can go backward in time and find the costates and save them for online control. This process is summarized in Algorithm 1.

Remark 1. The convergence of the inner loop in step 3 of Algorithm 1 was studied in theorem 1 of [13] for systems with conventional dynamics. The proof can be modified to include tracking in switched systems.

Remark 2. 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 word, one does not need to integrate again when the initial condition is changed. This is unlike Method 1 that propagation should be done for each initial condition separately. However, as the order of system increases, this method becomes very complicated. In other words, this method is only 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 time. 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

---

Algorithm 1 : *Finding the Costates (Approach 1)*

step 1: Initialize the neural network weights, $\widehat{W}_{\hat{k}}^{0}, \hat{k} \in \{1, \ldots, N-1\}$. Also select a small positive number γ as a convergence tolerance. Select η random training samples $x^{[l]} \in \Omega$ where $l \in \{1, 2, ..., \eta\}$ and $t_1^{[l]} \in [t_0, t_f]$ where $l \in \{1, 2, ..., \eta\}$.

step 2: Repeat the outer loop for $\hat{k} = N-1$ to $\hat{k} = 1$:

step 3: Set $i = 0$ and repeat the following inner loop:

step 3-1: Select η random training samples $\{(t_1^{[l]}, x^{[l]}) \in (t_0, t_f) \times \Omega$ where $l \in \{1, 2, ..., \eta\}$. Substitute all the training samples in $\phi(.,.)$ and find a $\lambda_{\hat{k}+1}(t_1, x_{\hat{k}})$. With $\lambda_{\hat{k}+1}(t_1, x_{\hat{k}})$ find $u_{\hat{k}}$ and propagate the states along it to find $x_{\hat{k}+1}$. Also, find $r_{\hat{k}+1}$.

step 3-2: If $\hat{k} = N-1$ set the target as $\widehat{\lambda}_{\hat{k}+1}(t_1, x_{\hat{k}+1}) = S(x_{N'} - r_{N'})$ and train $\widehat{W}_{\hat{k}}^{i+1}$ such that $\widehat{W}_{\hat{k}}^{i+1^T}\phi(t_1, x_{\hat{k}}) = \widehat{\lambda}_{\hat{k}+1}(t_1, x_{\hat{k}+1})$ with least squares on the entire set of training samples. Otherwise, set the target as right-hand side of (17) and use the least squares to find $\widehat{W}_{\hat{k}}^{i+1^T}$.

step 3-3: If $\|\widehat{W}_{\hat{k}}^{i+1} - \widehat{W}_{\hat{k}}^{i}\| \le \gamma$, go to step 4. Otherwise, set $i = i+1$ and go back to step 3-1.

step 4: If $\hat{k} = 1$, stop the training. Otherwise, store $\widehat{W}_{\hat{k}}^{i+1}$, set $\hat{k} = \hat{k} - 1$ and go to step 2.

---

might become very time-consuming.

4 Numerical Simulation

In this section some simulation results are provided to evaluate the effectiveness of the solutions discussed in this paper. Consider a system with two modes. For the first mode, a benchmark system, Van der Pol oscillator was selected. The dynamics of this mode can be shown as

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= \left(1 - x_1^2(t)\right)x_2(t) - x_1(t) + u(t) \end{aligned} \tag{22}$$

For the second mode, a linear subsystem was selected as

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= 2x_1 - x_2 + u(t) \end{aligned} \tag{23}$$

The reference signal is chosen as

$$\begin{aligned} \dot{r}_1(t) &= \sin(\pi t) \\ \dot{r}_2(t) &= \pi\cos(\pi t) \end{aligned} \tag{24}$$

For the cost function, $S = \mathrm{diag}(10^5, 10^5)$, $\bar{Q} = \mathrm{diag}(10^5, 10^7)$ were selected, where $\mathrm{diag}(a,b)$ denotes a diagonal matrix with a and b on the main diagonal and 0 elsewhere. Also, $\bar{R} = 1/\delta\hat{t}$ was selected as the control penalizing term. The basis functions were selected as all possible combination of t_1, x_1, x_2 up to the power 3 without repetition. For training, 1000 random samples were generated in $\Omega = \{(t_1, x_1, x_2) | t_1 \in (0,3), x_1 \in [-4,4], x_2 \in [-4,4]\}$. For descritization, sampling time was selected as $\delta\hat{t} = 0.001$ (sec). Using approach 1, The training process concluded in 22.6177 seconds using an office desktop with 16 GB of RAM and Intel(R) Core(TM) i7-3770 CPU @ 3.4 GHz. The history of the weights of the neural networks to approximate the costates is shown in Fig. 1. As one can see from Fig 1, the history of the weights shows a jump at $\hat{t} = 1$ which is the switching time.

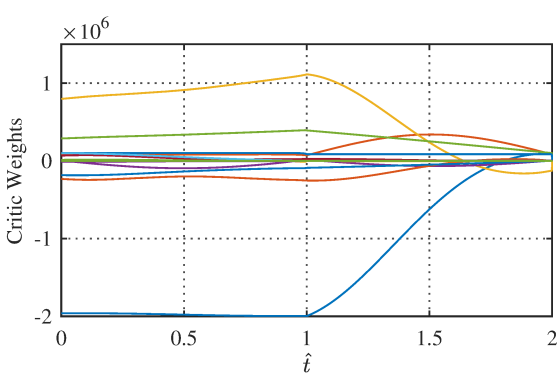

FIGURE 1: The history of the weights of the neural networks to approximate costates. The jump at $\hat{t} = 1$ shows the switching at this time.

Once the training concluded, the trained costates were used to find the optimal switching time. For this example, Method 2 in Remark 2, i.e., integration of the costate analytically, was used to find the value function and then minimized the value function with nonlinear programming methods. Through integration, one can find the analytical value function as $V(t_1, x_1, x_2) = 67377t_1^3x_1 + 568.7t_1^3x_2 + 49288t_1^2x_1^2 + 849t_1^2x_1x_2 - 2.322 \times 10^5 t_1^2x_1 + 57.67t_1^2x_2^2 - 2077t_1^2x_2 + 4210t_1x_1^3 + 162.2t_1x_1^2x_2 - 92177t_1x_1^2 + 1.432t_1x_1x_2^2 - 1691t_1x_1x_2 + 7.968 \times 10^5 t_1x_1 + 0.3332t_1x_2^3 - 2411t_1x_2^2 + 3337t_1x_2 + 2413x_1^4 + 97.95x_1^3x_2 - 4013x_1^3 - 434.8x_1^2x_2^2 - 182.2x_1^2x_2 + 1.44 \times 10^5 x_1^2 - 2.424x_1x_2^3 + 2.186x_1x_2^2 + 1789x_1x_2 - 1.961 \times 10^6 x_1 - 0.1953x_2^4 - 0.08602x_2^3 + 50288x_2^2 + 77666x_2$. Substituting the initial conditions for $x_1 = 1$ and $x_2 = -0.5$, one can find the value function as a function of the switching time as $V(t_1) = 67088t_1^3 - 1.823 \times 10^5 t_1^2 + 7.073 \times 10^5 t_1 - 1.846 \times 10^6$. Using nonlinear programming, the best switching time is sought as $t_1 = 2.654$ (sec).

The history of the states in the transformed dynamics is shown in Fig. 2. As one can see, the controller had a very good performance in forcing $x_2(.)$ to track $r_2(.)$ as in the cost function it was emphasized.

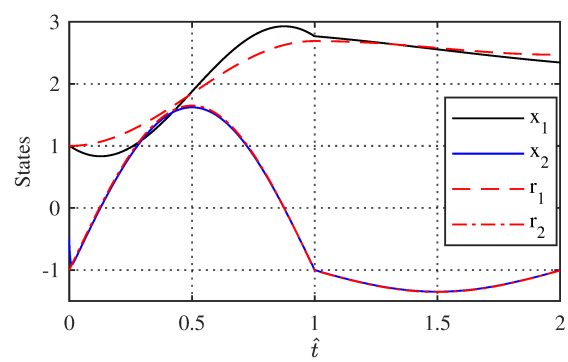

FIGURE 2: The history of the states in the transformed time.

5 Conclusion

An approximate solution for optimal control of switched systems with fixed mode sequence and controlled subsystems was presented. The method includes two levels of control. In the upper level, optimal switching times were sought. In the lower level, continuous control for each mode was generated in a feedback form. To find the continuous control, a single network adaptive critic was used to find the optimal costates while treating the switching times as parameters. Simulation results confirmed the effectiveness of the solution.

ACKNOWLEDGMENT

This research was partially supported by the National Academies of Sciences, Engineering and Medicine GRP Early Career Research Fellow Award.

REFERENCES

[1] Sardarmehni, T., and Heydari, A., 2019. "Sub-optimal switching in anti-lock brake systems using approximate dynamic programming". *IET Control Theory & Applications*, January.

[2] Khiabani, A. G., and Heydari, A., 2018. "Optimal switching of voltage source inverters using approximate dynamic programming". In ASME 2018 Dynamic Systems and Control Conference (DSCC 2018), pp. V001T01A006–V001T01A015.

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

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

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

[6] Heydari, A., 2015. "Optimal scheduling for reference tracking or state regulation using reinforcement learning". *Journal of the Franklin Institute,* ***352***(8), pp. 3285–3303.

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

[8] Sardarmehni, T., and Heydari, A., 2018. "Suboptimal scheduling in switched systems with continuous-time dynamics: A least squares approach". *IEEE Transactions on Neural Networks and Learning Systems,* ***29***(6), June, pp. 2167–2178.

[9] Lu, W., Zhu, P., and Ferrari, S., 2015. "A hybrid-adaptive dynamic programming approach for the model-free control of nonlinear switched systems". *IEEE Transactions on Automatic Control,* ***PP***(99), pp. 1–1.

[10] Rinehart, M., Dahleh, M., and Kolmanovsky, I., 2009. "Value iteration for (switched) homogeneous systems". *IEEE Transactions on Automatic Control,* ***54***(6), pp. 1290–1294.

[11] Xu, X., and Antsaklis, P. J., 2004. "Optimal control of switched systems based on parameterization of the switching instants". *IEEE Transactions on Automatic Control,* ***49***(1), pp. 2–16.

[12] Heydari, A., and Balakrishnan, S. N., 2014. "Optimal switching and control of nonlinear switching systems using approximate dynamic programming". *IEEE Transactions on Neural Networks and Learning Systems,* ***25***(6), June, pp. 1106–1117.

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

[14] Kiumarsi, B., and Lewis, F. L., 2015. "Actor–critic-based optimal tracking for partially unknown nonlinear discrete-time systems". *IEEE Transactions on Neural Networks and Learning Systems,* ***26***(1), Jan, pp. 140–151.

[15] Luo, B., Liu, D., Huang, T., and Wang, D., 2016. "Model-free optimal tracking control via critic-only Q-learning". *IEEE Transactions on Neural Networks and Learning Systems,* ***27***(10), Oct, pp. 2134–2144.

[16] Kiumarsi, B., Lewis, F. L., Naghibi-Sistani, M., and Karimpour, A., 2015. "Optimal tracking control of unknown discrete-time linear systems using input-output measured data". *IEEE Transactions on Cybernetics,* ***45***(12), Dec, pp. 2770–2779.

[17] Gan, M., Zhang, C., and Zhao, J., 2019. "Data-driven optimal switching of switched systems". *Journal of the Franklin Institute,* ***356***(10), pp. 5193 – 5221.

[18] Zhang, K., Zhang, H., Liang, Y., and Wen, Y., 2019. "A new robust output tracking control for discrete-time switched constrained-input systems with uncertainty via a critic-only iteration learning method". *Neurocomputing*.

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

[20] Hornik, K., Stinchcombe, M., and White, H., 1989. "Multilayer feedforward networks are universal approximators". *Neural Networks,* ***2***(5), pp. 359–366.