



# Article Data-Driven Suboptimal Scheduling of Switched Systems

# Chi Zhang<sup>D</sup>, Minggang Gan \*, Jingang Zhao<sup>D</sup> and Chenchen Xue

State Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Automation, Beijing Institute of Technology, Beijing 100081, China; zhangchibit@foxmail.com (C.Z.); zhaojingang521@126.com (J.Z.); xuechchcalt@163.com (C.X.)

\* Correspondence: agan@bit.edu.cn

Received: 5 February 2020; Accepted: 25 February 2020; Published: 27 February 2020



**Abstract:** In this paper, a data-driven optimal scheduling approach is investigated for continuous-time switched systems with unknown subsystems and infinite-horizon cost functions. Firstly, a policy iteration (PI) based algorithm is proposed to approximate the optimal switching policy online quickly for known switched systems. Secondly, a data-driven PI-based algorithm is proposed online solely from the system state data for switched systems with unknown subsystems. Approximation functions are brought in and their weight vectors can be achieved step by step through different data in the algorithm. Then the weight vectors are employed to approximate the switching policy and the cost function. The convergence and the performance are analyzed. Finally, the simulation results of two examples validate the effectiveness of the proposed approaches.

Keywords: optimal switching; data-driven control; switched systems; policy iteration; continuous time

# 1. Introduction

Switched systems consisting of several subsystems and a switching policy ruling the switching among them [1,2] arise in certain application situations [3,4] such as a system which has to collect data sequentially from a number of sensory sources and switches its attention among the data sources [5,6]. The switching among subsystems complicates the control problems and many of the problems remain to be open such as the optimal control problems. Optimal control [7,8] problems of switched systems have attracted considerable attention over the past few years. Thereinto, the optimal scheduling problem of switched systems is investigated in this paper.

Multiple approaches have been introduced to solve the optimal control problems for switched systems. Gradient-based approaches are investigated to solve the optimal switching time problems [9] and optimal scheduling problems [10,11] directly, usually in a finite time horizon. By utilizing control inputs to represent the switching policy, embedding approaches transform the optimal control problems of switched systems to traditional optimal control problems to address [12–14]. Adaptive dynamic programming (ADP) [15] approaches are introduced to solve the optimal control problems for switched systems with different initial conditions directly [16–22]. For optimal scheduling of switched systems with infinite-horizon cost functions, ADP approaches perform well to provide approximate global optimal solutions directly.

Approximate global optimal solutions are derived in feedback forms through ADP for optimal scheduling of switched systems with discrete-time dynamics and finite-horizon cost functions [17] or infinite-horizon cost functions [18]. Then further research is conducted for problems with switching cost [19] or state jumps [20]. For optimal scheduling of switched systems with continuous-time dynamics, an approximate feedback solution is proposed based on policy iteration (PI) algorithm

with its offline, online, and concurrent implementation [21]. Then, a PI algorithm with recursive least squares is proposed and modified into a single loop PI algorithm to reduce the computational burden [22].

The aforementioned optimal control approaches are deduced based on the a priori knowledge of system models. However, not all system models can be completed acquired so that approaches independent of system models [23–25] require investigated. For this purpose, some model-free optimal control approaches have been studied for switched systems. Adaptive dynamic programming approaches are presented respectively for a continuous-time switched system with an infinite-horizon cost function [26] and a discrete-time switched system with a finite-horizon cost function [27] under the assumption that dynamic equations can be evaluated at some sets. Gradient-decent approaches only employing state data are proposed to solve optimal switching problems for continuous-time switched systems with finite-horizon cost functions [28,29]. Data-driven research utilizes real-life data measured by sensory sources to achieve the intrinsic information of systems [30] and can solve the problems of switched systems with unknown subsystems, such as the data-driven framework for discovering cyber-physical systems directly from the data [31]. Thereinto, data-driven ADP approaches provide possible solutions for optimal control problems of switched systems [32–34] and based on that, the following approach is designed to adapt to the complexity of switched systems which is brought by switching. Then in this paper, the data-driven optimal scheduling approach is investigated for continuous-time switched systems with unknown subsystems. At first, an online PI-based algorithm inspired by the off-policy learning method [35,36] is proposed to approximate the optimal solution quickly for optimal scheduling problems with known system models first and based on that, a data-driven PI-based algorithm is formulated for optimal scheduling of continuous-time switched systems with infinite-horizon cost functions, which don't require that dynamic equations can be evaluated or known at some sets. Moreover, common online algorithms usually keep collecting data with the updating switching policies applied to the system at each iteration while the online algorithms in this paper take advantages of the data produced only by the initial switching policy.

The contribution of this paper is stated as follows: (1) with the data produced by the initial switching policy, an online PI-based algorithm is proposed to approximate the optimal solution quickly for optimal scheduling of known switched systems. (2) A data-driven PI-based algorithm is designed to solve optimal scheduling problems for switched systems with infinite-horizon cost functions and unknown subsystems, solely from the data produced by the initial switching policy, which has not been achieved well in existing literature as far as we know. (3) The convergence is proved and the optimality is analyzed.

The remainder of the paper is organized as follows. In Section 2, the problem is stated and the classic PI approach is introduced for the optimal scheduling problem. In Section 3, online PI-based algorithms are proposed for switched systems with known subsystems and unknown subsystems. In Section 4, simulation results are shown to indicate the effectiveness of the algorithms. Finally, the conclusion is drawn in Section 5.

## 2. Preliminaries

Consider the switched system as follows:

$$\dot{x}(t) = f_v(x(t)), x(0) = x_0,$$
(1)

where  $x(t) \in \mathbb{R}^n$  is the system state,  $v \in V$  represents the index of the active subsystem,  $V = \{1, 2, ..., N\}$  is the index set of all subsystems, N is the number of subsystems, and  $f_v : \mathbb{R}^n \to \mathbb{R}^n$  denotes the unknown dynamics of subsystem v.  $f_v(v \in V)$  is Lipschitz continuous in  $\Omega$  where  $\Omega \subset \mathbb{R}^n$  including the origin is the region of interest and there exists  $v \in V$  such that  $f_v(0) = 0$ .

In this paper, the problem to be addressed consists in seeking out the optimal switching policy  $v^*$  to minimize the following cost function:

$$J = \int_0^\infty Q(x(\tau))d\tau,$$
(2)

where  $Q : \mathbb{R}^n \to \mathbb{R}$  is a positive definite function. During the time interval, the cost-to-go from the time *t* to infinity with the state x(t) at time *t* can be described as [37]:

$$V(x(t)) = \int_{t}^{\infty} Q(x(\tau)) d\tau$$
(3)

and then

$$V(x(t)) = \int_{t}^{t+\delta t} Q(x(\tau))d\tau + V(x(t+\delta t)).$$
(4)

For the optimal problem, the admissible switching policy should be introduced and the relevant assumption is made as [22].

**Definition 1.** For system (1), a switching policy is called admissible with respect to the cost (2), if it stabilizes the system in  $\Omega$  and for all  $x_0 \in \Omega$ , the cost  $V(x_0)$  is finite.

**Assumption 1.** There exists at least one admissible policy for the system.

According to the Bellman principle of optimality, the optimal cost-to-go can be represented by

$$V^{*}(x(t)) = \min_{v} (\int_{t}^{t+\delta t} Q(x(\tau))d\tau + V^{*}(x(t+\delta t))),$$
(5)

and when  $\delta t \to 0$ , the corresponding optimal switching policy  $v^*$  can be represented in a state-feedback form:

$$v^{*}(x) = \arg\min_{v} (\int_{t}^{t+\delta t} Q(x(\tau)) d\tau + V^{*}(x(t+\delta t))).$$
(6)

When  $\delta t \rightarrow 0$ , with first-order Taylor expansion of  $V^*(x(t + \delta t))$  applied in (5) and (6), the HJB equation can be given as [21]:

$$\min_{v(x)}(Q(x) + (\frac{\partial V^*(x)}{\partial x})^T f_{v(x)}(x)) = Q(x) + (\frac{\partial V^*(x)}{\partial x})^T f_{v^*(x)}(x) = 0,$$

where x = x(t),  $\frac{\partial(\cdot)}{\partial x} = [\frac{\partial(\cdot)}{\partial x_1}, \frac{\partial(\cdot)}{\partial x_2}, \dots, \frac{\partial(\cdot)}{\partial x_n}]^T$  when  $(\cdot)$  is a scalar and the corresponding optimal switching policy  $v^*$  is given as [21]:

$$v^*(x) = \arg\min_{i \in V} \left( \left( \frac{\partial V(x)}{\partial x} \right)^T f_i(x) \right).$$

Then, a PI approach can be applied to solve the optimal scheduling problem. Given an admissible switching policy  $v^0$ , the PI approach [21,22] is stated as follows:

$$Q(x) + \left(\frac{\partial V^k(x)}{\partial x}\right)^T f_{v^k(x)}(x) = 0$$
(7)

$$v^{k+1}(x) = \arg\min_{i \in V} \left( \left( \frac{\partial V^k(x)}{\partial x} \right)^T f_i(x) \right), \tag{8}$$

where *k* is the iteration number. The cost-to-go is solved in policy evaluation (7) and the switching policy is updated in policy improvement (8) iteratively. The stability and convergence are stated in the following theorem which has been proved in [21,38].

**Theorem 1.** For the system (1) with the cost (2), if the value function sequence  $\{V^k\}_{k=0}^{\infty}$  and the switching policy sequence  $\{v^k\}_{k=0}^{\infty}$  are generated through (7) and (8) initiating from a stabilizing initial policy  $v^0$ , the value functions  $\{V^k\}_{k=0}^{\infty}$  converge to the optimal value function  $V^*$ , and the switching policies  $\{v^k\}_{k=0}^{\infty}$  stabilize the system in  $\Omega$ .

## 3. Main Results

#### 3.1. PI-Based Algorithm for Known Switched Systems

The offline, online and concurrent implementation of the aforementioned PI approach for switched systems with known dynamics has been investigated in references [21,22,38]. In this subsection, a novel online PI-based algorithm is proposed to approximate the optimal switching policy quickly.

Common online algorithms keep collecting data from the systems to evaluate and improve the switching policy. To be specific, once a new switching policy is produced at each iteration, it is applied to the system and the newly produced data is collected to calculate a new cost and a new switching policy. It entails considerable time to apply the new switching policy, collect new data and calculate a new cost and a new switching policy sequentially at each iteration. Aiming at this, inspired by the off-policy ADP methods for ordinary systems [36,39,40], an online PI-based algorithm is proposed to approximate the optimal switching policy quickly with only an initial switching policy applied and only the data produced by the initial switching policy is required. Next, the algorithm starts from the initial admissible switching policy.

With the initial admissible switching policy  $v^0$  applied only, large amounts of data can be produced in the process and the state trajectory x(t) corresponding to the switching policy  $v^0$  can be acquired which will be employed in the subsequent derivation. Along the acquired state trajectory x(t),  $V^k(x(t + \delta t)) - V^k(x(t))$  can be represented as

$$V^{k}(x(t+\delta t)) - V^{k}(x(t)) = \int_{t}^{t+\delta t} \left(\frac{\partial V^{k}(x)}{\partial x}\right)^{T} f_{v^{0}(x)}(x) d\tau,$$
(9)

where  $\delta t > 0$  is very small. To combine the policy iteration and the cost at the acquired state, integrating (7) along the acquired state trajectory x(t) and adding the integration to (9) yield

$$V^{k}(x(t+\delta t)) - V^{k}(x(t)) = \int_{t}^{t+\delta t} \left(\frac{\partial V^{k}(x)}{\partial x}\right)^{T} (f_{v^{0}(x)}(x) - f_{v^{k}(x)}(x))d\tau - \int_{t}^{t+\delta t} Q(x)d\tau.$$
(10)

The cost  $V^k(x)$  is unknown. However, for all  $x \in \Omega$ , it can be expressed by:

$$V^k(x) = W^k \Phi(x) + e^k_{\Phi}(x), \tag{11}$$

where  $\Phi(x) = [\Phi_1(x), \Phi_2(x), \dots, \Phi_{N_w}(x)]^T$  is a vector concerning a set of linearly independent basis functions  $\Phi_j(x) : \mathbb{R}^n \to \mathbb{R}(j = 1, 2, \dots, N_w), W^k \in \mathbb{R}^{1 \times N_w}$  is the weight vector and  $e_{\Phi}^k(x)$  is the approximation error.  $N_w$  is the number of the basis functions. A set of basis functions can constitute a particular basis of a function space and can almost approximate any function in the function space. With the approximation function (11) applied, Equation (10) can be transformed into

$$W^{k}[\Phi(x(t+\delta t)) - \Phi(x(t)) + \int_{t}^{t+\delta t} \frac{\partial \Phi(x)}{\partial x} (f_{v^{k}(x)}(x) - f_{v^{0}(x)}(x))d\tau] = -\int_{t}^{t+\delta t} Q(x)d\tau + e_{1}^{k}(t), \quad (12)$$

where  $\frac{\partial(\cdot)}{\partial x} = \begin{bmatrix} \frac{\partial(\cdot)_1}{\partial x}, \frac{\partial(\cdot)_2}{\partial x}, \dots, \frac{\partial(\cdot)_m}{\partial x} \end{bmatrix}^T \in \mathbb{R}^{m \times n}$  when  $(\cdot) = [(\cdot)_1, (\cdot)_2, \dots, (\cdot)_m]^T \in \mathbb{R}^m$  and  $e_1^k(t) = e_{\Phi}^k(x(t)) - e_{\Phi}^k(x(t+\delta t)) + \int_t^{t+\delta t} \frac{\partial e_{\Phi}^k(x)}{\partial x} (f_{v^0(x)}(x) - f_{v^k(x)}(x)) d\tau$  is the approximation error. Then the estimation can be achieved:

$$\hat{W}^{k}[\Phi(x(t+\delta t)) - \Phi(x(t)) + \int_{t}^{t+\delta t} \frac{\partial \Phi(x)}{\partial x} (f_{\vartheta^{k}(x)}(x) - f_{\vartheta^{0}(x)}(x)) d\tau] = -\int_{t}^{t+\delta t} Q(x) d\tau,$$
(13)

where the estimate  $\hat{v}^k(x)$  is achieved through substituting the estimate  $\hat{W}^k$  and the approximation (11) into (8) as follows:

$$\hat{v}^{k+1}(x) = \arg\min_{i \in V} (\hat{W}^k \frac{\partial \Phi(x)}{\partial x} f_i(x)), \tag{14}$$

with the initial switching policy estimate  $\hat{v}^0(x) = v^0(x)$ .

To employ the acquired state data corresponding to some selected instants, some data matrices can be defined as follows:  $\overline{\Phi}(x_i) = \Phi(x_i(x_i), x_i(x_i)) = \Phi(x_i(x_i))$ 

$$\Phi(t_r) = \Phi(x(t_r + \delta t)) - \Phi(x(t_r))$$
$$b(t_r) = \int_{t_r}^{t_r + \delta t} \frac{\partial \Phi(x)}{\partial x} (f_{\partial^k(x)}(x) - f_{v^0(x)}(x)) d\tau$$
$$d(t_r) = \int_{t_r}^{t_r + \delta t} Q(x) d\tau$$

where  $t_1 < t_2 < \cdots < t_l$  are the selected instants, l is a positive integer,  $r = 1, 2, \ldots l$ ,  $\bar{\Phi}(t_r) \in \mathbb{R}^{N_w}$ ,  $b(t_r) \in \mathbb{R}^{N_w}$  and  $d(t_r) \in \mathbb{R}$ . The following assumption concerning the data matrices is made as [40,41]:

**Assumption 2.** There exist a positive integer  $\overline{L}$  and a positive number  $\alpha$  such that for all  $L \ge \overline{L}$ , the following equality holds:

$$\frac{1}{L}\sum_{r=1}^{L}\left(\bar{\Phi}(t_r)+b(t_r)\right)\left(\bar{\Phi}(t_r)+b(t_r)\right)^T\geq \alpha I_{N_w}.$$

In optimal control of ordinary systems, this kind of assumption can be satisfied by exerting an exploration noise in the input [39,40]. In the case of the switched systems, according to [21,22], it can be satisfied through random switching.

Then based on Assumption 2,  $\hat{W}^k$  can be achieved with the data matrices as the following formula:

$$\hat{W}^{k} = -\left[\sum_{r=1}^{L} d(t_{r})(\bar{\Phi}(t_{r}) + b(t_{r}))^{T}\right] \cdot \left[\sum_{r=1}^{L} \left(\bar{\Phi}(t_{r}) + b(t_{r})\right)(\bar{\Phi}(t_{r}) + b(t_{r}))^{T}\right]^{-1}.$$
(15)

According to the analysis, the online PI-based algorithm for switched systems with known dynamics can be formulated in Algorithm 1.

# Algorithm 1 Online policy iteration (PI)-based Algorithm.

Step 1. Start with an initial admissible switching policy  $v^0(x)$  and set the iteration index k = 0. Step 2. Apply  $v^0(x)$  in the switched systems and acquire the state data. Set  $\hat{v}^0(x) = v^0(x)$ . Calculate  $\bar{\Phi}(t_r)$  and  $d(t_r)$  for r = 1, 2, ..., l according to their definition with the state data. Step 3. Calculate  $b(t_r)$  for r = 1, 2, ..., l according to its definition with the state data and then calculate  $\hat{W}^k$  from (15). Step 4. Update the switching policy  $\hat{v}^{k+1}(x)$  as (14). Step 5. If  $\|\hat{W}^k - \hat{W}^{k-1}\| < \varepsilon$ , set  $\hat{W}^* = \hat{W}^k$  and exit. Otherwise, set k = k + 1 and go back to Step 3.

In the algorithm, when the switching policy  $v^0(x)$  is applied, the corresponding system state data can be obtained. With multiple samples, multiple data matrix  $\bar{\Phi}(t_r)$  and  $b(t_r)$  will be calculated.

Sufficient samples should be employed to satisfy that  $\frac{1}{L}\sum_{r=1}^{L} (\bar{\Phi}(t_r) + b(t_r))(\bar{\Phi}(t_r) + b(t_r))^T \ge \alpha I_{N_w}$ with a proper  $\alpha$  so that the sampling stage must be long enough to ensure that. Moreover, since the weight vector  $\hat{W}^k$  to be solved in Equation (15) has  $N_w$  components, at least  $N_w$  instants  $t_r$  should be employed to solve  $\hat{W}^k$  so that r is no less than  $N_w$ . Then through the algorithm, the approximate optimal cost function can be achieved as  $\hat{V}^*(x) = \hat{W}^* \Phi(x)$  with the corresponding approximate optimal switching policy  $\hat{\vartheta}^*(x) = \arg\min_{i \in V} (\hat{W}^* \frac{\partial \Phi(x)}{\partial x} f_i(x))$ . It can be seen from Algorithm 1 that only the initial switching policy is applied and then Algorithm 1 can approximate the optimal switching policy with the data produced by the initial switching policy.

#### 3.2. Data-Driven PI-Based Algorithm for Unknown Switched Systems

In this subsection, based on the proposed PI approach dependent on system models, a data-driven PI-based algorithm is proposed for switched systems with unknown subsystems. The algorithm takes full advantage of data produced by an initial switching policy to approximate the optimal switching policy quickly.

From Section 3.1, with the initial admissible switching policy  $v^0$  applied and along the acquired state trajectory x(t), (10) can be achieved and the cost function can be represented by (11). In the process,  $\frac{\partial V^k(x)}{\partial x}$  can be represented by  $(W^k \frac{\partial \Phi(x)}{\partial x})^T$  and solved with known system models. However, due to the unknown subsystem models,  $W^k \frac{\partial \Phi(x)}{\partial x} f_i(x)$  representing  $(\frac{\partial V^k(x)}{\partial x})^T f_i(x)$  cannot work well as Section 3.1. So another approximation function is brought in to solve the problem. For all  $x \in \Omega$ , the unknown variable  $(\frac{\partial V^k(x)}{\partial x})^T f_i(x)$ ,  $i \in V$  can be expressed by:

$$\left(\frac{\partial V^k(x)}{\partial x}\right)^T f_i(x) = C_i^k \Psi(x) + e_{\Psi,i}^k(x), \tag{16}$$

where  $\Psi(x) = [\Psi_1(x), \Psi_2(x), \dots, \Psi_{N_c}(x)]^T$  is a vector concerning a set of linearly independent basis functions  $\Psi_j(x) : \mathbb{R}^n \to \mathbb{R}(j = 1, 2, \dots, N_c), C_i^k, i \in V$  are the weight vectors and  $e_{\Psi_i}^k(x)$  is the approximation error.  $N_c$  is the number of the basis functions.

With the approximation functions (11) and (16) applied, Equation (10) can be transformed into

$$W^{k}(\Phi(x(t+\delta t)) - \Phi(x(t))) + \int_{t}^{t+\delta t} (C^{k}_{v^{k}(x)} - C^{k}_{v^{0}(x)})\Psi(x)d\tau = -\int_{t}^{t+\delta t} Q(x)d\tau + e^{k}_{2}(t), \quad (17)$$

where  $e_2^k(t) = \int_t^{t+\delta t} (e_{\Psi,v^0(x)}^k(x) - e_{\Psi,v^k(x)}^k(x))d\tau + e_{\Psi}^k(x(t)) - e_{\Psi}^k(x(t+\delta t))$  is the approximation error. Since  $\delta t$  is very small, the values of  $v^k(x)$  and  $v^0(x)$  can be seen to be constant during the time interval  $[t, t + \delta t)$  so that  $C_{v^0(x)}^k$  and  $C_{v^k(x)}^k$  are constant during the time interval  $[t, t + \delta t)$ . Therefore, the estimation can be made as (18):

$$\hat{W}^{k}(\Phi(x(t+\delta t)) - \Phi(x(t))) + (\hat{C}^{k}_{\hat{\sigma}^{k}(x)} - \hat{C}^{k}_{v^{0}(x)}) \int_{t}^{t+\delta t} \Psi(x)d\tau = -\int_{t}^{t+\delta t} Q(x)d\tau.$$
(18)

For the subsequent algorithm, in addition to the data matrices defined in Section 3.1, some more matrices are required which are defined as follows:

$$g(t_r) = \int_{t_r}^{t_r + \delta t} \Psi(x(\tau)) d\tau$$

where  $t_1 < t_2 < \cdots < t_l$ , *l* is a positive integer,  $g(t_r) \in \mathbb{R}^{N_c}$  and  $r = 1, 2, \ldots l$ .

At first, the following assumption concerning the data matrices is made as Assumption 2:

**Assumption 3.** There exist a positive integer  $\overline{L}'$  and a positive number  $\alpha$  such that for all  $L \ge \overline{L}'$ , the following equalities hold:

$$\frac{1}{L}\sum_{r=1}^{L}\bar{\Phi}(t_r)\bar{\Phi}(t_r)^T \ge \alpha I_{N_w}$$

where the time instants  $t_r$  satisfies the condition that  $\hat{v}^k(x(t_r)) = v^0(x(t_r));$ 

$$\frac{1}{L}\sum_{r=1}^{L}\Psi(x(t_r))\Psi(x(t_r))^T \ge \alpha I_{N_d}$$

where the time instants  $t_r$  satisfies the condition that  $v^0(x(t_r)) = i$  for  $\forall i \in V$ ;

$$\frac{1}{L}\sum_{1}^{L}g(t_r)g(t_r)^T \ge \alpha I_{N_c}$$

where the time instants  $t_r$  satisfies the conditions that  $\hat{v}^k(x(t_r)) = i$  and  $v^0(x(t_r)) = j$  for  $\forall i \neq j, (i, j \in V)$ .

**Remark 1.** Since the vector  $\hat{C}_{\vartheta^k(x)}^k - \hat{C}_{\vartheta^0(x)}^k$  changes as x changes, the weight vector  $\hat{W}^k$  and  $\hat{C}_{\vartheta^k(x)}^k$  cannot be solved directly from (18) with the data matrices as Section 3.1. The difficulty in this problem is mainly calculating the weight vector  $\hat{W}^k$  and  $\hat{C}_{\vartheta^k(x)}^k$  or achieve enough useful knowledge about the weight vectors through the state data.

Based on the above analysis, the following approach is designed to acquire useful knowledge about the weight vector  $\hat{W}^k$  and  $\hat{C}^k_{\hat{v}^k(x)}$  step by step through different state data. The weight vector  $W^k$  is discussed firstly. The state data satisfying the condition that  $\hat{v}^k(x) = v^0(x)$  is selected from all the state data and then utilized in (18). It is obvious that when  $\hat{v}^k(x) = v^0(x)$ ,  $\hat{C}^k_{\hat{v}^k(x)} = \hat{C}^k_{v^0(x)}$  and (18) can be simplified to:

$$\hat{W}^k(\Phi(x(t+\delta t)) - \Phi(x(t))) = -\int_t^{t+\delta t} Q(x)d\tau.$$
(19)

Under Assumption 3, the estimate of the weight vector  $W^k$  can be achieved from (19) with the data matrices defined in Section 3.1 concerning certain states as follows:

$$\hat{W}^{k} = -\left(\sum_{r=1}^{L} d(t_{r}) \bar{\Phi}(t_{r})^{T}\right) \left(\sum_{r=1}^{L} \bar{\Phi}(t_{r}) \bar{\Phi}(t_{r})^{T}\right)^{-1},$$
(20)

where the time instants  $t_r$  satisfies the condition that  $\hat{v}^k(x(t_r)) = v^0(x(t_r))$ .

Then, we concentrate on estimating the weight vector of  $C_i^k$ . Along the acquired state trajectory x(t) produced by the switching policy  $v^0$ , it can be obtained from (16) and (7) that when k = 0 the following formula holds:

$$\int_{t}^{t+\delta t} C^{0}_{v^{0}(x)} \Psi(x) d\tau + \int_{t}^{t+\delta t} e^{0}_{\Psi,v^{0}(x)}(x) d\tau + \int_{t}^{t+\delta t} Q(x) d\tau = 0.$$
(21)

The estimation can be made as follows:

$$\hat{C}^0_{v^0(x)}\int_t^{t+\delta t}\Psi(x)d au+\int_t^{t+\delta t}Q(x)d au=0.$$

Under Assumption 3, the estimate of the weight vector  $C_i^0$  can be easily achieved as follows:

$$\hat{C}_{i}^{0} = -\left[\sum_{1}^{L} d(t_{r})g(t_{r})^{T}\right]\left(\sum_{1}^{L} g(t_{r})g(t_{r})^{T}\right)^{-1},$$
(22)

where the time instants  $t_r$  satisfies the condition that  $v^0(x(t_r)) = i(i \in V)$ .

Though the weight vector  $C_i^k$  is not easy to estimate, the estimate of  $C_i^k - C_j^k$  ( $i \neq j$ ) can be easily achieved from (18) with the data matrices concerning certain states under Assumption 3 as follows:

$$\hat{C}_{i}^{k} - \hat{C}_{j}^{k} = -\left[\sum_{1}^{L} \left(d(t_{r}) + \hat{W}^{k}\bar{\Phi}(t_{r})\right)g(t_{r})^{T}\right]\left(\sum_{1}^{L} g(t_{r})g(t_{r})^{T}\right)^{-1},\tag{23}$$

where the time instants  $t_r$  satisfies the conditions that  $\hat{v}^k(x(t_r)) = i$  and  $v^0(x(t_r)) = j$   $(i, j \in V$  and  $i \neq j$ ).

For a certain state x,  $\left(\frac{\partial V^k(x)}{\partial x}\right)^T f_{v^0(x)}(x)$  is constant and  $\arg\min_{i \in V} \left(\left(\frac{\partial V^k(x)}{\partial x}\right)^T f_i(x)\right) = \arg\min_{i \in V} \left(\left(\frac{\partial V^k(x)}{\partial x}\right)^T f_i(x) - \left(\frac{\partial V^k(x)}{\partial x}\right)^T f_{v^0(x)}(x)\right)$ . Therefore, on the basis of (8), employing the estimate of (22) and (23), the switching policy  $v^{k+1}(x)$  is estimated by

$$\hat{v}^{k+1}(x) = \begin{cases} \arg\min(\hat{C}_i^0 \Psi(x)) & k = 0\\ \underset{i \in V}{i \in V} \\ \arg\min(\hat{C}_i^k - \hat{C}_{v^0(x)}^k) \Psi(x) & k > 0 \end{cases}$$
(24)

with the initial switching policy estimate  $\hat{v}^0(x) = v^0(x)$ .

The approach utilizes different parts of the acquired state data to calculate the weight vector  $\hat{W}^k$ ,  $C_i^0$  and  $C_i^k - C_j^k$  ( $i \neq j$ ) respectively. Then the weight vectors are employed to calculate the switching policy and the cost function.

According to the aforementioned analysis, the data-driven PI-based algorithm for switched systems can be formulated as follows:

Then, the approximate optimal cost function can be achieved as  $\hat{V}^*(x) = \hat{W}^*\Phi(x)$  with the corresponding approximate optimal switching policy  $\hat{v}^*(x) = \arg\min_{i \in V} ((\hat{C}_i^k - \hat{C}_{v^0(x)}^k) \Psi(x)).$ 

**Remark 2.** In Algorithm 2, only the initial switching policy  $v^0(x)$  requires to be applied in the system and the produced state data is collected at the beginning. The data matrices  $\overline{\Phi}(t_r)$ ,  $d(t_r)$  and  $g(t_r)$  are calculated once at the beginning and don't require to be calculated repeatedly at each iteration. It is very convenient and timesaving to operate online according to Algorithm 2 and the calculation time is very short. Therefore, the optimal cost can be approximated rapidly in Algorithm 2. Moreover, Algorithm 2 is only based on data with no need for the knowledge of subsystems.

Algorithm 2 Data-driven PI-based algorithm.

Step 1. Start with an initial admissible switching policy  $v^0(x)$  and set the iteration index k = 0. Step 2. Apply  $v^0(x)$  in the switched systems and acquire the state data. Set  $\hat{v}^0(x) = v^0(x)$ . Calculate  $\bar{\Phi}(t_r)$ ,  $d(t_r)$  and  $g(t_r)$  for r = 1, 2, ..., l according to their definition with the state data. Step 3. Calculate  $\hat{C}_i^0$  for all  $i \in V$  from (22) and then calculate  $\hat{v}^1(x)$  from (24). Set k = 1. Step 4. Calculate  $\hat{W}^k$  from (20) and then calculate  $\hat{C}_i^k - \hat{C}_j^k$  from (23) for all  $i \neq j$  ( $i, j \in V$ ). Step 5. Update the switching policy  $\hat{v}^{k+1}(x)$  as (24). Step 6. If  $\|\hat{W}^k - \hat{W}^{k-1}\| < \varepsilon$ , set  $\hat{W}^* = \hat{W}^k$  and exit. Otherwise, set k = k + 1 and go back to Step 4.

Next, the convergence of Algorithm 2 is analyzed. Based on Theorem 1, Theorem 2 is stated for the convergence analysis of Algorithm 2.

**Theorem 2.** For system (1) with cost (2) under Assumptions 1 and 3, if the value function sequence  $\{V^k\}_{k=0}^{\infty}$  and the switching policy sequence  $\{v^k\}_{k=0}^{\infty}$  are generated through (7) and (8) initiating from an initial admissible

policy  $v^0$ , for  $\forall \varepsilon > 0$  and all  $x \in \Omega$ , there exists  $\overline{N} > 0$  such that for  $\forall N > \overline{N}$ , when  $\delta t$  approaches to zero, the following inequalities hold:

$$\left|\hat{V}^{k}(x)-V^{k}(x)\right|$$

where  $N = \min\{N_w, N_c\}$ ,  $\hat{V}^k(x) = \hat{W}^k \Phi(x)$ ,  $\hat{W}^k$  and  $\hat{v}^k$  are generated through (20) and (24) in Algorithm 2, and k = 0, 1, ...

# Proof of Theorem 2. Mathematical induction is utilized to prove the convergence.

Firstly, we discuss the theorem when k = 0. When k = 0,  $\hat{v}^0(x) = v^0(x)$ ,  $\hat{C}^k_{\hat{v}^0(x)} = \hat{C}^k_{v^0(x)}$  and it can be achieved from (17) and (18) that:

$$(\hat{W}^0 - W^0)(\Phi(x(t+\delta t)) - \Phi(x(t))) = -e_2^0(t)$$

and

$$\sum_{r=1}^{L} (\hat{W}^{0} - W^{0}) \bar{\Phi}(t_{r}) (\bar{\Phi}(t_{r}))^{T} (\hat{W}^{0} - W^{0})^{T} = \sum_{r=1}^{L} e_{2}^{0} (t_{r})^{2}$$

According to Assumption 3, it follows that

$$\frac{1}{L}\sum_{r=1}^{L}\bar{\Phi}(t_r)(\bar{\Phi}(t_r))^T \ge \alpha I_{N_w}$$

and then  $\|\hat{W}^0 - W^0\|^2 \leq \frac{1}{\alpha L} \sum_{r=1}^{L} e_2^0(t_r)^2$ . The approximation theory [42] yields that for all  $x \in \Omega$ ,  $\lim_{N_w \to \infty} e_{\Phi}^0(x) = 0$ , then  $\lim_{N_w \to \infty} e_2^0(t_r) = 0$ . Therefore, for  $\forall \varepsilon > 0$  and  $\forall x \in \Omega$ , there exists  $\bar{N}_w > 0$  such that for  $\forall N_w > \bar{N}_w$ ,  $\|\hat{W}^0 - W^0\|^2 \leq \varepsilon$  and then  $|\hat{V}^0(x) - V^0(x)| < \varepsilon$ . It follows that  $\lim_{N_w \to \infty} \|\hat{W}^0 - W^0\| = 0$ and  $\lim_{N_w \to \infty} \hat{V}^0(x) = V^0(x)$  for  $\forall x \in \Omega$ . Then, it can be achieved from (21) and (22) that:

$$(\hat{C}^{0}_{v^{0}(x)} - C^{0}_{v^{0}(x)}) \int_{t}^{t+\delta t} \Psi(x) d\tau = \int_{t}^{t+\delta t} e^{0}_{\Psi,v^{0}(x)}(x) d\tau + \int_{t}^{t+\delta t} (C^{0}_{v^{0}(x(\tau))} - C^{0}_{v^{0}(x(t))}) \Psi(x) d\tau.$$

Similarly, the approximation theory [42] yields that for all  $x \in \Omega$ ,  $\lim_{N_w \to \infty} e^0_{\Psi,i}(x) = 0 (i \in V)$  and then  $\lim_{\delta t \to 0} \int_t^{t+\delta t} (C^0_{v^0(x(\tau))} - C^0_{v^0(x(t))}) \Psi(x) d\tau = 0$ . Under Assumption 3, it can be deduced that for  $\forall x \in \Omega$ ,  $\lim_{\substack{N_c \to \infty \\ \delta t \to 0}} (\frac{\partial V^0(x)}{\partial x})^T f_i(x) = (\frac{\partial V^0(x)}{\partial x})^T f_i(x)$ . It implies that  $\lim_{\substack{N_c \to \infty \\ \delta t \to 0}} \hat{v}^1(x) = v^1(x)$  for  $\forall x \in \Omega$ .

Secondly, we consider the theorem when k = 1. When k = 1,  $\lim_{N_c \to \infty} \hat{C}^1_{\hat{v}^1(x)} = \hat{C}^1_{v^1(x)}$ . When  $\hat{W}^1$  is calculated, for  $\forall x \in \Omega$  and it can be achieved from (17) to (19) that:

$$\begin{aligned} (\hat{W}^1 - W^1)(\Phi(x(t+\delta t)) - \Phi(x(t))) + (\hat{C}^1_{\hat{v}^1(x)} - \hat{C}^1_{v^0(x)}) \int_t^{t+\delta t} \Psi(x) d\tau - \\ \int_t^{t+\delta t} (C^1_{v^1(x)} - C^1_{v^0(x)}) \Psi(x) d\tau &= -e_2^1(t), \end{aligned}$$

where  $\hat{v}^1(x) = v^0(x)$  and thus  $\lim_{N_c \to \infty} v^1(x) = v^0(x)$  for  $\forall x \in \Omega$ . Then it can be inferred that when  $N_c \to \infty$ 

$$(\hat{W}^1 - W^1)(\Phi(x(t+\delta t)) - \Phi(x(t))) = -e_2^1(t)$$

Under Assumption 3, it can be deduced that for all  $x \in \Omega$ ,  $\lim_{N_w \to \infty} e_2^1(t_r) = 0$ ,  $\lim_{N_w,N_c\to\infty} \|\hat{W}^1 - W^1\| = 0 \text{ and } \lim_{N_w,N_c\to\infty} \hat{V}^1(x) = V^1(x). \text{ Then, it can be achieved from (17), (18) and (23) that:}$ 

$$\begin{aligned} & (\hat{C}^{1}_{\hat{v}^{1}(x)} - \hat{C}^{1}_{v^{0}(x)}) \int_{t}^{t+\delta t} \Psi(x) d\tau - \int_{t}^{t+\delta t} (C^{1}_{v^{1}(x)} - C^{1}_{v^{0}(x)}) \Psi(x) d\tau \\ &= -e^{1}_{2}(t) - (\hat{W}^{1} - W^{1}) (\Phi(x(t+\delta t)) - \Phi(x(t))). \end{aligned}$$

Since it can be achieved that  $\int_{t}^{t+\delta t} (C_{v^{1}(x)}^{1} - C_{v^{0}(x)}^{1}) \Psi(x) d\tau = (C_{v^{1}(x)}^{1} - C_{v^{0}(x)}^{1}) \int_{t}^{t+\delta t} \Psi(x) d\tau + e_{3}^{1}(t)$  where  $e_{3}^{1}(t) = \int_{t}^{t+\delta t} ((C_{v^{1}(x(\tau))}^{1} - C_{v^{1}(x(t))}^{1}) - (C_{v^{0}(x(\tau))}^{1} - C_{v^{0}(x(t))}^{1}) \Psi(x) d\tau$ , it can be inferred that

$$\begin{bmatrix} (\hat{C}^{1}_{\vartheta^{1}(x)} - \hat{C}^{1}_{\vartheta^{0}(x)}) - (C^{1}_{\vartheta^{1}(x)} - C^{1}_{\vartheta^{0}(x)}) \end{bmatrix} \int_{t}^{t+\delta t} \Psi(x) d\tau \\ = e^{1}_{3}(t) - e^{1}_{2}(t) - (\hat{W}^{1} - W^{1}) (\Phi(x(t+\delta t)) - \Phi(x(t))).$$

Since  $\lim_{\delta t \to 0} e_3^1(t) = 0$ ,  $\lim_{N_w \to \infty} e_2^1(t) = 0$  and  $\lim_{N_w, N_c \to \infty} \|\hat{W}^1 - W^1\| = 0$ , under Assumption 3, it can be deduced that  $\lim_{\substack{N_w, N_c \to \infty \\ \delta t \to 0}} \|(\hat{C}_{\hat{v}^1(x)}^1 - \hat{C}_{v^0(x)}^1) - (C_{v^1(x)}^1 - C_{v^0(x)}^1)\| = 0$  and then  $\lim_{\substack{N_w, N_c \to \infty \\ \delta t \to 0}} (\hat{C}_{\hat{v}^1(x)}^1 - \hat{C}_{v^0(x)}^1) \Psi(x)$  for  $\forall x \in \Omega$ . It implies that  $\lim_{\substack{N_w, N_c \to \infty \\ \delta t \to 0}} \hat{v}^2(x) = v^2(x)$  for  $\forall x \in \Omega$ .

Thirdly, we suppose the theorem holds when k = k. That is to say,  $\lim_{\substack{N_w, N_c \to \infty \\ \delta t \to 0}} \hat{V}^k(x) = V^k(x)$  and  $\lim_{\substack{N_w,N_c\to\infty\\\delta t\to 0}} \hat{v}^{k+1}(x) = v^{k+1}(x) \text{ for } \forall x \in \Omega. \text{ When } \hat{W}^{k+1} \text{ is calculated, it can be achieved from (17) to}$ 

(19) that:

$$(\hat{W}^{k+1} - W^{k+1})(\Phi(x(t+\delta t)) - \Phi(x(t))) + (\hat{C}^{k+1}_{\vartheta^{k+1}(x)} - \hat{C}^{k+1}_{v^0(x)}) \int_t^{t+\delta t} \Psi(x)d\tau - \int_t^{t+\delta t} (C^{k+1}_{v^{k+1}(x)} - C^{k+1}_{v^0(x)})\Psi(x)d\tau = -e_2^{k+1}(t)$$

where  $\hat{v}^{k+1}(x) = v^0(x)$  and  $\lim_{\substack{N_w, N_c \to \infty \\ \delta t \to 0}} v^{k+1}(x) = v^0(x)$  for  $\forall x \in \Omega$ . Then it can be inferred that  $(\hat{W}^{k+1} - W^{k+1})(\Phi(x(t+\delta t)) - \Phi(x(t))) = -e_2^{k+1}(t)$ 

Under Assumption 3, it can be deduced that  $\lim_{\substack{N_w, N_c \to \infty \\ \delta t \to 0}} \left\| \hat{W}^{k+1} - W^{k+1} \right\| = 0$  and  $\lim_{\substack{N_w, N_c \to \infty \\ \delta t \to 0}} \hat{V}^{k+1}(x) = V^{k+1}(x)$  for  $\forall x \in \Omega$ . Then, it can be achieved from (17), (18) and (23) that:

$$\begin{split} & [(\hat{C}_{\vartheta^{k+1}(x)}^{k+1} - \hat{C}_{v^0(x)}^{k+1}) - (C_{v^{k+1}(x)}^{k+1} - C_{v^0(x)}^{k+1})] \int_t^{t+\delta t} \Psi(x) d\tau \\ & = -(\hat{W}^{k+1} - W^{k+1}) (\Phi(x(t+\delta t)) - \Phi(x(t))) - e_2^{k+1}(t) + e_3^{k+1}(t)) \end{split}$$

where  $e_3^{k+1}(t) = \int_t^{t+\delta t} \left( (C_{v^{k+1}(x(\tau))}^{k+1} - C_{v^{k+1}(x(t))}^{k+1}) - (C_{v^0(x(\tau))}^{k+1} - C_{v^0(x(t))}^{k+1}) \Psi(x) d\tau \right)$ . Since  $\lim_{\delta t \to 0} e_3^{k+1}(t) = 0$ ,  $\lim_{N_w, N_c \to \infty} e_2^{k+1}(t) = 0$  and  $\lim_{N_w, N_c \to \infty} \left\| \hat{W}^{k+1} - W^{k+1} \right\| = 0$ , under Assumption 3, it can be deduced that  $\lim_{\substack{N_{w},N_{c}\to\infty\\\delta t\to0}\\(C_{v^{k+1}(x)}^{k+1}-C_{v^{0}(x)}^{k+1})-(C_{v^{k+1}(x)}^{k+1}-C_{v^{0}(x)}^{k+1})\|=0 \text{ and then } \lim_{\substack{N_{w},N_{c}\to\infty\\\delta t\to0}}(\hat{C}_{v^{k+1}(x)}^{k+1}-C_{v^{0}(x)}^{k+1})\Psi(x) = 0 \text{ and then } \lim_{\substack{N_{w},N_{c}\to\infty\\\delta t\to0}}\hat{C}_{v^{k+1}(x)}^{k+1}-\hat{C}_{v^{0}(x)}^{k+1})\Psi(x) = 0 \text{ and then } \lim_{\substack{N_{w},N_{c}\to\infty\\\delta t\to0}}\hat{C}_{v^{k+1}(x)}^{k+1}-\hat{C}_{v^{k$ can be deduced that the theorem holds when k = k + 1 from the supposition that the theorem holds when k = k.

The proof is completed through mathematical induction.  $\Box$ 

It can be achieved from Theorem 2 that the value function  $\hat{V}^k(x)$  generated through Algorithm 2 is an approximation of  $V^k(x)$  and the corresponding switching policy is  $\hat{v}^k(x) = v^k(x)$  if the preconditions are satisfied. Theorem 2 combined with Theorem 1 indicates that the value function  $\hat{V}^*(x) = \hat{W}^*\Phi(x)$  is the approximate optimal cost function with the corresponding approximate optimal switching policy  $\hat{v}^*(x) = \arg\min_{i \in V} ((\hat{C}_i^k - \hat{C}_{v^0(x)}^k))\Psi(x)).$ 

**Remark 3.** In practice, the error of the approximation exists and the small parameter  $\delta t$  can not get infinitely close to zero so that the solution calculated from Algorithm 2 is suboptimal.

### 4. Example

In this section, two examples are illustrated to validate the effectiveness of the suboptimal scheduling approach of Algorithm 1 and the data-driven suboptimal scheduling approach of Algorithm 2 in this paper.

**Example 1.** Consider a switched system as [21,38] consisting of the following subsystems:

$$\dot{x}(t) = f_1(x(t)) = -x(t), \dot{x}(t) = f_2(x(t)) = -x^3(t)$$

with x(0) = 2 and  $Q(x(t)) = x^2(t)$ .

The optimal switching policy can be known from [17] as

$$v^*(x) = \begin{cases} 1 & x \le 1\\ 2 & x > 1 \end{cases}$$

Choose the initial switching policy  $v^0(x) = 1$  when  $x \le 1.5$  and  $v^0(x) = 2$  when x > 1.5. The basis functions are  $\Phi(x) = [x^2, x^4, x^6, x^8, x^{10}]^T$  selected as [21]. Set the sample period  $\delta t = 0.002$  s.

Apply Algorithm 1 and utilize the online state data from t = 0 to 0.2 s. Then after calculation in 0.02 s, the approximate optimal cost is achieved with the corresponding approximate optimal switching policy through 3 iterations. The initial cost  $\hat{V}^0$ , the approximate optimal cost  $\hat{V}^k$  and the optimal cost  $V^*$  are demonstrated in Figure 1. The largest error between  $\hat{V}^k$  and  $V^*$  is 0.1621 and it is obvious that the approximate optimal cost  $\hat{V}^k$  is very close to the optimal cost of  $V^*$ . The state trajectories with the initial switching policy  $v^0$ , the approximate optimal switching policy  $\vartheta^k$  and the optimal switching policy  $v^*$  applied in the system after t = 0.25 s are demonstrated in Figure 2, where the state trajectory corresponding to  $\vartheta^k$  coincides with the one corresponding to  $v^*$  and their largest error is zero while the largest error between the state trajectory corresponding to  $v^0$  and the one corresponding to  $v^*$  is 0.0563. The online trajectories are too close to show the superiority of the proposed algorithm so that the switching policies  $v^0$ ,  $\vartheta^k$  and  $v^*$  when  $x \in [-2, 2]$  are illustrated in Figure 3. Apparently,  $\vartheta^k$  and  $v^*$  are the same. The similarity rate of the schedules  $\vartheta^k$  and  $v^*$  is 100% while the similarity rate of the schedules  $\vartheta^0$  and  $v^*$  is 75.31%.

Apply Algorithm 2 and utilize the online state data from t = 0 to 0.2 s. Then after calculation in 0.05 s, the approximate optimal cost is achieved with the corresponding approximate optimal switching policy through 6 iterations. The costs  $\hat{V}^0$ ,  $\hat{V}^k$  and  $V^*$  are demonstrated in Figure 4. It can be obtained that compared with the initial cost of  $\hat{V}^0$ , the approximate optimal cost of  $\hat{V}^k$  is close to the optimal cost of  $V^*$ . The state trajectories with  $v^0$ ,  $\hat{v}^k$  and  $v^*$  applied in the system after t = 0.25 s are demonstrated in Figure 5, where the state trajectory corresponding to  $\hat{v}^k$  coincides with the one corresponding to  $v^*$  and their largest error is 0.0155 while the largest error between the state trajectory corresponding to  $v^0$  and the one corresponding to  $v^*$  is 0.0563. The online trajectories also are too close to show the superiority of the proposed algorithm so that the switching policies  $v^0$ ,  $\hat{v}^k$  and  $v^*$  when  $x \in [-2, 2]$  are illustrated in Figure 6. It can be seen that  $\hat{v}^k$  is approximate to  $v^*$ . The similarity rate of the schedules  $\hat{v}^k$  and  $v^*$  is 95.06% while the similarity rate of the schedules  $v^0$  and  $v^*$  is 75.31%.



Figure 1. Costs achieved by Algorithm 1 applied in Example 1.



Figure 2. State trajectories with Algorithm 1 applied in Example 1.



Figure 3. Switching policies achieved by Algorithm 1 applied in Example 1.



Figure 4. Costs achieved by Algorithm 2 applied in Example 1.



Figure 5. State trajectories with Algorithm 2 applied in Example 1.



Figure 6. Switching policies achieved by Algorithm 2 applied in Example 1.

**Example 2.** Consider a mass-spring-damper system as [21,22]:

$$\dot{x}_1(t) = x_2(t),$$
  
 $\dot{x}_2(t) = F_v - 0.1x_1(t) - 0.1x_2(t)$ 

with  $v \in \{1, 2, 3\}$ ,  $F_1 = 1$ ,  $F_2 = -1$  and  $F_3 = 0$ . Here, the state  $x_1(t)$  is the displacement of the mass measured from the relaxed length of the spring.  $F_v$  is the external force acting on the mass. The initial state is x(0) = [2, 2] and the function Q is

Choose the initial switching policy  $v^0(x) = 3$  when  $|x_1| \le 0.5$ ,  $v^0(x) = 2$  when  $x_1 > 0.5$  and  $v^0(x) = 1$  when  $x_1 < -0.5$ . The basis functions are polynomials with all possible combinations of the state variables up to the 4th degree without repetitions selected as [21,22]. Set the sample period  $\delta t = 0.02$  s.

Apply Algorithm 1 and utilize the online state data from t = 0 to 23 s. Then after calculation in 7 s, the approximate optimal cost is achieved with the corresponding approximate optimal switching policy through 15 iterations. The initial cost  $\hat{V}^0$  and the approximate optimal cost  $\hat{V}^k$  are demonstrated in Figure 7. It can be seen that the approximate optimal cost  $\hat{V}^k$  is less than the initial cost  $\hat{V}^0$ . The state trajectories with the initial switching policy  $v^0$  and the approximate optimal switching policy  $\hat{v}^k$  applied in the system after t = 30 s are demonstrated in Figure 8, where the state trajectory corresponding to  $\hat{v}^k$  converges to the origin quickly after t = 30 s while the trajectory corresponding to  $v^0$  converges slowly with decreasing oscillation amplitude. The corresponding switching policies  $v^0$  and  $\hat{v}^k$  are illustrated in Figure 9. It can be seen that  $v^0$  which switches among three subsystems and finally stays at subsystem 3 results in that the trajectory corresponding to  $v^0$  converges slowly with decreasing oscillation applitude, while  $v^*$  which switches between subsystem 1 and 2 results in that the state trajectory corresponding to  $\hat{v}^k$  converges to the origin quickly after t = 30 s.



Figure 7. Costs achieved by Algorithm 1 applied in Example 2.



Figure 8. State trajectories with Algorithm 1 applied in Example 2.

Apply Algorithm 2 and utilize the online state data from t = 0 to 23 s. Then after calculation in 7 s, the approximate optimal cost is achieved with the corresponding approximate optimal switching policy through 21 iterations. The costs  $\hat{V}^0$  and  $\hat{V}^k$  are demonstrated in Figure 10. It can be seen that the approximate optimal cost  $\hat{V}^k$  is less than the initial cost  $\hat{V}^0$ . The state trajectories with  $v^0$  and  $\hat{v}^k$  applied in the system after t = 30 s are demonstrated in Figure 11, where the state trajectory corresponding to  $\hat{v}^k$  converges to the origin relatively quickly after t = 30 s while the trajectory corresponding to  $v^0$  converges slowly with decreasing oscillation amplitude. The corresponding switching policies  $v^0$  and  $\hat{v}^k$  are illustrated in Figure 12. It can be seen that  $v^0$  which switches among three subsystems and finally stays at subsystem three results in that the trajectory converges slowly with decreasing oscillation amplitude, while  $v^*$  which switches between subsystem 1 and 2 results in that the state trajectory converges to the origin relatively quickly after t = 30 s.

**Remark 4.** In algorithm 2, the value of the initial switching policy  $v^0(x)$  should include every element of V and there should exist enough data to calculate  $\hat{C}_i^0$  for every  $i \in V$  so that the sampling stage must be long enough to ensure that.

**Remark 5.** In the calculation, the fourth order Runge-Kutta algorithm is adopted to numerically evaluate the integrals which are necessary in certain terms such as  $b(t_r)$  and  $d(t_r)$ .

**Remark 6.** In Example 1, the proposed algorithms converge in 0.25 s while the online algorithm investigated in [38] requires more time. In Example 2, the proposed algorithms converge in 30 s while the online algorithms investigated in [21,22] require more time.



Figure 9. Switching policies with Algorithm 1 applied in Example 2.



Figure 10. Costs achieved by Algorithm 2 applied in Example 2.



Figure 11. State trajectories with Algorithm 2 applied in Example 2.



Figure 12. Switching policies with Algorithm 2 applied in Example 2.

In the two examples, the effectiveness of Algorithm 2 has been validated. The superiority of Algorithm 2 lies in that it can work for switched systems with unknown subsystems while Algorithm 1 can not work if the subsystems of switched systems are unknown. Practical examples of this kind of switched systems appear in a wide range of applications such as cyber-physical systems which embed software into the physical world and have proved resistant to modeling due to their intrinsic complexity arising from the combination of physical and cyber components and the interaction between them in [31]. Thereinto, data-driven research has been conducted for some specific examples such as complex electronics switching among low-voltage, middle-voltage and high-voltage models, and smart grid switching between base configuration and changed configuration. These examples require the data-driven approaches of Algorithm 2 where Algorithm 1 is inapplicable.

Simulation results show that Algorithms 1 and 2 can approximate the optimal solution effectively and efficiently. Algorithm 2 achieves the approximate optimal solution for unknown switched systems with infinite-horizon cost function, which has not been achieved well in existing literature as far as we know.

#### 5. Conclusions

In this paper, an online PI-based algorithm inspired by the off-policy learning method and based on that, a data-driven PI-based algorithm, are proposed to approximate the optimal solution quickly for optimal scheduling problems. Only data produced by an initial switching policy is necessary and the approximation time is relatively short. The data-driven algorithm acquires useful knowledge of the weight vectors step by step through different data and solves the optimal scheduling

problem for switched systems with unknown subsystems, only taking advantage of data. However, the dwell-time problems, which are important in practical applications, are not incorporated in this paper. So, optimal scheduling problems with dwell-time constraints will be investigated in the future.

**Author Contributions:** Methodology, C.Z. and M.G.; formal analysis, C.Z. and J.Z.; validation, C.Z. and C.X.; writing—original draft preparation, C.Z.; supervision, M.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China under grant number 61673065.

Conflicts of Interest: The authors declare no conflict of interest.

#### References

- 1. Liberzon, D. Switching in Systems and Control; Springer Science & Business Media: Boston, MA, USA, 2003.
- 2. Sun, Z. Switched Linear Systems: Control and Design; Springer Science & Business Media: London, UK, 2006.
- Zha, W.; Guo, Y.; Wu, H.; Sotelo, M.A.; Ma, Y.; Yi, Q.; Li, Z.; Sun, X. A new switched state jump observer for traffic density estimation in expressways based on hybrid-dynamic-traffic-network-model. *Sensors* 2019, 19, 3822. [CrossRef] [PubMed]
- Liu, Y.; Zhao, Y.; Zhu, J.; Wang, J.; Tang, B. A Switched-Element System Based Direction of Arrival (DOA) Estimation Method for Un-Cooperative Wideband Orthogonal Frequency Division Multi Linear Frequency Modulation (OFDM-LFM) Radar Signals. *Sensors* 2019, *19*, 132. [CrossRef] [PubMed]
- Egerstedt, M.; Wardi, Y.; Axelsson, H. Transition-time optimization for switched-mode dynamical systems. *IEEE Trans. Autom. Control* 2006, *51*, 110–115. [CrossRef]
- 6. Axelsson, H.; Wardi, Y.; Egerstedt, M.; Verriest, E.I. Gradient Descent Approach to Optimal Mode Scheduling in Hybrid Dynamical Systems. *J. Optim. Theory Appl.* **2008**, *136*, 167–186. [CrossRef]
- 7. Koksal, N.; Jalalmaab, M.; Fidan, B. Adaptive linear quadratic attitude tracking control of a quadrotor UAV based on IMU sensor data fusion. *Sensors* **2019**, *19*, 46. [CrossRef]
- 8. Aranda-Escolástico, E.; Salt, J.; Guinaldo, M.; Chacón, J.; Dormido, S. Optimal control for aperiodic dual-rate systems with time-varying delays. *Sensors* **2018**, *18*, 1491. [CrossRef]
- 9. Stellato, B.; Ober-Blobaum, S.; Goulart, P.J. Second-Order Switching Time Optimization for Switched Dynamical Systems. *IEEE Trans. Autom. Control* 2017, *62*, 5407–5414. [CrossRef]
- 10. Wardi, Y.; Egerstedt, M.; Hale, M. Switched-mode systems: Gradient-descent algorithms with Armijo step sizes. *Discret. Event Dyn. Syst.* **2015**, *25*, 571–599. [CrossRef]
- Ruffler, F.; Hante, F.M. Optimal switching for hybrid semilinear evolutions. *Nonlinear Anal. Hybrid Syst.* 2016, 22, 215–227. [CrossRef]
- 12. Das, T.; Mukherjee, R. Optimally switched linear systems. Automatica 2008, 44, 1437–1441. [CrossRef]
- 13. Zhang, W.; Hu, J.; Abate, A. Infinite-horizon switched LQR problems in discrete time: A suboptimal algorithm with performance analysis. *IEEE Trans. Autom. Control* **2012**, *57*, 1815–1821. [CrossRef]
- 14. Caldwell, T.M.; Murphey, T.D. Projection-based iterative mode scheduling for switched systems. *Nonlinear Anal. Hybrid Syst.* **2016**, *21*, 59–83. [CrossRef]
- 15. Li, C.; Liu, D.; Wang, D. Data-based optimal control for weakly coupled nonlinear systems using policy iteration. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2016**, *48*, 511–521. [CrossRef]
- 16. Heydari, A.; Balakrishnan, S.N. Optimal switching and control of nonlinear switching systems using approximate dynamic programming. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 1106–1117.
- Heydari, A.; Balakrishnan, S. Optimal switching between autonomous subsystems. J. Frankl. Inst. 2014, 351, 2675–2690. [CrossRef]
- Heydari, A. Optimal scheduling for reference tracking or state regulation using reinforcement learning. J. Frankl. Inst. 2015, 352, 3285–3303. [CrossRef]
- 19. Heydari, A. Feedback solution to optimal switching problems with switching cost. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 2009–2019. [CrossRef]
- 20. Heydari, A. Optimal switching of DC–DC power converters using approximate dynamic programming. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 586–596. [CrossRef]

- 21. Sardarmehni, T.; Heydari, A. Sub-optimal scheduling in switched systems with continuous-time dynamics: A gradient descent approach. *Neurocomputing* **2018**, *285*, 10–22. [CrossRef]
- 22. Sardarmehni, T.; Heydari, A. Suboptimal Scheduling in Switched Systems With Continuous-Time Dynamics: A Least Squares Approach. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 2167–2178. [CrossRef]
- 23. Wei, Q.; Liu, D. Data-driven neuro-optimal temperature control of water–gas shift reaction using stable iterative adaptive dynamic programming. *IEEE Trans. Ind. Electron.* **2014**, *61*, 6399–6408. [CrossRef]
- 24. Wei, Q.; Song, R.; Yan, P. Data-driven zero-sum neuro-optimal control for a class of continuous-time unknown nonlinear systems with disturbance using ADP. *IEEE Trans. Neural Netw. Learn. Syst.* 2015, 27, 444–458. [CrossRef] [PubMed]
- 25. Zhang, X.; Ma, H.; Zhang, X.; Li, Y. Compact Model-Free Adaptive Control Algorithm for Discrete-Time Nonlinear Systems. *IEEE Access* 2019, *7*, 141062–141071. [CrossRef]
- Lu, W.; Ferrari, S. An approximate dynamic programming approach for model-free control of switched systems. In Proceedings of the 52nd IEEE Conference on Decision and Control, Florence, Italy, 10–13 December 2013; pp. 3837–3844.
- 27. Lu, W.; Zhu, P.; Ferrari, S. A Hybrid-Adaptive Dynamic Programming Approach for the Model-Free Control of Nonlinear Switched Systems. *IEEE Trans. Autom. Control* 2016, *61*, 3203–3208. [CrossRef]
- 28. Zhang, C.; Gan, M.; Zhao, J. Data-driven optimal control of switched linear autonomous systems. *Int. J. Syst. Sci.* **2019**, *50*, 1275–1289. [CrossRef]
- 29. Gan, M.; Zhang, C.; Zhao, J. Data-driven optimal switching of switched systems. *J. Frankl. Inst.* 2019, 356, 5193–5221. [CrossRef]
- 30. Kulin, M.; Fortuna, C.; De Poorter, E.; Deschrijver, D.; Moerman, I. Data-driven design of intelligent wireless networks: An overview and tutorial. *Sensors* **2016**, *16*, 790. [CrossRef]
- 31. Yuan, Y.; Tang, X.; Zhou, W.; Pan, W.; Li, X.; Zhang, H.T.; Ding, H.; Goncalves, J. Data driven discovery of cyber physical systems. *Nat. Commun.* **2019**, *10*, 1–9. [CrossRef]
- 32. Wang, W.; Chen, X.; Fu, H.; Wu, M. Data-driven adaptive dynamic programming for partially observable nonzero-sum games via Q-learning method. *Int. J. Syst. Sci.* **2019**, *50*, 1338–1352. [CrossRef]
- 33. Liu, Y.; Zhang, H.; Yu, R.; Xing, Z. H∞ Tracking Control of Discrete-Time System with Delays via Data-Based Adaptive Dynamic Programming. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2019**. [CrossRef]
- 34. Radac, M.B.; Precup, R.E. Data-Driven model-free tracking reinforcement learning control with VRFT-based adaptive actor-critic. *Appl. Sci.* **2019**, *9*, 1807. [CrossRef]
- 35. Lagoudakis, M.G.; Parr, R. Least-squares policy iteration. J. Mach. Learn. Res. 2003, 4, 1107–1149.
- 36. Jiang, Y.; Jiang, Z.P. Robust Adaptive Dynamic Programming; John Wiley & Sons: Hoboken, NJ, USA, 2017.
- 37. Vrabie, D.; Pastravanu, O.; Abu-Khalaf, M.; Lewis, F.L. Adaptive optimal control for continuous-time linear systems based on policy iteration. *Automatica* **2009**, *45*, 477–484. [CrossRef]
- Sardarmehni, T.; Heydari, A. Policy iteration for optimal switching with continuous-time dynamics. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2016; pp. 3536–3543.
- 39. Jiang, Y.; Jiang, Z.P. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica* **2012**, *48*, 2699–2704. [CrossRef]
- 40. Bian, T.; Jiang, Y.; Jiang, Z.P. Adaptive dynamic programming and optimal control of nonlinear nonaffine systems. *Automatica* **2014**, *50*, 2624–2632. [CrossRef]
- Bian, T.; Jiang, Z.P. Value iteration, adaptive dynamic programming, and optimal control of nonlinear systems. In Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016; pp. 3375–3380.
- 42. Powell, M.J.D. Approximation Theory and Methods; Cambridge University Press: London, UK, 1981.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).