Short Papers

A Fuzzy-Model-Based Approach to Optimal Control for Nonlinear Markov Jump Singularly Perturbed Systems: A Novel Integral Reinforcement Learning Scheme

Hao Shen, *Member, IEEE*, Yun Wang, Jing Wang, and Ju H. Park, *Senior Member, IEEE*

***Abstract*—A fuzzy-model-based approach is developed to investigate the reinforcement learning-based optimization for nonlinear Markov jump singularly perturbed systems. As the first attempt, an offline parallel iteration learning algorithm is presented to solve the coupled algebraic Riccati equations with singular perturbation and jumping parameters. Furthermore, based on the integral reinforcement learning approach, a novel online parallel learning algorithm is proposed by employing the slow and fast sampled data simultaneously, where the impacts of stochastic jumping and ill-conditioned numerical problems are avoided. Meanwhile, the convergence of the proposed learning algorithms is proved. Finally, we present a tunnel diode circuit model to demonstrate the efficacy of the proposed methods.**

***Index Terms*—Fuzzy-model-based approach, Markov jump singularly perturbed systems (MJSPSs), parallel algorithm, reinforcement learning (RL).**

I. Introduction

A. Motivation

Over the past few decades, Markov jump systems (MJSs) have been applied in many fields as a special kind of stochastic systems, such as robust control [1], DC motors control [2], and traffic flow control [3]. The characteristic of MJSs lies in that they contain many subsystems, which obey the Markov process [4], [5]. Furthermore, the coexistence of fast and slow dynamics, named the two-time-scale phenomenon, is also widely met in the engineering field. However, this phenomenon usually brings the ill-conditioned problem for the controller design and performance analysis. Therefore, with the ability to study these two typical phenomena simultaneously, Markov jump singularly perturbed systems (MJSPSs) have earned a lot of attention [6], [7], [8], [9], [10], [11]. In [9], Wang et al. designed a hybrid H_∞ and dissipative asynchronous controller for MJSPSs. A sliding mode control (SMC) method for a class of nonlinear MJSPSs was presented in [10]. In [11], the H_∞ control problem for MJSPSs was discussed. It is worth noting that the abovementioned studies require the exact information of system dynamics, which is difficult to realize in real-world situations. To overcome the abovementioned limitation, the reinforcement learning (RL) method serves as a powerful tool to solve optimal control problems without system dynamics, which has rapidly attracted plenty of attention [12], [13], [14], [15]. Nevertheless, little attention has been paid to the learning control problem for MJSPSs, which is one of the motivations for this article.

B. Related Work

For RL-based control, Vrabie et al. [16] proposed an integral RL (IRL)-based policy iteration algorithm, which can obtain the optimal controller when the system dynamics information is partially unknown. After that, Jiang et al. [17] developed the algorithm proposed in [16], so that the optimal controller can be obtained with completely unknown system dynamics. For MJSs, it is difficult to design the optimal controller by employing system data due to the existence of Markov jumping parameters. Recently, several studies have been reported related to the learning controller design of MJSs. For example, He et al. [18] proposed a parallel algorithm to deal with the optimal control problem for MJSs with partially unknown dynamics. Based on the research in [18], the work in [19] settled the optimal control problem of completely unknown MJSs in view of the off-policy algorithm. Along with this line, Tu et al. [20] used a parallel control algorithm to solve the tracking control problem for MJSs. Besides, through seeking the Nash equilibrium for zero-sum game, the H_∞ control problem of MJSs was studied in [21]. Similarly, Xin et al. [22] investigated multiplayer nonzero sum games of MJSs. In another field, for singularly perturbed systems (SPSs), Zhao et al. [23] proposed a model-free-based algorithm to obtain the optimal controller, while the ill-conditioned problem caused by the singular perturbation parameter (SPP) was solved. Zhang et al. [24] employed the SMC technique to study the optimal control problem for SPSs when actuators fail.

Manuscript received 15 October 2022; revised 26 February 2023; accepted 3 April 2023. Date of publication 11 April 2023; date of current version 6 October 2023. The work of Ju H. Park was supported by the National Research Foundation of Korea (NRF), Korea Government (Ministry of Science and Information and Communications Technology) under Grant 2019R1A5A8080290. This work was supported in part by the National Natural Science Foundation of China under Grant 62273006, Grant 62173001, and Grant 61873002, in part by the Natural Science Foundation for Distinguished Young Scholars of Higher Education Institutions of Anhui Province under Grant 2022AH020034, in part by the Natural Science Foundation for Excellent Young Scholars of Higher Education Institutions of Anhui Province under Grant 2022AH030049, in part by the Major Natural Science Foundation of Higher Education Institutions of Anhui Province under Grant KJ2020ZD28, in part by the Natural Science Foundation for Excellent Young Scholars of Anhui Province under Grant 2108085Y21, in part by the Major Technologies Research and Development Special Program of Anhui Province under Grant 202003a05020001, in part by the Key research and development projects of Anhui Province under Grant 202104a05020015, and in part by the Open Project of China International Science and Technology Cooperation Base on Intelligent Equipment Manufacturing in Special Service Environment under Grant ISTC2021KF04. *(Corresponding author: Ju H. Park.)*

Hao Shen, Yun Wang, and Jing Wang are with the AnHui Province Key Laboratory of Special Heavy Load Robot and the School of Electrical and Information Engineering, Anhui University of Technology, Ma'anshan 243002, China (e-mail: haoshen10@gmail.com; wy18355156234@163.com; jingwang08@126.com).

Ju H. Park is with the Department of Electrical Engineering, Yeungnam University, Kyongsan 38541, South Korea (e-mail: jessie@ynu.ac.kr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TFUZZ.2023.3265666.

Digital Object Identifier 10.1109/TFUZZ.2023.3265666

Furthermore, the Takagi–Sugeno (T-S) fuzzy model provides convenience for the analysis of nonlinear systems, where the nonlinear term can be regionally linearized by using a set of IF–THEN fuzzy rules [25], [26], [27], [28]. Accordingly, many efforts have been devoted to nonlinear MJSPSs [29], [30]. Zhang et al. [31] proposed a fuzzy learning control method for T-S fuzzy systems with partially unknown dynamics. Expanded the work of [31], a decentralized fuzzy tracking learning control method was designed in [32] for interconnected systems. Fang et al. [33] developed an off-policy learning control method for discrete-time Markov jump fuzzy systems. To the best of our knowledge, there is little attention that focuses on designing a learning controller for fuzzy MJSPSs, which stimulates our work.

C. Contribution

Summarizing the abovementioned considerations, based on the T-S fuzzy model, we devote to designing an RL-based optimal controller for nonlinear MJSPSs. The main contributions of this article can be organized as follows.

1. As the first attempt, the optimal controller design problem for the nonlinear MJSPSs is solved under the framework of IRL.
2. Compared with the existing learning control method for MJSs [18], [19], [20], [21], [22], in this article, the characteristics of nonlinearity and two-time-scale are taken into consideration. On this basis, a model-based parallel iteration learning method and an off-policy IRL-based parallel control scheme are presented to obtain the optimal controller for nonlinear MJSPSs.
3. From a new perspective, the slow and fast sampled system states data are employed during the process of the optimal controller designing for nonlinear MJSPSs, where the numerical ill-conditioned problem and mode coupled problem can be avoided.

D. Structure and Notation

The main contents of this article are organized as follows. The problem formulation and an offline parallel algorithm are given in Section II. Section III proposes a novel online parallel algorithm. In Section IV, a tunnel diode circuit model is given to show the efficacy of the developed algorithms. Section V gives the conclusion. Throughout this article, the used notations are shown in the following: $\mathbb{R}^n$ denotes an n-dimensional real vector. $\mathbb{E}\{\cdot\}$ is the mathematical expectation. $\otimes$ means the Kronecker product operator. For a column vector $L \in \mathbb{R}^n$, $vecv(L) = [l_1^2, \ldots, l_1 l_n, l_2^2, l_2 l_3, \ldots, l_{n-1} l_n, l_n^2]^T \in \mathbb{R}^{n(n+1)/2}$. For a matrix $Q \in \mathbb{R}^{n\times m}$, $vec(Q) = [q_{11}, q_{12}, \ldots, q_{1m}, q_{21}, \ldots, q_{nm}]^T \in \mathbb{R}^{nm}$. For an asymmetric matrix $V \in \mathbb{R}^{n\times n}$, $vecs(V) = [v_{11}, 2v_{12}, \ldots, 2v_{1n}, v_{22}, 2v_{23}, \ldots, 2v_{n-1,n}, v_{nn}]^T \in \mathbb{R}^{n(n+2)/2}$.

II. Problem Formulation

A. Problem Statement

We consider the nonlinear MJSPSs described by the T-S fuzzy model as follows.

Plant rule λ*:* **IF** $\alpha_1(\ell)$ is $M_{\lambda 1}, \ldots, \alpha_b(\ell)$ is $M_{\lambda b}$, **THEN**

$$\begin{bmatrix}\dot{x}_1(\ell)\\ \varepsilon\dot{x}_2(\ell)\end{bmatrix} = \begin{bmatrix}A_\lambda^{11}(r(\ell)) & A_\lambda^{12}(r(\ell))\\ A_\lambda^{21}(r(\ell)) & A_\lambda^{22}(r(\ell))\end{bmatrix}\begin{bmatrix}x_1(\ell)\\ x_2(\ell)\end{bmatrix} + \begin{bmatrix}B_\lambda^{1}(r(\ell))\\ B_\lambda^{2}(r(\ell))\end{bmatrix}u(\ell) \quad (1)$$

with the transition probabilities

$$\Pr\{r(\ell+\triangle\ell) = j | r(\ell) = i\} = \begin{cases}\pi_{ij}\Delta\ell + o(\Delta\ell) & j \neq i\\ 1 + \pi_{ii}\Delta\ell + o(\Delta\ell) & j = i\end{cases}$$

where $x_1(\ell) \in \mathbb{R}^{n_1}$ and $x_2(\ell) \in \mathbb{R}^{n_2}$ $(n_1 + n_2 = n)$ represent the slow and fast state vectors, respectively, $u(\ell) \in \mathbb{R}^m$ is the control input vector, $\lambda = 1, 2, \ldots, s,\ s > 1$ denotes the number of **IF–THEN** rules, $M_{\lambda p}, p = 1, 2, \ldots, b$ is the fuzzy set, $\alpha(\ell) = [\alpha_1(\ell), \ldots, \alpha_b(\ell)]^T$ is the premise variable, $0 < \varepsilon \ll 1$ denotes the SPP, and $\{r(\ell), \ell \geq 0\}$ is the system mode subject to Markov stochastic process and takes values in $\mathbb{S} = \{1, 2, \ldots, N\}$. Moreover, for $i,\ j \in \mathbb{S}, \Delta\ell > 0, \lim_{\Delta\ell\to 0}\frac{o(\Delta\ell)}{\Delta\ell} = 0$, and $\pi_{ij} \geq 0$ denotes the system transition rate when time from ℓ to $\ell + \Delta\ell$, and the system mode i to mode j, with $\pi_{ii} = -\sum_{i\neq j}\pi_{ij}$. For $\forall r(\ell) = i$, we denote the following matrices:

$$A_{i,\lambda} \triangleq \begin{bmatrix}A_{i,\lambda}^{11} & A_{i,\lambda}^{12}\\ A_{i,\lambda}^{21} & A_{i,\lambda}^{22}\end{bmatrix}, B_{i,\lambda} \triangleq \begin{bmatrix}B_{i,\lambda}^{1}\\ B_{i,\lambda}^{2}\end{bmatrix}$$

$$A_{i,\lambda}^{\varepsilon} \triangleq \begin{bmatrix}A_{i,\lambda}^{11} & A_{i,\lambda}^{12}\\ \varepsilon^{-1}A_{i,\lambda}^{21} & \varepsilon^{-1}A_{i,\lambda}^{22}\end{bmatrix}, B_{i,\lambda}^{\varepsilon} \triangleq \begin{bmatrix}B_{i,\lambda}^{1}\\ \varepsilon^{-1}B_{i,\lambda}^{2}\end{bmatrix}$$

where $A_{i,\lambda} \in \mathbb{R}^{n\times n}$, $A_{i,\lambda}^{\varepsilon} \in \mathbb{R}^{n\times n}$, $B_{i,\lambda} \in \mathbb{R}^{n\times m}$, and $B_{i,\lambda}^{\varepsilon} \in \mathbb{R}^{n\times m}$ are system matrices. Letting $x(\ell) \triangleq [x_1^T(\ell) \quad x_2^T(\ell)]^T$, the overall T-S fuzzy system is given

$$\dot{x}(\ell) = \sum_{\lambda=1}^{s} g_\lambda(\alpha(\ell))\left[A_{i,\lambda}^{\varepsilon}x(\ell) + B_{i,\lambda}^{\varepsilon}u(\ell)\right]$$

where $g_\lambda(\alpha(\ell))$ is the fuzzy weighting functions which are defined as

$$g_\lambda(\alpha(\ell)) = \frac{\prod_{p=1}^{b} M_{\lambda p}(\alpha_p(\ell))}{\sum_{\lambda=1}^{s}\prod_{p=1}^{b} M_{\lambda p}(\alpha_p(\ell))}$$

with $M_{\lambda p}(\alpha_p(\ell))$ representing the grade of membership of $\alpha_p(\ell)$ in $M_{\lambda p}$. Thus, we have

$$0 \leq g_\lambda(\alpha(\ell)) \leq 1,\ \sum_{\lambda=1}^{s} g_\lambda(\alpha(\ell)) = 1.$$

In the following, a parallel distributed compensation (PDC) technology is introduced to design the fuzzy controller.

Control rule λ*:* **IF** $\alpha_1(\ell)$ is $M_{\lambda 1}, \ldots, \alpha_b(\ell)$ is $M_{\lambda b}$, **THEN**

$$\begin{cases}u(\ell) = -K_{i,\lambda}^{\varepsilon}x(\ell)\\ J_\lambda(\ell) = \mathbb{E}\left\{\int_\ell^\infty\left[x^T(\varsigma)Q_i x(\varsigma) + u^T(\varsigma)R_i u(\varsigma)\right]d\varsigma\right\}\end{cases}$$

where $Q_i > 0$ and $R_i > 0$ are positive definite weighting matrices, and $J_\lambda(\ell)$ is the cost function. Then, we have the overall fuzzy controller expressed as

$$u(\ell) = -\sum_{\lambda=1}^{s} g_\lambda(\alpha(\ell))K_{i,\lambda}^{\varepsilon}x(\ell).$$

B. Offline Learning Scheme Design for Nonlinear MJSPSs

By the optimal control theory, the matrix $P_{i,\lambda}^{\varepsilon}$ is the solution of the following coupled algebraic Riccati equations (CAREs):

$$(A_{i,\lambda}^{\varepsilon})^T P_{i,\lambda}^{\varepsilon} + P_{i,\lambda}^{\varepsilon}A_{i,\lambda}^{\varepsilon} + \sum_{j=1}^{N}\pi_{ij}P_{j,\lambda}^{\varepsilon} + Q_i - P_{i,\lambda}^{\varepsilon}B_{i,\lambda}^{\varepsilon}R_i^{-1}(B_{i,\lambda}^{\varepsilon})^T P_{i,\lambda}^{\varepsilon} = 0 \quad (2)$$

where $P^{\varepsilon}_{i,\lambda}$ has the following form:

$$P^{\varepsilon}_{i,\lambda} = \begin{bmatrix} P^{(1,1)}_{i,\lambda} & \varepsilon\left(P^{(2,1)}_{i,\lambda}\right)^T \\ \varepsilon P^{(2,1)}_{i,\lambda} & \varepsilon P^{(2,2)}_{i,\lambda} \end{bmatrix}$$

with $P^{(1,1)}_{i,\lambda} \in \mathbb{R}^{n_1 \times n_1}$, $P^{(2,1)}_{i,\lambda} \in \mathbb{R}^{n_2 \times n_1}$, and $P^{(2,2)}_{i,\lambda} \in \mathbb{R}^{n_2 \times n_2}$. Note that there exists a coupled term $\sum_{j=1}^{N} \pi_{ij} P^{\varepsilon}_{j,\lambda}$ in CAREs (2). Thus, let $\bar{A}^{\varepsilon}_{i,\lambda} \triangleq A^{\varepsilon}_{i,\lambda} + \frac{\pi_{ii}}{2} I$, which implies

$$(\bar{A}^{\varepsilon}_{i,\lambda})^T P^{\varepsilon}_{i,\lambda} + P^{\varepsilon}_{i,\lambda}\bar{A}^{\varepsilon}_{i,\lambda} + \sum_{j=1,i\neq j}^{N} \pi_{ij} P^{\varepsilon}_{j,\lambda} + Q_i - P^{\varepsilon}_{i,\lambda} B^{\varepsilon}_{i,\lambda} R_i^{-1} (B^{\varepsilon}_{i,\lambda})^T P^{\varepsilon}_{i,\lambda} = 0. \tag{3}$$

Then, the following lemma presents an offline parallel iteration algorithm to solve CAREs (3).

Lemma 1 ([18]): For every fuzzy rule λ, give initial stabilizing matrices $K^{\varepsilon}_{i,\lambda(0)}$, matrices $P^{\varepsilon}_{i,\lambda}$ are the solutions of the following Lyapunov equations:

$$(\tilde{A}^{\varepsilon}_{i,\lambda(k)})^T P^{\varepsilon}_{i,\lambda(k)} + P^{\varepsilon}_{i,\lambda(k)} \tilde{A}^{\varepsilon}_{i,\lambda(k)} + Q_i + \sum_{j=1,j\neq i}^{N} \pi_{ij} P^{\varepsilon}_{j,\lambda(k)} + \left(K^{\varepsilon}_{i,\lambda(k)}\right)^T R_i K^{\varepsilon}_{i,\lambda(k)} = 0 \tag{4}$$

where $\tilde{A}^{\varepsilon}_{i,\lambda(k)} = \bar{A}^{\varepsilon}_{i,\lambda} - B^{\varepsilon}_{i,\lambda} K^{\varepsilon}_{i,\lambda(k)}$, and

$$K^{\varepsilon}_{i,\lambda(k)} = R_i^{-1} \left(B^{\varepsilon}_{i,\lambda}\right)^T P^{\varepsilon}_{i,\lambda(k-1)}. \tag{5}$$

Then, we have $\lim_{k\to\infty} P^{\varepsilon}_{i,\lambda(k)} = P^{\varepsilon *}_{i,\lambda}$, where the $P^{\varepsilon *}_{i,\lambda}$ denote the optimal values.

Remark 1: Note that the solution of (3) can be parallelly approximated by solving N Lyapunov equations (4). However, due to the existence of the SPP, solving (4) would lead to the ill-conditioned problem. To deal with this problem, one way is to design a composite controller for different time scales, which may bring some computation complexity [23]. Hence, in this article, attention is focused on designing the optimal controller for MJSPSs based on the full-order model.

Letting $U^{\varepsilon} \triangleq \mathrm{diag}\{I_{n1}, \varepsilon I_{n2}\}$, matrices $P^{\varepsilon}_{i,\lambda}$ can be rewritten as

$$P^{\varepsilon}_{i,\lambda} = U^{\varepsilon}\tilde{P}^{\varepsilon}_{i,\lambda} = \left(\tilde{P}^{\varepsilon}_{i,\lambda}\right)^T U^{\varepsilon}$$

where $\tilde{P}^{\varepsilon}_{i,\lambda} = \begin{bmatrix} P^{(1,1)}_{i,\lambda} & \varepsilon(P^{(2,1)}_{i,\lambda})^T \\ P^{(2,1)}_{i,\lambda} & P^{(2,2)}_{i,\lambda} \end{bmatrix}$, then CAREs (3) can be rewritten as

$$\mathbb{A}^T_{i,\lambda}\tilde{P}^{\varepsilon}_{i,\lambda} + \left(\tilde{P}^{\varepsilon}_{i,\lambda}\right)^T \mathbb{A}_{i,\lambda} + \bar{Q}_i - \left(\tilde{P}^{\varepsilon}_{i,\lambda}\right)^T B_{i,\lambda} R_i^{-1} B^T_{i,\lambda} \tilde{P}^{\varepsilon}_{i,\lambda} = 0 \tag{6}$$

where

$$\mathbb{A}_{i,\lambda} = A_{i,\lambda} + \frac{\pi_{ii}}{2} U^{\varepsilon}, \bar{Q}_i = Q_i + \sum_{j=1,i\neq j}^{N} \pi_{ij} U^{\varepsilon} \tilde{P}^{\varepsilon}_{j,\lambda}.$$

Based on the transformation of matrices $P^{\varepsilon}_{i,\lambda}$, the following Algorithm 1 is proposed to solve (6).

Remark 2: Compared with existing studies about MJSs [34], [35], [36], we take the effect of SPP into account for the MJSs, and propose a novel offline parallel learning algorithm, which can effectively avoid the ill-conditioned problem. Moreover, Algorithm 1 gives the exact solutions of $\tilde{P}^{\varepsilon *}_{i,\lambda}$ when $k \to \infty$. Meanwhile, it is worth emphasizing that Algorithm 1 is a model-based learning control method, which needs information of the system dynamics.

Algorithm 1: Offline Model-Based Parallel Algorithm for Nonlinear MJSPSs.

Initialization
Select the initial stabilizing matrices sequence $\tilde{P}^{\varepsilon}_{i,\lambda(0)}$. ϵ is a small positive constant.
for *fuzzy rule* $\lambda = 1 : s$ **do**
  for $i = 1 : N$ **do**
    while $\max\{||\mathrm{vecs}(\tilde{P}^{\varepsilon}_{i,\lambda(k+1)}) - \mathrm{vecs}(\tilde{P}^{\varepsilon}_{i,\lambda(k)})||\} \leq \epsilon$ **do**
      Parallel solve

$$\tilde{\mathbb{A}}^T_{i,\lambda(k)}\tilde{P}^{\varepsilon}_{i,\lambda(k)} + \left(\tilde{P}^{\varepsilon}_{i,\lambda(k)}\right)^T \tilde{\mathbb{A}}_{i,\lambda(k)} + \bar{Q}_{i(k)} + \tilde{P}^{\varepsilon}_{i,\lambda(k-1)} B_{i,\lambda} R_i^{-1} B^T_{i,\lambda} \tilde{P}^{\varepsilon}_{i,\lambda(k-1)} = 0 \tag{7}$$

      where

$$\begin{cases} \tilde{\mathbb{A}}_{i,\lambda(k)} = \mathbb{A}_{i,\lambda} - B_{i,\lambda} R_i^{-1} B^T_{i,\lambda} \tilde{P}^{\varepsilon}_{i,\lambda(k-1)}, \\ \bar{Q}_{i(k)} = Q_i + \sum_{j=1,j\neq i}^{N} \pi_{ij} U^{\varepsilon} \tilde{P}^{\varepsilon}_{j,\lambda(k)} \end{cases} \tag{8}$$

    end
  end
end

III. Main Results

A novel online parallel IRL algorithm is proposed to solve the optimal feedback control problem by employing system data, which means that Algorithm 2 is model-free. Moreover, the convergence of this novel online parallel IRL algorithm is also discussed in this section.

A. Subsystems Transformation

Considering that the coupling term $\sum_{j=1}^{N} \pi_{ij} P^{\varepsilon}_{j,\lambda}$ exists in CAREs (2), which is difficult to be obtained. To deal with this problem, the following Theorem 1 is proposed. The subsystem transformation technology is utilized to decompose the nonlinear MJSPSs, so that the information of π_{ij} is separated.

Theorem 1: Assume that the solutions of the following N reconstructed subsystems are $P^{\varepsilon}_{i,\lambda(k)}$. According to [18], we have

$$\begin{cases} \dot{x}_{i,\lambda}(\ell) = \bar{A}^{\varepsilon}_{i,\lambda} x_{i,\lambda}(\ell) + B^{\varepsilon}_{i,\lambda} u_{i,\lambda}(\ell) \\ x_{i,\lambda}(0) = x_0,\ \ell_0 = 0 \end{cases} \tag{9}$$

where $x_{i,\lambda}$ and $u_{i,\lambda}$ denote the system state and control input of the ith subsystem, respectively, and matrices $\bar{Q}_{i(k)}$ satisfy the following form:

$$\bar{Q}_{i(k)} = Q_i + \sum_{j=1,i\neq j}^{N} \pi_{ij} U^{\varepsilon} \tilde{P}^{\varepsilon}_{j,\lambda(k)}.$$

Then, we have $\lim_{k\to\infty} P^{\varepsilon}_{i,\lambda(k)} = P^{\varepsilon *}_{i,\lambda}$, where the $P^{\varepsilon *}_{i,\lambda}$ are the solutions of (3).

Proof: The solutions of the N reconstructed subsystems (9) are equal to the solutions of the following equation:

$$\left(\tilde{A}^{\varepsilon}_{i,\lambda(k)}\right)^T P^{\varepsilon}_{i,\lambda(k)} + P^{\varepsilon}_{i,\lambda(k)} \tilde{A}^{\varepsilon}_{i,\lambda(k)} + \bar{Q}_{i(k)} + \left(K^{\varepsilon}_{i,\lambda(k)}\right)^T R_i K^{\varepsilon}_{i,\lambda(k)} = 0 \tag{10}$$

where $K^{\varepsilon}_{i,\lambda(k)} = R_i^{-1}(B^{\varepsilon}_{i,\lambda(k)})^T P^{\varepsilon}_{i,\lambda(k-1)}$. Obviously, (10) is equivalent to (4). This ends the proof. ■

Algorithm 2: Online Model-Free Parallel IRL Algorithm for Nonlinear MJSPSs.

Initialization

Give initial stabilizing sequence $K^{\varepsilon}_{i,\lambda(0)}$, and select a threshold $\epsilon > 0$.

for *fuzzy rule* $\lambda = 1 : s$ **do**

  for $i = 1 : N$ **do**

    Data collection:

    Apply initial control policies $u_{i,\lambda}(\ell) = -K^{\varepsilon}_{i,\lambda(0)} x_{i,\lambda}(\ell) + e_{i,\lambda}$ in time interval $[\ell, \ell+T]$, where $e_{i,\lambda}$ represents the exploration noise. Compute $I_{i\lambda xx}, I_{i\lambda xu}$.

    while $\max\{||\text{vecs}(\tilde{P}^{\varepsilon}_{i,\lambda(k+1)}) - \text{vecs}(\tilde{P}^{\varepsilon}_{i,\lambda(k)})||\} \geq \epsilon$ **do**

      Iterative computation:

      Parallel solve $K^{\varepsilon}_{i,\lambda(k)}$ and $P^{\varepsilon}_{i,\lambda(k+1)}$ from

$$\begin{aligned} &x^T_{i,\lambda}(\ell+T)U^{\varepsilon}\tilde{P}^{\varepsilon}_{i,\lambda(k)}x_{i,\lambda}(\ell+T) - x^T_{i,\lambda}(\ell)U^{\varepsilon} \\ &\times \tilde{P}^{\varepsilon}_{i,\lambda(k)}x_{i,\lambda}(\ell) = -\int_{\ell}^{\ell+T} x^T_{i,\lambda}(\varsigma)\hat{Q}_{i(k)}x_{i,\lambda}(\varsigma)d\varsigma \\ &+2\int_{\ell}^{\ell+T}[(u_{i,\lambda}(\varsigma) + K^{\varepsilon}_{i,\lambda(k)}x_{i,\lambda}(\varsigma))^T R_i \\ &\times K^{\varepsilon}_{i,\lambda(k+1)}x_{i,\lambda}(\varsigma)]d\varsigma \end{aligned} \tag{11}$$

      where

$$\begin{aligned} \hat{Q}_{i(k)} = Q_i + \textstyle\sum_{j=1,j\neq i}^{N} \pi_{ij}U^{\varepsilon}\tilde{P}^{\varepsilon}_{j,\lambda(k-1)} \\ +(K^{\varepsilon}_{i,\lambda(k)})^T R_i K^{\varepsilon}_{i,\lambda(k)} \end{aligned} \tag{12}$$

      $k \leftarrow k+1;$

    end

  end

end

The overall control policy is

$$u(\ell) = -\sum_{\lambda=1}^{s} g_{\lambda}(\alpha(\ell))K^{\varepsilon *}_{i,\lambda}x(\ell)$$

B. IRL-Based Online Parallel Algorithm Design

Based on Theorem 1, the system states equation can be rewritten as

$$\begin{cases} \dot{x}_{i,\lambda}(\ell) = \tilde{A}^{\varepsilon}_{i,\lambda}x_{i,\lambda}(\ell) + B^{\varepsilon}_{i,\lambda}(u_{i,\lambda}(\ell) \\ +K^{\varepsilon}_{i,\lambda(k)}x_{i,\lambda}(\ell)) \\ x_{i,\lambda}(0) = x_0,\ \ell_0 = 0 \end{cases}$$

where the pair $(\tilde{A}^{\varepsilon}_{i,\lambda}, B_{i,\lambda})$ is stabilizable. Furthermore, an online model-free parallel IRL control method is presented in Algorithm 2.

C. Online Implementation of Algorithm 2

For MJSPSs, using the value function as $V_{i,\lambda}(x_{i,\lambda}(\ell)) = x^T_{i,\lambda}(\ell)P^{\varepsilon}_{i,\lambda}x_{i,\lambda}(\ell)$ and weak infinitesimal operator Γ, we have

$$\begin{aligned} &\Gamma V_{i,\lambda}(x_{i,\lambda}(\ell)) \\ &= x^T_{i,\lambda}(\ell)\left(\left(\tilde{A}^{\varepsilon}_{i,\lambda(k)}\right)^T P^{\varepsilon}_{i,\lambda} + P^{\varepsilon}_{i,\lambda}\tilde{A}^{\varepsilon}_{i,\lambda(k)}\right)x_{i,\lambda}(\ell) \\ &\quad + 2(u_{i,\lambda}(\ell) + K^{\varepsilon}_{i,\lambda(k)}x_{i,\lambda}(\ell))^T\left(B^{\varepsilon}_{i,\lambda}\right)^T P^{\varepsilon}_{i,\lambda}x_{i,\lambda}(\ell) \\ &= -x^T_{i,\lambda}(\ell)\hat{Q}_{i(k)}x_{i,\lambda}(\ell) + 2\left(u_{i,\lambda}(\ell) + K^{\varepsilon}_{i,\lambda(k)}x_{i,\lambda}(\ell)\right)^T \\ &\quad \times \left(B^{\varepsilon}_{i,\lambda}\right)^T P^{\varepsilon}_{i,\lambda}x_{i,\lambda}(\ell) \end{aligned} \tag{13}$$

where $\hat{Q}_{i(k)} = Q_i + \sum_{j=1,i\neq j}^{N} \pi_{ij}U^{\varepsilon}\tilde{P}^{\varepsilon}_{j,\lambda(k)} + (K^{\varepsilon}_{i,\lambda(k)})^T R_i K^{\varepsilon}_{i,\lambda(k)}$. For integral interval $[\ell, \ell+T]$, by integrating (12), one can get

$$\begin{aligned} &x^T_{i,\lambda}(\ell+T)U^{\varepsilon}\tilde{P}^{\varepsilon}_{i,\lambda(k)}x_{i,\lambda}(\ell+T) - x^T_{i,\lambda}(\ell)U^{\varepsilon}\tilde{P}^{\varepsilon}_{i,\lambda(k)}x_{i,\lambda}(\ell) \\ &= -\int_{\ell}^{\ell+T} x^T_{i,\lambda}(\varsigma)\hat{Q}_{i(k)}x_{i,\lambda}(\varsigma)d\varsigma + 2\int_{\ell}^{\ell+T}[(u_{i,\lambda}(\varsigma) \\ &\quad + K^{\varepsilon}_{i,\lambda(k)}x_{i,\lambda}(\varsigma))^T R_i K^{\varepsilon}_{i,\lambda(k+1)}x_{i,\lambda}(\varsigma)]d\varsigma. \end{aligned} \tag{14}$$

To guarantee the symmetry of $U^{\varepsilon}\tilde{P}^{\varepsilon}_{i,\lambda}$, transform the term $x^T_{i,\lambda}U^{\varepsilon}\tilde{P}^{\varepsilon}_{i,\lambda(k)}x_{i,\lambda}$ into the following form:

$$x^T_{i,\lambda}(\ell)U^{\varepsilon}\tilde{P}^{\varepsilon}_{i,\lambda(k)}x_{i,\lambda}(\ell) = \text{vec}v(x_{i,\lambda}(\ell))^T E_{\varepsilon}\text{vec}s(\tilde{P}^{\varepsilon}_{i,\lambda(k)})$$

where $E_{\varepsilon} = \text{diag}\{I_{n_m}, \varepsilon I_{n_k}\}$, $n_m = n_1(n_1+1)/2$, and $n_k = n_1 n_2 + n_2(n_2+1)/2$. Combining with (14), we have

$$\begin{aligned} &x^T_{i,\lambda}(\ell+T)U^{\varepsilon}\tilde{P}^{\varepsilon}_{i,\lambda(k)}x_{i,\lambda}(\ell+T) - x^T_{i,\lambda}(\ell)U^{\varepsilon}\tilde{P}^{\varepsilon}_{i,\lambda(k)}x_{i,\lambda}(\ell) \\ &= \left(\text{vec}v(x_{i,\lambda}(\ell+T))^T - \text{vec}v(x_{i,\lambda}(\ell))^T\right)E_{\varepsilon}\text{vec}s(\tilde{P}^{\varepsilon}_{i,\lambda(k)}). \end{aligned}$$

Similar to the abovementioned method, we can obtain

$$\begin{aligned} &x^T_{i,\lambda}(\ell)\hat{Q}_{i(k)}x_{i,\lambda}(\ell) = (x^T_{i,\lambda}(\ell)\otimes x^T_{i,\lambda}(\ell))\text{vec}(\hat{Q}_{i(k)}), \\ &(u_{i,\lambda}(\ell) + K^{\varepsilon}_{i,\lambda(k)}x_{i,\lambda}(\ell))^T R_i K^{\varepsilon}_{i,\lambda(k+1)}x_{i,\lambda}(\ell) \\ &= (x^T_{i,\lambda}(\ell)\otimes x^T_{i,\lambda}(\ell))(I_n\otimes K^{\varepsilon}_{i,\lambda(k)}R_i)]\text{vec}(K^{\varepsilon}_{i,\lambda(k+1)}) \\ &\quad + (x^T_{i,\lambda}(\ell)\otimes u^T_{i,\lambda}(\ell))(I_n\otimes R_i)]\text{vec}(K^{\varepsilon}_{i,\lambda(k+1)}). \end{aligned}$$

Define the following operators:

$$I_{i\lambda xx} = \begin{bmatrix} \int_{\ell_0}^{\ell_1} x^T_{i,\lambda}(\varsigma)\otimes x^T_{i,\lambda}(\varsigma)d\varsigma \\ \int_{\ell_1}^{\ell_2} x^T_{i,\lambda}(\varsigma)\otimes x^T_{i,\lambda}(\varsigma)d\varsigma \\ \vdots \\ \int_{\ell_{l-1}}^{\ell_l} x^T_{i,\lambda}(\varsigma)\otimes x^T_{i,\lambda}(\varsigma)d\varsigma \end{bmatrix}$$

$$I_{i\lambda xu} = \begin{bmatrix} \int_{\ell_0}^{\ell_1} x^T_{i,\lambda}(\varsigma)\otimes u^T_{i,\lambda}(\varsigma)d\varsigma \\ \int_{\ell_1}^{\ell_2} x^T_{i,\lambda}(\varsigma)\otimes u^T_{i,\lambda}(\varsigma)d\varsigma \\ \vdots \\ \int_{\ell_{l-1}}^{\ell_l} x^T_{i,\lambda}(\varsigma)\otimes u^T_{i,\lambda}(\varsigma)d\varsigma \end{bmatrix}$$

where $0 \leq \ell_0 < \ell_1 < \cdots < \ell_l$. Then, (13) can be transformed into the compact form as

$$\Theta_{i,\lambda(k)}\begin{bmatrix} \text{vec}s(\tilde{P}^{\varepsilon}_{i,\lambda(k)}) \\ \text{vec}(K^{\varepsilon}_{i,\lambda(k+1)}) \end{bmatrix} = \Xi_{i,\lambda(k)},\ k = 0, 1, 2, \ldots$$

with $\Theta_{i,\lambda(k)} \in \mathbb{R}^{l\times[\frac{1}{2}n(n+1)+nm]}$. $\Xi_{i,\lambda(k)} \in \mathbb{R}^l$ can be denoted as

$$\Theta_{i,\lambda(k)} = \begin{bmatrix} E_{\varepsilon}\left(\text{vec}v(x_{i,\lambda}(\ell+T)) - \text{vec}v(x_{i,\lambda}(\ell))\right) \\ -2[I_{i\lambda xx}(I_n\otimes(K^{\varepsilon}_{i,\lambda(k)})^T R_i) \\ +I_{i\lambda xu}(I_n\otimes R_i)]^T \end{bmatrix}^T$$

$$\Xi_{i,\lambda(k)} = -I_{i\lambda xx}\text{vec}(\hat{Q}_{i(k)}).$$

If $\Theta_{i,\lambda(k)}$ has full rank, $\tilde{P}^{\varepsilon}_{i,\lambda(k)}$ and $K^{\varepsilon}_{i,\lambda(k+1)}$ can be obtained by

$$\begin{bmatrix} \text{vec}s(\tilde{P}^{\varepsilon}_{i,\lambda(k)}) \\ \text{vec}(K^{\varepsilon}_{i,\lambda(k+1)}) \end{bmatrix} = \left(\Theta^T_{i,\lambda(k)}\Theta_{i,\lambda(k)}\right)^{-1}\Theta^T_{i,\lambda(k)}\Xi_{i,\lambda(k)}. \tag{15}$$

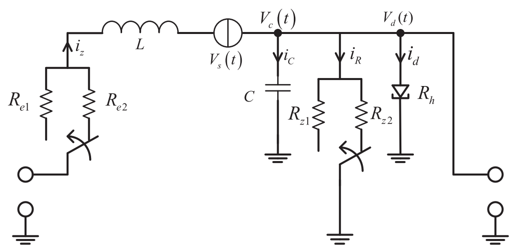

Fig. 1. Structure of tunnel diode circuit system.

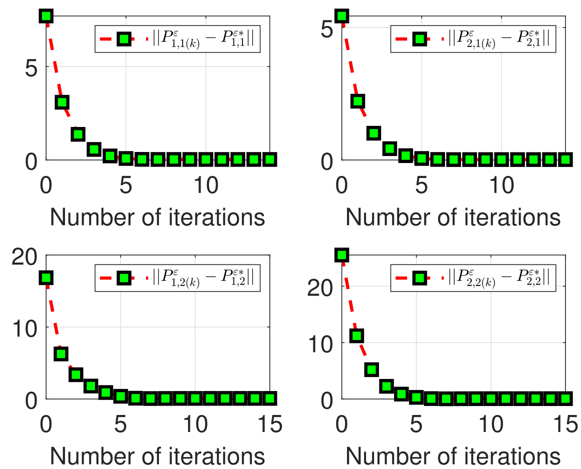

Fig. 2. Convergence of matrices $P^{\varepsilon}_{i,\lambda}$.

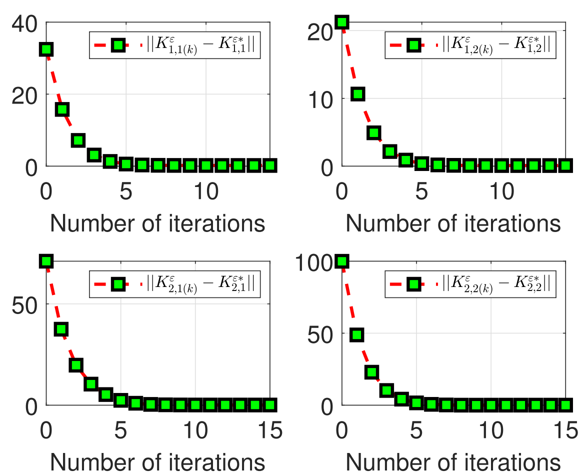

Fig. 3. Convergence of matrices $K^{\varepsilon}_{i,\lambda}$.

Accordingly, the rank condition of $\Theta_{i,\lambda(k)}$ can be ensured by the following Lemma 2.

Lemma 2 ([23]): If there is a constant $l_0 > 0$, for all $l \geq l_0$

$$\text{rank}\left(\left[I_{i\lambda xx} \quad I_{i\lambda xu}\right]\right) = \frac{n(n+1)}{2} + mn$$

then $\Theta_{i,\lambda(k)}$ has full column rank for all k.

Remark 3: Note that the proposed online parallel learning control algorithm to solve the CAREs of nonlinear MJSPSs is a model-free control method. The IRL-based Bellman equation in the Algorithm 2 is utilized to avoid the information of system dynamics. Compared with existing learning control algorithms, the improvement lies in the following twofolds. 1) By means of the PDC technology, the learning controller for nonlinear MJSPSs is designed by solving a set of parallel CAREs, which reduces the computational complexity. 2) The matrix $\tilde{P}^{\varepsilon}_{i,\lambda}$ rather than $P^{\varepsilon}_{i,\lambda}$ is determined by solving the well-posed CAREs in Algorithm 2, which can improve the calculation accuracy.

Remark 4: In Algorithm 2, although the proposed method is model free, it is obvious that the transition probability still exists in (15), which means that transition probability is assumed to be known in this article. Therefore, how to eliminate transition probability by employing the IRL method in MJSPSs deserves our further study.

D. Analysis of Convergence

The convergence of the online parallel learning Algorithm 2 is proved in the following theorem.

Theorem 2: Assume that $\Theta_{i,\lambda(k)}$ is full column rank and gives stable values $K^{\varepsilon}_{i,\lambda(0)}$ $\forall i \in \mathbb{S}$. Then, the solutions of (5) and (6) are equal to (15).

Proof: Given initial stabilizing control gains $K^{\varepsilon}_{i,\lambda(0)}$, the solutions of CAREs (3) can be obtained. According to Lemma 1, we can obtain the exact solutions $P^{\varepsilon *}_{i,\lambda}$ from (4). Denote the cost function as $V_{i,\lambda}(x_{i,\lambda}(\ell)) = x^T_{i,\lambda}(\ell) P^{\varepsilon}_{i,\lambda} x_{i,\lambda}(\ell)$. For N reconstructed subsystems (9) satisfy (7) and (8), the following condition is satisfied:

$$\begin{aligned}
& x^T_{i,\lambda}(\ell+T) U^{\varepsilon} \tilde{P}^{\varepsilon}_{i,\lambda(k)} x_{i,\lambda}(\ell+T) - x^T_{i,\lambda}(\ell) U^{\varepsilon} \tilde{P}^{\varepsilon}_{i,\lambda(k)} x_{i,\lambda}(\ell) \\
&= \int_{\ell}^{\ell+T} \Big[x^T_{i,\lambda}(\varsigma) ((\tilde{A}^{\varepsilon}_{i,\lambda(k)})^T \tilde{P}^{\varepsilon}_{i,\lambda(k)} + \tilde{P}^{\varepsilon}_{i,\lambda(k)} \tilde{A}^{\varepsilon}_{i,\lambda(k)}) x_{i,\lambda}(\varsigma) \\
&\quad + 2(u_{i,\lambda}(\varsigma) + K^{\varepsilon}_{i,\lambda(k)} x_{i,\lambda}(\varsigma))^T (B^{\varepsilon}_{i,\lambda})^T \tilde{P}^{\varepsilon}_{i,\lambda(k)} \Big] d\varsigma \\
&= -\int_{\ell}^{\ell+T} x^T_{i,\lambda}(\varsigma) \hat{Q}_{i(k)} x_{i,\lambda}(\varsigma) d\varsigma \\
&\quad + 2\int_{\ell}^{\ell+T} (u_{i,\lambda}(\varsigma) + K^{\varepsilon}_{i,\lambda(k)} x_{i,\lambda}(\varsigma))^T R_i K^{\varepsilon}_{i,\lambda(k+1)} x_{i,\lambda}(\varsigma) d\varsigma.
\end{aligned}$$

Since $\Theta_{i,\lambda(k)}$ has full column rank, (15) are the online implement of (11). Therefore, the solutions of (15) are equivalent to (5) and (6). The proof is completed. ∎

IV. Simulation

In this example, a tunnel diode circuit model employed from [37] is shown in Fig. 1. The state equation can be described as

$$\begin{cases} C\dot{V}_c(\ell) = -\frac{1}{R_z} V_c(\ell) - \frac{1}{R_h} V_c(\ell) + i_z(\ell) \\ L\dot{i}_l(\ell) = -V_c(\ell) - R_{ei} i_z(\ell) + V_s(\ell) + u(\ell) \end{cases}$$

where the capacitance $C = 0.2$F, the inductance $L = 1$mH, $V_s(\ell) = 1.5V_c(\ell)$, and $R_h = \frac{1}{0.002+0.01V_d^2(t)}$. Assume that the resistances R_z and R_{ei} are influenced by the environment. This system has two working status: 1) $R_{z1} = 180\Omega$ and $R_{e1} = 9\Omega$, and 2) $R_{z2} = 200\Omega$ and $R_{e2} = 10\Omega$. Let $x_1(\ell) = V_c(\ell) \in [-5, 5]$, $x_2(\ell) = i_z(\ell)$, and SPP $\varepsilon = 0.01$, then the T-S fuzzy model can be described as follows.

Fuzzy rule 1: **IF** $x_1(\ell) = -5$ **THEN**

$$U^{\varepsilon} \dot{x}(\ell) = A_{i,1} x(\ell) + B_{i,1} u(\ell).$$

Fuzzy rule 2: **IF** $x_1(\ell) = 5$ **THEN**

$$U^{\varepsilon} \dot{x}(\ell) = A_{i,2} x(\ell) + B_{i,2} u(\ell)$$

where $x(\ell) = [\ x_1^T(\ell) \quad x_2^T(\ell)\ ]^T, U^\varepsilon = \text{diag}\{1, 0.01\}, i = 1, 2$, and

$$A_{i,1} = \begin{bmatrix} -\frac{1}{R_{zi}C} - \frac{0.002+0.01\times 25}{C} & \frac{1}{C} \\ 0.5 & -R_{ei} \end{bmatrix}, B_{i,1} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$A_{i,2} = \begin{bmatrix} -\frac{1}{R_{zi}C} & \frac{1}{C} \\ 0.5 & -R_{ei} \end{bmatrix}, B_{i,2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The transition probability matrix and the fuzzy weighting functions are selected as follows:

$$\Pi = \begin{bmatrix} -3 & 3 \\ 1.5 & -1.5 \end{bmatrix}$$

$$g_1(x_1(\ell)) = \frac{x_1^2(\ell)}{25},\ g_2(x_1(\ell)) = \frac{25 - x_1^2(\ell)}{25}.$$

In addition, let weighting matrices as

$$Q_1 = Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ R_1 = R_2 = 0.1I.$$

By employing Algorithm 1, the following optimal matrices are calculated as:

$$P_{1,1}^{\varepsilon *} = \begin{bmatrix} 0.3405 & 0.0017 \\ 0.0017 & 0.0005 \end{bmatrix}, P_{2,1}^{\varepsilon *} = \begin{bmatrix} 0.3453 & 0.0016 \\ 0.0016 & 0.0005 \end{bmatrix}$$

$$P_{1,2}^{\varepsilon *} = \begin{bmatrix} 0.7162 & 0.0037 \\ 0.0037 & 0.0005 \end{bmatrix}, P_{2,2}^{\varepsilon *} = \begin{bmatrix} 0.7454 & 0.0035 \\ 0.0035 & 0.0005 \end{bmatrix}$$

$$K_{1,1}^{\varepsilon *} = \begin{bmatrix} 1.7981 & 0.5487 \end{bmatrix}, K_{2,1}^{\varepsilon *} = \begin{bmatrix} 1.6585 & 0.4960 \end{bmatrix}$$

$$K_{1,2}^{\varepsilon *} = \begin{bmatrix} 3.7751 & 0.5590 \end{bmatrix}, K_{2,2}^{\varepsilon *} = \begin{bmatrix} 3.5723 & 0.5051 \end{bmatrix}.$$

From Algorithm 1, the optimal control gains can be calculated by accurate system model information. Compared with Algorithm 1, by employing system data $x(\ell)$ and $u(\ell)$, the online Algorithm 2 can obtain the optimal control policies without using system dynamics. The convergences of matrices $P_{i,\lambda}^{\varepsilon}$ and $K_{i,\lambda}^{\varepsilon}$ are presented in Figs. 2 and 3, respectively, which means that the approximated optimal solutions can be obtained after online learning interaction.

V. Conclusion

This article has investigated the optimal control problem of nonlinear MJSPSs based on IRL. The T-S fuzzy model has been used to approximate the nonlinear feature of MJSPSs. Besides, a novel offline parallel learning algorithm has been proposed, where system dynamics should be known as a priori. To solve this problem, based on IRL method, we have presented an online model-free parallel algorithm to obtain the optimal control policy for nonlinear MJSPSs. The convergence has also been proven. Finally, a tunnel diode circuit model has been employed to demonstrate the effectiveness of the proposed method. It is worth noting that the methods proposed in this article will invalid when the information of Markov chain is not available. Hence, it is significant to extend the proposed control methods to hidden MJSPSs. Actually, stochastic jump and two-time-scale phenomena widely exist in real engineering applications. Thus, we will extend the proposed RL-based control methods to practical application systems, such as looper hydraulic servo systems in hot strip rolling.

References

[1] V. Ugrinovskii and H. R. Pota, "Decentralized control of power systems via robust control of uncertain Markov jump parameter systems," *Int. J. Control*, vol. 78, no. 9, pp. 662–677, 2005.

[2] A. N. Vargas, E. F. Costa, and J. B. do Val, "On the control of Markov jump linear systems with no mode observation: Application to a DC motor device," *Int. J. Robust Nonlinear Control*, vol. 23, no. 10, pp. 1136–1150, 2013.

[3] L. Zhang and C. Prieur, "Stochastic stability of Markov jump hyperbolic systems with application to traffic flow control," *Automatica*, vol. 86, pp. 29–37, 2017.

[4] S. Wang, Z.-G. Wu, P. Shi, and Z. Wu, "Asynchronous control for discrete-time Markovian jump systems with multiplicative noise," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, early access, Jan. 18, 2023, doi: 10.1109/TCSII.2023.3237902.

[5] Z.-G. Wu, P. Shi, Z. Shu, H. Su, and R. Lu, "Passivity-based asynchronous control for Markov jump systems," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 2020–2025, Apr. 2017.

[6] X. Zhou, Y. Tang, J. Cheng, J. Cao, C. Xue, and D. Yan, "Nonstationary quantized control for discrete-time Markov jump singularly perturbed systems against deception attacks," *J. Franklin Inst.*, vol. 358, no. 6, pp. 2915–2932, 2021.

[7] G. Wang, C. Huang, Q. Zhang, and C. Yang, "Stabilisation bound of stochastic singularly perturbed systems with Markovian switching by noise control," *IET Control Theory Appl.*, vol. 8, no. 5, pp. 367–374, 2014.

[8] G. Wang and L. Xu, "Stability and stabilization for singularly perturbed systems with Markovian jumps," *Int. J. Robust Nonlinear Control*, vol. 30, no. 12, pp. 4690–4707, 2020.

[9] Y. Wang, F. Chen, G. Zhuang, and G. Yang, "Dynamic event-based mixed H_∞ and dissipative asynchronous control for Markov jump singularly perturbed systems," *Appl. Math. Comput.*, vol. 386, 2020, Art. no. 125443.

[10] J. Wang, C. Yang, H. Shen, J. Cao, and L. Rutkowski, "Sliding-mode control for slow-sampling singularly perturbed systems subject to Markov jump parameters," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 51, no. 12, pp. 7579–7586, Dec. 2021.

[11] V. Drăgan, P. Shi, and E.-K. Boukas, "Control of singularly perturbed systems with Markovian jump parameters: An H_∞ approach," *Automatica*, vol. 35, no. 8, pp. 1369–1378, 1999.

[12] F. Tang, B. Niu, G. Zong, X. Zhao, and N. Xu, "Periodic event-triggered adaptive tracking control design for nonlinear discrete-time systems via reinforcement learning," *Neural Netw.*, vol. 154, pp. 43–55, 2022.

[13] Y. Zhao, H. Wang, N. Xu, G. Zong, and X. Zhao, "Reinforcement learning-based decentralized fault tolerant control for constrained interconnected nonlinear systems," *Chaos, Solitons Fractals*, vol. 167, 2023, Art. no. 113034.

[14] H. Zhang, X. Zhao, H. Wang, G. Zong, and N. Xu, "Hierarchical sliding-mode surface-based adaptive actor–critic optimal control for switched nonlinear systems with unknown perturbation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 14, 2022, doi: 10.1109/TNNLS.2022.3183991.

[15] H. Zhang, H. Wang, B. Niu, L. Zhang, and A. M. Ahmad, "Sliding-mode surface-based adaptive actor-critic optimal control for switched nonlinear systems with average dwell time," *Inf. Sci.*, vol. 580, pp. 756–774, 2021.

[16] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, 2009.

[17] Y. Jiang and Z.-P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699–2704, 2012.

[18] S. He, J. Song, Z. Ding, and F. Liu, "Online adaptive optimal control for continuous-time Markov jump linear systems using a novel policy iteration algorithm," *IET Control Theory Appl.*, vol. 9, no. 10, pp. 1536–1543, 2015.

[19] S. He, M. Zhang, H. Fang, F. Liu, X. Luan, and Z. Ding, "Reinforcement learning and adaptive optimization of a class of Markov jump systems with completely unknown dynamic information," *Neural Comput. Appl.*, vol. 32, no. 18, pp. 14311–14320, 2020.

[20] Y. Tu, H. Fang, H. Wang, K. Shi, and S. He, "Reinforcement learning-based adaptive optimal tracking algorithm for Markov jump systems with partial unknown dynamics," *Optim. Control Appl. Methods*, vol. 43, no. 5, pp. 1435–1449, doi 10.1002/oca.2903.

[21] J. Song, S. He, Z. Ding, and F. Liu, "A new iterative algorithm for solving H_∞ control problem of continuous-time Markovian jumping linear systems based on online implementation," *Int. J. Robust Nonlinear Control*, vol. 26, no. 17, pp. 3737–3754, 2016.

[22] X. Xin et al., "Online reinforcement learning multiplayer non-zero sum games of continuous-time Markov jump linear systems," *Appl. Math. Comput.*, vol. 412, 2022, Art. no. 126537.

[23] J. Zhao, C. Yang, and W. Gao, "Reinforcement learning based optimal control of linear singularly perturbed systems," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 3, pp. 1362–1366, Mar. 2022.

[24] Y. Zhang, L. Ma, C. Yang, and W. Dai, “Reinforcement learning based sliding mode tracking control for the two-time-scale systems: Dealing with actuator attacks,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 9, pp. 3819–3823, Sep. 2022.

[25] Y. Ma and C. Kong, “Dissipative asynchronous TS fuzzy control for singular semi-Markovian jump systems,” *IEEE Trans. Cybern.*, vol. 52, no. 6, pp. 5454–5463, Jun. 2022.

[26] C.-C. Ku, W.-J. Chang, M.-H. Tsai, and Y.-C. Lee, “Observer-based proportional derivative fuzzy control for singular Takagi-Sugeno fuzzy systems,” *Inf. Sci.*, vol. 570, pp. 815–830, 2021.

[27] X. Su, Y. Wen, P. Shi, S. Wang, and W. Assawinchaichote, “Event-triggered fuzzy control for nonlinear systems via sliding mode approach,” *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 2, pp. 336–344, Feb. 2021.

[28] J. Wang, C. Yang, J. Xia, Z.-G. Wu, and H. Shen, “Observer-based sliding mode control for networked fuzzy singularly perturbed systems under weighted try-once-discard protocol,” *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 6, pp. 1889–1899, Jun. 2022.

[29] H. Li, Y. Wang, D. Yao, and R. Lu, “A sliding mode approach to stabilization of nonlinear Markovian jump singularly perturbed systems,” *Automatica*, vol. 97, pp. 404–413, 2018.

[30] H. Shen, X. Hu, J. Wang, J. Cao, and W. Qian, “Non-fragile H_∞ synchronization for Markov jump singularly perturbed coupled neural networks subject to double-layer switching regulation,” *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 6, 2021, doi: 10.1109/TNNLS.2021.3107607.

[31] K. Zhang, H. Zhang, Y. Mu, and S. Sun, “Tracking control optimization scheme for a class of partially unknown fuzzy systems by using integral reinforcement learning architecture,” *Appl. Math. Comput.*, vol. 359, pp. 344–356, 2019.

[32] K. Zhang, H. Zhang, Y. Mu, and C. Liu, “Decentralized tracking optimization control for partially unknown fuzzy interconnected systems via reinforcement learning method,” *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 4, pp. 917–926, Apr. 2021.

[33] H. Fang et al., “Fuzzy-based adaptive optimization of unknown discrete-time nonlinear Markov jump systems with off-policy reinforcement learning,” *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 12, pp. 5276–5290, Dec. 2022.

[34] M. Xue, H. Yan, H. Zhang, X. Zhan, and K. Shi, “Compensation-based output feedback control for fuzzy Markov jump systems with random packet losses,” *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 12759–12770, Dec. 2022.

[35] H. Yang, Y. Jiang, and S. Yin, “Adaptive fuzzy fault-tolerant control for Markov jump systems with additive and multiplicative actuator faults,” *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 4, pp. 772–785, Apr. 2021.

[36] P. Cheng, S. He, X. Luan, and F. Liu, “Finite-region asynchronous H_∞ control for 2D Markov jump systems,” *Automatica*, vol. 129, 2021, Art. no. 109590.

[37] Y. Wang, P. Shi, and H. Yan, “Reliable control of fuzzy singularly perturbed systems and its application to electronic circuits,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 10, pp. 3519–3528, Oct. 2018.