

To appear in Int J RNC, 2016

INTERNATIONAL JOURNAL OF  
**Robust and Nonlinear Control**

**A New Iterative Algorithm for Solving  $H_\infty$  Control Problem  
of Continuous-Time Markovian Jumping Linear Systems  
Based on Online Implementation**

Journal:	<i>International Journal of Robust and Nonlinear Control</i>
Manuscript ID	RNC-15-0222.R1
Wiley - Manuscript type:	Research Article
Date Submitted by the Author:	12-Jan-2016
Complete List of Authors:	Song, Jun; School of Electrical Engineering and Automation, Anhui University,; School of Information Science and Engineering, East China University of Science and Technology He, Shuping; School of Electrical Engineering and Automation,, Anhui University, Ding, Zhengtao; University of Manchester, School of Electrical and Electronic Engineering Liu, Fei; Institute of Automation, Jiangnan University
Keywords:	Markov jump linear systems (MJLSs);, H-infinite control;; Subsystems Transformation;; Coupled game algebraic Riccati equation (CGARE);, Online

SCHOLARONE™  
Manuscripts

## Novelty

This paper studied the H-infinite control problem of continuous-time Markovian jumping linear systems via a new online iterative algorithm. The main study of this paper are within the scope of International Journal of Robust and Nonlinear Control, such as the model based control design, the H-infinite controller design, the Lyapunov based techniques, the adaptive learning control methods, the relevant iterative algorithm and etc.

The main contribution of this paper are mainly as follows:

1. This paper first studied a new online iterative algorithm to solve the H-infinite control problem of continuous-time Markovian jumping linear systems.
2. Based on the on-line iterative algorithm and a new online decoupling technique named Subsystems Transformation method, a set of linear subsystems which implementation in parallel are obtained.
3. By means of the online Adaptive Dynamic Programming technique, the two-player zero-sum game and the coupled game algebraic Riccati equation (CGARE) are solved. The convergence of the novel policy iteration algorithm is also established.
4. We also proposed the relevant simulation results to show the effectiveness and applicability of the proposed methods.

The relevant Keywords which correspond with the topics covered by the journal and dealt with in the paper are as follows:

control design, control applications, adaptive control, H-infinite control

# A New Iterative Algorithm for Solving $H_\infty$ Control Problem of Continuous-Time Markovian Jumping Linear Systems Based on Online Implementation

Jun Song<sup>1,2</sup>, Shuping He<sup>1\*</sup>, Zhengtao Ding<sup>3</sup>, Fei Liu<sup>4</sup>

<sup>1</sup>School of Electrical Engineering and Automation, Anhui University, Hefei 230601, China

<sup>2</sup>School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China

<sup>3</sup>Control Systems Center, School of Electrical and Electronic Engineering, University of Manchester, Sackville Street Building, Manchester M13 9PL, UK

<sup>4</sup>Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Institute of Automation, Jiangnan University, Wuxi, 214122, China

**Abstract:** A new online iterative algorithm for solving the  $H_\infty$  control problem of continuous-time Markovian jumping linear systems (MJLSs) is developed. For comparison, an available off-line iterative algorithm for converging to the solution of the  $H_\infty$  control problem is firstly proposed. Based on the off-line iterative algorithm and a new online decoupling technique named *Subsystems Transformation* method, a set of linear subsystems which implementation in parallel are obtained. There, by means of the online Adaptive Dynamic Programming (ADP) technique, the two-player zero-sum game and the coupled game algebraic Riccati equation (CGARE) are solved. The convergence of the novel policy iteration algorithm is also established. At last, simulation results have illustrated the effectiveness and applicability of these two methods.

**Keywords:** Markov jump linear systems (MJLSs);  $H_\infty$  control; Subsystems Transformation; Coupled game algebraic Riccati equation (CGARE); Online

## 1 Introduction

Over the last few years, continuous-time Approximate/Adaptive Dynamic Programming (ADP) approach has been widely applied to the optimal controller design for linear/nonlinear systems (see, for instance [1, 16, 23, 31, 34, 35]). In recent years, this method has been widely used to solve the zero-sum infinite horizon differential games control problems [32, 33, 38]. The greatest advantage of this algorithm is that it can solve the optimal problems without using the systems dynamics [23], and the convergence speed of the online iterative algorithm shows much faster [38].

Markovian jumping linear systems (MJLSs) have attracted much research attention because they may effectively represent a class of plants with abrupt variations in their structures, such as random failures of components,

\*Corresponding author. Tel: +86 0551 63861413, fax: +86 0551 63861413. Email addresses: shuping.he@ahu.edu.cn (S. He), sjky5211@163.com (J. Song), zhengtao.ding@manchester.ac.uk (Z. Ding), fliu@jiangnan.edu.cn (F. Liu).

sudden environmental disturbances, changing subsystems interconnections and abrupt variations in the operating point of a nonlinear plant. Specially, a number of jump systems were modeled by the *continuous-time* MJLSs, which belong to the sort of hybrid systems since the states take values continuously from the real number field and the modes takes discrete values [15]. Some relevant results on the stability, stabilization, and optimal control of continuous-time MJLSs can be found in [5, 7], and the references therein.

As one of the most important robust control methods,  $H_\infty$  control has attracted much attention over the past decades [12, 25], since it can eliminate the effect of disturbance and minimize a desired performance when the worst case disturbance is implemented. It is well known that finding the  $H_\infty$  controller of MJLSs is equivalent to finding the unique positive-definite sequence solution for a set of coupled game algebraic Riccati equation (CGARE) [8]. Parallel Algorithm (PA) with a sequence of Lyapunov algebraic equations has been proposed by Gajic and Borno in [10] to converge to the solution of the coupled algebraic Riccati equations that associated with the  $H_2$  optimal control problem for continuous-time MJLSs. Recently, in [24], an implicit sequential algorithm was developed for solving coupled Lyapunov matrix equations of continuous-time MJLSs. To solve the coupled stochastic Riccati equations, a heuristic iteration algorithm has been proposed for  $H_\infty$  control of stochastic systems in [29].

It should be pointed out that almost all the aforementioned results are proposed by the off-line numerical algorithms, which implicitly assumes that the exactly information of the MJLSs dynamics is known a priori. However, such assumption is no longer true in many practical applications. Motivated by the fact, the main objective of this work is to develop an algorithm for solving  $H_\infty$  control problem for the continuous-time MJLSs without using the information about the system matrices by using continuous-time ADP approach.

At first, we first extend the framework of PA in [4, 10] to an off-line iterative algorithm for solving the CGARE for the  $H_\infty$  control of continuous-time MJLSs. Subsequently, a new decoupling technique called *Subsystem Transformation* [13] is introduced to separate the embedded state-coupled information (*i.e.* transition probabilities associated jumping modes) of MJLSs into a explicit formula. Then, motivated by the work [38], we present a continuous-time ADP algorithm associated with the integral reinforcement learning to find the Nash equilibrium solution for the two-player zero-sum differential game. The new ADP iterative computing algorithm is based on the framework of two-step policy iteration. At last, the new policy iteration algorithm is implemented online by means of the Recursive Least Squares (RLS), and the relevant simulation examples are also given.

**Notation.** In the sequel, if not explicitly stated, matrices are assumed to have compatible dimensions.  $\|\cdot\|$  denotes the Euclidean norm for vectors or the spectral norm of matrices.  $L_2^n[0 N]$  is the space of  $n$ -dimensional square integrable function vector over  $[0 N]$ .  $E\{\cdot\}$  denotes the mathematics statistical expectation of the stochastic process or vector. The superscript "T" represents the transpose.  $vec(A)$  is defined to be the  $mn$ -vector formed by stacking the columns of  $A \in \mathbb{R}^{n \times m}$  on top of one another, *i.e.*  $vec(A) = \begin{bmatrix} a_1^T & a_2^T & \cdots & a_m^T \end{bmatrix}^T$ , where  $a_i \in \mathbb{R}^n$  are the columns of  $A$ .  $\otimes$  represents the Kronecker product.

## 2 Problem Formulation and Off-line Iterative Algorithm

### 2.1 Problem formulation

For a given probability space  $(\Omega, \mathcal{F}, \mathcal{P})$ , where  $\Omega$  is the sample space,  $\mathcal{F}$  is the algebra of events, and  $\mathcal{P}$  is the probability measure defined on  $\mathcal{F}$ . Define the  $\{r(t), t \geq 0\}$  is a continuous-time discrete-state Markovian process with left continuous trajectories [26] and taking values from a finite set  $\mathbf{S} = \{1, 2, \dots, N\}$  with transition

probabilities given by

$$P_r\{r(t + \Delta t) = j | r(t) = i\} = \begin{cases} \pi_{ij}\Delta t + o(\Delta t) & j \neq i, \\ 1 + \pi_{ii}\Delta t + o(\Delta t) & j = i, \end{cases} \quad (1)$$

where  $\Delta t > 0$ ,  $\lim_{\Delta t \rightarrow 0} \frac{o(\Delta t)}{\Delta t} = 0$ , and  $\pi_{ij} \geq 0$  ( $j \neq i$ ) is the transition rate from mode  $i$  at time  $t$  to mode  $j$  at time  $t + \Delta t$  and

$$\pi_{ii} = - \sum_{j \in \mathbf{S}, j \neq i} \pi_{ij}. \quad (2)$$

Consider the following continuous-time MJLSs with external disturbance over the given probability space  $(\Omega, \mathcal{F}, \mathcal{P})$ :

$$\begin{cases} \dot{x}(t) = A(r(t))x(t) + B_1(r(t))d(t) + B_2(r(t))u(t) \\ y(t) = \begin{bmatrix} C(r(t))x(t) \\ D(r(t))u(t) \end{bmatrix} \end{cases} \quad (3)$$

where  $x(t) \in \mathbb{R}^n$  is the system state,  $u(t) \in \mathbb{R}^m$  is the control input,  $d(t) \in \mathbb{R}^p$  is the disturbance input which belongs to  $L_2[0, \infty)$ ,  $y(t) \in \mathbb{R}^q$  is the control output.

The coefficient matrices in MJLSs (1)-(3) are mode-dependent real matrices with appropriate dimensions. To simplify the mathematical notation, when  $r(t) = i$  ( $i \in \mathbf{S}$ ), denoting  $A(r(t))$ ,  $B_1(r(t))$ ,  $B_2(r(t))$ ,  $C(r(t))$  and  $D(r(t))$  as  $A_i$ ,  $B_{1i}$ ,  $B_{2i}$ ,  $C_i$  and  $D_i$ , respectively. In addition, we suppose that the pair  $(A_i, B_{2i})$  is stochastically stabilizable.

Subject to the MJLSs (1)-(3), our aim is to find the following mode-dependent state-feedback controller:

$$u(t) = K_i x(t), \quad (4)$$

such that, for all nonzero  $d(t) \in L_\infty[0, \infty) \cap L_2[0, \infty)$ ,

$$\|y(t)\|_{2,E} \leq \gamma \|d(t)\|_2, \quad (5)$$

where  $\gamma > 0$  is a prescribed level of disturbance attenuation to be achieved and

$$\|y(t)\|_{2,E} = \mathbb{E} \left\{ \int_0^\infty y^T(t)y(t)dt \right\}^{1/2}, \quad \|d(t)\|_2 = \left[ \int_0^\infty d^T(t)d(t)dt \right]^{1/2}.$$

It is well known that the solution to the  $H_\infty$  control problem is the saddle point solution to a two-player zero-sum game, where the control input  $u(t)$  is a minimizing mode-dependent player and the disturbance input  $d(t)$  is a maximizing mode-dependent player [9, 22]. Define an infinite horizon quadratic performance index as

$$V(u, w) = \mathbb{E} \int_0^\infty \left( \|y(t)\|^2 - \gamma^2 \|d(t)\|^2 \right) dt = \mathbb{E} \int_0^\infty \left[ x^T(t)Q_i x(t) + u^T(t)R_i u(t) - \gamma^2 d^T(t)d(t) \right] dt, \quad (6)$$

where  $Q_i = C_i^T C_i$  and  $R_i = D_i^T D_i$ . The goal of the game problem is to find the following Nash equilibrium solution  $(u^*(t), d^*(t))$ :

$$u^*(t) = K_i x(t) = -R_i^{-1} B_{2i}^T P_i x(t), \quad (7)$$

$$d^*(t) = L_i x(t) = \gamma^{-2} B_{1i}^T P_i x(t), \quad (8)$$

where matrix  $P_i^T = P_i > 0$  is the solution of the following CGARE:

$$A_i^T P_i + P_i A_i + \sum_{j=1}^N \pi_{ij} P_j + Q_i + \gamma^{-2} P_i B_{1i} B_{1i}^T P_i - P_i B_{2i} R_i^{-1} B_{2i}^T P_i = 0. \quad (9)$$

Obviously, finding the Nash equilibrium solution to the game problem of MJLSs (1)-(3) is equivalent to solving the above CGARE (9). Generally, in order to solve the CGARE (9), the exact information of the system matrix  $A_i$  is required.

In this work, our primary objective is to remove this limitation by using continuous-time ADP approach. Before proceeding with the study, we will introduce an off-line iterative algorithm for solving the CGARE (9). The off-line iterative algorithm will be used to develop subsystems transformation technique and finally help us to prove the proposed online iteration algorithm.

## 2.2 An off-line iterative algorithm for solving the CGARE

In [10], PA was developed to solve the coupled algebraic Riccati equations of continuous-time MJLSs. In the spirit of this work, we propose a iteration method based on PA to converge to the solutions of the CGARE of the  $H_\infty$  control problem for an MJLSs (1)-(3). Thereafter, we will prove that the PA is essentially the Newton's method. The following Assumption 1 is needed to obtain the off-line algorithm.

**Assumption 1** [10] The triples  $(A_i, B_{2i}, C_i)$ ,  $i \in \mathbf{S}$ , are stabilizable-detectable and

$$\max_{i \in \mathbf{S}} \left\{ \inf_{\Gamma_i} |\lambda_{\max} [ \int_0^\infty e^{\tilde{M}^T t} \times e^{\tilde{M} t} dt ] \right\} < 1, \quad (10)$$

where  $\tilde{M} = A_i + \gamma^{-2} B_{1i} B_{1i}^T \Gamma_i - B_{2i} R_i^{-1} B_{2i}^T \Gamma_i + \frac{\pi_{ii}}{2} I$ ,  $\Gamma_i$  are arbitrary real matrices.

Assumption 1 assumes that there exists a unique stabilizing  $P_i$  of the CGARE (9). The PA is given as follows in terms of decoupled algebraic Lyapunov equations.

**Remark 1.** *It is noticed that the solution of the nonlinear CGARE (9) can be obtained by iteratively solving a sequence of decoupled linear algebraic Lyapunov equations. At each iteration step, the value of  $N$   $Q_i(k)$  (11) are updated and the relevant  $N$  Lyapunov equations (12) are solved. Till now, many numerical algorithms to solve (12) have been imbedded in MATLAB functions, such as, `lyap` [3].*

Next, we will prove that Algorithm 1 converges to the solution of CGARE (9). The proof is motivated by [20]. To this end, the following lemmas for positive operator are introduced:

**Lemma 1.** [2] *An operator  $\mathcal{T} : \mathbb{E} \rightarrow \mathbb{F}$  between two ordered matrix spaces is said to be **positive operator** (in symbols  $\mathcal{T} > 0$ ) if  $\mathcal{T}A > 0, \forall A > 0$ .*

**Lemma 2.** [27] *Let  $\mathcal{T} : \mathbb{W} \rightarrow \mathbb{W}$  be positive operator and  $\rho(\mathcal{T})$  be the spectral radius of  $\mathcal{T}$ . Then, the following statements are equivalent:*

- (i).  $\rho(\mathcal{T}) < 1$ ;
- (ii). *There exists an  $X > 0$  such that  $\mathcal{T}X - X < 0$ .*

Define an operator as follows:

$$\begin{cases} \mathcal{F} : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n} \\ \mathcal{F}(P_i(k)) = [A_i + \frac{\pi_{ii}}{2} I]^T P_i(k) + P_i(k) [A_i + \frac{\pi_{ii}}{2} I] - P_i(k) S_i P_i(k) + \sum_{j=1, j \neq i}^N \pi_{ij} P_j(k) + Q_i \end{cases} \quad (13)$$

Thus, we have the Fréchet differential of  $\mathcal{F}(P_i(k))$  at  $P_i(k)$  is

$$\mathcal{F}'_{P_i(k)}(M) = [A_i + \frac{\pi_{ii}}{2} I - S_i P_i(k)]^T M + M [A_i + \frac{\pi_{ii}}{2} I - S_i P_i(k)], \quad (14)$$

---

**Algorithm 1:** Using PA to solve CGARE (9).

---

**Input:** Give a set of initial stabilizing matrices  $\{P_i(0), i = 1, 2, \dots, N\}$

**Output:** The solution of CGARE (9):  $\{P_i^*, i = 1, 2, \dots, N\}$

$k = 0;$

$P_i(k) \in \mathbb{R}^{n \times n};$

**while**  $k \geq 0$  **do**

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

$$Q_i(k) = Q_i + \sum_{j=1, j \neq i}^N \pi_{ij} P_j(k); \quad (11)$$

        Compute:

$$\begin{aligned} & [A_i + \frac{\pi_{ii}}{2} I - S_i P_i(k)]^T P_i(k+1) + P_i(k+1) [A_i + \frac{\pi_{ii}}{2} I - S_i P_i(k)] \\ & = -P_i(k) S_i P_i(k) - Q_i(k), \end{aligned} \quad (12)$$

        where  $S_i = B_{2i} R_i^{-1} B_{2i}^T - \gamma^{-2} B_{1i} B_{1i}^T;$

$\{P_i(k+1)\};$

**if**  $\max\{\|P_1(k+1) - P_1(k)\|, \|P_2(k+1) - P_2(k)\|, \dots, \|P_N(k+1) - P_N(k)\|\} \geq \epsilon$  **then**

$k = k + 1;$

**else**

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

$P_i^* = P_i(k+1);$

**break;**

return  $\{P_i^*\};$

---

where  $M \in \mathbb{R}^{n \times n}$ . Based upon operators  $\mathcal{F}$  and  $\mathcal{F}'_{P_i(k)}$ , we define another operator in form of:

$$\mathcal{L} = \mathcal{I} - (\mathcal{F}'_{P_i(k)})^{-1} \mathcal{F}. \quad (15)$$

**Lemma 3.** *The proposed Algorithm 1 (PA) can be rewritten as:*

$$P_i(k+1) = \mathcal{L}(P_i(k)). \quad (16)$$

*Proof.* The equation (16) is equivalent to the following equation:

$$\mathcal{F}'_{P_i(k)}(P_i(k+1)) = \mathcal{F}'_{P_i(k)}(P_i(k)) - \mathcal{F}(P_i(k)), \quad (17)$$

By means of (14), we have

$$\begin{aligned} \mathcal{F}'_{P_i(k)}(P_i(k+1)) &= [\bar{A}_i - S_i P_i(k)]^T P_i(k+1) + P_i(k+1) [\bar{A}_i - S_i P_i(k)], \\ \mathcal{F}'_{P_i(k)}(P_i(k)) &= [\bar{A}_i - S_i P_i(k)]^T P_i(k) + P_i(k) [\bar{A}_i - S_i P_i(k)], \end{aligned} \quad (18)$$

where  $\bar{A}_i = A_i + \frac{\pi_{ii}}{2} I$ .

Substituting (13) and (18) into (17), it follows that:

$$\left[ \bar{A}_i - S_i P_i(k) \right]^T P_i(k+1) + P_i(k+1) \left[ \bar{A}_i - S_i P_i(k) \right] = -Q_i - \sum_{j=1, j \neq i}^N \pi_{ij} P_j(k) - P_i(k) S_i P_i(k). \quad (19)$$

Obviously, the above equation is just the iterative (11) and (12) of the Algorithm 1. This completes the proof.  $\square$

**Lemma 4.** *The operator  $\mathcal{L}$  is an positive operator.*

*Proof.* Since  $P_i(k+1)$  and  $P_i(k)$  are positive matrix, we know:  $\mathcal{L}(P_i(k)) > 0$ . According to Lemma 1, it is easy to see that the operator  $\mathcal{L}$  is an positive operator. This completes the proof.  $\square$

Based upon the above four lemmas, we can prove the following result regarding the convergence of Algorithm 1.

**Theorem 1.** *Let  $P_i^*$  ( $i = 1, 2, \dots, N$ ) be the solution of the CGARE (9) and  $\{P_i(k)\}_{k=0}^{\infty}$  be the sequence generated by Algorithm 1 (PA). Then, Algorithm 1 monotonically converges to nonincreasing sequences and  $\lim_{k \rightarrow \infty} P_i(k) = P_i^*$ .*

*Proof.* Along a similar line to [11], one can find a initial *stabilizing* sequence  $\{P_i(0), \forall i = 1, 2, \dots, N\}$  such that it satisfies

$$P_i(0) - P_i(1) > 0, \forall i. \quad (20)$$

Since  $P_i(0)$  must be positive, we get

$$\mathcal{L}(P_i(0)) - P_i(0) = P_i(1) - P_i(0) < 0. \quad (21)$$

By Lemma 2, we know that

$$\rho(\mathcal{L}) < 1. \quad (22)$$



Therefore, Algorithm 1 monotonically converges to nonincreasing sequences by the theorem of monotonic convergence of positive operators [11, 17].

On the other hand, we let  $\lim_{k \rightarrow \infty} P_i(k) = P_i^*$ . Taking limit on the both sides of (19) and considering the relation of  $\lim_{k \rightarrow \infty} P_i(k+1) = \lim_{k \rightarrow \infty} P_i(k)$ , we have

$$A_i^T P_i^* + P_i^* A_i + \sum_{j=1}^N \pi_{ij} P_j^* + Q_i + \gamma^{-2} P_i^* B_{1i} B_{1i}^T P_i^* - P_i^* B_{2i} R_i^{-1} B_{2i}^T P_i^* = 0. \quad (23)$$

Denote  $\Delta P = P_i^* - P_i^*$ . Substituting (23) into (9) with  $P^*$ , it has

$$A_i^T \Delta P_i + \Delta P_i A_i + \sum_{j=1}^N \pi_{ij} \Delta P_j + \gamma^{-2} \Delta P_i B_{1i} B_{1i}^T \Delta P_i - \Delta P_i B_{2i} R_i^{-1} B_{2i}^T \Delta P_i = 0.$$

Recalling  $\rho(\mathcal{L}) < 1$  for  $Q_i \equiv 0$ , it implies that the above equations have a unique solution 0, that is,  $P_i^* = P_i^*$ . This completes the proof. □

**Remark 2.** By using Nash equilibrium strategies, the mixed  $H_2/H_\infty$  control problem for continuous-time stochastic MJLSs with multiplicative noise has been studied in [28] and a heuristic algorithm has been proposed to solve the CAREs in [29]. However, similar to the PA proposed in Algorithm 1, the exact knowledge of the system dynamics must be known a priori for these off-line algorithms. In the following, based on ADP techniques [32, 33, 34], we present a new approach to solve the CGARE (9) without using the system matrices  $A_i$ .

### 3 Online Solving $H_\infty$ Control Problem for MJLSs

Associating with the continuous-time linear/nonlinear systems, the online ADP techniques has been proposed in [32, 33, 34], which are implemented by online measuring the system states  $x(t)$ . The information regarding to the system matrix  $A$  is embedded in the measured system states. However, subject to the continuous-time MJLSs (1)-(3), the coupled relation between working subsystems in different modes can not be known by online measuring the MJLSs states since the system modes take values from a discrete-state set with known transition probabilities. Namely,  $P_j$  ( $j \in \mathbf{S}, j \neq i$ ) need to be known while we find the solution  $P_i$  from the CGARE (9).

In order to deal with the  $H_\infty$  control problem for MJLSs (1)-(3) by means of online ADP techniques, we first introduce a new online decoupling approach, which so-called *Subsystems Transformation* [13]. In the sequel, this technique is used to separate the coupled relation from CGARE (9) based on PA (Algorithm 1) and related results presented in [21, 38].

#### 3.1 Subsystems Transformation based on Algorithm 1

Algorithm 1 shows that the sequence  $\{P_i(k)\}$  generated by iterations (11)-(12) converges to the solution sequence  $\{P_i\}$  of CGARE (9). Subject to the  $i$ th iteration equation in (11), the value of  $Q_i(k)$  need to be reset at each iteration step by Eq. (12) and the coupled information of MJLSs (1)-(3) is contained in  $Q_i(k)$ . Before proceeding the study, the following Lemma 5 is required:

**Lemma 5.** (Off-line SPUA [38]) For the  $H_\infty$  control problem of the linear time-invariant continuous-time systems which described as:

$$\Sigma : \begin{cases} \dot{x}(t) = Ax(t) + B_1w(t) + B_2u(t) \\ y(t) = \begin{bmatrix} Cx(t) \\ Du(t) \end{bmatrix} \end{cases} \quad (24)$$

where  $w(t)$  is the norm-bounded disturbance input. Then, the sequence  $\{\Pi(k)\}$  given in following iteration equation:

$$\begin{aligned} & [A + \rho^{-2}B_1B_1^T\Pi(k) - B_2R^{-1}B_2^T\Pi(k)]^T\Pi(k+1) + \Pi(k+1)[A + \rho^{-2}B_1B_1^T\Pi(k) - B_2R^{-1}B_2^T\Pi(k)] \\ & = -Q + \rho^{-2}\Pi(k)B_1B_1^T\Pi(k) - \Pi(k)B_2R^{-1}B_2^T\Pi(k), \end{aligned} \quad (25)$$

can converge to the unique solution of the following  $H_\infty$  ARE:

$$A^T\Pi + \Pi A + Q + \rho^{-2}\Pi B_1B_1^T\Pi - \Pi B_2R^{-1}B_2^T\Pi = 0 \quad (26)$$

where  $\rho > 0$  is a prescribed level of disturbance attenuation.

Comparing Algorithm 1 with Lemma 5, we can get:

- At each iteration step  $k$ , Algorithm 1 need to calculate  $N$  parallel linear Lyapunov equations and Lemma 5 only computes one linear Lyapunov equation;
- At each iteration step  $k$ , Algorithm 1 need to reset  $Q_i(k)$  (embed the coupled relation into the computation) for every parallel equation but  $Q$  in Lemma 5 is a constant.

Simultaneously, notice that the  $H_\infty$  controller for system  $\Sigma$  in (24) can be obtained from the solution of (26). Based on the above analysis, we give the following Theorem 2 for the  $H_\infty$  control problem of MJLSs (1)-(3) by using a new decoupling method called *Subsystems Transformation*:

**Theorem 2.** Subject to a set of  $N$  linear subsystems  $(\Sigma_1, \Sigma_2, \dots, \Sigma_i, \dots, \Sigma_N)$  which run simultaneously, we can describe  $\Sigma_i$  ( $\forall i = 1, 2, \dots, N$ ) as follows:

$$\begin{cases} \dot{x}_i = (A_i + \frac{\pi_{ii}}{2}I)x_i + B_{1i}d + B_{2i}u, \\ y_i = \begin{bmatrix} C_i x_i \\ D_i u \end{bmatrix}. \end{cases} \quad (27)$$

By utilizing the iteration (25) to solve  $H_\infty$  control problem for subsystems  $\{\Sigma_i\}$  with same disturbance input  $d(t)$  and same disturbance attenuation level  $\gamma$ . And, at each iteration step  $k$ ,  $\{Q_i\}$  is updated by (11). Then, the obtained sequence  $\{\Pi_i(k)\}$  is convergent and

$$\lim_{k \rightarrow \infty} \Pi_i(k) = P_i, \quad (28)$$

where  $P_i$  is the solution of CGARE (9).

*Proof.* Subject to the  $i$ th subsystem, we use Lemma 5 to find the  $H_\infty$  controller gain. Replacing the system matrix  $A$  with  $A_i + \frac{\pi_{ii}}{2}I$  in (25). And updating  $Q$  by (11) at every iteration step. Then it is easy to see that the obtained equation is into (12). According to Theorem 1, it follows that (28). This completes the proof.  $\square$

**Remark 3.** In this work, we separate MJLSs (1)-(3) into  $N$  reconstructed linear subsystems and a explicit coupled relation among these reconstructed linear subsystems is proposed by the *Subsystems Transformation* technique

[13]. Notice that the exact information of system matrices  $A_i$  are contained in the linear subsystems matrices  $A_i + \frac{\pi_{ii}}{2}I$ . Therefore, the subsystems states  $x_i(t)$  which can be observed online contain the information regarding to the system matrix  $A_i$ . It implies that we can develop an online two-step iteration algorithm without using the system matrix  $A_i$ . A new online policy iteration algorithm will be studied in the following Subsection 3.2.

Define the following control input player  $u_i(t)$  and disturbance input player  $d_i(t)$  associated with a two-player zero-sum game for subsystems (27):

$$u_i(t) = K_i(k)x(t) = -R_i^{-1}B_{2i}^T\Pi_i(k)x(t), \quad (29)$$

$$d_i(t) = L_i(k)x(t) = \gamma^{-2}B_{1i}^T\Pi_i(k)x(t). \quad (30)$$

Based on Theorem 2, we give the following Theorem 3:

**Theorem 3.** *If  $\Pi_i(k)$  is one of the obtained sequence  $\{\Pi_i(k)\}$  in Theorem 2 which satisfies (28), the following equations will be held:*

$$\lim_{k \rightarrow \infty} K_i(k) = K_i, \quad (31)$$

$$\lim_{k \rightarrow \infty} L_i(k) = L_i. \quad (32)$$

where  $K_i$  and  $L_i$  are the solution of (7) and (8), respectively.

*Proof.* Combining (28) and the predefined equations (29) and (30), one can get (31) and (32) directly. The proof is completed. □

### 3.2 A novel online policy iteration algorithm for the solution of CGARE

In this subsection, we propose a novel policy iteration algorithm which can be implemented online for learning the solution of CGARE (9) without using the exact knowledge of system matrix  $A_i$  of MJLSs (1)-(3). The novel policy iteration algorithm is based on a two-step iteration framework which named Policy Evaluation and Policy Improvement. The proof of convergence of the new algorithm is also provided.

Based on Theorem 2 and Theorem 3, an online policy iteration algorithm is developed for learning the solution of CGARE (9) as follows:

**Remark 4.** *It can be seen from the Policy Evaluation (33) that the dynamic matrices  $\{A_i\}$  of MJLSs (1)-(3) are not involved. Actually, the information regarding to each subsystem matrix  $A_i$  is embedded in the relevant subsystems state.*

**Remark 5.** *In Algorithm 2, at every iteration step  $k$ , overall  $N$  policy evaluation equations need to be solved and the relevant policy improvement equations also need to be calculated. A remarkable difference between (33), (34) and (35) is that,  $\{\Pi_j(k)\}$  ( $j \in \mathbf{S}, j \neq i$ ) are used to update  $Q_i(k)$  while (33) and (35) only require  $\Pi_i(k)$  to improve the control input player gain  $K_i(k)$  and the disturbance input player gain  $L_i(k)$ .*

**Remark 6.** *From a viewpoint of computation, the differences of our Policy Iteration Algorithm with the existing results proposed for continuous-time linear/nonlinear systems (e.g. [31, 32, 33, 34, 38]), are mainly focused on*

- *In Algorithm 2, Policy Evaluation step is solved by utilizing some discrete data while sampling the constructed linear subsystems state trajectories. But in [34], the used data is direct sampled on original system state trajectories;*

---

**Algorithm 2:** Using online policy iteration algorithm to solve CGARE (9).

---

**Input:** Give initial matrices  $\{\Pi_i(0)\}$ ,  $\{K_i(0)\}$  and  $\{L_i(0)\}$  ( $i = 1, 2, \dots, N$ )

**Output:** Optimal controller gain  $\{K_i^*\}$  and optimal disturbance controller gain  $\{L_i^*\}$

$k = 0$ ;  $Q_i(0) = Q_i + \Pi_i(0)$ ;

**while**  $k \geq 0$  **do**

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

    Online solve: **(Policy Evaluation)**

$$\left\{ \begin{array}{l} x_{1,t}^T \Pi_1(k+1) x_{1,t} = x_{1,t+T}^T \Pi_1(k+1) x_{1,t+T} \\ \quad + \int_t^{t+T} x_{1,\tau}^T [Q_1(k) + K_1^T(k) R_1 K_1(k) - \gamma^2 L_1^T(k) L_1(k)] x_{1,\tau} d\tau, \\ \vdots \\ x_{i,t}^T \Pi_i(k+1) x_{i,t} = x_{i,t+T}^T \Pi_i(k+1) x_{i,t+T} \\ \quad + \int_t^{t+T} x_{i,\tau}^T [Q_i(k) + K_i^T(k) R_i K_i(k) - \gamma^2 L_i^T(k) L_i(k)] x_{i,\tau} d\tau, \\ \vdots \\ x_{N,t}^T \Pi_N(k+1) x_{N,t} = x_{N,t+T}^T \Pi_N(k+1) x_{N,t+T} \\ \quad + \int_t^{t+T} x_{N,\tau}^T [Q_N(k) + K_N^T(k) R_N K_N(k) - \gamma^2 L_N^T(k) L_N(k)] x_{N,\tau} d\tau, \end{array} \right. \quad (33)$$

    where  $x_{i,t}$  is used to denote the states of the corresponding subsystems (27) at the moment of  $t$ ;

  Obtain  $\{\Pi_i(k+1)\}$ ;

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

    Coupling Relation Update:

$$Q_i(k+1) = Q_i + \sum_{j=1, j \neq i}^N \pi_{ij} \Pi_j(k+1) \quad (34)$$

  Get  $\{Q_i(k+1)\}$ ;

**if**  $\max\{\|\Pi_1(k+1) - \Pi_1(k)\|, \|\Pi_2(k+1) - \Pi_2(k)\|, \dots, \|\Pi_N(k+1) - \Pi_N(k)\|\} \geq \epsilon$  **then**

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

      Compute: **(Policy Improvement)**

$$\left\{ \begin{array}{l} K_i(k+1) = -R_i^{-1} B_{2i}^T \Pi_i(k); \\ L_i(k+1) = \gamma^2 B_{1i}^T \Pi_i(k); \end{array} \right. \quad (35)$$

    Obtain  $\{K_i(k+1)\}$  and  $\{L_i(k+1)\}$ ;

    Refresh  $k = k + 1$ ;

**else**

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

$$\left[ \begin{array}{l} \Pi_i^* = \Pi_i(k+1); \\ K_i^* = -R_i^{-1} B_{2i}^T \Pi_i^*; \\ L_i^* = \gamma^2 B_{1i}^T \Pi_i^*; \end{array} \right.$$

    break;

  return  $\{K_i^*\}$  and  $\{L_i^*\}$ ;

---

- In Algorithm 2, at every iteration step, overall the  $4N$  equations need to be computed. But only 2 equations were computed at each iteration step, see [32, 34, 38] for example;
- At Policy Improvement step of Algorithm 2, we need to consider the coupled relation formula (28). However, the Policy Improvement step presented in [31] did not contain this equation.

In fact, the above differences are determined by the fact that MJLSs are stochastic hybrid systems.

Then, we will propose a theorem to prove the convergence of Algorithm 2:

**Theorem 4.** The sequence  $\{\Pi_i(k)\}$  generated by Algorithm 2 converges to the solution  $P_i$  of CGARE (9), that is,  $\lim_{k \rightarrow \infty} \Pi_i(k) = P_i$ .

*Proof.* Let  $k = 0$ , and choose a stabilizing initial conditions  $\{\Pi_i(0)\}$ ,  $\{K_i(0)\}$  and  $\{L_i(0)\}$  ( $i = 1, 2, \dots, N$ ). By Policy Evaluation (34), we have

$$x_{i,t}^T \Pi_i(1) x_{i,t} = x_{i,t+T}^T \Pi_i(1) x_{i,t+T} + \int_t^{t+T} x_{i,\tau}^T \left[ Q_i(0) + K_i^T(0) R_i K_i(0) - \gamma^2 L_i^T(0) L_i(0) \right] x_{i,\tau} d\tau \quad (36)$$

Solving (36) by using the discrete data which sampled on subsystem (27), one can obtain a set of solutions  $\Pi_i(1)$ .

On the other hand, the solutions of (36) have been proven is equivalent to the solution of the following equation [34, 38]:

$$\left[ A_i + \frac{\pi_{ii}}{2} I - S_i \Pi_i(0) \right]^T \Pi_i(1) + \Pi_i(1) \left[ A_i + \frac{\pi_{ii}}{2} I - S_i \Pi_i(0) \right] = -Q_i - \sum_{j=1, j \neq i}^N \pi_{ij} \Pi_j(0) - \Pi_i(0) S_i \Pi_i(0), \quad (37)$$

Obviously, (37) leads to (12) by letting  $k = 0$  in Algorithm 1.

Performing the same procedure, we can get that, at every iteration step  $k$ , the solution  $\Pi_i(k)$  of Algorithm 2 is equivalent to the corresponding  $k$ th iteration solution of Algorithm 1. According to the Theorem 2, we can conclude that the sequence  $\Pi_i(k)$  generated by Algorithm 2 converges to the solution of CGARE (9). This completes the proof.  $\square$

### 3.3 Online implementation of Algorithm 2

In this subsection, we derive the implement approach for Algorithm 2 based on RLS [6, 34]. With respect to the  $i$ th equation in (33), the following relation by Kronecker product representation, we obtain:

$$(x_{i,t}^T \otimes x_{i,t}^T - x_{i,t+T}^T \otimes x_{i,t+T}^T) \text{vec}(\Pi_i(k+1)) = \int_t^{t+T} x_{i,\tau}^T [Q_i(k) + K_i^T(k) R_i K_i(k) - \gamma^2 L_i^T(k) L_i(k)] x_{i,\tau} d\tau, \quad (38)$$

Considering  $\Pi_i = [\Lambda_{lr}]_i \in \mathbb{R}^{n \times n}$  is positive-definite symmetric matrix, we define the following two vectors:

$$\hat{x}_i = [x_{i,1}^2, x_{i,1} x_{i,2}, \dots, x_{i,1} x_{i,n}, x_{i,2}^2, x_{i,2} x_{i,3}, \dots, x_{i,n}^2] \in \mathbb{R}^{1 \times \frac{1}{2}n(n+1)}, \quad (39)$$

$$\hat{\Pi}_i = [\Lambda_{i,11}, 2\Lambda_{i,12}, \dots, 2\Lambda_{i,1n}, \Lambda_{i,22}, 2\Lambda_{i,23}, \dots, \Lambda_{i,nn}]^T \in \mathbb{R}^{\frac{1}{2}n(n+1) \times 1}. \quad (40)$$

Using (39) and (40), then (38) can be rewritten as

$$(\hat{x}_{i,t} - \hat{x}_{i,t+T}) \hat{\Pi}_i(k+1) = \int_t^{t+T} x_{i,\tau}^T [Q_i(k) + K_i^T(k) R_i K_i(k) - \gamma^2 L_i^T(k) L_i(k)] x_{i,\tau} d\tau, \quad (41)$$

Notice that the symmetric matrix  $\Pi_i(k)$  has  $n(n+1)/2$  unknown independent elements. Hence, there require at least  $M \geq n(n+1)/2$  independent equations to get  $\Pi_i(k+1)$ . Then, we should sample  $M$  state vectors in each time interval  $T$  from the linear subsystems  $\Sigma_i$ . The relevant  $\Pi_i(k+1)$  can be found by the following RLS sense:

$$\hat{\Pi}_i(k+1) = [X_i^T X_i]^{-1} X_i^T Y_i, \quad (42)$$

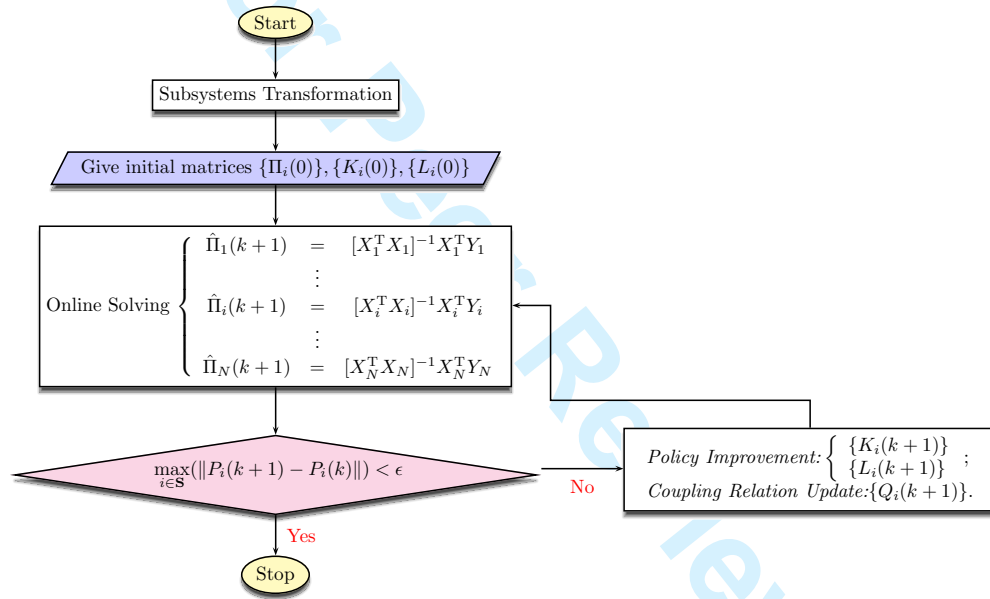
where

$$X_i = \begin{bmatrix} \hat{x}_i(t) - \hat{x}_i(t + \delta t) & \cdots & \hat{x}_i(t + (M-1)\delta t) - \hat{x}_i(t + M\delta t) \end{bmatrix}^T \in \mathbb{R}^{M \times \frac{1}{2}n(n+1)}, \quad (43)$$

$$Y_i = \begin{bmatrix} y_i^{(0)}(k) & y_i^{(1)}(k) & \cdots & y_i^{(M-1)}(k) \end{bmatrix}^T \in \mathbb{R}^{M \times 1}, \quad (44)$$

and with  $\delta t = T/M$  and  $y_i^{(r)}(k) = \int_{t+r\delta t}^{t+(r+1)\delta t} x_{i,\tau}^T [Q_i(k) + K_i^T(k)R_iK_i(k) - \gamma^2 L_i^T(k)L_i(k)] x_{i,\tau} d\tau$  ( $r = 0, 1, \dots, M-1$ ).

Figure 1: The flowchart of the new online policy iteration algorithm.



Rearrange the policy evaluation (33) of Algorithm 2 based upon online implementation approach (42), then we have a online iterative algorithm associates to solve the  $H_\infty$  control problem for continuous-time MJLS (1)-(3) without using the exact knowledge of  $A_i$  ( $\forall i$ ). The flowchart of the new online iteration algorithm is shown in Figure 1.

**Remark 7.** In this work, a Subsystems Transformation-based online policy iteration algorithm has been proposed to solve the  $H_\infty$  optimal control problem for continuous-time MJLSs without using the information on system matrices  $A_i$ , which shows significantly different from some existing approaches, e.g. [14]. However, the results obtained here for MJSs are intrinsically conservative due to constant transition rates and thus have many limitations in applications [30]. Unlike the MJSs, semi-Markov jump systems (semi-MJSs) have much broader applications owing to their relaxed conditions on the probability distributions [18, 19]. Therefore, it is very meaningful to extend the results in this work to semi-MJSs in further research.

**Remark 8.** *It is worth to mention that the discrete-time Markov chain can be effectively used to model the various network-induced phenomena in networked control systems (NCSs) so as to the concerned system can be rewritten as a finite dimensional discrete-time MJLS [30]. Recently, the problem of adaptive optimal control of unknown discrete-time nonlinear MJLSs has regained some research interest [40]. Nevertheless, the  $H_\infty$  optimal control problem for NCSs by using discrete-time ADP approaches [37] has attracted little research attention, not to mention the case with randomly occurring incomplete information [36, 39]. These issues deserve further research.*

## 4 Simulation Example

In this section, we study the  $H_\infty$  controller design problem for a fourth-order continuous-time MJLSs by using both off-line method (Algorithm 1: off-line iterative algorithm) and online method (Algorithm 2: a new online policy iteration algorithm). Comparing with the computing results and the observing simulation figures, one can see that the both algorithms are effective while finding the optimal  $H_\infty$  controller for the following given MJLSs.

The matrix coefficients of the MJLSs are directly taken from [10] and presented as follows

$$A_1 = \begin{bmatrix} -2.1051 & -1.1648 & 0.9347 & 0.5194 \\ -0.0807 & -2.8949 & 0.3835 & 0.8310 \\ 0.6914 & 10.5940 & -36.8199 & 3.8560 \\ 1.0692 & 13.4230 & 22.1185 & -13.1801 \end{bmatrix}, B_{21} = \begin{bmatrix} 0.7564 \\ 0.9910 \\ 9.8255 \\ 7.2266 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} -2.6430 & -1.2497 & 0.5269 & 0.6539 \\ -0.7910 & -2.8570 & 0.0920 & 0.4160 \\ 21.0357 & 22.8659 & -26.4655 & -1.7214 \\ 27.3096 & 7.8736 & -3.8604 & -29.5345 \end{bmatrix}, B_{22} = \begin{bmatrix} 0.3653 \\ 0.2470 \\ 7.5336 \\ 6.5152 \end{bmatrix},$$

$$Q_1 = Q_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \Pi = \begin{bmatrix} -2 & 2 \\ 1.5 & -1.5 \end{bmatrix}, R_1 = R_2 = 1.$$

Among them, we assume that  $C_1 = C_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix}$  and  $D_1 = D_2 = 1$ . In addition, as for the  $H_\infty$  control problem in form of (1)-(3), we set the disturbance attenuation level  $\gamma = 1.5$  and the disturbance input matrices  $B_{11} = B_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$ .

### 4.1 Using Algorithm 1 to solve the $H_\infty$ control problem

While the exact information of system matrix  $A_i$  is known, we use Algorithm 1 (proposed in Section 2.2) to solve the control problem by the parallel-iterative idea of mathematics of computation. The practical stopping criterion of Algorithm 1 is defined as

$$\max \left\{ \|P_1(k+1) - P_1(k)\|, \|P_2(k+1) - P_2(k)\| \right\} < \epsilon, \quad (45)$$

where  $\epsilon$  is the iterative error of Algorithm 1.

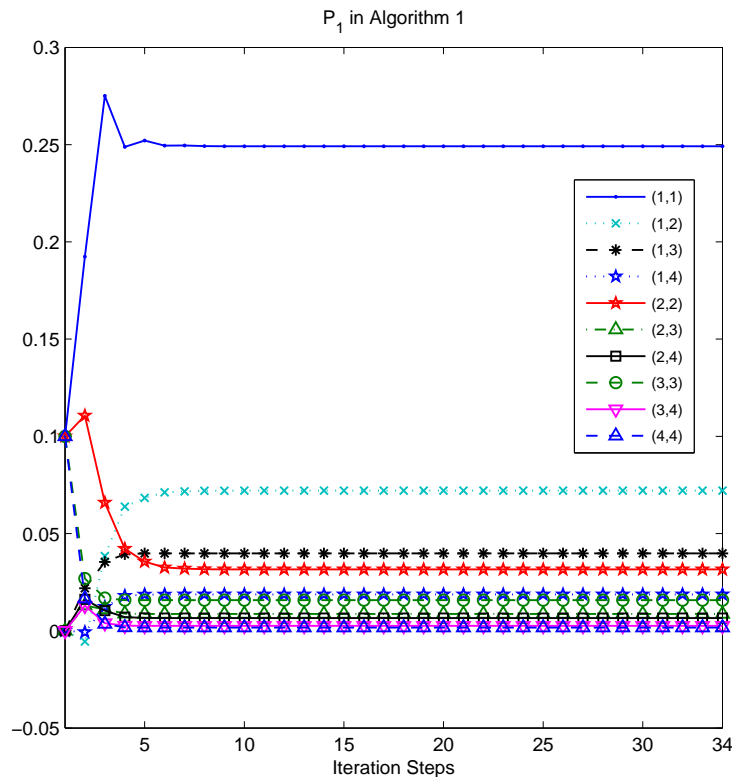
Letting  $\epsilon = 10^{-15}$  and the initial matrices  $P_1(1) = P_2(1) = 0.1I$ , we obtain the following results after 34 iteration steps:

$$P_1^* = P_1(34) = \begin{bmatrix} 0.2492 & 0.0721 & 0.0398 & 0.0187 \\ 0.0721 & 0.0315 & 0.0087 & 0.0066 \\ 0.0398 & 0.0087 & 0.0157 & 0.0026 \\ 0.0187 & 0.0066 & 0.0026 & 0.0017 \end{bmatrix},$$

$$P_2^* = P_2(34) = \begin{bmatrix} 0.5272 & 0.1376 & 0.0525 & 0.0103 \\ 0.1376 & 0.0495 & 0.0139 & 0.0027 \\ 0.0525 & 0.0139 & 0.0193 & 2.4656 \times 10^{-4} \\ 0.0103 & 0.0027 & 2.4656 \times 10^{-4} & 2.8652 \times 10^{-4} \end{bmatrix}.$$

Moreover, we have  $\|P_1(34) - P_1(33)\| = 3.4591 \times 10^{-16}$ , and  $\|P_2(34) - P_2(33)\| = 4.3843 \times 10^{-16}$ . Figures 2 and 3 shows the convergence of the Algorithm 1.

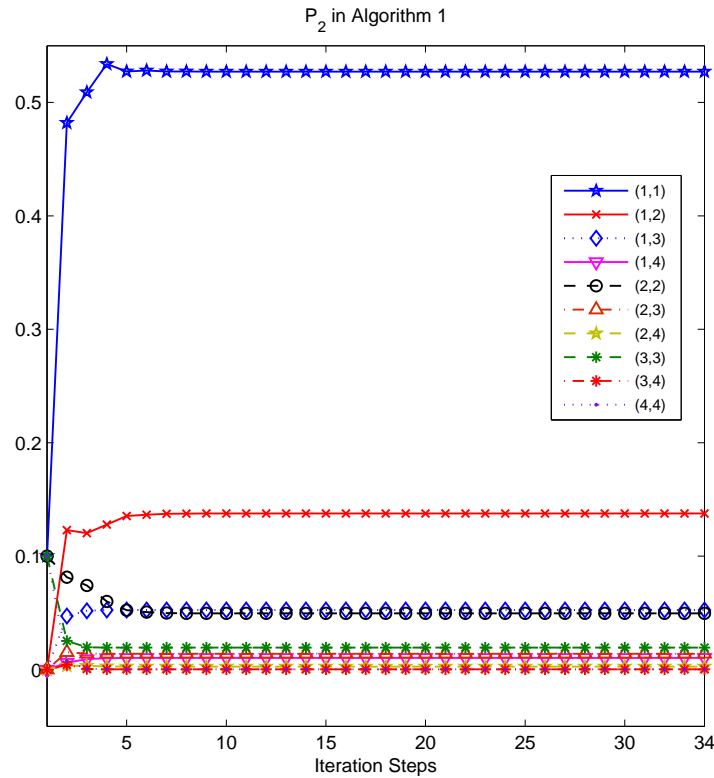
Figure 2: The convergence of  $P_1$  in Algorithm 1.



## 4.2 Using Algorithm 2 to solve the $H_\infty$ control problem with partially unknown dynamics

The solution  $P_i$  of the CGARE (9) associated with this forth-order MJLSs has 10 independent elements. Therefore, one needs to sample at least 10 measurements of the subsystems states at each update time interval  $T$ .



Figure 3: The convergence of  $P_2$  in Algorithm 1.

In this note, we select the sample time interval  $\delta t = 0.03$ , and at each RLS problem (42), 20 sample datas are utilized, i.e. the  $H_\infty$  controller is updated every 0.6.

We start the new online iterative algorithm with the initial symmetric matrices  $P_1(0) = P_2(0) = 0.2I$ , the initial controller gains  $K_1(0) = K_2(0) = 0$ , and the initial disturbance input controller gains  $L_1(0) = L_2(0) = 0I$ . Set both the subsystems initial states as  $x_0 = \begin{bmatrix} 0.1 & -0.1 & 0.1 & -0.1 \end{bmatrix}^T$ . At the first iterative step of Algorithm 2, the following results is obtained:

$$P_1(1) = \begin{bmatrix} 0.2217 & -0.0283 & 0.0189 & -0.0043 \\ -0.0283 & 0.1754 & 0.0182 & 0.0253 \\ 0.0189 & 0.0182 & 0.0497 & 0.0313 \\ -0.0043 & 0.0253 & 0.0313 & 0.0369 \end{bmatrix},$$

$$P_2(1) = \begin{bmatrix} 0.5650 & 0.1533 & 0.0507 & 0.0121 \\ 0.1533 & 0.1358 & 0.0209 & 0.0085 \\ 0.0507 & 0.0209 & 0.0405 & 0.0157 \\ 0.0121 & 0.0085 & 0.0157 & 0.0212 \end{bmatrix}.$$

According to the above digital, a set of  $\gamma$ -suboptimal  $H_\infty$  state-feedback controller gains for the concerned MJLSs can be found:

$$K_1(1) = \begin{bmatrix} -0.2943 & -0.5141 & -0.7469 & -0.5960 \end{bmatrix},$$

$$K_2(1) = \begin{bmatrix} -0.7050 & -0.3024 & -0.4311 & -0.2629 \end{bmatrix}.$$

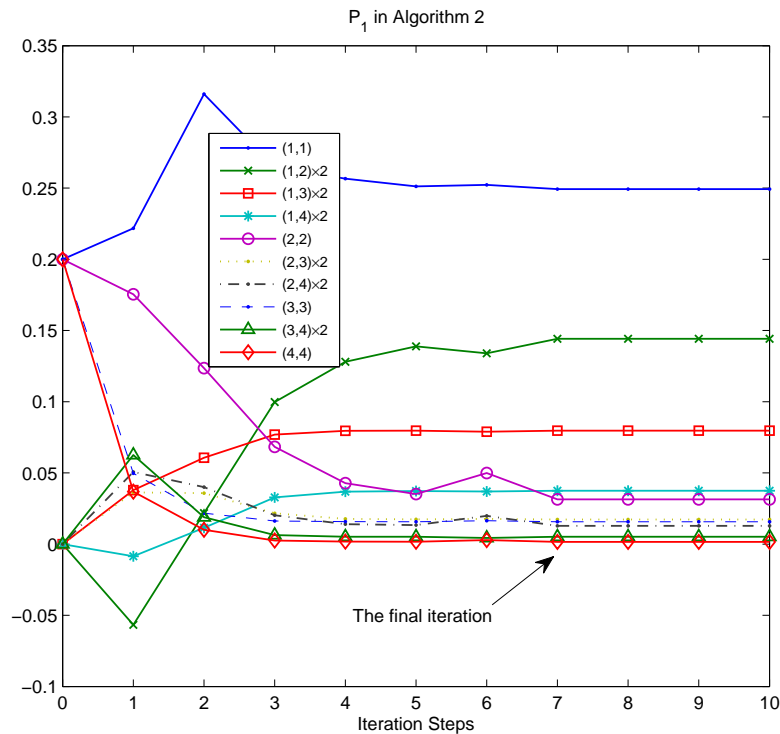
Keeping learning the the solution of the CGARE online by means of Algorithm 2, after seven updates, a set of optimal solutions  $P_i$  for the CGARE (9) can be obtained as

$$P_1(7) = \begin{bmatrix} 0.2493 & 0.0721 & 0.0398 & 0.0187 \\ 0.0721 & 0.0314 & 0.0087 & 0.0064 \\ 0.0398 & 0.0087 & 0.0157 & 0.0026 \\ 0.0187 & 0.0064 & 0.0026 & 0.0016 \end{bmatrix},$$

$$P_2(7) = \begin{bmatrix} 0.5280 & 0.1358 & 0.0528 & 0.0102 \\ 0.1358 & 0.0529 & 0.0147 & 0.0026 \\ 0.0528 & 0.0147 & 0.0189 & 0.0005 \\ 0.0102 & 0.0026 & 0.0005 & 0.0002 \end{bmatrix}.$$

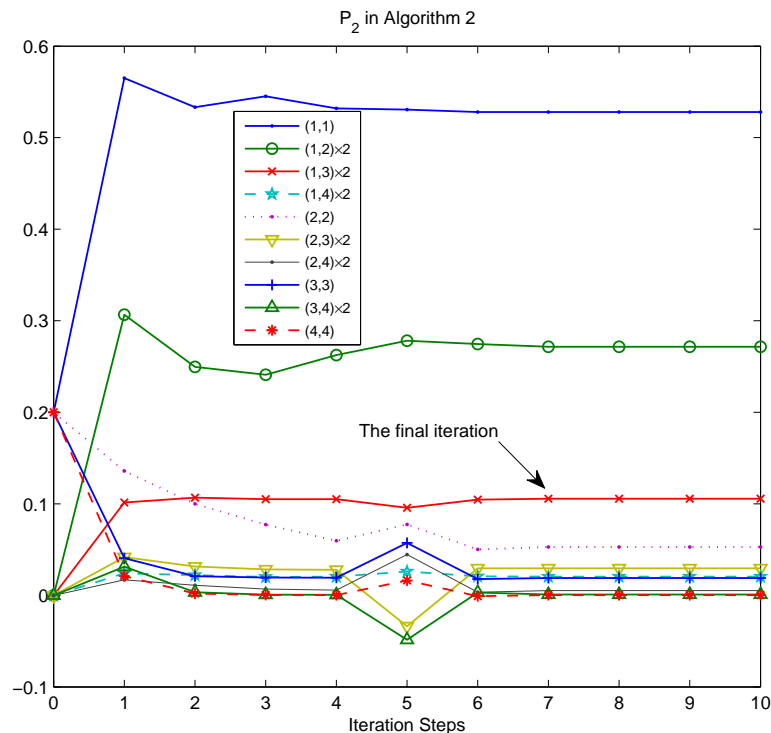
The update histories of matrices  $P_1(k)$  and  $P_2(k)$  in Algorithm 2 are shown in Figures 4 and 5, respectively. From the two figures, it can be seen that Algorithm 2 converges to the solution of the CGARE efficiently with a precision  $\varepsilon = 0.005$ , that is,  $\max\{\|P_1(7) - P_1^*\|, \|P_2(7) - P_2^*\|\} < \varepsilon$ .

Figure 4: The convergence of  $P_1$  in Algorithm 2.



### 4.3 Comparisons and Discussions

From the simulation results shown in Figures 2-5, it is found that Algorithm 2 is better than Algorithm 1 since the less iteration steps are required in Algorithm 2 and Algorithm 2 does not need the exact information

Figure 5: The convergence of  $P_2$  in Algorithm 2.

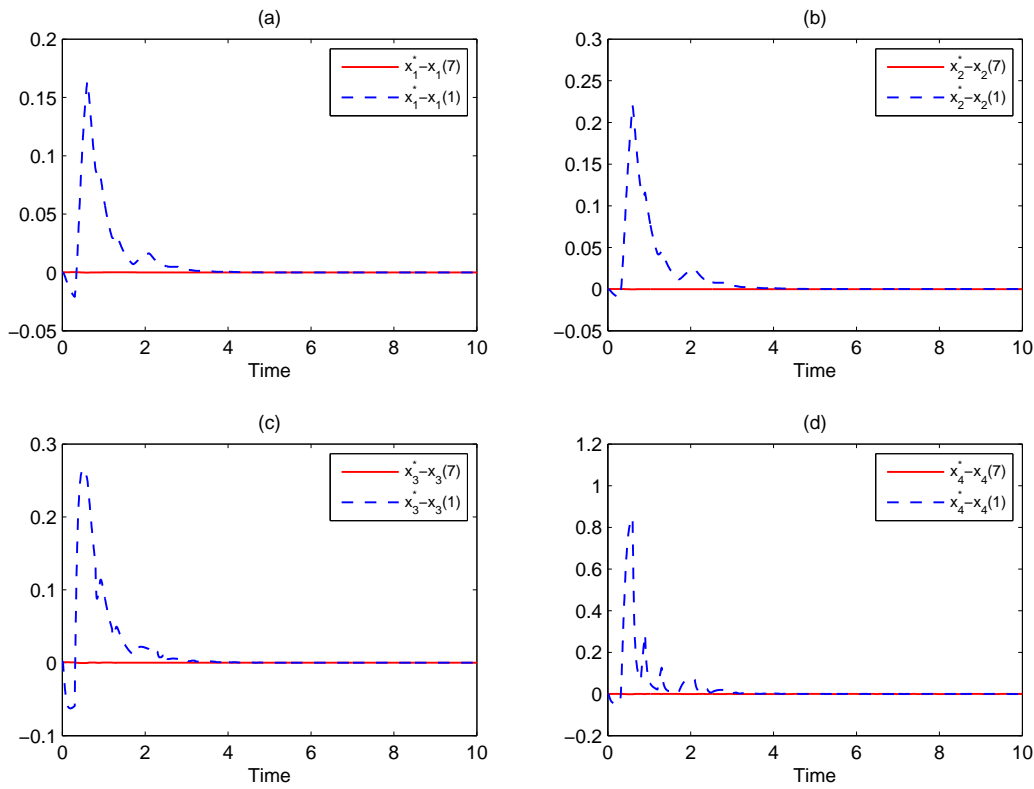
of system matrices  $A_i$ . Nevertheless, based on the same MATLAB platform, the computer operating time for Algorithm 2 is 46.340571sec which is significantly higher than 0.065051sec of Algorithm 1. This is because of there exist overall  $N \times M$  integral operations (e.g. ode45, ode23) and  $N$  RLS for each iteration step in Algorithm 2. In contrast, there only exist overall  $N$  algebraic operations in each step of Algorithm 1.

With the disturbance input  $d(t) = 0.5e^{-3t} \sin t$  and the initial condition  $x(0) = [-2 \ 1 \ -1 \ 1]^T$ , Figure 6 shows the state errors of the continuous-time MJLSs in a simulation experiment, in which  $x^*$  means the optimal controller  $K_i^*$  in Algorithm 1 is applied,  $x_i^*(1)$  the suboptimal controller  $K_i(1)$  in Algorithm 2 is applied, and  $x_i^*(7)$  means the final controller  $K_i(7)$  in Algorithm 2 is applied. The simulation results have illustrated the effectiveness of the proposed algorithms and approaches.

## 5 Conclusions

In this paper, a new online policy iteration algorithm has been developed to solve the  $H_\infty$  control problem of continuous-time MJLSs without using the exact knowledge of the systems matrices  $A_i$ . Based on a proposed off-line iteration algorithm and a new online decoupling technique called *Subsystems Transformation* method, we constructed a set of new coupled parallel-running linear subsystems. Then, we utilized a continuous-time ADP approach to solve the two-player zero-sum game of overall  $N$  linear subsystems based on online implementation. The proof for the convergence of the novel policy iteration algorithm was also given. Simulation results have illustrated the effectiveness and applicability of the proposed off-line iterative algorithm and online iterative algorithm.

Figure 6: A simulation experiment for the state responses in different controllers applied .



## 6 Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61203051), and the Foundation for Distinguished Young Scholars of Anhui Province (No. 1608085J05).

## References

- [1] A. AI-Tamimi and F.L. Lewis. Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(4):943–949, 2008.
- [2] C.D. Aliprantis and O. Burkinshaw. *Positive operators*. New York: Academic Press, 1985.
- [3] R.H. Bartels and G.W. Stewart. Solution of the matrix equation  $AX + BX = C$ . *Communication of ACM*, 15(9):820–826, 1972.
- [4] I. Borno. Parallel computation of the solutions of coupled algebraic Lyapunov equations. *Automatica*, 31(9):1345–1347, 1995.
- [5] E.K. Boukas. *Stochastic Switching Systems: Analysis and Design*. Birkhauser, 2005.

- 1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60
- [6] S.J. Bradtke, B.E. Ydstie, and A.G. Barto. Adaptive linear quadratic control using policy iteration. *Proceedings of the American Control Conference*, 3:3475–3479, 1994.
  - [7] O.L.V. Costa, J.B.R. do Val, and J.C. Geromel. Continuous-time state-feedback  $H_2$ -control of Markovian jump linear systems via convex analysis. *Automatica*, 35(2):259–268, 1999.
  - [8] V. Dragan and I. Ivanov. A numerical procedure to compute the stabilising solution of game theoretic Riccati equations of stochastic control. *International Journal of Control*, 84(4):783–800, 2011.
  - [9] Y. Feng and B.D.O. Anderson. An iteration algorithm to solve state-perturbed stochastic algebraic Riccati equations in LQ zero-sum games. *Systems and Control Letters*, 59(1):50–56, 2010.
  - [10] Z. Gajic and I. Borno. Lyapunov iterations for optimal control of jump linear systems at steady state. *IEEE Transactions on Automatic Control*, 40(11):1971–1975, 1995.
  - [11] Z. Gajic and R. Losada. Monotonicity of algebraic Lyapunov iterations for optimal control of jump parameter linear systems. *System and Control Letters*, 41(3):175–181, 2000.
  - [12] S. He and F. Liu. Finite-time  $H_\infty$  fuzzy control of nonlinear jump systems with time delays via dynamic observer-based state feedback. *IEEE Transactions on Fuzzy systems*, 20(4):605–614, 2012.
  - [13] 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 and Applications*, 9(10):1536–1543, 2015.
  - [14] H. Huang, F. Long, and C. Li. Stabilization for a class of Markovian jump linear systems with linear fractional uncertainties. *International Journal of Innovative Computing, Information and Control*, 11(1):295–307, 2015.
  - [15] Y.D. Ji and H.J. Chizeck. Controllability, stabilizability, and continuous-time Markovian jump linear quadratic control. *IEEE Transactions on Automatic Control*, 35(7):777–788, 1990.
  - [16] Y. Jiang and Z. Jiang. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamic. *Automatica*, 48(10):2699–2704, 2012.
  - [17] L. Kantorovich and G. Akilov. *Functional analysis in normed spaces*. New York: Macmillan, 1964.
  - [18] F. Li, P. Shi, L. Wu, M.V. Basin, and C.C. Lim. Quantized control design for cognitive radio networks modeled as nonlinear semi-Markovian jump systems. *IEEE Transactions on Industrial Electronics*, 62(4):2330–2340, 2015.
  - [19] F. Li, L. Wu, P. Shi, and C.C. Lim. State estimation and sliding mode control for semi-Markovian jump systems with mismatched uncertainties. *Automatica*, 51:385–393, 2015.
  - [20] Z. Li, B. Zhou, J. Lam, and Y. Wang. Positive operator based iterative algorithms for solving Lyapunov equations for Itô stochastic systems with Markovian jumps. *Applied Mathematics and Computation*, 217(21):8179–8195, 2011.
  - [21] B. Luo and H. Wu. Computationally efficient simultaneous policy update algorithm for nonlinear  $H_\infty$  state feedback control with Galerkin’s method. *International Journal of Robust and Nonlinear Control*, 23(9):991–1012, 2013.

- 1  
2  
3 [22] L. Mou and J. Yong. Two-person zero-sum linear quadratic stochastic differential games by a Hilbert space  
4 method. *Journal of Industrial and Management Optimization*, 2(1):93–115, 2006.  
5  
6 [23] J.J. Murray, C.J. Cox, G.G. Lendaris, and R. Saeks. Adaptive dynamic programming. *IEEE Transactions*  
7 *on Systems, Man and Cybernetics*, 32(2):140–153, 2002.  
8  
9 [24] Y. Qian and W. Pang. An implicit sequential algorithm for solving coupled lyapunov equations of continuous-  
10 time Markovian jump systems. *Automatica*, 60:245–250, 2015.  
11  
12 [25] M.A. Rami and E.El. Ghaoui. Robust state-feedback stabilization of jump linear systems via *LMI*s. *Inter-*  
13 *national Journal of Robust Nonlinear Control*, 6(910):1015–1022, 1996.  
14  
15 [26] V. Robinson and D. Sworder. A computational algorithm for design of regulators for linear jump parameter  
16 systems. *IEEE Transactions on Automatic Control*, 19(1):47–49, 1974.  
17  
18 [27] H. Schneider. Positive operators and an inertia theorem. *Numerische Mathematik*, 7(1):11–17, 1965.  
19  
20 [28] L. Sheng, W. Zhang, and M. Gao. Relationship between Nash equilibrium strategies and  $H_2/H_\infty$  control  
21 of stochastic Markov jump systems with multiplicative noise. *IEEE Transactions on Automatic Control*,  
22 59(9):2592–2597, 2014.  
23  
24 [29] L. Sheng, W. Zhang, and M. Gao. Some remarks on infinite horizon stochastic  $H_2/H_\infty$  control with  $(x, u, v)$ -  
25 dependent noise and Markov jumps. *Journal of the Franklin Institute*, 352(10):3929–3946, 2015.  
26  
27 [30] P. Shi and F. Li. A survey on Markovian jump systems: Modeling and design. *International Journal of*  
28 *Control, Automation, and Systems*, 13(1):1–16, 2015.  
29  
30 [31] K.G. Vamvoudakis and F.L. Lewis. Online actor-critic algorithm to solve the continuous-time infinite horizon  
31 optimal control problem. *Automatica*, 46(5):878–888, 2010.  
32  
33 [32] K.G. Vamvoudakis and F.L. Lewis. Online solution of nonlinear two-player zero-sum games using synchronous  
34 policy iteration. *International Journal of Robust and Nonlinear Control*, 22(13):1460–1483, 2012.  
35  
36 [33] D. Vrabie and F.L. Lewis. Adaptive dynamic programming for online solution of a zero-sum differential  
37 game. *Journal of Control Theory and Applications*, 9(3):353–360, 2011.  
38  
39 [34] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F.L. Lewis. Adaptive optimal control for continuous-time  
40 linear systems based on policy iteration. *Automatica*, 45(2):477–484, 2009.  
41  
42 [35] F. Wang, N. Jin, D. Liu, and Q. Wei. Adaptive dynamic programming for finite-horizon optimal control of  
43 discrete-time nonlinear systems with  $\varepsilon$ -error bound. *IEEE Transactions on Neural Networks*, 22(1):24–26,  
44 2011.  
45  
46 [36] Z. Wang, Y. Niu, and B. Shen. Analysis and control with randomly occurring incomplete information.  
47 *International Journal of Systems Science*, 45(7):1333–1336, 2014.  
48  
49 [37] Q. Wei, D. Liu, and H. Lin. Value iteration adaptive dynamic programming for optimal control of discrete-  
50 time nonlinear systems. *IEEE Transactions on Cybernetics*, DOI: 10.1109/TCYB.2015.2492242, 2015.  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

- 1  
2  
3 [38] H. Wu and B. Luo. Simultaneous policy update algorithms for learning the solution of linear continuous-time  
4  $H_\infty$  state feedback control. *Information Sciences*, 222:472–485, 2013.  
5  
6  
7 [39] Z. Wu, P. Shi, H. Su, and J. Chu. Asynchronous  $L_2$ - $L_\infty$  filtering for discrete-time stochastic Markov jump  
8 systems with randomly occurred sensor nonlinearities. *Automatica*, 50(1):180–186, 2014.  
9  
10 [40] X. Zhong, H. He, H. Zhang, and Z. Wang. Optimal control for unknown discrete-time nonlinear Markov  
11 jump systems using adaptive dynamic programming. *IEEE Transactions on Neural Networks and Learning*  
12 *Systems*, 25(12):2141–2155, 2014.  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

For Peer Review