Neurocomputing 190 (2016) 165–171

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Event-based H_∞ control for discrete Markov jump systems ☆

Anke Xue [a], Huijiao Wang [b,1], Renquan Lu [a,c]

[a] Institute of Information and Control, Hangzhou Dianzi University, Hangzhou 310018, PR China
[b] Institute of Automation, Faculty of Mechanical Engineering and Automation, Zhejiang Sci-Tech University, Hangzhou 310018, PR China
[c] School of Automation, Guangdong University of Technology, Guangdong Key Laboratory of IoT Information Processing, Guangzhou 510006, PR China

ARTICLE INFO

Article history:
Received 13 September 2015
Received in revised form
6 November 2015
Accepted 7 January 2016
Communicated by Guang Wu Zheng
Available online 31 January 2016

Keywords:
Event-based
Discrete Markov jump systems
H_∞ control
Network-induced delays

ABSTRACT

The problem of event-based H_∞ control for discrete Markov jump systems is investigated in this paper. A time interval analysis approach is used to transform the system, the event-triggered scheme and network-induced delays into a time-delay system. Based on the time-delay system analysis method, criteria are derived to guarantee the discrete networked Markov jump system stochastically stable with an H_∞ norm bound. The correspondent state feedback controller and the event-based parameters are also given. An illustrative example is given to show that the proposed analysis and design techniques are effective.

1. Introduction

Markov jump systems, which have randomly jumping parameters and the jumps are modeled by the transitions of a Markov chain, have received much attention [1–3]. Due to its capacity of capturing the abrupt mode changes for the plant, the Markov jump system has a wide range of application [4–7].

With the rapid development of information technology, networked control systems (NCSs) have received considerable attention in recent years [8–10]. NCSs are flexible to install and maintain, but the limited bandwidth leads to some challenges, such as network-induced delay, data dropout, and data disordering [11–14], which may deteriorate the system's performance. Traditionally, the sensor and the controller are updated uniformly in a constant sampling period, which is called as time-based sampling [15–18]. Since sampling happens at a fixed interval regardless whether it is really necessary or not, time-based sampling may lead to higher system costs. Recently, event-based sampling has been proposed as an effective means to reduce the sampling [19]. In event-based sampling scheme, the necessary sampling is determined by the occurrence of an "event" rather than "time". So event-based sampling can reduce the release times of the sensor and the burden of network communication.

Event-based control or estimation for different systems has received considerable attention in the past years. For continuous-time systems, the event-based analysis and design are shown in [19–27] and the references therein. In [19], the constructed event-triggers rely on continuous monitor of the system state and an additional hardware is needed to detect whether the current state exceeds the triggering threshold. Compared with the event-triggered method in [19], the self-triggered method used in [20,21] can save energy for the sensors and is less complexity to implement. However, its release period is often smaller than the event-triggered scheme, thus more constraints are needed for controller design or implementation under self-triggered scheme. Meanwhile, the event-triggered scheme that requires only supervision of the system state in discrete instants has been proposed and analyzed in [22,28–30]. Since NCSs are generally sampled-data systems, the discrete event-based scheme is more practical. For discrete-time systems, the event-triggered control or estimation was considered in [31–35]. Event-based control or filtering for discrete-time systems has received little attention. To the best of the authors' knowledge, there is no result reported in the open literature on the event-based H_∞ control for discrete networked Markov jump systems. The theoretical results for such systems would be appealing and have wide practical use, this motivates the research presented in this paper.

In this paper, we focus on the problem of event-based H_∞ control for discrete networked Markov jump systems. The event detector is positioned between the sensor and the controller for

☆This work was partially supported by the National Natural Science Foundation of China (61473264, 61473107, 61427808), the National Funds for Distinguished Young Scientists of China (61425009), the Zhejiang Provincial Natural Science Foundation of China (R1100716), and the Open Foundation of first level Zhejiang Province Key in Key Discipline of Control Science and Engineering.

E-mail address: hjwang@zstu.edu.cn (H. Wang).
[1] Tel.: +86 571 86843341.

http://dx.doi.org/10.1016/j.neucom.2016.01.021
0925-2312/

determining whether the newly sampled state should be sent out to the controller. The dynamic event-triggered scheme is designed to reduce the utilization of limited network resources. The time-delay method is employed to analysis and design the event-based H_∞ controller for discrete networked Markov jump systems.

The main contributions of this paper are threefold. (i) In order to reduce the use of limited network resources, the dynamic discrete event-triggered scheme, where there are different triggered thresholds for different Markov modes, is presented. (ii) The network-induced delay, data packet dropouts, and the Markov jump system are unified into the networked Markov jump system with time-varying delay based on the analysis of time interval. (iii) The H_∞ performance criterion is derived, and the co-design method of the event detector and the H_∞ controller is given.

The remaining of the paper is organized as follows. Section 2 formulates the problem under consideration. H_∞ control performance analysis and the method of state controller design based on state feedback control are presented in Section 3. Illustrative examples are given in Section 4, and the paper is concluded in Section 5.

*Notations:*Through this paper, the superscripts "T" and "-1" stand for the transpose of a matrix and the inverse of a matrix; $\mathbb{R}^n$ denotes n-dimensional Euclidean space; $\mathbb{R}^{n\times m}$ is the set of all real matrices with m rows and n columns; $P>0$ means that P is positive definite; I is the identity matrix with appropriate dimensions; the space of square-integrable vector functions over $[0,\infty)$ is denoted by $\mathcal{L}_2[0,\infty)$, and for $w(k)\in\mathcal{L}_2[0,\infty)$; for a symmetric matrix, $*$ denotes the matrix entries implied by symmetry.

2. Problem formulation

The framework of event-based H_∞ control for discrete Markov jump system considered in this paper is shown in Fig. 1.

2.1. Plant

The plant is assumed to be described by the following discrete Markov jump system:

$$\begin{cases} x(k+1)=A(r(k))x(k)+B(r(k))u(k)+E(r(k))w(k) \\ z(k)=C(r(k))x(k)+D(r(k))u(k) \end{cases} \tag{1}$$

where $x(k)\in\mathbb{R}^n$ is the state of the plant, $u(k)\in\mathbb{R}^m$ is the control input, $z(k)\in\mathbb{R}^p$ is the output to be controlled, and $w(k)\in\mathcal{L}_2[0,\infty)$ is the external disturbance; $r(k)$ represents a discrete-time homogeneous Markov chain, which takes values in a finite set $\wp$

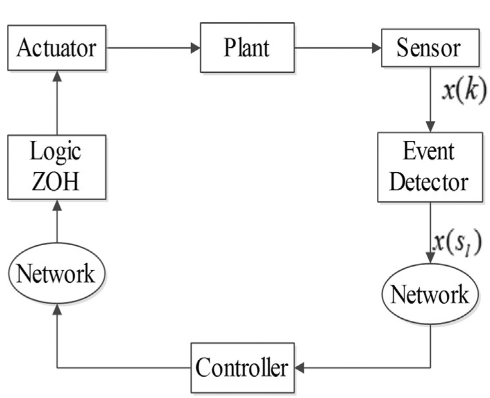

Fig. 1. Framework of event-based networked Markov jump system.

$=\{1,2,3,\dots,N\}$ with the following mode transition probabilities:

$$P_r\{r(k+1)=j|\,r(k)=i\}=\pi_{ij} \tag{2}$$

where $0\le\pi_{ij}\le 1,\ \forall i,j\in\wp$ and $\sum_{j=1}^{N}\pi_{ij}=1,\ \forall i\in\wp$.

For notational simplicity, in this paper, when $r(k)=i, i\in\wp$, a matrix $M(r(k))$ will be denoted by M_i; for example, $A(r(k))$ is denoted by A_i, $B(r(k))$ by B_i and so on.

2.2. Event detector

We suppose that the signal in the network is transmitted with a single packet. The framework of an event-based scheme for a NCS is shown in Fig. 1, where the "event detector" has the logic function to determine whether or not the current sampled packet should be transmitted. Let $x(k)$ being the current sampled packet, and $x(s_l)(l=0,1,2,\dots,\infty, s_0=0)$ being the latest transmitted packet, then the current sampled packet $x(k)$ is transmitted only when the following dynamic event-based scheme satisfies

$$[x(k)-x(s_l)]^T\Phi_i[x(k)-x(s_l)]>\varepsilon_i x^T(k)\Phi_i x(k) \tag{3}$$

where Φ_i is a positive-definite weighting matrix to design, and $\varepsilon_i\in[0,1)$ is the given scalar parameter.

Remark 1. Different from the traditional H_∞ control problem, the data packet transmission in event-based H_∞ control is mediated by event detector. The set of transmitted instants is a subset of the sampled instants, that is, $\{s_0,s_1,s_2,\dots\}\subseteq\{0,1,2,\dots\}$. When $\Phi_i=0$, $\{s_0,s_1,s_2,\dots\}=\{0,1,2,\dots\}$, that is, all the sampled signals are transmitted. It reduces to the traditional H_∞ control problem.

Remark 2. Note that s_l denotes the current triggered instant and s_{l+1} refers to the next triggered instant, and according to event-based scheme (3), for any $k\in[s_l,s_{l+1}-1]$, we have

$$[x(k)-x(s_l)]^T\Phi_i[x(k)-x(s_l)]\le\varepsilon_i x^T(k)\Phi_i x(k) \tag{4}$$

that is, only part of sampled packet is transmitted to the controller. So the amount of data transmission is reduced and the limited network-bandwidth is saved.

2.3. Time-delay modelling

Because of the limited network bandwidth, time-delay is inevitable in the process of signal transmission. We suppose that the transmission delay is τ_k at time instant k, and it is bounded, that is, $\tau_k\in[0,\tilde{\tau}]$, where $\tilde{\tau}$ is a positive integer. Consider the effect of transmission delay, the triggered data $x(s_l)$ reach the controller at time instant $s_l+\tau_{s_l}$. With the help of zero-order-holder (ZOH), the event-based controller considered in this paper can be written as

$$u(k)=K_i x(s_l),\quad k\in[s_l+\tau_{s_l}, s_{l+1}+\tau_{s_{l+1}}-1] \tag{5}$$

where $K_i\in\mathbb{R}^{m\times n}$ is the feedback gain to be designed later. Substituting (5) into (1) leads to the following closed-loop Markov jump system:

$$\begin{cases} x(k+1)=A_i x(k)+B_iK_i x(s_l)+E_i w(k) \\ z(k)=C_i x(k)+D_iK_i x(s_l), k\in[s_l+\tau_{s_l}, s_{l+1}+\tau_{s_{l+1}}-1]. \end{cases} \tag{6}$$

Remark 3. The logic ZOH stores the latest data packet in the holding interval $[s_l+\tau_{s_l}, s_{l+1}+\tau_{s_{l+1}}-1]$, which means that the actuator keeps the control input unchanged until the output of the logic ZOH is being updated to a new data packet. Because the