



Separation principle for networked control systems with multiple-packet transmission

D. Wu¹ J. Wu¹ S. Chen²

¹State Key Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, People's Republic of China

²School of Electronics and Computer Science, University of Southampton, Highfield, Southampton SO17 1BJ, UK
 E-mail: sqc@ecs.soton.ac.uk

Abstract: The authors investigate a class of observer-based discrete-time networked control systems (NCSs) with multiple-packet transmission where random packet dropouts occur independently in both the sensor-to-controller (S/C) and controller-to-actuator (C/A) channels. The authors first propose and prove the separation principle for the NCSs where packet dropouts in the C/A and S/C channels are governed by two independent Markov chains, respectively. Secondly, the authors derive a sufficient condition, in terms of linear matrix inequalities (LMIs), for stabilisation control of the Markov chain-driven NCSs. The authors also derive the necessary and sufficient condition for stabilisation control of the memoryless process-driven NCSs as a special case. A numerical example is provided to illustrate the effectiveness of our method.

1 Introduction

An important concept in control theory is the separation principle [1]. When the separation principle holds, the design of observer-based controller can be separated into a state feedback control and an observer design. The separation principle has been widely used in optimal control and estimation problems, such as the linear quadratic Gaussian (LQG) optimal control [2]. For non-linear observer-based control design, the separation principle has also been proposed, for example, by developing a high-gain observer approach [3]. Networked control systems (NCSs) have drawn much attention in the control community recently [4–7]. An NCS is a control system in which the control loop is closed via a shared communication network. Compared to the point-to-point system connection, the use of an NCS has the advantages of low installation cost, reducing system wiring, simple system diagnosis and easy maintenance. However, some inherent shortcomings of NCSs, such as bandwidth constraints, packet dropouts and packet delays, will degrade the performance of NCSs or even cause instability.

Packet dropouts, which can randomly occur because of node failures or network congestion, impose one of the most critical problems in NCSs. Stochastic approaches based on the mean square stability [8, 9] are typically adopted to deal with network delay and packet dropout. Under such a stochastic approach, the network is modelled as a Bernoulli process [5, 10, 11] or a Markov chain [12, 13], and the system is considered as a special case of a discrete-time jump linear system. In some other works [14, 15], NCSs with arbitrary packet dropouts are modelled as switched systems. An effective strategy for combating packet dropouts is to compensate them. The works [16, 17] use the model

predictive controller to send the current and future control inputs in a single packet. The authors of [18] consider the compensation for dropped feedback measurements, while the authors of [19] propose a compensation scheme for the control packet dropout at the actuator using the past control signals. The work [20] studies the optimal state estimation at the controller for open-loop NCSs, while the work [21] designs an agent-based networked control estimator. These methods provide better performance, as they compensate the dropped packets. However, these schemes complicate the controller design task for closed-loop NCSs, or they require the actuator node to have a complex computational capability.

Since most of the actuator nodes in practice do not have the required computational capability to implement a sophisticated compensation scheme for control packet dropouts, two simple methods are typically adopted for dealing with packet dropouts at the actuator node. The first one is the zero-input scheme where the actuator value is set to zero when the current control packet is lost in the controller-to-actuator (C/A) channel [5, 10, 15], while the other one is the hold-input scheme where the actuator holds the previous control input when the current control packet is lost [12, 14, 22]. Recently, Schenato [23] has made a detailed comparison of the zero-input and hold-input schemes. The hold-input scheme can provide a smoother sequence of control inputs. Nevertheless, in the case of large packet dropout probability, long consecutive packet dropouts cause the problem of overshooting and oscillations under the hold-input scheme. In general, none of these two schemes can be claimed to be superior to the other [23]. However, for certain industrial process control systems where it is critical to maintain a smooth operation of an industrial process, the zero-input scheme may not be applicable.

In addition, there have been some studies on the effects of packet dropouts to NCSs under multiple-packet transmission [11, 12, 15, 22]. In a multiple-packet transmission, individual sensor or controller data are transmitted in separate network packets that may not all arrive at the controller or actuator simultaneously because of packet dropouts. By contrast, in a single-packet transmission, all the sensors or controllers data are lumped together into one packet and transmitted at the same time. There are two reasons for adopting multiple-packet transmission. Firstly, a large amount of data must be broken into multiple packets because of the packet size constraint. Secondly and more importantly, sensors and actuators in an NCS may be distributed over a large physical area.

Owing to its importance to control system analysis and synthesis, the separation principle has been studied for some special NCSs with nominal plant. For example, the separation principle is known to hold for the NCSs under TCP-like protocols, where packet dropouts in each communication channel is modelled as a Bernoulli process [5]. In [6], the authors prove the separation principle for the system with nominal plant, where the network is only located in the sensor-to-controller (S/C) channel. In [24], the author derives the separation principle for the time-varying sampled data system under the assumption that when the plant output is available at the controller, the actuator also receives the related new control data as well. In reality, however, the packets may be lost independently in the S/C and C/A channels.

To the best of our knowledge, no work to date has proposed the separation principle for observer-based discrete-time NCSs with multiple-packet transmission where packet dropouts in the S/C and C/A channels are driven by two independent Markov chains. Hence, the focus of the present work is to investigate the separation principle for such NCSs. Our contributions are twofold. Firstly, we present and prove the separation principle for Markov chain-driven NCSs with multiple-packet transmission. We formulate such NCSs as the Markovian jump linear system by modelling the behaviours of packet dropouts in the S/C and C/A channels as independent Markov chains. Secondly, we derive a sufficient condition for synthesising the stabilisation controller for such NCSs, formulated in terms of linear matrix inequalities (LMIs) [25]. The necessary and sufficient condition is also presented for the memoryless process-driven NCSs as a special case. An example is provided to demonstrate our proposed approach.

2 Preliminaries

Throughout this paper \mathbb{R} stands for real numbers and \mathbb{N} for non-negative integers, while $\mathbf{W} > \mathbf{0}$ indicates that \mathbf{W} is a positive-definite matrix. Furthermore, \mathbf{I} and $\mathbf{0}$ represent the identity and zero matrices of appropriate dimension, respectively. For positive integer n , define the set $\mathcal{S}_n \triangleq \{\boldsymbol{\Omega} | \boldsymbol{\Omega} = \text{diag}(\omega_1, \dots, \omega_n), \omega_i \in \{0, 1\}, i \in \{1, \dots, n\}\}$. The number of elements in the set \mathcal{S}_n is 2^n . Further define the one-to-one mapping f_n from \mathcal{S}_n to $\{1, 2, \dots, 2^n\}$ as

$$\rho = f_n(\boldsymbol{\Omega}) = 1 + \sum_{i=1}^n \omega_i \cdot 2^{i-1} \tag{1}$$

and define the inverse mapping as $\boldsymbol{\Omega} = \mathbf{H}_n(\rho)$, which can be implemented by the following iterative procedure:

- Step 1: Set $v = \rho - 1, i = 1$.
- Step 2: Find $g \in \mathbb{N}$ and $d \in \{0, 1\}$ to satisfy $v = 2g + d$. Then $\omega_i = d$.
- Step 3: If $i < n$, then $v = g, i = i + 1$, return to Step 2.
- Step 4: $\boldsymbol{\Omega} = \text{diag}(\omega_1, \dots, \omega_n)$, End.

Lemma 1 (see [25]): Given constant matrices $\mathbf{V}_1, \mathbf{V}_2$ and \mathbf{V}_3 of appropriate dimensions where \mathbf{V}_1 and \mathbf{V}_2 are symmetric, then

$$\begin{bmatrix} \mathbf{V}_2 & \mathbf{V}_3^T \\ \mathbf{V}_3 & \mathbf{V}_1 \end{bmatrix} > \mathbf{0}$$

if and only if $\mathbf{V}_1 > \mathbf{0}$ and $\mathbf{V}_2 - \mathbf{V}_3^T \mathbf{V}_1^{-1} \mathbf{V}_3 > \mathbf{0}$.

For a discrete-time stochastic signal $\mathbf{r} = \{\mathbf{r}(k)\}_{k \in \mathbb{N}}$ with $\mathbf{r}(k)$ a \mathbb{R}^n -valued random variable, let us define

$$\|\mathbf{r}\|_{2s} \triangleq \sqrt{\sum_{k=0}^{\infty} \mathbf{E}[\mathbf{r}^T(k)\mathbf{r}(k)]} \tag{2}$$

where $\mathbf{E}[\cdot]$ denotes the expectation. Let ℓ_{2s}^n be the set of \mathbf{r} with $\|\mathbf{r}\|_{2s} < \infty$. For positive integer N , define $\mathcal{N} = \{1, 2, \dots, N\}$. Consider the following stochastic system, denoted as \hat{F}

$$\begin{cases} \mathbf{x}_f(k+1) = \mathbf{A}_{\theta(k)}\mathbf{x}_f(k) + \mathbf{B}_{\theta(k)}\mathbf{u}_f(k), \\ \mathbf{y}_f(k) = \mathbf{C}_{\theta(k)}\mathbf{x}_f(k) + \mathbf{D}_{\theta(k)}\mathbf{u}_f(k), \end{cases} k \in \mathbb{N} \tag{3}$$

where $\mathbf{x}_f(k) \in \mathbb{R}^n, \mathbf{u}_f(k) \in \mathbb{R}^m$ and $\mathbf{y}_f(k) \in \mathbb{R}^q$ are the state, input and output vectors, respectively. The stochastic process $\theta(k)$ is driven by an \mathcal{N} -valued Markov chain [26]. The transition probabilities of $\theta(k)$ are given by $p_{i,j} = \text{Prob}(\theta(k+1) = j | \theta(k) = i)$ with $p_{i,j} \geq 0$ and $\sum_{j \in \mathcal{N}} p_{i,j} = 1$ for each $i \in \mathcal{N}$. The following result defines the stochastic stability of \hat{F} .

Lemma 2 (see [8, 9]): For the system \hat{F} defined in (3), the following are equivalent:

1. \hat{F} with $\mathbf{u}_f(k) \equiv \mathbf{0}$ is stochastically stable;
2. $\forall \mathbf{x}_f(0) \in \mathbb{R}^n, \forall \theta(0) \in \mathcal{N}, \mathbf{x}_f \in \ell_{2s}^n$ with $\mathbf{u}_f(k) \equiv \mathbf{0}$;
3. $\exists \mathbf{0} < \mathbf{P}_i \in \mathbb{R}^{n \times n}$ with $i \in \mathcal{N}$ such that $\mathbf{P}_i - \mathbf{A}_i^T (\sum_{j \in \mathcal{N}} p_{i,j} \mathbf{P}_j) \mathbf{A}_i > \mathbf{0}, \forall i \in \mathcal{N}$;
4. $\exists \mathbf{0} < \mathbf{W}_j \in \mathbb{R}^{n \times n}$ with $j \in \mathcal{N}$ such that $\mathbf{W}_j - \mathbf{A}_j (\sum_{i \in \mathcal{N}} p_{i,j} \mathbf{W}_i) \mathbf{A}_j^T > \mathbf{0}, \forall j \in \mathcal{N}$.

It is interesting to note that under a memoryless process assumption the condition simplifies as follows.

Lemma 3 (see [9]): Suppose $p_{i,j} = p_j$ for all $i, j \in \mathcal{N}$. Then the system \hat{F} in (3) is stochastically stable if and only if $\exists \mathbf{0} < \mathbf{P} \in \mathbb{R}^{n \times n}$ such that $\mathbf{P} - \sum_{j \in \mathcal{N}} p_j \mathbf{A}_j^T \mathbf{P} \mathbf{A}_j > \mathbf{0}$.

Lemma 4 (see [13]): If there exist $0 < \beta \in \mathbb{R}$ and $\mathbf{0} < \mathbf{P}_i \in \mathbb{R}^{n \times n}$ with $i \in \mathcal{N}$ such that

$$\begin{bmatrix} \mathbf{P}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} - \begin{bmatrix} \mathbf{A}_i & \beta \mathbf{B}_i \\ \mathbf{C}_i & \beta \mathbf{D}_i \end{bmatrix}^T \begin{bmatrix} \sum_{j \in \mathcal{N}} p_{i,j} \mathbf{P}_j & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \times \begin{bmatrix} \mathbf{A}_i & \beta \mathbf{B}_i \\ \mathbf{C}_i & \beta \mathbf{D}_i \end{bmatrix} > \mathbf{0}, \quad \forall i \in \mathcal{N}$$

then the system \hat{F} of (3) is stochastically stable, and

furthermore for $\mathbf{x}_f(0) = \mathbf{0}$, $\forall \mathbf{u}_f \in \ell_{2s}^m$ and $\forall \theta(0) \in \mathcal{N}$, we have $\|\mathbf{y}_f\|_{2s} \leq \|\mathbf{u}_f\|_{2s}/\beta$.

3 Problem formulation

The NCS \hat{P}_K , depicted in Fig. 1, consists of a discrete-time linear time-invariant (LTI) plant \hat{P} and a discrete-time LTI controller \hat{K} with the control loop closed via a shared communication network. The plant \hat{P} , including sensors and actuators, is described by

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k), \end{cases} \quad k \in \mathbb{N} \quad (4)$$

where $\mathbf{x}(k) \in \mathbb{R}^n$, $\mathbf{u}(k) = [u_1(k) \dots u_m(k)]^T \in \mathbb{R}^m$ and $\mathbf{y}(k) = [y_1(k) \dots y_q(k)]^T \in \mathbb{R}^q$ are the state, input and output vectors, respectively, while $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$ and $\mathbf{C} \in \mathbb{R}^{q \times n}$.

Assume that the m actuators and q sensors of plant (4) are physically distributed. At each instant $k \in \mathbb{N}$, m control input packets from the controller are transmitted through the C/A channel to the m actuators, one for each element of $\hat{\mathbf{u}}(k)$. Each packet may be successfully transmitted or dropped. The packet transmission delay through the network for a successful transmission is assumed to be negligible, that is, there is no delay in communication. TCP-like protocol is assumed, in which there is acknowledgement for a received packet. Thus, at each instant $k \in \mathbb{N}$, the controller has indicators of whether the current control input packets are received or not by the actuators. Referring to Fig. 1, if the actuator j receives the control input packet $\hat{u}_j(k)$ at instant k , then $u_j(k) = \hat{u}_j(k)$, otherwise $u_j(k) = 0$. Thus, we have

$$\mathbf{u}(k) = \Theta_a(k)\hat{\mathbf{u}}(k) \quad (5)$$

where

$$\Theta_a(k) \triangleq \text{diag}(\theta_{a,1}(k), \theta_{a,2}(k), \dots, \theta_{a,m}(k)) \in S_m$$

and $\theta_{a,j}(k) \in \{0, 1\}$ for $j \in \{1, \dots, m\}$ are the indicators of single-packet dropout in the C/A channel for $\hat{u}_j(k)$. Specifically, a value 0 indicates that the packet is dropped while a value 1 indicates that the packet is transmitted successfully. It is clear that we adopt the zero-input scheme for compensating the dropped control packet. An alternative method is the hold-input scheme, which keeps

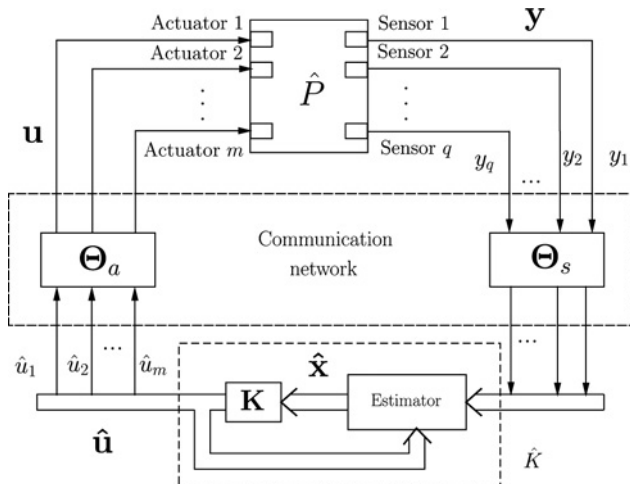


Fig. 1 Networked control system \hat{P}_K

$u_j(k) = \hat{u}_j(k-1)$ if $\hat{u}_j(k)$ is lost. These two schemes are the two most natural methods for dealing with control packet dropouts at the actuator node [23].

Also at each instant $k \in \mathbb{N}$, q output packets from the q sensors are transmitted through the S/C channel to the controller, one for each element of $\mathbf{y}(k)$. Each packet may be successfully transmitted or dropped and hence let us define

$$\Theta_s(k) \triangleq \text{diag}(\theta_{s,1}(k), \theta_{s,2}(k), \dots, \theta_{s,q}(k)) \in S_q$$

with $\theta_{s,i}(k) \in \{0, 1\}$ for $i \in \{1, \dots, q\}$ denoting the indicators of single-packet dropout in the S/C channel for $y_i(k)$. Since there is no delay in communication, \hat{K} receives those $y_i(k)$ having the indicators $\theta_{s,i}(k) = 1$ at the instant k . The controller \hat{K} adopts a smart control mechanism [6] to update the estimator state $\hat{\mathbf{x}}(k+1)$ and controller output $\hat{\mathbf{u}}(k+1)$ as follows

$$\hat{\mathbf{x}}(k+1) = \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\Theta_a(k)\hat{\mathbf{u}}(k) + \mathbf{L}\Theta_s(k)(\hat{\mathbf{y}}(k) - \mathbf{y}(k)) \quad (6)$$

$$\hat{\mathbf{y}}(k) = \mathbf{C}\hat{\mathbf{x}}(k) \quad (7)$$

$$\hat{\mathbf{u}}(k+1) = \mathbf{K}\hat{\mathbf{x}}(k+1) \quad (8)$$

where \mathbf{K} is the state feedback gain matrix and \mathbf{L} the observer gain matrix. The vector $\hat{\mathbf{u}}(k+1)$ will be transmitted to the actuators at the next instant $k+1$. The underlying idea of this smart control mechanism is as follows. With $\theta_{s,i}(k) = 1$, the controller receives $y_i(k)$ and it utilises the error $\hat{y}_i(k) - y_i(k)$ to correct $\hat{\mathbf{x}}(k+1)$, which is similar to the standard observation law, while with $\theta_{s,i}(k) = 0$, $y_i(k)$ is dropped and no correction on $\hat{\mathbf{x}}(k+1)$ is made, which is just generated from an imitation of the plant.

The sequences $\{\Theta_s(k)\}_{k \in \mathbb{N}}$ and $\{\Theta_a(k)\}_{k \in \mathbb{N}}$ specify the packet dropout processes in the S/C and C/A channels, respectively, and they can be mapped into the two sequences $\{s_k\}_{k \in \mathbb{N}}$ with $s_k = f_q(\Theta_s(k))$ and $\{a_k\}_{k \in \mathbb{N}}$ with $a_k = f_m(\Theta_a(k))$. The inverse mappings of f_q and f_m are simply $\Theta_s(k) = \mathbf{H}_q(s_k)$ and $\Theta_a(k) = \mathbf{H}_m(a_k)$, respectively. We consider the case where $\{s_k\}_{k \in \mathbb{N}}$ and $\{a_k\}_{k \in \mathbb{N}}$ are driven by two independent discrete-time stochastic processes with the following assumption.

Assumption 1: The stochastic processes s_k and a_k are driven by the two independent homogeneous Markov chains, which take values in the set $\mathcal{Q} \triangleq \{1, \dots, 2^q\}$ with the transition probability matrix $\Pi \triangleq [\pi_{t,r}]$ and in the set $\mathcal{M} \triangleq \{1, \dots, 2^m\}$ with the transition probability matrix $\Lambda \triangleq [\lambda_{i,j}]$, respectively. Here the probabilities $\pi_{t,r}$ and $\lambda_{i,j}$ are defined, respectively, by

$$\pi_{t,r} = \text{Prob}(s_{k+1} = r | s_k = t) \quad (9)$$

$$\lambda_{i,j} = \text{Prob}(a_{k+1} = j | a_k = i) \quad (10)$$

with $\sum_{r \in \mathcal{Q}} \pi_{t,r} = 1$, $\pi_{t,r} \geq 0$, $\forall t, r \in \mathcal{Q}$ and $\sum_{j \in \mathcal{M}} \lambda_{i,j} = 1$, $\lambda_{i,j} \geq 0$, $\forall i, j \in \mathcal{M}$.

Remark 1: The multiple-packet transmission policy considered in this contribution is a general framework for packet dropouts. This multiple-packet transmission policy, for example, is valid in the following two scenarios. In the first case several output or input variables are lumped into one packet, while in the second case some output or input variables have dedicated links and are always transmitted

successfully. These two cases can simply be tackled by setting the related probabilities to be 0. For the first case, for instance, assume that $q = 3$ and the output variables $y_1(k)$ and $y_2(k)$ are transmitted together in one packet. In this scenario $\Theta_s(k)$ never takes the values $\text{diag}(1, 0, 0)$, $\text{diag}(0, 1, 0)$, $\text{diag}(1, 0, 1)$ or $\text{diag}(0, 1, 1)$. Note that the corresponding values of s_k for these matrices are 2, 3, 6 and 7, respectively. Clearly, this case can be presented by setting $\pi_{t,r} = 0$, when $t \in \{2, 3, 6, 7\}$ and setting $\pi_{t,r} = 0$, when $r \in \{2, 3, 6, 7\}$. For the second case, assume that $q = 3$ sensors and $y_1(k)$ is never dropped. Then $\Theta_s(k)$ never takes the values $\text{diag}(0, 0, 0)$, $\text{diag}(0, 1, 0)$, $\text{diag}(0, 0, 1)$ or $\text{diag}(0, 1, 1)$. Since the corresponding values of s_k for these matrices are 1, 3, 5 and 7, respectively, we can represent this case by setting $\pi_{t,r} = 0$, when $t \in \{1, 3, 5, 7\}$ and setting $\pi_{t,r} = 0$, when $r \in \{1, 3, 5, 7\}$.

Remark 2: We mainly consider packet dropouts for the discrete-time LTI plant \hat{P} of (4), which is directly derived from the continuous-time plant by sampling. Therefore our NCS model can naturally deal with the case where the maximum packet transmission delay in the S/C channel and the maximum packet transmission delay in the C/A channel are within one sampling period. In addition, most of the present NCSs are configured over local area networks (LANs), such as wired Ethernet and wireless LAN (WLAN). In such NCSs, the most significant communication delay is because of access delay, which again is naturally taken into account in our model, while the communication delay of a successful packet transmission is negligible.

Remark 3: In general, the S/C and C/A channels are most likely to physically locate at the different segments of a LAN, which forms the communication network of the system, owing to the spatial distribution nature of the control system. Therefore Assumption 1 is a generic assumption. Even in the case that both the S/C and C/A channels are located in the same segment of the LAN, they may still be regarded as independent, as they may have different bandwidth assignments and/or other network policy. Thus, many important NCSs, including the works of [5, 10, 12, 15], are developed based on the assumption of two independent stochastic processes for the S/C and C/A channels.

Let us define the state of the NCS \hat{P}_K as

$$\bar{\mathbf{x}}(k) \triangleq [\mathbf{x}^T(k) \mathbf{e}^T(k)]^T \quad (11)$$

where $\mathbf{e}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k)$. From (4)–(8), the NCS \hat{P}_K can be described by

$$\bar{\mathbf{x}}(k+1) = \bar{\mathbf{A}}_{s_k a_k} \bar{\mathbf{x}}(k), \quad \forall k \in \mathbb{N} \quad (12)$$

where

$$\bar{\mathbf{A}}_{s_k a_k} = \begin{bmatrix} \mathbf{A} + \mathbf{B}\mathbf{H}_m(a_k)\mathbf{K} & -\mathbf{B}\mathbf{H}_m(a_k)\mathbf{K} \\ \mathbf{0} & \mathbf{A} + \mathbf{L}\mathbf{H}_q(s_k)\mathbf{C} \end{bmatrix} \quad (13)$$

Our main objective is to derive the separation principle and stabilisation control for the NCSs with multiple-packet transmission where packet dropouts in the S/C and C/A channels are driven by two independent Markov chains. We also investigate the NCSs driven by the two memoryless processes under a special case of Assumption 1 with $\pi_{t,r} = \pi_r$ and $\lambda_{i,j} = \lambda_j$.

Assumption 2: The stochastic processes s_k and a_k are driven by the two independent memoryless processes. The probability mass function of s_k is denoted by $\pi_r = \text{Prob}(s_k = r)$, $\forall r \in \mathcal{Q}$, with the state probability matrix $\mathbf{\Pi}_M \triangleq [\pi_r]$, while the probability mass function of a_k is denoted by $\lambda_j = \text{Prob}(a_k = j)$, $\forall j \in \mathcal{M}$, with the state probability matrix $\mathbf{\Lambda}_M \triangleq [\lambda_j]$.

4 Separation principle

We first derive the separation principle for Markov chain-driven NCSs. Noting that the NCS \hat{P}_K (12) is a Markovian jump linear system, we prove our result in the following theorem.

Theorem 1 (Separation Principle): The NCS \hat{P}_K (12) under Assumption 1 is stochastically stable if and only if the following two conditions hold.

(i) $\exists \mathbf{0} < \mathbf{Q}_t \in \mathbb{R}^{n \times n}$ with $t \in \mathcal{Q}$ such that

$$\mathbf{Q}_t - \mathbf{\Psi}_t^T \left(\sum_{r \in \mathcal{Q}} \pi_{t,r} \mathbf{Q}_r \right) \mathbf{\Psi}_t > \mathbf{0}, \quad \forall t \in \mathcal{Q} \quad (14)$$

where

$$\mathbf{\Psi}_t = \mathbf{A} + \mathbf{L}\mathbf{H}_q(t)\mathbf{C} \quad (15)$$

(ii) $\exists \mathbf{0} < \mathbf{P}_i \in \mathbb{R}^{n \times n}$ with $i \in \mathcal{M}$ such that

$$\mathbf{P}_i - \mathbf{\Phi}_i^T \left(\sum_{j \in \mathcal{M}} \lambda_{i,j} \mathbf{P}_j \right) \mathbf{\Phi}_i > \mathbf{0}, \quad \forall i \in \mathcal{M} \quad (16)$$

where

$$\mathbf{\Phi}_i = \mathbf{A} + \mathbf{B}\mathbf{H}_m(i)\mathbf{K} \quad (17)$$

Proof: (If). From (11) to (15), a sub-system for \mathbf{e} can readily be obtained as

$$\mathbf{e}(k+1) = \mathbf{\Psi}_{s_k} \mathbf{e}(k), \quad \forall k \in \mathbb{N} \quad (18)$$

According to Lemma 2 and (14), the sub-system (18) is stochastically stable. Hence, $\forall \bar{\mathbf{x}}(0) \in \mathbb{R}^{2n}$, $\mathbf{e} \in \ell_{2s}^n$. From (11) to (17), a sub-system for \mathbf{x} can readily be obtained as

$$\mathbf{x}(k+1) = \mathbf{\Phi}_{a_k} \mathbf{x}(k) + \mathbf{\Gamma}_{a_k} \mathbf{e}(k), \quad k \in \mathbb{N} \quad (19)$$

where

$$\mathbf{\Gamma}_{a_k} = -\mathbf{B}\mathbf{H}_m(a_k)\mathbf{K} \quad (20)$$

Partition \mathbf{x} as

$$\mathbf{x}(k) = \mathbf{x}_a(k) + \mathbf{x}_b(k) \quad (21)$$

with

$$\mathbf{x}_a(k+1) = \mathbf{\Phi}_{a_k} \mathbf{x}_a(k), \quad \mathbf{x}_a(0) = \mathbf{x}(0) \quad (22)$$

$$\mathbf{x}_b(k+1) = \mathbf{\Phi}_{a_k} \mathbf{x}_b(k) + \mathbf{\Gamma}_{a_k} \mathbf{e}(k), \quad \mathbf{x}_b(0) = \mathbf{0} \quad (23)$$

From Lemma 2 and (16), it is easy to show that $\forall \bar{\mathbf{x}}(0) \in \mathbb{R}^{2n}$, $\mathbf{x}_a \in \ell_{2s}^n$. If (16) is satisfied, then there exists a sufficiently small constant $0 < \gamma \in \mathbb{R}$ such that $\forall i \in \mathcal{M}$

$$\begin{bmatrix} \mathbf{P}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} - \begin{bmatrix} \Phi_i & \gamma \Gamma_i \\ \gamma \mathbf{I} & \mathbf{0} \end{bmatrix}^\top \times \begin{bmatrix} \sum_{j \in \mathcal{M}} \lambda_{i,j} \mathbf{P}_j & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Phi_i & \gamma \Gamma_i \\ \gamma \mathbf{I} & \mathbf{0} \end{bmatrix} > \mathbf{0} \quad (24)$$

This together with Lemma 4 yields

$$\|\gamma \mathbf{x}_b\|_{2s} < \|\mathbf{e}\|_{2s} / \gamma \quad (25)$$

namely, $\forall \bar{\mathbf{x}}(0) \in \mathbb{R}^{2n}$, $\mathbf{x}_b \in \ell_{2s}^n$. Thus, $\forall \bar{\mathbf{x}}(0) \in \mathbb{R}^{2n}$, $\mathbf{x} \in \ell_{2s}^n$. According to Lemma 2, the NCS \hat{P}_K is stochastically stable.

(Only If). As the NCS \hat{P}_K is stochastically stable, then $\forall \bar{\mathbf{e}}(0) \in \mathbb{R}^n$, $\mathbf{e} \in \ell_{2s}^n$. Thus, the sub-system (18) for \mathbf{e} is stochastically stable, and the condition (i) holds according to Lemma 2. On the other hand, for the sub-system (18), $\mathbf{e}(k) \equiv \mathbf{0}$ when $\mathbf{e}(0) = \mathbf{0}$. In other words, the sub-system (19) for \mathbf{x} becomes

$$\mathbf{x}(k+1) = \Phi_{a_k} \mathbf{x}(k) \quad (26)$$

in the case of $\mathbf{e}(0) = \mathbf{0}$. The stochastic stability of \hat{P}_K means that for $\mathbf{e}(0) = \mathbf{0}$, $\forall \mathbf{x}(0) \in \mathbb{R}^n$, $\mathbf{x} \in \ell_{2s}^n$. Thus, the sub-system (19) is stochastically stable and hence the condition (ii) holds. \square

Note that (14) only involves \mathbf{L} while (16) only involves \mathbf{K} . Hence, the state feedback control and observer can be designed independently by assuring each separated part be stochastically stable. This property is in accordance with the separation principle.

For the memoryless process-driven NCSs under Assumption 2, we can simplify the separation principle as follows.

Theorem 2: The NCS \hat{P}_K (12) under Assumption 2 is stochastically stable if and only if the following two conditions hold.

(i) $\exists \mathbf{0} < \mathbf{Q} \in \mathbb{R}^{n \times n}$ such that

$$\mathbf{Q} - \sum_{r \in \mathcal{Q}} \pi_r \Psi_r^\top \mathbf{Q} \Psi_r > \mathbf{0}, \quad \forall r \in \mathcal{Q} \quad (27)$$

(ii) $\exists \mathbf{0} < \mathbf{P} \in \mathbb{R}^{n \times n}$ such that

$$\mathbf{P} - \sum_{j \in \mathcal{M}} \lambda_j \Phi_j^\top \mathbf{P} \Phi_j > \mathbf{0}, \quad \forall j \in \mathcal{M} \quad (28)$$

where Ψ_r and Φ_j are given in (15) and (17), respectively.

Proof: The proof is straightforward by directly applying Theorem 1 and Lemma 3. \square

5 Stabilisation control

The task of stabilisation control for Markov chain-driven NCSs is as follows. Given \mathbf{A} , \mathbf{B} and \mathbf{C} as well as the transition probabilities $\pi_{r,t}$ and $\lambda_{i,j}$ under Assumption 1, determine the observer gain \mathbf{L} and state feedback gain \mathbf{K} such that \hat{P}_K is stochastically stable.

Based on the separation principle derived in the previous section and with the introduction of freedom matrices [27, 28], such as \mathbf{G}_1 and \mathbf{G}_2 in the following theorem, we can synthesis the stabilisation control for NCSs in terms of LMI.

Theorem 3: For the NCS \hat{P}_K (12) under Assumption 1, suppose that there exist $\mathbf{0} < \mathbf{Q}_t \in \mathbb{R}^{n \times n}$ with $t \in \mathcal{Q}$, $\mathbf{0} < \mathbf{W}_j \in \mathbb{R}^{n \times n}$ with $j \in \mathcal{M}$, $\mathbf{Y}_1 \in \mathbb{R}^{n \times q}$, $\mathbf{Y}_2 \in \mathbb{R}^{m \times n}$, $\mathbf{G}_1 \in \mathbb{R}^{n \times n}$ and $\mathbf{G}_2 \in \mathbb{R}^{n \times n}$ such that $\forall t \in \mathcal{Q}$, $\forall j \in \mathcal{M}$

$$\begin{bmatrix} \mathbf{Q}_t & \Xi_\pi^\top(t, 1) & \cdots & \Xi_\pi^\top(t, 2^q) \\ \Xi_\pi(t, 1) & \mathbf{Q}_G(i) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \Xi_\pi(t, 2^q) & \mathbf{0} & \cdots & \mathbf{Q}_G(2^q) \end{bmatrix} > \mathbf{0} \quad (29)$$

$$\begin{bmatrix} \mathbf{W}_j & \Xi_\lambda(1, j) & \cdots & \Xi_\lambda(2^m, j) \\ \Xi_\lambda^\top(1, j) & \mathbf{W}_G(1) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \Xi_\lambda^\top(2^m, j) & \mathbf{0} & \cdots & \mathbf{W}_G(2^m) \end{bmatrix} > \mathbf{0} \quad (30)$$

where

$$\Xi_\pi(t, i) = \sqrt{\pi_{t,i}} (\mathbf{G}_1 \mathbf{A} + \mathbf{Y}_1 \mathbf{H}_q(t) \mathbf{C}), \quad i \in \mathcal{Q}$$

$$\mathbf{Q}_G(i) = \mathbf{G}_1 + \mathbf{G}_1^\top - \mathbf{Q}_i, \quad i \in \mathcal{Q}$$

$$\Xi_\lambda(i, j) = \sqrt{\lambda_{i,j}} (\mathbf{A} \mathbf{G}_2 + \mathbf{B} \mathbf{H}_m(j) \mathbf{Y}_2), \quad i \in \mathcal{M}$$

$$\mathbf{W}_G(i) = \mathbf{G}_2 + \mathbf{G}_2^\top - \mathbf{W}_i, \quad i \in \mathcal{M}$$

Then $\mathbf{L} = \mathbf{G}_1^{-1} \mathbf{Y}_1$ and $\mathbf{K} = \mathbf{Y}_2 \mathbf{G}_2^{-1}$ make \hat{P}_K stochastically stable.

Proof: Firstly, the non-singularity of \mathbf{G}_1 and \mathbf{G}_2 is guaranteed by (29) and (30) that include $\mathbf{G}_1 + \mathbf{G}_1^\top > \mathbf{0}$ and $\mathbf{G}_2 + \mathbf{G}_2^\top > \mathbf{0}$, respectively. The constraint $\mathbf{G}_1 + \mathbf{G}_1^\top > \mathbf{0}$ does not allow any singular \mathbf{G}_1 . In fact, if $\det(\mathbf{G}_1) = 0$, there exists a non-zero $\mathbf{v} \in \mathbb{R}^n$ such that $\mathbf{G}_1 \mathbf{v} = \mathbf{0}$ and hence $\mathbf{v}^\top \mathbf{G}_1^\top = \mathbf{0}$. Thus, $\mathbf{v}^\top (\mathbf{G}_1^\top + \mathbf{G}_1) \mathbf{v} = \mathbf{0}$ which contradicts $\mathbf{G}_1 + \mathbf{G}_1^\top > \mathbf{0}$. Similarly, \mathbf{G}_2 is non-singular. Secondly, set $\mathbf{L} = \mathbf{G}_1^{-1} \mathbf{Y}_1$ and $\Psi_t = \mathbf{A} + \mathbf{L} \mathbf{H}_q(t) \mathbf{C}$. Then, pre- and post- multiplying (29) by

$$\begin{bmatrix} \mathbf{I} & -\sqrt{\pi_{t,1}} \Psi_t^\top & \cdots & -\sqrt{\pi_{t,2^q}} \Psi_t^\top \end{bmatrix}$$

and its transpose, we arrive at condition (i) of Theorem 1. Thirdly, set $\mathbf{K} = \mathbf{Y}_2 \mathbf{G}_2^{-1}$ and $\Phi_j = \mathbf{A} + \mathbf{B} \mathbf{H}_m(j) \mathbf{K}$. Then, pre- and post- multiplying (30) by

$$\begin{bmatrix} \mathbf{I} & -\sqrt{\lambda_{1,j}} \Phi_j & \cdots & -\sqrt{\lambda_{2^m,j}} \Phi_j \end{bmatrix}$$

and its transpose, we have

$$\mathbf{W}_j - \Phi_j \left(\sum_{i \in \mathcal{M}} \lambda_{i,j} \mathbf{W}_i \right) \Phi_j^\top > \mathbf{0}$$

From Lemma 2, the above inequality is equivalent to condition (ii) of Theorem 1. Finally, by Theorem 1, we know that \hat{P}_K is stochastically stable. \square

Clearly, Theorem 3 only gives a sufficient condition for controller design. For the memoryless process-driven NCSs, we can derive the necessary and sufficient condition for controller design.

Theorem 4: For the NCS \hat{P}_K (12) under Assumption 2, there exist L and K such that \hat{P}_K is stochastically stable if and only if there exist $0 < Q \in \mathbb{R}^{n \times n}$, $0 < X \in \mathbb{R}^{n \times n}$, $Y_1 \in \mathbb{R}^{n \times q}$ and $Y_2 \in \mathbb{R}^{m \times n}$ such that

$$\begin{bmatrix} Q & \# & \dots & \# \\ \sqrt{\pi_1}(QA + Y_1H_q(1)C) & Q & \dots & \# \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{\pi_{2^q}}(QA + Y_1H_q(2^q)C) & 0 & \dots & Q \end{bmatrix} > 0 \quad (31)$$

$$\begin{bmatrix} X & \# & \dots & \# \\ \sqrt{\lambda_1}(AX + BH_m(1)Y_2) & X & \dots & \# \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{\lambda_{2^m}}(AX + BH_m(2^m)Y_2) & 0 & \dots & X \end{bmatrix} > 0 \quad (32)$$

where $\#$ denotes the symmetric term in a symmetric matrix. Moreover, if (31) and (32) have solutions, the observer and state feedback gains are given by $L = Q^{-1}Y_1$ and $K = Y_2X^{-1}$, respectively.

Proof: By pre- and post- multiplying (31) with $\text{diag}\{I, Q^{-1}, \dots, Q^{-1}\}$, setting $L = Q^{-1}Y_1$ and applying Lemma 1, we can see that (31) is equivalent to (27). Similarly, set $P = X^{-1}$ and $K = Y_2X^{-1}$. By pre- and post- multiplying (32) with $\text{diag}\{P, I, \dots, I\}$ and applying Lemma 1, we can see that (32) is equivalent to (28). This completes the proof according to Theorem 2. \square

6 Numerical example

To illustrate the effectiveness of the proposed stabilisation approach, we considered the following system with the plant parameters

$$A = \begin{bmatrix} 1.4 & 1 & 1 & -1.1 \\ -1.3 & -0.9 & 0.5 & 0.5 \\ 0.3 & -0.2 & -1 & 0 \\ -0.5 & -0.3 & -0.5 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} -0.7 & -1 \\ 0 & -0.9 \\ 0.8 & 0.6 \\ 0.1 & 0 \end{bmatrix} \quad (33)$$

$$C = \begin{bmatrix} -1 & 0.6 & -0.3 & 0 \\ 0.5 & 0.5 & 0.2 & -1 \end{bmatrix} \quad (34)$$

The eigenvalues of the plant were 1.1783, -1.1818 and $-0.7482 \pm 0.5171i$. At each instant k , the plant output vector $y(k)$ was transmitted by two packets through the S/C channel and the input vector $u(k)$ was transmitted by two packets via the C/A channel. Thus, the numbers of elements in the sets Q and M were both $2^2 = 4$.

Assume that the above system was under Assumption 1 with

$$\Pi = \begin{bmatrix} 0.1 & 0.2 & 0.2 & 0.5 \\ 0.2 & 0.4 & 0.3 & 0.1 \\ 0.1 & 0.2 & 0.4 & 0.3 \\ 0 & 0.2 & 0.3 & 0.5 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} 0.4 & 0.3 & 0.2 & 0.1 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0.1 & 0.2 & 0.5 & 0.2 \\ 0 & 0.1 & 0.2 & 0.7 \end{bmatrix}$$

By applying the Matlab LMI Control Toolbox to solve (29) and (30), we obtained the solution

$$Y_1 = \begin{bmatrix} 4.4367 & -0.8720 \\ 2.0832 & -0.1689 \\ 3.5632 & -0.0516 \\ -2.1079 & -0.6023 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} 9.4017 & 8.0215 & 5.9738 & -4.8219 \\ 6.7010 & 13.1040 & 12.3461 & -4.2863 \\ 5.2419 & 13.5195 & 33.3694 & 0.5246 \\ -4.6326 & -4.5293 & 0.4402 & 3.8743 \end{bmatrix}$$

$$Y_2 = \begin{bmatrix} -0.0278 & 1.3988 & 1.8676 & 0.0163 \\ 0.7544 & -0.4493 & 0.2046 & -0.5139 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} 6.8887 & -8.8750 & -1.6751 & 1.1331 \\ -8.2399 & 31.4891 & -12.2438 & -1.8277 \\ -2.1167 & -10.7430 & 11.2849 & -0.5062 \\ 1.0734 & -2.1387 & -0.1161 & 0.5962 \end{bmatrix}$$

The matrix solutions for Q_1, Q_2, Q_3 and Q_4 as well as W_1, W_2, W_3 and W_4 were omitted here for space economy. It followed from Theorem 3 that the synthesis of stochastic stabilisation control was solvable by

$$L = G_1^{-1}Y_1 = \begin{bmatrix} 0.3390 & -0.7328 \\ -0.5661 & -0.5095 \\ 0.2960 & 0.3462 \\ -0.8342 & -1.6666 \end{bmatrix}$$

$$K = Y_2G_2^{-1} = \begin{bmatrix} 2.4052 & 1.7665 & 2.4696 & 2.9684 \\ 0.4531 & 0.0690 & 0.1459 & -1.3878 \end{bmatrix}$$

We further considered the memoryless process-driven NCS with the plant parameters given in (33) and (34). Assume that the state probabilities in Assumption 2 were

$$\Pi_M = [0.1 \quad 0.25 \quad 0.25 \quad 0.4]$$

$$\Lambda_M = [0.1 \quad 0.15 \quad 0.25 \quad 0.5]$$

By applying the the Matlab LMI Control Toolbox to solve (31) and (32), we obtained the solution

$$Y_1 = \begin{bmatrix} 55.1808 & -101.4306 \\ 40.8748 & 21.4914 \\ 37.2105 & 19.7553 \\ -12.3726 & -128.3100 \end{bmatrix}$$

$$Q = \begin{bmatrix} 131.4554 & 78.6004 & 57.8315 & 13.7671 \\ 78.6004 & 124.7493 & 120.4823 & -22.2888 \\ 57.8315 & 120.4823 & 249.0080 & -45.2236 \\ 13.7671 & -22.2888 & -45.2236 & 144.0558 \end{bmatrix}$$

$$\mathbf{Y}_2 = \begin{bmatrix} -41.6925 & 151.2122 & 461.0763 & -39.8225 \\ 81.5724 & -364.0850 & 581.1922 & -299.6153 \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} 0.8818 & -0.7804 & -0.0770 & 0.0675 \\ -0.7804 & 1.5382 & -0.4507 & 0.0277 \\ -0.0770 & -0.4507 & 0.8851 & -0.2959 \\ 0.0675 & 0.0277 & -0.2959 & 0.2543 \end{bmatrix}$$

It followed from Theorem 4 that the synthesis of stochastic stabilisation control was solvable by

$$\mathbf{L} = \mathbf{Q}^{-1}\mathbf{Y}_1 = \begin{bmatrix} 0.4013 & -1.2337 \\ 0.0378 & 1.1150 \\ 0.0174 & -0.2998 \\ -0.1129 & -0.6944 \end{bmatrix}$$

$$\mathbf{K} = \mathbf{Y}_2\mathbf{X}^{-1} = \begin{bmatrix} 0.7848 & 0.9029 & 1.4633 & 1.2394 \\ 0.1758 & -0.0109 & 0.4211 & -0.7337 \end{bmatrix}$$

7 Conclusions

In this contribution we have studied a class of NCSs with multiple-packet transmission where random packet dropouts occur independently in both the S/C and C/A channels. Our new contributions have been twofold. Firstly, we have established and proved a separation principle for the generic NCS where packet dropouts in the S/C and C/A channels are driven by two independent Markov chains, respectively. According to this result, the state feedback control and the observer for the underlying NCS can be analysed and synthesised independently. Secondly, we have presented a sufficient condition in the terms of LMI for the stabilisation control of the Markov chain-driven NCSs. We have also derived the necessary and sufficient condition for the stabilisation control of memoryless process-driven NCSs. Finally, A numerical example has been included to illustrate our proposed design approach.

In this study, our results were derived by adopting the zero-input scheme for compensating the dropped control packet. As mentioned in the introduction section, the zero-input scheme might be inapplicable for certain industrial process control applications. An alternative method for compensating control packet dropouts is known to be the hold-input scheme. Currently, we are investigating the conditions for guaranteeing the separation principle and stabilisation control under the hold-input scheme.

8 Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grants No. 60774001, No. 60736021 and No. 60721062), the 973 program of China (Grant No. 2009CB320603), the 863 program of China (Grant No. 2008AA042602), the 111 Project of China (Grant No. B07031) and the UK Royal Academy of Engineering.

9 References

- 1 Antsaklis, P.J., Michel, A.N.: 'Linear systems' (Springer, Boston, 2006, 2nd edn.)
- 2 Maybeck, P.S.: 'Stochastic models, estimation, and control' (Academic Press, New York, 1979), vol. 1
- 3 Atassi, A.N., Khalil, H.K.: 'A separation principle for the stabilization of a class of nonlinear systems', *IEEE Trans. Autom. Control*, 1999, **44**, pp. 1672–1687
- 4 Hespanha, J.P., Naghshtabrizi, P., Xu, Y.-G.: 'A survey of recent results in networked control systems', *Proc. IEEE*, 2007, **95**, pp. 138–162
- 5 Schenato, L., Sinopoli, B., Franceschetti, M., Poolla, K., Sastry, S.S.: 'Foundations of control and estimation over lossy networks', *Proc. IEEE*, 2007, **95**, pp. 163–187
- 6 Zhivoglyadov, P.V., Middleton, R.H.: 'Networked control design for linear systems', *Automatica*, 2003, **39**, pp. 743–750
- 7 Montestruque, L.A., Antsaklis, P.: 'Stability of model-based networked control systems with time-varying transmission times', *IEEE Trans. Autom. Control*, 2004, **49**, pp. 1562–1572
- 8 Ji, Y., Chizeck, H.J., Feng, X., Loparo, K.A.: 'Stability and control of discrete-time jump linear systems', *Control Theory Adv. Technol.*, 1991, **7**, pp. 247–270
- 9 Costa, O.L.V., Fragoso, M.D.: 'Stability results for discrete-time linear systems with Markovian jumping parameters', *J. Math. Anal. Appl.*, 1993, **179**, pp. 154–178
- 10 Imer, O.C., Yüksel, S., Başar, T.: 'Optimal control of LTI systems over unreliable communication links', *Automatica*, 2006, **42**, pp. 1429–1439
- 11 Hu, S., Yan, W.-Y.: 'Stability of networked control systems under a multiple-packet transmission policy', *IEEE Trans. Autom. Control*, 2008, **53**, pp. 1706–1711
- 12 Wu, J., Chen, T.: 'Design of networked control systems with packet dropouts', *IEEE Trans. Autom. Control*, 2007, **52**, pp. 1314–1319
- 13 Seiler, P., Sengupta, R.: 'An H_∞ approach to networked control', *IEEE Trans. Autom. Control*, 2005, **50**, pp. 356–364
- 14 Zhang, W.-A., Yu, L.: 'Output feedback stabilization of networked control systems with packet dropouts', *IEEE Trans. Autom. Control*, 2007, **52**, pp. 1705–1710
- 15 Ishii, H.: ' H^∞ control with limited communication and message losses', *Syst. Control Lett.*, 2008, **57**, pp. 322–331
- 16 Quevedo, D.E., Silva, E.I., Goodwin, G.C.: 'Packetized predictive control over erasure channels'. Proc. 2007 American Control Conf., New York, USA, 9–13 July 2007, pp. 1003–1008
- 17 Muñoz de la Peña, D., Christofides, P.D.: 'Lyapunov-based model predictive control of nonlinear systems subject to data losses', *IEEE Trans. Autom. Control*, 2008, **53**, pp. 2076–2089
- 18 Ling, Q., Lemmon, M.D.: 'Optimal dropout compensation in networked control systems'. Proc. 42nd IEEE Conf. on Decision and Control, Maui, Hawaii, USA, 9–12 December 2003, pp. 670–675
- 19 Tian, Y.-C., Levy, D.: 'Compensation for control packet dropout in networked control systems', *Inf. Sci.*, 2008, **178**, pp. 1263–1278
- 20 Schenato, L.: 'Optimal estimation in networked control systems subject to random delay and packet loss'. Proc. 45th IEEE Conf. on Decision and Control, San Diego, CA, USA, 13–15 December 2006, pp. 5615–5620
- 21 Soglo, A.B., Yang, X.: 'Compensation for network-induced delays and packet dropping in control systems', *Eur. J. Control*, 2006, **12**, pp. 296–306
- 22 Hristu-Varsakelis, D.: 'Short-period communication and the role of zero-order holding in networked control systems', *IEEE Trans. Autom. Control*, 2008, **53**, pp. 1285–1290
- 23 Schenato, L.: 'To zero or to hold control inputs with lossy links?', *IEEE Trans. Autom. Control*, 2009, **54**, pp. 1093–1099
- 24 Sala, A.: 'Computer control under time-varying sampling period: an LMI gridding approach', *Automatica*, 2005, **41**, pp. 2077–2082
- 25 Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V.: 'Linear matrix inequalities in systems and control theory' (SIAM, Philadelphia, PA, 1994)
- 26 Karlin, S., Taylor, H.M.: 'A first course in stochastic processes' (Academic Press, New York, 1975, 2nd edn.)
- 27 de Oliveira, M.C., Bernussou, J., Geromel, J.C.: 'A new discrete-time robust stability condition', *Syst. Control Lett.*, 1999, **37**, pp. 261–265
- 28 Yu, M., Wang, L., Chu, T., Xie, G.: 'Modelling and control of networked systems via jump system approach', *IET Control Theory Appl.*, 2008, **2**, pp. 535–541