



Research article

Stability analysis of networked control systems with bounded random delay and state compensation: How large is the actual system scale?

Yipeng Yang*

Department of Mathematics and Statistics, University of Houston - Clear Lake, Houston, TX 77058

* **Correspondence:** Email: yangy@uhcl.edu; Tel: +12812833737.

Abstract: In order to overcome the constraints of Networked Control Systems (NCSs) such as random packet delays or dropouts, it is a natural idea to estimate the system state and compensate for the time delays on the controller side. This paper provides an estimate on the scale of the complete dynamical system that uses this idea of control. The structure of the complete system is clearly illustrated. Then a concise sufficient and necessary stability condition is provided. In the numerical example, it is shown that this seemingly small system turns out to have a very large scale.

Keywords: networked control system; regime switching system; random delay; state compensation; stability analysis

1. Introduction and preliminaries

This work is motivated by the networked control system (NCS) or remote control system where the media that connects the controller and actuator causes random packet delays or dropouts. This type of control systems are anticipated to have vast applications. The readers are referred to the review paper [1]. The common features of NCS are random packet delays or dropouts, competition of multiple nodes in the network, data quantization, etc., and this makes the modeling and stability analysis very challenging. Markovian type regime switching models have been proved to be very successful in modeling NCS [2, 3]. However, the stability of such systems still remains a challenge. Most stability criteria are given as complicated LMI conditions [4, 5] via Lyapunov function approach, or dwell time [6], or delay Riccati equations [7]. A stability condition using hybrid system analysis is found in [8], which is again a sufficient condition and is not easy to verify. A neat sufficient and necessary condition is in great need.

In order to overcome the constraints of NCS, such as random packet delays or dropouts, it is a natural idea to estimate system state and compensate for packet disorders [9]. One believes that in this way better system performance and better system stability can be achieved. Predictive control has

been effectively applied to NCS [10–13]. This control method generates a sequence of future control variables, which can be used to compensate for packet disorder or dropouts [14], and to estimate unknown system state as well [15, 16]. However, as mentioned earlier, the stability conditions are usually hard to verify. To analyze the stability issue, one must have a thorough understanding of the dynamics of NCS with random time delays and state estimate.

In this paper we propose a predictive control method of NCS with random packet delays and state estimate/compensation, and clearly show the structure of this system. This control method has much less computational burden compared to classical predictive control method. Then the system is modeled as a regime switching system, and an upper bound on the number of regimes is provided. By the word “scale” we mean the number of distinct regimes (denoted by M), and the size of minimal state representation in each regime (denoted by d). Then we provide a concise stability criterion that is sufficient and necessary. Both M and d play a role in the validation of this criterion (see Section 4).

Throughout this paper we use $\mathbf{x}_k, \mathbf{u}_k$ to denote the system state on the actuator side, and the control variable, respectively, where k is time index. Note that this is not the state representation in the overall system. The controller and actuator are connected through a network. The random time delay in actuator-to-controller channel is called the backward delay, denoted by τ_1 . The delay in the controller-to-actuator channel is called the forward delay, denoted by τ_2 . Markov models have been shown to be very general for the transition of these random delays [13], which we shall adopt.

Unlike most NCSs where the objective systems on the actuator side are time invariant, here we assume that the objective system is a regime switching system with Markovian jumps, and we call it a Markov Jump System (MJS). The application of switching systems in engineering can be found, for example, in [17–19]. We let $i_k \in \{1, 2, \dots, m\}$ denote the system mode/regime at time k . P_m with size $m \times m$ denotes the transition probability matrix among the m modes. Entries in P_m are denoted by λ_{ij} 's, $i, j \in \{1, 2, \dots, m\}$. Thus, this NCS has another layer of complexity.

We begin with simple and practically reasonable assumptions on this system.

- **The actuator receives signals with random delays.**

The actuator receives control signals from the controller and performs the required actions. Due to the random packet delays, the actuator might receive no signal, or receive a packet that is randomly delayed, or receive multiple packets. If more than one packet arrive at the same time, the actuator executes the most recent one.

- **Obsolete signals are discarded.**

If, at a certain time instant, a control variable has been applied, then any control signal older than it but arrives at later times will be discarded. For example, if \mathbf{u}_5 has been applied to the system at the present time and at the next time step \mathbf{u}_4 arrives, then \mathbf{u}_4 is discarded.

- **The delay times are bounded.**

The backward time delay τ_1 , and the forward time delay τ_2 are independent and bounded. This can be done by designing a transmission protocol such that, when a packet has been delayed for a certain amount of time, it is upgraded to the highest priority at a relay station and is transmitted immediately. Therefore we assume that $\tau_1 \in \{0, 1, \dots, T_1\}$ and $\tau_2 \in \{0, 1, 2, \dots, T_2\}$.

- **The controller sends out signals at every discrete instant.**

The controller receives state feedback from the actuator via randomly delayed packets, and the controller estimates the future system state and sends out a new control signal. If no state feedback is received at a certain time instant, the controller simply sends a zero control signal.

- **Past control signals are recorded.**

The controller keeps a record of past control signals that have been sent. This information is used to calculate a new control signal whenever a feedback packet arrives.

- **Packet size is limited.**

The network bandwidth is very limited so that each time the controller and actuator just send out one small packet to the other party. That means, sending out a large packet containing a long sequence of controls or system states is not applicable. (Otherwise this system reduces to classical control system without time delay.)

The controlled system on the actuator side becomes

$$\mathbf{x}_{k+1} = \mathbf{a}(i_k)\mathbf{x}_k + \mathbf{b}^{(j)}(i_k)\mathbf{u}_{k-\tau_2}, \quad (1.1)$$

where k is present time, and the system dynamics $\mathbf{a}(i_k)$, $\mathbf{b}^{(j)}(i_k)$ depend on the mode i_k . The superscript j is an indicator function such that if $j = 0$, $\mathbf{b}^{(j)}(i_k) = \mathbf{0}$ which means no control signal is received; and if $j = 1$, $\mathbf{b}^{(j)}(i_k) = \mathbf{b}(i_k)$, $i_k \in \{1, 2, \dots, m\}$ meaning at least one (possibly delayed) control signal is received. In the complete state representation to be introduced shortly it can be seen that the information of j is absorbed by a variable $C_2^{(k)}$. Recall that the control variable \mathbf{u}_k depends on delayed feedback $\mathbf{x}_{k-\tau_1}$ and $i_{k-\tau_1-1}$, and the controller does not know the system information between $k - \tau_1 - 1$ and k , except the past controls up to \mathbf{u}_{k-1} . In what follows we shall show that this seemingly simple model actually has a very large scale.

2. The scale of NCS with random delay and state estimate

The NCS is a special dynamical system that might be easy to describe in words, but is hard to understand because its dynamics looks chaotic. One must get down to the bottom and have a careful dissemination to obtain a clear understanding. In what follows we shall do this work.

We use the triplet $(C_2^{(k)}, \tau_2^{(k)}, i_k)$ to denote the situation at the actuator side at time k , where $C_2^{(k)} \in \{0, 1, 2, \dots, T_2\}$. Without the script k , C_2, τ_2, i just denote generic information, respectively.

If no control signal arrives at time k but there was one packet arrived at $k - 1$, it means that this is a new round of receiving no signal, hence $C_2^{(k)} = 1$. And at the same time we set $\tau_2^{(k)}$ in this triplet equal to its counterpart in the proceeding packet, i.e., $\tau_2^{(k)} = \tau_2^{(k-1)}$. Thus the information of the last applied control index is recorded. If at the next time step, say $k + 1$, no control signal arrives, $C_2^{(k+1)} = 2$, $\tau_2^{(k+1)} = \tau_2^{(k)}$ and so forth. If a new packet arrives at time $k + 2$, we set $C_2^{(k+2)} = 0$ and $\tau_2^{(k+2)}$ equals the time delay in control in this received packet. Recall that the forward time delay is bounded. So before or by the time $C_2 + \tau_2 = T_2$, a new control signal is guaranteed to arrive. And finally i_k denotes the mode of the system at time k . For example, $(3, \tau_2^{(k)}, i_k)$ denotes the situation that for the past 3 time instants, no control signal was received, and the last applied control was delayed by $\tau_2^{(k)}$ (the total delay at this time is $3 + \tau_2^{(k)}$). $(0, 2, i_k)$ denotes the situation that at least one control signal arrives, and \mathbf{u}_{k-2} is the most recent one (if more than one signal have arrived).

The expression $(1, \tau_2^{(k)}, i_k) \rightarrow (2, \tau_2^{(k+1)}, i_{k+1})$ represents a state transition. It should be pointed out that the states $(C_2^{(k)}, \tau_2^{(k)}, i_k)$ do not have full reachability. For example, if the current state is $(0, 0, i_k)$, the next one cannot be $(0, 1, i_{k+1})$. That is because, the state $(0, 0, i_k)$ indicates that at time k the control \mathbf{u}_k arrived with no delay and was applied to the system, while $(0, 1, i_{k+1})$ means that at time $k + 1$, the

one step delayed control (which is again \mathbf{u}_k) arrives and is to be applied. But this is a contradiction. The transition probability matrix for the transitions $(C_2^{(k)}, \tau_2^{(k)}) \rightarrow (C_2^{(k+1)}, \tau_2^{(k+1)})$ is denoted by P_2 .

We use a similar mechanism to represent the situation at the controller side. The pair $(C_1^{(k)}, \tau_1^{(k)})$ is constructed where $C_1^{(k)} \in \{0, 1, 2, \dots, T_1\}$. If $C_1^{(k)} = 0$, that means the controller receives a feedback packet from the actuator and the time delay in this packet is $\tau_1^{(k)}$, which is the most recent one if multiple packets have arrived. If $C_1^{(k)} > 0$, that means no feedback packet arrives and $C_1^{(k)}$ is a counter recording the time steps of receiving no signal, while $\tau_1^{(k)}$ equals its counterpart in the previously received packet. Because the backward time delay is bounded, before or by the time $C_1 + \tau_1 = T_1$ a feedback packet is guaranteed to arrive. The transition probability matrix for the transitions $(C_1^{(k)}, \tau_1^{(k)}) \rightarrow (C_1^{(k+1)}, \tau_1^{(k+1)})$ is denoted by P_1 .

Notice that the received information depends on the sequence of the past states $(C_2^{(j)}, \tau_2^{(j)}, i_j)$, $j < k$, from the actuator side. Then we have the following proposition on the number of different regimes/scenarios of this control system.

Proposition 2.1. *An upper bound on the number of different regimes of this problem (denoted by M), or equivalently, an upper bound on the number of different scenarios of the system, is given by*

$$m^{2+T_1} \cdot (1 + T_2)^{2+T_1} \cdot (2 + T_2)/2 \cdot (1 + T_1) \cdot (2 + T_1)/2 \quad (2.1)$$

Proof. On the controller side we consider the pair $(C_1^{(k)}, \tau_1^{(k)})$, where $C_1^{(k)}, \tau_1^{(k)}$ both take values in $\{0, 1, 2, \dots, T_1\}$. Notice that any combination of $C_1^{(k)}, \tau_1^{(k)}$ such that $C_1^{(k)} + \tau_1^{(k)} \leq T_1$ is possible, thus there are $1 + 2 + \dots + (1 + T_1) = (1 + T_1)(2 + T_1)/2$ different scenarios in total for the pair $(C_1^{(k)}, \tau_1^{(k)})$.

Similarly, on the actuator side, any pair (C_2, τ_2) satisfying $C_2 + \tau_2 \leq T_2$, $C_2, \tau_2 \in \{0, 1, 2, \dots, T_2\}$, is a possible scenario. There are $1 + 2 + \dots + (1 + T_2) = (1 + T_2)(2 + T_2)/2$ different combinations. Furthermore, $i_k \in \{1, 2, \dots, m\}$ also determines the system dynamics. So there are $m \cdot ((1 + T_2)T_2/2 + 1 + T_2) = m \cdot (1 + T_2) \cdot (2 + T_2)/2$ different cases.

Due to the feedback delay, a sequence with length $1 + T_1$ of past system states $(C_2^{(j)}, \tau_2^{(j)}, i_j)$, $j = k - T_1 - 1, k - T_1, \dots, k - 1$, must be maintained. Thus, a state $(C_1^{(k)}, \tau_1^{(k)}) = (0, \tau_1^{(k)})$ indicates that the controller receives the information $\mathbf{x}_{k-\tau_1^{(k)}}, C_2^{(k-\tau_1^{(k)}-1)}, \tau_2^{(k-\tau_1^{(k)}-1)}$ and $i_{k-\tau_1^{(k)}-1}$ at time k .

The controller then calculates and sends out a new control variable \mathbf{u}_k based on this received information. When or whether will this control be applied by the actuator depends on the situation $(C_2^{(k)}, \tau_2^{(k)}, i_k)$ at time k (which the controller does not know), and this contributes another triplet to the representation. That means, there should be a total of $2 + T_1$ of the triplets (C_2, τ_2, i) in the representation of the system regime.

However, recall that not all the transitions $(C_2^{(k)}, \tau_2^{(k)}, i_k) \rightarrow (C_2^{(k+1)}, \tau_2^{(k+1)}, i_{k+1})$ are possible. The sum $C_2^{(k)} + \tau_2^{(k)}$ (which is less than or equal to T_2) indicates the time index of the last applied control, and only the controls newer than it are likely to be accepted by the actuator. When the time index shifts to the next time instant, it automatically gives one more spot for future control. Therefore, given a triplet $(C_2^{(k)}, \tau_2^{(k)}, i_k)$ in a sequence, there are at most $m \cdot (1 + T_2)$ successors. Thus, given that the first triplet has $m \cdot (1 + T_2) \cdot (2 + T_2)/2$ possibilities, the rest $1 + T_1$ triplets each has $m \cdot (1 + T_2)$ possibilities. This contributes

$$m \cdot (1 + T_2) \cdot (2 + T_2)/2 \cdot [m \cdot (1 + T_2)]^{1+T_1} = m^{2+T_1} \cdot (1 + T_2)^{2+T_1} \cdot (2 + T_2)/2.$$

Now the proof is complete. □

It is clear that to represent the complete system scenario, we need $2 + T_1$ triplets (C_2, τ_2, i) and a pair (C_1, τ_1) . We call this complete scenario/regime representation (R) .

3. State estimate and new control signal

3.1. State estimate

In each scenario in (R) , we need to represent the system dynamics. Firstly we write a column vector \mathbf{z}_k as follows:

$$\mathbf{z}_k = [\mathbf{x}_k(i_k), \mathbf{x}_{k-1}(i_{k-1}), \dots, \mathbf{x}_{k-T_1}(i_{k-T_1}), \mathbf{u}_{k-1}, \mathbf{u}_{k-2}, \dots, \mathbf{u}_{k-T_1-T_2}]^T, \quad (3.1)$$

where the superscript T means transpose, then we get

$$\mathbf{z}_{k+1} = \mathbf{A}(C_2^{(k)}, \tau_2^{(k)}, i_k)\mathbf{z}_k + \mathbf{B}(C_2^{(k)}, \tau_2^{(k)}, i_k)\mathbf{u}_k, \quad (3.2)$$

which is an extension of (1.1). Since each entry in (3.1) is necessary to describe the system dynamics in a certain regime, we are clear that d (as mentioned in Section 1) equals the length of \mathbf{z}_k . Notice that if $T_1 = T_2 = 0$, there is no need to keep any control variable in \mathbf{z}_k (and \mathbf{z}_k reduces to \mathbf{x}_k). The matrices \mathbf{A}, \mathbf{B} are explained in the following.

If $C_2^{(k)} > 0$, that means no control signal is received at time k , we simply write

$$\mathbf{A}(C_2^{(k)}, \tau_2^{(k)}, i_k) = \begin{pmatrix} \mathbf{a}(i_k) & 0 & 0 & 0 & 0 \cdots \\ & 1 & 0 & 0 & 0 \cdots \\ & 0 & 1 & 0 & 0 \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0}^{(*)} \cdots & 0 & 0 & 0 & 0 \cdots \\ & 0 \cdots & 0 & 1 & 0 \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix},$$

$$\mathbf{B}(C_2^{(k)}, \tau_2^{(k)}, i_k) = [0 \ 0 \ 0 \ \cdots \ 1^{(*)} \ 0 \ 0 \ \cdots]^T,$$

where $(*)$ corresponds to the $T_1 + 2$ nd row. Indeed it corresponds to the entry \mathbf{u}_k in \mathbf{z}_{k+1} on the left side of (3.2). The information $C_2^{(k)}$ affects the scenario representation in (R) only and plays a role as a counter, and it does not alter the system dynamics. $\tau_2^{(k)}$ records the delay in the last received control packet.

If $C_2^{(k)} = 0$, it means at least one control signal is received, then there are two cases. If $\tau_2^{(k)} = 0$, \mathbf{u}_k arrives at the actuator side without delay, then $\mathbf{A}(0, 0, i_k)$ remains the same as $\mathbf{A}(C_2^{(k)}, \tau_2^{(k)}, i_k)$ shown above, and $\mathbf{B}(0, 0, i_k) = [\mathbf{b}(i_k) \ 0 \ 0 \ \cdots \ 1^{(*)} \ 0 \ 0 \ \cdots]^T$.

If $\tau_2^{(k)} > 0$ while $C_2^{(k)} = 0$, then

$$\mathbf{A}(0, \tau_2^{(k)}, i_k) = \begin{pmatrix} \mathbf{a}(i_k) & 0 & \dots & \mathbf{b}(i_k) & 0 & 0 & \dots \\ 1 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 0 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0^{(*)} & \dots & 0 & 0 & 0 & 0 & \dots \\ 0 & \dots & 0 & 1 & 0 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix},$$

$$\mathbf{B}(0, \tau_2^{(k)}, i_k) = [0 \ 0 \ 0 \ \dots \ 1^{(*)} \ 0 \ 0 \ \dots]^T,$$

where the position of $\mathbf{b}(i_k)$ corresponds to the entry $\mathbf{u}_{k-\tau_2^{(k)}}$ in \mathbf{z}_k .

On the controller side we have

$$\mathbf{u}_k = -\mathbf{F}(C_1^{(k)}, \tau_1^{(k)}, C_2^{(k-\tau_1^{(k)}-1)}, \tau_2^{(k-\tau_1^{(k)}-1)}, i_{k-\tau_1^{(k)}-1})\mathbf{z}_k. \tag{3.3}$$

If $C_1^{(k)} > 0$, it means no feedback packet is received at time k , so $\mathbf{F} = \mathbf{0}$ and $C_1^{(k)}$ plays the role as a counter. If $C_1^{(k)} = 0$, the controller receives at least one feedback packet among which the most recent one is $\mathbf{x}_{k-\tau_1^{(k)}}$, together with the last applied control whose index is $k - \tau_1^{(k)} - \tau_2^{(k-\tau_1^{(k)}-1)} - 1$. Based on the information $\mathbf{x}_{k-\tau_1^{(k)}}$, $i_{k-\tau_1^{(k)}-1}$, $\tau_2^{(k-\tau_1^{(k)}-1)}$ and the past controls

$$\mathbf{u}_{k-\tau_1^{(k)}-\tau_2^{(k-\tau_1^{(k)}-1)}}, \mathbf{u}_{k-\tau_1^{(k)}-\tau_2^{(k-\tau_1^{(k)}-1)}+1}, \dots, \mathbf{u}_{k-1}, \tag{3.4}$$

the controller is able to estimate the system state \mathbf{x}_k and sends out a new control variable \mathbf{u}_k . It is thus given in the feedback form (3.3). Notice that when $C_1^{(k)} = 0$, $\tau_1^{(k)}$ is the only random variable: it determines which past triplet $(C_2^{(j)}, \tau_2^{(j)}, i_j)$ in (R) is received by the controller. The controller also keeps a record of $\tau_1^{(k)}$ in the last received packet such that any future arrivals with indices older than $k - \tau_1^{(k)}$ will be discarded.

Recall that if the controller does not receive any feedback packet at present time k , it simply sends $\mathbf{u}_k = 0$. If the controller receives a packet that contains $\mathbf{x}_{k-\tau_1^{(k)}}$ with the triplet $(C_2^{(k-\tau_1^{(k)}-1)}, \tau_2^{(k-\tau_1^{(k)}-1)}, i_{k-\tau_1^{(k)}-1})$, and if the time index $k - \tau_1^{(k)}$ is the newest, it accepts this packet. Then the controller estimates the system state \mathbf{x}_k and calculates a new control \mathbf{u}_k that is to be sent.

We need to point out that there is much flexibility in calculating the new control variable, which might be the most resourceful research in NCS. In this paper we present our control method as follows. Firstly, based on the most recent information $\mathbf{x}_{k-\tau_1^{(k)}}$, $C_2^{(k-\tau_1^{(k)}-1)}$, $\tau_2^{(k-\tau_1^{(k)}-1)}$, $i_{k-\tau_1^{(k)}-1}$, where k is present time, the controller knows that the last applied control to the system was $\mathbf{u}_{k-\tau_1^{(k)}-\tau_2^{(k-\tau_1^{(k)}-1)}-1}$, and any controls older than it will be discarded by the actuator, while the controls $\mathbf{u}_{k-\tau_1^{(k)}-\tau_2^{(k-\tau_1^{(k)}-1)}}, \mathbf{u}_{k-\tau_1^{(k)}-\tau_2^{(k-\tau_1^{(k)}-1)}+1}, \dots, \mathbf{u}_{k-1}$ are either on their way, or have been (possibly partially) applied by time k . Since this sequence of controls were sent out in order, it is reasonable to assume

that they were or will be applied to the system one by one. A letter with a hat denotes the estimate of that corresponding quantity. Thus, $\hat{\mathbf{x}}_{k-\tau_1^{(k)}+1}$ is estimated at

$$\hat{\mathbf{x}}_{k-\tau_1^{(k)}+1} = \mathbf{a}(\hat{i}_{k-\tau_1^{(k)}})\mathbf{x}_{k-\tau_1^{(k)}} + \mathbf{b}(\hat{i}_{k-\tau_1^{(k)}})\mathbf{u}_{k-\tau_1^{(k)}-\tau_2^{(k-\tau_1^{(k)}-1)}, \quad (3.5)$$

and $\hat{\mathbf{x}}_{k-\tau_1^{(k)}+2}$ is estimated at

$$\hat{\mathbf{x}}_{k-\tau_1^{(k)}+2} = \mathbf{a}(\hat{i}_{k-\tau_1^{(k)}+1})\hat{\mathbf{x}}_{k-\tau_1^{(k)}+1} + \mathbf{b}(\hat{i}_{k-\tau_1^{(k)}+1})\mathbf{u}_{k-\tau_1^{(k)}-\tau_2^{(k-\tau_1^{(k)}+1)}+1}, \quad (3.6)$$

and so forth until \mathbf{u}_{k-1} is applied to yield $\hat{\mathbf{x}}_{k+\tau_2^{(k-\tau_1^{(k)}-1)}}$.

One may notice that the system mode information $\hat{i}_{k-\tau_1^{(k)}}, \hat{i}_{k-\tau_1^{(k)}+1}, \dots, \hat{i}_{k+\tau_2^{(k-\tau_1^{(k)}-1)}-1}$ are not known by the controller. So the controller needs to predict a future path of modes.

Based on the probability transition matrix P_m and the received mode $i_{k-\tau_1^{(k)}-1}$, one can easily calculate the probability for each path to occur. We hope to determine one path over which we may estimate system states up to $\hat{\mathbf{x}}_{k+\tau_2^{(k-\tau_1^{(k)}-1)}}$, so we apply a greedy search algorithm as follows. Starting from mode $i_{k-\tau_1^{(k)}-1}$, we search in P_m the next connected mode $\hat{i}_{k-\tau_1^{(k)}}$ so that $\lambda_{i_{k-\tau_1^{(k)}-1}, \hat{i}_{k-\tau_1^{(k)}}}$ is the maximum in the row of P_m corresponding to mode $i_{k-\tau_1^{(k)}-1}$. If there is a tie, we just randomly choose one mode. Then starting from $\hat{i}_{k-\tau_1^{(k)}}$ we search in P_m the next connected mode $\hat{i}_{k-\tau_1^{(k)}+1}$ so that $\lambda_{\hat{i}_{k-\tau_1^{(k)}}, \hat{i}_{k-\tau_1^{(k)}+1}}$ is the maximum in the row of P_m corresponding to the mode $\hat{i}_{k-\tau_1^{(k)}}$. We repeat this process to obtain a sequence of future modes $\hat{i}_{k-\tau_1^{(k)}}, \hat{i}_{k-\tau_1^{(k)}+1}, \dots, \hat{i}_{k+\tau_2^{(k-\tau_1^{(k)}-1)}-1}$. With the past controls (3.4) we obtain an estimate for $\hat{\mathbf{x}}_{k+\tau_2^{(k-\tau_1^{(k)}-1)}}$ via (3.5) and (3.6) recursively.

3.2. New control signal

To calculate a new control \mathbf{u}_k , we apply the idea of predictive control, which is also referred to as receding horizon control (RHC). Firstly we extend the prediction of future modes to get

$$\hat{i}_{k-\tau_1^{(k)}}, \hat{i}_{k-\tau_1^{(k)}+1}, \dots, \hat{i}_{k+\tau_2^{(k-\tau_1^{(k)}-1)}-1}, \hat{i}_{k+\tau_2^{(k-\tau_1^{(k)}-1)}}, \dots, \hat{i}_{k-\tau_1^{(k)}+L-1}, \quad (3.7)$$

where $L (> T_1 + T_2)$ is the total length of prediction. The future controls $\hat{\mathbf{u}}_k, \hat{\mathbf{u}}_{k+1}, \dots, \hat{\mathbf{u}}_{k+L-\tau_1^{(k)}-\tau_2^{(k-\tau_1^{(k)}-1)}-1}$ are chosen to minimize the following cost function:

$$\begin{aligned} J(\hat{\mathbf{x}}_{k+\tau_2}, \hat{i}_{k+\tau_2^{(k-\tau_1^{(k)}-1)}-1}, \tau_1^{(k)}, \tau_2^{(k-\tau_1^{(k)}-1)}) \\ = \sum_{j=\tau_2^{(k-\tau_1^{(k)}-1)}+1}^{L-\tau_1^{(k)}} \hat{\mathbf{x}}_{k+j}^T \mathbf{Q}(k+j) \hat{\mathbf{x}}_{k+j} + \sum_{j=0}^{L-\tau_1^{(k)}-\tau_2^{(k-\tau_1^{(k)}-1)}-1} \hat{\mathbf{u}}_{k+j}^T \mathbf{R}(k+j) \hat{\mathbf{u}}_{k+j}, \end{aligned} \quad (3.8)$$

where \mathbf{Q}, \mathbf{R} 's are symmetric positive definite (SPD) matrices. A sequence of future controls are thus determined and the control \mathbf{u}_k to be sent equals the first control variable in this sequence, i.e., $\mathbf{u}_k = \hat{\mathbf{u}}_k$.

Unlike the usual cost function which is the expected value along all possible future paths, this cost function (3.8) is calculated along the deterministic path (3.7), and because the optimization is performed on one path, a huge amount of calculation is saved. To see this in more details, let us consider a control method which chooses future controls $\hat{\mathbf{u}}_k, \hat{\mathbf{u}}_{k+1}, \dots, \hat{\mathbf{u}}_{k+L-\tau_1^{(k)}-\tau_2^{(k-1)}-1}$ to minimize the following expected cost function:

$$\begin{aligned}
 & J(\hat{\mathbf{x}}_{k+\tau_2}, \hat{\mathbf{i}}_{k+\tau_2}^{(k-\tau_1^{(k)}-1)}, \tau_1^{(k)}, \tau_2^{(k-\tau_1^{(k)}-1)}) \\
 &= \sum_{j=\tau_2^{(k-\tau_1^{(k)}-1)}+1}^{L-\tau_1^{(k)}} E \left[\hat{\mathbf{x}}_{k+j}^T Q(k+j) \hat{\mathbf{x}}_{k+j} | \mathbf{x}_{k-\tau_1^{(k)}} \right] + \sum_{j=0}^{L-\tau_1^{(k)}-\tau_2^{(k-\tau_1^{(k)}-1)}-1} \hat{\mathbf{u}}_{k+j}^T R(k+j) \hat{\mathbf{u}}_{k+j},
 \end{aligned} \tag{3.9}$$

where $E[\cdot]$ denotes conditional expectation. Since the system has m modes, from $k-\tau_1^{(k)}$ to $K+L-\tau_1^{(k)}$ the calculation will have to consider m^L different paths. Furthermore, this calculation is based on the assumption that the past controls that have been sent before will be applied to the actuator one by one in order, which is certainly not guaranteed in NCS with random time delay. Even if this calculation is carried out to send the new control variable \mathbf{u}_k , this control may not be applied to the actuator at expected instant. Therefore, it looks to have no advantage to carry out this heavy calculation to optimize the expected cost as in classical predictive control.

But for our proposed simplified predictive control method, a natural question remains: what if the path we predicted does not actually occur? Remember that so far we have not used rolling optimization which is the true power of RHC. The calculation of future control sequences is repeated at each sample instant, and if a prediction error occurs, it will be corrected by re-planning at the next sample instant.

Actually we can make this calculation faster by noticing that the control format in RHC with quadratic criterion function (3.8) is given by linear feedback form, see, e.g., [20]. We borrow the iterative algorithm in [20] for the calculation of state feedback control in time varying case, then we obtain

$$\mathbf{u}_k = -F(\hat{\mathbf{i}}_{k+\tau_2}^{(k-\tau_1^{(k)}-1)}, \tau_1^{(k)}, \tau_2^{(k-\tau_1^{(k)}-1)}) \hat{\mathbf{x}}_{k+\tau_2}^{(k-\tau_1^{(k)}-1)}. \tag{3.10}$$

Recall that $\hat{\mathbf{x}}_{k+\tau_2}^{(k-\tau_1^{(k)}-1)}$ was determined by $\mathbf{x}_{k-\tau_1^{(k)}}, \hat{\mathbf{i}}_{k-\tau_1^{(k)}-1}$ and the past controls (3.4), via (3.5) and (3.6) recursively. So we construct the following matrix

$$D_{k-\tau_1^{(k)}+1} = \begin{pmatrix} 0 & \dots & 0 & \mathbf{a}(\hat{\mathbf{i}}_{k-\tau_1^{(k)}}) & 0 & \dots & 0 & \mathbf{b}(\hat{\mathbf{i}}_{k-\tau_1^{(k)}}) & \dots \\ 0 & \dots & \dots & \dots & \dots & 0 & 1 & 0 & \dots \\ 0 & \dots & \dots & \dots & \dots & 1 & 0 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix},$$

where $\mathbf{a}(\hat{\mathbf{i}}_{k-\tau_1^{(k)}}), \mathbf{b}(\hat{\mathbf{i}}_{k-\tau_1^{(k)}})$ in the first row correspond to the entries $\mathbf{x}_{k-\tau_1^{(k)}}$ and $\mathbf{u}_{k-\tau_1^{(k)}-\tau_2^{(k-\tau_1^{(k)}-1)}}$ in \mathbf{z}_k , respectively. The entry 1 in the second row corresponds to the entry $\mathbf{u}_{k-\tau_1^{(k)}-\tau_2^{(k-\tau_1^{(k)}-1)}+1}$ in \mathbf{z}_k , and so forth up to the entry \mathbf{u}_{k-1} . Then

$$D_{k-\tau_1^{(k)}+1} \mathbf{z}_k = (\hat{\mathbf{x}}_{k-\tau_1^{(k)}+1} \mathbf{u}_{k-\tau_1^{(k)}-\tau_2^{(k-\tau_1^{(k)}-1)}+1} \dots \mathbf{u}_{k-1})^T$$

by (3.5).

Now we define

$$D_{k-\tau_1^{(k)}+2} = \begin{pmatrix} \mathbf{a}(\hat{i}_{k-\tau_1^{(k)}+1}) & \mathbf{b}(\hat{i}_{k-\tau_1^{(k)}+1}) & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & \cdots \\ 0 & 0 & 0 & 1 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix},$$

and get

$$D_{k-\tau_1^{(k)}+2} D_{k-\tau_1^{(k)}+1} \mathbf{z}_k = (\hat{\mathbf{x}}_{k-\tau_1^{(k)}+2} \mathbf{u}_{k-\tau_1^{(k)}-\tau_2}^{(k-\tau_1^{(k)-1)}+2} \cdots \mathbf{u}_{k-1})^T$$

by (3.6). Continue this process until we reach $D_{k+\tau_2}^{(k-\tau_1^{(k)-1})}$ so that

$$D_{k+\tau_2}^{(k-\tau_1^{(k)-1})} \cdots D_{k-\tau_1^{(k)}+1} \mathbf{z}_k = \hat{\mathbf{x}}_{k+\tau_2}^{(k-\tau_1^{(k)-1})}. \quad (3.11)$$

Substituting (3.11) in (3.10) yields

$$\mathbf{u}_k = -F(\hat{i}_{k+\tau_2}^{(k-\tau_1^{(k)-1})-1}, \tau_1^{(k)}, \tau_2^{(k-\tau_1^{(k)-1})}) \cdot D_{k+\tau_2}^{(k-\tau_1^{(k)-1})} \cdots D_{k-\tau_1^{(k)}+1} \mathbf{z}_k.$$

Since $\hat{i}_{k+\tau_2}^{(k-\tau_1^{(k)-1})-1}$ is determined by $i_{k-\tau_1^{(k)-1}}$ via (3.7) and $C_1^{(k)}, C_2^{(k-\tau_1^{(k)-1})}$ as well, we obtain by (3.3)

$$\begin{aligned} & \mathbf{F}(C_1^{(k)}, \tau_1^{(k)}, C_2^{(k-\tau_1^{(k)-1})}, \tau_2^{(k-\tau_1^{(k)-1})}, i_{k-\tau_1^{(k)-1}}) \\ &= F(\hat{i}_{k+\tau_2}^{(k-\tau_1^{(k)-1})-1}, \tau_1^{(k)}, \tau_2^{(k-\tau_1^{(k)-1})}) \cdot D_{k+\tau_2}^{(k-\tau_1^{(k)-1})} \cdots D_{k-\tau_1^{(k)}+1}. \end{aligned} \quad (3.12)$$

In the next section we shall investigate the stability of this control algorithm.

4. Stability analysis

Putting (3.3) in (3.2) yields

$$\mathbf{z}_{k+1} = \mathbf{A}(C_2^{(k)}, \tau_2^{(k)}, i_k) \mathbf{z}_k - \mathbf{B}(C_2^{(k)}, \tau_2^{(k)}, i_k) \cdot \mathbf{F}(C_1^{(k)}, \tau_1^{(k)}, C_2^{(k-\tau_1^{(k)-1})}, \tau_2^{(k-\tau_1^{(k)-1})}, i_{k-\tau_1^{(k)-1}}) \mathbf{z}_k. \quad (4.1)$$

Denote

$$\begin{aligned} \mathbf{H} &= \mathbf{H}(C_2^{(k)}, \tau_2^{(k)}, i_k, C_1^{(k)}, \tau_1^{(k)}, C_2^{(k-\tau_1^{(k)-1})}, \tau_2^{(k-\tau_1^{(k)-1})}, i_{k-\tau_1^{(k)-1}}) \\ &= \mathbf{A}(C_2^{(k)}, \tau_2^{(k)}, i_k) - \mathbf{B}(C_2^{(k)}, \tau_2^{(k)}, i_k) \cdot \mathbf{F}(C_1^{(k)}, \tau_1^{(k)}, C_2^{(k-\tau_1^{(k)-1})}, \tau_2^{(k-\tau_1^{(k)-1})}, i_{k-\tau_1^{(k)-1}}), \end{aligned}$$

then we obtain the following dynamics

$$\mathbf{z}_{k+1} = \mathbf{H} \cdot \mathbf{z}_k, \quad (4.2)$$

where \mathbf{H} depends on the representation $C_2^{(k)}, \tau_2^{(k)}, i_k, C_1^{(k)}, \tau_1^{(k)}, C_2^{(k-\tau_1^{(k)}-1)}, \tau_2^{(k-\tau_1^{(k)}-1)}, i_{k-\tau_1^{(k)}-1}$. This result also explains the scale estimate stated in (2.1). Now it can be seen that the system becomes a pure Markov Jump System where the total number of regimes/scenarios is bounded by (2.1). Let \mathbf{P} be the transition probability matrix for the regimes in the complete representation (R). The sufficient and necessary stability tests provided in [21] can be directly applied. Before we introduce those tests, we shall introduce the stability criteria for MJS.

The usual stability criteria on MJSs are mean convergence (MC) and mean square convergence (MSC). We say that a system is uniformly MC if $E(\mathbf{z}_k|\mathbf{z}_0) \rightarrow \mathbf{0}$ as $k \rightarrow \infty$ for any initial condition \mathbf{z}_0 and any initial regime, where E denotes expectation. The system is said to be uniformly MSC if $E(\|\mathbf{z}_k\|^2|\mathbf{z}_0) \rightarrow 0$ as $k \rightarrow \infty$ for any initial condition \mathbf{z}_0 and any initial regime. Let I denote the identity matrix, and define

$$\begin{aligned}\mathcal{F} &= \text{diag}(\mathbf{H}) \cdot (\mathbf{P}' \otimes I_d), \\ \mathcal{A} &= \text{diag}(\mathbf{H} \otimes \mathbf{H}) \cdot (\mathbf{P}' \otimes I_{d^2}),\end{aligned}\quad (4.3)$$

where diag denotes the diagonal block matrix, i.e., the \mathbf{H} 's corresponding to different regimes are put in the diagonal entries. The symbol \otimes denotes the Kronecker product of two matrices, and d is the dimension of the dynamical system (the size of \mathbf{z}_k) in each regime. Then \mathcal{F} has size $Md \times Md$ and \mathcal{A} has size $Md^2 \times Md^2$, where M is the total number of regimes in the complete representation (bounded by (2.1)). Let $\rho(\cdot)$ denote the spectral radius of a matrix, then we have the following result.

Theorem 4.1. *The MJS (4.1) is uniformly MC if and only if $\rho(\mathcal{F}) < 1$ and uniformly MSC if and only if $\rho(\mathcal{A}) < 1$*

Proof. The proof can be found in [22]. Here we just point out that the matrices \mathcal{F} and \mathcal{A} are deterministic, but not stochastic. Also, MSC implies MC but not the other way. \square

In the numerical example we shall test both stabilities.

Remark 4.1. *We are able to provide this sufficient and necessary condition on stability, and this is largely due to the fact that the structure of this NCS has been clearly revealed.*

5. Transition probabilities

For the complete representation (R), the transition to the next regime depends on the most recent regime thanks to the Markovian property of this system, i.e.,

$$(C_2^{(k)}, \tau_2^{(k)}, i_k, C_1^{(k)}, \tau_1^{(k)}) \rightarrow (C_2^{(k+1)}, \tau_2^{(k+1)}, i_{k+1}, C_1^{(k+1)}, \tau_1^{(k+1)}). \quad (5.1)$$

As we mentioned earlier, the regimes do not have full reachability, but the transition (5.1) can be decomposed into three parts that are treated independently. The first part $i_k \rightarrow i_{k+1}$ is determined by P_m . The other two parts, $(C_2^{(k)}, \tau_2^{(k)}) \rightarrow (C_2^{(k+1)}, \tau_2^{(k+1)})$, and $(C_1^{(k)}, \tau_1^{(k)}) \rightarrow (C_1^{(k+1)}, \tau_1^{(k+1)})$ are determined by P_2, P_1 , respectively. In this way the overall transition matrix \mathbf{P} can be constructed.

6. Numerical example

We consider an example where the objective system, which is a MJS, has two modes ($m = 2$),

$$\begin{bmatrix} x_{k+1}^{(1)} \\ x_{k+1}^{(2)} \end{bmatrix} = \begin{bmatrix} 1.2 & 0.2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \end{bmatrix} + \begin{bmatrix} 0.05 \\ 0.1 \end{bmatrix} u_{k-\tau_2},$$

and

$$\begin{bmatrix} x_{k+1}^{(1)} \\ x_{k+1}^{(2)} \end{bmatrix} = \begin{bmatrix} 0.7 & 0.2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.3 \end{bmatrix} u_{k-\tau_2},$$

respectively. This system is controlled by a controller through a network with random time delays. It is assumed that $\tau_1 \in \{0, 1\}$ and $\tau_2 \in \{0, 1, 2\}$. We shall see how big the actual size of this system will be.

The matrix P_m for the two modes is given as follows:

$$P_m = \begin{pmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{pmatrix}.$$

To construct P_2 , we need to keep in mind that not all the transitions $(C_2^{(k)}, \tau_2^{(k)}) \rightarrow (C_2^{(k+1)}, \tau_2^{(k+1)})$ are possible. For instance, the transition $(0, 0) \rightarrow (0, 0)$ means that at time k , \mathbf{u}_k is received, and at time $k + 1$, \mathbf{u}_{k+1} is received with no delay. $(0, 0) \rightarrow (0, 1)$ is impossible because this is a contradiction on \mathbf{u}_k . $(0, 0) \rightarrow (0, 2)$ has zero probability because at time k , \mathbf{u}_k is received, and at time $k + 1$, \mathbf{u}_{k-1} is received (delay of 2) but is discarded since its index is older than k . $(0, 0) \rightarrow (1, 0)$ means at time k , \mathbf{u}_k is received, and at time $k + 1$ no control signal is received and the last applied control (which is \mathbf{u}_k) has 0 delay. The matrix P_2 is given in the following.

$$\begin{array}{cccccc} & (0, 0) & (0, 1) & (0, 2) & (1, 0) & (1, 1) & (2, 0) \\ (0, 0) & 0.7 & 0 & 0 & 0.3 & 0 & 0 \\ (0, 1) & 0.4 & 0.4 & 0 & 0 & 0.2 & 0 \\ (0, 2) & 0.5 & 0.2 & 0.3 & 0 & 0 & 0 \\ (1, 0) & 0.3 & 0.4 & 0 & 0 & 0 & 0.3 \\ (1, 1) & 0.5 & 0.2 & 0.3 & 0 & 0 & 0 \\ (2, 0) & 0.4 & 0.2 & 0.4 & 0 & 0 & 0 \end{array} \quad (6.1)$$

Now that $T_1 = 1$, (C_1, τ_1) has three possibilities, namely, $(0, 0)$, $(0, 1)$, $(1, 0)$, and P_1 is given below.

$$P_1 = \begin{pmatrix} 0.6 & 0 & 0.4 \\ 0.5 & 0.5 & 0 \\ 0.7 & 0.3 & 0 \end{pmatrix}.$$

By the derivation in this paper, we need to keep $2 + T_1 = 3$ consecutive triplets (C_2, τ_2, i) in the complete representation. Because the mode information i is independent, we consider 3 consecutive pairs (C_2, τ_2) that are derived from (6.1). Some sample paths are illustrated below:

$$\begin{aligned}
(0, 0) &\rightarrow (0, 0) \rightarrow (0, 0) \\
(0, 0) &\rightarrow (0, 0) \rightarrow (1, 0) \\
(0, 0) &\rightarrow (1, 0) \rightarrow (0, 0) \\
(0, 0) &\rightarrow (1, 0) \rightarrow (0, 1) \\
(0, 0) &\rightarrow (1, 0) \rightarrow (2, 0) \\
&\dots\dots\dots,
\end{aligned}$$

and the total number of all possible 3-pair paths is 45. The estimate from (2.1) on this part is given by $(1 + T_2)^{2+T_1}(2 + T_2)/2 = 54$. With the 3 possibilities on (C_1, τ_1) , there are $45 \times 3 = 135$ cases. Now that the system itself has $m = 2$ modes, then the total number of regimes in the complete representation is $2^3 \times 135 = 1080$. That is, in the regime switching representation (4.2) there are $M = 1080$ regimes for this dynamical system. Formula (2.1) yields an upper bound of $2^3 \times 54 \times 3 = 1296$ on M .

The vector $\mathbf{z}_k = [x_k^{(1)} \ x_k^{(2)} \ x_{k-1}^{(1)} \ x_{k-1}^{(2)} \ u_{k-1} \ u_{k-2} \ u_{k-3}]^T$, so $d = 7$ (see (4.3)), and $Md = 7560$ $Md^2 = 52920$. That is to say, the size of the matrix \mathcal{F} in (4.3) is 7560×7560 and that of \mathcal{A} is 52920×52920 .

Without any control we have $\rho(\mathcal{F}) = 1.1861$ and $\rho(\mathcal{A}) = 1.5237$. So the system is not stable in either sense. Figure 1 shows 10 sample paths of system state without control, and the initial states are $x_1 = x_2 = 2$.

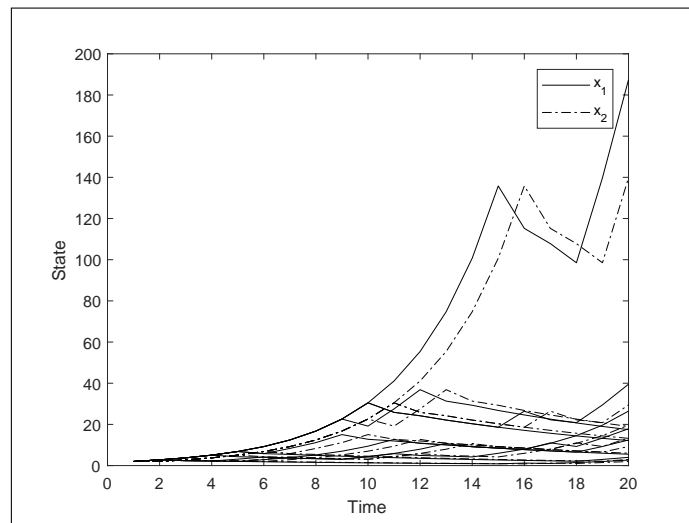


Figure 1. System state without control, 10 sample paths.

We choose the prediction horizon $L = 20$ in RHC, and calculate the feedback law F as in (3.10), then we obtain \mathbf{F} as in (3.12) and further obtain \mathbf{H} as in (4.2). Then we calculate the spectral radii $\rho(\mathcal{F})$ and $\rho(\mathcal{A})$ as in (4.3). In our example, $\rho(\mathcal{F}) = 0.9195$, $\rho(\mathcal{A}) = 1.0358$. Thus the controlled system is MC, but not MSC. Figure 2 shows the system state and control of one sample path while Figure 3 shows the results of 10 sample paths.

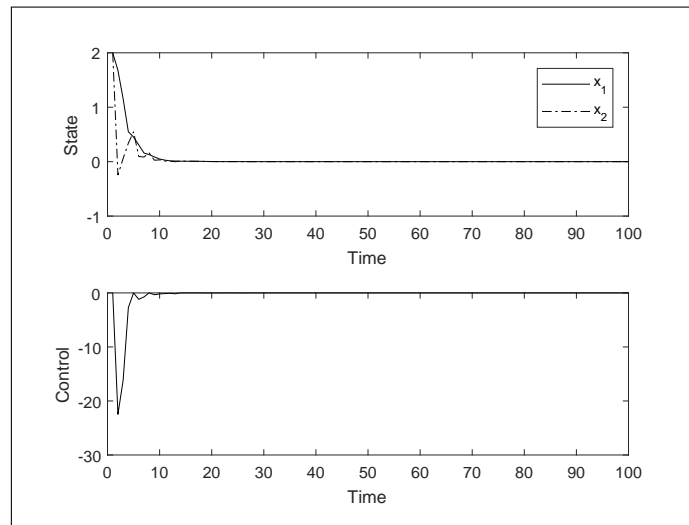


Figure 2. State and control of one sample path.

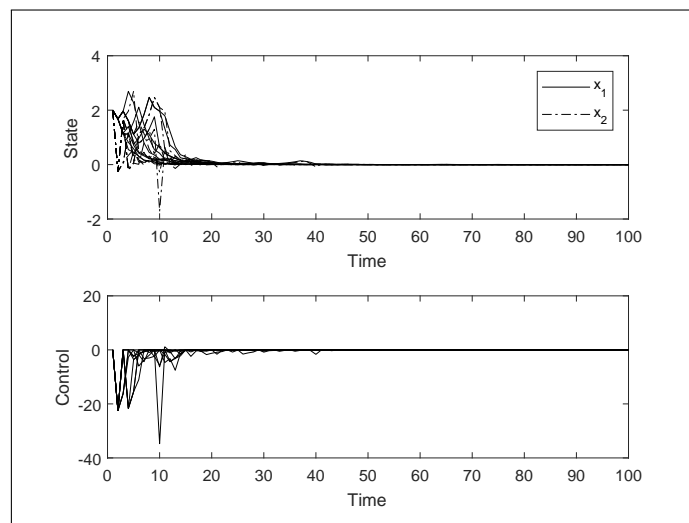


Figure 3. State and control of 10 sample paths.

The MJS stability certainly depends on the transition probabilities among the regimes. For instance, if we use

$$P_m = \begin{pmatrix} 0.85 & 0.15 \\ 0.25 & 0.75 \end{pmatrix}, \quad P_1 = \begin{pmatrix} 0.7 & 0 & 0.3 \\ 0.6 & 0.4 & 0 \\ 0.8 & 0.2 & 0 \end{pmatrix},$$

and

$$P_2 = \begin{pmatrix} 0.8 & 0 & 0 & 0.2 & 0 & 0 \\ 0.5 & 0.3 & 0 & 0 & 0.2 & 0 \\ 0.6 & 0.3 & 0.1 & 0 & 0 & 0 \\ 0.4 & 0.3 & 0 & 0 & 0 & 0.3 \\ 0.5 & 0.2 & 0.3 & 0 & 0 & 0 \\ 0.7 & 0.2 & 0.1 & 0 & 0 & 0 \end{pmatrix},$$

then again, without control we have $\rho(\mathcal{F}) = 1.2301$ and $\rho(\mathcal{A}) = 1.6022$. Figure 4 shows the system output with no control.

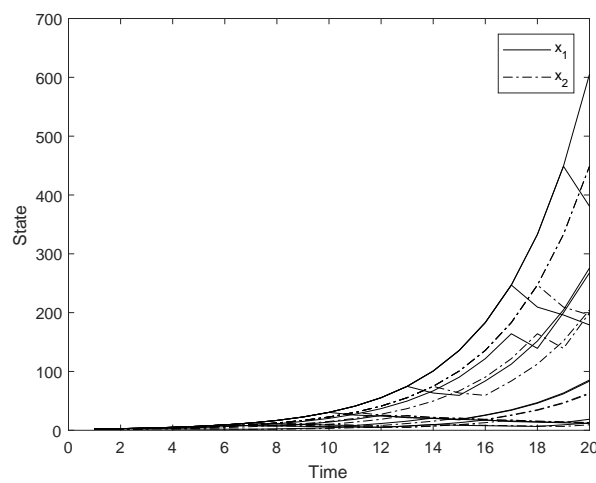


Figure 4. System state of 10 sample paths without control.

Then we apply the proposed predictive control method and have $\rho(\mathcal{F}) = 0.8769, \rho(\mathcal{A}) = 0.9466$. Thus the controlled system is MC as well as MSC. Figure 5 shows the system state and control of one sample path while Figure 6 shows the results of 10 sample paths.

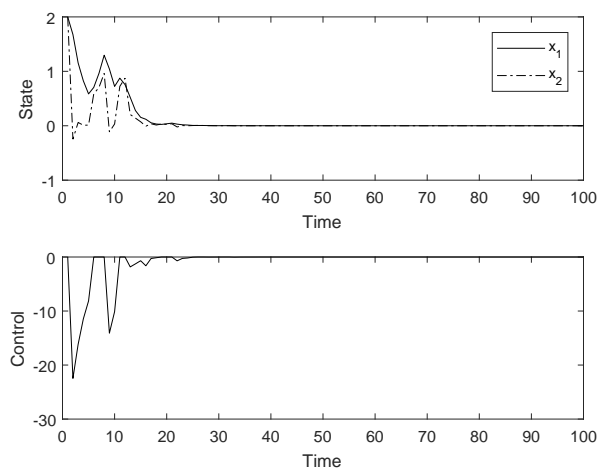


Figure 5. State and control of one sample path.

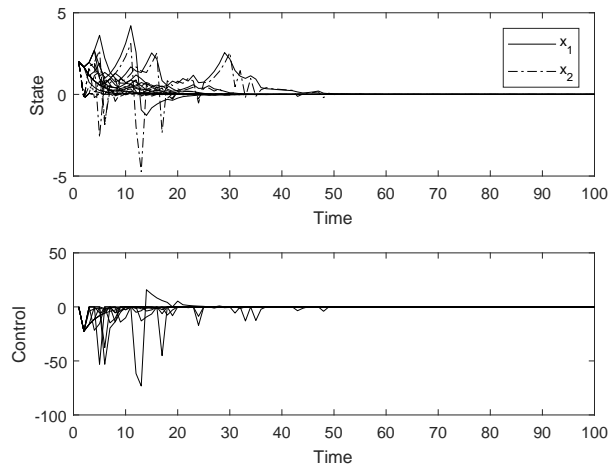


Figure 6. State and control of 10 sample paths.

7. Conclusions

To overcome the constraints of NCS such as random packet delays/disorders, it is a natural idea to estimate the system states and send out a control signal that compensates the time delay, hence the expectation of a better control performance. In this paper we modeled the NCS as a regime switching system and proposed a simplified predictive control method. The regime estimate (2.1), which is one of the main contributions of this paper, illustrates that this seemingly small system actually has a very large scale. The structure of this system has been clearly revealed, and a concise sufficient and necessary condition on stability is obtained. Numerical examples clearly show the features of this dynamical system.

Acknowledgments

This work is supported by Barrios Technology Faculty Fellowship from University of Houston - Clear Lake, 2017-2018.

Conflict of interest

The author declares no conflict of interest in this paper.

References

1. Zhang L, Gao H, Kaynak O (2013) Network-Induced Constraints in Networked Control Systems—A Survey. *IEEE T Ind Inform* 9: 403–416.
2. Zhang L, Shi Y, Chen T, et al. (2005) A New Method for Stabilization of Networked Control Systems with Random Delays. *IEEE T Automat Contr* 50: 1177–1181.

3. Xiao L, Hassibi A, How JP (2000) Control with Random Communication Delays via a Discrete-Time Jump System Approach. *Proceedings of American Control Conference*, Chicago, Illinois, USA.
4. Donkers MCF, Heemels WPMH, Wouw N, et al. (2011) Stability Analysis of Networked Control Systems Using a Switched Linear Systems Approach. *IEEE T Automat Contr* 56: 2101–2115.
5. Heijmans SHJ, Postoyan R, Noroozi N, et al (2016) Stability Analysis of Networked Control Systems with Direct-Feedthrough Terms: Part II - the Linear Case. *Proceedings of IEEE Conference on Decision and Control*, Las Vegas, Nevada, USA.
6. Dai SL, Lin H, Ge SS (2010) Scheduling-and-Control Codesign for a Collection of Networked Control Systems With Uncertain Delays. *IEEE Trans on Contr Sys Tech*, 18: 66–78.
7. Tan C, Zhang H (2017) Necessary and Sufficient Stabilizing Conditions for Networked Control Systems With Simultaneous Transmission Delay and Packet Dropout. *IEEE Trans on Automat Contr* 62: 4011–4016.
8. Branicky MS, Phillips SM, Zhang W (2000) Stability of Networked Control Systems: Explicit Analysis of Delay. *Proceedings of American Control Conference*, Chicago, Illinois, USA.
9. Schenato L (2008) Optimal Estimation in Networked Control Systems Subject to Random Delay and Packet Drop. *IEEE T Automat Contr* 53: 1311–1317.
10. Liu GP, Xia Y, Chen J, et al (2007) Networked Predictive Control of Systems With Random Network Delays in Both Forward and Feedback Channels. *IEEE T Ind Electron* 54: 1282–1297.
11. Nagahara M, Quevedo DE, Østergaard J (2014) Sparse Packetized Predictive Control for Networked Control Over Erasure Channels. *IEEE T Automat Contr* 59: 1899–1905.
12. Quevedo DE, Nešić D (2011) Input-to-State Stability of Packetized Predictive Control Over Unreliable Networks Affected by Packet-Dropouts. *IEEE T Automat Contr* 56: 370–375.
13. Yang R, Liu GP, Shi P, et al. (2014) Predictive Output Feedback Control for Networked Control Systems. *IEEE T Ind Electron* 61: 512–520.
14. Henriksson E, Sandberg H, Johansson KH (2008) Predictive Compensation for Communication Outages in Networked Control Systems. *Proceedings of IEEE Conference on Decision and Control*, Cancun, Mexico.
15. Zhao YB, Liu GP, Rees D (2010) Actively Compensating for Data Packet Disorder in Networked Control Systems. *IEEE T Circuits-II* 57: 913–917.
16. Ulusoy A, Gurbuz O, Onat A (2011) Wireless Model-Based Predictive Networked Control System Over Cooperative Wireless Network. *IEEE T Ind Inform* 7: 41–51.
17. Mhaskar P, El-Farra N, Christofides P (2005) Predictive control of switched nonlinear systems with scheduled mode transitions. *IEEE T Automat Contr* 50: 1670–1680.
18. Müller M, Martius P, Allgöwer F (2012) Model predictive control of switched nonlinear systems under average dwell-time. *J Process Contr* 22: 1702–1710.
19. Essick R, Lee J, Dullerud GE (2014) Control of linear switched systems with receding horizon modal information. *IEEE T Automat Contr* 59: 2340–2352.

-
20. Yang Y, Nesbitt ND (2017) Concise Iterative Algorithms On the State Feedback Form for Model Predictive Control and Stability Analysis of Regime Switching Systems. *Proceedings of IEEE Symposium Series on Computational Intelligence*, Honolulu, Hawaii, USA.
 21. Costa OLV, Fragoso MD, Marques RP (2005) Discrete-Time Markov Jump Linear Systems. *Springer-Verlag*, New York, USA.
 22. Costa OLV, Fragoso MD (1993) Stability Results for Discrete-Time Linear Systems with Markovian Jumping Parameters. *J Math Anal Appl* 179: 154–178.



AIMS Press

©2019 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)