Chapter 3

Sequential output information based predictive control for networked control systems

3.1 Introduction

NCSs find diverse applications, including industrial control, surveillance systems, and process control engineering [7]. Utilizing communication networks, an NCS establishes closed-loop control, enabling a cost-effective, remotely controllable, and adaptable system [79, 80]. However, the presence of random time delays, packet losses, and limited bandwidth in shared communication networks poses challenges to achieving optimal control performance [9, 85, 86].

Various control methods have been devised to mitigate the adverse effects of uncertain network delays and packet losses in Networked Control Systems (NCS) [110, 111]. The analysis framework can adopt either a deterministic or stochastic approach. The deterministic approach integrates uncertain variations, such as delays or dropouts, into the system model, leading to resilient designs [1,101]. In contrast, the stochastic approach utilizes statistical data to formulate control designs [102, 103]. This chapter focuses on employing the deterministic approach.

Given the presence of delays in Networked Control Systems (NCS), it is anticipated that predictive controllers would outperform non-predictive ones in terms of control performance [96]. In this direction, classical control design methods based on the Smith

predictor have been explored in the literature [99, 100]. However, their effectiveness is constrained by the variable nature of the delay. Consequently, various predictive control methods have been developed, utilizing either deterministic [11] or stochastic analysis frameworks [104].

The limitation imposed by frequent transmission through a network with limited bandwidth is often alleviated through the utilization of the event-triggering technique (ETT) for data transmission [47–50, 129]. ETT allows data transmission to be initiated only when a specific event or change in the data occurs, thereby reducing the frequency of data transmission and conserving network bandwidth usage [34, 51, 52]. In this study, ETT is employed in both the feedback and forward channels to enhance data transmission efficacy, resulting in a lighter network load.

In the context of transmitting state or output information in an NCS, it is often more advantageous to share output information due to its smaller data size, especially in systems with a large number of states. The size of a data packet is known to impact the network load, causing transmission delay and reduced reliability [18, 19]. A smaller data packet reduces transmission delay, making it advantageous for NCS. Despite the frequent use of the ETT in NCSs, the literature does not adequately address the reduction of data packet size. As packet size has not been a significant focus in existing literature, most works have considered the transmission of observed state information over the network [13–16], even for applications with a large number of states [17].

On the contrary, if output information is transmitted via the feedback channel, it necessitates the implementation of an observer on the controller side. This observer is responsible for deriving state estimates based on intermittently received output information, posing challenges to state estimation. Additionally, the data packet size may not be optimally utilized when transmitting such output information, especially for systems with few outputs. To tackle this concern, this study proposes a sequential observer for NCS, which harnesses successive output information in a single packet to enhance state prediction from delayed data.

This study focuses on an NCS incorporating event-triggered network communication in both the feedback and forward channels. Event-triggered communication is implemented to alleviate frequent network usage. Moreover, the approach involves sharing only output information over the network, in contrast to conventional schemes that trans-

mit state information. The conventional transmission of state information necessitates an observer on the plant side for state observation. In this work, as only the output is transmitted, a state observer is employed on the remote side, resulting in a reduced packet size for transmission. However, the system's performance experiences degradation due to states being observed from delayed output information.

In an effort to enhance the performance of these schemes, a sequential output transmission strategy is explored, where sequential output information is consolidated into a single packet and transmitted collectively. A sequential observer is established on the remote side to refine state estimation by leveraging the sequential information received in a single transmission. Subsequently, state predictions are generated from the observed state information to address delays stemming from network-induced random delays, dropouts, and the ETT in both the feedback and forward channels. Utilizing the predictive control scheme, the comprehensive system dynamics are derived, necessitating distinct designs for the controller and observer gains. The state feedback controller employs the renowned Linear Quadratic Regulator (LQR) gain, while the observer gain is determined through the solution of Linear Matrix Inequalities (LMIs). The efficacy of this approach is illustrated by applying it to the networked control of both an inverted pendulum and a DC motor. The results demonstrate that the performance of the closed-loop system is enhanced when multiple sequences of output information are transmitted in a single packet.

The subsequent sections of this chapter are structured as follows. Section 2 introduces the proposed networked control scheme and the corresponding system model. Within this section, the components of the overall system, including the sequential observer and prediction mechanism, are delineated. Section 3 focuses on obtaining the overall system dynamics. The design of the controller and observer gain matrices is expounded in Section 4. Subsequently, Section 5 demonstrates the efficacy of the proposed method through design cases and simulation results. The chapter concludes with Section 6, summarizing the findings and drawing conclusions.

3.2 System description

3.2.1 NCS overview

The NCS considered in this work is shown in Figure 3.1, which comprises a plant and a controller with two-way Event-Triggered (ET) communication. The output information from the plant to the controller and the designed control information from the controller to the plant are communicated through the network. The data transmitted in the form of packets will be affected simultaneously by network-induced delays and packet dropout. ET communication is considered in feedback and forward channels to conserve the limited communication resources.

An output buffer employed at the plant side holds a sequence of output data. The packets containing this sequence are then delivered across the feedback channel through an ET mechanism. Due to the network and ETT, these transmitted packets experience random delays and dropouts. The building blocks are assumed to be time-synchronized, and time-stamped data is sent over the network.

On the controller side, an observer receives delayed packets containing the delayed output sequence and utilizes this sequence to observe the states. To compensate for network delays and dropouts in the feedback and forward channels, a state predictor is used to construct predicted states, which are then utilized in creating predictive state feedback control.

A buffer is used to gather the consecutive predictive control inputs transmitted intermittently to the plant side through the forward path ET mechanism. The control input sequence required by the observer and predictor are retrieved from this buffer.

Due to the forward path network, packets containing the sequence of control inputs may experience delays and drops. On the plant side, a selector is employed to receive these delayed packets containing the array of inputs and choose an appropriate control input from the array based on the forward channel delay experienced by that packet.

To this end, note that the observer can be employed either on the controller (remote) side [20, 22] or on the plant side of the NCS [12]. Having an observer on the controller side of an NCS is usually desirable since output data requires smaller packet sizes than state data, especially for systems with many states. However, the state observation using a remote observer encounters variation in the observed states because the output values

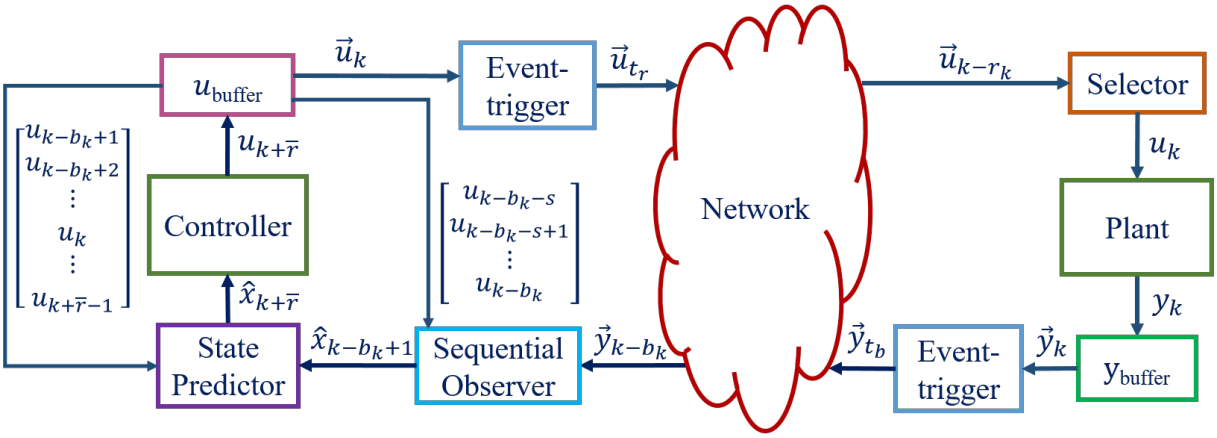

Figure 3.1: The sequential output information-based event-triggered predictive control scheme for NCS

are intermittently obtained. A sequence of output information embedded in a single transmitted packet has the potential to improve the observer's performance, and the same is addressed in this work.

The plant to be controlled has a discrete-time model:

$$x_{k+1} = Ax_k + Bu_k, \qquad y_k = Cx_k \tag{3.1}$$

Here the system state, input, and output are denoted by $x_k \in \mathcal{R}^n$, $u_k \in \mathcal{R}^m$, and $y_k \in \mathcal{R}^p$, respectively, and time instants are denoted by k. , B, and C are the system matrices fulfilling the controllability and observability of the plant.

Transmitting a set of output values in a single packet requires a buffer, called y_{buffer}, to hold the sequential output values as follows:

$$\vec{y}_k = \begin{bmatrix} y_k & y_{k-1} & \cdots & y_{k-s} \end{bmatrix}, s \in \mathcal{N}$$

In a simpler situation, only y_k is delivered through the network, i.e., $s = 0$ represents the case that only the instantaneous output is transmitted. In this scenario, the state information is estimated by a typical Luenberger observer, which is denoted here as a level one sequential observer. When $s > 0$, in addition to y_k, the subsequent s numbers of delayed information $(y_{k-i}, i = 1, \cdots, s)$ are also communicated in a single packet. Such sequential information is then received and utilized by the sequential observer, a succession of Luenberger observers with $s + 1$ level, to observe the states. An increase in the order of the sequential observer results in improved quality of the observed state information, as demonstrated later in Section 5. A schematic of the sequential observer with $(s + 1)$

level of observation and use of different buffers is shown in Figure 3.2. The dynamics of the sequential observer are discussed later in Section 2.2.

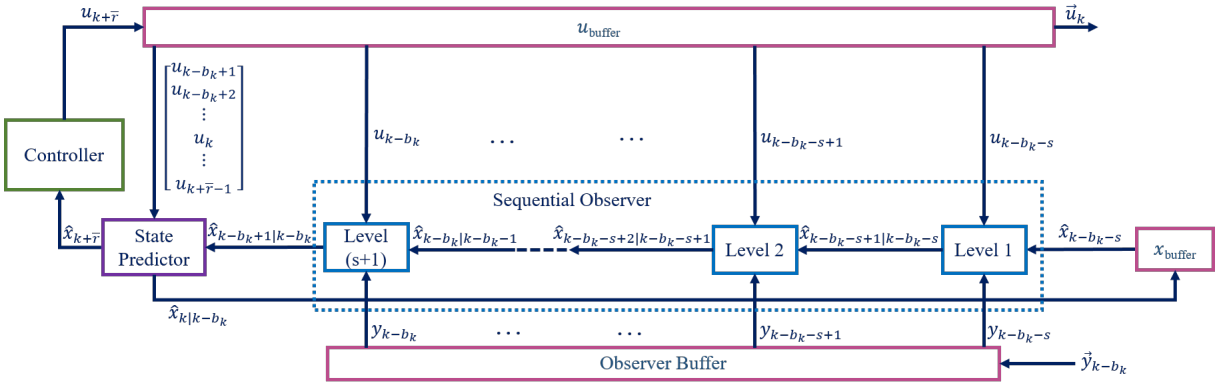

Figure 3.2: The internal structure of sequential observer and the whole prediction method on the remote side of the NCS

Remark 4. *The level of sequential observer corresponds to the number of output sequences transmitted in a single packet. The larger the sequence size, the better the observer's performance. Conversely, a larger sequence increases the packet size, the network load, and the corresponding complexity in the observer. This trade-off can be exploited to define the sequence size for a particular application.*

Remark 5. *Since the observer is located on the controller side of the network, the observer's performance is influenced by the network-induced delays and drops. In the case of delays and drops, the latest among the received packets is utilized, which may aggravate the observer error. However, due to the sequential output information available within the received packet, the state observation is improved based on the level of the sequential information or that of the sequential observer. From Figure 2, it can be understood that the packet $\vec{y}_{t_1}$ is received at time instant k and the packet $\vec{y}_{t_3}$ is received at $(k+6)$. In between, no packets are received due to delays and drops. As the $\vec{y}_{t_3}$ packet is received at $(k+6)$, the packet contains a sequence of output information, some of which are not transmitted in the conventional scheme ($s=0$). Such a sequence embedded in a single packet is used to improve the state estimation using a sequential observer.*

It is considered that the control loop experiences delays and packet dropouts (both active and passive. Passive dropouts are the dropouts caused by the network, which are unavoidable. Active dropouts are those brought on by ETT, which is enforced to

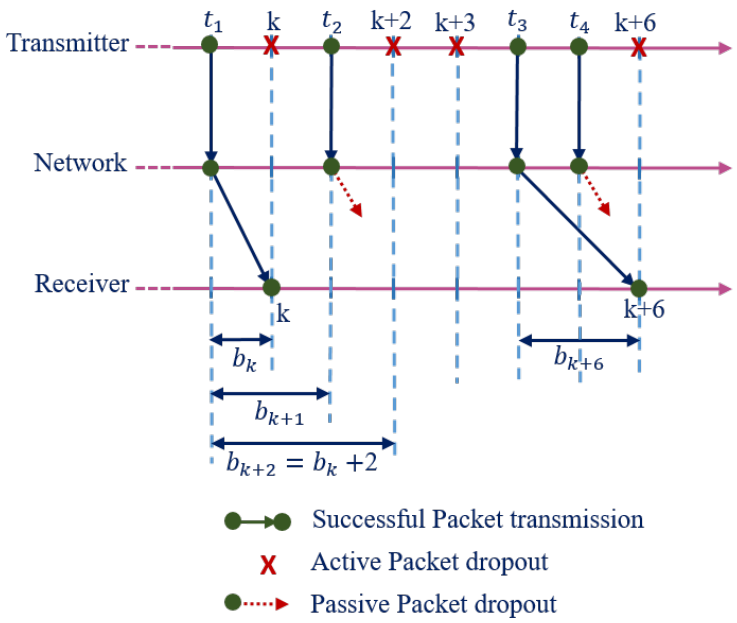

Figure 3.3: A representation of event-triggered packet transmission in the network with delays and dropouts

conserve communication bandwidth). Let us consider that the feedback ET triggers at a specific instant denoted as $t_b, b \in \mathcal{N}$, and uses this triggering event to transmit the output-sequence packet y_{t_b} through the network as depicted in Figure 3.3.

When a packet dropout (active or passive) occurs, the latest packet at the receiver (sequential observer) is used again, increasing the delay value by one. Let b_k be the random delay in the feedback path; then it obeys that $b_{k+1} \leq b_k + 1$. Thus, the active or passive dropout count is added to the delay value. As each received packet has a distinct time stamp t_b, and the system is synchronized, the delay value b_k can be obtained at the receiver side by computing the difference between the instantaneous value of k and the time stamp of the received packet, i.e., $b_k = k - t_b$. If D_b represents the upper bound of the feedback network's delay and P_{b_a} and P_{b_p} represent the upper bounds of the feedback network's consecutive active and passive dropouts, respectively, then $\bar{b} = D_b + P_{b_a} + P_{b_p}$ represents the largest possible delay in the feedback path. Similarly, the largest possible delay in the forward path is defined as $\bar{r} = D_r + P_{r_a} + P_{r_p}$. Table 3.1 lists the notations and NCS parameters used in this work.

Table 3.1: Notations used

D_b	Maximum network delay in the feedback path
P_{b_a}	Maximum consecutive active dropouts in the feedback path
P_{b_p}	Maximum consecutive passive dropouts in the feedback path
b_k	Instantaneous feedback path delay
$t_b, b \in \mathcal{N}$	Event-triggered time-instants in feedback path
$\vec{y}_k$	Sequential output information
$\vec{y}_{t_b}$	Event-triggered output sequence
$\bar{b}$	$D_b + P_{b_a} + P_{b_p}$
b_k	$k - t_b$
D_r	Maximum network delay in the forward path
P_{r_a}	Maximum consecutive active dropouts in the forward path
P_{r_p}	Maximum consecutive passive dropouts in the forward path
r_k	Instantaneous forward path delay
$t_r, r \in \mathcal{N}$	Event-triggered time-instants in forward path
$\vec{u}_{t_r}$	Event-triggered control packets
$\bar{r}$	$D_r + P_{r_a} + P_{r_p}$
r_k	$k - t_r$
$\mathcal{N}$	The set of all natural numbers
$\mathcal{R}^*$	The set of all *-tuples of real numbers

3.2.2 Sequential observer

The sequential observer block (see Fig. 3.2) comprises an observer buffer and a sequential observer. When a new packet arrives, the observer buffer compares its timestamp with the last one stored in memory and keeps only the most recent ones. In case of a packet drop, the observer buffer retrieves the previously stored data as shown below:

$$\vec{y}_{k-b_k} = \begin{cases} \vec{y}(max\{t_b\}), & (k-b_k) \geq (k-1-b_{k-1}) \\ \vec{y}_{k-1-b_{k-1}}, & (k-b_k) < (k-1-b_{k-1}) \end{cases}$$

Here, the expressions $k-b_k$ and $k-1-b_{k-1}$ correspond to the time stamps of the currently received packet and the previously stored packet, respectively. The term $max\{t_b\}$ denotes the most recent time stamp. The observer utilizes the latest sequence of output

information and observes the corresponding one-step-ahead state value by implementing a series of operations:

$$\begin{aligned}\hat{x}_{k-b_k-s+1|k-b_k-s} =& A\hat{x}_{k-b_k-s} + Bu_{k-b_k-s} + L(y_{k-b_k-s} - C\hat{x}_{k-b_k-s}) \\ &\vdots \\ \hat{x}_{k-b_k+1|k-b_k} =& A\hat{x}_{k-b_k|k-b_k-1} + Bu_{k-b_k} + L(y_{k-b_k} - C\hat{x}_{k-b_k|k-b_k-1}),\end{aligned}$$

where L is the observer gain matrix to be designed. The above can be generalized as:

$$\hat{x}_{k-b_k+i+1|k-b_k+i} = A\hat{x}_{k-b_k+i|k-b_k+i-1} + Bu_{k-b_k+i} + L(y_{k-b_k+i} - C\hat{x}_{k-b_k+i|k-b_k+i-1}) \quad \text{for } i = -s, -(s-1), \cdots, -2, -1, 0 \tag{3.2}$$

Depending on the instantaneous delay value b_k, the values of $\hat{x}_{k-b_k+i}$ and u_{k-b_k+i}, $i = -s, -(s-1), -(s-2), \cdots, -2, -1, 0$, and $s \in \mathcal{N}$ are obtained from x_{buffer} and u_{buffer}, respectively. The x_{buffer} and u_{buffer} are the two buffers for storing the consecutive predicted state $\hat{x}_{k|k-b_k}$ and control $u_{k+\bar{r}}$ values. The sizes of these buffers are derived from the maximum delay values $\bar{b}$ and $\bar{r}$ and the level of the sequential observer.

After obtaining the final estimate at the sequential observer and defining the observer error as $e_k = x_k - \hat{x}_{k|k-1}$, from (3.1) and (3.2), one can write

$$\begin{aligned}e_{k-b_k+1} &= x_{k-b_k+1} - \hat{x}_{k-b_k+1|k-b_k} \\ &= Ax_{k-b_k} + Bu_{k-b_k} - A\hat{x}_{k-b_k} - Bu_{k-b_k} - L(Cx_{k-b_k} - C\hat{x}_{k-b_k}) \\ &= (A - LC)e_{k-b_k}\end{aligned} \tag{3.3}$$

3.2.3 Predictor and controller

It is necessary to account for the delay once the new output data has been integrated into the observed state through the observer. This is accomplished through model-based predictions. The sum of the instantaneous feedback delay b_k and the maximum delay $\bar{r}$ in the forward path determines the number of prediction steps. The predictor is constructed as follows:

$$\hat{x}_{k-b_k+n|k-b_k} = A\hat{x}_{k-b_k+n-1} + Bu_{k-b_k+n-1}, \quad \text{for } n = 2, 3 \cdots, b_k + \bar{r} \tag{3.4}$$

To make these state predictions, the control input values $u_{k-b_k+1}, \cdots, u_k, \cdots, u_{k+\bar{r}-1}$, as shown in Figure 3.2, are taken from the u_{buffer}. It can be noted that both the observed

and predicted states are denoted as $\hat{x}_k$ despite different updates being employed. Now, using (3.2) and (3.4), one obtains

$$\hat{x}_{k+\bar{r}|k-b_k} = A^{b_k+\bar{r}}\hat{x}_{k-b_k} + A^{b_k+\bar{r}-1}L(y_{k-b_k} - C\hat{x}_{k-b_k}) + \sum_{m=1}^{b_k+\bar{r}} A^{m-1}Bu_{k+\bar{r}-m} \tag{3.5}$$

The u_{buffer} and selector are then used to select the control input, compensating for the varying forward path delay.

First, the predicted state information is used to generate the predictive control input $u_{k+\bar{r}|k-b_k}$ following the state feedback control law $u_k = Kx_k$, where $K \in \mathcal{R}^{m\times n}$ is the controller gain matrix to be designed. The predicted control input value is $u_{k+\bar{r}} = K\hat{x}_{k+\bar{r}|k-b_k}$ and defining $d_k = \bar{r} + b_{k-\bar{r}}$, one can write

$$u_k = K\hat{x}_{k|k-d_k} \tag{3.6}$$

The u_{buffer} is then updated with the predicted control input $u_{k+\bar{r}}$, which includes $\vec{u}_k = [u_k^T \; u_{k+1}^T \; \cdots \; u_{k+\bar{r}}^T]^T$. The oldest value u_{k-1}, from the array $\vec{u}_{k-1} = [u_{k-1}^T \; u_k^T \; \cdots \; u_{k-1+\bar{r}}^T]^T$ produced at the preceding instant, is dropped during the procedure. Therefor , the vector $\vec{u}$ always has length $\bar{r}+1$.

Then, the packets containing $\vec{u}_k$ are transmitted to the plant side through the forward path ET mechanism. Like the feedback path case, the packets undergo active dropouts due to ET transmission, delay, and drop due to the forward path network. On the plant side, a selector block receives the delayed control packets.

The selector block is made up of a selector and a selector buffer. The buffer retains the packet with the most recent time-stamp when it receives new packets, i.e.

$$\vec{u}_{k-r_k} = \begin{cases} \vec{u}(max\{t_r\}), & k - r_k \geq k - 1 - r_{k-1} \\ \vec{u}_{k-1-r_{k-1}}, & k - r_k < k - 1 - r_{k-1} \end{cases}$$

As can be seen from the packets produced by the u_{buffer} at each time-step, the controller's predicted control input, $u_{k+\bar{r}}$, is made available in $\bar{r}$ successive packets. The selection logic chooses the appropriate value u_k applicable for the present instant based on the instantaneous value of the forward path delay r_k.

3.2.4 Event-triggering

Event-triggering intermittently communicates $\vec{y}_k$ to the controller side. The following logical condition is used to implement the event-triggering mechanism at the feedback channel:

$$t_{b+1} = \begin{cases} min\{k|(\tilde{y}_k)^T H_b(\tilde{y}_k) > \epsilon_b(y_k^T H_b y_k)\}, \\ \qquad\qquad\qquad \text{when } k < t_b + E_b + 1 \\ t_b + E_b + 1, \qquad \text{when } k \geq t_b + E_b + 1 \end{cases} \tag{3.7}$$

where $\tilde{y}_k = y_k - y_{t_b}$ and

$$\vec{y}_{t_{b+1}} = \begin{cases} \vec{y}_k, & \text{when } k < t_b + E_b + 1 \\ \vec{y}_{k-1}, & \text{when } k \geq t_b + E_b + 1 \end{cases} \tag{3.8}$$

Assuming that t_b is the most recent triggering time-instant, the next triggering instant to be identified is t_{b+1}. The dimension of $H_b > 0$ is an appropriate one and $0 < \epsilon_b < 1$. Given that E_b represents the maximum number of dropouts that are currently active, the maximum amount of time that can pass between any two triggering instants is $E_b + 1$. It may be denoted that H_b and ϵ_b are chosen by trading off the system performance and communication bandwidth availability [11].

Similarly, the forward channel's event-triggering mechanism is described as

$$t_{r+1} = \begin{cases} min\{k|(\tilde{\vec{u}}_k)^T H_r(\tilde{\vec{u}}_k) > \epsilon_r(\vec{u}_k^T H_r \vec{u}_k)\}, \\ \qquad\qquad\qquad \text{when } k < t_r + E_r + 1 \\ \hline t_r + E_r + 1, \qquad \text{when } k \geq t_r + E_r + 1 \end{cases} \tag{3.9}$$

where $\tilde{\vec{u}}_k = \vec{u}_k - \vec{u}_{t_r}$ and

$$\vec{u}_{t_{r+1}} = \begin{cases} \vec{u}_k, & \text{when } k < t_r + E_r + 1 \\ \vec{u}_{k-1}, & \text{when } k \geq t_r + E_r + 1 \end{cases} \tag{3.10}$$

The notations H_r, E_r, ϵ_r and t_r have equivalent correspondence to the event-triggering condition in the feedback channel.

3.3 Overall system dynamics

This section presents the overall system dynamics subjected to the predictive control scheme employed at the remote terminal. With the control input at an instant given by

$u_k = K\hat{x}_{k|k-d_k}$, the closed-loop system dynamics is

$$x_{k+1} = (A + BK)x_k - BK(x_k - \hat{x}_{k|k-d_k}) \tag{3.11}$$

Let $\alpha_k = x_k - \hat{x}_{k|k-d_k}$ represents the error, then

$$\alpha_k = A(x_{k-1} - \hat{x}_{k-1|k-d_k})$$

By successive replacements, one can write

$$\alpha_k = A^{d_k-1}(x_{k-d_k+1} - \hat{x}_{k-d_k+1|k-d_k}) \tag{3.12}$$

Now, at the observer end, in a similar fashion, letting $\beta_k = x_{k-d_k+1} - \hat{x}_{k-d_k+1|k-d_k}$, one can write

$$\beta_k = Ax_{k-d_k} + Bu_{k-d_k} - A\hat{x}_{k-d_k} - Bu_{k-d_k} - L(Cx_{k-d_k} - C\hat{x}_{k-d_k}) \tag{3.13}$$

Defining $e_{k-d_k} = x_{k-d_k} - \hat{x}_{k-d_k}$, one obtains

$$\beta_k = (A - LC)e_{k-d_k} \tag{3.14}$$

Replacing (3.14) into (3.12) and then (3.12) into (3.11) yields

$$x_{k+1} = (A + BK)x_k - BKA^{d_k-1}(A - LC)e_{k-d_k} \tag{3.15}$$

The dynamics of e_{k-d_k} will be studied for adjudging the input due to the error dynamics. For the purpose, let us define the variable $w_k = b_{k-\bar{r}} - b_{k-\bar{r}-1}$. Then, it holds that $w_k \in \{-\bar{b}, -(\bar{b}-1), -(\bar{b}-2), \cdots, -1, 0, 1\}$ and the relationship between e_k and w_k is given as following.

Case-I: $w_k = 1$ indicates a dropout, thereby $b_{k+1-\bar{r}} = b_{k-\bar{r}} + 1$. Since $d_k = \bar{r} + b_{k-\bar{r}}$, one can write

$$\begin{aligned} e_{k+1-d_{k+1}} &= x_{k+1-d_{k+1}} - \hat{x}_{k+1-d_{k+1}|k-d_k} \\ &= x_{k+1-\bar{r}-b_{k-\bar{r}+1}} - \hat{x}_{k+1-\bar{r}-b_{k-\bar{r}+1}|k-d_k} \\ &= x_{k-\bar{r}-b_{k-\bar{r}}} - \hat{x}_{k-\bar{r}-b_{k-\bar{r}}|k-d_k} \\ &= x_{k-d_k} - \hat{x}_{k-d_k|k-d_k} \\ &= e_{k-d_k} \end{aligned} \tag{3.16}$$

Case-II: $w_k \in \{-\bar{b}, \cdots, 0\}$. In this case, it is clear that $b_{k-\bar{r}+1} = b_{k-\bar{r}} + w_k$ and we have

$$\hat{x}_{k+1-\bar{r}-b_{k-\bar{r}+1}} = \hat{x}_{k+1-\bar{r}-b_{k-\bar{r}}-w_k|k-d_k}$$

$$\begin{aligned} e_{k+1-d_{k+1}} &= x_{k+1-d_{k+1}} - \hat{x}_{k+1-d_{k+1}|k-d_k} \\ &= x_{k+1-d_k-w_k} - \hat{x}_{k+1-d_k-w_k|k-d_k} \\ &= A(x_{k-d_k-w_k} - \hat{x}_{k-d_k-w_k|k-d_k}) \\ &\vdots \\ &= A^{-w_k}(A-LC)e_{k-d_k} \end{aligned} \tag{3.17}$$

Letting $z_k = \begin{bmatrix} x_k^T & e_{k-d_k}^T \end{bmatrix}^T$, the augmented dynamics for the above two cases can be derived as:

$$z_{k+1} = \Phi z_k, \tag{3.18}$$

where

$$\Phi = \begin{cases} \begin{bmatrix} A+BK & -BKA^{d_k-1}(A-LC) \\ 0 & I \end{bmatrix}, \\ \qquad \text{when } w_k = 1 \qquad \text{(3.19a)} \\ \begin{bmatrix} A+BK & -BKA^{d_k-1}(A-LC) \\ 0 & A^{-w_k}(A-LC) \end{bmatrix}, \\ \qquad \text{when } w_k \in \{-\bar{b}, \cdots, 0\} \qquad \text{(3.19b)} \end{cases}$$

The stability of the closed-loop NCS can be analyzed through the augmented system dynamics (3.18).

3.4 Controller and observer design

This section considers the dynamics (3.18) and adopts a design of the gains L and K guaranteeing the stability of the closed-loop NCS. It may be noted that Case-I ($w_k = 1$) may occur consecutively at most $\bar{b}$ times. Again, when the maximum count of dropouts is reached, the system dynamics switches to Case-II. Therefore, Case-I cannot happen at all times. The stability condition for (3.18) is derived next, considering only Case-II.

The following result can be determined by using the separation principle.

Theorem 3. *The augmented system dynamics of the NCS* (3.18) *along with* (3.19b) *is asymptotically stable if* $(A + BK)$ *is Schur and the system with the following switched system matrix is stable.*

$$A^{-w_k}(A-LC), \quad w_k \in \{-\bar{b}, \cdots, 0\} \tag{3.20}$$

Proof. As described in Section 3, the proposed scheme yields an overall system dynamics (3.18) with stability to be ensured for Φ as defined in (3.19b). Since Φ is block-diagonal, the stability can be ensured by the separation principle [125] that the individual stability of $(A + BK)$ and $A^{-w_k}(A - LC)$ will ensure the overall stability of the system. □

It can be seen that the above theorem separates the controller and observer design problems that can now be carried out independently.

To ensure stability of $A^{-w_k}(A-LC)$ with w_k being uncertain, the common Lyapunov function [126,130] is used. Then, the following result can be constructed, which is further used later for the controller and observer design.

Theorem 4. *The NCS with the augmented system dynamics (3.18) with (3.19b) is asymptotically stable if $(A + BK)$ is Schur and, for chosen $Q > 0$, there exists a $P = P^T > 0$ satisfying the following inequality.*

$$[A^{-w_k}(A - LC)]^T P[A^{-w_k}(A - LC)] - P + Q < 0, \quad w_k \in \{-\bar{b}, \cdots, 0\} \tag{3.21}$$

Proof. The Schur-ness of $(A+BK)$ ensures its stability. On the other hand, the stability of $A^{-w_k}(A - LC)$ can be analyzed from the perspective of switched systems. Using the common Lyapunov function method [130] for linear systems, a Lyapunov function is chosen as:

$$V_k = e_{k-d_k}^T P e_{k-d_k}. \tag{3.22}$$

Then (3.21) ensures that $V_{k+1} - V_k < -e_{k-d_k}^T Q e_{k-d_k} < 0$, which ensures the stability of the error dynamics of the observer. □

Remark 6. *Note that the common Lyapunov equation is used to ensure the stability of the observer error dynamics, yielding a sufficient stability condition. Though a multiple Lyapunov function-based analysis is known to yield less conservative results [130], the analysis is comparatively complex [131]. Employing such analysis on the proposed NCS scheme may improve the observer's performance. Such e tensions, as well as using a variable observer gain based on the delay information, can be further studied in the future.*

In this work, the discrete-time LQR controller [125] is used for controller design, although other performance-based design methods can also be used to ensure some performance of $(A + BK)$. Several simulation studies are carried out to validate the efficacy

of the LQR controller for the proposed system, as illustrated in the next section through case studies.

For observer design, in (3.21), Q can be considered similar to the Q matrix of the LQR theory [125] that governs the convergence of the error dynamics of the observer. Choosing a high value of Q will enforce faster convergence of the observer error and otherwise.

For designing L, note that (3.21) is a nonlinear matrix inequality. Employing the Schur complement, it is equivalent to:

$$\begin{bmatrix} -P+Q & * \\ A^{-w_k}(A-LC) & -P^{-1} \end{bmatrix} < 0 \tag{3.23}$$

Here, '*' represents the symmetric term of the corresponding diagonal one. It is still nonlinear due to the involvement of P and P^{-1}. In this work, the cone-complementarity algorithm [128] is used for solving (3.23) for $w_k \in \{-\bar{b}, \cdots, 0\}$. For this purpose, the following steps are used.

Let $X = P^{-1}$, then (3.23) can be written as:

$$\begin{bmatrix} -P+Q & * \\ A^{-w_k}(A-LC) & -X \end{bmatrix} < 0 \tag{3.24}$$

Since $PX = I$, one may write

$$\begin{bmatrix} P & * \\ I & X \end{bmatrix} \geq 0 \tag{3.25}$$

Note that, for the exact solution, (3.25) should be rank-deficient. Using the above, in line with the cone-complementarity algorithm, Algorithm 1 (presented below) is used for designing L.

3.5 Simulation results

To validate the effectiveness of the proposed predictive control scheme, the networked control setup in Figure 3.1 is used for simulation studies. Two plants are considered — (i) an inverted pendulum and (ii) a DC motor. The design and simulation results are detailed in this section as follows.

Algorithm 2

Step 1: Set $j = 0$ as iteration index. Find a feasible set of $(P_0, X_0, L_0) = (P, X, L)|_{j=0}$ satisfying (3.24) and (3.25).

Step 2: Solve the following LMI optimization problem:

$$\begin{aligned} &\min_{P,X,L} && \text{trace}(XP_j + PX_j) \\ &\text{subject to} && (3.24) \quad \text{and} \quad (3.25). \end{aligned}$$

Set $j = j + 1$ and then $P_j = P, X_j = X$.

Step 3: Solve (3.23) for L with $P = P_j$. If it yields a feasible solution, L is chosen as the desired observer gain and stop. Else, go to Step 2 and iterate.

Example 1. *Consider the discrete-time model of an inverted pendulum [11] with state variables as* $x = \begin{bmatrix} \zeta & \dot{\zeta} & \theta & \dot{\theta} \end{bmatrix}^T$, *where* ζ *and* θ *represent the displacement of the cart and the pendulum angle from the vertical position, respectively. With the sampling period* $T = 0.01s$, *the plant dynamics can be described by* (3.1) *with*

$$A = \begin{bmatrix} 1.0000 & 0.0100 & 0.0001 & 0 \\ 0 & 0.9982 & 0.0267 & 0.0001 \\ 0 & 0 & 1.0016 & 0.0100 \\ 0 & -0.0045 & 0.3119 & 1.0016 \end{bmatrix}, \; B = \begin{bmatrix} 0.0001 \\ 0.0182 \\ 0.0002 \\ 0.0454 \end{bmatrix}, \text{ and } C^T = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

In the feedback and forward networks, the random delays and successive dropouts are considered in the range of $[0, 10]$ *and* $[0, 4]$, *respectively. The values* $\epsilon_b = 0.01$ *and* $\epsilon_r = 0.09$ *are set for the feedback and forward path ET, respectively, with the consecutive maximum dropout count set to* $E_b = E_r = 8$. *Thereby, maximum delays (including the packet losses) in the feedback and forward paths are* $\bar{b} = \bar{r} = 22$. *For the design of* K, *LQR is used with* $Q = I$ *and* $R = 5$. *The following gain matrices are obtained for the controller and observer:*

$$K^T = \begin{bmatrix} 0.4201 \\ 1.1089 \\ -16.8297 \\ -3.1228 \end{bmatrix} \quad \text{and} \quad L = \begin{bmatrix} 1.0023 & 0.0005 \\ 0.2295 & 0.0753 \\ 0.0006 & 1.0345 \\ 0.0629 & 3.6068 \end{bmatrix}$$

with

$$P = \begin{bmatrix} 0.1875 & 0.0429 & -0.0036 & 0.0001 \\ 0.0429 & 0.0409 & 0.0004 & 0.0114 \\ -0.0036 & 0.0004 & 0.0244 & 0.0799 \\ 0.0001 & 0.0114 & 0.0799 & 0.2898 \end{bmatrix}.$$

For comparison, a local controller that implements the state feedback controller along with the observer without the network communication is considered. The local controller, being an ideal one for the predictive control of the NCS, is expected to yield a better response. The state trajectories using the proposed sequential observer-based state feedback control (with $s = 0, 1, 2$ and 4) and their comparison with the local control are shown in Figure 3.4, with the initial state values of the plant and observer being $x_0 = \begin{bmatrix} 0.98 & 0 & 0.2 & 0 \end{bmatrix}^T$ and $\hat{x}_0 = \begin{bmatrix} 0.1 & 0 & 0 & 0 \end{bmatrix}^T$, respectively. It is observed that the control performance is improved when the number of levels in the sequential observer is increased utilizing the sequential output information. This corroborates the efficacy of using the sequential observer, though only output information is transmitted through the network.

Since practical systems suffer from model uncertainties and disturbances, the predictive control performance is also evaluated for additive noise in the system model. An additive noise n_k is considered in the plant dynamics as $x_{k+1} = Ax_k + Bu_k + n_k$ with n_k having uniform distribution in $[0, 0.05]$. The simulation results for the NCS with additive system noise with the predictive controller are shown in Fig. 3.5. It can be seen that the controller yields similar performance even in the presence of disturbances in the plant.

Example 2. *Next, control of a DC motor [132] is considered. The network environment (event-triggering conditions, delays, and packet losses) is kept the same as in Example 1. The plant dynamics 3.1 obtained for $T = 1$ ms as* $A = \begin{bmatrix} 1.0087 & -0.0051 \\ 0.0711 & 0.9667 \end{bmatrix}$, $B = \begin{bmatrix} 0.0581 & -0.1684 \end{bmatrix}^T$, $C = \begin{bmatrix} 4.2400 & 2.6720 \end{bmatrix}$, *with state variables as* $x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$, *and with initial state values set as* $x_0 = \begin{bmatrix} 100 & 10 \end{bmatrix}^T$. *The initial state of the observer and the values of Q and R are the same as in Example 1. Then, he designed values are obtained as*

$$K^T = \begin{bmatrix} -11.5284 \\ 0.8614 \end{bmatrix} \quad \textit{and} \quad L = \begin{bmatrix} 0.1697 \\ 0.1148 \end{bmatrix}, \quad \textit{with} \quad P = \begin{bmatrix} 1.0053 & -0.0023 \\ -0.0023 & 0.9981 \end{bmatrix}.$$

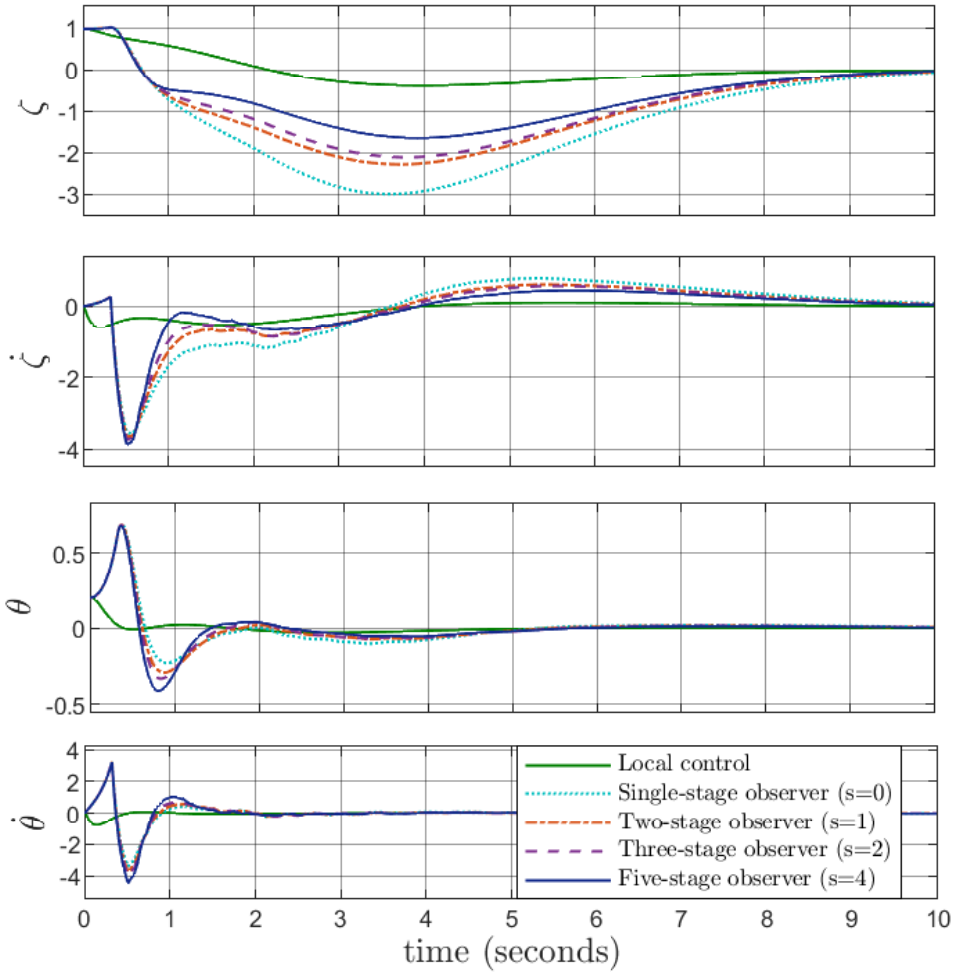

Figure 3.4: Comparison of inverted pendulum state variations without using network (Local control) and with network and using various number of stages of sequential observer

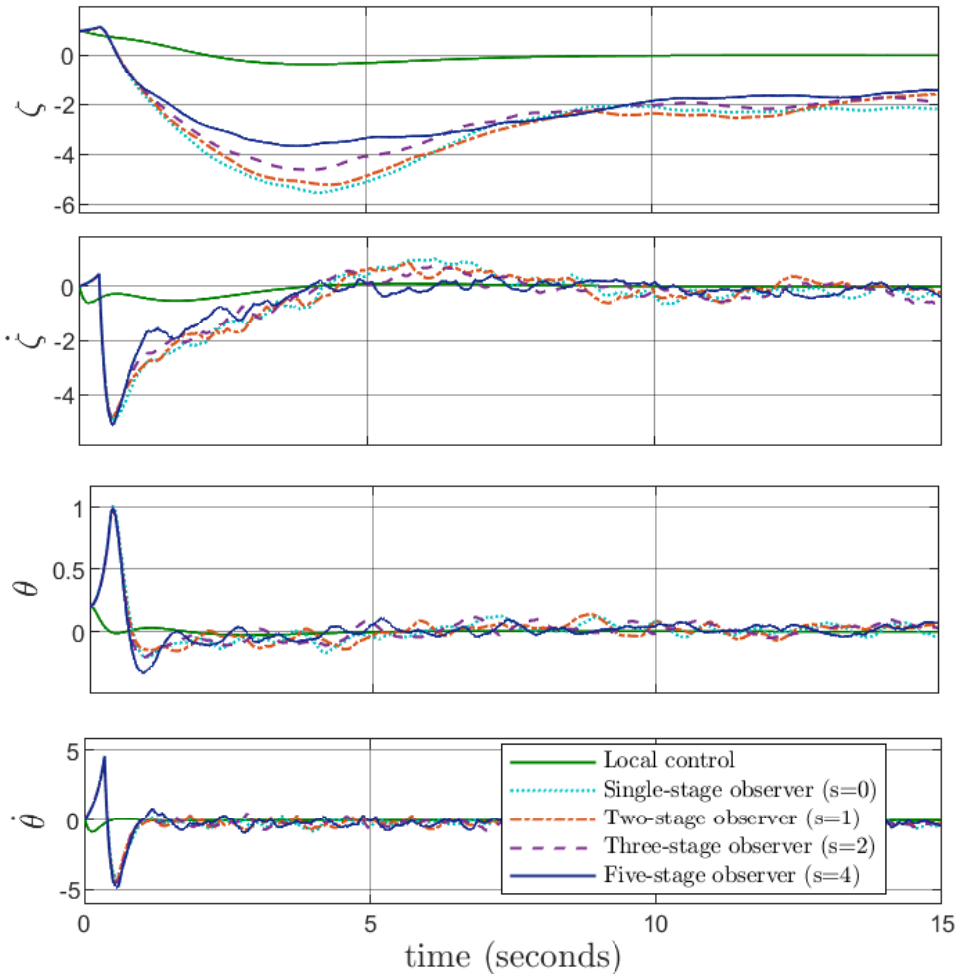

Figure 3.5: State variations of the inverted pendulum with disturbances for the NCS utilizing various number of stages of sequential observer

Figure 3.6 shows the state propagation of the NCS along with the local control (without the network). From the figure, one may observe that the initial free response for the delay duration and then the predictive controller leads to a transient response similar to that with the local control. In the case of smaller delay values, the response gets better than is observed in this example.

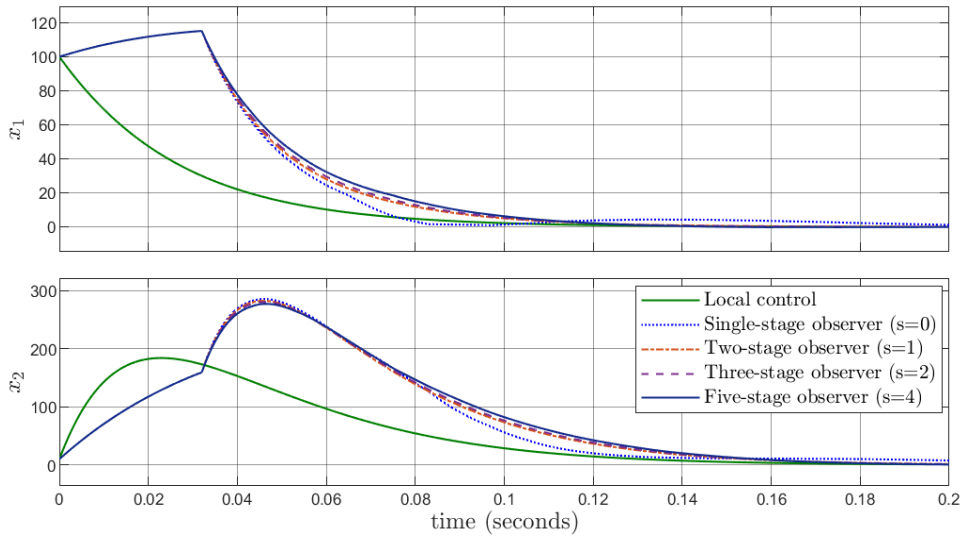

Figure 3.6: Comparison of DC motor state variations for various number of stages of sequential observers

3.6 Summary

This chapter proposes a new approach to improve the performance of a predictive controller for NCSs affected by network-induced time delay and packet dropout. Transmission of output information is considered for reducing the packet size, and trade-offs for using a sequential observer along with the transmission of sequential output information in a single packet are studied. Further, event-triggering is utilized in both the feedback and forward paths to address the challenges of limited bandwidth. A predictive controller is designed to compensate for the delays and dropouts introduced by the network and event-triggering. A remote-side sequential observer is designed to use successive output data transmitted in a single packet to mitigate the challenges of transmitting output information. The effectiveness of the proposed approach is evaluated by comparing its performance with that of a plant with local control (without the network) and with the

NCS that uses different levels of observer scenarios, including only the instantaneous output transmission. The simulation results demonstrate improved control performance of the NCS with the sequential observer.

Although event-triggering has been successfully implemented for the NCS, efficiently utilizing the unused network channels remains a challenging issue. The next chapter will address this challenge by exploring the implementation of predictive triggering to further improve communication efficiency in the NCS.