Chapter 2

Predictive control of networked control system with event-triggering in two channels

2.1 Introduction

Recently, NCSs have garnered significant interest due to their extensive range of applications, including the collaborative control of mobile robots, intelligent transportation systems, unmanned aerial vehicles, remote surgery, human surveillance systems, and visual servo control [7, 119]. An NCS implements a closed-loop control mechanism that utilizes communication networks to enable a more cost-effective, remotely operated, and adaptable system [78]. Nevertheless, the presence of unpredictable time delays and packet losses, as well as the limited availability of bandwidth resources in shared communication networks, pose significant challenges in the implementation, design of control systems, and overall performance of an NCS [82–84].

Many control techniques have been devised to enhance the efficiency of NCSs when dealing with the impact of unpredictable delays and dropouts. In [93], a criterion for delay-dependent stability analysis has been developed for the purpose of designing a static output feedback stabilizing controller. The work in [1] provides the necessary conditions for stability and feedback stabilization in NCS. The stability conditions of NCS experiencing both arbitrary and Markovian packet losses have been established by [94]. These conditions were derived using a Lyapunov approach that takes into account the

dependence of packet loss on system stability. In the study conducted by [95], an output-feedback controller was developed that utilizes an estimation technique to handle the impact of random packet loss, which is represented by a Bernoulli process.

Moreover, due to the presence of network delays and packet drops, NCS makes extensive use of predictive control. NCS was subjected to a delay compensation scheme utilizing Smith predictors in [97, 98]. With bounded delays and packet drops in both the forward and feedback channels, the performance of digital Smith predictor-based NCS has been demonstrated in [99]. NCS also employs Networked Predictive Control (NPC) widely [106, 107, 120]. Such predictive controllers produce control predictions by incorporating delay and packet loss data in order to mitigate the impact of delay. A time-varying predictive output feedback controller has been developed in [104] to guarantee system stability. The predictive controller of the scheme comprises two components — (i) a control prediction generator and (ii) a network delay compensator designed to address communication delays and dropouts. Similarly, [105] and [20] have utilized NPC to guarantee the stability of the overall system. The preceding methods utilize model-based NPC by incorporating system model information into the predictor. In contrast, an alternative model-free approach, employing data-based NPC, has been implemented in [106].

It is worth noting that the majority of the aforementioned studies incorporate a state feedback controller with an observer as the central component of the control scheme. This observer can be applied either on the plant side, as demonstrated in [11, 12], or on the controller (remote) side, as illustrated in [20–22]. In the case of a plant-side observer, the transmission of state information occurs over the network, contrasting with the controller-side approach, where only the output is transmitted. Importantly, employing an observer on the controller side in an NCS proves advantageous. This is because transmitting output information requires a smaller packet size compared to transmitting state information, especially for systems with a large number of states, given the limited size of a packet [18, 23].

Conversely, within an NCS that employs a communication network, the use of a high rate of periodic communication can impede the optimal utilization of shared network resources. Instead of relying on such periodic communication, Event-Triggered Transmissions (ETT) are employed to construct a resource-aware NCS. ETT dynamically determines and utilizes network transmission intermittently based on specific requirements.

The studies presented in [38–40,42–45,121–123] have integrated ETT into the NCS framework. In a novel approach, the co-design of an observer-based coding-decoding scheme and the N-step MPC is introduced in [46] to achieve the desired control performance within the constraints of a limited-bandwidth network.

To address the challenges posed by random delays, dropouts in the network, and limited network bandwidth, recent studies such as [11, 12, 58, 59] have underscored the integration of NPC in control alongside ETT. However, a prevalent limitation in most of these works is the incorporation of Event Triggering (ET) solely in either the feedback or forward channel. This approach restricts the optimal utilization of ET, and the communication network remains strained unless ETT is implemented in both channels. The scenario where the controller is remote and networked in the feedback channel while the forward path employs a dedicated communication channel represents a limited application context. A more general situation involves both the feedback and forward paths utilizing the same, often ad-hoc, network channel. Consequently, using a communication network in both channels becomes a necessity from an application standpoint.

In this context, when the feedback path employs event-triggering to alleviate network load, the forward path still communicates in a time-triggering fashion, exerting a similar load on the network. Furthermore, when the same communication channel is employed in both paths, the advantages gained by ET in the feedback path may be compromised due to time-triggering in the forward path. This chapter addresses the comprehensive challenge of employing ET in both channels to achieve a more balanced and effective network utilization.

While a few studies, such as the one presented in [124], have explored the use of predictive control and ET in both the forward and feedback channels, the potential advantages of having an observer on the controller side have not been adequately addressed. In comparison with the findings in [124], a significant departure in the proposed work lies in the placement of the observer on the controller side of the network and the inclusion of passive dropouts.

This work introduces the utilization of ET in both channels, coupled with the placement of the observer on the controller side. This configuration enables the transmission of only the output through the feedback channel, utilizing a comparatively reduced packet size.

Compensating for delays and dropouts is achieved through the incorporation of a predictive controller, while the conservation of communication resources in both the feedback and forward channels is realized by implementing ET. This approach is denoted as Two-channel Event-triggered Predictive Control (TEPC). Additionally, a remote observer situated on the controller side is devised to counteract the network's impact on state estimates. The proposed TEPC scheme effectively addresses delays and packet dropouts within a unified framework.

A brief outline of this chapter is organized as follows. In Section II, the system model and the proposed TEPC scheme are shown. Section III presents the control design of the proposed TEPC scheme. The required gain matrices are obtained in section IV. Section V illustrates the effectiveness of the proposed scheme by comparing the proposed scheme with existing results by simulating the results of a double integrator and an inverted pendulum system. The conclusions are finally drawn in Section VI.

2.2 Preliminaries and problem formulation

The TEPC scheme considered in this chapter is shown in Fig. 2.1. The discrete-time model of the plant to be controlled is

$$x_{k+1} = Ax_k + Bu_k, \qquad y_k = Cx_k. \tag{2.1}$$

where $x_k \in \mathcal{R}^n$, $u_k \in \mathcal{R}^m$, and $y_k \in \mathcal{R}^p$ represent the system state, input, and output, respectively, at k^{th} time-instant. A, B, and C are the system matrices satisfying the controllability and observability of the plant model (2.1). The information transmitted through the communication network is considered to be time-stamped, and all the blocks that are using the time-stamps are assumed to be synchronized. Event-triggered communication is considered in both the feedback and forward channels. An observer is designed on the remote side to estimate the states from the delayed output information. Then a state predictor is used to construct the predicted states, and thereby the state feedback control is generated. The consecutive predicted control inputs are stored in a buffer, and an array of advanced control inputs is generated to communicate them to the plant side. Finally, a selector chooses the appropriate control input based on the delay in the forward channel. The notations used in this chapter and the NCS parameters are given in Table 2.1.

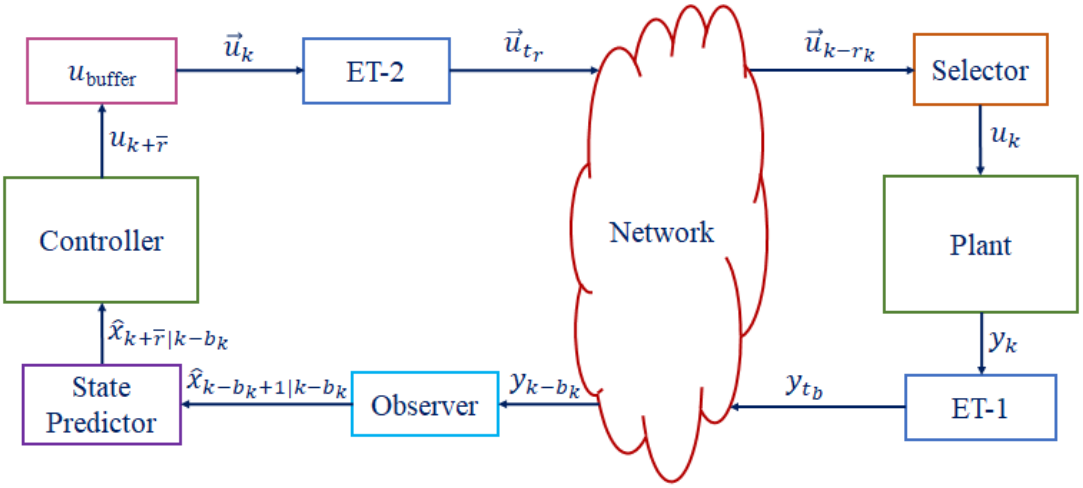

Figure 2.1: The two-channel event-triggered predictive control (TEPC) scheme for NCS

The plant output information y_k is fed to the ET-1 block, which decides whether to transmit y_k through the network or not. Consider y_{t_b} denotes the event-triggered output with the suffix t_b, $b \in \mathcal{N}$, represents the time-instants of y_k at which the transmission event is triggered. In addition, these transmitted packets undergo random delays and dropouts due to the network. The dropouts that occur due to the network are unavoidable and are considered passive dropouts, whereas the dropouts that occur due to the use of ET are obtained consciously to save communication bandwidth and are considered active dropouts. An illustration of active dropouts due to ET, network delays, and passive dropouts is shown in Fig. 2.2. The dropouts that occur at the ET are the active ones, and passive dropouts occur at the network level. Whenever a packet dropout (either active or passive) occurs, the latest packet at the receiver (observer input) is used again, so the delay value due to the dropout is increased by one at most. Thus, the count of dropouts is added to the delay value, and the data is treated accordingly. Let b_k be the random delay in the feedback path, then it obeys that $b_{k+1} \leq b_k + 1$. The value of t_b is related to the delay value b_k by $b_k = k - t_b$, i.e., the delay value at the receiver side is obtained by taking the difference between the instantaneous value of k and the time-stamp of the received packet.

Let D_b be the upper bound of delay in the feedback network; P_{b_a} and P_{b_p} be the upper bounds of the consecutive active and passive dropouts, respectively. Then $\bar{b} = D_b + P_{b_a} + P_{b_p}$ is the maximum value of the total delay in the feedback path.

Similarly, in the forward path, the ET-2 block takes care of the ETT. Following

Table 2.1: Notations pertaining to the NCS

k	The present time-instant
D_b	Maximum amount of feedback network delay
P_{b_a}	Maximum consecutive no. of feedback active dropouts
P_{b_p}	Maximum consecutive no. of feedback passive dropouts
b_k	Instantaneous feedback path delay
$t_b, b \in \mathcal{N}$	Event-triggered time-instants in feedback path
y_{t_b}	Event-triggered output values
$\bar{b}$	$D_b + P_{b_a} + P_{b_p}$
b_k	$k - t_b$
D_r	Maximum amount of forward network delay
P_{r_a}	Maximum consecutive no. of forward active dropouts
P_{r_p}	Maximum consecutive no. of forward passive dropouts
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 natural numbers
$\mathcal{R}^*$	The set of all *-tuples of real numbers

Table 2.1, the maximum delay in the forward path is $\bar{r} = D_r + P_{r_a} + P_{r_p}$.

Remark 1. *The main advantage of using an observer on the controller side is the reduction in the packet size. If the observer is on the plant side, then all the state information has to be communicated through the network, which could be large for systems with a considerable number of states. When the observer is on the controller side, then the transmission of only the output is carried out with a comparatively smaller packet size. It may be noted that packet size also affects the network performance [23], [18].*

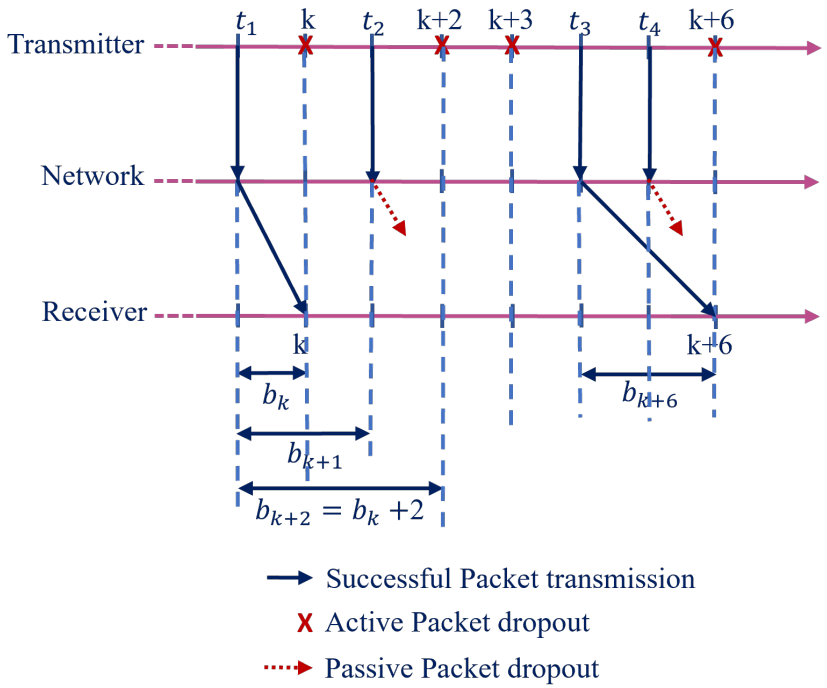

Figure 2.2: A representation of event-triggered packet transmission in the network with delays and dropouts

2.2.1 Event-triggering at feedback channel

ET is incorporated to intermittently transmit y_k to the controller side through the following logical condition:

$$t_{b+1} = \begin{cases} \min\{k|(\tilde{y}^T H_b \tilde{y}) > \epsilon_b (y_k^T H_b y_k)\}, \\ \text{when } k < t_b + E_b + 1 \\ t_b + E_b + 1, \text{ when } k \geq t_b + E_b + 1 \end{cases} \tag{2.2}$$

where $\tilde{y} = y_k - y_{t_b}$; t_b is an ET instant and t_{b+1} represents the next ET instant to be determined; $H_b > 0$ is of appropriate dimension and $0 < \epsilon_b < 1$ is a scalar to be chosen; E_b is the maximum number of active dropouts, and therefore the maximum time-interval between any two triggering instants is $E_b + 1$.

The choice of H_b and ϵ_b is often decided based on the trade-off between the system performance and communication bandwidth availability [11].

2.2.2 Observer

The observer block consists of an observer buffer and then an observer. Upon receiving a new packet, the observer buffer compares the time stamp with the available one and retains only the latest one. In case of a packet drop, the old data is used, i.e., the output information received at the observer can be written as:

$$y_{k-b_k} = \begin{cases} y(\max\{t_b\})|k-b_k \geq k-1-b_{k-1} \\ y_{k-1-b_{k-1}}|k-b_k < k-1-b_{k-1} \end{cases}$$

With y_{k-b_k} as the input, the functionality of the observer is shown in Fig. 2.3. The observer uses the output data from the buffer to construct the one-step-ahead state estimate as:

$$\hat{x}_{k-b_k+1|k-b_k} = A\hat{x}_{k-b_k|t_{b_l}} + Bu_{k-b_k} + L(y_{k-b_k} - C\hat{x}_{k-b_k|t_{b_l}}) \tag{2.3}$$

where L is the observer gain matrix to be designed and $\hat{x}_{k-b_k|t_{b_l}}$ is equivalent to the observed state value at $(k-1)$ instant given the latest packet received then t_{b_l} indicates the timestamp of the latest packet received at $(k-1)$. The values of $\hat{x}_{k-b_k|t_{b_l}}$ and u_{k-b_k} are obtained from x_{buffer} and u_{buffer}, respectively, depending on the delay value b_k. The x_{buffer} and u_{buffer} are the two buffers for storing the predicted state $\hat{x}_{k|k-b_k}$ and control $u_{k+\bar{r}}$ values. The sizes of these buffers are derived according to the maximum delay values, which are considered to be known.

Defining the observer error as $e_k = x_k - \hat{x}_{k|k-1}$, from (2.1) and (2.3), one can write

$$\begin{aligned} e_{k-b_k+1} &= x_{k-b_k+1} - \hat{x}_{k-b_k+1|k-b_k} && \text{(2.4a)} \\ &\approx Ax_{k-b_k} + Bu_{k-b_k} - A\hat{x}_{k-b_k|t_{b_l}} - Bu_{k-b_k} \\ &\quad - L(Cx_{k-b_k} - C\hat{x}_{k-b_k|t_{b_l}}) && \text{(2.4b)} \\ &= (A-LC)(x_{k-b_k} - \hat{x}_{k-b_k|t_{b_l}}) && \text{(2.4c)} \\ &= (A-LC)e_{k-b_k} \\ & && \text{(2.4d)} \end{aligned}$$

Remark 2. *As an example case, a situation of network delays and active and passive dropouts is shown in Fig. 2.2. One can see that the observer can use only y_{t_1} for $(k+1)$ to $(k+5)$ time-instants since it is the latest packet received till $(k+5)$. Then, at $(k+6)$ instant, new data arrives with timestamp t_3. Therefore, the observed state value at time*

$(k{+}6)$ *depends on* y_{t_3}. *Then the definitions of observer error at* $(k{+}5)$ *and* $(k{+}6)$ *are given as* $e_{k+5-b_{k+5}} = x_{k+5-b_{k+5}} - \hat{x}_{k+5-b_{k+5}|t_1}$ *and* $e_{k+6-b_{k+6}} = x_{k+6-b_{k+6}} - \hat{x}_{k+6-b_{k+6}|t_3}$, *respectively. This shows that the observed state values are not obtained given the previous output value all the time because of the delays and packet drops. Therefore, two consecutive errors may not be the same as defined in* (2.4a) *and* (2.4c). *Depending on the instantaneous delay value, the estimate differs, and thereby the* e_k *and* e_{k-1} *may not be exact. Such a difference may introduce errors in the state estimate though the delay compensation is carried out through the prediction. Therefore, the choice of the controller and observer gains is tricky in the sense that these should be considered robust to accommodate such errors in the closed-loop operation.*

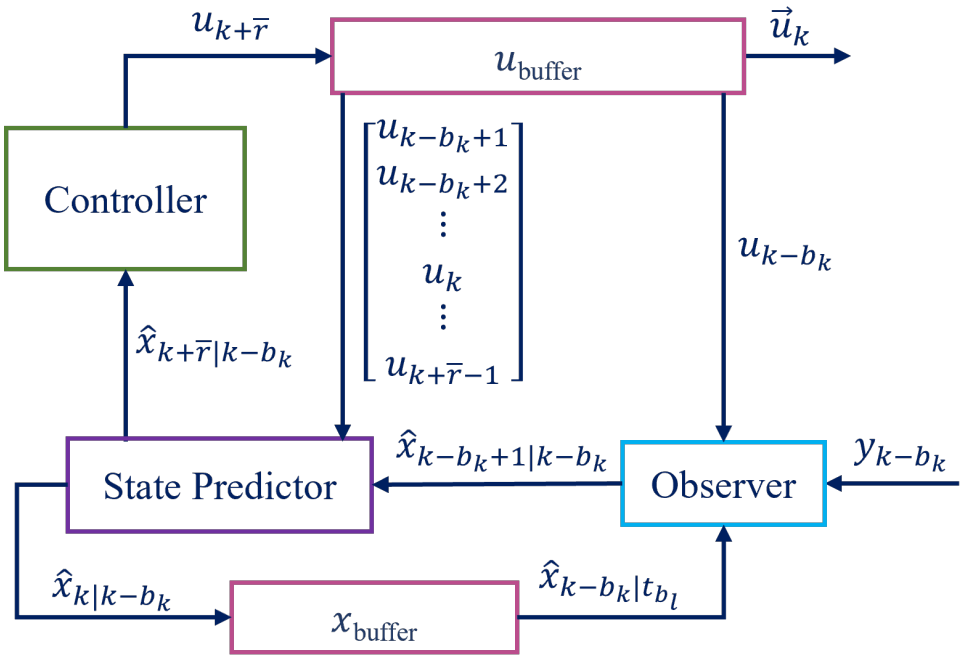

Figure 2.3: The block components at the remote side of the proposed TEPC scheme

2.2.3 State predictor

Once the new output information has been incorporated into the observed state through the observer, compensation for the delay is carried out based on model-based prediction. The number of prediction steps is equal to the sum of the values of instantaneous feedback delay b_k and the maximum delay $\bar{r}$ in the forward path. The predictor can be written as:

$$\hat{x}_{k-b_k+n|k-b_k} = A\hat{x}_{k-b_k+n-1|k-b_k} + Bu_{k-b_k+n-1}, \quad \text{for } n = 2, 3 \cdots, b_k + \bar{r} \tag{2.5}$$

The control inputs $u_{k-b_k+1}, \cdots, u_k, \cdots, u_{k+\bar{r}-1}$ required to make these predictions are drawn from the u_{buffer} as shown in Fig. 2.3. Then, combining (2.3) and (2.5) results in

$$\hat{x}_{k+\bar{r}|k-b_k} = A^{b_k+\bar{r}}\hat{x}_{k-b_k|t_{b_l}} + \sum_{m=1}^{b_k+\bar{r}} A^{m-1}Bu_{k+\bar{r}-m} + A^{b_k+\bar{r}-1}L(y_{k-b_k} - C\hat{x}_{k-b_k|t_{b_l}}) \tag{2.6}$$

2.2.4 State-feedback controller

The predicted state information is then used to construct the predicted control input $u_{k+\bar{r}|k-b_k}$ incorporated through a state feedback control law $u_k = Kx_k$, where $K \in \mathcal{R}^m$ is the controller gain matrix to be designed. The predicted control input value is given by

$$u_{k+\bar{r}} = K\hat{x}_{k+\bar{r}|k-b_k} \tag{2.7}$$

or

$$u_k = K\hat{x}_{k|k-d_k} \tag{2.8}$$

where $d_k = \bar{r} + b_{k-\bar{r}}$. This predicted control input $u_{k+\bar{r}}$ is then incorporated into the u_{buffer} that contains $\vec{u}_k = [u_k^T \; u_{k+1}^T \; \cdots \; u_{k+\bar{r}}^T]^T$. The oldest value u_{k-1} from the array created at the previous instant i.e. $\vec{u}_{k-1} = [u_{k-1}^T \; u_k^T \; \cdots \; u_{k-1+\bar{r}}^T]^T$ is discarded in the process. This way, the length of the vector $\vec{u}$ is always $\bar{r} + 1$. Such a prediction helps compensate for the forward path delay.

2.2.5 Event-triggering at forward channel

The mechanism for ET-2 in the forward channel is given by

$$t_{r+1} = \begin{cases} \min\{k | (\tilde{u}^T H_r \tilde{u}) > \epsilon_r (\vec{u}_k^T H_r \vec{u}_k)\}, \\ \text{when } k < t_r + E_r + 1 \\ t_r + E_r + 1, \text{when } k \geq t_r + E_r + 1 \end{cases} \tag{2.9}$$

where $\tilde{u} = \vec{u}_k - \vec{u}_{t_r}$. The notations H_r, E_r, ϵ_r, and t_r are all similar to that of the event-triggering condition taken at the feedback channel.

2.2.6 Selector

The selector block consists of a buffer and a selector. When the buffer receives a new packet, it compares the time stamps and retains only the packet with the latest time

stamp that can be expressed as:

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

As it is observed from the packets that are generated by the u_{buffer} at each time-step,

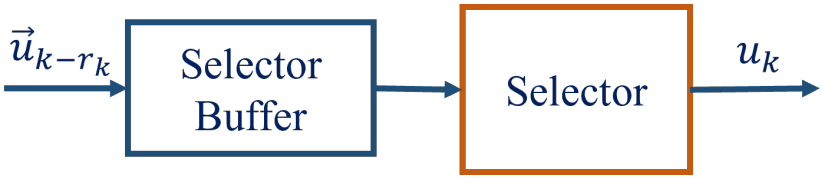

Figure 2.4: Buffer at the selector of the TEPC scheme

the predicted control input $u_{k+\bar{r}}$, which is calculated by the controller, is available in $\bar{r}$ consecutive packets. Depending on the instantaneous value of forward path delay r_k (including both the active and passive dropout), the u_k is selected. This way, the constant $\bar{r}$ number of predictions for compensating the forward path delay is achieved using the combined logic of u_{buffer} and the selector.

Remark 3. *The forward path delays and dropouts (including the active dropouts due to forward path ET) can be compensated for any forward path delay by utilizing the* u_{buffer}*. By maintaining the size of* $u_{buffer} = \bar{r} + 1$ *where* $\bar{r} = D_r + P_{r_a} + P_{r_p}$*, the transmission of advanced control inputs suitable for these values can always be facilitated. However, the size of the packet due to large delay may cause implementation issues due to limited packet size.*

2.3 Predictive control scheme

This section presents 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 can be:

$$x_{k+1} = (A + BK)x_k - BK(x_k - \hat{x}_{k|k-d_k}) \tag{2.10}$$

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

$$\begin{aligned} \alpha_k &= Ax_{k-1} + Bu_{k-1} - (A\hat{x}_{k-1|k-d_k} + Bu_{k-1}) \\ &= A(x_{k-1} - \hat{x}_{k-1|k-d_k}) \end{aligned}$$

By successive replacements, one can write

$$\alpha_k = A^{d_k-1}(x_{k-d_k+1} - \hat{x}_{k-d_k+1|k-d_k}) \tag{2.11}$$

At the observer end, in a similar fashion, let $\beta_k = x_{k-d_k+1} - \hat{x}_{k-d_k+1|k-d_k}$. Then,

$$\begin{aligned}\beta_k = &Ax_{k-d_k} + Bu_{k-d_k} - A\hat{x}_{k-d_k|t_{b_l}} + Bu_{k-d_k} \\ &- L(Cx_{k-d_k} - C\hat{x}_{k-d_k|t_{b_l}})\end{aligned}$$

where t_{b_l} indicates the timestamp of the latest packet received at $k-d_k$. Defining $e_{k-d_k} = x_{k-d_k} - \hat{x}_{k-d_k|t_{b_l}}$, one obtains

$$\beta_k = (A - LC)e_{k-d_k} \tag{2.12}$$

Replacing (2.12) into (2.11) and then (2.11) into (2.10) yields

$$x_{k+1} = (A + BK)x_k - BKA^{d_k-1}(A - LC)e_{k-d_k} \tag{2.13}$$

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

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

$$\begin{aligned}e_{k+1-d_{k+1}} &= x_{k+1-d_{k+1}} - \hat{x}_{k+1-d_{k+1}|k-d_k} \\ &= x_{k+1-\bar{r}-b_{k-\bar{r}+1}} - \hat{x}_{k+1-\bar{r}-b_{k-\bar{r}+1}|k-d_k} \\ &= x_{k-\bar{r}-b_{k-\bar{r}}} - \hat{x}_{k-\bar{r}-b_{k-\bar{r}}|k-d_k} \\ &= x_{k-d_k} - \hat{x}_{k-d_k|k-d_k} \\ &= e_{k-d_k}\end{aligned} \tag{2.14}$$

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

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

$$\begin{aligned}e_{k+1-d_{k+1}} &= x_{k+1-d_{k+1}} - \hat{x}_{k+1-d_{k+1}|k-d_k} \\ &= x_{k+1-d_k-s_k} - \hat{x}_{k+1-d_k-s_k|k-d_k} \\ &= A(x_{k-d_k-s_k} - \hat{x}_{k-d_k-s_k|k-d_k}) \\ &\vdots \\ &= A^{-s_k}(x_{k+1-d_k} - \hat{x}_{k+1-d_k|k-d_k}) \\ &= A^{-s_k}(A - LC)e_{k-d_k}\end{aligned} \tag{2.15}$$

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:

$$z_{k+1} = \Phi z_k$$

where

$$\Phi = \begin{cases} \begin{bmatrix} A+BK & -BKA^{d_k-1}(A-LC) \\ 0 & I \end{bmatrix}, \\ \text{when } s_k = 1 \end{cases} \tag{2.16a}$$

$$\begin{cases} \begin{bmatrix} A+BK & -BKA^{d_k-1}(A-LC) \\ 0 & A^{-s_k}(A-LC) \end{bmatrix}, \\ \text{when } s_k \in \{-\bar{b}, \cdots, 0\} \end{cases} \tag{2.16b}$$

It is clear that the stability of the closed-loop NCS can be analyzed through the augmented system dynamics (2.16).

2.4 Controller and observer design

This section considers the dynamics (2.16) 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 for s_k may occur at most $\bar{b}$ times consecutively. Again, when the maximum count of dropouts is reached, the system dynamics switch to Case-II. Therefore, Case-I cannot happen at all times. The stability condition for (2.16) is derived next only for Case-II.

The following result is well known through the separation principle [125] applied through (2.16b).

Theorem 1. *The augmented system* (2.16b) *is asymptotically stable if $A+BK$ is Schur and the system with the following switched system matrix is stable.*

$$A^{-s_k}(A-LC), \quad s_k \in \{-\bar{b}, \cdots, 0\} \tag{2.17}$$

Proof. Straightforward using the separation principle due to the block-diagonal nature of the system matrix. □

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

For controller design, LQR controller [125] is considered in this work, while other methods of state feedback controller design can also be employed that guarantee the stability of $(A + BK)$. Following Remark 2, it can be seen that the overall system dynamics are not exact, for which stability is ensured. A robust controller would work well for this purpose. Several simulation studies were carried out to confirm that the LQR controller works well for such systems, as demonstrated in the next section through case studies.

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

Theorem 2. *The NCS with the augmented system dynamics (2.16b) is asymptotically stable if $A+BK$ is Schur and there exists a $P = P^T > 0$ satisfying the following inequality.*

$$[A^{-s_k}(A-LC)]^T P[A^{-s_k}(A-LC)] - P < 0, \quad s_k \in \{-\bar{b}, \cdots, 0\} \tag{2.18}$$

Proof. The proof follows the common Lyapunov function method [126] for linear systems and hence omitted. □

For designing L, note that (2.18) is a nonlinear matrix inequality. Employing the Schur complement, it can be rewritten as:

$$\begin{bmatrix} -P & * \\ A^{-s_k}(A-LC) & -P^{-1} \end{bmatrix} < 0 \tag{2.19}$$

Here '$*$' indicates the symmetric term of the corresponding off-diagonal one. It is still nonlinear due to the involvement of P and P^{-1}. In this work, the cone-complementarity algorithm [127, 128] is used for solving (2.19) for $s_k \in \{-\bar{b}, \cdots, 0\}$. The algorithm to follow for designing the observer is presented next.

Let $X = P^{-1}$, then (2.19) can be written as:

$$\begin{bmatrix} -P & * \\ A^{-s_k}(A-LC) & -X \end{bmatrix} < 0 \tag{2.20}$$

Since $PX = I$, one may write

$$\begin{bmatrix} P & * \\ I & X \end{bmatrix} \geq 0 \tag{2.21}$$

Note that, for the exact solution, (2.21) should be rank-deficient. Now, the cone-complementarity algorithm can be invoked as given in the following Algorithm 1.

Algorithm 1

Step 1: Set $j = 0$ as the iteration index. Find a feasible set of $(P_0, X_0, L_0) = (P, X, L)|_{j=0}$ satisfying (2.20) and (2.21).

Step 2: Solve the following LMI optimization problem:

$$\begin{aligned} &\min_{P,X,L} && \text{trace}(XP_j + PX_j) \\ &\text{subject to} && (2.20)\text{and}(2.21). \end{aligned}$$

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

Step 3: Solve (2.19) for L with $P = P_j$. If it yields a feasible solution, then L is chosen as the desired observer gain and stop. Else go to Step 2 and iterate.

2.5 Case studies

In this section, the proposed control scheme is applied for two example cases demonstrating its effectiveness.

Example 1: Many systems can be modeled as a double integrator, e.g. rolling mill [1]. A double integrator system is inherently an unstable one. Consider the following double integrator system:

$$\dot{x}_1(t) = x_2(t), \dot{x}_2(t) = u(t), \quad y(t) = x_1(t) \tag{2.22}$$

The discretized system model with sampling period $T_s = 1$ms, corresponding to (A.6) is obtained as $A = \begin{bmatrix} 1.0000 & 0.0010 \\ 0 & 1.0000 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 0.0010 \end{bmatrix}$, and $C = \begin{bmatrix} 1 & 0 \end{bmatrix}$.

The random values of delays and the consecutive dropouts in both the feedback and forward network are considered as $[0, 10]$ and $[0, 4]$, respectively. The consecutive dropout count in the feedback and forward path ET logics is set to $E_b = E_r = 8$. Then the maximum delay values in both the feedback and forward path are $\bar{b} = \bar{r} = 22$. The state feedback LQR gain with $Q = I$, $R = 5$ is obtained as:

$$K = \begin{bmatrix} -0.4449 & -1.0429 \end{bmatrix}$$

The cone-complementarity algorithm is used to design the observer gain as follows:

$$L = \begin{bmatrix} 1.0030 & 0.1221 \end{bmatrix}^T$$

Then the simulation of the proposed NCS is carried out with the parameters in the feedback and forward path ET logic chosen as $\epsilon_b = 0.01$ and $\epsilon_r = 0.09$, respectively. With initial plant and observer states as $x_0 = \begin{bmatrix} 0.98 & 0 \end{bmatrix}^T$ and $\hat{x}_0 = \begin{bmatrix} 0.1 & 0 \end{bmatrix}^T$, respectively, the state trajectories of the controlled system are shown in Fig. 2.5. For a comparison, simulation results for local state feedback control without the network and predictive control are also shown. It can be seen that the state trajectories are almost the same as the locally controlled system.

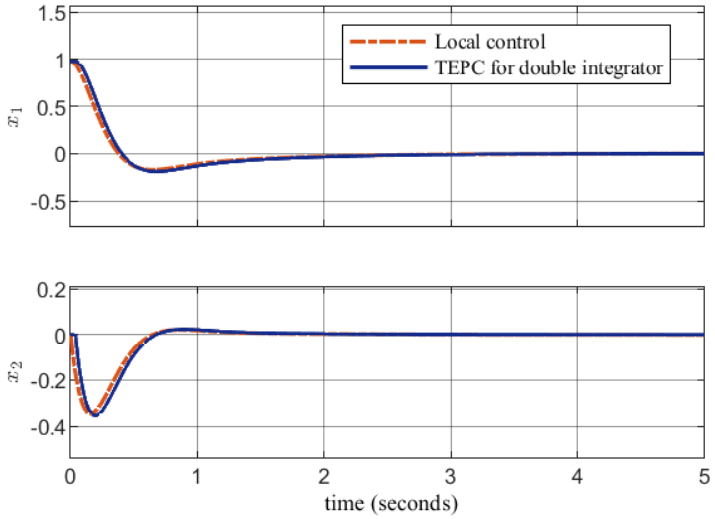

Figure 2.5: The double-integrator system states without network (Local control) and with network along with proposed TEPC scheme

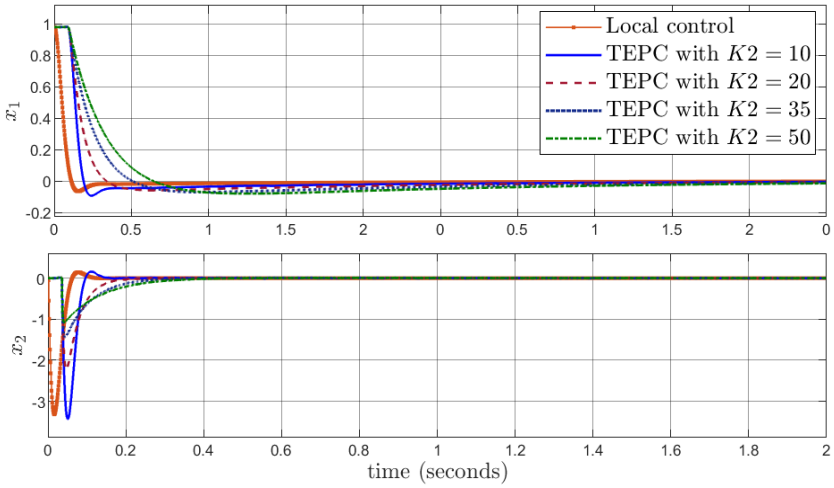

Figure 2.6: System states without network (Local control) and with network and TEPC scheme for the work given in [1] with $K2 = 10, 20, 35$ and 50

For a comparison, consider the results obtained in [1]. For $K = \begin{bmatrix} 50 & 15 \end{bmatrix}$, using only state feedback (no predictive control), the system can not be stabilized for more than a unit delay value in the network loop. However, using the TEPC scheme here, the system remains stable for a much larger value of delays as the simulation shown in Fig. 2.6. This shows the effectiveness of the predictive control scheme compared to a state feedback controller.

The random network delays and the consecutive passive dropouts considered in the simulation are shown in Fig. 2.7. These random values are generated using the MATLAB command *randi*, which generates integers from a random uniform distribution. Note that, for the sake of clarity in presentation, it is demonstrated for a shorter duration. The

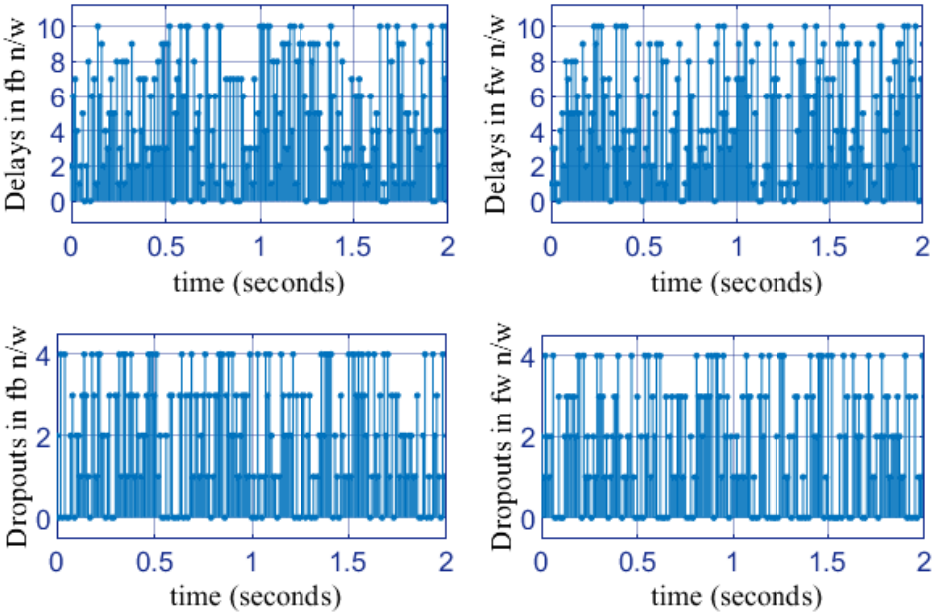

Figure 2.7: Feedback (fb) and forward (fw) network (n/w) delays and dropouts

corresponding ET instants are shown in Fig. 2.8 corresponding to the TEPC simulation in Fig. 2.5. The same Fig. 2.8 is reproduced in Fig. 2.9 and Fig. 2.10 for shorter duration where the data transmission intervals are better visible. Using the proposed method, the transmissions through the feedback and forward channels are reduced significantly by actively dropping 852 out of 1000 samples by the feedback ET, and 766 out of 1000 samples by the forward ET, giving a significant reduction in the transmissions of about 85.2% and 76.6% through the feedback and forward channels, respectively.

The communication and control computation costs are traded off in networked con-

trol. Due to the advancement in cheap computing resources, the computational cost could be sacrificed compared to the fixed and limited communication bandwidth. The event-triggered logic that is incorporated in both channels takes care of this trade-off between the control-computation cost and the communication cost.

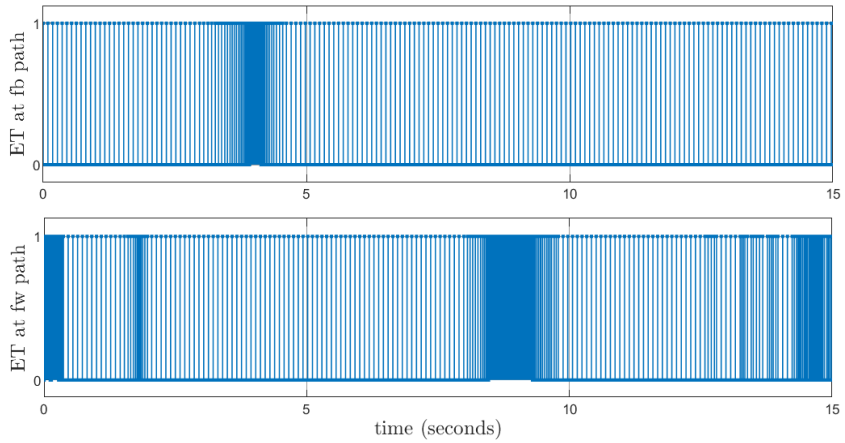

Figure 2.8: Trigger times of ET for Example 1 at feedback (fb) and forward (fw) paths

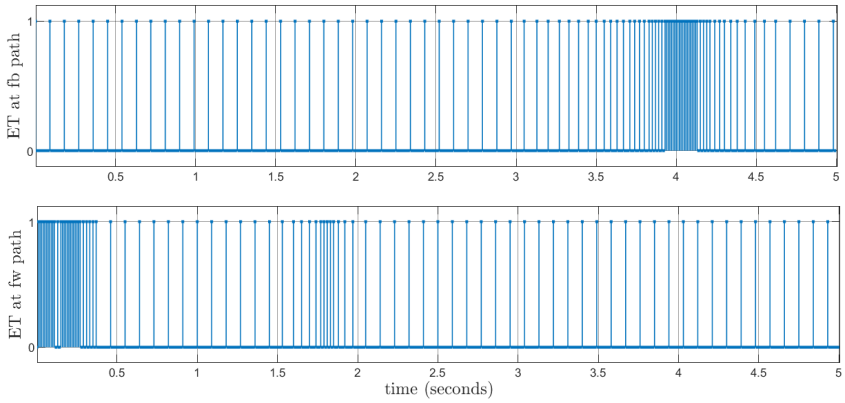

Figure 2.9: Trigger times of ET for Example 1 at feedback (fb) and forward (fw) paths up to 5s

Example 2: Next, the stabilization of an inverted pendulum system [11] through the proposed method is considered. With the state vector, $x = \begin{bmatrix} s & \dot{s} & \theta & \dot{\theta} \end{bmatrix}^T$ and the sampling period $T = 0.01$s, the discrete-time model of the inverted pendulum is given by (2.1) with

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

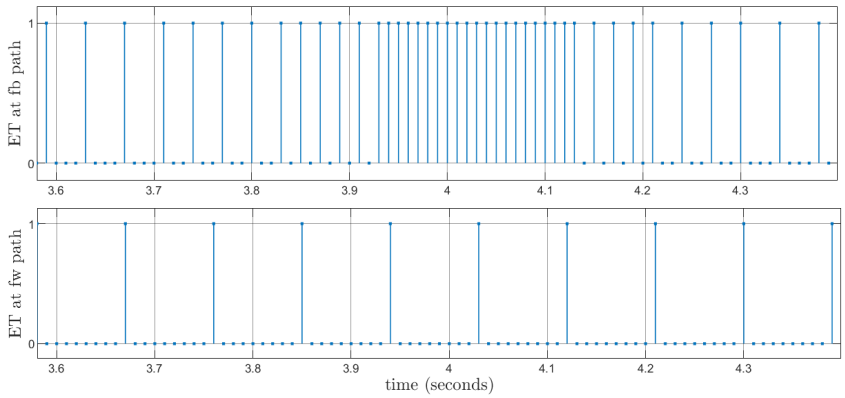

Figure 2.10: Trigger times of ET for Example 1 at feedback (fb) and forward (fw) paths with zoom in around 4s

The number of network delays and dropouts in the system is considered the same as in Example 1. The controller and observer gain matrices are obtained as:

$$K^T = \begin{bmatrix} 0.4201 \\ 1.1089 \\ -16.8297 \\ -3.1228 \end{bmatrix} \quad \text{and} \quad L = \begin{bmatrix} 1.0070 & 0.0036 \\ 0.7470 & 0.2197 \\ 0.0006 & 1.0284 \\ 0.1215 & 3.5018 \end{bmatrix}.$$

Then, with the initial state values of the plant and observer as $x_0 = \begin{bmatrix} 0.98 & 0 & 0.2 & 0 \end{bmatrix}^T$ and $\hat{x}_0 = \begin{bmatrix} 0.1 & 0 & 0 & 0 \end{bmatrix}^T$, respectively, the state trajectories with the local observer-based state feedback control and using the proposed NCS are shown in Fig. 2.11. Fig. 2.12 shows the $2-$norm of the state estimation error, $||e_k||_2$, which indicates the effectiveness of the overall predicted control scheme of the NCS.

The corresponding ET instants with $\epsilon_b = 0.1$ and $\epsilon_r = 0.9$ are shown in Fig. 2.13. Only 121 and 118 times out of 1000 sample times did the ET occur through the feedback and forward channels, respectively. Hence, the channel occupancy is reduced significantly by 87.9% and 88.2%, respectively. The Fig. 2.13 is reproduced in Fig. 2.14 and Fig. 2.15 for shorter durations for the sake of the better visibity of the data transmission intervals.

The performance of the TEPC scheme, which involves the ET in both the feedback and the forward paths, is compared with ET only in the feedback path and when there is no ET involved. The simulation comparison is shown in Fig. 2.16, and the corresponding

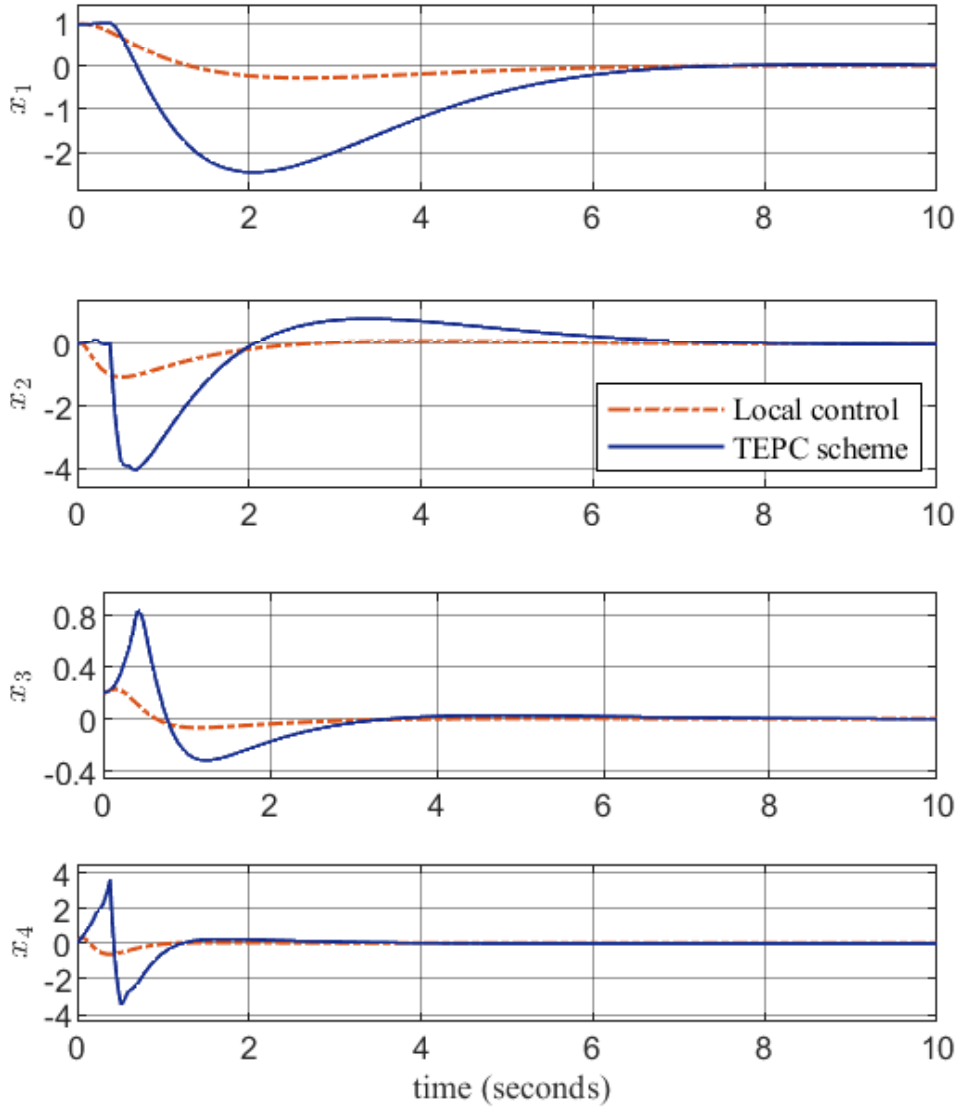

Figure 2.11: The inverted pendulum system states without network (Local control) and with network and the proposed TEPC scheme

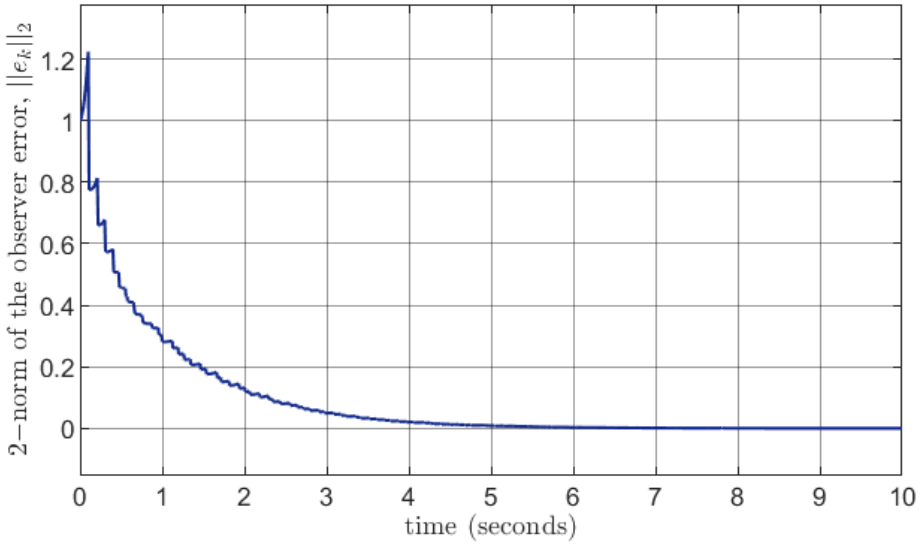

Figure 2.12: The 2−norm of the observer error for the inverted pendulum model with the proposed TEPC scheme

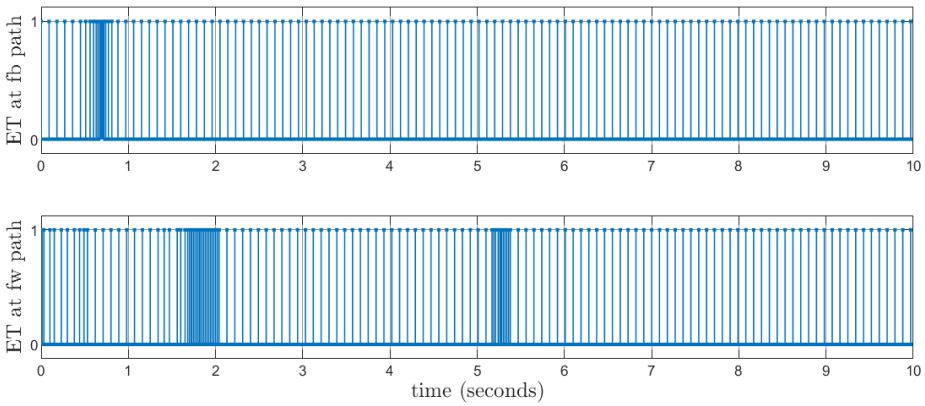

Figure 2.13: Trigger times of ET for Example 2 at feedback (fb) and forward (fw) channels

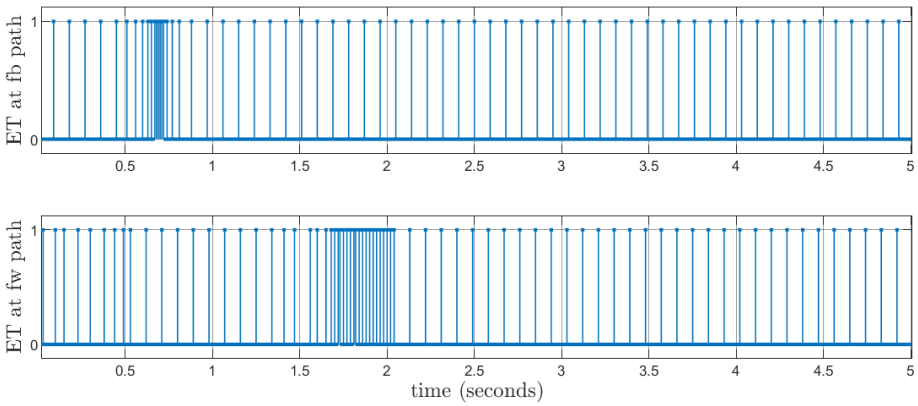

Figure 2.14: Trigger times of ET for Example 2 at feedback (fb) and forward (fw) channels up to 5s

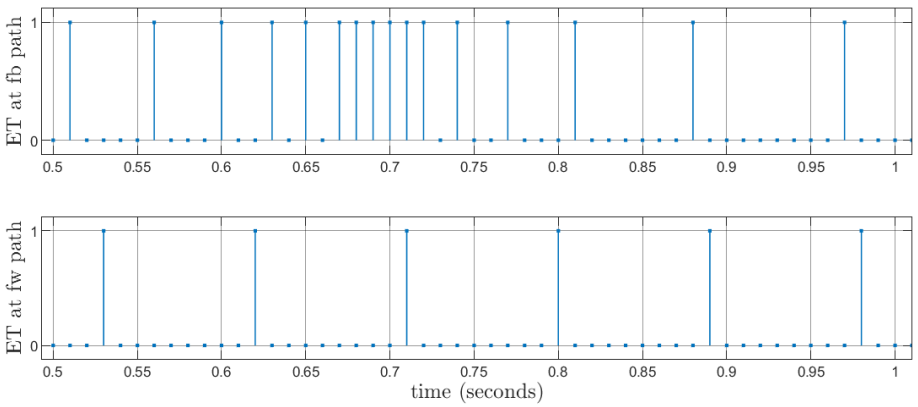

Figure 2.15: Trigger times of ET for Example 2 at feedback (fb) and forward (fw) channels with zoom in about 0.75s

state estimation error is shown in Fig. 2.17. It is clear from both figures that when the system is affected by only network delays and dropouts and not by active drops due to ET, there is not much deviation of the state trajectories compared with the local control performance. When ET is introduced in the feedback path, the effect of active dropouts deviates from the performance a little more. As expected, the performance still deteriorates when ET is utilized on both the feedback and forward paths. Nevertheless, with the proposed method, the performance of the system is maintained even after using the event-triggering in both the feedback and forward channels.

2.5.1 Comparison with existing schemes

The proposed TEPC scheme is compared with the recent works, EPC [11] and ETPC [12] schemes, as shown in Fig. 2.18 and Fig. 2.19, respectively. Similar to the work of the EPC scheme, the NCS of the TEPC scheme is rebuilt by considering ET only on the feedback side with $E_b = 8$ and $\epsilon_b = 0.1$. The observer gain values of both the schemes are taken as they are designed for the scheme in each work, and the LQR controller is considered for both cases to make a comparison of control effects.

$$K^T = \begin{bmatrix} 0.4201 \\ 1.1089 \\ -16.8297 \\ -3.1228 \end{bmatrix}, \quad L_{EPC} = \begin{bmatrix} 1.1360 & -0.0135 \\ 10.9912 & -0.3548 \\ -0.0143 & 1.1612 \\ -0.4609 & 12.0159 \end{bmatrix}, \quad \text{and} \quad L_{TEPC} = \begin{bmatrix} 1.0070 & 0.0036 \\ 0.7470 & 0.2197 \\ 0.0006 & 1.0284 \\ 0.1215 & 3.5018 \end{bmatrix}.$$

From the comparison, it can be seen that the TEPC scheme has achieved a stabilizing performance similar to the EPC scheme, where state transmissions are employed instead of output transmissions. Hence, such a comparative response is expected.

Another comparison is carried out with the ETPC scheme by taking the proposed TEPC scheme with only the feedback network and ET with $E_b = 8$ and $\epsilon_b = 0.1$. The K value is designed using the pole placement method as in [12] for both the schemes and the L value is taken according to the individual schemes.

$$K^T = \begin{bmatrix} 0.4201 \\ 1.1089 \\ -16.8297 \\ -3.1228 \end{bmatrix}, \quad L_{ETPC} = \begin{bmatrix} 1.0125 & -0.0112 \\ 1.2684 & -1.1030 \\ -0.0163 & 1.0346 \\ -1.6361 & 3.6202 \end{bmatrix}, \text{and} \quad L_{TEPC} = \begin{bmatrix} 1.0070 & 0.0036 \\ 0.7470 & 0.2197 \\ 0.0006 & 1.0284 \\ 0.1215 & 3.5018 \end{bmatrix}.$$

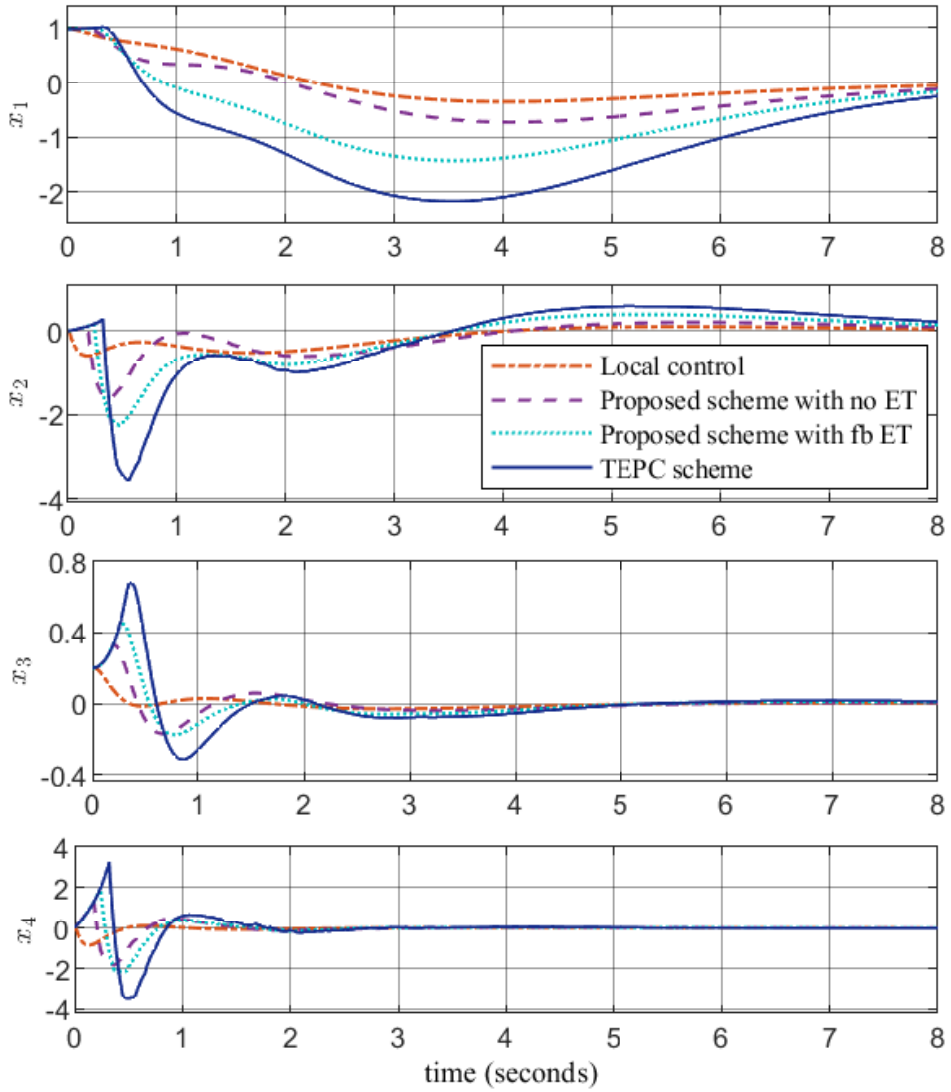

Figure 2.16: Comparison of state values of inverted pendulum system without network (Local control), with network and no ET on both the paths, with network and ET on only feedback path and with no ET on both the paths and with network and TEPC scheme that uses ET on both feedback and forward paths

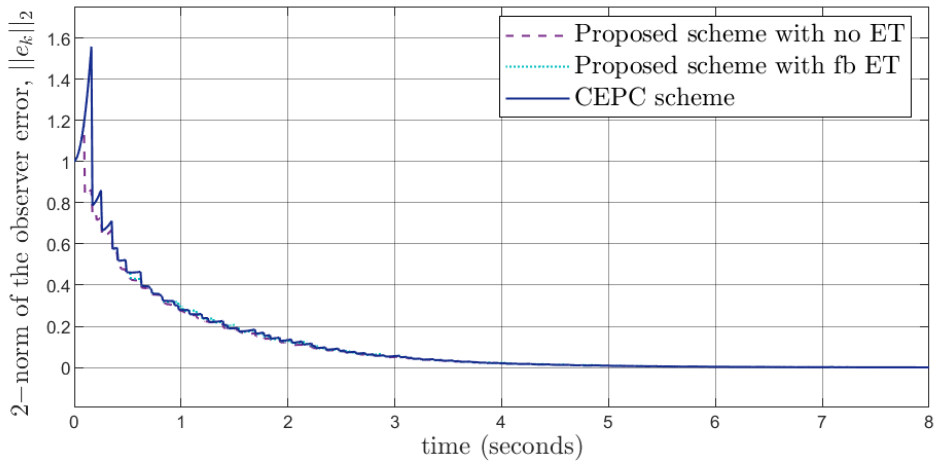

Figure 2.17: Comparison of 2−norm of the observer error with no ET on both paths, with ET on only feedback path and with TEPC scheme

The ETPC scheme utilizes ET to transmit the output information to the observer, which is placed on the plant side of the communication network. The communication network is considered to have random delays. The proposed TEPC scheme with only a feedback network has used an observer on the controller side of the communication network, considered the network unreliable with both delays and dropouts and also utilized ET before the packet transmission through the network to save the channel bandwidth, which in turn causes active dropouts.

Even though both the works involve ET and network delays and use the same controller gain value, due to the placement of the observer on the controller side and the difference in observer design, the proposed work performs better (in the sense that the response is closer to local control one) than the work in the ETPC scheme even though packet dropouts involved and comparatively large values of delays and dropouts occur.

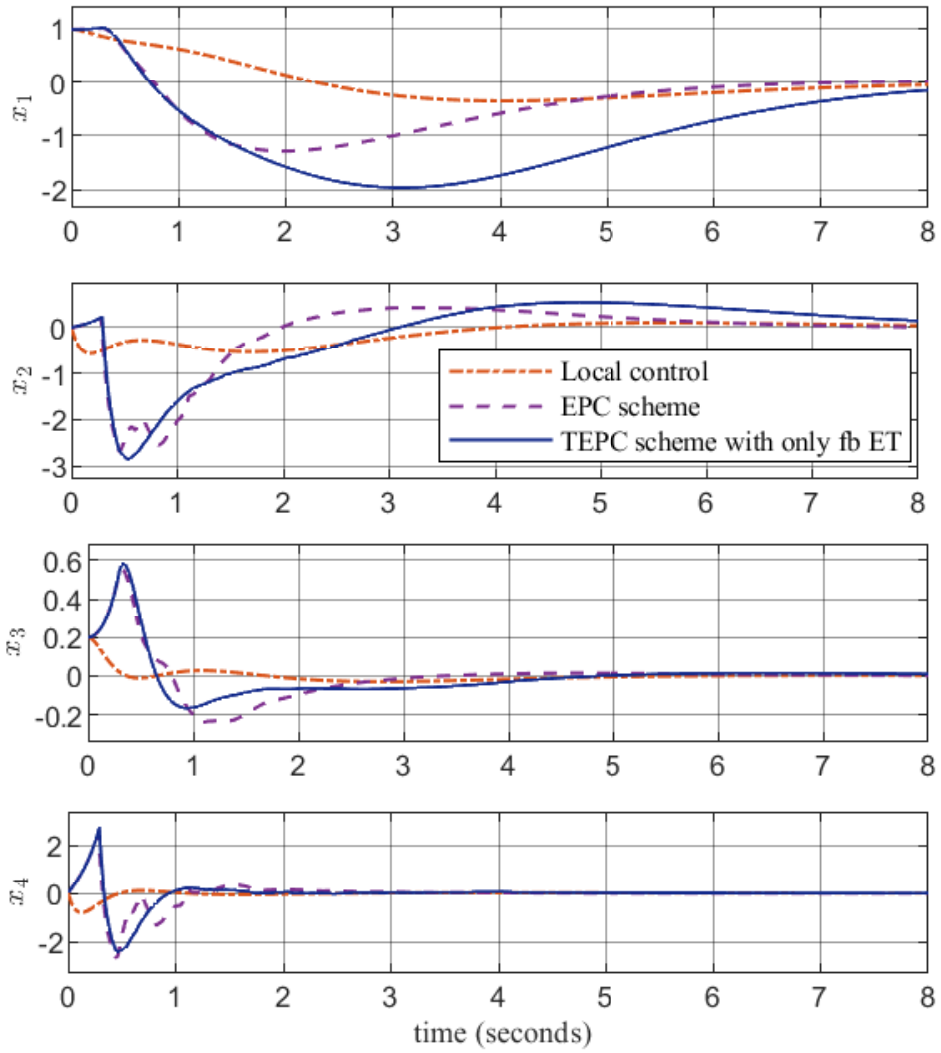

Figure 2.18: Comparison of inverted pendulum system state values without network (Local control), with EPC scheme, and with TEPC scheme with only feedback (fb) ET

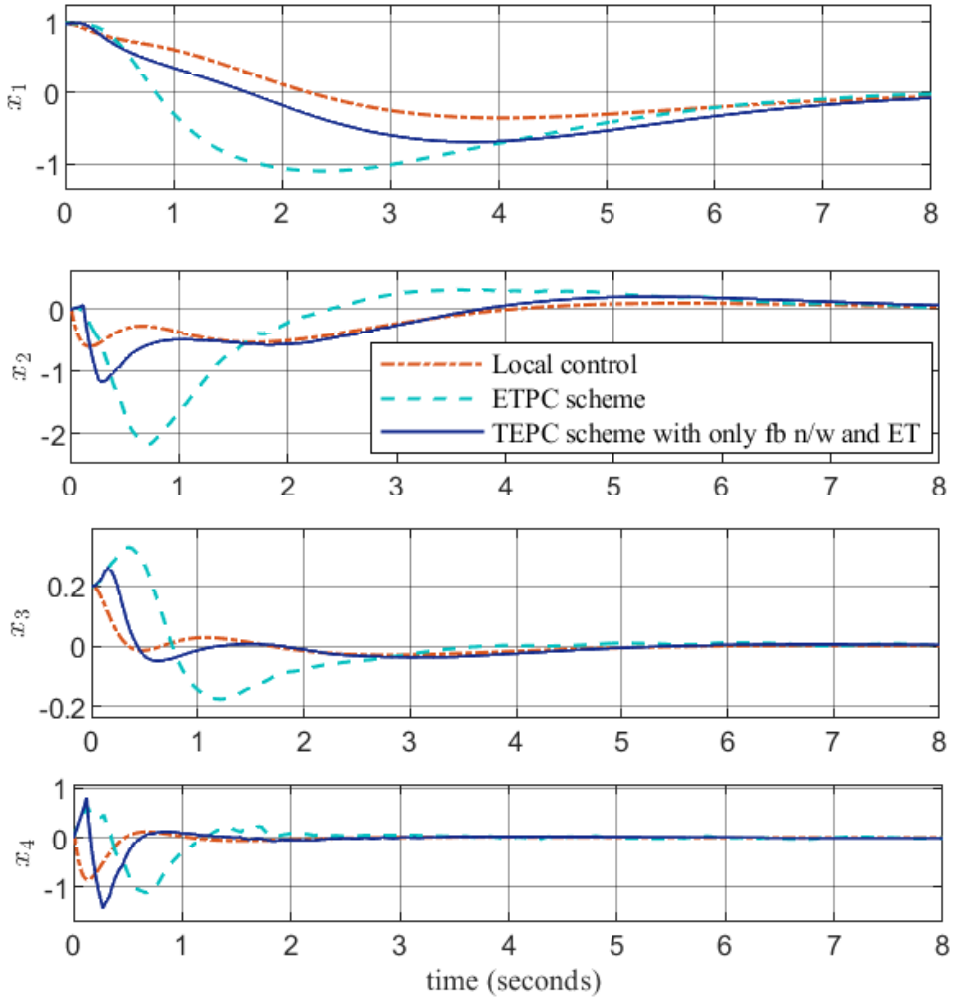

Figure 2.19: Comparison of inverted pendulum system state values without network (Local control), with ETPC scheme, and with TEPC scheme with only feedback (fb) ET

2.6 Summary

In this chapter, a stabilization challenge in a discrete-time NCS characterized by random network delays, dropouts, and limited communication resources is explored. Introducing a TEPC scheme, primarily featuring event-triggered state estimation via a remote predictive controller, the performance of the closed-loop NCS is demonstrated. Comparing the performance of the proposed NCS setup, employing complete ET, with existing approaches that incorporate ET only in the feedback channel reveals that implementing ET in both the feedback and forward channels allows the system to effectively compensate for delays and dropouts, ensuring overall stability. Furthermore, a comparative analysis between the proposed scheme and recent event-triggered predictive control approaches is conducted, highlighting the efficacy of the method in stabilizing the NCS, underscored by the additional advantage of having an observer on the controller side.

The challenge of constructing the state information using delayed output data without access to the control input has been addressed in this chapter, and the system performance can still be enhanced by transmitting more output information without increasing the number of transmissions. Building on these findings, the next chapter proposes a novel approach to improve the performance of predictive controllers in NCSs by mitigating network-induced delays and packet dropouts through sequential output transmission.