Chapter 4

Predictive-triggered compensating control for networked control systems with random delays and dropouts

4.1 Introduction

NCSs utilize communication networks to exchange information among functional blocks of a closed-loop control system. This leads to their rapid development along with flexibility, scalability, cost-effectiveness, and distributed control suitable for various industrial applications, e.g., manufacturing, transportation, robotics, and process control [7, 8, 77, 133]. However, NCSs involve random time delays, packet losses and disorder, time-varying transmission intervals, and competition of multiple nodes accessing networks that causes control performance degradation [87–89].

Model-based delay-compensating control through prediction is considered one of the main approaches for improving NCS performance against random network delays and packet losses [104, 107, 108]. On the other hand, event-triggering (ET) is used to overcome the challenges posed by limited bandwidth availability of the network [47, 53, 134]. ET reduces the instants of network communication by determining the usability of new information by comparing the present information with the past. To address network-induced delays and dropouts and the limited communication resource problem, NCSs use

ET integrated along with delay-compensating control through prediction [11, 12, 54, 55]. In this regard, only a few studies have utilized ET in both the forward and feedback paths [42, 56, 57].

In comparison to ET, predictive triggering (PT) offers to trigger (network communication) through prediction (looking at future state evolution), which makes the triggering more effective, trading off the computational cost involved in the prediction mechanism [66]. It also allows the operator to be notified of available channels in advance, facilitating better network scheduling, particularly for shared networks. There are variants of predictive-triggering, such as Self-Triggering (ST) and Periodic Event-Triggering (PET) [71,72] that are special cases of generic PT mechanism by restricting or periodically scheduling prediction parameters [73, 74].

As far as reduced network communication is concerned, an NCS configuration is desired to have the following features: (i) Shared network in both the forward and feedback channels, (ii) Reduced network communication through ET, PT, or their variants, and (iii) Reduced packet size. In this work, all the above considerations are made in the NCS configuration. Particularly, a PT involving the predicted states from the plant model and observer is used to trigger communication-based on a predefined threshold. Performance evaluation of both the fixed horizon PT and variable horizon PT are evaluated by considering them in the feedback channel of the NCS. Further, PET is employed in the forward channel of the NCS to allocate channels at fixed and predetermined intervals. In addition, transmitting output rather than state information is incorporated to reduce packet size. To address the dropped packets caused by triggering and network-induced delays and drops, state prediction and control input generation are incorporated at the remote end. Due to this, the proposed scheme will be referred to as Predictive Triggered Predictive Control (PTPC) in this chapter. The proposed PTPC approach is applied to both state and output information transmission for constructing a comparison.

4.2 System description

This study examines an NCS with two-way PT communication between the plant and the controller, as shown in Fig. 4.1. The information transmitted through the communication network is considered time-stamped, assuming all blocks utilizing the time stamps

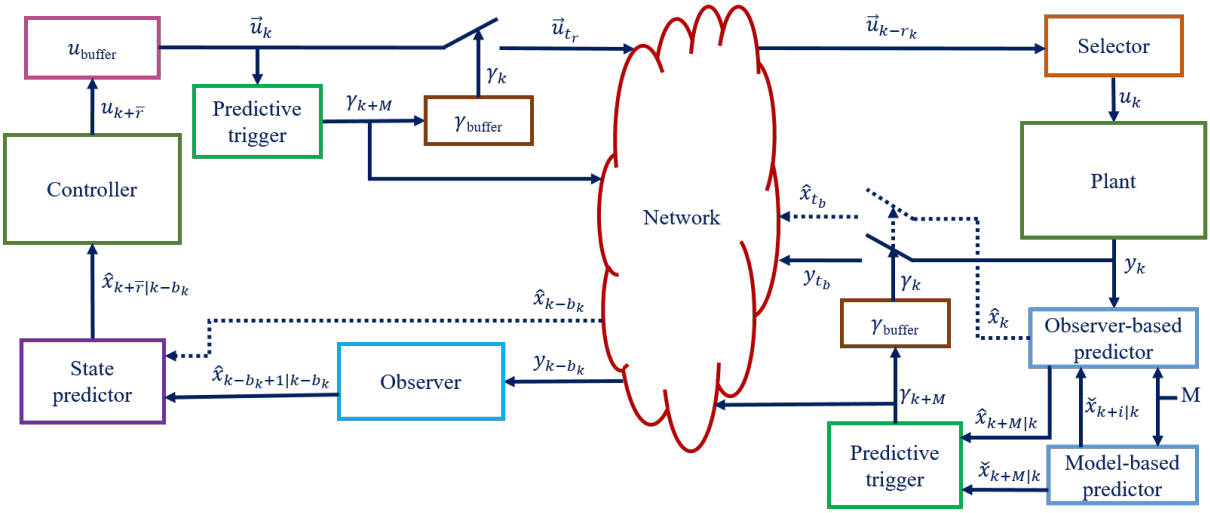

Figure 4.1: NCS configuration with PT in both the forward and feedback channels incorporating transmission of input and output information. Transmission of the predicted state instead of the output is also shown using dotted lines.

are synchronized. Both the feedback and forward channels use PT communication. The transmitted packets of the plant information in the feedback path and the control information from the controller in the forward path experience random network delays and dropouts. The network-induced dropouts are called passive dropouts, while dropouts caused by implementing triggering are called active dropouts. In Fig. 4.1, an observer at the output side of the plant yields the state estimate. Either of the two transmission situations is considered: transmitting (i) the output or (ii) the state. In the case of output transmission, an observer on the controller side reconstructs states from the delayed output and feeds them to a state predictor. In the case of state transmission, a predictor receives the delayed packets and predicts lost states due to network delays and dropouts. The predicted control inputs are stored in a buffer, and an array of advanced control inputs is generated to communicate with the plant side. A selector on the plant side chooses the appropriate control input based on the forward channel delay.

Table 4.1 presents the notations used in this work. Fig. 4.2 depicts the analysis of data packet transmission from the transmitter (PT) on the plant side of the NCS to the receiver (controller-side observer) through the communication network. The feedback PT triggers at a specific instant denoted as $t_b, b \in \mathcal{N}$, enabling the transmission of the output packet y_{t_b} through the network. Whenever a packet dropout occurs, whether active or passive, the receiver utilizes the latest received packet, resulting in an incremented delay

value. Thus the random delay in the feedback path, represented by b_k, follows the relation $b_{k+1} \leq b_k+1$. Synchronization and the time stamps t_b associated with each received packet allow the computation of the delay value b_k at the receiver side by subtracting the packet's timestamp from the instantaneous value of k, i.e., $b_k = k - t_b$. The maximum total delay in the feedback path, denoted as $\bar{b} = D_b + P_{b_a} + P_{b_p}$, is determined by the upper bound of delay in the feedback network (D_b) and the upper bounds of consecutive active and passive dropouts, P_{b_a} and P_{b_p}, respectively. Similarly, for the forward path, the maximum total delay is expressed as $\bar{r} = D_r + P_{r_a} + P_{r_p}$.

4.2.1 Plant model

The following discrete-time linear system describes the plant in the predictive-triggered predictive controlled NCS.

$$x_{k+1} = Ax_k + Bu_k, \qquad y_k = Cx_k. \tag{4.1}$$

where $x_k \in \mathcal{R}^n$ is the state of the plant, $u_k \in \mathcal{R}^m$ is the control signal applied to the plant and $y_k \in \mathcal{R}^p$ is the plant's output. A, B and C are constant matrices of appropriate dimensions. The pair (A, B) is assumed to be controllable, and the pair (A, C) is assumed to be observable for the plant (4.1).

4.2.2 State predictions

Two methods are used for state predictions — (i) model-based and (ii) observer-based (denoted as $\check{x}_k$ and $\hat{x}_k$, respectively). A state feedback controller with gain K is considered for control implementation as:

$$u_k = Kx_k \tag{4.2}$$

For a prediction horizon of M, the model-based state predictions can be obtained by solving the system model (4.1) subjected to the state feedback (4.2) with $u_k = K\check{x}_k$ as:

$$\check{x}_{k+M|k} = \bar{A}^M \check{x}_k \tag{4.3}$$

where $\bar{A} = A + BK$.

For observer-based control, the control input is realized through $u_k = K\hat{x}_k$. A Luenberger observer [28] with observer gain L is considered for implementation as:

$$\hat{x}_{k+1|k} = A\hat{x}_k + BK\hat{x}_k + L(y_k - C\hat{x}_k) \tag{4.4}$$

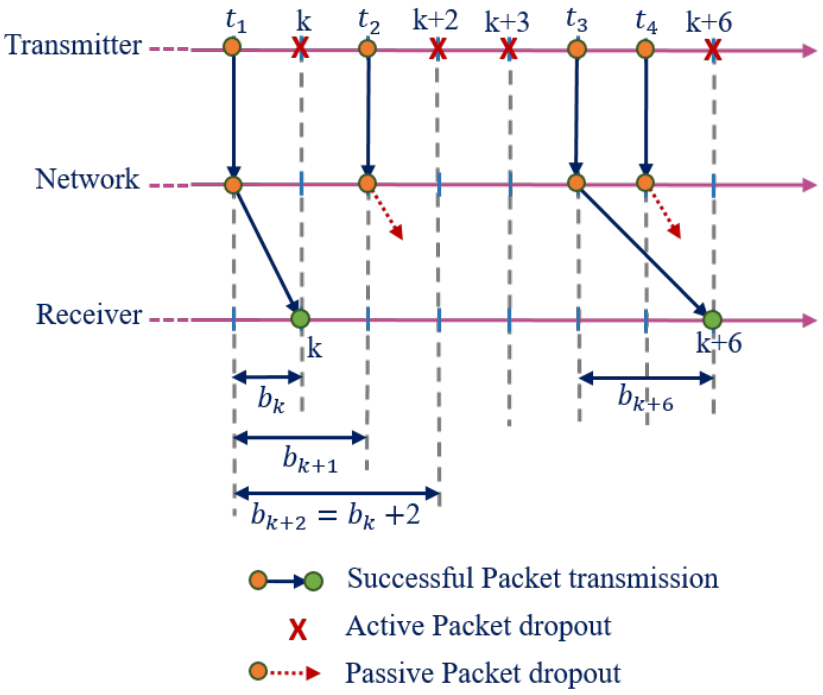

Figure 4.2: Illustration of packet transmission with the effects of event-triggering at the transmitter and the communication network in between transmitter and receiver

Using an approximation that $y_k = Cx_k \approx C\check{x}_k$, the above can be written as

$$\hat{x}_{k+1|k} = \bar{A}\hat{x}_k + LCe_k \tag{4.5}$$

where $e_k = \check{x}_k - \hat{x}_k$ is the prediction error w.r.t. observed state. Corresponding observer-based M^{th} predicted state is

$$\hat{x}_{k+M|k} = \bar{A}^M\hat{x}_k + \sum_{m=1}^{M} \bar{A}^{m-1}LCe_{k+M-m} \tag{4.6}$$

where $e_{k+M-m} = \check{x}_{k+M-m|k} - \hat{x}_{k+M-m|k}$

Remark 7. *The prediction error $\check{e}_k = x_k - \check{x}_k$ is inherently present in the prediction error w.r.t. the observed state $e_k = \check{x}_k - \hat{x}_k$ because of the approximation $x_k \approx \check{x}_k$ in (4.5). Due to this, the error e_k might be larger than that of $\check{e}_k$ and may lead to more frequent triggering. However, the main problem of predicting future events to utilize the bandwidth resources is fulfilled, and the system's overall performance is maintained.*

4.2.3 Predictive-triggering at feedback channel

Fixed-horizon predictive-triggering

For establishing a triggering law, let us define the prediction error as

$$e_{k+M} = \check{x}_{k+M} - \hat{x}_{k+M}. \tag{4.7}$$

Then the predictive-triggering (PT) law is defined as:

$$\gamma_{k+M} = \begin{cases} 1, & \text{if} \quad E_{k+M|k} \geq C_{k+M} \\ 0, & \text{else} \end{cases} \tag{4.8}$$

where γ_{k+M} indicates the M-step ahead predicted value of γ at k^{th} instant and the predicted estimation cost $E_{k+M|k}$ is given by $E_{k+M} = (e_{k+M})^T(e_{k+M})$. The communication cost is defined by C_{k+M}, which is application-specific and may be related to, for instance, the consumption of bandwidth or energy. Also, the maximum interval of two triggering instants t_b and t_{b+1} is limited to P_{b_a} to avoid error accumulation over a long time. Note that, for $M = 0$, the above-mentioned triggering law is the conventional ET [57].

The predicted γ values, γ_{k+M} are stored in γ_{buffer}. Then the instantaneous γ values are retrieved to operate the switch-like triggering mechanism to transmit the output or state values. The observed state $\hat{x}_k$ or the output y_k shown in Fig. 4.1 is transmitted based on either state transmission or output transmission and can be defined as follows:

$$\hat{x}_{t_b} \quad \text{or} \quad y_{t_b} = \begin{cases} \hat{x}_k \quad \text{or} \quad y_k, & \text{if} \quad \gamma_k = 1 \\ \hat{x}_{k-1} \quad \text{or} \quad y_{k-1}, & \text{if} \quad \gamma_k = 0 \end{cases}$$

Variable horizon predictive triggering

In this scheme, the prediction horizon M is computed at each instant based on an error condition given as:

$$\text{Find } M \text{ for which } E_{k+M|k} \geq C_{k+M} \text{ and enforce all } \gamma_k, \cdots, \gamma_{k+M-1} = 0 \text{ and } \gamma_{k+M} = 1. \tag{4.9}$$

It may be noted that this triggering scheme is similar to the self-triggering one in [66].

Like the fixed horizon PT technique, the obtained γ_{k+M} values are stored in the γ_{buffer}, and the instantaneous γ values are accessed based on the variable M. Based on the instantaneous values of γ_k, the packets containing either state values or output values are transmitted through the network.

4.2.4 Observer on the controller side

In the case of output transmission, the transmitted packets are received by the observer block. The observer block on the controller side differs from that of the observer on the plant side. It is composed of an observer buffer and an observer. When a new packet is received, the observer buffer checks its timestamp against the existing one and keeps only the most recent one. When a packet is lost, the previous data is utilized, and the observer's output information can be expressed as:

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

Here, $k - b_k$ and $k - 1 - b_{k-1}$ represent the time stamps of the currently received packet and the previously stored packet, respectively, and $\max\{t_b\}$ denotes the most recent time stamp. Fig. 4.3 illustrates the operation of the observer when y_{k-b_k} is provided as input. The observer generates a one-step-ahead state estimate given by:

$$\hat{x}_{k-b_k+1|k-b_k} = A\hat{x}_{k-b_k|t_{b_l}} + Bu_{k-b_k} + L(y_{k-b_k} - C\hat{x}_{k-b_k|t_{b_l}}) \tag{4.10}$$

Here, L denotes the observer gain matrix to be designed, and $\hat{x}_{k-b_k|t_{b_l}}$ represents the observed state value at instant $(k-1)$, given the most recent received packet with a timestamp of t_{b_l}. The values of $\hat{x}_{k-b_k|t_{b_l}}$ and u_{k-b_k} are extracted from x_{buffer} and u_{buffer}, respectively, based on the delay value b_k. The buffers x_{buffer} and u_{buffer} are used to store the predicted state and control values, and the sizes of these buffers are determined according to the maximum delay values.

4.2.5 Predictor and controller

After the observed state of delayed output is obtained, the delay is compensated using a state predictor with model-based predictions. The number of prediction steps the predictor executes equals the sum of the instantaneous feedback delay b_k and the maximum forward path delay $\bar{r}$. The predictor can be represented as:

$$\hat{x}_{k-b_k+n|k-b_k} = A\hat{x}_{k-b_k+n-1|k-b_k} + Bu_{k-b_k+n-1}, \quad \text{for } n = 2, 3, \cdots, b_k + \bar{r} \tag{4.11}$$

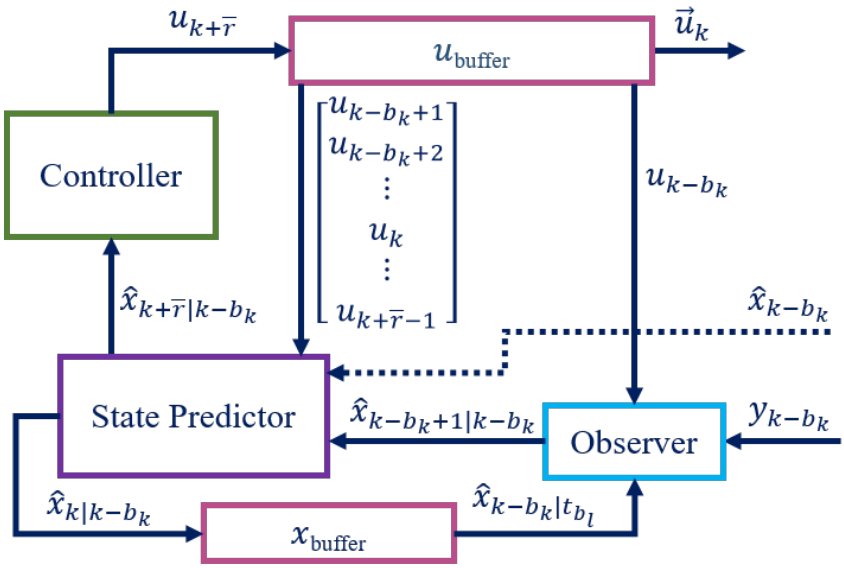

Figure 4.3: The block components at the remote side of the PTPC scheme for output and state transmission

To generate these predictions, the control inputs i.e. $u_{k-b_k+1}, \cdots, u_k, \cdots, u_{k+\bar{r}-1}$ are retrieved from the u_{buffer}, as depicted in Fig. 4.3. Combining (4.10) and (4.11) one obtains

$$\hat{x}_{k+\bar{r}|k-b_k} = A^{b_k+\bar{r}}\hat{x}_{k-b_k|t_{b_l}} + \sum_{m=1}^{b_k+\bar{r}} A^{m-1}Bu_{k+\bar{r}-m} \quad + A^{b_k+\bar{r}-1}L(y_{k-b_k} - C\hat{x}_{k-b_k|t_{b_l}}) \quad (4.12)$$

In the case of state transmissions, the state predictor receives the packets, and the delayed packet containing the state information is compared with the previous latest packet to update the current latest packet.

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

Then the state predictor utilizes this latest packet and makes predictions based on the delay undergone to compensate for the forward path delays and drops. In this case, the predictions are made as given below:

$$\hat{x}_{k-b_k+n|k-b_k} = A\hat{x}_{k-b_k+n-1|t_{b_l}} + Bu_{k-b_k+n-1}, \quad \text{for } n = 1, 2, \cdots, b_k + \bar{r} \quad (4.13)$$

and it follows that:

$$\hat{x}_{k+\bar{r}|k-b_k} = A^{b_k+\bar{r}}\hat{x}_{k-b_k|t_{b_l}} + \sum_{m=1}^{b_k+\bar{r}} A^{m-1}Bu_{k+\bar{r}-m}$$

The control inputs $u_{k-b_k}, u_{k-b_k+1}, \cdots, u_k, \cdots, u_{k+\bar{r}-1}$ involved in these predictions are obtained from the u_{buffer}.

Subsequently, the predicted state information is utilized to generate the predicted control input $u_{k+\bar{r}|k-b_k}$, which is incorporated through a state feedback control law (4.2). By defining d_k as $\bar{r} + b_{k-\bar{r}}$, the expression $u_{k+\bar{r}} = K\hat{x}_{k+\bar{r}|k-b_k}$ can be rewritten as:

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

The control input $u_{k+\bar{r}}$ is then added to the u_{buffer} containing $\vec{u}_k = [u_k^T \; u_{k+1}^T \; \cdots \; u_{k+\bar{r}}^T]^T$. In this process, the oldest value u_{k-1} from the array generated in 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. Consequently, the length of the vector $\vec{u}$ is always $(\bar{r} + 1)$. This prediction technique aids in compensating for the forward path delay.

4.2.6 Predictive-triggering at the forward channel and the selector

The triggering in the forward channel slightly differs from that in the feedback channel. Here, the triggering is scheduled periodically with a period N, $2 \leq N \leq P_{r_a}$ and is added to the maximum delay and dropout count value. Therefore, the maximum delay value in the forward path equals $\bar{r} = D_r + N + P_{r_p}$. The value of N can be chosen based on the changing demands of the network resources. Since the triggering decision is made after every constant N number of time instants, it is a class of periodic event triggering as follows:

$$\vec{u}_{t_r} = \vec{u}_{k+nN}, \forall \gamma_k = \gamma_{k+nN}, n \in \mathcal{N}, 2 \leq N \leq P_{r_a} \tag{4.14}$$

The selector block consists of both a selector and a selector buffer. Upon receiving new packets, the selector buffer only retains the one with the latest timestamp.

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

From the packets generated by u_{buffer} at each time-step, it is evident that the predicted control input of the controller, u_k, is distributed over $(\bar{r} + 1)$ consecutive packets. The selection logic selects the suitable value u_k for the current time k, considering the instantaneous forward path delay r_k.

4.3 Predictive control scheme

This section introduces the PTPC scheme employed at the remote terminal. The control input at an instant is given by $u_k = K\hat{x}_{k|k-d_k}$, and it leads to the closed-loop system dynamics described by:

$$x_{k+1} = (A + BK)x_k - BK(x_k - \hat{x}_{k|k-d_k}) \tag{4.15}$$

To facilitate the analysis, we define $\alpha_k = x_k - \hat{x}_{k|k-d_k}$, and

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

which allows the derivation of the expression

$$\alpha_k = A^{d_k-1}(x_{k-d_k+1} - \hat{x}_{k-d_k+1|k-d_k}) \tag{4.16}$$

through successive replacements. Similarly, by defining $\beta_k = x_{k-d_k+1} - \hat{x}_{k-d_k+1|k-d_k}$, it can be obtained as

$$\beta_k = Ax_{k-d_k} + Bu_{k-d_k} - A\hat{x}_{k-d_k|t_{b_l}} + Bu_{k-d_k} - L(Cx_{k-d_k} - C\hat{x}_{k-d_k|t_{b_l}}) \tag{4.17}$$

By considering t_{b_l} as the timestamp of the most recent packet received at $k - d_k$, we define $e_{k-d_k} = x_{k-d_k} - \hat{x}_{k-d_k|t_{b_l}}$. Consequently, one obtains

$$\beta_k = (A - LC)e_{k-d_k} \tag{4.18}$$

Substituting (4.18) into (4.16), and subsequently inserting (4.16) into (4.15), we derive the following expression:

$$x_{k+1} = (A + BK)x_k - BKA^{d_k-1}(A - LC)e_{k-d_k} \tag{4.19}$$

To study the dynamics of e_{k-d_k}, we introduce the variable $s_k = b_{k-\bar{r}} - b_{k-\bar{r}-1}$. It is worth noting that s_k takes values in the range $\{-\bar{b}, \cdots, 1\}$, and its relationship with e_k is explored in two cases.

In Case-I, when $s_k = 1$, it signifies a dropout, resulting in the increment of $b_{k+1-\bar{r}}$ by one compared to $b_{k-\bar{r}}$. As a consequence, based on the relationship $d_k = \bar{r} + b_{k-\bar{r}}$, it

can be expressed as:

$$\begin{aligned} e_{k+1-d_{k+1}} &= x_{k+1-d_{k+1}} - \hat{x}_{k+1-d_{k+1}|k-d_k} \\ &= x_{k+1-\bar{r}-b_{k-\bar{r}+1}} - \hat{x}_{k+1-\bar{r}-b_{k-\bar{r}+1}|k-d_k} \\ &= x_{k-\bar{r}-b_{k-\bar{r}}} - \hat{x}_{k-\bar{r}-b_{k-\bar{r}}|k-d_k} \\ &= x_{k-d_k} - \hat{x}_{k-d_k|k-d_k} \\ &= e_{k-d_k} \end{aligned} \tag{4.20}$$

In Case-II, when $s_k \in \{-\bar{b}, \cdots, 0\}$, it is evident that $b_{k-\bar{r}+1}$ can be obtained by adding s_k to $b_{k-\bar{r}}$, and $\hat{x}_{k+1-\bar{r}-b_{k-\bar{r}+1}} = \hat{x}_{k+1-\bar{r}-b_{k-\bar{r}}-s_k|k-d_k}$. Therefore,

$$\begin{aligned} e_{k+1-d_{k+1}} &= x_{k+1-d_{k+1}} - \hat{x}_{k+1-d_{k+1}|k-d_k} \\ &= x_{k+1-d_k-s_k} - \hat{x}_{k+1-d_k-s_k|k-d_k} \\ &= A(x_{k-d_k-s_k} - \hat{x}_{k-d_k-s_k|k-d_k}) \\ &\vdots \\ &= A^{-s_k}(x_{k+1-d_k} - \hat{x}_{k+1-d_k|k-d_k}) \\ &= A^{-s_k}(A - LC)e_{k-d_k} \end{aligned} \tag{4.21}$$

To capture the overall behavior, we define the augmented state vector as $z_k = \begin{bmatrix} x_k^T & e_{k-d_k}^T \end{bmatrix}^T$ and the augmented dynamics for the above two cases can be derived as:

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

where

$$\Phi = \begin{cases} \begin{bmatrix} A+BK & -BKA^{d_k-1}(A-LC) \\ 0 & I \end{bmatrix}, \\ \text{when } s_k = 1 \end{cases} \tag{4.22a}$$

$$\Phi = \begin{cases} \begin{bmatrix} A+BK & -BKA^{d_k-1}(A-LC) \\ 0 & A^{-s_k}(A-LC) \end{bmatrix}, \\ \text{when } s_k \in \{-\bar{b}, \cdots, 0\} \end{cases} \tag{4.22b}$$

The stability of the closed-loop NCS can be analyzed by examining the augmented system dynamics represented by (4.22).

4.4 Controller and observer design

This section focuses on the dynamics described by (4.22) and introduces a design approach for the gains L and K to ensure the stability of the closed-loop NCS. It should be noted that Case-I, for s_k, can occur at most $\bar{b}$ consecutive times. Once the maximum dropout count is reached, the system dynamics switch to Case-II. Consequently, Case-I cannot occur continuously. To establish the stability condition for (4.22), the following well-known result, derived using the separation principle [125], is considered:

Theorem 5. *The augmented system* (4.22b) *is asymptotically stable if $A + BK$ is Schur and the switched system matrix with the following form is stable:*

$$A^{-s_k}(A - LC), \quad s_k \in \{-\bar{b}, \cdots, 0\} \tag{4.23}$$

Proof. The proof is straightforward due to the block-diagonal nature of the system matrix. □

The above result demonstrates the separation between the controller and observer designs, allowing them to be carried out independently.

For the controller design, the LQR controller [125] is employed in this work, although other state feedback controller design methods that ensure the stability of $(A + BK)$ can also be utilized. Several simulation studies have been conducted to confirm the effectiveness of the LQR controller for the given NCS, as demonstrated in the subsequent section through case studies.

To ensure the stability of $A^{-s_k}(A - LC)$ with uncertain s_k, a common Lyapunov function [126] is utilized. This leads to the construction of the following result, which is later used for the observer design:

Theorem 6. *The NCS with the augmented system dynamics (4.22b) is asymptotically stable if $A + BK$ is Schur and there exists a positive definite matrix P that satisfies the following inequality:*

$$[A^{-s_k}(A - LC)]^T P[A^{-s_k}(A - LC)] - P < 0, \quad s_k \in \{-\bar{b}, \cdots, 0\} \tag{4.24}$$

Proof. The proof follows the standard Lyapunov function method [126] for linear systems and is omitted for brevity. □

For the design of L, it is important to note that (4.24) represents a nonlinear matrix inequality. By employing the Schur complement, it can be rewritten as:

$$\begin{bmatrix} -P & * \\ A^{-s_k}(A-LC) & -P^{-1} \end{bmatrix} < 0 \tag{4.25}$$

Here, the symbol '*' represents the symmetric term of the corresponding diagonal element. However, (4.25) remains nonlinear due to the involvement of P and P^{-1}. The cone-complementarity algorithm [127,128] is employed to solve (4.25) for $s_k \in \{-\bar{b}, \cdots, 0\}$. The subsequent algorithm outlines the procedure for designing the observer.

Let $X = P^{-1}$, (4.25) can be rewritten as:

$$\begin{bmatrix} -P & * \\ A^{-s_k}(A-LC) & -X \end{bmatrix} < 0 \tag{4.26}$$

Since $PX = I$, it can be expressed as:

$$\begin{bmatrix} P & * \\ I & X \end{bmatrix} \geq 0 \tag{4.27}$$

It is important to note that (4.27) should be rank-deficient for an exact solution. The cone-complementarity algorithm can now be employed as Algorithm 1 to solve this problem.

Algorithm 3

Step 1: Initialize the iteration index j to 0 and find a feasible set of $(P_0, X_0, L_0) = (P, X, L)|_{j=0}$ that satisfies (4.26) and (4.27).

Step 2: Optimize the following linear matrix inequality (LMI) problem:

$$\min_{P,X,L} \quad \text{trace}(XP_j + PX_j)$$
$$\text{subject to} \quad (4.26) \quad \text{and} \quad (4.27).$$

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

Step 3: Iteratively solve (4.18) for L with $P = P_j$. If feasible, choose L as the observer gain and stop; otherwise, proceed to Step 2.

4.5 Simulation results

In this section, an inverted pendulum [11] subjected to networked control is considered to illustrate the stability, control, and resource conservation using triggering techniques and the proposed PTPC method. The pendulum system dynamics is as in (4.1) with

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

with the sampling period $T = 0.01$s and with the state vector $x_k = \begin{bmatrix} s & \dot{s} & \theta & \dot{\theta} \end{bmatrix}^T$. The initial state values of the plant and observer are taken 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. In both the feedback and forward networks, the random delays and successive dropouts are taken as $[0, 10]$ and $[0, 4]$, respectively. The maximum value of consecutive active dropout count in the feedback PT is $P_{b_a} = 8$, and for PET in the forward path, the period $N = P_{r_a} = 8$ is considered. Consequently, the maximum allowable delays for the feedback and forward paths are set to $\bar{b} = \bar{r} = 22$. For the controller and observer, the following gain matrices are obtained:

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

Trajectories of the state $x_k(1)$ are shown in Fig. 4.4 for the three cases of ET, PT with constant M, and PT with variable M using state transmissions with C_k set to 0.5. They are compared to the ideal case where data transmission is done directly without the network (wonw). The triggering instants of the respective schemes are shown in Fig. 4.5. When the output transmission is considered in place of state transmission, then a similar set of results are obtained, as shown in Fig. 4.8 and Fig. 4.9. Hence, only the output transmission can also be effective.

Fig. 4.6 illustrates the scenario of feedback PT, demonstrating a constant prediction value of $M = 4$ for state transmissions. It displays the predicted triggering instants, γ_{k+M}, as well as the corresponding instantaneous triggering instants, γ_k, which are obtained by

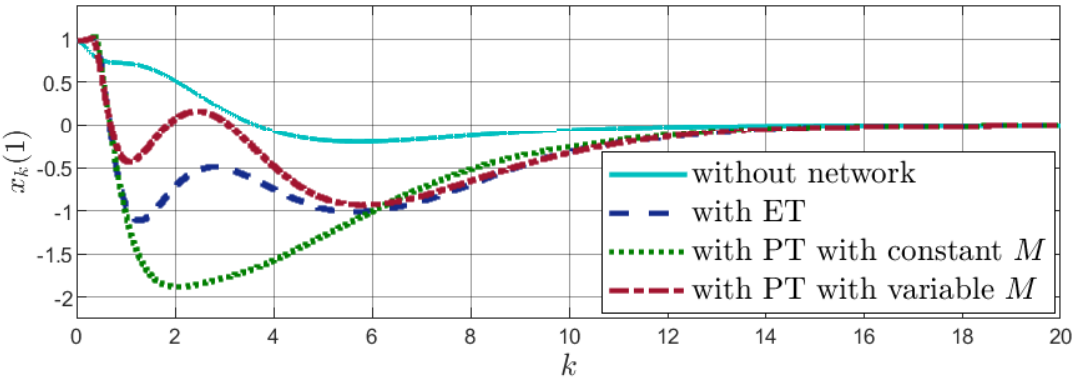

Figure 4.4: The comparison of $x_k(1)$ state trajectory of the inverted pendulum system without network, with ET, and with PT, utilizing predicted state transmission

delaying the predictions γ_{k+M} by M instants. This information enables the network provider to make informed decisions regarding channel allocation, with the ability to plan four steps ahead.

Fig. 4.7 illustrates the variable prediction horizon for state transmissions in the Predictive Triggering (PT) mechanism. The Y-axis represents the prediction horizon values, M, which denote the number of intervals until the next triggering. Due to the long text, the Y-axis label is not used, rather it is represented as the legend. Each triggering instant with $\gamma_k = 1$ (value 1 represents triggering occurs and value 0 represents no triggering) corresponds to the transmission instant.

For example, a value of $M = 1$ indicates that the next triggering will occur after one interval, represented by $\gamma = 1$. Similarly, $M = 2$ suggests that the next triggering will occur after two instants and $M = 9$ indicates that triggering will happen nine instants later. This mechanism helps in efficiently allocating network resources by informing the network operator about the intervals between triggering instants and optimizing channel usage.

Upon analyzing the two feedback-PT schemes depicted in Fig. 4.6 and Fig. 4.7, it can be acknowledged that both schemes are effective in providing advanced channel requirement information. The scheme with constant M in PT measures the error at every time step, whereas the scheme with variable M measures the error only during triggering or transmission events. However, the selection between these two schemes depends on the characteristics of the networked system, particularly whether it is soft-constrained or hard-constrained with respect to the error signal.

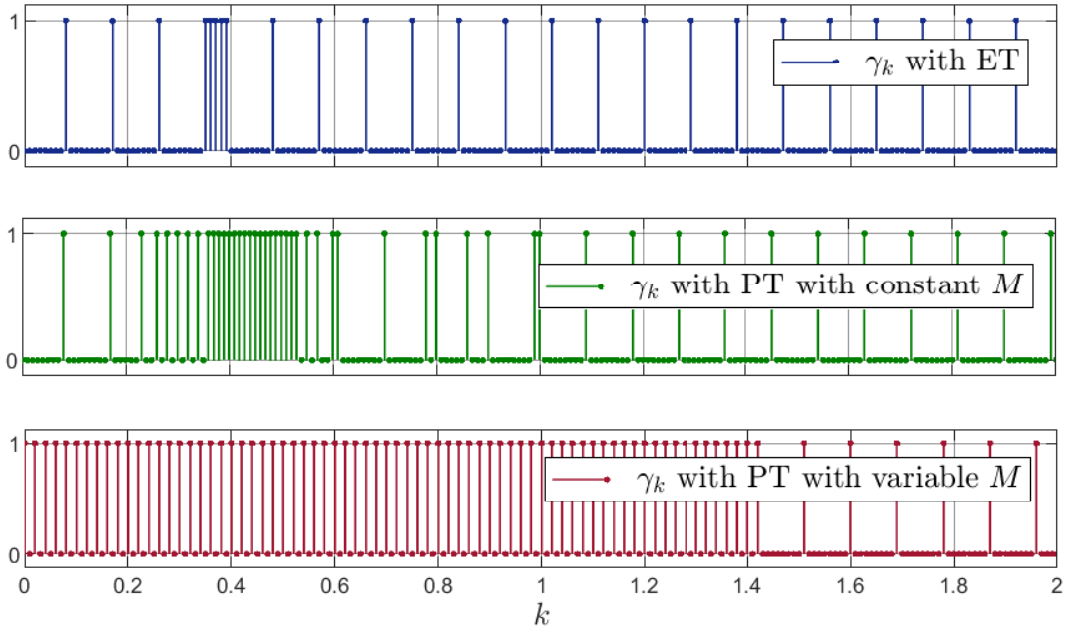

Figure 4.5: Triggering in the feedback path for state transmission with different triggering schemes

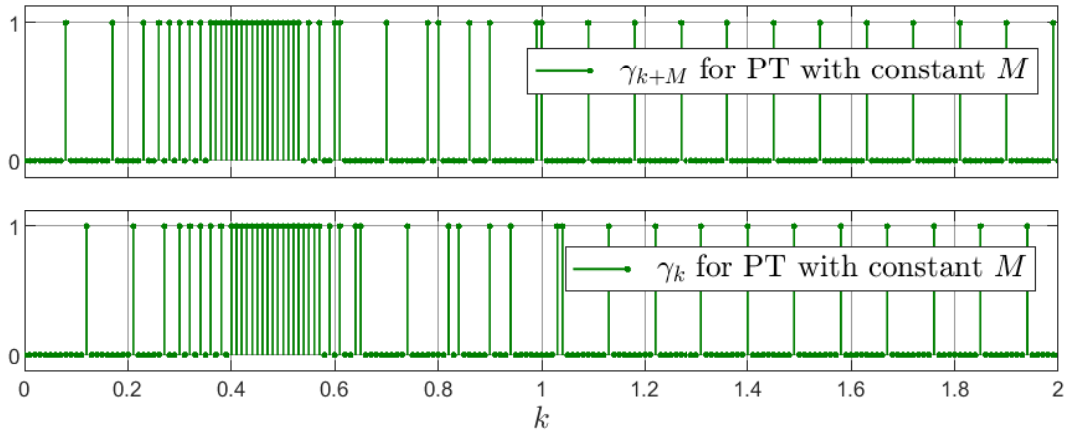

Figure 4.6: γ_{k+M} and γ_k for PT with constant M

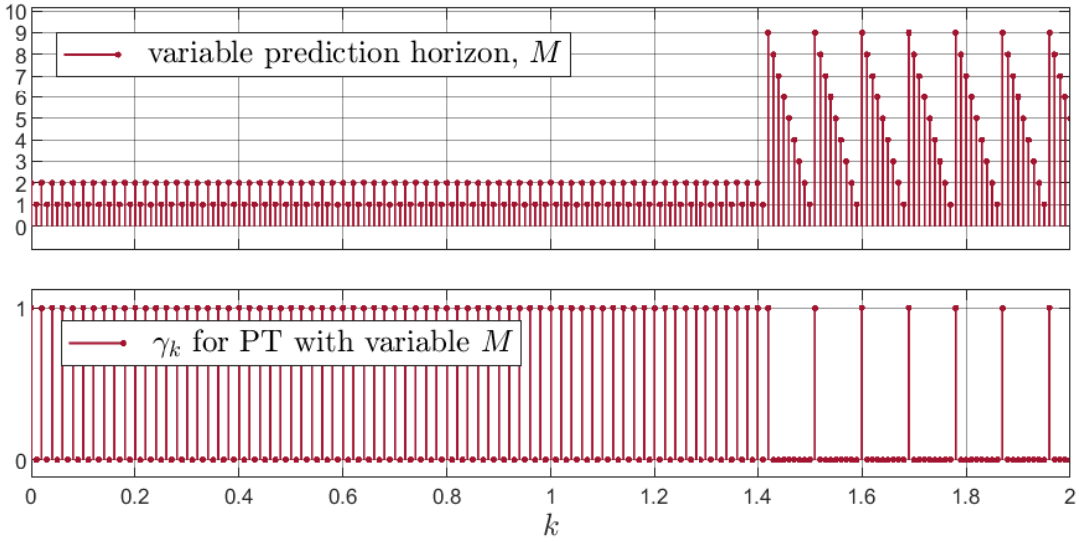

Figure 4.7: Variable prediction horizon, M, and the corresponding triggering instants denoted by γ_k values for PT with variable M

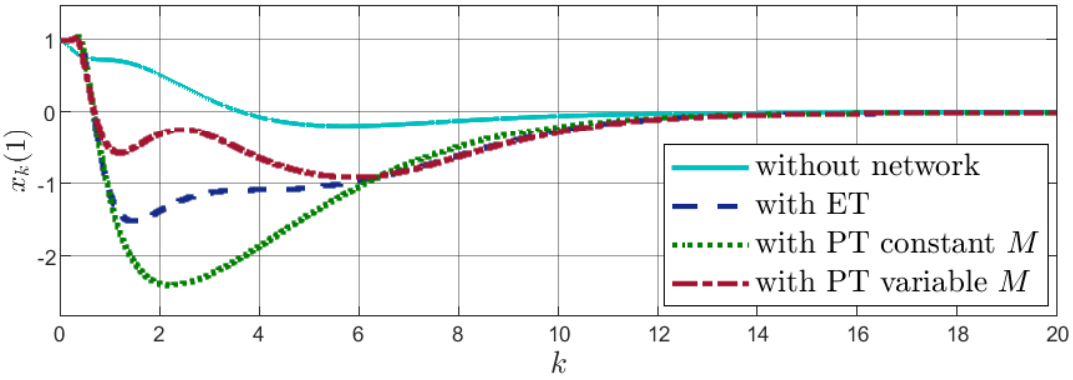

Figure 4.8: The comparison of $x_k(1)$ state trajectory of the inverted pendulum system without network, with ET, and with PT, utilizing output transmission

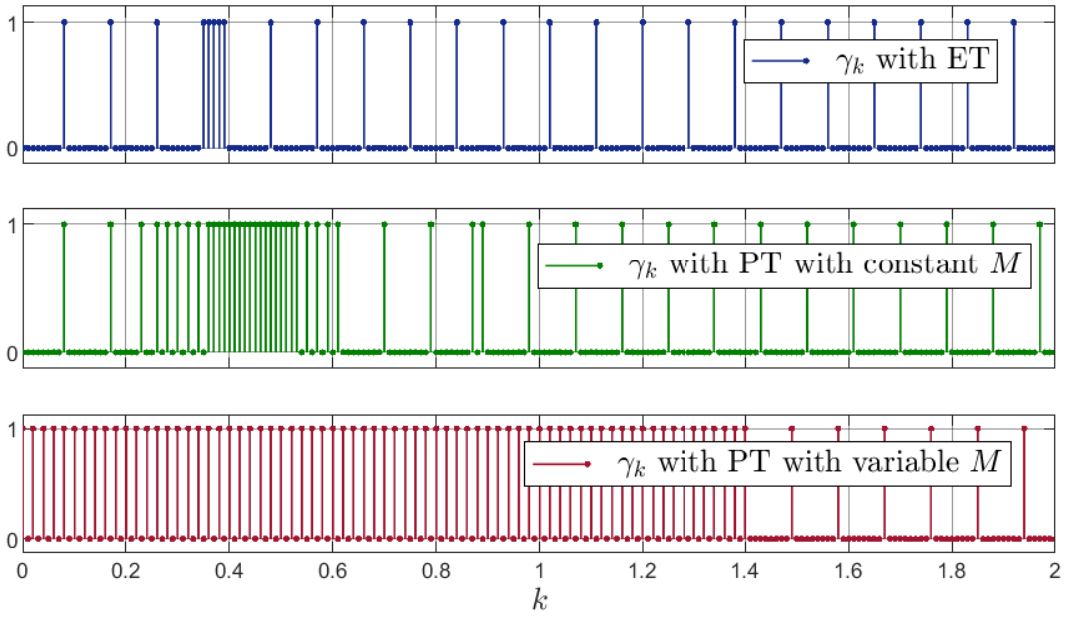

Figure 4.9: Triggering in the feedback path for output transmission with different triggering schemes

In Table 4.2, the drop rates for various communication cost values are presented. The results indicate that the PT with variable M yields more packet transmissions than other methods while performing better than PT with constant M. The table clearly illustrates that as the cost value C_k increases, the drop rate increases for both schemes. Furthermore, it is observed that transmitting output values requires fewer transmissions compared to state transmission. This highlights that the transmission of output values is worth reducing the size of packets, thus reducing the delays and drops in the network while saving the bandwidth resources better than that of state transmission.

The triggering resulting from the PET mechanism in the forward path can be obtained using (4.14), where the period is set to $N = 8$. Fig. 4.10 illustrates how PET facilitates convenient channel allocation through periodic transmissions, occurring every 0.08 s. This periodicity enables efficient utilization of the time slots, allowing the allocation of the remaining slots to other users.

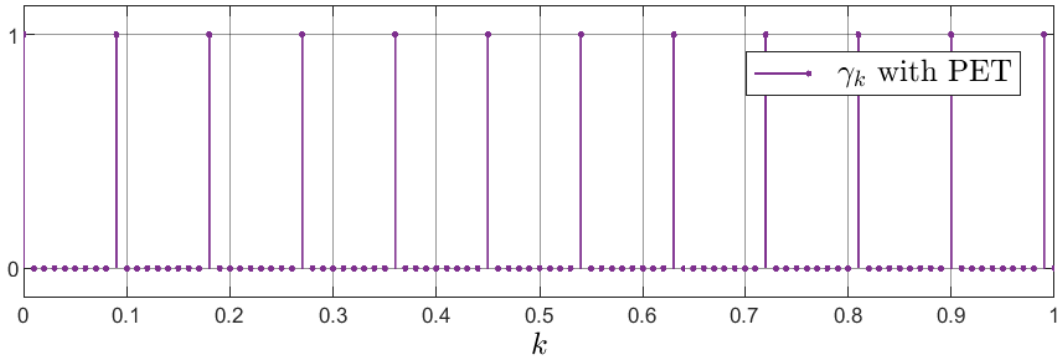

Figure 4.10: Triggering instants with PET in the forward path

Table 4.1: Notations used

k	The present discrete-time-instant
$t_b, b \in \mathcal{N}$	Triggering time-instants in the feedback path
y_{t_b}	Triggered output values
D_b	Maximum amount of random delay due to feedback network
P_{b_a}	Maximum no. of consecutive active dropouts due to triggering in feedback path
P_{b_p}	Maximum no. of consecutive passive dropouts due to feedback network
b_k	Instantaneous random delay in the feedback path, $k - t_b$
$\bar{b}$	Maximum amount of random delay in the feedback path, $D_b + P_{b_a} + P_{b_p}$
$t_r, r \in \mathcal{N}$	Triggering time-instants in the forward path
$\vec{u}_{t_r}$	Triggered control packets
D_r	Maximum amount of random delay due to forward network
P_{r_a}	Maximum no. of consecutive active dropouts due to triggering in forward path
P_{r_p}	Maximum no. of consecutive passive dropouts due to forward network
r_k	Instantaneous random delay in the forward path, $k - t_r$
$\bar{r}$	Maximum amount of random delay in the forward path, $D_r + P_{r_a} + P_{r_p}$
γ_k	Communication decision at k^{th} instant
M	Prediction horizon
C_k	Communication cost
E_k	Estimation cost
$\mathcal{N}$	The set of all natural numbers
$\mathcal{R}^*$	The set of all *-tuples of real numbers

Table 4.2: Packet drop percentage with ET and PT for different transmission and triggering cases in the feedback path

Transmission	Triggering Scheme	$C_k = 0.1$		$C_k = 0.5$		$C_k = 1$	
		No. of Packets	Drop	No. of Packets	Drop	No. of Packets	Drop
Observed State ($\hat{x}_{t_b}$)	ET	120	88.0%	117	88.3%	113	88.7%
	PT (constant M)	183	81.7%	148	85.2%	133	86.7%
	PT (variable M)	500	50.0%	248	75.2%	243	75.7%
Output (y_{t_b})	ET	120	88.0%	116	88.4%	113	88.7%
	PT (constant M)	181	81.9%	145	85.5%	131	86.9%
	PT (variable M)	370	63.0%	237	76.3%	250	75.0%

4.6 Summary

This chapter implements predictive triggering in the feedback and forward paths allowing for efficient communication channel utilization in an NCS. A delay compensation method is implemented through state predictions to compensate for the delays and dropouts at feedback and forward paths. Simulation results show the efficiency of the proposed control method in establishing system stability while fetching benefits in network bandwidth utilization. Moreover, the work also projects the benefit of having an observer on the controller side, which leads to packet transmissions of reduced size trading off higher computational costs at both ends of the network.