A Novel Robust Predictive Control System over Imperfect Networks

Truong. Q. Dinh, Kyoung. K. Ahn, *Member, IEEE* and James Marco

***Abstract*—This paper aims to study on feedback control for a networked system with both uncertain delays, packet dropouts and disturbances. Here, a so-called robust predictive control (RPC) approach is designed as follows: 1- delays and packet dropouts are accurately detected online by a network problem detector (NPD); 2- a so-called PI-based neural network grey model (PINNGM) is developed in a general form for a capable of forecasting accurately in advance the network problems and the effects of disturbances on the system performance; 3- using the PINNGM outputs, a small adaptive buffer (SAB) is optimally generated on the remote side to deal with the large delays and/or packet dropouts and, therefore, simplify the control design; 4- based on the PINNGM and SAB, an adaptive sampling-based integral state feedback controller (ASISFC) is simply constructed to compensate the small delays and disturbances. Thus, the steady-state control performance is achieved with fast response, high adaptability and robustness. Case studies are finally provided to evaluate the effectiveness of the proposed approach.**

***Index Terms*—Networked control system, time delay, state feedback control, predictor, neural network, buffer.**

I. INTRODUCTION

NETWORKED control systems (NCSs) are spatially distributed systems, in which actuators and sensors at the plant side are connected to a controller at the remote side. Due to having many advantages over the traditional control schemes via point-to-point wiring, NCSs have been deploying worldwide in various applications, such as power grids, transportation systems, teleoperation or remote systems.

However for practical uses of NCSs, the most important issues are network-included random time-varying delays and packet dropouts which deteriorate the control performances and, easily cause the instability [1], [5], [25], and [29]. Generally, time delays can be classified into three components: computation delays at the controller and communication delays at the forward and backward channels.

Many studies in literature addressed stability of NCSs with delay problem only [1]-[6]. Here, the controller designs were depended on the assumptions in which the time delay was constant [1], bounded [2]-[5] or had a probability distribution function [6]. Other researchers focused on analyzing time delay problem at communication channels to evaluate its effects on the system performances as well as to compensate these undesirable effects [7] and [8]. For NCSs with large delays, a new control concept based on variable sampling periods using neural network or prediction theories has been recently adopted [9]-[11]. Nevertheless, the observation of real delay data to train and construct the NCSs was not appropriately discussed in these studies. To solve this problem, an advanced variable sampling period control concept has been developed for systems containing random delays [29]. The effectiveness of this concept in accurately detecting and predicting the delays without requiring a training process was proved through real-time experiments.

To adapt NCSs to not only time delays but also packet dropouts, many important methodologies, such as predictive control [12]-[16], adaptive control [17], [18], hybrid control [18], and robust control [19]-[26], were proposed. In an early study with unknown networked systems [18], the randomly missing output feature was considered and modelled as a Bernoulli process and the Kalman filter-based adaptive control scheme was successfully conducted to estimate both the missing output measurements and system parameters. For systems with network problems existing on both the forward and backward channels, Markov chains were employed to model the delays and packet dropouts and, subsequently, incorporated into the well-known two-mode-dependent state feedback controller in the general and practical way [20]. The robust stability of this control method was significantly improved through the implementation of the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms to perform the robust $\mathcal{H}_2$ and robust mixed $\mathcal{H}_2/\mathcal{H}_\infty$ two-mode-dependent controller [21], [22]. Several studies also developed the robust controllers based on the other advanced concepts, in which the network delays are presented in the general form and bounded by both upper and lower limits to reduce the conservatism problem [23]-[25]. Or, the integral action was used to compensate the nonzero disturbance issue [25]. By using these techniques, the NCS performances were remarkably improved over the traditional methods. However in most studies, delays and packet dropouts were assumed to be priorly known and bounded to design the controllers. The complex control designs to deal with both the communication problems and disturbances also restrict their applicability. Another trend to improve networked control performances is known as control based on efficient networks with adaptive

Manuscript received November 29, 2015; revised March 05, 2016; accepted April 15, 2016. This work was supported by the 2017 Research Fund of the University of Ulsan.

T. Q. Dinh and J. Marco are with the Warwick Manufacturing Group (WMG), University of Warwick, Coventry, UK (e-mail: q.dinh@warwick.ac.uk; james.marco@warwick.ac.uk).

K. K. Ahn is with the School of Mechanical Engineering, University of Ulsan, Namgu Muger2dong, Ulsan 680-749, Korea (corresponding author to provide phone: 82-52-259-2282; fax: 82-52-259-1680; e-mail: kkahn@ulsan.ac.kr).

buffers [27] or with a compensation strategy [28]. Although these methods could achieved good control results, the applicability may be limited because of the complex network designs. Furthermore, the uses of fixed sampling period and dynamic buffers with large sizes at both channels could limit the control performances. Additionally, delays due to a complex computation of a control system is another important factor affecting directly on the performance, especially in NCSs [29]. Recently, a robust control approach named RVSPC [36] developed from [29] has been proposed as a feasible solution to address all of these problems. Its controllability has been clearly proven through both simulations and real-time experiments. Although the results are remarkable, the RVSPC still remains some open problems:

- The robust state feedback control module is basically designed for a linear NCS of which the system state is clearly known. It is actually not easy to be achieved in practical applications where most of system states are time-variant. In addition, disturbance cancelation are not considered in this control module design;
- With the use of ZOH for updating instants, the control performance may be degraded or even, unstable when existing long delays and/or high packet loss rate on the communication channels;
- The role of grey model-based prediction is limited to only switch between the control modules and to vary the sampling period.

In order to overcome these drawbacks as well as to improve the control quality, this study aims to develop a novel robust predictor control (RPC) approach based on the advantages of the previous studies, [29] and [36], to deal with networked systems included both of three delay components and packet dropouts. The merit of the RPC can be expressed as bellows:

1- Different from [36], and other studies on network analysis, delays and packet dropouts are quickly and accurately detected online by a network problem detector (NPD) using a new rule to detect network problems.

2- Developed from [29] and [36], a so-called PI-based neural network grey model (PINNGM) is robustly constructed in a general form for a capable of forecasting precisely in advance the network problems and impacts of disturbances on the system response.

3- Using the NPD and PINNGM, a hybrid time-event-driven scheme is derived to avoid long delays during the communication.

4- Different from other networked control techniques normally using ZOH or larger buffers, a small adaptive buffer (SAB), of which the buffer size is online optimized based on the PINNGM, is generated on the plant side to compensate packet dropouts on the forward link.

5- By using the PINNGM and SAB, an adaptive sampling-based integral state feedback controller (ASISFC) is built with four advanced characteristics:

- To compensate packet dropouts on the forward channel, s_k-step-ahead control technique is suggested to combine with the SAB to produce the control input in time;

- To compensate packet dropouts on the backward channel, a state observer is employed;

- To compensate the estimated influences of disturbances, an integral term is added to the state feedback control algorithm;

- To avoid the conservatism as well as to save the computational time in producing the control input online, a set of the control gains is pre-derived for a set of system states with small delay regions and then, stored in a look-up table.

As a result, the steady-state control performance is guaranteed with high adaptability and robustness. The rest of this paper is organized as follows. Section II introduces the concerned problem. The RPC architecture is shown in Section III. The design procedures for the PINNGM and ASISFC are presented in Section IV and Section V, respectively. Experiments are provided and discussed in Section VI. Finally, we conclude the paper in Section VII.

II. Problem Description

A generic system without delays can be described as:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k+1) = Cx(k) \end{cases} \tag{1}$$

where $x(k) \equiv x_k \in R^{mx}, u(k) \equiv u_k \in R^{mu}$ and $y(k) \equiv y_k \in R^{my}$ denote the state vector, control input and system output, respectively; A, B and C are known matrices. Without loss of generality, the reference $r(k)$ is assumed to be zero.

Remark 1: The RPC scheme is developed for system (1) over an imperfect network with following issues:

- τ^{ca}, τ^{sc}, and τ^{com} in turn denote communication delays on forward and backward channels, and computation delay. The computation delay can be bounded, $\tau_k^{com} \le \bar{\tau}^{com}$. Two threshold values, $\bar{\tau}^{ca}$ and $\bar{\tau}^{sc}$, are used to classify delays in which large delays are treated as packet dropouts:

$$\begin{cases} \tau_k^{ca} \le \bar{\tau}^{ca} \text{ or } \tau_k^{sc} \le \bar{\tau}^{sc} \rightarrow \text{Small delay} \\ \tau_k^{ca} > \bar{\tau}^{ca} \text{ or } \tau_k^{sc} > \bar{\tau}^{sc} \rightarrow \text{Packet dropout} \\ \tau_k = \tau_{k-1}^{sc} + \tau_k^{com} + \tau_k^{ca}; \end{cases} \tag{2}$$

- "Virtual" switches, S^{ca} and S^{sc}, in turn represent the packet dropouts in the forward and backward channels: S_k^{ca} (or S_k^{sc}) is opened (=1) when a packet loss event exists; p_k^{ca} and p_k^{sc} denote the numbers of continuous packet dropouts on forward and backward channels up to step k^{th}.
- The sensors are time-driven with a variable sampling period T_k. This period is online regulated in advance to ensure that it covers the total system delay in (2), $\tau_k \le T_k$;
- The controller and actuators are hybrid time-event-driven. They are event-driven if the delays satisfy (2); otherwise, they are time-driven to avoid the packet dropout problem.

III. Robust Predictive Control Approach

A generic NCS architecture using the proposed RPC scheme is depicted in Fig. 1. Generally, the desired tasks consists of a main task to control system (1) (including n

actuators and n sensors) using the RPC and, other execution tasks which cause unavoidable computation delays. The RPC mainly consists of five modules: NPD detector, PINNGM-based predictor, variable sampling period adjuster (VSPA), SAB buffer and ASISFC controller.

A. NPD Detector

The NPD is to detect online accurately both the time delays and packet dropouts based on (2) and then, to support these information to the other control modules.

Remark 2: using the same hardware and logics developed in [36], the NPD employs the PIC18F4620 MCU (from Microchip) equipped with the 4-MHz oscillator and the embedded time stamp-based detection logic to manage the networked system. Thus, real-time measurement accuracy is $50\mu s$ [36] and, the NDP outputs for each step are the delay set, $\{\tau_k^{ca}, \tau_k^{sc}, \tau_k^{com}\}$, and packet dropout set, $\{p_k^{ca}, p_k^{sc}\}$.

Based on Remark 1, the controller and actuators are selected as time or event -driven according to the real-time measurement using the NDP and condition (2). Thus, it is possible to use a so-called hybrid period T^C (or T^A) to represent the driven state of the controller (or actuators). From Remark 2, T^C and T^A can be derived as

$$\begin{cases} T_k^C = \tau_k^{com} + \min\left(\tau_k^{ca}, \bar{\tau}^{ca}\right) + \min\left(\tau_k^{sc}, \bar{\tau}^{sc}\right) \le \bar{\tau}^d \\ T_k^A = \min\left(\tau_{k-1}^{sc}, \bar{\tau}^{sc}\right) + \tau_k^{com} + \min\left(\tau_k^{ca}, \bar{\tau}^{ca}\right) \le \bar{\tau}^d \\ \bar{\tau}^d = \bar{\tau}^{com} + \bar{\tau}^{ca} + \bar{\tau}^{sc}; \end{cases} \tag{3}$$

B. PINNGM-Based Predictor

Grey model is known as a feasible solution for online prediction while requiring only a few historical data about the predicted object [29]-[36]. Although the AGM(1,1) proposed in [36] could overcome all drawbacks of typical grey models, it requires the fuzzy control knowledge from the user once employing this model for each specific application. In this study, a grey model is efficiently developed in a general form of first order – N variables and denoted as PINNGM(1,N).

Remark 3: The PINNGM is developed with following characteristics:

- The model is able to forecast any random time-series data by using a simple data conversion with additive factors.
- A recurrent signal is integrated as a model input to exhibit the dynamic temporal behavior.
- The model consists of a main grey model, tagged as MGM(1,N), and a proportional-integral (PI) -based neural network weight tuner, tagged as PINNWT.
- Lyapunov stability constrain is used to guarantee the robust prediction.

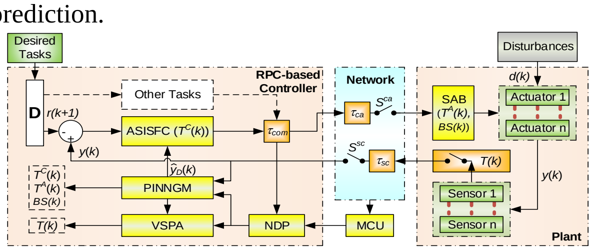

Fig. 1. A generic NCS architecture using proposed RPC approach.

As stated in the Introduction and from Fig. 1, the PINNGM(1,N) is employed with two main functions:

- First, to perform s_k-step-ahead prediction of the system delays, $\{\hat{\tau}_{k+i_s}^{com}, \hat{\tau}_{k+i_s}^{ca}, \hat{\tau}_{k+i_s}^{sa}\}$, and packet dropouts, $\{\hat{p}_{k+i_s}^{ca}\}, i_s = 1, \ldots, s_k$ where s_k is defined as:

$$\begin{cases} s_k \ge 2 \\ s_k = \min\left(i\right) \text{ at which } \hat{p}_{k+i}^{ca} = 0;\ i = 2, 3, \ldots \end{cases} \tag{4}$$

  These predicted values are used to set the sampling period for the sensors and the size of the SAB buffer.
- Second, to estimate the current system response (when existing a packet dropout on the backward channel) as well as the s_k-step-ahead responses. This information is then fed into the ASISFC to derive in advance the s_k-step-ahead control inputs to compensate the packet dropouts on the forward channels.

C. VSPA Adjuster

Based on Remark 1, it is necessary to regulate the sensors' sampling period synchronously with the delays defined by (2). By considering the system process time constant as T_p, the sampling period for a future step $(k+i_s)^{\text{th}}$ is defined based on the PINNGM and a MAX-MIN rule as

$$\begin{aligned} &T_{k+i_s} = k_T T_0,\ k_T = \left\lceil T_{new} / T_0 \right\rceil \\ &T_{new} = \max\left[T_0, \min\left(\hat{\tau}_{k+i_s-1}^{sc}, \bar{\tau}^{sc}\right) + \hat{\tau}_{k+i_s}^{com} + \min\left(\hat{\tau}_{k+i_s}^{ca}, \bar{\tau}^{ca}\right)\right] \\ &T_0 = 0.1T_p,\ \left(T_0 \text{is initial sampling period}\right) \end{aligned} \tag{5}$$

where $\lceil * \rceil$ is ceiling function to return nearest integer value.

Meanwhile, the hybrid periods, T^C (or T^A), are estimated in advanced in order to perform the s_k-step-ahead control:

$$\begin{cases} \hat{T}_k^C = \hat{\tau}_k^{com} + \min\left(\hat{\tau}_k^{ca}, \bar{\tau}^{ca}\right) + \min\left(\hat{\tau}_k^{sc}, \bar{\tau}^{sc}\right) \le \bar{\tau}^d \\ \hat{T}_k^A = \min\left(\hat{\tau}_{k-1}^{sc}, \bar{\tau}^{sc}\right) + \hat{\tau}_k^{com} + \min\left(\hat{\tau}_k^{ca}, \bar{\tau}^{ca}\right) \le \bar{\tau}^d \end{cases} \tag{6}$$

D. SAB Buffer

To eliminate forward packet dropout effect, the SAB is only utilized at the plant side. Selection of the buffer size (BS) affects directly on the control performance. With the ZOH, the controlled system is difficult to follow the desired goal when there is a large number of forward packet dropouts. On the contrary, the control performance becomes poor when using a large size buffer (designed for the worst network state). Thus, the key principle here is to optimize online the buffer size SB based on the PINNGM to store enough commands to drive the actuators during each p_k^{ca} continuous forward packet dropouts. For each working step, the commands for actuators are then extracted from the SAB following the first-in-first-out rule.

The procedure to tune the SAB size can be described as:

Initial step: Define the minimum size of the SAB, BS_{min}. In this case, it is selected as, BS_{min} = min(s_k)= 2. Then, set the initial size of the SAB: $BS_0 = BS_{min}$.

Step 1: For each step $(k+1)^{th}$ at the controller side, based on the s_k-step-ahead prediction to:

- Assign value of $BS_{k+1} = s_k$;

- Generate a set of time-stamped control inputs, $\{u_{k+1},...,u_{k+s_k}\}$ using the ASISFC.

Step 2: For each step $(k+1)^{th}$ at the plant side:

- If a new package sent from the controller is arrived:
  + Re-size the SAB with the new size BS_{k+1}.
  + Update the new control inputs and store in the SAB.
- Extract data from the SAB to drive the actuators based on the first-in-first-out principle.

E. ASISFC controller

The ASISFC takes part in driving the actuators to reach the desired target. As mentioned above, the packet dropout problem is compensated by the use of NDP, PINNGM and SAB with s_k-step-ahead control technique. Thus, the ASISFC is designed to address only the small delays defined by (2) with following characteristics:

- The state feedback control algorithm is used to generate the main control input;
- An integral term is added to the control algorithm to compensate the bad effects of disturbances.
- A state observer is implemented to adapt to the time variant system states.
- Based on the threshold values of small delays, a set of the control gains is pre-derived for a set of system states with small delay regions and stored in a look-up table.

IV. PINNGM-BASED PREDICTOR

A. Main Grey Model MGM(1,N)

This section is to generate a grey model between N relevant objects (or characteristics), $\{y_{obj_1}, y_{obj_2},..., y_{obj_N}\}$, of a system. Without loss of generality, object 1st is assumed to be the object which needs to be estimated based on a historical data set of the relevant objects. The procedure to predict this object using the MGM(1,N) can be designed with following steps:

Step 1: Prepare an input grey sequence for each object obj_i:

- Update the data sequence with the newest values:

$$\left\{y_{obj_i}(t_{O1}), y_{obj_i}(t_{O2}),..., y_{obj_i}(t_{Om})\right\};\ i=1,...,N; m\geq 5. \tag{7}$$

- Define corresponding raw input grey sequences as

$$y^{(0)}_{raw_i} = \left\{y^{(0)}_{raw_i}(t_1), y^{(0)}_{raw_i}(t_2),..., y^{(0)}_{raw_i}(t_n)\right\}; i=1,...,N; n\geq 5. \tag{8}$$

- Define a data distribution checking condition:

$$\begin{gathered}\Delta t_{Oi}/T_{MGM} < 2;\ \Delta t_{Oi} = t_{Oi} - t_{O(i-1)};\ \forall i\in[2,...,m] \\ \Delta t_{O1} = t_{O1} - t_{Olast}; t_{Olast}: \text{previous time stamp of } y_{obj_i}(t_{O1});\end{gathered} \tag{9}$$

here T_{MGM} is the desired prediction sampling period;

- If (9) is satisfied, then update directly sequence (8) as the (7); Otherwise, sequence (8) is created as sequence (10) with approximately equal time intervals based on sequence (7) and the spline function, SP, introduced in [36]:

$$\begin{gathered}y^{(0)}_{raw_i}(t_1) \equiv y_{obj_i}(t_{O1}); y^{(0)}_{raw_i}(t_k) = SP(t_k - t_{Oj});\ t_{Oj}\leq t_k \leq t_{O(j+1)}; \\ i=1,...,N; k=2,...,n;\ n=\lfloor (t_{Om}-t_{O1})/T_{MGM}\rfloor; \mathrm{j}=1,...,m-1;\end{gathered} \tag{10}$$

here $\lfloor * \rfloor$ is floor function to return nearest integer value.

- To exhibit the dynamic temporal behavior of the model, a recurrent signal as the current-step predicted value of the predicted object, $\hat{y}^{(0)-}_{raw_1}(t_n)$, is generated and added to its sequence $y^{(0)}_{raw}(t_n)$ to become:

$$Y^{(0)}_{raw_1} = \left\{y^{(0)}_{raw_1}(t_1), y^{(0)}_{raw_1}(t_2),..., \hat{y}^{(0)-}_{raw_1}(t_n)\right\};\ n\geq 5. \tag{11}$$

Remark 4: By adding two non-negative factors c_1 and c_2 derived by Theorem 1 in [36], sequence (11) satisfies the two checking conditions of a grey model: positive sequence and quasi-smooth – quasi-exponential law checking condition.

- Generated a grey sequence (12) using Remark 4:

$$\begin{gathered}y^{(0)} = \left\{y^{(0)}(t_1), y^{(0)}(t_2),..., y^{(0)}(t_n)\right\} > 0;\ n\geq 5 \\ y^{(0)}(t_k) = Y^{(0)}_{raw_1}(t_k) + c_1 + c_2;\ k=1,...,n.\end{gathered} \tag{12}$$

Step 2: Generate a new series $y^{(1)}$ by generating from $y^{(0)}$:

$$\begin{aligned}y^{(1)}(k) &= \sum_{i=1}^{k} y^{(0)}(t_i)\times \Delta t_i, k=1,2,...,n \\ &= \sum_{i=1}^{k}\left(Y_{raw}(t_i)+c_1+c_2\right)\times\Delta t_i.\end{aligned} \tag{13}$$

Step 3: Create the background series $z^{(1)}$ from $y^{(1)}$ as

$$z^{(1)}(t_k) = w^1_k y^{(1)}(t_k) + w^2_k y^{(1)}(t_{k-1}); k=2,...,n \tag{14}$$

where $\left\{w^1_k, w^2_k\right\}$ is a weight factor set which is designed as

$$\begin{cases} w^1_k = 0.5(1-\delta_1) + \delta_1\delta_2 w^{PINN}_k + \delta_1(1-\delta_2)\left(1-w^{PINN}_k\right) \\ w^2_k = 0.5(1-\delta_1) + \delta_1(1-\delta_2) w^{PINN}_k + \delta_1\delta_2\left(1-w^{PINN}_k\right)\end{cases} \tag{15}$$

where $0 < w^{PINN}_k < 1$ is derived using the PINNWT tuner; $\{\delta_1,\delta_2\}$ is a set of activated factors and defined as

$$\{\delta_1,\delta_2\} = \begin{cases}\{1,0\}, \text{IF}: y^{(0)}_{raw_1}(t_k) = SP(t_k); y^{(0)}_{raw_1}(t_{k-1}) = y_{obj_1}(t_{k-1}) \\ \{1,1\}, \text{IF}: y^{(0)}_{raw_1}(t_k) = y_{obj_1}(t_k); y^{(0)}_{raw_1}(t_{k-1}) = SP(t_{k-1}) \\ \{0,1\}, \text{Others}.\end{cases} \tag{16}$$

Step 4: Establish the grey differential equation

$$y^{(0)}(t_k) + az^{(1)}(t_k) = \delta_3 b_1 + (1-\delta_3)\sum_{i=2}^{N} b_i y_{raw_i}(t_k). \tag{17}$$

where δ_3 is an activated factor which is enabled (or 1) when N=1, and vice versa.

By employing the least square estimation ([30],[36]), one has:

$$\hat{\beta}_{ab} = \begin{bmatrix}\hat{a} & \hat{b}_1 & \cdots & \hat{b}_N\end{bmatrix}^T = (B^T B)^{-1}B^T Y \tag{18}$$

where

$$B = \begin{bmatrix} -z^{(1)}(t_2) & \delta_3 & (1-\delta_3)y_{raw_2}(t_2) & \cdots & (1-\delta_3)y_{raw_N}(t_2) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ -z^{(1)}(t_n) & \delta_3 & (1-\delta_3)y_{raw_2}(t_n) & \cdots & (1-\delta_3)y_{raw_N}(t_n)\end{bmatrix},$$

$$Y = \begin{bmatrix} y^{(0)}(t_2) & y^{(0)}(t_3) & \cdots & y^{(0)}(t_n)\end{bmatrix}^T.$$

Step 5: The MGM(1,N) prediction is then setup as:

$$\hat{y}^{(0)}(t_k) = \frac{\delta_3\hat{b}_1 + (1-\delta_3)\sum_{i=2}^{N}\hat{b}_i y_{raw_i}(t_k) - \hat{a}\left(w^1_k + w^2_k\right)y^{(1)}(t_{k-1})}{1 + w^1_k \hat{a}\Delta t_k}. \tag{19}$$

Step 6: Combining (19), (13) and (12), the predicted value of y_{obj_1} at step $(k+p_k)^{th}$ can be computed as

$$\hat{y}^{(0)}_{raw_1}\left(t_{k+p_k}\right) = \hat{y}^{(0)}\left(t_{k+p_k}\right) - c_1 - c_2 \quad (20)$$

where p_k is the step size of the grey predictor.

B. PINNWT Tuner and MGM(1,N) Stability Analysis

By considering the prediction of y_{obj_1} using the MGM(1,N) as a tracking control problem with the control input is w_k^{PINN} (in (15)), it is possible to derive a robust controller to enhance the robust prediction of this model. Here based on Remark 3, the PINNWT controller is designed with a Lyapunov stability condition to regulate the 'control input' w_k^{PINN}.

Here, the PINNWT consists of three layers: an input layer as a prediction error sequence $e_k^{MGM} = \{e_k^1, ..., e_k^{n-1}\}$, $e_k^i = y^{(0)}_{raw}(t_i) - \hat{y}^{(0)}_{raw}(t_i)$, a hidden layer with two nodes, P and I, following PI algorithm, and an output layer to compute the weight factor, w_k^{PINN}. Define $\{w_k^{Pi}, w_k^{Ii}\}$ is a weight vector of the hidden nodes with respect to input i^{th}, and $\{w_k^P, w_k^I\}$ is the weight vector of the output layer. Therefore, the output from each hidden node is derived based on PI algorithm:

$$\begin{cases} O_k^P = \sum_{i=1}^{n-1} w_k^{Pi} e_k^i : \text{Node P} \\ O_k^I = O_{k-1}^I + \sum_{i=1}^{n-1} w_k^{Ii} e_k^i : \text{Node I} \end{cases} \quad (21)$$

Then, the output from the network or the control input for 'plant' MGM is obtained using sigmoid activation function:

$$w_k^{PINN} = f\left(O_k^{NN}\right) = \left(1 + e^{-O_k^{NN}}\right)^{-1}, O_k^{NN} = w_k^P O_k^P + w_k^I O_k^I \quad (22)$$

In order to ensure the robust prediction, the back-propagation algorithm based on the Lyapunov stability condition is used to tune the PINNWT. Define a prediction error function as

$$E_k^{MGM} = 0.5\sum_{i=1}^{n-1}\left(y^{(0)}_{raw_1}(t_i) - \hat{y}^{(0)}_{raw_1}(t_i)\right)^2 = 0.5\sum_{i=1}^{n-1}\left(e_k^i\right)^2. \quad (23)$$

Thus, the weight factors of the PINNWT are online tuned for each step, $(k+1)^{th}$, as follows:

$$\begin{cases} w_{k+1}^{P/I} = w_k^{P/I} - \eta_k^{P/I} \partial E_k^{MGM} / \partial w_k^{P/I} \\ w_{k+1}^{Pi/Ii} = w_k^{Pi/Ii} - \eta_k^{Pi/Ii} \partial E_k^{MGM} / \partial w_k^{Pi/Ii} \end{cases} \quad (24)$$

where $\eta_k^{P/I}, \eta_k^{Pi/Ii}$ are learning rates within range [0,1]; the other factors in (24) are derived using the partial derivative of the error function (23) with respect to each decisive parameter and chain rule method [37].

Theorem 1: By selecting properly the learning rates $\eta_k^{Pi/Ii} \equiv \eta_k^{P/I} = \eta_k$ for step $(k+1)^{th}$ to satisfy (25), then the stability of the PINNGM prediction is guaranteed.

$$\sum_{i=1}^{n-1}\left(e_k^i F_k + 0.5 F_k^2 \eta_k\right) \le 0 \quad (25)$$

with

$$F_k = -\sum_{j=1}^{n-1}\left(\frac{e_k^j}{w_k^{Pi}}\frac{\partial E_k^{MGM}}{\partial w_k^{Pj}} + \frac{e_k^j}{w_k^{Ii}}\frac{\partial E_k^{MGM}}{\partial w_k^{Ij}}\right).$$

Proof: Define a Lyapunov function as

$$V_k^{MGM} = 0.5\sum_{i=1}^{n-1}\left(y^{(0)}_{raw}(t_i) - \hat{y}^{(0)}_{raw}(t_i)\right)^2 = 0.5\sum_{i=1}^{n-1}\left(e_k^i\right)^2. \quad (26)$$

The change of this Lyapunov function is derived as

$$\begin{aligned} \Delta V_{k+1}^{MGM} &= 0.5\sum_{i=1}^{n-1}\left(\left(e_{k+1}^i\right)^2 - \left(e_k^i\right)^2\right) \\ &= \sum_{i=1}^{n-1}\left(e_k^i \Delta e_k^i + 0.5\left(\Delta e_k^i\right)^2\right), \left(e_{k+1}^i = e_k^i + \Delta e_k^i\right). \end{aligned} \quad (27)$$

From the PINNWT structure, one has:

$$\Delta e_k^i = \sum_{j=1}^{n-1}\left(\frac{\partial e_k^i}{\partial w_k^{Pj}}\Delta w_k^{Pj} + \frac{\partial e_k^i}{\partial w_k^{Ij}}\Delta w_k^{Ij}\right). \quad (28)$$

The terms $\Delta w_k^{Pj}, \Delta w_k^{Ij}$ are from (24). By using the partial derivative and selecting $\eta_k^{Pj/Ij} \equiv \eta_k^{P/I} = \eta_k$, (28) becomes:

$$\Delta e_k^i = \eta_k F_k. \quad (29)$$

From (29), (27) is rewritten as

$$\Delta V_{k+1}^{MGM} = \sum_{i=1}^{n-1}\left(e_k^i F_k + 0.5 F_k^2 \eta_k\right). \quad (30)$$

The MGM(1,1) prediction is guaranteed to be stable only if $\Delta V_{k+1}^{MGM} \le 0, \forall k$. It is clear that except η_k, the other factors in (30) can be determined online based on the prediction error and the chain rule method [37]. Hence for each working step, it is easy to select a proper value of η_k to make (25) satisfy. Therefore, the proof is completed.

C. PINNGM(1,N) Application to RPC and Examples

As presented in Section III.B, the PINNGM model is utilized for the RPC design as:

- To perform s_k-step-ahead prediction of the system delays/packet dropout, PINNGM(1,1) is employed, in which the model input is a vector of historical delays/packet dropouts;
- To perform s_k-step-ahead prediction of the system response, PINNGM(1,2) is chosen, in which the model inputs are the historical data of system response and control input. This is because the system response is relevant to the control input.

The applicability of the PINNGM models is investigated via the following examples.

Example 3.1: A comparative study of four grey models, GM(1,1), SAUIGM(1,1) [29], AGM(1,1) [36] and PINNGM(1,1), has been carried out to investigate the capability in predicting the network delay problem studied in [29] and [36]. Because the GM is limited to equal-time-series prediction, the real-time estimations were performed for a 30-second period in which there were only time delays in the network. The delays were observed by the NPD with a fixed sampling rate of 10ms [29]. The one-step-ahead estimation results were obtained and analyzed in Table I using three evaluation criteria: average relative error (ARE), root mean square error (RMSE) and coefficient of determination (R2).

It is clear that the prediction accuracy was low using the typical GM due to its drawbacks [29]. Although the performance was improved by using the SAUIGM, there was no sufficient condition to ensure the prediction stability. Meanwhile by using the AGM and PINNGM, of which the model parameters were optimized via the Lyapunov conditions, the highest prediction accuracies were achieved as in Table I. However, the fuzzy inference of the AGM may need to be re-designed once changing the application or prediction step size. This is investigated through the second example.

Example 3.2: Series of delay predictions using the four grey models in Example 3.1 have been done with different prediction step sizes (varied from 1 to 30) to evaluate the s_k-step-ahead prediction ability. The comparison result is then displayed in Fig. 2. Here, the GM prediction accuracy was totally reduced according to the step size increase. Although the SAUIGM could improve the accuracy, its fitness was not kept stably. Only with the AGM and PINNGM which were optimized by the robust conditions, the precise estimations were almost guaranteed. However, the performance of the PINNGM was better than that of the AGM due to the adaptability of the neural network, PINNWT, over the change of prediction step sizes.

V. ASISFC Controller Design

A. ASISFC-based NCS Analysis

As described in the Introduction, Remark 1 and Section III, the ASISFC controller is designed to create the s_k-step-ahead control inputs for networked system (1) with only small delays defined in (2). Without loss of generative, the followings are to design the control gains for the networked system (1) only at step $(k+1)^{th}$ which can be simplified to the following delayed system:

$$x_{k+2} = A_{k+1}x_{k+1} + B^1_{k+1}u_{k+1} + B^2_{k+1}u_k \tag{31}$$

where the matrices in (31) are derived from (3) and (5):

$$A_{k+1} = e^{AT_{k+1}}, B^1_{k+1} = \int_0^{T_{k+1}-T^A_{k+1}} e^{At}Bdt, B^2_{k+1} = \int_{T_{k+1}-T^A_{k+1}}^{T_{k+1}} e^{At}Bdt;$$

(with step $(k+i_s)^{\text{th}}$ (i_s =2,…,s_k), T^A_{k+1} and T^A_{k+1} in (31) are replaced by $\hat{T}^A_{k+i_s}$ and $\hat{T}^A_{k+i_s}$).

From (3), $T^A_{k+1} \le \overline{\tau}^d$, and Section III.E, it is possible to consider the control design for system (31) similarly as for a set of system with D small delay regions having the same intervals:

TABLE I

Real-Time One-Step-Ahead Delay Predictions

Evaluation criteria	Prediction models			
	GM	SAUIGM	AGM	PINNGM
ARE (%)	8.6720	0.9664	0.5271	0.4783
RMSE (10^{-4})	28.4937	3.7635	2.026	1.8398
R2	0.8818	0.9679	0.9822	0.9891

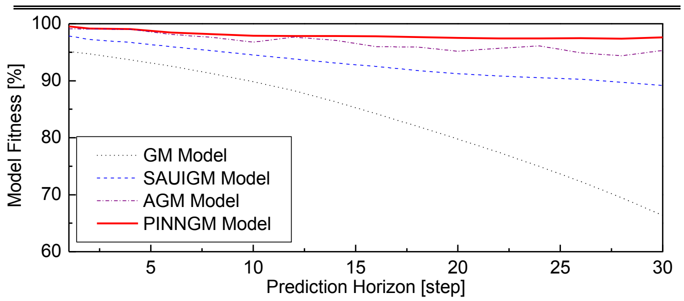

Fig. 2. Delay prediction model fitness vs. prediction step size s_k.

$$x_{k+2} = A_{k+1}x_{k+1} + \sum_{i=1}^{D}\delta^i_4\left(B^{1i}_{k+1}u_{k+1} + B^{2i}_{k+1}u_k\right) \tag{32}$$

where $B^{1i}_{k+1} = \int_0^{T_{k+1}-\tau^d_i} e^{At}Bdt, B^{2i}_{k+1} = \int_{T_{k+1}-\tau^d_i}^{T_{k+1}} e^{At}Bdt, \tau^d_i = \dfrac{\overline{\tau}^d}{D}$;

$\delta^i_4 = 1$(activated) if $\tau^d_{i-1} \le \hat{T}^A_{k+1} \le \tau^d_i$.

To ensure a robust tracking performance with disturbance attenuation, the control rule for system (32) is designed as a state feedback gain combine with an integral gain:

$$u_{k+1} = K^1x_{k+1} + K^2x^d_{k+1} \tag{33}$$

where $K^1 \in R^{mu\times mx}$ and $K^2 \in R^{mu\times md}$ are the desired control gain matrices; $x^d_{k+1} \in R^{md}$ is the disturbance integration factor:

$$x^d_{k+2} = x^d_{k+1} + \left(y_{k+1} - r\right) = x^d_{k+1} + Cx_{k+1} \tag{34}$$

Replacing (33) and (34) in to (32), one has:

$$\begin{aligned} x_{k+2} = &\left[A_{k+1} + \sum_{i=1}^{D}\delta^i_4 B^{1i}_{k+1}K^1\right]x_{k+1} + \left(\sum_{i=1}^{D}\delta^i_4 B^{2i}_{k+1}\left(K^1 - K^2C\right)\right)x_k \\ &+\left[\sum_{i=1}^{D}\delta^i_4\left(B^{1i}_{k+1} + B^{2i}_{k+1}\right)K^2\right]x^d_{k+1} \end{aligned} \tag{35}$$

or:

$$X_{k+2} = \Psi_{k+1}X_{k+1} \tag{36}$$

where, $X_{k+1} = \begin{bmatrix} x_{k+1} \\ x_k \\ x^d_{k+1} \end{bmatrix}; \Psi_{k+1} = \begin{bmatrix} \psi_{11} & \psi_{12} & \psi_{13} \\ I & 0 & 0 \\ C & 0 & I \end{bmatrix};$

$$\psi_{11} = A_{k+1} + \sum_{i=1}^{D}\delta^i_4\left(B^{1i}_{k+1}K^1 - B^{2i}_{k+1}K^2C\right),$$

$$\psi_{12} = \sum_{i=1}^{D}\delta^i_4 B^{2i}_{k+1}K^1, \psi_{13} = \sum_{i=1}^{D}\delta^i_4\left(B^{1i}_{k+1} + B^{2i}_{k+1}\right)K^2$$

To adapt to the time-variation of the system state, a state observer is implemented into system (32):

$$\hat{x}_{k+2} = A_{k+1}\hat{x}_{k+1} + \sum_{i=1}^{D}\delta^i_4\left(B^{1i}_{k+1}u_{k+1} + B^{2i}_{k+1}u_k\right) + L\left(y_{k+1} - \hat{y}_{k+1}\right) \tag{37}$$

where $\hat{x}_{k+1} \in R^{mx}, \hat{y}_{k+1} \in R^{my}$ and $L \in R^{mx \times my}$ denote the estimated state, estimated system output and observer gain, respectively. The state estimation error is $e_{k+1} = x_{k+1} - \hat{x}_{k+1}$.

The packet dropouts on the backward channel can be compensated by the PINNGM(1,2) (Section IV.C) to estimate the system response (denoted as $\hat{y}^M$). Thus, system (37) can be re-written in a general form as

$$\hat{x}_{k+2} = A_{k+1}\hat{x}_{k+1} + \sum_{i=1}^{D}\delta^i_4\left(B^{1i}_{k+1}u_{k+1} + B^{2i}_{k+1}u_k\right) + L\left(\overline{y}_{k+1} - \hat{y}_{k+1}\right) \tag{38}$$

$$\overline{y}(k+1) = \begin{cases} y(k+1), & if\ S^{sc}_k = 0 \\ \hat{y}^M(k+1), & if\ S^{sc}_k = 1 \end{cases} \tag{39}$$

Let $E_{k+1} = \begin{bmatrix} e_{k+1} & e_k & 0 \end{bmatrix}^T$, the estimation error can be represented as

$$E_{k+2}=\begin{bmatrix} A_{k+1}-LC & 0 & 0 \\ I & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} E_{k+1} \tag{40}$$

or

$$E_{k+2}=\Psi_{k+1}^{E} E_{k+1} \tag{41}$$

where, $\Psi_{k+1}^{E}=\begin{bmatrix} A_{k+1}-LC & 0 & 0 \\ I & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.

By combining (36) and (41), the closed-loop NCS using both state feedback control gain, disturbance integration gain and observer gain is expressed by

$$X_{k+2}=\Psi_{k+1} X_{k+1}-LCE_{k+1} \tag{42}$$

B. ASISFC Control Gain Design and Stability Analysis

Theorem 2 [25]: Separation principle – If there exit two positive matrices, P and P^E, such that:

$$\Psi_{k+1}^{T} P \Psi_{k+1}-P<0 \tag{43}$$

and,

$$(\Psi_{k+1}^{E})^{T} P^{E} \Psi_{k+1}^{E}-P^{E}<0 \tag{44}$$

hold, then the closed-loop NCS (42) is asymptotically stable.

Proof: The proof of this theorem is addressed in [25].

Based on Theorem 3, the control gain design for system (42) is now simplified to the control gain designs for system (36) and system (41).

Theorem 3: If there exists a positive definite matrix $P>0$, such that (43) hold, then the NCS (36) is asymptotically stable.

Proof: Select the following Lyapunov function:

$$V_{k+1}\left(X_{k+1}\right) \equiv V_{k+1}=X_{k+1}^{T} P X_{k+1}. \tag{45}$$

Using (36), the derivative of this function is obtained as

$$\begin{aligned} \Delta V_{k+1} &= V_{k+2}-V_{k+1}=X_{k+2}^{T} P X_{k+2}-X_{k+1}^{T} P X_{k+1} \\ &= X_{k+1}^{T}\left(\Psi_{k+1}^{T} P \Psi_{k+1}-P\right) X_{k+1}. \end{aligned} \tag{46}$$

Due to P>0, it is clear that $\Delta V_{k+1}<0$ if (43) holds, thus the system (36) is asymptotically stable, the proof is completed.

The sufficient condition for designing the controller is pointed out by Theorem 3. However when designing the control gains, the inequalities (43) may do not has the form of LMIs. In order to design the control gains, Theorem 4 is therefore introduced:

Theorem 4: If there exists a positive definite matrix $P>0$, such that the inequalities:

$$\begin{bmatrix} -P & \Psi_{k+1}^{T} \\ \Psi_{k+1} & -Q \end{bmatrix}<0 \tag{47}$$

hold with $PQ = I$, then the NCS (36) is asymptotically stable.

Proof: Applying the Schur complement [38] to the equalities (43), one has:

$$\begin{bmatrix} -P & \Psi_{k+1}^{T} \\ \Psi_{k+1} & -P^{-1} \end{bmatrix}<0. \tag{48}$$

Letting $P^{-1}=Q$, (48) becomes (47). The proof is finished.

Using Theorem 4 and the cone complementary linearization algorithm [39], the control gains can be solved by the following minimization problem involving LMI conditions:

$$\begin{aligned} &\text{minimize}\; trace(PQ) \\ &\text{subject to}\,(47), P>0, Q>0. \end{aligned} \tag{49}$$

Remark 5: by applying Theorem 2, system (41) is asymptotically stable if existing a positive matrix P^E which satisfies (44). Thus in this case, the stability of system (41) can be guaranteed by design the observer gain L in order to placing all eigenvalues of the characteristic (A_{k+1}-LC) inside the unit circle. This can be done easily by pole placement method [40].

Finally, the ASISFC control procedure for a NCS can be summarized as:

Step 1 – control gain preparation (off-line):

- Divide the delay range $\bar{\tau}^{d}$ into D small delay regions to input to (30);
- Design a set of control gains K^1 and K^2 for system (36) using Theorem 4 and (49);
- Design a control gain L for system (41) using the pole placement method;
- Generate a look-up table to store the designed control gains with respect to the set of delay regions.

Step 2 – s_k-step-ahead control command generation (online):

- Based on the predicted system delays, select properly the control gains from the gain look-up table;
- Generate the s_k-step-ahead driving commands with time stamps and send to the SAB.

VI. Case Study

This section considers a case study with speed tracking control of a networked DC servomotor system. The proposed RPC approach has been evaluated in a comparison with two controllers: the RVSPC developed in [36] and, the SSFC in [36] combined with a fixed buffer and the VSPA module, tagged as SSFC-FB.

A. System Configuration and Problem Investigation

Configuration of the networked DC servomotor system is presented in Fig. 3 and clearly described in [29] and [36]. The servomotor input-output transfer function can be expressed as

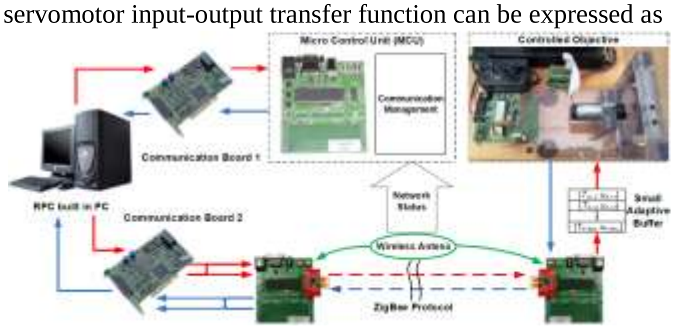

Fig. 3. Configuration of the networked servomotor control system.

$$G_p(s) = \frac{32300}{0.001s^2 + 20.93s + 493.4}.$$

The system can be also represented in the form of (1) with:

$$A = \begin{bmatrix} -235870 & -4934200 \\ 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 0 \\ 32298000 \end{bmatrix}^T.$$

Furthermore for the full evaluation, computation delays and two disturbance sources were applied to the system. The computation delays were observed from the control system of the real machine [36]. The first disturbance source came as the magnetic load applied to the motor shaft while the second source was simulated and added to the system feedback speed:

$$N(t) = \mathrm{Rand}_N(t) - A_N \sin(2\pi f_N t) - 1$$

here A_N was given randomly from 0.5 to 1; f_N was varied from 1 to 5 Hz; Rand_N was a noise with power 0.005.

B. Control Parameter Setting

First, the RVSPC control gains were derived as presented in [36]. Second, based on the network condition, the SSFC-FB control gains were computed by considering the total delay as $\overline{\tau}^d_{SSFC} = 0.1s$. Then, the SSFC-FB control gain was found as $K_{SSFC-FB} = \begin{bmatrix} 0.3652 & 0.2539 \end{bmatrix}$ while its buffer size was fixed as 5. Third for the RPC design, based on Remark 1, the delay threshold values were selected as: $\overline{\tau}^{com} = 0.025s$, $\overline{\tau}^{ca} = 0.035s$ and $\overline{\tau}^{sc} = 0.035s$. The state feedback and integral control gains were, therefore, derived for a set of the total delay regions ranging from 0 to 0.095 with interval of 0.01s as shown in Table II. Meanwhile, the observer gain was derived as [1; 1.3217].

C. Experimental Results and Discussion

The real-time speed tracking control evaluation in the imperfect conditions, including both the three delays, packet dropouts, and disturbances, has been then performed using the comparative controllers. Here, the first disturbance source always applied to the motor shaft while the second disturbance source was only added to the feedback speed after 15 seconds.

Firstly, the experiments on the system without the simulated disturbances were done in which the desired motor speed was set as a constant of 10rad/s. The tracking results were then obtained as demonstrated in Fig. 4. The one-step-ahead delay prediction using the PINNGM was observed as in Fig. 5.

TABLE II
RPC CONTROL GAINS ACCORDING TO TOTAL DELAY

$\hat{T}^A$	K^1	K^2
0~0.01	[0.2511 0.2001]	0.0038
0.01~0.02	[0.2587 0.2009]	0.0037
0.02~0.03	[0.2644 0.2037]	0.0037
0.03~0.04	[0.2723 0.2065]	0.0039
0.04~0.05	[0.2805 0.2086]	0.0038
0.05~0.06	[0.2869 0.2101]	0.0038
0.06~0.07	[0.3028 0.2152]	0.0039
0.07~0.08	[0.3213 0.2189]	0.0041
0.08~0.95	[0.3471 0.2234]	0.0040

In this case, the SSFC-FB controller could not ensure the smooth tracking result due to the use of fixed control gains and buffer, which were designed for the worst network conditions. Furthermore, the SSFC-FB lacked of the predictability of the system response, the controller could not derive the control inputs once there existed packet dropouts on the feedback channel. The tracking performance was significantly degraded when the packet dropout ratio increased and the second disturbance source was invoked into the system.

By using the RVSPC controller, the performance was improved by the advanced switching control based on the QFT and state control algorithms. However, without the use of buffer, the control performance was degraded in some regions with high packet dropout ratio. Additionally, without the disturbance compensation in designing the control gains, both the SSFC-FB and RVSPC could not cancelled the steady state error in the state feedback performances.

Only by using the proposed RPC approach which possesses the advantages of the SAB and s_k-step-ahead control, which combines both the state feedback control, disturbance rejection, and observer-based control, the best tracking result was achieved with fast response and high robustness even facing with the high ratio of packet dropouts.

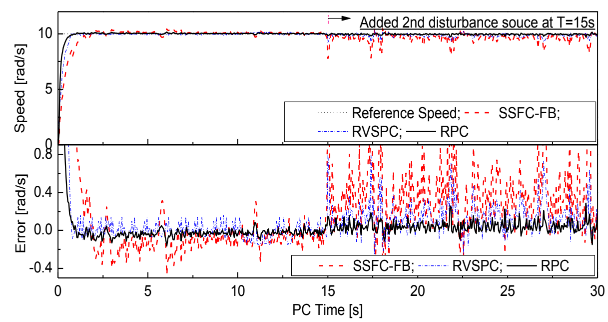

Fig. 4. Step tracking: performances comparison.

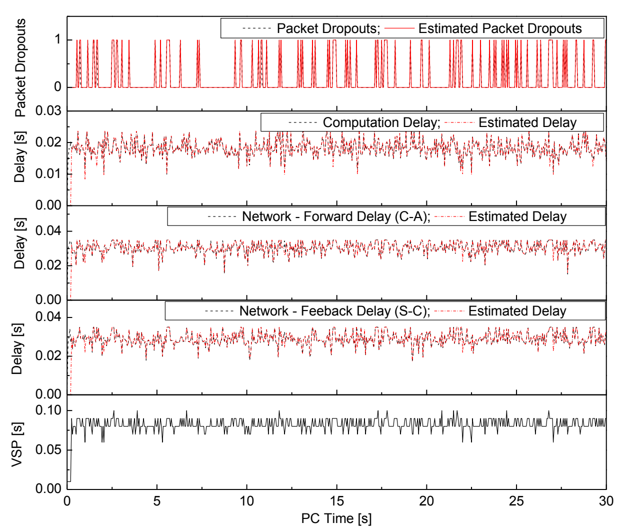

Fig. 5. Step tracking: one-step-ahead predicted delays and adaptive sampling period based on the PINNGM.

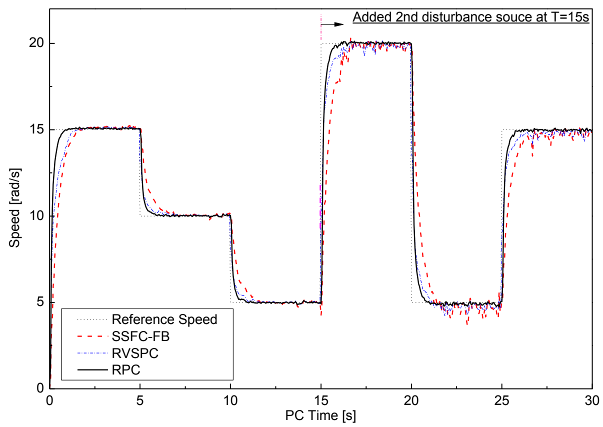

Fig. 6. Multi-step tracking: performance comparison.

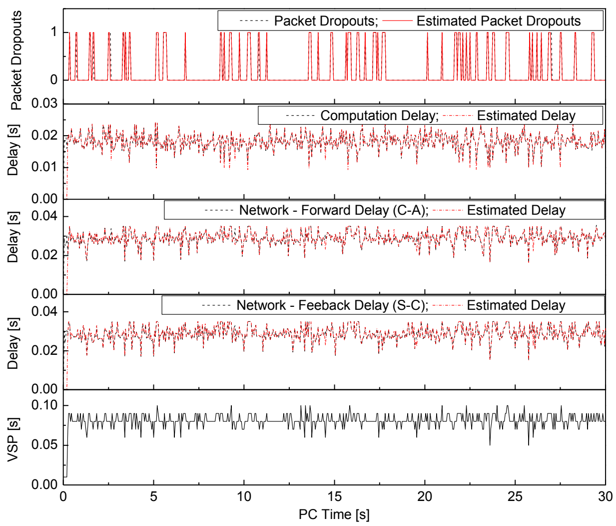

Fig. 7. Multi-step tracking: one-step-ahead predicted delays and adaptive sampling period based on the PINNGM.

TABLE III
COMPARISON OF NCS PERFORMANCES USING DIFFERENT CONTROLLERS

Controller	Step responses			
	Percent of Overshoot [%]	Settling Time [s]	Steady State Error [%] [15~30s]	Mean Square Error $[rad/s]^2$ 10^{-2} [15~30s]
SSFC-FB	0.35	1.65	9.07	30.73
RVSPC	0.45	0.95	6.01	4.71
RPC	0.47	0.74	1.63	1.14

Secondly, another experiment series were done for a multi-step speed tracking profile using the compared controllers. The results were then obtained in Fig. 6 and Fig. 7. These results again show the best performance was achieved by using proposed control method. This comes as no surprise, because the proposed method is based on both the advanced control modules as presented in Section III. The tracking performances using these three controllers were finally analyzed in Table III to state evidently the efficiency of the proposed approach.

VII. CONCLUSIONS

This paper presents the novel robust predictive control approach for NCSs included both the three delay components, packet dropouts and disturbances. Based on the robust prediction using the PINNGM, this control approach is the combination of both the advanced techniques, the small adaptive buffer with s_k-step-ahead control to compensate the packet dropouts, the observer-based integral state feedback control with smooth switching gains and adaptive sampling period to compensate the delays and disturbances.

The capability of the proposed RPC control scheme has been validated through series of real-time experiments with the speed tracking of the networked servomotor system. In addition, the comparative study with other advanced controllers has been done. The results proved convincingly the effectiveness as well as the applicability of the proposed method to NCSs.

REFERENCES

[1] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Control Syst. Mag.*, vol. 2, pp. 84-99, Feb. 2001.

[2] C. Huang, Y. Bai, X. Liu, "H-Infinity State Feedback Control for a Class of Networked Cascade Control Systems With Uncertain Delay," *IEEE Trans. Ind. Informat.*, vol. 6, no. 1, pp. 62-72, Feb. 2010.

[3] B. Yang, U. X. Tan, A. B. McMillan, R. Gullapalli, and J. P. Desai, "Design and Control of a 1-DOF MRI-Compatible Pneumatically Actuated Robot With Long Transmission Lines," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 6, pp. 1040-1048, Dec. 2011.

[4] D. Tian, D. Yashiro, and K. Ohnishi, "Wireless Haptic Communication Under Varying Delay by Switching-Channel Bilateral Control With Energy Monitor," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 3, pp. 488-498, Jun. 2012.

[5] C. C. Hua and X. P. Liu, "A New Coordinated Slave Torque Feedback Control Algorithm for Network-Based Teleoperation Systems," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 2, pp. 764-774, Apr. 2013.

[6] F. Yang, Z. Wang, Y. S. Hung, and G. Mahbub, "H^{∞} control for networked control systems with random communication delays," *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 511-518, Mar. 2006.

[7] C. C. Choi and W. T. Lee, "Analysis and Compensation of Time Delay Effects in Hardware-in-the-Loop Simulation for Automotive PMSM Drive System," *IEEE Trans. Ind. Electron.*, vol. 59, no. 9, pp. 3403-3410, Sep. 2012.

[8] K. H. Kim, M. Y. Sung, and H. W. Jin, "Design and Implementation of a Delay-Guaranteed Motor Drive for Precision Motion Control," *IEEE Trans. Ind. Informat.*, vol. 8, no. 2, pp. 351-365, May 2012.

[9] J. Q. Yi, Q. D. Wang, B. Zhao, and J. T. Wen, "BP neural network prediction-based variable-period sampling approach for networked control systems," *Appl. Math. Computa.*, vol. 185, pp. 976-988, Feb. 2007.

[10] L. L. Chien and L. H. Pau, "Design the remote control system with the time-delay estimator and the adaptive Smith predictor," *IEEE Trans. Ind. Informat.*, vol. 6, no. 1, pp. 73-80, Feb. 2010.

[11] L. E. Daniel and E. M. Mario, "Variable Sampling Approach to Mitigate Instability in Networked Control Systems with Delays," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 1, pp. 119-126, Jan. 2012.

[12] H. Zhang, J. Yang, and C.-Y Su, "T-S Fuzzy-Model-Based Robust H_{∞} Design for Networked Control Systems With Uncertainties," *IEEE Trans. Ind. Informat.*, vol. 3, no. 4, pp. 289-301, Nov. 2007.

[13] G. S. Tian, F. Xia1, and Y. C. Tian, "Predictive compensation for variable network delays and packet losses in networked control systems," *Comput. and Chemical Eng.*, vol. 39, pp. 152– 162, Apr. 2012.

[14] R. Yang, G.-P. Liu, P. Shi, C. Thomas, and M. V. Basin, "Predictive Output Feedback Control for Networked Control Systems," *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 512-520, Jan. 2014.

[15] T. Chai, L. Zhao, J. Qiu, F. Liu, and J. Fan, "Integrated Network-Based Model Predictive Control for Setpoints Compensation in Industrial Processes," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 417-426, Feb. 2013.

[16] Z.-H. Pang, G.-P. Liu, D. Zhou, and M. Chen, "Output Tracking Control for Networked Systems: A Model-Based Prediction Approach," *IEEE Trans. Ind. Electron.*, vol. 61, no. 9, pp. 4867-4877, Sep. 2014.

[17] Y. Shi, H. Fang and M. Yan, "Kalman filter based adaptive control for networked systems with unknown parameters and randomly missing

outputs," *Int. J. Robust Nonlinear Control*, vol. 19, no. 18, pp. 1976–1992, Dec. 2009.
[18] P. Ignaciuk, "Nonlinear Inventory Control With Discrete Sliding Modes in Systems With Uncertain Delay," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 559-568, Feb. 2014.
[19] A. Baños, F. Perez, and J. Cervera, "Network-Based Reset Control Systems With Time-Varying Delays," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 514-522, Feb. 2014.
[20] Y. Shi and B. Yu, "Output feedback stabilization of networked control systems with random delays modeled by Markov chains," *IEEE Trans. Autom. Control*, vol. 54, no. 7, pp. 1668 – 1674, Jul. 2009.
[21] Y. Shi and B. Yu, "Robust mixed H_2/H_∞ control of networked control systems with random time delays in both forward and backward communication links," *Automatica*, vol. 47, no. 4, pp. 754–760, Jan. 2011.
[22] Y. Shi, J. Huang, and B. Yu, "Robust Tracking Control of Networked Control Systems: Application to a Networked DC Motor," *IEEE Trans. Ind. Electron.*, vol. 60, no. 12, pp. 5864-5874, Dec. 2013.
[23] H. Gao, T. Chen, and J. Lam, "A new delay system approach to network-based control," *Automatica*, vol. 44, no. 1, pp. 39-52, Jan. 2008.
[24] H. Gao and T. Chen, "Network-Based H_∞ Output Tracking Control," *IEEE Trans. Autom. Control*, vol. 53, no. 3, pp. 655 – 667, Apr. 2008.
[25] H. Li, Z. Sun, M.-Y. Chow, and F. Sun, "Gain-Scheduling-Based State Feedback Integral Control for Networked Control Systems," *IEEE Trans. Ind. Electron.*, vol. 58, no. 6, pp. 2465-2472, Jun. 2011.
[26] H. Zhang, Y. Shi, and M. Liu, "H_∞ Step Tracking Control for Networked Discrete-Time Nonlinear Systems with Integral and Predictive Actions," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 337-345, Feb. 2013.
[27] L. Repele, R. Muradore, D. Quaglia, and P. Fiorini, "Improving Performance of Networked Control Systems by Using Adaptive Buffering," *IEEE Trans. Ind. Electron.*, vol. 61, no. 9, pp. 4847-4856, Sep. 2014.
[28] H. Li and Y. Shi, "Network-Based Predictive Control for Constrained Nonlinear Systems with Two-Channel Packet Dropouts," *IEEE Trans. Ind. Electron.*, vol. 61, no. 3, pp. 1574-1582, Mar. 2014.
[29] D. Q. Truong, K. K. Ahn, and N. T. Trung, "Design of An Advanced Time Delay Measurement and A Smart Adaptive Unequal Interval Grey Predictor for Real-time Nonlinear Control Systems," *IEEE Trans. Ind. Electron.*, vol. 60, no. 10, pp. 4574-4589, Oct. 2013.
[30] J. L. Deng, "Control problem of grey system," *Syst. Control Lett.*, vol. 1, no. 5, pp. 288-294, May 1982.
[31] R. C. Luo and T. M. Chen, "Autonomous Mobile Target Tracking System Based on Grey-Fuzzy Control Algorithm," *IEEE Trans. Ind. Electron.*, vol. 47, no. 4, pp. 920-931, Aug. 2000.
[32] E. Kayacan, Y. Oniz, and O. Kaynak, "A Grey System Modeling Approach for Sliding-Mode Control of Antilock Braking System," *IEEE Trans. Ind. Electron.*, vol. 56, no. 8, pp. 3244-3252, Aug. 2009.
[33] R. Guo, "Modeling imperfectly repaired system data via grey differential equations with unequal-gapped times," *Rel. Eng. & Sys. Safety*, vol. 92, no. 3, pp. 378-391, Mar. 2007.
[34] D. Q. Truong and K. K. Ahn, "Force control for hydraulic load simulator using self-tuning grey predictor – fuzzy PID," *Mechatronics*, vol. 19, no. 2, pp. 233-246, Mar. 2009.
[35] E. Kayacan, B. Ulutas, and O. Kaynak, "Grey system theory-based models in time series prediction," *Expert Syst. Appl.*, vol. 37, no. 2, pp. 1784-1789, Mar. 2010.
[36] D. Q. Truong and K. K. Ahn, "Robust Variable Sampling Period Control for Networked Control Systems," *IEEE Trans. Ind. Electron.*, vol. 62, no. 9, pp. 5630-5643, Sep. 2015.
[37] D. Q. Truong and K. K. Ahn, "Nonlinear black-box models and force-sensorless damping control for damping systems using magneto-rheological fluid dampers," *Sens. Actuators A, Phys.*, vol. 167, no. 2, pp. 556-573, Jun. 2011.
[38] J. G. VanAntwerp and R. D. Braatz, "A tutorial on linear and bilinear matrix inequalities," *J. Process Control*, vol. 10, no. 4, pp. 363-385, Aug. 2000.
[39] E. I. Ghaoui, F. Oustry, and M. A. Rami, "A cone complementarity linearization algorithm for static output-feedback and related problems," *IEEE Trans. Autom. Control*, vol. 42, no. 8, pp. 1171–1176, Aug. 1997.
[40] K. Ogata, *Modern Control Engineering*, 5th ed. Englewood Cliffs, NJ, USA: Prentice-Hall. Sep. 2009.

Truong Quang Dinh received the B.S degree in the department of Mechanical Engineering from Hochiminh City University of Technology in 2001, and the Ph.D. degree in the School of Mechanical Engineering from University of Ulsan in 2010.

He is currently a Research Fellow at the Warwick Manufacturing Group (WMG), University of Warwick, Coventry, UK. His research interests focus on control theories and applications, fluid power control systems, energy saving and power management systems, low carbon technologies for transports and construction sectors, and smart sensors and actuators.

Kyoung Kwan Ahn (M'06) received the B.S. degree in the department of Mechanical Engineering from Seoul National University in 1990, the M. Sc. degree in Mechanical Engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1992 and the Ph.D. degree with the title "A study on the automation of out-door tasks using 2 link electro-hydraulic manipulator from Tokyo Institute of Technology in 1999, respectively.

He is currently a Professor in the School of Mechanical Engineering, University of Ulsan, Ulsan, Korea. His research interests are design and control of smart actuator using smart material, fluid power control and active damping control, renewable energy. He is a Member of IEEE, ASME, SICE, RSJ, JSME, KSME, KSPE, KSAE, KFPS, and JFPS.

James Marco received the Ph.D. from the University of Warwick.

He is currently an Associate Professor at the Warwick Manufacturing Group (WMG), University of Warwick, Coventry, UK. His research interests include systems engineering, real-time control, systems modelling, design optimisation, design of energy management control systems. In particular, these generic and enabling technologies relate to the design of new low carbon technologies for the transport and energy storage sectors.