

Original citation:

Truong, Dinh Quang, Marco, James, Greenwood, David, Ahn, Kyoung Kwan and Yoon, Yong Il. (2018) Data-based predictive hybrid driven control for a class of imperfect networked systems. IEEE Transactions on Industrial Informatics .

Permanent WRAP URL:

<http://wrap.warwick.ac.uk/98227>

Copyright and reuse:

The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions. Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

Publisher's statement:

© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting /republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

A note on versions:

The version presented here may differ from the published version or, version of record, if you wish to cite this item you are advised to consult the publisher's version. Please see the 'permanent WRAP url' above for details on accessing the published version and note that access may require a subscription.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk

Data-Based Predictive Hybrid Driven Control for A Class of Imperfect Networked Systems

Truong Q. Dinh, *Member, IEEE*, James Marco, David Greenwood,
Kyoung K. Ahn, *Member, IEEE*, and Jong I. Yoon

Abstract—A data-based predictive hybrid driven control (DPHDC) approach is presented for a class of networked systems compromising both computation and communication delays, packet dropouts and disturbances. First, network problems are classified in a generic way which is used to design a network problem detector (NPD) capable of detecting online current delays and packet dropouts. Second, a single-variable first-order proportional-integral (PI) -based adaptive grey model (PIAGM(1,1)) is designed to predict future network problems and, to predict system disturbances. Third, a hybrid driven scheme integrated an optimal small buffer (OSB) is constructed to allow the system to operate without any interrupts due to large delays or packet dropouts. Furthermore, the OSB size is online optimized using adaptive grey fuzzy cognitive map technique. Forth, a prediction-based model-free adaptive controller (PMFAC) is developed to compensate for network problems. The DPHDC stability is theoretically proved while its effectiveness is demonstrated through a case study.

Index Terms— Networked control system, time delays, packet dropouts, prediction, model free, stability.

I. INTRODUCTION

RECENTLY with the rapid development of networking technologies, network-based remote control gains more and more attention due to its advantages over the traditional wiring control. However because of computation and communication resources restriction, time delay and packet

dropout are two critical and unavoidable factors existing in any networked control system (NCS). Only a small time delay can deteriorate the control performance and, even worse, cause the instability [1]–[3].

To address stability of NCSs with communication delays, many studies were carried out in which the controller designs were depended on assumptions that the time delay was constant [4], was bounded [2], [5]–[7], had a probability distribution function [8], or was represented based on time delay analyses [9], [10]. For NCSs with large delays, control concepts based on variable sampling periods using neural network or prediction theories were adopted [11]–[13]. Nevertheless, the observation of real delay data to train and construct the NCSs was not appropriately discussed. To solve this problem, a variable sampling period control concept for systems containing both random computation and communication delays has been developed and validated [3].

To deal with NCSs compromising both time delays and packet dropouts, many important methodologies, such as predictive control [14]–[16], adaptive control [17], hybrid control [18], robust state feedback control [19]–[21], and model predictive control (MPC) [22], were proposed. The concept of model based on the updating instants was carefully considered and implemented to derive the controllers in [19], [20]. Additionally, the network delays were presented in the general form and bounded by both upper and lower limits to reduce the conservatism problem. To minimize the communication effort as well as to improve the energy efficiency of NCSs, a robust state feedback control approach based on a so-called self-triggered sampling scheme was suggested [21]. To tackle both system state with network problems and tracking error, an extended form of MPC has been introduced in [22]. By using these techniques, although the NCS performances were remarkably improved over the traditional techniques, delays and packet dropouts were assumed to be priorly known and bounded. Another trend is known as control based on efficient networks with adaptive buffers or with compensation strategy [23], [24]. Though 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. As a feasible solution for these problems, a robust variable sampling period control approach (RVSPC) [25] has been proposed. Its controllability

Manuscript received August 1st, 2017; revised December 9th, 2017; accepted December 31st, 2017. This work was supported by Innovate UK through the Agile Power Management System (APMS), project number: 102437, in collaboration with the WMG Centre of High Value Manufacturing (HVM), Babcock and Potenza, and the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE) of Korea (No. 20162020107450).

T. Q. Dinh, J. Marco and D. Greenwood are with the Warwick Manufacturing Group (WMG), University of Warwick, Coventry, UK (email: james.marco@warwick.ac.uk; d.greenwood@warwick.ac.uk; corresponding author to provide phone: +44-24-765-74902; e-mail: q.dinh@warwick.ac.uk;).

K. K. Ahn is with the School of Mechanical Engineering, University of Ulsan, Namgu Muger2dong, Ulsan 680-749, Korea (email: kkahn@ulsan.ac.kr)

J. I. Yoon is with the Korea Construction Equipment Technology Institute, 36, Sandan-ro, Gunsan 54004, Korea (email: jiyoona@koceti.re.kr)

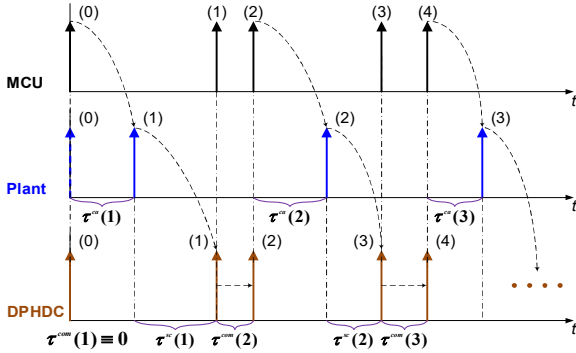


Fig. 2. Working principle of NPD.

Based on Assumptions 1 and 2, system (1) can be represented in a compacted form dynamic linearization:

$$\Delta y(k+1) = \Phi^T(k) \Delta U(k) \quad (2)$$

Thus, the incremental control algorithm is derived for system (1) using the MFAC theory [27] as follows:

$$\hat{\Phi}(k) = \hat{\Phi}(k-1) + \frac{\eta \Delta U(k-1)(\Delta y(k) - \hat{\Phi}^T(k-1) \Delta U(k-1))}{\mu + \|\Delta U(k-1)\|^2} \quad (3)$$

$$\hat{\phi}_i(k) = \hat{\phi}_i(0) \text{ if } |\hat{\phi}_i(k)| \leq \varepsilon, \text{ or } |\Delta u_i(k-1)| \leq \varepsilon \quad (4)$$

$$(U(k) = \{u_i(k)\}; \Phi(k) = \{\phi_i(k)\}; i = 1, \dots, m_u)$$

$$\Delta U(k) = G(k)(y^r(k+1) - y(k)), G(k) = \frac{\rho \hat{\Phi}(k)}{\lambda + \|\hat{\Phi}(k)\|^2} \quad (5)$$

where $y^r(k+1)$ is the tracking target of $y(k)$; $\hat{\phi}_i(k)$ is the estimation of $\phi_i(k)$ with initial value $\hat{\phi}_i(0)$; η, ρ are the step-sizes which are normally set within $(0, 1]$; μ, λ are the weight factors; ε is a small positive constant.

Definition 1: the problems in a network are defined as:

- Communication delays in forward and backward channels are in turn denoted τ^{ca} and τ^{sc} . Two threshold values, $\bar{\tau}^{ca}$ and $\bar{\tau}^{sc}$, are defined to classify small and large delays. A large delay is treated as a packet dropout:

$$\begin{cases} \tau^{ca}(k) \leq \bar{\tau}^{ca} \text{ or } \tau^{sc}(k) \leq \bar{\tau}^{sc} \rightarrow \text{Small delay} \\ \tau^{ca}(k) > \bar{\tau}^{ca} \text{ or } \tau^{sc}(k) > \bar{\tau}^{sc} \rightarrow \text{Packet dropout} \end{cases} \quad (6)$$

- A packet disorder is also considered as a packet dropout.
- “Virtual” switches, S^{ca} and S^{sc} , are in turn used to represent the packet dropouts in the forward and backward channels. S^{ca} (S^{sc}) is opened (or 1) once a packet dropout event exists.
- A set of continuous packet dropouts at timestamp k^{th} is denoted as $p^{ca}(k)$ or $p^{sc}(k)$.
- The computation delay, τ^{com} , is bounded, $\tau^{com}(k) \leq \bar{\tau}^{com}$.

The objective is to develop a networked control scheme of system (1) using the DPHDC approach in which the network problems are classified using Definition 1.

III. DATA-BASED PREDICTIVE HYBRID DRIVEN CONTROL

Architecture of a NCS using the DPHDC approach is depicted in Fig. 1. The system plant including a number of

actuators and sensors are driven by the DPHDC, for example, embedded in a personal computer (PC), via the network to follow given tasks. The network is with two modules attached to the controller and plant sides while the desired tasks consist of a main task to control the plant (1) using the DPHDC and other execution tasks, which could cause computation delays. Additionally, a micro control unit (MCU) is employed to equip with the NCS in order to measure the system run time.

Remark 1: The DPHDC approach is developed for system (1) over an imperfect network with following functionalities:

- The NPD based on Definition 1 and the MCU output is used to detect online precisely the network problems.
- The PIAGM(1,1) with prediction step size s_k employs the NPD outputs to predict the s_k -step ahead network problems.
- The OSB buffer is only placed at the plant side. This buffer size, b_{OSB} , is online regulated using the AGFCM based on the network problems and the system tracking performance.
- Both the controller, sensors and actuators are hybrid time-event-driven. They are event driven when there exist only small delays; otherwise, they are time driven. The sensors and actuators are synchronous.
- The PMFAC is to compensate for both the network problems classified by Definition 1 and the impacts of disturbances. According to the OSB size, b_{OSB} , a vector of control inputs are derived and sent to the OSB to guarantee the robust tracking performance of the actuators.

A. NPD Detector

Remark 2: From Remark 1 to construct the NPD, first, the MCU is selected as PIC18F4620 from Microchip equipped with the 4-MHz oscillator [3] to function as a reference clock to measure the real time and; second, timestamp-based detection logics developed through [25], [26], [31] are used to detect the system delays. As analyzed in [3], time resolution of the reference clock using the MCU is $50\mu s$. The MCU is connected to the DPHDC-embedded PC through a PCI card [3] and, therefore, their communication delay is neglected. The delay detection principle is depicted in Fig. 2. Once system is started, an activation flag is sent from the DPHDC - MCU to the plant to activate a clock at the plant side to support the detection of forward communication delay. For the synchronization of the clocks between the MCU and plant, the first delay between the MCU and plant, $\tau^{ca}(1)$, is determined using the well-known IEEE 1588-based precision time protocol, timestamps and the assumption that the forward and backward transport delay at the beginning of system operation is symmetric [32], [33]. Here, the clock frequency deviation of the crystal oscillators used in the NCS are not considered. This impact on the clock synchronization can be effectively compensated using the clock model presented in [33].

From Remark 2 and Definition 1, for each working step, the NPD detects accurately the current network states with a delay set, $\tau^{ca}(k)$, $\tau^{sc}(k)$ and $\tau^{com}(k)$, and a set of continuous packet dropouts, $p^{ca}(k)$ and $p^{sc}(k)$.

B. PIAGM(1,1) Predictor and Hybrid Driven Method

Grey model is known as a feasible tool for online prediction

while requiring only a few historical data about the predicted object [3], [25], [26], [34], [35].

Remark 3: In this study, the PIAGM(1,1) is constructed based on the general model - PINNGM(1,N) which has been successfully developed in [26]. The PIAGM(1,1) is therefore setup in the efficient way using an adaptive weight which is online regulated by a simple PI-type neural network with respect to the prediction error minimization. The robust prediction is guaranteed by a Lyapunov stability condition.

From the DPHDC architecture and Remark 1, the first PIAGM, PIAGM₁(1,1), uses the historical data of network problems (come from the NPD) to perform the s_k -step-ahead prediction of network delays, $\hat{\tau}^{ca}(k+i_s)$, $\hat{\tau}^{sc}(k+i_s)$ and $\hat{\tau}^{com}(k+i_s)$, and packet dropouts, $\hat{p}^{ca}(k+i_s)$ and $\hat{p}^{sc}(k+i_s)$ in which $i_s=2, \dots, s_k$, s_k is the smallest value satisfying: $(\hat{p}^{ca}(k+2)=0) \vee [(\hat{p}^{ca}(k+i_s-1) \neq 0) \wedge (\hat{p}^{ca}(k+i_s)=0)]$.

For a coming step $(k+1)^{th}$, using Definition 1 and the PIAGM₁(1,1) outputs, the hybrid time-event driven scheme can be expressed through dynamic sampling rates, $T^C(k+1)$ and $T^A(k+1)$, for the controller and plant sides, respectively:

$$\begin{cases} T^C(k+1) = \hat{\tau}^{com}(k+1) + \min(\hat{\tau}^{ca}(k+1), \bar{\tau}^{ca}) + \min(\hat{\tau}^{sc}(k+1), \bar{\tau}^{sc}) \\ T^A(k+1) = \min(\hat{\tau}^{sc}(k), \bar{\tau}^{sc}) + \hat{\tau}^{com}(k+1) + \min(\hat{\tau}^{ca}(k+1), \bar{\tau}^{ca}) \end{cases} \quad (7)$$

In addition, once a package loss is detected on the backward channel, the PIAGM is applied (denoted as PIAGM₂(1,1)) to estimate the current impact of noises and disturbances on the system performance, $\hat{y}^{ND}(k)$. The historical data of system disturbances is the set of time-series differences between last values of the actual system response and the corresponding system response which is calculated based on (2) (denoted as \hat{y}^{PMFAC}). The estimated disturbance impact is then fed into the PMFAC to speed up the convergence speed.

C. OSB Buffer

Without any buffer, the controlled system is difficult to follow the desired goal when there is a large numbers of packet dropouts. On the contrary, the control performance becomes poor when using a large size buffer. Since, the key concept is to regulate online the OSB size, $b_{OSB}(k+1) \geq s_k$, to store enough commands to drive the actuators during a period consisting of $\hat{p}^{ca}(k+s_k)$ continuous packet dropouts.

D. PMFAC Controller

The PMFAC takes part in driving the actuators to reach the desired targets based on Remark 1:

- For each step defined by the hybrid time-event driven method ($T^C(k)$) and based on OSB size, a set of the control inputs are generated for the next b_{OSB} steps using an iterative estimation scheme.
- Actual and estimated time delay factors and estimated disturbance impacts are invoked into the controller design process to compensate for these problems and, therefore, to ensure the stability.

IV. PIAGM(1,1) MODEL

A. PIAGM(1,1) Design

Remark 4: Any random data set of a predicted object, y_{obj} , can be converted into a grey sequence which satisfies both the grey checking conditions using the cubic spline interpolation (SP, see Appendix A) and two non-negative additive factors, c_1 and c_2 , derived by Theorem 1 in [25]. Next using [26], the PIAGM(1,1) model with the adaptive weight is constructed.

Based on Remark 4, the prediction procedure for an object, y_{obj} , using the PIAGM(1,1) can be expressed as follows:

Algorithm 1 PIAGM(1,1) Model to Predict An Object y_{obj}

Step 1: For any object with a data sequence $Y_{obj} = \{y_{obj}(t_{O1}), \dots, y_{obj}(t_{Om})\} (m \geq 4)$, the raw input grey sequence, $Y_{raw} = \{y_{raw}(t_1), \dots, y_{raw}(t_n)\} (n \geq 4)$, is derived as

$$y_{raw}^{(0)}(t_i) = \delta_i^0 y_{obj}(t_i) + (1 - \delta_i^0) SP(y_{obj}, t_i), i = 1, \dots, n \quad (8)$$

where δ_i^0 is the activating factor following Remark 4;

The input grey sequence is then computed as

$$y^{(0)} = \{y^{(0)}(t_1), y^{(0)}(t_2), \dots, y^{(0)}(t_n)\} > 0; n \geq 4 \quad (9)$$

where $y^{(0)}(t_i) = y_{raw}(t_i) + c_1 + c_2$;

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

$$\begin{aligned} y^{(1)}(t_k) &= \sum_{i=1}^k y^{(0)}(t_i) \times \Delta t_i, k = 1, \dots, n \\ &= \sum_{i=1}^k (y_{obj}(t_i) + c_1 + c_2) \times \Delta t_i; \end{aligned} \quad (10)$$

Step 3: Define a background series $z^{(1)}$ as

$$z^{(1)}(t_k) = w^1(t_k) y^{(1)}(t_k) + w^2(t_k) y^{(1)}(t_{k-1}); k = 2, \dots, n \quad (11)$$

$$\begin{cases} w^1(t_k) = 0.5(1 - \delta^1) + \delta^1 \delta^2 w^4(t_k) + \delta^1(1 - \delta^2)(1 - w^4(t_k)) \\ w^2(t_k) = 0.5(1 - \delta^1) + \delta^1(1 - \delta^2) w^4(t_k) + \delta^1 \delta^2(1 - w^4(t_k)) \end{cases} \quad (12)$$

where $0 < w^4(t_k) < 1$ is the adaptive factor; $\{\delta^1, \delta^2\}$ is the set of activated factors and given as

$$\{\delta^1, \delta^2\} = \begin{cases} \{1, 0\}, \text{ IF } : y_{raw}^{(0)}(t_k) = SP(t_k); y_{raw}^{(0)}(t_{k-1}) = y_{obj}(t_{k-1}) \\ \{1, 1\}, \text{ IF } : y_{raw}^{(0)}(t_k) = y_{obj}(t_k); y_{raw}^{(0)}(t_{k-1}) = SP(t_{k-1}) \\ \{0, 1\}, \text{ Others; } \end{cases} \quad (13)$$

Step 4: Establish the grey differential equation:

$$y^{(0)}(t_k) + az^{(1)}(t_k) = b \quad (14)$$

By employing the least square estimation [25], one has:

$$\hat{\beta}_{ab} = [\hat{a} \quad \hat{b}]^T = (B^T B)^{-1} B^T Y \quad (15)$$

$$B = \begin{bmatrix} -z^{(1)}(t_2) & 1 \\ \vdots & \vdots \\ -z^{(1)}(t_n) & 1 \end{bmatrix}, Y = \begin{bmatrix} y^{(0)}(t_2) \\ \vdots \\ y^{(0)}(t_n) \end{bmatrix};$$

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

$$\hat{y}^{(0)}(t_k) = \frac{\hat{b} - \hat{a}(w^1(t_k) + w^2(t_k)) y^{(1)}(t_{k-1})}{1 + w^1(t_k) \hat{a} \Delta t_k}; \quad (16)$$

Step 6: Produce the s_k -step-ahead prediction y_{obj} (at step $(k+s_k)^{th}$) based on (9) and (16):

$$\hat{y}_{obj}^{(0)}(k+s_k) \equiv \hat{y}_{raw}^{(0)}(t_{k+s_k}) = \hat{y}^{(0)}(t_{k+s_k}) - c_1 - c_2. \quad (17)$$

Step 7: If prediction is continued, check the prediction accuracy

at the coming step to optimize the factor $w^A(t_{k+1})$ and return to Step 1; Otherwise, terminate the procedure.

B. Prediction Stability Analysis

The prediction using Algorithm 1 can be considered as a closed-loop tracking control problem in which the ‘control input’ $w^A(t_k)$ needs to be designed to ensure the ‘system response’ $\hat{y}_{raw}^{(0)}(k+s_k)$ follows the ‘desired target’ $y_{raw}(k+s_k)$. Here, the PI-type neural network (PINN) controller with Lyapunov stability condition [26] is implemented to regulate the ‘control input’ $w^A(t_k)$ for the robust prediction.

The PINN controller consists of three layers: an input layer as the prediction error sequence $\{e_i^P(t_k)\}$ with $e_i^P(t_k) = y_{raw}^{(0)}(t_{k-n+i}) - \hat{y}_{raw}^{(0)}(t_{k-n+i})$, $i = 1, \dots, n$, a hidden layer with two nodes P and I following PI algorithm, and an output layer to compute the adaptive factor $w^A(t_k)$ in (12). Define $\{w_i^P(t_k), w_i^I(t_k)\}$ and $\{w^P(t_k), w^I(t_k)\}$ are in turn the weight vectors of the hidden layer with respect to input i^{th} and the output layer. Then by using sigmoid activation function to ensure $0 < w^A(t_k) < 1$, the output from each hidden node is derived in (18) while the network output is derived in (19):

$$\begin{cases} O^P(t_k) = \sum_{i=1}^n w_i^P(t_k) e_i^P(t_k) : \text{Node P} \\ O^I(t_k) = O^I(t_k - 1) + \sum_{i=1}^n w_i^I(t_k) e_i^P(t_k) : \text{Node I} \end{cases} \quad (18)$$

$$w^A(t_k) = \left(1 + e^{-O^{NN}(t_k)}\right)^{-1}, \quad (19)$$

$$O^{NN}(t_k) = w^P(t_k) O^P(t_k) + w^I(t_k) O^I(t_k).$$

Next, a prediction error function is defined as

$$E^P(t_k) = 0.5 \sum_{i=1}^n \left(e_i^P(t_k)\right)^2. \quad (20)$$

By employing the back-propagation algorithm, the weight factors of the PINN controller are online regulated as

$$\begin{cases} w^{P/I}(t_k + 1) = w^{P/I}(t_k) - \eta_P(t_k) \partial E^P(t_k) / \partial w^{P/I}(t_k) \\ w_i^{P/I}(t_k + 1) = w_i^{P/I}(t_k) - \eta_P(t_k) \partial E^P(t_k) / \partial w_i^{P/I}(t_k) \end{cases} \quad (21)$$

where $\eta_P(t_k)$ is the learning rate within range (0,1]; the other factors in (21) are derived using the chain rule.

Theorem 1: By selecting the learning rate $\eta_P(t_k)$ to satisfy (22), the stability of the PIAGM(1,1) prediction is guaranteed.

$$\sum_{i=1}^{n-1} \left(e_i^P(t_k) F(t_k) + 0.5 F^2(t_k) \eta_P(t_k)\right) \leq 0 \quad (22)$$

$$F(t_k) = - \sum_{j=1}^n \left(\frac{e_j^P(t_k)}{w_j^P(t_k)} \frac{\partial E^P(t_k)}{\partial w_j^P(t_k)} + \frac{e_j^P(t_k)}{w_j^I(t_k)} \frac{\partial E^P(t_k)}{\partial w_j^I(t_k)} \right).$$

Proof: Define a Lyapunov function as (23). The theorem can be then proofed using the method presented in [26].

$$V^P(t_k) = 0.5 \sum_{i=1}^n \left(e_i^P(t_k)\right)^2. \quad (23)$$

V. PMFAC CONTROLLER

A. PMFAC Design

Remark 5: For the networked system (1) with problems defined using Definition 1, the PMFAC is proposed in which

the PIAGM_{1,2}(1,1), the sampling rates (7) and the NPD are utilized to produce the control input. The last plant information received at the controller side is defined as $y^*(k)$:

Case 1 - there is only a delay $\tau^{sc}(k)$ and no packet dropout in the backward channel at the current step, one has:

$$\begin{cases} y^*(k) = y(k - \Delta t_k) \\ \Delta t_k = \tau^{sc}(k) \end{cases} \quad (24)$$

Case 2 - there exist i continuous packet dropouts in the backward channel up to the current step, one has:

$$\begin{cases} y^*(k) = \hat{y}^{PMFAC}(y(k - \Delta t_k)) + \hat{y}^{ND}(k) \\ \Delta t_k = \hat{\tau}^{sc}(k) + \sum_{j=k-i}^k T^C(k) \end{cases} \quad (25)$$

where $\hat{y}^{PMFAC}(y(k - \Delta t_k))$ is the estimated system response at present using (2) based on the last actual system response $y(k - \Delta t_k)$; $\hat{y}^{ND}(k)$ is the estimated system disturbances using PIAGM₂(1,1) (Section III.B).

Based on Remark 1 and Remark 5, both the current network problems and system disturbances have been taken in to account. The control problem is now simplified to design a set of control inputs $\{U(k + \tau_i)\}$ using Algorithm 2 for future steps $\{(k + \tau_i)\}$ in which τ_i is defined as

$$\tau_i = \sum_{j=1}^i T^A(k + j), 2 \leq i \leq s_k^*, s_k^* \equiv b_{OSB} \quad (26)$$

Algorithm 2 Control Input Calculation Using PMFAC

Step 1: Update the current system response using (24) or (25).

Step 2: Compute the control input for step $(k + \tau_1)^{\text{th}}$:

$$\Delta U(k) = G(k) \left(y^r(k + \tau_1) - y^*(k) \right) \quad (27)$$

Step 3: Using (3) to (5), compute the set of control inputs $\{U(k + \tau_i)\}$, $i = 2, \dots, s_k^*$ using iterative method ($\tau_0 = 0$):

$$\Delta y(k + \tau_i) = \hat{\Phi}^T(k) \Delta U(k + \tau_{i-1}) \quad (28)$$

$$\Delta U(k + \tau_i) = G(k) \left(y^r(k + \tau_{i+1}) - y(k + \tau_i) \right) \quad (29)$$

$$U(k + \tau_{i+1}) = U(k + \tau_i) + \Delta U(k + \tau_i) \quad (30)$$

B. Control Stability Analysis

Lemma 1 [30]: Consider the following scalar linear system:

$$\begin{aligned} x(k+1) &= x(k) - \alpha(k)x(k - \tau_k), 0 \leq \tau_k \leq \bar{\tau} \\ x(k) &= \psi(k), k = -\bar{\tau}, -\bar{\tau} + 1, \dots, 0 \end{aligned} \quad (31)$$

where $x(k)$ is the scalar state, $\alpha(k)$ is the time-varying parameter, and $\psi(k)$ is the initial condition.

System (31) is asymptotically stable if $0 < \alpha(k) < 2(2\bar{\tau} + 1)^{-1}$.

Proof: Selecting the Lyapunov function as [30]

$$V(k) = V_1(k) + V_2(k) \quad (32)$$

$$V_1(k) = px(k)^2; V_2(k) = \bar{\tau} \sum_{i=-\bar{\tau}}^{k-1} \sum_{j=k+i}^{k-1} q\eta_x^2(j)$$

where p and q are positive scalars, $\eta_x^2(k) = -\alpha(k)x(k - \tau_k)$.

By taking the change of this Lyapunov function and defined $X(k) = [x(k) \ x(k - \tau_k)]^T$, bellow relation is obtained [30]:

$$\Delta V(k) = V(k+1) - V(k) = X^T(k) \Psi X(k) \quad (33)$$

$$\Psi = \begin{bmatrix} -q & -p\alpha(k) + q \\ -p\alpha(k) + q & (p + q\bar{\tau}^2)\alpha^2(k) - q \end{bmatrix}$$

The system is asymptotically stable once $\Delta V(k) < 0$. This lead to the inequalities in (33), $\Psi < 0$. By applying the Schur complement lemma [36], following condition is achieved:

$$0 < \alpha(k) < 2pq(pq + p^2 + q^2\bar{\tau}^2)^{-1} \leq 2(1 + 2\bar{\tau})^{-1} \quad (34)$$

The requirement (34) is to ensure the stability of system (31), therefore, completes the proof.

From (26), it is clear that τ_i is definitely bounded for each working step, $\tau_1 \leq \tau_i \leq \tau_{s_k}^*$. The following is to carry out the stability analysis for the system at step $k^* = (k + \tau_{s_k}^*)$.

Theorem 2: By selecting the weight factor $\lambda \geq (2\tau_{s_k}^* + 1)^2 \rho^2 b^2 / 16$, the networked control of system (1) is guaranteed to be stable with zero steady state tracking error for a constant reference of $y, y^r = \text{const}$.

Proof: Applying the PMFAC for the networked control of system (1), the tracking error with output y is derived as

$$e(k^*) \equiv y^r - y(k^*) = y^r - y(k^* - 1) - \hat{\Phi}^T(k) \Delta U(k^* - 1). \quad (35)$$

Replacing (29) into (35):

$$e(k^*) = e(k^* - 1) - \hat{\Phi}^T(k) G(k) e(k^* - 1). \quad (36)$$

From (5) and (36), one has:

$$e(k^*) = e(k) \left(1 - \rho \frac{\|\hat{\Phi}(k)\|^2}{\lambda + \|\hat{\Phi}(k)\|^2} \right)^{s_k^*}. \quad (37)$$

Taking the change of control error, one has:

$$\Delta e(k^*) = -\Delta y(k^* + 1) = -\Phi^T(k) \Delta U(k^*); \quad (38)$$

or

$$\Delta e(k^*) = -e(k) \left(1 - \rho \frac{\|\hat{\Phi}(k)\|^2}{\lambda + \|\hat{\Phi}(k)\|^2} \right)^{s_k^*} \Phi^T(k) G(k). \quad (39)$$

Comparing (39) with Lemma 1, the system is stable with zero steady state tracking error only if:

$$0 < \left(1 - \rho \frac{\hat{\Phi}^T(k) \hat{\Phi}(k)}{\lambda + \|\hat{\Phi}(k)\|^2} \right)^{s_k^*} \left(\rho \frac{\Phi^T(k) \hat{\Phi}(k)}{\lambda + \|\hat{\Phi}(k)\|^2} \right) < \frac{2}{2\tau_{s_k}^* + 1}. \quad (40)$$

Based on the reset algorithm (4), without loss of generality, $\Phi(k)$ is assumed to be positive for all the time. It is clear that:

$$\left(1 - \rho \frac{\hat{\Phi}^T(k) \hat{\Phi}(k)}{\lambda + \|\hat{\Phi}(k)\|^2} \right)^{s_k^*} < 1, \text{ and } \Phi^T(k) \hat{\Phi}(k) > 0. \quad (41)$$

From Assumption 1 and (2), following relation is obtained:

$$0 < \rho \frac{\Phi^T(k) \hat{\Phi}(k)}{\lambda + \|\hat{\Phi}(k)\|^2} \leq \frac{\rho b}{\lambda \|\hat{\Phi}(k)\|^{-1} + \|\hat{\Phi}(k)\|} \leq \frac{\rho b}{2\sqrt{\lambda}} \quad (42)$$

From (40) to (42), to ensure the system stability, one has:

$$\frac{\rho b}{2\sqrt{\lambda}} \leq \frac{2}{2\tau_{s_k}^* + 1} \leftrightarrow \lambda \geq \frac{\rho^2 b^2 (2\tau_{s_k}^* + 1)^2}{16} \quad (43)$$

The relation (43), therefore, completes the proof.

Remark 6: It is obviously that the weight factor is able to be optimized online for each s_k^* -step-ahead control action, $\lambda^{opt}(k) = \rho^2 b^2 (2\tau_{s_k}^* + 1)^2 / 16$, using $\tau_{s_k}^*$ (26) and Theorem 2.

VI. OSB BUFFER

A. Adaptive Grey Fuzzy Cognitive Map

Fuzzy cognitive map (FCM) is the neuro-fuzzy-based decision making tool graphically represented by a frame of nodes (input and output concepts, C) and connection edges between nodes (or relation, w) [37]. Combining a FCM and grey numbers can perform an effective decision making tool for solving problems within environments with high uncertainties and/or incomplete information [38]. However, the use of grey numbers with fixed bounds could lead to the low convergence speed and, therefore, wrong or inappropriate decisions could be made. To address these issues, this study develops a novel adaptive grey fuzzy cognitive map AGFCM in which the grey weights with dynamic bounds.

AGFCM Design

The AGFCM is generally designed for a set of N^c concepts. Hence, the AGFCM can be represented by

$$\{\otimes C_i(t), \otimes w_{ij}(t), f^C\} \quad (44)$$

where $\otimes C_i(t)$ is the grey value of i^{th} concept (45); $\otimes w_{ij}(t)$ is the grey weight between concept i^{th} and j^{th} (46); $f^C(*)$ is the activation function (47) (ζ is the steepness parameter)

$$\otimes C_i(t) \in [\underline{C}_i(t), \bar{C}_i(t)] \in ((0,1) \text{ or } (-1,1)), \quad (45)$$

$$\otimes w_{ij}(t) \in [\underline{w}_{ij}(t), \bar{w}_{ij}(t)] \in ((0,1) \text{ or } (-1,1)), \quad (46)$$

$$f^C(*) = \begin{cases} (1 + e^{-\zeta(*)})^{-1}, & \text{if } (*) \in (0,1), \\ (e^{\zeta(*)} - e^{-\zeta(*)})(e^{\zeta(*)} + e^{-\zeta(*)})^{-1}, & \text{if } (*) \in (-1,1). \end{cases} \quad (47)$$

Remark 7: Bounds of grey numbers in the AGFCM (concepts' values and weights) are properly initialized within maximum range (0,1) or (-1,1) and then, are automatically adjusted online to minimize a pre-defined cost function.

By using FCM theory and Remark 7, the grey value of each concept can be updated for each step of time based on the influences of the other interconnected concepts:

$$\begin{aligned}\otimes C_i(t+1) &= f^C \left(\otimes C_i(t) + \sum_{j=1, j \neq i}^N \otimes w_{ji}(t) \otimes C_j(t) \right) \\ &= \text{sat} \left(C_i(t+1), \bar{C}_i(t+1) \right) \in ((0,1) \text{ or } (-1,1))\end{aligned}\quad (48)$$

where $\text{sat}(\cdot)$ is the saturation function of \cdot .

AGFCM Training Algorithm

To enrich the AGFCM accuracy when dealing with partial unknown systems and uncertainties, the grey weights are updated online using the nonlinear Hebbian-type rule [37]:

$$\begin{aligned}\otimes w_{ji}(t+1) &= \otimes w_{ji}(t) + \Delta \otimes w_{ji}(t) \\ &= \otimes w_{ji}(t) + \eta_{AFGCM} \otimes C_j(t) \times \\ &\quad (\otimes C_i(t) - \otimes w_{ji}(t) \otimes C_j(t)) \\ &= \text{sat} \left(\otimes w_{ji}(t+1), \bar{\otimes w}_{ji}(t+1) \right) \in ((0,1) \text{ or } (-1,1))\end{aligned}\quad (49)$$

Remark 8: The AGFCM grey weights are updated until one of two following termination conditions are achieved:

- Cost function minimization condition:

$$J = \sum_{i=1}^m \left(C_{d_i}^{des} - C_i^{des} \right)^2 \leq J_{\min}, J_{\min} > 0 \quad (50)$$

- Minimum changing speed of decisive concepts' values:

$$\left| C_i^{des}(t+1) - C_i^{des}(t) \right| \leq \delta_C, \forall i = 1, \dots, m, \delta_C > 0 \quad (51)$$

where $C_{d_i}^{des}$ is the desired value of decisive concept i^{th} , C_i^{des} ; m is the number of decisive concepts; J_{\min} is the desired cost; δ_C is the small constant.

B. OSB Buffer Size Regulation Using AGFCM

To optimize online the OSB buffer size b_{OSB} (or s_k^*), the AGFCM is structured with four concepts within $(0, 1)$: three input concepts, and one output concept, α_k^s , which decides the increment of the buffer size compared to s_k estimated by the PIAGM₁(1,1). The three input concepts are selected as the current root mean square error (RMSE) of the prediction, the current package dropout ratio on the forward channel and the RSME of system tracking. Then, the OSB buffer size is given:

$$b_{OSB}(k+1) = s_k + 0.1 \text{round}(10\alpha_k^s)\Delta s \quad (52)$$

where Δs is the maximum allowable increase of buffer size.

Algorithm 3 OSB-based Control Procedure

Step 1: Set the initial size of the OSB, $b_{OSB}(0) = b_{\min} = 2$;

Step 2: For step $(k+1)^{\text{th}}$ at the controller side, based on the PIAGM₁(1,1) and AGFCM outputs, compute $b_{OSB}(k+1)$ using (52) and, correspondingly, generate a set of time-stamped control inputs, $\{U(k+t_1), \dots, U(k+t_{s_k}^*)\}$, using the

PMFAC;

Step 3: For step $(k+1)^{\text{th}}$ at the plant side:

- If a new package sent from the controller is arrived:
 - + Re-size the OSB with the new size $b_{OSB}(k+1)$
 - + Update the new control inputs and store in the OSB
- Following the given sampling period vector, $\{T_A\}$, (derived from PIAGM₁(1,1) and (7)), extract data from the OSB to drive the actuators based on the first-in-first-out principle.

VII. CASE STUDY

A. Problem and Hardware Setup

Here, the same networked DC servomotor system using Zigbee protocol developed throughout the previous studies [25], [26] was used to conduct a comparative study on the motor speed tracking using different advanced control methods.

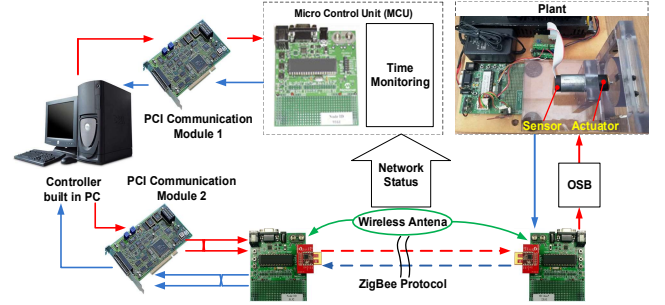


Fig. 3. Configuration of the networked servomotor control system.

TABLE I
CHARACTERISTICS OF THE COMPARATIVE CONTROLLERS

Characteristic	MFAC [30]	IMPC [22]	RPC [26]	DPHDC
NPD-based estimation	N/A	N/A	Yes	Yes
Buffer at plant side	Fixed size designed for worst case (max delay)	N/A	Buffer size is based on $\hat{p}^{ca}(k+s_k)$ only	Buffer size is based on $\hat{p}^{ca}(k+s_k)$ and AGFCM
Model-based	Not required	Yes	Yes	Not required
Model-free	Yes, control gain is designed for worst case	N/A	N/A	Yes, control gain is optimized online
Controller event	Event driven	Fixed for worst case	Hybrid driven scheme	Hybrid driven scheme
Actuator event	Time driven using buffer	Event driven	Hybrid driven scheme	Hybrid driven scheme
Sensor event	Time driven using buffer	Fixed for worst case	Hybrid driven scheme	Hybrid driven scheme

Here, the robust predictive control (RPC) [26], the improved model predictive control (IMPC) [22], the typical MFAC with fixed weight factor λ [30] combined with a fixed buffer (tagged as MFAC), and the proposed DPHDC. The characteristics of these controllers are summarized in Table I. Configuration of the system is presented in Fig. 3 and clearly described in [25], [26]. The system plant can be expressed as

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

with $d(k)$ is the unknown time-variant disturbance; and

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

To tackle the tracking control with computation delays and disturbances, these were added to the system during the experiments. The computation delays were simulated using the real machine computation delays in [25] while two disturbance sources [26] were invoked. The first source was the time-variant magnetic load applied to the motor shaft while the second source was added to the feedback speed as:

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

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.

For the control comparison, the real-time network problems were detected and recorded through an experiment with a selected controller. This problem source was then re-used to generate the same working conditions for the other controllers.

B. Setting of Controller Parameters

Using Definition 1, the delay thresholds were selected as: $\bar{\tau}^{com} = 0.025s$, $\bar{\tau}^{ca} = 0.035s$ and $\bar{\tau}^{sc} = 0.035s$. The RPC and IMPC were constructed based on the linearized system model. The state feedback and integral control gains of the RPC were derived for the set of the total delay regions ranging from 0 to 0.095 with interval of 0.01s [31]. Meanwhile, its observer gain was derived as [1; 1.3217]. For the IMPC, based on its design [22], the sampling rate was selected as 0.1s to cover the total delay. The prediction horizon P and control horizon M were in turn selected in this case as 5 and 3, their corresponding weighting matrices was chosen as, $Q = \text{diag}(0.4, 0, 0, 1)$ and $R = \text{diag}(0.01, 0.01, 0.01)$. The MFAC and DPHDC could be constructed without using the linearized plant model. For the MFAC, the buffer size was fixed as 5. As described in Section II and [30], the control parameters were set as

$\eta = 1, \rho = 1, \mu = 1, \hat{\Phi}(0) \equiv \hat{\phi}(0) = 2, \varepsilon = 10^{-5}$, and the weight factor was set to $\lambda = 20.5$ corresponding to the worst network state. For the DPHDC, the control parameters were set as $\eta = 1, \rho = 1, \mu = 1, \hat{\Phi}(0) = 2, \varepsilon = 10^{-5}, \lambda(0) = 6.5, \zeta = 1, \Delta s = 10$.

C. Experiments and Discussions

The real-time speed tracking control experiments in the imperfect conditions have been then carried out 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.

The first experiment series were done to track a constant speed at 10rad/s. The results were then obtained in Fig. 4. During the first 15 seconds with small perturbations, both the controllers could drive the motor speed to follow the desired level well (steady-state-error (SSE) within 2%). With the MFAC, once packet dropouts in the backward channel were detected, the system response could be ideally estimated using (2) and the iteration algorithm. Thus, the vector of five-step-ahead control inputs was always generated and sent to the buffer to drive the motor. However, the MFAC could only ensure the system converge to the disturbed speed level around the desired set point because of lacking disturbance compensation.

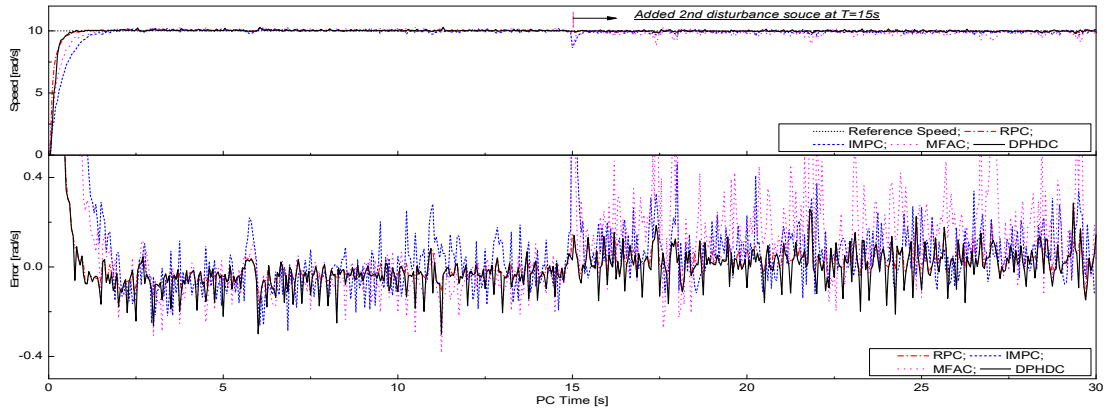


Fig. 4. Step tracking: performances comparison.

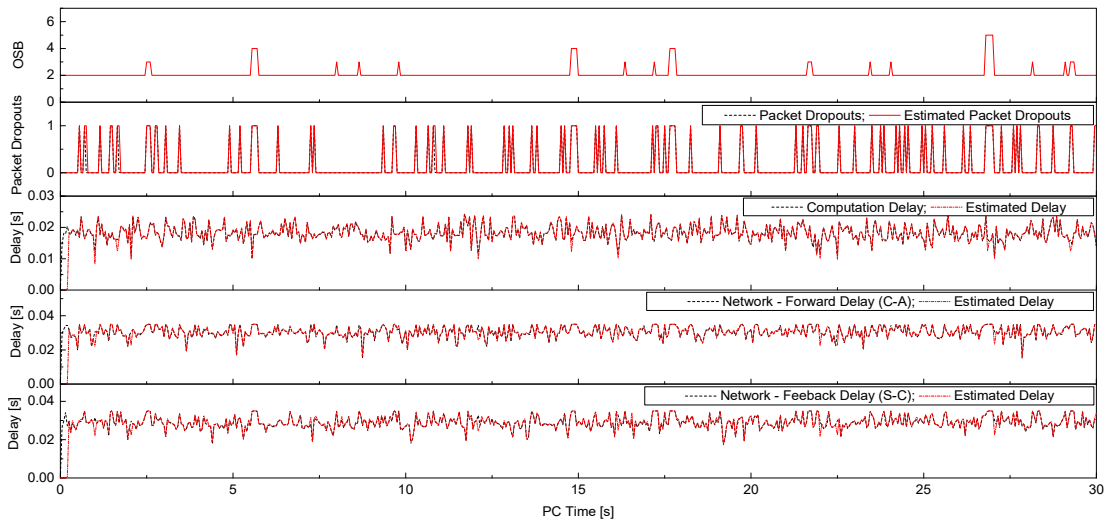


Fig. 5. Step tracking: one-step-ahead prediction of the delays and forward packet dropouts and OSB size of the DPHDC.

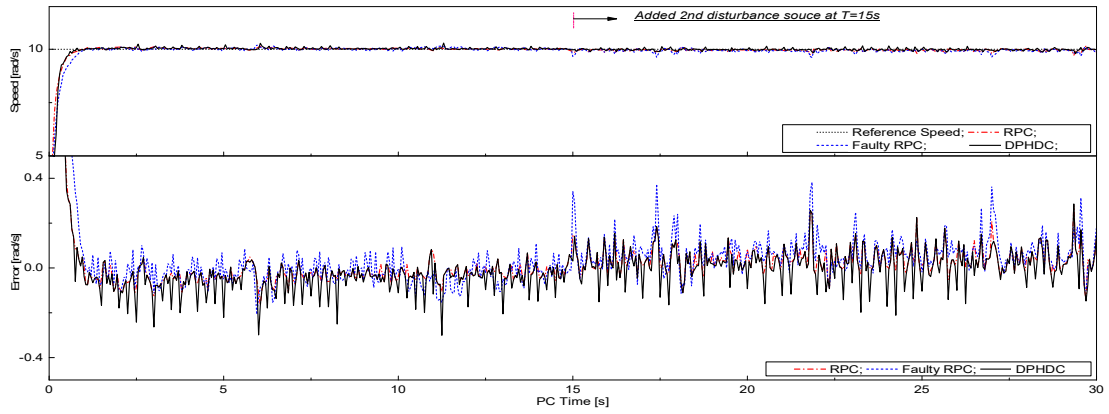


Fig. 6. Step tracking: a comparison with the faulty RPC.

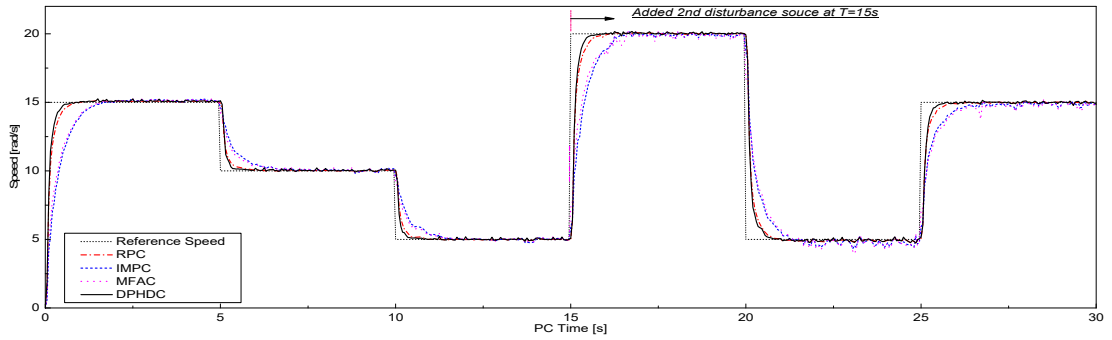


Fig. 7. Multi-step tracking: performance comparison.

Furthermore, due to use of fixed buffer size and fix weight factor, this controller could not adapt quickly to the system changes. Therefore once the disturbance level increased during the last 15 seconds, the tracking performance significantly deteriorated. With the IMPC, the system adaptation was slower than the others because the fixed large sampling period was used to cover the total system delays. Nevertheless, the steady response over the test was better than that of the MFAC due to the integration of speed tracking error minimization in conducting the optimal control signal. The most precise performances were achieved by the last two control options. This came as no surprise because these controllers possess the unique characteristics of the grey models, adaptive buffers, hybrid driven mechanism and adaptive control gains based on the network problem detection and disturbance rejection mechanisms. Due to the same use of prediction technique, the one-step-ahead delay prediction with high accuracy could be observed by either the RPC or the DPHDC scheme as plotted in sub-plots 1 to 4 from bottom of Fig. 5. The packet dropouts' plot shows the detection/estimation results on both the channels) while the OSB size of the DPHDC was observed as plotted in the top sub-plot of Fig. 5. The tracking results of these two methods were almost similar. Nonetheless in the RPC, the buffer with the tight selection of buffer size based on the package dropout prediction only could lead to small delays and even small increases of SSE in system response if the network problem was not properly estimated. This could be clearly demonstrated through Fig. 6. Here, the RPC was tested in which a random faulty signal was added to the network problem estimation of this controller. The tracking result was then

compared against the performances of the RPC and DPHDC. In addition, the RPC required the system knowledge well represented in the form of linearized model and the observer gains needed to be off-line calculated to cover the delay range. These could hinder the applicability of this method. On the contrary, the DPHDC is capable of producing the robust control actions without the system knowledge. With the proposed DPHDC, the buffer size was optimally adjusted based on both the prediction and tracking performances and the network status. Thus, the continuous optimal control inputs to drive the motor could be ensured. The tracking performances using the compared controllers were then summarized in Table II.

Next, to confirm the capability of the proposed approach over the other ones, the second experiment series were done for a multi-step speed tracking control profile under the same perturbed environment. The tracking results were obtained as plotted in Fig. 7. Similar to the first test, the slowest adaptation was observed by the IMPC while the lowest stability was obtained by the MFAC. In some regions, the IMPC and MFAC tracking results were quite similar as they were both designed for the worst network conditions (Table I) and, therefore, the system adaptation was slowly achieved. The RPC and DPHDC showed their superiority performances over the IMPC and MFAC. As numerically shown in Table II, although the RPC could bring to the smallest SSE and mean square error, the smooth response and quick adaptation were only guaranteed by the DPHDC. Both of these characteristics and the high applicability make the proposed control scheme to be the best solution for the considered NCS.

TABLE II
COMPARISON OF NCS PERFORMANCES USING DIFFERENT CONTROLLERS

Controller	Step responses			
	Percent of Overshoot [%]	Settling Time [s]	SSE [%] [15~30s]	Mean Square Error [rad/s] ² 10 ⁻² 15~30s]
RPC	0.47	1.74	1.63	1.14
IMPC	0.25	1.53	3.56	6.48
MFAC	0.38	1.36	8.44	21.88
DPHDC	0.39	0.61	1.87	1.83

VIII. CONCLUSIONS

In this paper, the data-based predictive hybrid driven control approach has been successfully introduced for a class of NCSs under random delays, packet dropouts and disturbances. The combination of the advanced modules, including the PIAGM(1,1), the hybrid driven mechanism, the AGFCM-based OSB buffer and the PMFAC, could compensate well all the system issues without requiring the plant model. The robust tracking performance is theoretically proved and guaranteed by tuning online the control inputs via the weight factor and buffer size regulation. The applicability and capability of the proposed DPHDC have been validated through real-time experiments on the networked servomotor system with difference control methods.

APPENDIX A

As Remark 4, for an object with a data sequence $Y_{obj} = \{y_{obj}(t_{O1}), y_{obj}(t_{O2}), \dots, y_{obj}(t_{Om})\}$, the raw input grey sequence Y_{raw} is derived as (8). From (8), define $\Delta t_{Ok} = t_{Ok} - t_{Ok-1}$; $k = 2, \dots, m$, the following two cases are considered:

- Case 1 – the object has equal time-interval signal, or unequal time-interval signal satisfying a time sequence checking in (A1), each data point of Y_{raw} is obtained as (A2)

$$\frac{\Delta t_{Ok}}{\Delta T} \leq 1.1; \Delta t_{Ok} = t_{Ok} - t_{O(k-1)}; \forall k \in [2, \dots, m] \quad (A1)$$

here ΔT is the desired sample time.

$$y_{raw}^{(0)}(t_i) = y_{obj}(t_{Oi}); \delta_i^0 = 1; i = 1, \dots, n; n \equiv m. \quad (A2)$$

- Case 2 – the object having unequal time-interval signal which does not satisfy (A1), each data point of Y_{raw} is derived using the SP algorithm A3 ($\delta_i^0 = 0$)

$$\begin{aligned} y_{raw}^{(0)}(t_1) &= y_{Object}(t_1); t_i = t_{i-1} + \Delta T; \\ y_{raw}^{(0)}(t_i) &= Sa_k + Sb_k(t_i - t_{Ok})^1 + Sc_k(t_i - t_{Ok})^2 + Sd_k(t_i - t_{Ok})^3 \\ t_{Ok} \leq t_i \leq t_{O(k+1)}; i &= 2, \dots, n; n = \text{floor}\left(\frac{t_{Om} - t_1}{\Delta T}\right); k = 1, \dots, m-1 \end{aligned} \quad (A3)$$

where: $\{Sa_k, Sb_k, Sc_k, Sd_k\} (k = 1, \dots, m-1)$ is a $[4 \times m-1]$ coefficient matrix of the spline function going through the object data set $\{(t_1, y_{obj}(t_1)), \dots, (t_m, y_{obj}(t_m))\}$.

From [3], the coefficient matrix can be computed using the following S-procedure:

S-step 1: Obtain the first column of the coefficient matrix as

$$Sa_k = y_{obj}(t_{Ok}); k = 1, \dots, m-1. \quad (A4)$$

S-step 2: Generate a vectors h in size $[1 \times m-1]$

$$h_k \equiv h(k) = t_{O(k+1)} - t_{Ok}. \quad (A5)$$

S-step 3: Generate a vectors g in size $[1 \times m-2]$

$$\begin{aligned} g_k \equiv g(k) &= \frac{3}{h_{k+1}} \left(y_{obj}(t_{O(i+2)}) - y_{obj}(t_{O(k+1)}) \right) \\ &\quad - \frac{3}{h_k} \left(y_{obj}(t_{O(k+1)}) - y_{obj}(t_{Ok}) \right); k = 1, \dots, m-2. \end{aligned} \quad (A6)$$

S-step 4: Perform a tri-diagonal matrix algorithm to calculate the third column of the coefficient matrix as

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ va_1 & vb_1 & vc_1 & 0 & 0 \\ 0 & va_2 & vb_2 & vc_2 & 0 \\ 0 & 0 & va_{m-2} & vb_{m-1} & vc_{m-2} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Sc_1 \\ Sc_2 \\ Sc_3 \\ Sc_{m-1} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ g_1 \\ g_2 \\ g_{m-2} \\ 0 \end{bmatrix} \quad (A7)$$

here $va_k = h_k$; $vb_k = 2(h_k + h_{k+1})$; $vc_k = h_{k+1}$; $k = 1, \dots, m-2$.

By solving (A7) based on Gaussian elimination, the vector $\{Sc_k\}$ is then easily obtained ($Sc_m = 0$).

S-step 5: Calculate the second and fourth column

$$Sb_k = \frac{y_{obj}(t_{O(k+1)}) - y_{obj}(t_{Ok})}{h_k} - \frac{(2Sc_k + Sc_{k+1})h_k}{3} \quad (A8)$$

$$Sd_k = \frac{Sc_{k+1} - Sc_k}{3h_k}. \quad (A9)$$

REFERENCES

- [1] J. Qiu, S. Member, H. Gao, and S. X. Ding, "Recent Advances on Fuzzy-Model-Based Nonlinear Networked Control Systems: A Survey," *IEEE Trans. Ind. Electron.*, vol. 63, no. 2, pp. 1207–1217, 2016.
- [2] 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, 2013.
- [3] 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, 2013.
- [4] B. W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Control Syst.*, vol. 21, no. 1, pp. 84–99, 2001.
- [5] C. Huang, Y. Bai, and X. Liu, "H-Infinity State Feedback Control for a Class of Networked Cascade Control Systems With Uncertain Delay," *IEEE Trans. Ind. Informatics*, vol. 6, no. 1, pp. 62–72, 2010.
- [6] B. Yang, U. 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, 2011.
- [7] 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, 2012.
- [8] F. Yang, Z. Wang, Y. S. Hung, and M. Gani, "H ∞ control for networked control systems with random communication delays.pdf," *IEEE Trans. Automat. Contr.*, vol. 51, no. 3, pp. 511–518, 2006.
- [9] C. Choi and W. 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, 2012.
- [10] K. Kim, M. Sung, and H. Jin, "Design and Implementation of a Delay-Guaranteed Motor Drive for Precision Motion Control," *IEEE Trans. Ind. Informatics*, vol. 8, no. 2, pp. 351–365, 2012.
- [11] J. Yi, Q. Wang, D. Zhao, and J. T. Wen, "BP neural network prediction-

- based variable-period sampling approach for networked control systems," *Appl. Math. Comput.*, vol. 185, pp. 976–988, 2007.
- [12] C. Lai and P. Hsu, "Design the Remote Control System With the Time-Delay Estimator and the Adaptive Smith Predictor," *IEEE Trans. Ind. Informatics*, vol. 6, no. 1, pp. 73–80, 2010.
 - [13] D. López-echeverría and M. E. Magaña, "Variable Sampling Approach to Mitigate Instability in Networked Control Systems with Delays," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 23, no. 1, pp. 119–126, 2012.
 - [14] R. Yang, G. 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, 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. Informatics*, vol. 9, no. 1, pp. 417–426, 2013.
 - [16] Z. Pang, G. Liu, D. Zhou, S. Member, 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, 2014.
 - [17] P. Ignaciuk, "Nonlinear Inventory Control With Discrete Sliding Modes in Systems With Uncertain Delay," *IEEE Trans. Ind. Informatics*, vol. 10, no. 1, pp. 559–568, 2014.
 - [18] A. Baños, F. Perez, and J. Cervera, "Network-Based Reset Control Systems With Time-Varying Delays," *IEEE Trans. Ind. Informatics*, vol. 10, no. 1, pp. 514–522, 2014.
 - [19] H. Gao, T. Chen, and J. Lam, "A new delay system approach to network-based control," *Automatica*, vol. 44, pp. 39–52, 2007.
 - [20] H. Gao and T. Chen, "Network-Based H_∞ Output Tracking Control," *IEEE Trans. Automat. Contr.*, vol. 53, no. 3, pp. 655–667, 2008.
 - [21] C. Peng and Q. L. Han, "On Designing a Novel Self-Triggered Sampling Scheme for Networked Control Systems with Data Losses and Communication Delays," *IEEE Trans. Ind. Electron.*, vol. 63, no. 2, pp. 1239–1248, 2016.
 - [22] R. Lu, Y. Xu, and R. Zhang, "A New Design of Model Predictive Tracking Control for Networked Control System under Random Packet Loss and Uncertainties," *IEEE Trans. Ind. Electron.*, vol. PP, no. 99, pp. 6999–7007, 2016.
 - [23] 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, 2014.
 - [24] 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, 2014.
 - [25] 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, 2015.
 - [26] D. Q. Truong, K. K. Ahn, and J. Marco, "A Novel Robust Predictive Control System Over Imperfect Networks," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1751–1761, 2017.
 - [27] Z. Hou and Z. Wang, "From model-based control to data-driven control : Survey , classification and perspective," *Inf. Sci. (Ny).*, vol. 235, pp. 3–35, 2013.
 - [28] Z. Hou and X. Bu, "Model free adaptive control with data dropouts," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 10709–10717, 2011.
 - [29] X. Bu, F. Yu, Z. Hou, and H. Zhang, "Model-Free Adaptive Control Algorithm with Data Dropout Compensation," *Math. Probl. Eng.*, vol. 2012, pp. 1–14, 2012.
 - [30] Z. Pang, G. Liu, D. Zhou, and D. Sun, "Data-Based Predictive Control for Networked Nonlinear Systems With Network-Induced Delay and Packet Dropout," *IEEE Trans. Ind. Electron.*, vol. 63, no. 2, pp. 1249–1257, 2016.
 - [31] N. Vataniski, J. P. Georges, C. Aubrun, E. Rondeau, and S. L. Jämsä-Jounela, "Networked control with delay measurement and estimation," *Control Eng. Pract.*, vol. 17, no. 2, pp. 231–244, 2009.
 - [32] J. Jasperneite and K. Shehab, "Enhancements to the Time Synchronization Standard IEEE-1588 for a System of Cascaded Bridges," in *5th IEEE Int. Workshop on Factory Communication Systems (WFCS'04)*, 2004, pp. 239–244.
 - [33] X. Chen, D. Li, S. Wang, H. Tang, and C. Liu, "Frequency-Tracking Clock Servo for Time Synchronization in Networked Motion Control Systems," *IEEE Access*, vol. 5, pp. 11606–11614, 2017.
 - [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, 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, 2010.
 - [36] J. G. Vanantwerp and R. D. Braatz, "A tutorial on linear and bilinear matrix inequalities," *J. Process Control*, vol. 10, pp. 363–385, 2000.
 - [37] E. I. Papageorgiou and P. P. Groumpos, "A new hybrid method using evolutionary algorithms to train Fuzzy Cognitive Maps," *Appl. Soft Comput.*, vol. 5, no. 4, pp. 409–431, 2005.
 - [38] J. L. Salmeron, "Modelling grey uncertainty with fuzzy grey cognitive maps," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 7581–7588, 2010.



Truong Quang Dinh (M'18) received the B.S degree in the department of Mechanical Engineering from Hochiminh City University of Technology, Hochiminh, Vietnam, in 2001, and the Ph.D. degree in the School of Mechanical Engineering from University of Ulsan, Ulsan, Korea, in 2010.

He is currently an Assistant Professor at 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. He is a Member of IEEE, and IES.



James Marco received the Eng.D. degree 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.



Dave Greenwood received M.A. degree from the University of Cambridge, UK, in 1993, the M.Sc. degree from Chalmers University of Technology, Sweden, in 2008.

He joined WMG as a Professor in 2014 from Ricardo UK Ltd. He currently leads the Advanced Propulsion Systems team at Warwick Manufacturing Group (WMG), University of Warwick, Coventry, UK. His research interests include vehicle powertrain, energy storage, energy conversion and management. He is a Board Member at the Low Carbon Vehicle Partnership (LowCVP) and a member of the Automotive Council Technology Group. He is also a Member of the EPSRC's Energy Scientific Advisory Committee and the Advanced Propulsion Centre (APC) Advisory Board.



Kyoung Kwan Ahn (M'02) 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.



Jong Il Yoon received the Ph.D. from the University of Ulsan, Ulsan, Korea, in 2014.

He is currently a Principal Researcher at the Korea Construction Equipment Technology Institute (KOCETI), Gunsan, Korea. His research interests include fluid power systems, construction machines and energy saving technologies, systems engineering, systems design, modelling and control. He is a Member of SICE, KSME, KSPE, KROS, ICROS, KFPS and

JFPS.