A Data-Based Hybrid Driven Control for Networked-based Remote Control Applications

Truong Quang Dinh, James Marco, David Greenwood
Warwick Manufacturing Group (WMG)
University of Warwick
Coventry CV4 7 AL, UK
q.dinh@warwick.ac.uk, james.marco@warwick.ac.uk

Kyoung Kwan Ahn
School of Mechanical Engineering
University of Ulsan
Ulsan 680-749, Korea
kkahn@ulsan.ac.kr

Jong Il Yoon
KOCETI
KOCETI
Gunsan 573-540, Korea
jiyoon@koceti.re.kr

***Abstract*—This paper develops a data-based hybrid driven control (DHDC) approach for a class of networked nonlinear systems compromising delays, packet dropouts and disturbances. First, the delays and/or packet dropouts are detected and updated online using a network problem detector. Second, a single-variable first-order proportional-integral (PI) -based adaptive grey model is designed to predict in a near future the network problems. Third, a hybrid driven scheme integrated a small adaptive buffer is used to allow the system to operate without any interrupt due to the large delays or packet dropouts. Forth, a prediction-based model-free adaptive controller is developed to compensate for the network problems. Effectiveness of the proposed approach is demonstrated through a case study.**

Keywords—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 because of its advantages over wiring control. Time delay (computation delays at the controller and communication delays at the forward and backward channels) and packet dropout are two important and unavoidable factors existing in any networked control system (NCS) [1], [10].

To address stability of NCSs with communication delays, many studies were carried out in which the controller designs were depended on the assumptions, that the time delay was constant or bounded [2], had a probability distribution function [3], or based on time delay analyses [4]. In other studies, control concepts based on variable sampling periods using neural network or prediction theories were adopted for NCSs with large delays [5]. Although the favorable results were achieved, the observation of real delay data to train and construct the NCSs was not appropriately discussed.

For NCSs compromising both time delays and packet dropouts, many important methodologies, such as predictive control [6], adaptive control [7], hybrid control [8], and robust control [9],[10], were proposed in which delays and packet dropouts were assumed to be priorly known and bounded. Another trend is known as control based on efficient networks with compensation strategy [11]. However, their applicability may be limited due to the complex network designs. As a feasible solution for these problems, a robust variable sampling period control approach tagged RVSPC [12] has been proposed. Some limitations still exist in the RVSPC as: the state feedback control unit is basically designed for a well-defined networked linear system which is actually not easy to be achieved; for a NCS with large delays and/or high packet dropout ratio, the performance may be degraded when the driving commands and sensing signals are long delayed/dropped. Thanks to the development of model-free control, this advanced technology is successful utilized to construct NCSs with the good adaptability [13]. Nevertheless, these control schemes remain some open problems: the network problems are assumed to be bounded within a pre-defined maximum round trip delay; the large delay bound also leads to the slow convergence speed; the buffers need to be large enough to deal with the worst network states; the task must be given at the plant side while it is normally at the control side in actual remote applications.

This paper aims to address all the problems raised by [12], [13] by introducing a data-based hybrid driven control (DHDC) approach for a class of imperfect NSCs. The main contribution of this work can be expressed as:

1) Delays and packet dropouts are quickly and accurately detected online by a network problem detector (NPD) using a new concept to classify network problems.

2) A single-variable first-order PI-based adaptive grey model (PIAGM(1,1)) is more generally designed for a capable of forecasting precisely in a near future the network problems.

3) A hybrid driven scheme integrated a small adaptive buffer (SAB) is developed to operate the system smoothly. The buffer size is online optimized based on the estimated packet dropouts and subsequently, simplifies the control design.

4) A prediction-based model-free adaptive controller (PMFAC) is then constructed to compensate for the small delays and packet dropouts. By employing the hybrid driven scheme and SAB, a so-called s_k-step-ahead control technique is used to produce a dynamic vector of the control inputs according to the buffer size.

5) Lyapunov stability conditions are invoked in the control approach to guarantee the system stability.

II. Problem Description

An output y of a generic discrete-time nonlinear system with non-uniform sampling states can be described as:

$$y(k+1) = f(y(k),...,y(k-l_y),U(k),...,U(k-l_U)) \quad (1)$$

where k denotes the time stamp at working step k^{th}; $y(k) \in R^1$; $U(k) \in R^{m_u}$ is the control input vector; $f(*)$ is an unknown nonlinear function; l_y and l_u are unknown orders.

To build the free-model adaptive control (MFAC) for system (1), two following assumptions are made [13].

Assumption 1: the partial derivative of function $f(*)$ in (1) with respect to the control input $u(k)$ is continuous.

Assumption 2: there is a positive constant b such that system (1) is generalized Lipschitz, $\left|\Delta y(k+1)\right| \le b\left\|\Delta U(k)\right\|$, for any k, $\Delta y(k+1) = y(k+1) - y(k), \Delta U(k) = U(k) - U(k-1) \ne 0$.

Based on Assumptions 1 and 2, system (1) can be represented in a compacted form dynamic linearization [13]

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

The incremental control algorithm is derived for system (1) using the MFAC [13] as follows:

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

$$\hat{\phi}_i(k) = \hat{\phi}_i(0) \leftrightarrow \begin{cases} \left|\hat{\phi}_i(k)\right| \le \varepsilon, \text{ or } \left|\Delta u_i(k-1)\right| \le \varepsilon \\ sign(\hat{\phi}_i(k)) \ne sign(\hat{\phi}_i(0)) \end{cases} \quad (4)$$

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

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

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

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

- The communication delays in both forward and/or backward channels are denoted as τ^{ca} and τ^{sc}. Correspondingly, two threshold values, $\overline{\tau}^{ca}$ and $\overline{\tau}^{sc}$, are defined to classify small and large communication delays. Large delays are treated as packet dropouts:

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

- 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 time stamp k is denoted as $p^{ca}(k)$ or $p^{sc}(k)$.
- The computation delay, τ^{com}, is bounded, $\tau^{com}(k) \le \overline{\tau}^{com}$.

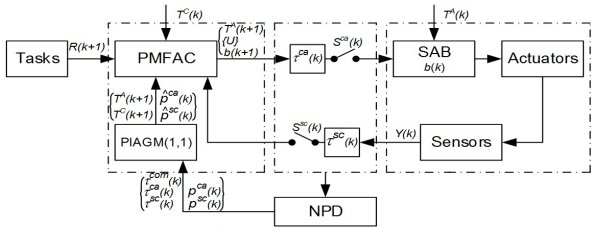

Fig. 1. A generic NCS architecture using the proposed DHDC approach.

III. Data-based Hybrid Driven Control Approach

A generic NCS using the DHDC is depicted in Fig. 1 in which the tasks consist of a main task to control the plant (1) and other tasks, which cause computation delays.

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

- The NPD is used to detect and classify the network problems using Definition 1.
- The PIAGM(1,1) with prediction step size s_k uses the NPD outputs to predict the network problems in a near future.
- Both the controller, sensors and actuators are hybrid time-event-driven. Event driven is activated once small delays are detected; otherwise, time driven is activated.
- The PMFAC is designed to compensate for small delays and packet dropouts. According to the s_k-step-ahead prediction, a vector of control inputs are derived and the SAB buffer at the plant side is used to drive the actuators.

A. NPD Detector

Remark 2: The hardware, PIC18F4620 MCU (from Microchip) equipped with the 4-MHz oscillator, and the embedded time stamp-based detection logics developed in [12] are used to build the NPD. Thus, the real-time measurement accuracy is $50\mu s$.

From Remark 2 and using 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

Remark 3: Grey model is known as a feasible tool for online prediction while requiring only a few historical data [12],[14]. The PIAGM(1,1) is designed as the followings:

- The model is developed a more efficient way using simple PI algorithm and neural network.
- It is capable of forecasting any random time-series data by using a simple data conversion with additive factors.
- The robust prediction is guaranteed by Lyapunov condition.

From Remark 1, the PIAGM(1,1) is employed to perform the s_k-step-ahead prediction of the network problems: 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,...,s_k, s_k is the smallest value corresponding to a case:

$$(\hat{p}^{ca}(k+2) = 0) \text{ or } (\hat{p}^{ca}(k+i_s-1) \ne 0) \,\&\, (\hat{p}^{ca}(k+i_s) = 0).$$

For a step $(k+1)^{th}$, using Definition 1 and PIAGM(1,1) outputs, the hybrid time-event driven method can be expressed through dynamic sampling rates, $T^C(k+1)$ and $T^A(k+1)$, which are derived for the controller side and plant side, respectively:

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

C. SAB Buffer

To eliminate packet dropout effect, the SAB is utilized at the plant side. The key concept here is to regulate online the buffer size, $b(k+1)$, to store enough commands to drive the actuators for $\hat{p}^{ca}(k+s_k)$ continuous packet dropouts.

Algorithm 1 SAB-based Control Procedure

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

Step 2: For step $(k+1)^{th}$ at the controller side, base on the s_k-step-ahead prediction using the PIAGM(1,1): set $b(k+1) = s_k$ and, correspondingly, generate a set of time-stamped control inputs, $\{U(k+t_1),...,U(k+t_{s_k})\}$, using the PMFAC;

Step 3: For 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 $b(k+1)$
  + Update the new control inputs and store in the SAB
- Extract data from the SAB to drive the actuators.

D. PMFAC Controller

As the main control unit, the PMFAC takes part in driving the actuators to reach the desired target based on Remark 1:

- For each step defined by the hybrid time-event driven method ($T^C(k)$) and based on the PIAGM(1,1) output, s_k, a set of the control inputs are generated for the next s_k steps using an iterative estimation scheme.
- A time delay factor is invoked into the controller design process to compensate for this problem.

IV. PIAGM(1,1) Predictor

A. PIAGM(1,1) Design

Remark 4: For any random data set of the predicted object, y_{obj}, it can be converted into a Grey sequence satisfying both the Grey checking conditions using the cubic spline interpolation (called as *SP*) and two non-negative smart additive factors, c_1 and c_2, derived by Theorem 1 in [12].

Prediction procedure for an object, y_{obj}, can be expressed as:

Algorithm 2 PIAGM(1,1) Model to Predict An Object y_{obj}

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

$$y^{(0)}_{raw_1}(t_i) = \delta_i^0 y_{obj}(t_i) + (1-\delta_i^0) SP(Y_{obj}, t_i), i = 1,...,n \quad (8)$$

where δ_i^0 is an activated factors based on Remark 4;

The input grey sequence is computed as

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

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

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

$$y^{(1)}(t_k) = \sum_{i=1}^{k} \left(y_{obj}(t_i) + c_1 + c_2\right) \times \Delta t_i, k = 1,...,n \quad (10)$$

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

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

where $\left\{w^1(t_k), w^2(t_k)\right\}$ is a set of weight factors computed as

$$\begin{cases} w^1(t_k) = 0.5\left(1-\delta^1\right) + \delta^1\delta^2 w^A(t_k) + \delta_1\left(1-\delta^2\right)\left(1-w^A(t_k)\right) \\ w^2(t_k) = 0.5\left(1-\delta^1\right) + \delta^1\left(1-\delta^2\right) w^A(t_k) + \delta^1\delta^2\left(1-w^A(t_k)\right) \end{cases} \quad (12)$$

where $0 < w^A(t_k) < 1$ is an adaptive factor; $\{\delta^1, \delta^2\}$ are:

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

Step 4: Establish the grey differential equation

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

By employing the least square estimation [12], one has:

$$\hat{\beta}_{ab} = \begin{bmatrix} \hat{a} & \hat{b} \end{bmatrix}^T = (B^T B)^{-1} B^T Y \quad (15)$$

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

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

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

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

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

B. Prediction Stability Analysis

A PI-based neural network (PINN) controller with three layers is designed to regulate the 'control input' $w^A(t_k)$. The input layer is the prediction error sequence $\{e_i^P(t_k)\}$ with $e_i^P(t_k) = y^{(0)}_{raw}(t_{k-n+i}) - \hat{y}^{(0)}_{raw}(t_{k-n+i})$. The hidden layer has two nodes P and I following PI algorithm and the output layer to compute the adaptive factor $w^A(t_k)$ in (12). Define $\{w_i^P(t_k), w_i^I(t_k)\}$ is a weight vector of the hidden nodes with respect to input i^{th} and $\{w^P(t_k), w^I(t_k)\}$ is a weight vector of the output layer. Since, the output from each hidden node is derived in (18) while the network output is derived in (19):

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

$$\begin{aligned} w^A(t_k) &= (1 + e^{-O^{NN}}(t_k))^{-1}, \\ O^{NN}(t_k) &= w^P(t_k) O^P(t_k) + w^I(t_k) O^I(t_k). \end{aligned} \quad (19)$$

A prediction error function is defined as

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

The weight factors of the PINN controller are online tuned using the back-propagation algorithm 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 partial derivative of the error function with respect to each weight.

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

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

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

Proof: Define a Lyapunov function as (23). The theorem can be then proofed by using the same method in [21].

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

V. PMFAC CONTROLLER

A. PMFAC Design

Remark 5: For the networked system (1) with problems defined using Definition 1, the PMFAC is built based on the PIAGM(1,1) prediction results to produce the control input. Thus, the last plant information received at the controller side can be described as $y^*(k) = y(k-\Delta t_k)$:

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

Case 2 – currently, there exist *i* continuous packet dropouts in the backward channel, $\Delta t_k = \bar{\tau}^{sc} + \sum_{j=k-i}^{k} T^A(j)$.

Based on Remarks 1 and 5, the control problem is now simplified to design a set of control inputs $\{U(k+\tau_i)\}$ (for Algorithm 1) at steps $\{(k+\tau_i)\}$ in which τ_i is defined as

$$\begin{aligned} \tau_1 &= (\Delta t + \hat{\tau}^{com}(k+1) + \hat{\tau}^{ca}(k+1)) \\ \tau_i &= \tau_1 + \sum_{j=2}^{i} T^A(k+i), 2 \le i \le s_k \end{aligned} \quad (28)$$

Algorithm 3 Control Input Calculation Using PMFAC

Step 1: Compute the control input for step $(k+\tau_1)^{\text{th}}$:

$$\Delta U(k) = G(k)(y^r(k+\tau_1) - y^*(k)) \quad (29)$$

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

$$\Delta y(k+\tau_i) = \hat{\Phi}^T(k)\Delta U(k+\tau_{i-1}) \quad (30)$$

$$\Delta U(k+\tau_i) = G(k)(y^r(k+\tau_{i+1}) - y(k+\tau_i)) \quad (31)$$

$$U(k+\tau_{i+1}) = U(k+\tau_i) + \Delta U(k+\tau_i) \quad (32)$$

B. Control Stability Analysis

Lemma 1 [25]: Consider the following linear system:

$$\begin{aligned} x(k+1) &= x(k) - \alpha(k)x(k-\tau_k), 0 \le \tau_k \le \bar{\tau} \\ x(k) &= \psi(k), k = -\bar{\tau}, -\bar{\tau}+1, ..., 0 \end{aligned} \quad (33)$$

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

System (32) is asymptotically stable if $0 < \alpha(k) < 2(2\bar{\tau}+1)^{-1}$.

Proof: Selecting the Lyapunov function as [25]

$$V(k) = V_1(k) + V_2(k) \quad (34)$$

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

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 $X(k) = [x(k) \quad x(k-\tau_k)]^T$, bellow relation is obtained [13]:

$$\Delta V(k) = V(k+1) - V(k) = X^T(k)\Psi X(k) \quad (35)$$

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

The system is asymptotically stable once $\Delta V(k) < 0$. This lead to the inequalities in (35), $\Psi < 0$. By applying the Schur complement lemma [12], following condition is achieved:

$$0 < \alpha(k) < 2pq\left(pq + p^2 + q^2\bar{\tau}^2\right)^{-1} \le 2\left(1+2\bar{\tau}\right)^{-1} \quad (36)$$

The requirement (36) to ensure the stability of system (33), therefore, completes the proof.

From (28), it is clear that τ_i is definitely bounded for each working step, $\tau_1 \le \tau_i \le \tau_{s_k}$. 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 $\lambda \ge \left(2\tau_{s_k}+1\right)^2\rho^2 b^2/16$, 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=const.

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

$$e(k^*) \equiv y^r - y(k^*) = y^r - y(k^*-1) - \hat{\Phi}^T(k)\Delta U(k^*-1). \quad (37)$$

Replacing (31) into (37) and using (5), one has:

$$e(k^*) = e(k^*-1)\left(1 - \rho\hat{\Phi}^T(k)\hat{\Phi}(k)(\lambda + \|\hat{\Phi}(k)\|^2)^{-1}\right); \text{ or} \quad (38)$$

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

Taking the change of control error, one has:

$$\Delta e(k^*) = -\Delta y(k^*+1) = -\Phi^T(k)\Delta U(k^*); \text{ or} \quad (40)$$

$$\Delta e(k^*) = -e(k)\left(1 - \rho\frac{\left\|\hat{\Phi}(k)\right\|^2}{\lambda + \left\|\hat{\Phi}(k)\right\|^2}\right)^{s_k}\Phi^T(k)G(k). \quad (41)$$

Comparing (41) with Lemma 1, the system is stable with zero steady state tracking error only if:

$$0<\left(1-\rho\frac{\hat{\Phi}^T(k)\hat{\Phi}(k)}{\lambda+\left\|\hat{\Phi}(k)\right\|^2}\right)^{s_k}\left(\rho\frac{\Phi^T(k)\hat{\Phi}(k)}{\lambda+\left\|\hat{\Phi}(k)\right\|^2}\right)<\frac{2}{2\tau_{s_k}+1}. \quad (42)$$

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:

$$\left(1-\rho\hat{\Phi}^T(k)\hat{\Phi}(k)\left(\lambda+\left\|\hat{\Phi}(k)\right\|^2\right)^{-1}\right)^{s_k}<1. \quad (43)$$

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

$$0<\rho\frac{\Phi^T(k)\hat{\Phi}(k)}{\lambda+\left\|\hat{\Phi}(k)\right\|^2}\le\frac{\rho b}{\lambda\left\|\hat{\Phi}(k)\right\|^{-1}+\left\|\hat{\Phi}(k)\right\|}\le\frac{\rho b}{2\sqrt{\lambda}} \quad (44)$$

From (42) to (44), to ensure the system stability, one has:

$$\rho b/(2\sqrt{\lambda})\le 2/(2\tau_{s_k}+1)\leftrightarrow\lambda\ge\rho^2b^2(2\tau_{s_k}+1)^2/16 \quad (45)$$

The relation (45), therefore, completes the proof.

VI. Case Study

A. Problem and Hardware Setup

Here, the network setup and DC servomotor system in [12] were used to carry out a comparative study on the motor speed tracking control (see Fig. 2). A comparative between the proposed DHDC approach and existing model-based and model-free methods has been considered. Two model-based methods were employed which were the robust variable sampling period controller (RVSPC) [12], and the static state feedback controller (SSFC) [10] combined with the NPD module and a fixed buffer, tagged as SSFC-FB. A model-free method - the MFAC [13] with fixed weight factor λ combined with a fixed buffer (named as MFAC-FB) was used to perform the comparison.

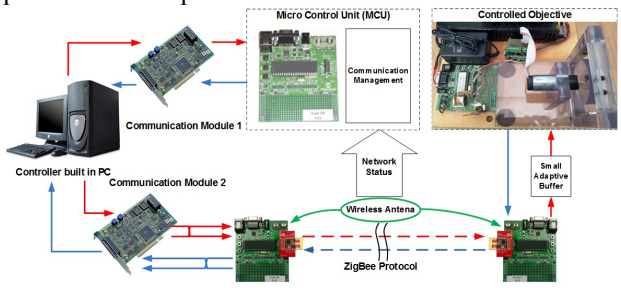

Fig. 2. Configuration of the networked servomotor control system.

B. Setting of Controller Parameters

To design the RVSPC and SSFC-FB, the system plant needed to be known. In this case, it was well addressed in [12]

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

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

The RVSPC control gains were derived in [12]. The SSFC-FB gains $K_{SSFC-FB}=[0.3652\quad 0.2539]$ were achieved by considering the total delay as $\bar{\tau}^d_{SSFC}=0.1s$. Its buffer size was fixed as 5. For the DHDC scheme, using Definition 1, the delay threshold values are selected as: $\bar{\tau}^{com}=0.025s;\bar{\tau}^{ca}=0.035s$; and $\bar{\tau}^{sc}=0.035s$. The controller was then constructed without using the linearized plant: $\eta=1,\rho=1,\mu=1,\hat{\Phi}(0)\equiv\hat{\phi}(0)=2,\varepsilon=10^{-5},\lambda(0)=6.5$. For the MFAC-FB, the buffer size was also fixed as 5. The control parameters were set as $\eta=1,\rho=1,\mu=1,\hat{\Phi}(0)\equiv\hat{\phi}(0)=2,$ $\varepsilon=10^{-5},\lambda=20.5$ corresponding to the worst network state.

C. Experiments and Discussions

Computation delays in [12] and two disturbance sources were added to create the control challenges. The first disturbance came as the magnetic load applied to the motor shaft while the second one was added to the system feedback speed after 12 seconds:

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

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.

Firstly, the constant speed tracking at 10rad/s tests were carried out and the tracking results were obtained in Fig. 3. The SSFC-FB controller could not ensure the smooth tracking due to the use of the fixed control gain and buffer. Furthermore, once existing the packet dropouts on the feedback channel, the SSFC-FB could not update in time the system states and, consequently, generated improperly the control inputs. The performance was significantly degraded when the packet dropout number increased and the second disturbance was applied. By using the RVSPC controller, the performance was improved by the advanced switching control based on 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, both the SSFC-FB and RVSPC could not cancelled the steady state error in the state feedback control performances. With the MFAC-FB, the system response could be estimated using the iteration algorithm when packet dropouts in the backward channel were detected. Thus, the vector of control input always generated to drive the motor. Subsequently, the tracking result was better than that of the SSFC-FB. Nevertheless, the use of fixed buffer size and fix weight factor, the controller could not adapt quickly to the system changes and the disturbance impacts. By using the proposed DHDC approach which possesses both the advantages of the NDP, PIAGM(1,1), PMFAC-based s_k-step-ahead control with SAB, the best tracking result was ensured with fast response, high robustness with zero steady state tracking error. The one-step-ahead delay prediction using the PIAGM was depicted in Fig. 4 while the tracking performances were analyzed in Table I.

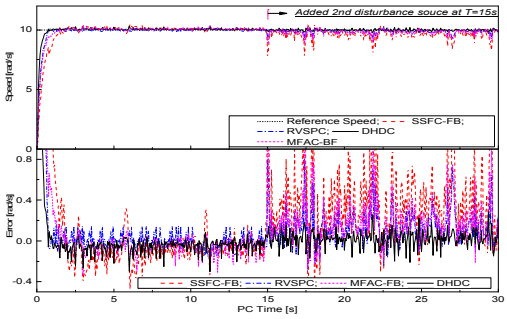

Fig. 3. Step tracking: performances comparison.

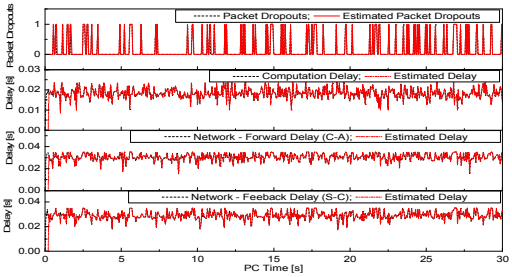

Fig. 4. Step tracking: one-step-ahead prediction of the delays and forward packet dropouts using the PIAGM(1,1).

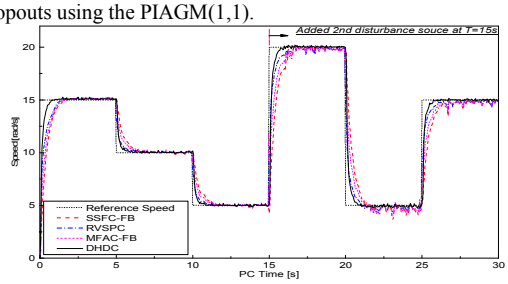

Fig. 5. Multi-step tracking: performance comparison.

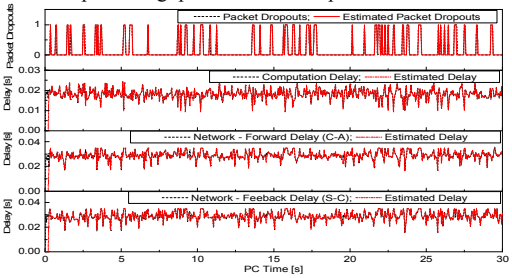

Fig. 6. Multi-step tracking: one-step-ahead prediction of the delays and forward packet dropouts using the PIAGM(1,1).

TABLE I. 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
MFAC-FB	0.38	1.36	8.44	21.88
DHDC	0.44	0.66	2.07	2.81

Secondly, the comparative experiments were done for a multi-step speed tracking control profile. The results were then obtained in Fig. 5 and Fig. 6. These results again show the best performance was achieved by using proposed control method. This comes as no surprise, since the proposed method possesses both the optimal design of SAB and the prediction-based adaptive controller, of which the performance is guaranteed by the robust stability constrains.

VII. CONCLUSIONS

This paper presents the advanced data-based hybrid driven control approach for NCSs. The robust tracking performance is theoretically proved and ensured by tuning online the control weight factor according to the estimated network problems. The capability of the proposed DHDC control scheme has been validated through real-time experiments on the networked servomotor system.

ACKNOWLEDGEMENT

This research are supported by: the Engineering and Physical Science Research Council (EPSRC, EP/P511432/1); Innovate UK through the Off-Highway Intelligent Power Management (OHIPM, 49951-373148) in collaboration with the WMG Centre High Value Manufacturing (HVM), JCB and Pektron; and the Koera's Small and Medium Business Administration (SMBA , S2428274).

REFERENCES

[1] J. Qiu, H. Gao, and S. X. Ding, "Recent Advances on Fuzzy-Model-Based Nonlinear Network Control Systems: A Survey," *IEEE Trans. Ind. Electron.*, vol. 63, no. 2, pp. 1207-1217, Feb. 2016.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] Z. H. Pang, G. P. Liu, D. Zhou, and D. Sun, 'Data-Based Predictive Control for Networked Nonlinear Systems with Network-Included Delay and Packet Dropout,' *IEEE Trans. Ind. Electron.*, vol. 63, no. 2, pp. 1249-1257, Jan. 2016.

[14] 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.