Predictive control for packet dropouts in Wireless Networked Control Systems

M. Chacón[1,*] and R. Katebi[2]

[1]Department of Electronic and Electrical Engineering, University of Strathclyde, 99 George Street, G1 1RD, Glasgow, UK

[2]Industrial Control Centre, Department of Electronic and Electrical Engineering, University of Strathclyde, 204 George Street, G1 1XW, Glasgow, UK

[*]mercedes.chacon-vasquez@strath.ac.uk

Abstract: A predictive PID controller is presented to achieve stability in wireless networked control systems, where the communication is subject to data packet dropouts in both communication routes: sensor to control and control to actuator transmission. The control strategy is based on General Predictive Control (GPC). A Kalman filter and a consecutive dropouts compensator algorithm have been added to the control scheme. The purpose of the algorithm is to develop an estimation and control system that maintains information of the sensor packets and the control actions. Several experiments using the TrueTime network simulator are provided to demonstrate the algorithm and its effectiveness.

1. Introduction

Networked Control Systems (NCS) and Wireless Networked Control Systems (WNCS) are control systems where controllers, sensors and final elements of control are connected to a mutual communication network. NCS applications are increasing as a result of stronger industrial and academic interests in the potential benefits that these systems can offer. However, the main issue that arises with the inclusion of network communication in the control system is the limited capacity of the common channel which, leads to several problems that attempt to degrade the reliability of the NCS. In particular, the network may introduce large communication delays and loss of information, which greatly influences the controller performance. During the last 30 years, the control engineering community has developed a full range of control schemes to cope with these problems. Recent advances in NCS are reviewed in [1], where NCS subject to limited network resources such as event-driven sampling, minimum rate coding and critical dropouts are addressed. Another survey focused on NCS with delay and dropouts can be found in [2]. Also, [3, 4] presented a categorised summary of control algorithms to cope with the networks constraints. Control system analysis and design, network architecture, protocol and scheduling and experimental and simulation studies have been outlined in [5].

Among the proposed methods the networked predictive control scheme stands out, which is considerably effective since it can actively compensate for the transmission delays and consecutive packet dropouts [6]. The first idea of exploiting the predictive controller in NCS can be referred to [7], where the design of a predictive control in communication channels with unbounded delays is presented.

In the case of predictive control for dropouts, a newer contribution is given in [8]. The authors proposed a robust networked predictive control for time-varying delay compensator. They used a multi-step prediction and linear interpolation, to predict the control sequence in combination with a receding optimisation. The design ended in a set of LMI that the authors solved for an illustrative example of a non-linear system and stated the effectiveness of the method.

In [9] a predictive control scheme was implemented to guarantee the stability of the control system under random delay and dropouts. They used Lyapunov theory to derive the conditions of stability. Simulations in Matlab/Simulink-based TrueTime network simulator and experimental results using an Ethernet network and a DC motor with a PD controller showed the effectiveness of this approach regarding control performance. They made real experiments using the IEEE802-11 wireless protocol and the performance of the design was similar. The predictive approach has been addressed for multivariable GPC in [10], where they presented an adaptive predictive control which showed a reasonable performance for an experimental test using UDP and TCP protocols. Stability conditions for networked predictive control with delay compensation have been proposed in [11–13]. They derived stability conditions using Lyapunov functions and proved its effectiveness with some illustrative examples.

In [14] a WNCS is designed focusing on the reduction of energy consumption. They studied the Media Access Control (MAC) parameters and the sampling time, dropouts and random delay of the network and derived a stability condition that reduces the energy consumption. The controller uses the Model-Based Network Control System (MB-NCS) cited in [15]. Another contribution that combined the design of MAC parameters and wireless channel conditions can be found in [16], where a model based predictive networked control system is implemented.

A different solution for predictive WNCS is presented in [17], where a Smith predictor combined with the Cerebellar Model Articulation Controller (CMAC) control is proposed. The predictor is combined with a non-linear PID control in [18] and with a PI control in [19].

While there are numerous approaches that have been reviewed such as fuzzy, predictive and robust control, the PID is the most successful, and has received the most attention in the history of process control. There are some methods for networked PID controllers to compensate for delays, dropouts and other network deficiencies. For instance, the PID tuning problem for varying time-delay systems has been addressed using multi-objective optimisation to develop rules that maximise the value of any additional varying time-delay in the control system [20–24].

Recently, some research has been focused on predictive PID controllers. For instance, [25] addressed the compensation of dropouts using GPC and pole placement structure to find the gains of a PID controller. Stability of the process was verified using Lyapunov theory. In [26], the predictive PID approach combined the optimal tracking control of the GPC and the simple PID structure, which made it easy to implement and gave a reliable controller. Another example is given by Friman [27], who designed a PID controller that keeps the last value in case that the systems do not have the last update. They introduced a simple estimator based on the past values of the output and the reference. The results showed a permanent error as a result of the trial and error mechanism used to select the design parameter of the estimator.

On the other hand, WNCS bring outstanding advantages but also their limited capacity leads to constraints and uncertainties that become more significant in the stability of the closed-loop system. For instance, the limited energy in the wireless nodes leads to non-periodic measurements [28]. Therefore, information is randomly dropped and delayed. These problems motivated the development of control systems that address the complexity and intricate estimation of the WNCS, meanwhile, the simplicity and efficiency of the controller are preferred. An industrial application cited in [29] showed the potential of using wireless communication in a closed-loop control. A PID controller was developed to reduce the power consumption by a reduction of the number of communications of wireless transmitters; at the same time, it managed to maintain a similar performance to a wired control system. Moreover, they showed the adequate performance of the controller to cope with the loss of communication. In [30] a PID tuned with AMIGO rules and combined with a time-varying Kalman filter was proposed and showed satisfactory results using IEEE 802.15.4 wireless equipment.

In this paper, the Model Predictive Control (MPC) technique is used. The motivation to use MPC is that the method has been proved to be robust to perturbations and leads to efficient controllers used in many industrial applications [31]. A predictive PID controller based on GPC scheme is developed to compensate dropouts in WNCS. A quadratic programming problem optimises a reduced GPC criterion to find the optimal PID gains at every sampling time. The velocity PID structure is used to set a general three term controller. The constraint handling is presented to stop input saturation. The problem of the occurrence of dropouts from sensor to controller is compensated by combining the controller with a Kalman filter. The sensed output $y(k)$ is switched to the Kalman estimation $\hat{y}_e(k)$ allowing the controller to have always information of the process even in the presence of dropouts. To compensate consecutive dropouts from controller to actuator, saved predictions of the control signal are calculated. The method is applied to typical second order and non-minimum phase systems with delays. The control system is implemented using the Matlab/Simulink-based TrueTime network simulator. Performance and robustness analysis is investigated. Finally, conclusions and future research work are presented.

The contribution of the proposed methods is that it provides a reliable design for WNCS. The reported predictive PID controller is stable with robustness to the system gain and pole variations. Moreover, it compensates higher occurrences of dropouts and respects constraints, which are significant problems for these systems.

2. The PID

The discrete parallel form of the conventional PID controller is given by:

$$u(k) = k_p e(k) + k_i\, t_s \sum_{i=1}^{k} e(k) + \frac{k_d}{t_s}[e(k) - e(k-1)] \tag{1}$$

where t_s is the sampling time.

The velocity form of the PID controller is considered:

$$\begin{aligned}\Delta u(k) = k_p[e(k) - e(k-1)] + k_i\, t_s e(k) + \\ \frac{k_d}{t_s}[e(k) - 2e(k-1) + e(k-2)]\end{aligned} \tag{2}$$

The matrix representation is selected to facilitate the computation:

$$\Delta u(k) = \mathbf{K^T e(k)} \tag{3}$$

where **e(k)** is the vector of control errors:

$$\mathbf{e(k)} = [e(k) \quad e(k-1) \quad e(k-2)]^T \tag{4}$$

The vector of gains $\boldsymbol{K}$ is defined as:

$$\boldsymbol{K} = \left[k_p + k_i\, t_s + \frac{k_d}{t_s} \quad -k_p - 2\frac{k_d}{t_s} \quad \frac{k_d}{t_s}\right]^T = [k_1\ k_2\ k_3]^T \tag{5}$$

The PID controller gains must be positive scalars:

$$k_p > 0, \qquad k_i > 0, \qquad k_d > 0 \tag{6}$$

Hence, the vector of gains $\boldsymbol{K}$ must fulfil the linear inequality constraints:

$$\begin{aligned} k_1 + k_2 + k_3 &> 0 \\ k_2 + 2k_3 &< 0 \\ k_3 &\geq 0 \end{aligned} \tag{7}$$

3. The GPC reduced criterion

This section defines the process model and the predictive algorithm for the GPC design. Consider the GPC cost function:

$$J = \sum\nolimits_{j=N_1}^{N_{ph}} [\hat{y}(k+j) - r(k+j)]^2 + \sum_{j=N_1}^{N_u} \lambda[\,\Delta u(k+j-1)]^2 \tag{8}$$

The equation is a weighted sum of square predictive future errors and square control signal increments, where: N_1 and N_{ph} are positive scalars indicating the initial and final predictive horizons. λ is the weighting parameter for the control input. A constant weight is used to penalise the control effort. N_u is the control horizon. Finally, $r(k+j)$ is the future reference trajectory that has been assumed to be known.

Now, from equation (8) it is required to find the prediction of process output $\hat{y}(k+j)$ to minimise the cost function. The Diophantine equations are used to find the prediction matrices.

Consider a linear SISO plant described by the Controlled Auto Regressive and Integrated Moving-Average (CARIMA) model, which is the well-known Auto Regressive Moving Average with exogenous terms (ARMAX) model in terms of control deviation variables. This model is represented by:

$$A(z^{-1})y(k) = z^{-d}B(z^{-1})u(k-1) + C(z^{-1})\frac{\xi(k)}{\Delta} \tag{9}$$

where $y(k)$ and $u(k)$ are the process output and the control input respectively. d is a delay of the process. A discrete-time setting is assumed and the current time is labelled as time instant k. A, B and C are polynomials function of z^{-1} with order na, nb and nc respectively. They are represented as follows:

$$\begin{aligned}A(z^{-1}) &= 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{na} z^{-na} \\ B(z^{-1}) &= b_0 + b_1 z^{-1} + b_2 z^{-2} + \\ &\cdots + b_{n_b} z^{-n_b} \\ C(z^{-1}) &= c_0 + c_1 z^{-1} + c_2 z^{-2} + \\ &\cdots + c_{n_c} z^{-n_c}\end{aligned} \tag{10}$$

The model represents the uncertainty of random disturbances in the process. $\xi(t)$ is a zero mean white noise, and $\Delta = 1 - z^{-1}$ is a difference operator, indicating the difference between the current time point and the previous time point. This makes the model more appropriate in industrial applications where disturbances are non-stationary.

Consider the following Diophantine equation:

$$1 = E_j(z^{-1})\Delta A(z^{-1}) + z^{-j} F_j(z^{-1}) \tag{11}$$

where E_j and F_j are polynomials.

Multiplying equation (9) by $\Delta E_j(z^{-1})z^j$ gives:

$$\begin{aligned}\Delta A(z^{-1})E_j(z^{-1})\hat{y}(k+j) = \\ E_j(z^{-1})B(z^{-1})\Delta u(k+j-d-1) + \xi(k+j)\end{aligned} \tag{12}$$

The best estimation of the future disturbance is obtained by selecting $\xi(t+j) = 0$. By substituting

$A(z^{-1})E_j$ from equation (11) in equation (12), it results:

$$\begin{aligned}(1 - z^{-j}F_j(z^{-1}))\hat{y}(k+j) = \\ E_j(z^{-1})B(z^{-1})\Delta u(k+j-d-1)\end{aligned} \tag{13}$$

Simplifying one get:

$$\begin{aligned}\hat{y}(k+j) = F_j(z^{-1})y(k) + \\ E_j(z^{-1})B(z^{-1})\Delta u(k+j-d-1)\end{aligned} \tag{14}$$

where

$$\begin{aligned}E_j(z^{-1}) = e_{d+j,0} + e_{d+j,1}z^{-1} + \cdots + e_{d+j,j-1}z^{-(d+j-1)} \\ F_j(z^{-1}) = f_{d+j,0} + f_{d+j,1}z^{-1} + \cdots + f_{d+j,na}z^{-na}\end{aligned} \tag{15}$$

Now, equation (14) can be expressed as:

$$\hat{y}(k+j) = F_j(z^{-1})y(k) + G(z^{-1})\Delta u(k+j-1) \tag{16}$$

where

$$G = E_j(z^{-1})B(z^{-1}) \tag{17}$$

The cost function J stated in equation (8) will now be formulated as the following quadratic problem, by replacing the output predictions in equation (16):

$$\begin{aligned}J(K) = (\mathbf{Gu} + \mathbf{Fy} - \mathbf{r})^{\mathbf{T}} \\ (\mathbf{Gu} + \mathbf{Fy} - \mathbf{r}) + \lambda \boldsymbol{u}^{\mathbf{T}}\boldsymbol{u}\end{aligned} \tag{18}$$

where

$$\boldsymbol{r} = [r(k+1) \quad r(k+2) \quad \cdots \quad r(k+N_{ph})]^T \tag{19}$$

$$\mathbf{y} = [\hat{y}(k+1) \quad \hat{y}(k+2) \quad \cdots \quad \hat{y}(k+N_u)]^T \tag{20}$$

$$\mathbf{u} = [\Delta u(k) \quad \Delta u(k+1) \quad \cdots \quad \Delta u(k+N_u-1)]^T \tag{21}$$

Note that the optimal input solution $\Delta u(k+j-1) = 0$ for $j > 1$. The control horizon has been selected as one because the PID law only computes $\Delta u(k)$.

4. The Predictive PID control

The PID control law is formulated using the result in equation (3). Simplifying equation (18):

$$\begin{aligned} J(K) = \mathbf{u^T}(\mathbf{G^T G} + \lambda \boldsymbol{I})\mathbf{u} \\ +\mathbf{u^T 2 G^T}(\mathbf{Fy} - \mathbf{r}) \end{aligned} \tag{22}$$

Now, replace Δu using equation (3):

$$\begin{aligned} J(K) = (\mathbf{e^T}\boldsymbol{K})^{\mathbf{T}}(\mathbf{G^T G} + \lambda \boldsymbol{I})\mathbf{e^T}\boldsymbol{K} \\ +(\mathbf{e^T}\boldsymbol{K})^{\mathbf{T}}\mathbf{2 G^T}(\mathbf{Fy} - \mathbf{r}) \end{aligned} \tag{23}$$

This is equivalent to:

$$\begin{aligned} J(K) = \boldsymbol{K}^{\mathbf{T}}\ \mathbf{e}\,(\mathbf{G^T G} + \lambda \boldsymbol{I})\mathbf{e^T}\ \boldsymbol{K} \\ +\boldsymbol{K}^{T}\ 2\,\mathbf{e}\,\mathbf{G^T}(\mathbf{Fy} - \mathbf{r}) \end{aligned} \tag{24}$$

The minimum cost is found by computing the gradient of J with respect to the vector of PID gains. Thus, equation (24) becomes:

$$\begin{aligned} min\, J(K) = \{\mathbf{e}\,(\mathbf{G^T G} + \lambda \boldsymbol{I})\mathbf{e^T} + [\mathbf{e}\,(\mathbf{G^T G} + \lambda \boldsymbol{I})\mathbf{e^T}]^{\mathbf{T}}\}\boldsymbol{K} \\ +2\,\mathbf{e}\,\mathbf{G^T}(\mathbf{Fy} - \mathbf{r}) = \mathbf{0} \end{aligned} \tag{25}$$

It can be reduced as:

$$\begin{aligned} min\, J(K) = 2\mathbf{e}\,(\mathbf{G^T G} + \lambda \boldsymbol{I})\mathbf{e^T}\,\boldsymbol{K} \\ +2\,\mathbf{e}\,\mathbf{G^T}(\mathbf{Fy} - \mathbf{r}) = \mathbf{0} \end{aligned} \tag{26}$$

For an unconstrained optimisation problem one can find easily the vector of PID gains. However, in this case Matlab optimisation Toolbox is used to compute the minimum cost at every sampling time. In specific, the function *quadprog* is selected. Using this quadratic program algorithm the optimal PID parameters that satisfy the constraints can be found. The quadratic function can be expressed as:

$$J = \quad \frac{1}{2} \cdot \mathbf{x}^{\mathrm{T}} \cdot H \cdot \mathbf{x} + m^{\mathrm{T}} \cdot \mathbf{x} \tag{27}$$

Comparing the results in equation (24) and equation (26) gives:

$$\begin{aligned}
&\mathbf{x} = \boldsymbol{K} \\
&\mathbf{H} = \mathbf{2}(\mathbf{G^T G} + \lambda \boldsymbol{I})\mathbf{e}(\mathbf{k})\mathbf{e}(\mathbf{k})^{\mathbf{T}} \\
&\mathbf{m} = \mathbf{2G^T}(\mathbf{F} - \mathbf{r})\mathbf{e}(\mathbf{k})
\end{aligned} \tag{28}$$

The optimisation will be carried out by minimising the cost function respect to the PID controller gains K:

$$\boldsymbol{\Delta}\mathbf{u_{cons}} = \min_{\boldsymbol{K}} J(\boldsymbol{K}, \mathbf{r}, \mathbf{F}, \mathbf{G}) \tag{29}$$

where Δu_{cons} is the optimal input trajectory at time instant k. Therefore, the design of the predictive PID controller is based on the following optimisation problem:

$$\begin{aligned}
\min_{\boldsymbol{K}} \quad & \frac{1}{2} \cdot \boldsymbol{K}^{\mathrm{T}} \cdot \mathbf{H} \cdot \boldsymbol{K} + \mathbf{m}^{\mathrm{T}} \cdot \boldsymbol{K} \\
s.t. \quad & \mathbf{a}(\mathbf{k}) \cdot \boldsymbol{K} \leq \mathbf{b}(\mathbf{k})
\end{aligned} \tag{30}$$

The constraints of equation (30) will guarantee the contributions of control input and rate input are applied according to the controller limitations. The design is extended in the next section.

The control law can be rewritten from equation (3) as:

$$\boldsymbol{\Delta}\mathbf{u}(\mathbf{k}) = \boldsymbol{K}(\mathbf{k})\mathbf{e}(\mathbf{k}) \tag{31}$$

Note that the vector of PID gains will change at every time instant k. Figure 1 shows the details of the Predictive PID implementation, where the main functional blocks of the controller are provided.

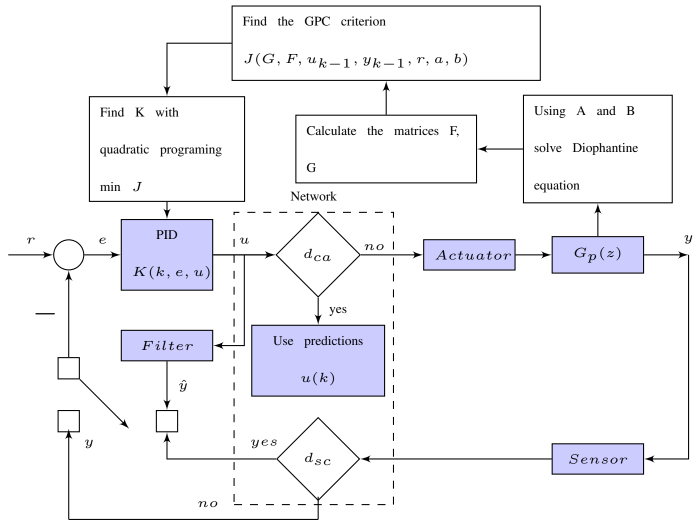

Figure 1. *Diagram of predictive PID controller structure*

5. Constraints for the control input and control input increment

Here the constraints to select the appropriate predictive PID gains to prevent input saturation are formulated. To introduce the constraint handling, the predictive PID control subject to linear constraints for the input and input increment is solved:

$$
\begin{aligned}
&-\Delta u_{min} \leq \ \Delta u(k) \leq \Delta u_{max} \\
&-u_{min} \leq \ u(k) \leq u_{max}
\end{aligned}
\tag{32}
$$

The predictive PID control law in equation (31) can be defined as:

$$
\Delta u = k_1\, e(k) + k_2\, e(k-1) + k_3\, e(k-2) \tag{33}
$$

By combining the previous result and $\Delta u = u(k) - u(k-1)$ in equation (33) the constraints can be written as:

$$
\begin{aligned}
&-\Delta u_{min} \leq \ k_1\, e(k) + k_2\, e(k-1) + k_3\, e(k-2) \leq \Delta u_{max} \\
&-u_{min} - u(k-1) \leq \ k_1\, e(k) + k_2\, e(k-1) + k_3\, e(k-2) \\
&\leq u_{max} - u(k-1)
\end{aligned}
\tag{34}
$$

The inequalities in (34) can be separated and a matrix arrangement is obtained. Moreover, by combining equation (7) with the previous result the final constraint matrix is found:

$$\begin{bmatrix} 0 & 1 & 2 \\ 0 & 0 & -1 \\ -1 & -1 & -1 \\ e(k) & e(k-1) & e(k-2) \\ -e(k) & -e(k-1) & -e(k-2) \\ e(k) & e(k-1) & e(k-2) \\ -e(k) & -e(k-1) & -e(k-2) \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} \leq \begin{bmatrix} -\epsilon \\ 0 \\ -\epsilon \\ \Delta u_{max} \\ -\Delta u_{min} \\ u_{max} - u(k-1) \\ -u_{min} + u(k-1) \end{bmatrix} \tag{35}$$

where ϵ is a small positive scalar. Note that, the constraints should be fulfilled for every $\Delta u(k)_j,\ j = 1, \ldots, N_{ph} - 1$.

Note that, the final constraint matrix in equation (35) has the form $\mathbf{a(k)} \cdot \boldsymbol{K} \leq \mathbf{b(k)}$ previously defined in the optimisation problem proposed in (30).

6. Dropouts from controller to actuator compensation

Predictions of the control signal are computed and saved in the actuator node to compensate the occurrence of dropouts from controller to actuator.

From equation (28) the matrix $\mathbf{G_1}$ is calculated instead of $\boldsymbol{G}$:

$$\mathbf{G1} = \mathbf{G(1 : N_{ph}, j)} \tag{36}$$

where j stands for columns of matrix $\boldsymbol{G}$ and $N_1 \leq j \leq N_{ph}$. Therefore, j predictions of the control signal $\Delta u(k)$ are calculated using the coefficients of $j - th$ column of matrix $\boldsymbol{G}$. For this, N_{ph} predictions of Δu are computed every sampling time and stored them for the next sampling instant. If there is a dropout, the next saved value is applied. In the case of consecutive dropouts, the maximum number of consecutive dropouts γ_{max} is selected to match the prediction horizon N_{ph}. Thus, the controller can determine the occurrence of consecutive dropouts and apply the past predictions until either the condition is over or γ_{max} has been reached.

Since the stability of the system will depend on the selection of N_u, the number of consecutive dropouts is measured for variations of the packet loss rate from 25% to 80%. A maximum value of $\gamma_{max} = 30$ is found.

7. Dropouts from sensor to controller compensation

7.1. Kalman Filter Construction

In this section, the design of the Kalman filter is presented. It will have an important role to improve the performance of the whole control system. The occurrence of dropouts during the transmission from the sensor to the controller results in an open-loop system which degrades the reliability of the WNCS. To solve this problem a Kalman filter is implemented to generate a prediction of the process output when dropouts are presented.

An estimator algorithm is used according to:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + v(k) \\ y(k) &= Cx(k) + w(k) \end{aligned} \tag{37}$$

where the process noise $v(k)$ and measurement noise $w(k)$ are independent Gaussian white sequences with covariance Q and R, respectively. The covariance is described as follows:

$$\mathbb{E}\left\{ \begin{bmatrix} \omega(k) \\ v(k) \end{bmatrix} \begin{bmatrix} \omega(k) & v(k) \end{bmatrix} \right\} = \begin{bmatrix} Q(k) & S(k) \\ S(k)^T & R(k) \end{bmatrix} \delta(k) \tag{38}$$

where $\delta(k)$ is the Kronecker delta function. The cross covariance $S(k)$ is assumed to be zero for independent processes. $\mathbb{E}$ stands for the estimation.

The matrices A, B, C are calculated using the discrete transformation from process transfer function to state-space representation. The Kalman filter gives the estimate of the state as follows:

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + K_f(y - \hat{y})$$
$$\hat{y} = C\hat{x} \tag{39}$$

where $\hat{y}(k)$ is the estimation of $y(k)$ at time k.

K_f represents the steady-state filter gain that is calculated as follows:

$$K_f = PC^T R^{-1} \tag{40}$$

where P is the covariance of estimation error that satisfies the following Riccati Difference Equation:

$$PA^T + AP - PC^T R^{-1} CP + Q = 0 \tag{41}$$

The filter gain is computed by selecting appropriate values for Q and R. The observer is implemented in Matlab/Simulink on the controller side using the estimate space-state model of the process. At time k, the predictive PID controller will read the sensed data y and if a dropout takes place, the information is obtained from the prediction $\hat{y}$.

7.2. Delay approach

The Kalman filtering results can be extended to handle delay in the system. A smoothed estimation can be used to study the delay. The Kalman filter algorithm assumes no delay exist between $y(k-1)$ and $y(k)$ measurements. If a delay d exists an estimation of $x(k)$ can be obtained by smoothed estimates for lags up to d samples, that is:

$$\hat{x}^d(k) = \mathbb{E}\left\{x(k-d)|y(0), \cdots, y(k-1)\right\} \tag{42}$$

This is called a smoothed estimate. The accuracy increases when more measurements are used in estimating the state. However, the greater the delay, the greater the complexity of the estimator.

A fixed-lag smoothing problem with a fixed delay d is chosen. It can be derived by augmenting the state vector by delayed versions of the state. The algorithm is:

$$\begin{bmatrix} x(k+1) \\ x^1(k+1) \\ \vdots \\ x^d(k+1) \end{bmatrix} = \begin{bmatrix} A & 0 & \cdots & 0 \\ I & 0 & \cdots & \vdots \\ 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & I & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ x^1(k) \\ \vdots \\ \vdots \\ x^d(k) \end{bmatrix} + \begin{bmatrix} I \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix} \upsilon(k) \tag{43}$$

$$y(k) = \begin{bmatrix} C & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ x^1(k) \\ \vdots \\ \vdots \\ x^d(k) \end{bmatrix} + \omega(k) \tag{44}$$

The gain K is obtained from the standard filter using equation (40).

8. Simulation studies

8.1. Numerical example 1: SO process

The performance of the predictive PID control is investigated using simulation studies. The proposed algorithm has been implemented and tested using the Matlab/Simulink-based TrueTime network simulator configured for wireless protocol 802.11b (WLAN), with a data rate of 800000 bits/s. The minimum frame size has been selected as 272 bits. The results have been compared with the solutions obtained by the classical GPC with constraints.

Consider the following system with a sampling time $T_s = 0.01\ s$:

$$G_p(z) = \frac{0.003319(z + 0.5215)}{(z - 0.9755)(z - 0.09748)} \tag{45}$$

Although different values of the penalty in the control action can be selected, for a faster response of the closed-loop $\lambda = 0.5$ is chosen. As explained before, the prediction horizon is $N_{ph} = 30$ and the control horizon $N_u = 1$. Control input constraints have been assumed as

$u_{max} = 3$, $u_{min} = -3$ and the rate input $\Delta u_{max} = 10$. The optimisation problem has been set using the command $quadprog(H, m, a, b)$, where H, m have been stated in equation (28) and a, b are the constraint matrices of the linear inequality in equation (35). The interior-point-convex algorithm is used. Lower and upper bounds have been added to the optimisation algorithm to find positive and finite PID gains. The optimal vector of PID parameters K is found and compared to equation (5) to find the optimal gains of the predictive PID every sampling time. Figure 3 shows the system outputs and constrained controller inputs for the predictive PID and GPC for a sin wave reference signal (black dashed line). The reference tracking is achieved and the control signal satisfies the constraints (red dotted line). Moreover, the percentage and occurrence of dropouts for the simulation are depicted in figure 2. A variable $d_p(k)\ \varepsilon\ [0, 1]$ is created to indicate if the

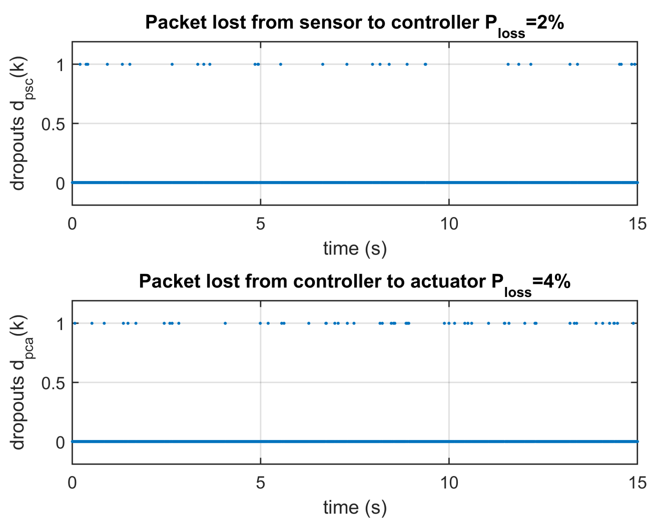

Figure 2. *Time instant of data dropouts*

packet containing the feedback signal $y(k)$ was received at the controller node $(d_{psc}(k) = 0)$ or if it was dropped $(d_{psc}(k) = 1)$. Similarly, packet dropouts from the controller to the actuator are represented as $(d_{pca}(k) = 1)$ and $(d_{pca}(k) = 0)$ if there is no dropouts.

The predictive PID (purple line) shows almost the same behaviour than GPC (light purple line). The performance of the controllers has been assessed using the Integral of Absolute Error (IAE). The criterion returned a value of $J = 1.8636$ for the predictive PID response which is better than the GPC: $J = 1.9293$.

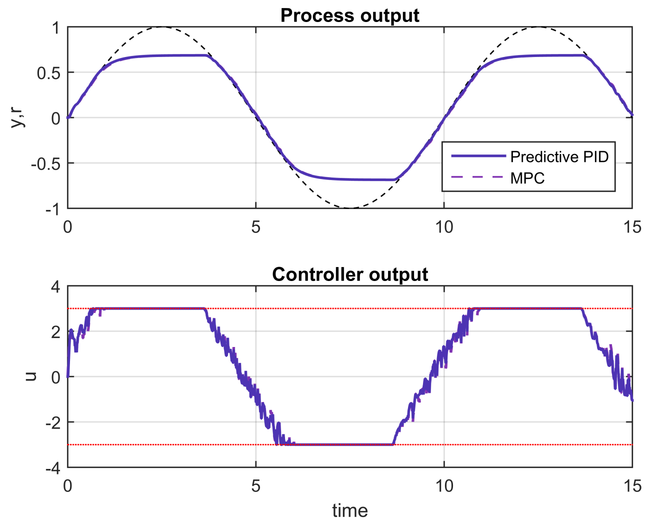

Figure 3. *System outputs for predictive PID constrained, TrueTime simulator*

Although minor oscillations are found in the signals, the controller is stable and works within the requirements even with the presence of dropouts.

8.2. Numerical example 2: Non-minimum phase process

The following non-minimum phase process with dead time and sampling time $T_s = 1\ s$, is considered to test the robustness of the design for a persistent disturbance input:

$$G_p(z) = \frac{-0.26785\,(z - 1.292)\,z^{-3}}{(z - 0.6065)(z - 0.006738)} \tag{46}$$

The designs are tested under a step disturbance of magnitude 1.1, introduced at time $t = 450\,s$. In this case, the closed-loop stability is achieved by selecting $\lambda = 25$. The input constrains are assumed $u_{max} = 10$, $u_{min} = -5$ and the rate input $\Delta u_{max} = 10$.

The predictive PID and GPC responses are shown in figure 4. The reference is a sequence of step input of magnitudes 2, -1 and 1. The percentages of dropouts from sensor to controller and from controller to actuator were 9% and 7%, respectively. The results demonstrated that disturbance rejection is achieved. Moreover, the input constraints are satisfied, however this leads to a slower rising time.

The performance of predictive PID and GPC responses for servo and regulatory responses has

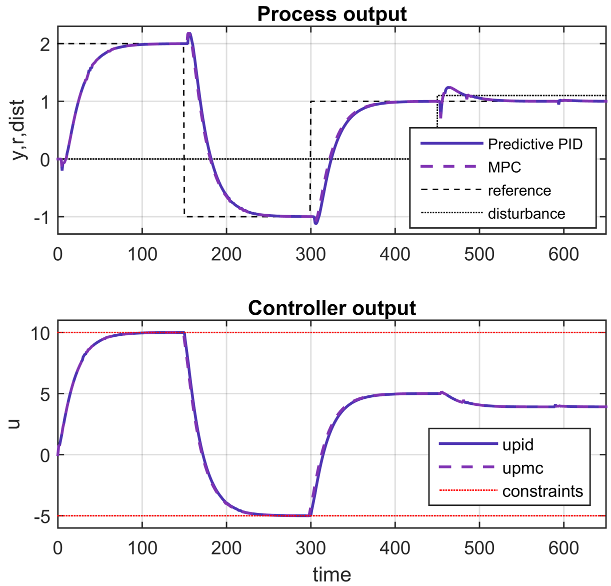

Figure 4. *System outputs for predictive PID and GPC*

been assessed using the IAE criterion. The results are summary in the Table 1.

Table 1 IAE values for step responses

Controller	J_r	J_d
Predictive PID	211.8	7.988
GPC	205	7.989

The predictive PID and the GPC performances present similar results.

9. Robustness stability

Process dynamics and percentage of dropouts will change during WNCS operation. Since the proposed design is based on a simplified model of the plant, the sensitivity of the closed-loop system to these variations is a fundamental issue. Hence, the stability of the method is investigated here studying the closed-loop responses for variations of the process model parameters and the percentage of dropouts. The following second order process is selected for this analysis:

$$G_p(z) = \frac{0.06347z^{-1} + 0.04807z^{-2}}{1 - 1.323z^{-1} + 0.4346z^{-2}} \tag{47}$$

The controller settings are the same than the previous example. A step of magnitude one is selected. Control input constraints have been chosen as $u_{max} = 1$, $u_{min} = 0$ and the rate input $\Delta u_{max} = 10$.

9.1. Study of stability for variations of percentage of dropouts

The percentage of dropouts from sensor to controller and from controller to actuator were varied to demonstrate the robustness of the design.

The step responses for different scenarios are showed in figure 5. The solid line shows that when the probability of loss is increased from zero to 65%, the responses are similar. Nevertheless, after 20 s, the control input for a 65% of packet lost presents small oscillations. If the probability keeps increasing, the oscillations continue to grow until the system response is unstable. The dashed line shows a higher percentage of dropouts and the performance of control system has decreased considerably.

Further validations reported that, the closed-loop response is stable for a maximum probability of dropouts from sensor to controller of 84%, that means the system is still stable when only 16% of process measurements are transmitted from sensor to controller. Moreover, the control system can compensate at the same time for a percentage of dropouts from controller to actuator of 13% which means that only 87% of the control inputs are received by the process.

9.2. Study of stability for variations of the gain

Figure 6 shows that even with the constraints, the closed-loop system is stable if the gain is increased and reduced to $\pm 35\%$ of the model process gain. Although the process presented a small oscillation and slower rising time, zero steady error and a good tracking performance are accomplished when the process gain changes within the given percentages.

9.3. Study of stability for variations of the poles

Figure 7 shows the closed-loop responses for variations in the non dominant pole called p_1. Note that, the effect of varying the p_2 is similar since the poles are closer to each other. It is evident

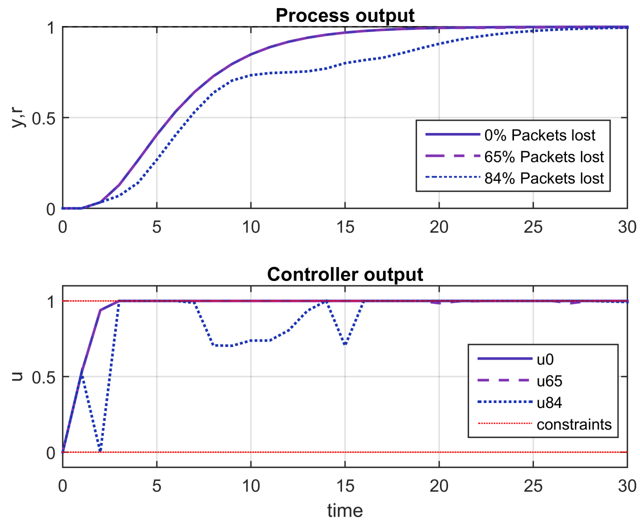

Figure 5. *Comparison of step responses with dropouts*

from the plot, that pole variations of $\pm 35\%$ are permitted without making the closed-loop system unstable.

9.4. Discussion

The performance stability of the control scheme is satisfactory for the selected prediction horizon. Further tests showed that a larger N_{ph} deteriorate the performance because the errors in the prediction are bigger for long prediction horizon. The sampling time of the wireless networked control system should be carefully chosen. The selected values for the previous validations guarantee the stability of the implementation. Especial attention has to be considered for the interaction between sensor, actuator and controller nodes. The sampling times of the first ones must be smaller than the later one. The constraint handling produces a reduction of the performance, but satisfactory results are still found. In most cases, a faster weight λ can improve the sluggish response of the control signal. However, there are some cases where the control strategy can not stabilise faster responses with high percentages of dropouts. The predictive PID controller and GPC show similar performances; in some cases, PID controller has better performance for higher percentages of dropouts.

For the given example, the control performance of the PID controller degrades gracefully when

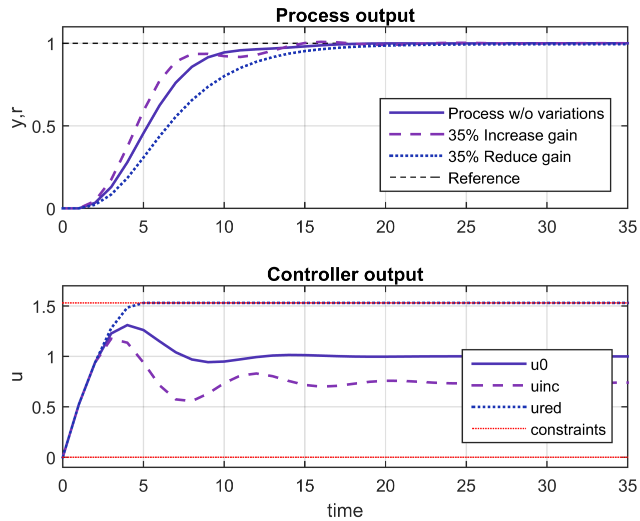

Figure 6. *Comparison of step responses with gain process model variations*

the percentage of dropouts is increased, nonetheless, after 84% it behaves poorly. Even with the constraints, the system has a good tracking performance when process gain and poles vary within the specified interval.

In conclusion, the fact that the closed-loop system is robust to process and dropouts variations obeys to the optimisation tool used to obtain the predictive PID controller gains. The design WNCS successfully minimises the error by changing the controller gains at every sampling time and allowing a maximum system parameter variation and percentage of dropouts.

10. Conclusions

A Predictive PID controller based on the GPC is presented. The results above demonstrate the ability of the proposed method to deal with higher dropouts and higher consecutive occurrence. The constraint handling is presented to stop input saturation. The problem of dropouts from sensor to controller was compensated by combining the controller with a Kalman filter. Moreover, saved predictions of the control signal are calculated to compensate consecutive dropouts from controller to actuator.

The control system is designed and implemented using the Matlab/Simulink-based TrueTime network simulator and the results showed that the complete approach successfully resolved two

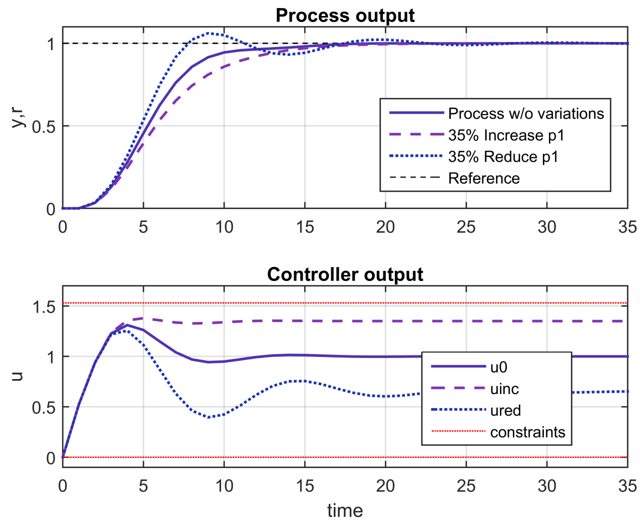

Figure 7. *Comparison of step responses with pole 1 process model variations*

main problems in the WNCS: missing sensor measurements and controller actions.

The design has been compared to GPC using typical second order and non-minimum phase systems with delays. The performance analysis has been done for the second order process by studying the performance index IAE. Results showed that both the predictive PID controller and GPC have similar performance.

Also, an analysis of the robustness of the system has been studied for a step disturbance and variations of the process model parameters. The stability of the control system has been proved under a very lossy wireless network.

In future works, this approach will be extended to complex WNCS with MIMO systems and decentralised control which results can be tested in an industrial application.

11. Acknowledgements

M. Chacón wishes to thank the financial support from the University of Costa Rica and MICITT/CONICT Costa Rica.

References

[1] Zhang Jin Peng Dacheng Guo Cancan, Peng Chen. A survey on networked control systems subject to limited network resources. *Control and Decision Conference (2014 CCDC), The 26th Chinese*.

[2] Y. L. Wang, W. T. Liu, X. L. Zhu, and Z. Du. A survey of networked control systems with delay and packet dropout. In *2011 Chinese Control and Decision Conference (CCDC)*, pages 2342–2346, 2011.

[3] Rachana Ashok Gupta and Mo-Yuen Chow. Networked control system: Overview and research trends. *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, 57:2527–2535, 2010.

[4] Yonggang Xu. Hespanha J. P., Naghshtabrizi P. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95:138–162, 2007.

[5] T. Yang. Networked control system: A brief survey. *IET ControlTheory Appl.*, 153:403412, 2006.

[6] X. M. Sun, D. Wu, C. Wen, and W. Wang. A novel stability analysis for networked predictive control systems. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 61(6):453–457, 2014.

[7] A. Bemporad. Predictive control of teleoperated constrained systems with unbounded communication delays. *Proceedings of the 37th IEEE Conference on Decision and Control Tampa, Florida USA December 1998*.

[8] Qiuxia Chen Ying Liu and Haoqi Zhu. Robust model predictive control for a class of networked control systems with timevarying delays. *2014 IEEE Workshop on Electronics, Computer and Applications*.

[9] E. Parlakay et al A. Onat, T. Naskali. Control over imperfect networks: Model-based predict-

ive networked control systems. *IEEE Transactions on Industrial Electronics*, 58:905–913, 2011.

[10] P. Tang and C. de Silva. Compensation for transmission delays in an ethernet-based control network using variable-horizon predictive control. *IEEE Trans. Control Syst. Technol*, 14.

[11] Lihua Dou Jian Sun, Jie Chen. Networked predictive control for linear systems with unknown communication delay. *2014 UKACC International Conference on Control 9th - 11th July 2014, Loughborough, U.K*, 61:668 – 672, 2014.

[12] G. Pin and T. Parisini. Networked predictive control of uncertain constrained nonlinear systems recursive feasibility and input-to-state stability analysis. *IEEE Trans. Autom. Control*, 56:72–87, 2011.

[13] David Rees Guo-Ping Liu, Yuanqing Xia and Wenshan Hu. Design and stability criteria of networked predictive control systems with random network delay in the feedback channel. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, 37:173–184, 2007.

[14] Huijun Gao Lixian Zhang and Okyay Kaynak. An adaptive tuning algorithm for ieee 802.15.4-based network control system. *Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference*.

[15] L. A. Montestruque and P. J. Antsaklis. On the model-based control of networked systems. *Automatica*, 39.

[16] O. Gurbuz A. Ulusoy and A. Onat. Wireless model-based predictive networked control system over cooperative wireless network. *IEEE Transactions on Industrial Informatics*, 7:41–51, 2011.

[17] Feng D. Wencai D. New smith predictor and cmac– pid control for wireless networked control systems. In *10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel Distributed Computing*, College of Information Sciences and Technology, Hainan University Haikou, Hainan, China, 2009. ACIS.

[18] F. Du, W. Du, and Lei Zhi. New smith predictor and nonlinear control for networked control systems. In *Engineers and Computer Scientists, 2009. IMECS 2009. International MultiConference on*, March 2009.

[19] Feng Du, Wencai Du, and Zhi Lei. A novel smith predictor for wireless networked control systems. In *Control, Automation and Systems Engineering, 2009. CASE 2009. IITA International Conference on*, pages 667–670, July 2009.

[20] L.M. Eriksson and M. Johansson. Simple pid tuning rules for varying time-delay systems. In *The 46th IEEE Conference on Decision and Control CDC 2007*, pages 12–14, LA, USA, 2007. International Institute of Electrical and Electronics Engineers (IEEE).

[21] T. Oksanen L. Eriksson. Pid controller tuning for integrating processes: Analysis and new design approach. In *Fourth International Symposium on Mechatronics and its Applications (ISMA07)*, pages 26–29, LA, USA, 2007.

[22] H. N. Koivo L. Eriksson. Tuning of discrete-time pid controllers in sensor network based control systems. In *in Proc. 2005 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA2005)*, page 6, Espoo, Finland, 2005. International Institute of Electrical and Electronics Engineers (IEEE).

[23] H. Koivo M. Pohjola, L. Eriksson. Tuning of pid controllers for networked control systems. In *in Proc. The 32nd Annual Conference of the IEEE Industrial Electronics Society (IECON060)*, page 6, Paris, France, 2006. International Institute of Electrical and Electronics Engineers (IEEE).

[24] L.M. Eriksson and M. Johansson. Pid controller tuning rules for varying time-delay systems. In *American Control Conference, 2007. ACC '07*, pages 619–625, July 2007.

[25] Eva Miklovi and Marián Mrosko. PID control strategy for sensor random packet dropouts in networked control system. *International Journal of Systems Applications, Engineering and Development*, 6(1), 2012.

[26] K. K. Tan, T. H. Lee, S. N. Huang, and F. M. Leu.

[27] Mats Friman and Joona Nikunen. A practical and functional approach to wireless pid control. *2013 21st Mediterranean Conference on Control and Automation (MED) Platanias-Chania, Crete, Greece, June*, pages 25–28, 2013.

[28] Araujo J. Johansson K. H. Tiberi, O. On event-based pi control of first-order processes. In *IFAC Conference on Advances in PID Control PID'12 Brescia (Italy)*, pages 28–30, Stockholm, Sweden, 2012. ACCESS Linnaeus Center, KTH Royal Institute of Technology.

[29] Nixon M. Blevins, T. and M. Zielinski. Using wireless measurements in control applications. *International Society of Automation (ISA)*, 2013.

[30] V. Ungan. Networked pid controllers for wireless systems. Masters degree project, KTH ELectrical Engineering, Stockholm, Sweden, November 2010.

[31] E. Camacho and A. Bordons. *Model Predictive Control*. Springer-Verlag, London, U.K., 2007.