Strathprints Institutional Repository

Comparison of PID methods for Networked Control Systems

Mercedes Chacón Vásquez
Technology and Innovation Centre
99 George Street
Glasgow, United Kingdom
mercedes.chacon-vasquez@strath.ac.uk

Reza Katebi
Industrial Control Centre
University of Strathclyde
Glasgow, United Kingdom
m.r.katebi@strath.ac.uk

Abstract—**This work presents significant developments in Networked Control Systems based on PID, Internal Model Control and Smith Predictor algorithms. The main purpose of this research paper is to study the performance and robustness offered by these control design methods in handling the challenging control problem encountered with systems subject to time-varying delays and dropouts. It is expected that proposed design methods achieve design requirements such as margins of robustness, performance criterions and stability conditions while the simplicity and flexibility of the controller are preferred. Performance of these controllers is evaluated and extensive simulations of these methods are presented using Matlab TrueTime toolbox.**

I. INTRODUCTION

It is harder to design Networked Control Systems (NCS) to meet performance objectives than simple process control systems. This is down to two important reasons. Firstly, it has to consider system dynamics and disturbances. Secondly, constraints of the network such as dropouts and delays limit the performance of the control system.

The compensation of time-varying delays in NCS is a complex problem that requires controllers with high levels of performance and robustness to ensure reliability in the control system. The need for a simple and flexible algorithm that fulfils these requirements has given birth to different methods of controlling NCS. In this paper, novel methods of designing PID controllers for NCS are presented.

A. Smith predictor controller design for NCS

The most important characteristic of this controller is that the design and performance will depend on the information of the process model and the time delay. The closed-loop transfer function for this system is depicted in (1).

$$\frac{y(s)}{r(s)} = \frac{C\,e^{-\tau_{ca}\,s}\,G_p\,e^{-\tau_p\,s}}{1 + C\,G_{pm} + C\,e^{-\tau_{ca}\,s}\,(G_p\,e^{-\tau_p\,s} - G_{pm}\,e^{-\tau_{pm}\,s})\,e^{-\tau_{sc}\,s}} \quad (1)$$

where $u(t)$ is the controller output, $y(t)$ stands for process output and $r(t)$ is the reference signal. The network delays are represented by the sensor-to-controller delay and controller-to-actuator time delay τ_{sc} and τ_{ca}, respectively. $C(s)$ is the controller, $G_{pm}(s)$ is the prediction model of the controlled plant $G_p(s)$ and τ_{pm} is the prediction value of the process time delay τ_p.

According to (1) if the prediction model matches accurately the dynamics of the process model, the delays of the network
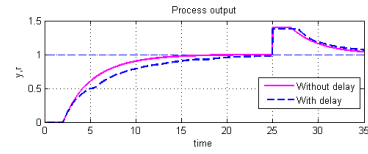

Fig. 1. System outputs for Smith predictor

can be effectively eliminated from the characteristic equation. However, for a real process subject to disturbances and variations of its parameters, the accuracy of the prediction model is not perfectly guaranteed. Furthermore, an analysis of this situation and effects on the performance of the controller is presented.

1) The PI controller tuning: For a good performance and robustness in the control design, a standard PI controller in the inner loop of the Smith predictor structure is tuned using the AMIGO rules [1].

2) Numerical example: Consider the following first order plus dead time (FOPDT) process which has been selected to test the effectiveness of the control system using the TrueTime simulator.

$$G_p(s) = \frac{1}{2\,s + 1}\,e^{-2\,s} \quad (2)$$

The TrueTime network has been configured for Ethernet protocol. The equivalent time-discrete controller has been found using the Euler backward approximation and a sample time $Ts = 0.01\,s$. The controller execution time is set 0.0002 s. The PI parameters are found using the AMIGO tuning rules as $K = 0.65, T_i = 2.1818$. For the Smith predictor, the model parameters were chosen equal to those of the process. Finally, to test the system subject to disturbances, a step disturbance signal which amplitude is 0.4, is introduced at t = 25 s.

Fig.1 shows the closed-loop response without delays. The solid line shows that the output can reach the desired value in a few seconds. Moreover, it responds quickly to the disturbance demonstrating a good prediction of the output. The performance of the design has been measured with the ITAE criterion as shown in Table I.

Next, the delays and dropouts have been set in TrueTime simulator. To simulate the effect of the time delay, an interfering node sending disturbing traffic over the network is implemented with an occupation of the 47% of the network bandwidth. The dropouts have been set up through a loss
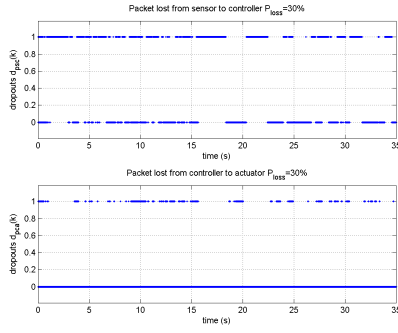
Fig. 2. Time instants of data dropouts $P_{loss} = 30\%$

Fig. 3. System outputs for adaptive IMC

probability $P_{loss} = 30\%$. The time instants of dropouts from sensor-to-controller $d_{psc}(k)$ and from controller-to-actuator $d_{pca}(k)$ are depicted in Fig. 2. From the simulation, the mean values for time delay from controller-to-actuator and from sensor-to-controller are $\tau_{ca} = 0.5\,s$ $\tau_{sc} = 1\,s$, respectively.

The dashed lines in Fig. 1 represent the scenario with delays. Although the process has a larger rise time, no overshoot and zero steady error are observed. On the other hand, it can be seen that the controller returned the system smoothly to the set point after the application of the disturbance. ITAE criterion returned a value of $J_r = 39.09$ for the servo control and $J_d = 62.34$ for the regulatory control. This value is bigger than the previous scenario ($J_r = 20.35$ and $J_d = 56.07$) which demonstrates the adverse effect of the delay in the NCS.

B. Adaptive IMC for NCS

An adaptive Internal Mode Control (IMC) algorithm is created to address the challenge of compensation of time delay and dropouts in NCS. A recursive least squares estimator is implemented to estimate the discrete process model on-line and adapt it during every sampling period. The model and a filter are used for the design of the IMC controller.

The closed-loop transfer function for this system is:

$$\frac{y(s)}{r(s)} = \frac{Q(s)e^{-\tau_{ca}s}G(s)}{1+Q(s)e^{-\tau_{ca}s}[G(s)-G_m(s)]e^{-\tau_{sc}s}} \quad (3)$$

where $Q(s)$, $G(s)$, is the function of the controller and the process respectively. $G_m(s)$ is the internal model of the process. The model can be represented as $G(s) = G_{m+}(s)\,G_{m-}(s)$, where G_{m-} is the invertible part of the process model and G_{m+} is the non-invertible part.

The design of the controller is performed by the cancellation of the invertible part of the process model and the addition of a filter. Thus, the transfer function $Q(s)$ is:

$$Q(s) = G_{m-}(s)F(s) \quad (4)$$

The typical transfer function of the filter is:

$$F(s) = 1/(\lambda s + 1)^n \quad (5)$$

where λ is the time constant of the filter and it is used to tune the closed-loop response. The value of n is chosen to obtain an appropriate transfer function.

According to (3) when $G(s) \neq G_m(s)$ the closed-loop transfer function becomes:

$$\frac{y(s)}{r(s)} = \frac{G_{m-}^{-1}(s)F(s)e^{-\tau_{ca}s}G(s)}{1 + G_{m-}^{-1}(s)\,F(s)e^{-\tau_{ca}s}[G(s) - G_m(s)]e^{-\tau_{sc}s}} \quad (6)$$
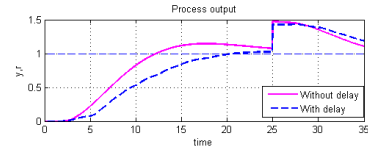
Equation (6) shows that the stability of the system decreases due to variations of process parameters. Thus, the IMC structure is improved with the combination of an adaptive algorithm. The essential part of the adaptive system is the identification of the process parameters. The identification algorithm is implemented using a recursive least squares estimation for discrete systems.

The process example used in (2) is considered here. The *forgetting factor* is set as 0.98, $Ts = 0.5\,s$ and a desired closed-loop time constant as $\lambda = 0.96$. The response of the system is depicted in Fig. 3. There is a large rising time as a result of the large value of λ.

In addition, network delays and dropouts have been configured in the same way as described in previous section and the results for the adaptive IMC controller are shown in Fig.3.

According to the simulation, the proposed method can tolerate the percentage of dropouts without becoming unstable. However, the response is very slow and it has a poor recovery after the application of the disturbance. The sluggish in the output is expected since the time constant value of the filter is high. Although some simulations were performed using smaller values the system became unstable, therefore, the slow response was preferred.

C. A design of robust PID controller using gain/phase margin

The study considers the implementation of a robust PID controller for a first order system subject to an uncertain time delay. Aiming this, the method in [2] is followed in this work. The characteristic equation of the system is modified by adding a gain-phase margin tester function. A set of stability equations is defined to find the desired gain margin and phase margin boundaries that are represented in a parameter plane. Then, the PID parameters that guarantee the required margins are obtained from the resulting admissible region in that plane.

Consider the open-loop transfer function as:

$$G_0(s) = N(s)/D(s) \quad (7)$$

where $N(s)$, $D(s)$ stands for the numerator and denominator, both polynomials function of s. By letting $s = j\omega$ and writing it in terms of magnitude A and phase ϕ, this is equivalent to:

$$D(j\omega) - \frac{1}{|G(j\omega)|\,e^{j\phi}}N(j\omega) = 0 \quad (8)$$

Define: $A = \frac{1}{|G(j\omega)|}$ and $\theta = \phi + 180\,°$. If A= 1 means that θ is the phase margin of the system and $\theta = 0$ results in A becoming the gain margin. Therefore, the gain and phase margin can be determined using the characteristic equation of the system with a gain-phase margin tester.

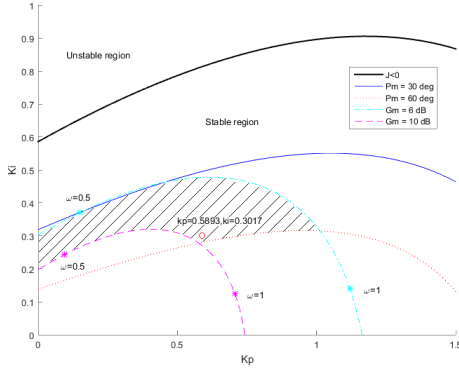$$F(j\omega) = D(j\omega) + Ae^{-j\theta}N(j\omega) = 0 \quad (9)$$

Fig. 4. $K_p - K_i$ plane

Consider the process in (2) and the PID controller with parallel structure:

$$C(s) = K_p + K_i/s + K_D s \qquad (10)$$

where K_p, K_i, K_D are the proportional, integral and derivative gains, respectively.

Substituting the previous equation and the PID parallel form in (9) the resulting characteristic equation is:

$$F(s) = 1 + Ae^{-j\theta}\left(K_p + \frac{K_i}{s} + K_D s\right)\left(\frac{1}{2}\frac{e^{-Ts}}{s+1}\right) = 0 \quad (11)$$

where T = 2+ τ, τ stands for the uncertain delay of the network and 2 s is the dead time of the process. The network delay is consider as $\tau = \tau_{sc} + \tau_c + \tau_{ca}$. τ_c is the controller execution time. By resorting the stability equations in (11) and letting K_D to be constant the controller parameters are defined by:

$$\begin{aligned} K_P &= (C_1 D_2 - C_2 D_1)/(B_1 C_2 - B_2 C_1) \\ K_I &= (D_1 B_2 - D_2 B_1)/(B_1 C_2 - B_2 C_1) \end{aligned} \qquad (12)$$

$$\begin{aligned} where: \quad & B_1 = -A cos(\theta_1)\omega, \quad C_1 = A sin(\theta_1), \\ & D_1 = -w - A sin(\theta_1)K_D\omega^2, \quad B_2 = A sin(\theta_1)\omega, \quad (13) \\ & C_2 = A cos(\theta_1), \quad D_2 = -2\omega^2 - A cos(\theta_1)K_D\omega^2 \end{aligned}$$

Using (12) and performing the same procedure as in [2] the locus is plotted and shown in Fig. 4. Initially, T is set 2 s and K_D is fixed as 0.1.

The first boundary, is found by setting $A = 1$ and $\theta = 0°$. The stability region has been marked in the figure. The boundaries for constant margins are also plotted.

As depicted in 4, the point $K_P = 0.5893, K_i = 0.3017$ is selected to guarantee a phase margin at least of $30°$ and a gain margin at least of 6 dB.

The system response to a step input for the closed-loop systems can be found in 5. It can be seen that the system has a good performance and good rejection to the disturbance. The margins are $PM = 56.5°$ and $GM = 9.51\ dB$. Therefore, the relative stability of the system fulfils the specified criteria.

Furthermore, the value of T have been changed to $T = 2.5\ s$, $T = 3\ s$ and $T = 3.5\ s$. This variation will cover the mean values of the network delay obtained in section A. The resultant intersection area is displayed in Fig. 6. The shaded region represents the admissible parameters for $K_p - K_i$ that will guarantee at least a phase margin of $30°$ and a gain
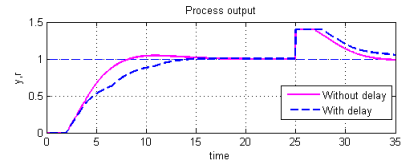

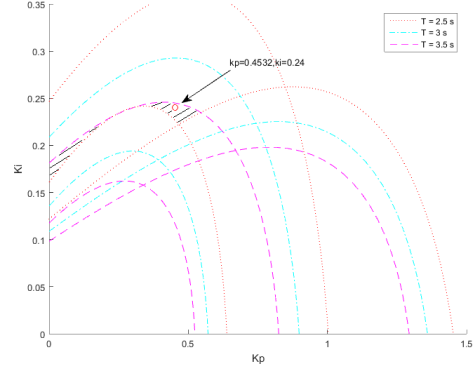Fig. 5. System outputs for robust PID

Fig. 6. $K_p - K_i$ plane for different values of time delay

margin of 6 dB. Based on this plot, a point of $K_p = 0.4532$ and $K_i = 0.24$ is selected for the design of the robust PID controller.

Finally, the system is tested using the TrueTime simulator. $Ts = 0.01s$. Results are shown in Fig. 5.

The system presented a sluggish response compared with the output without delays due to the presence of the delays and dropouts. In general terms, there is a good disturbance rejection and good enough set point tracking.

D. A design of an optimal PID controller for NCS with time-varying delays

An unconstrained optimisation problem is proposed and solved to find the parameters of a PID controller that minimises a cost function when the system has time-varying delays and random delays. The approach in [3] is further studied. Consider the PID controller with parallel structure:

$$u(t) = k_p e(t) + k_i \int_0^\tau e(\tau)d\tau + k_d \frac{de(t)}{dt} \qquad (14)$$

This is approximated to a discrete-time PID by using a backward approximation, a sampling time Ts and a filter for the derivative part. The algorithm is given by:

$$\begin{aligned} u(k) &= K_p\, e(k) + i(k-1) + K_i\, Ts\, e(k) + \frac{K_d}{K_d + K_p NTs} \\ & d(k-1) + \frac{K_P K_d N}{K_d + K_p NTs}[y(k-1) - y(k)] \end{aligned} \qquad (15)$$

where N is the filtering constant. N is selected to be a fraction of the derivative time constant T_d. To simulate the effect of the time-varying delay, the delay distribution of the network has been approximated by a gamma distribution. The gamma function Γ is defined as follows:

$$\Gamma(k) = \int_0^\infty x^{k-1}e^{-x}dx, \ k \in (0, \infty) \qquad (16)$$
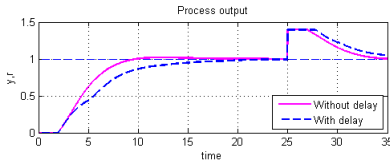
Fig. 7. Systems outputs for optimal PID

Fig. 8. System outputs for optimal robust PID

The general gamma distribution with shape parameter k and scale parameter b is given by:

$$f(x) = b^{-k}/\Gamma(k) \ x^{k-1}e^{-x/b}, \ x \in (0, \infty) \qquad (17)$$

The parameters of the gamma distribution were identified with properties of the network. k is the number of hops between the first and last node and the rate parameter is defined as $1/b = k/T$, where T is the mean delay. The delay with gamma probability distribution is generated by an S-function created in Matlab, that gives the value of the time delay each sample time.

The tuning of the PID controller is obtained by solving an optimisation problem. This is based on minimizing a cost function J. The Optimization Toolbox is used to generate the cost function and find the minimum value. The function *fminsearch* is selected since it finds the local minimum of the optimisation criterion J. The minimization cost criterion is chosen to be the ITAE (Integral of Time-weighted Absolute Error). This cost is given by:

$$J_{ITAE} = \int_0^\infty t \, |e(t)| \, dt = \int_0^\infty t|y_r(t) - y(t)|dt \qquad (18)$$

where $e(t)$ is the signal error, $y_r(t)$ is the reference signal and $y(t)$ is the system output.

1) Numerical example: Consider the example given in (2). Tuning by optimisation is carried out by running a script that calls the functions implemented in Matlab. For $k = 3$ and $T = 0.135 \ s$, the optimal results for the process are: $K_p = 0.4237$, $T_i = 1.5503 \, s$ and $T_d = 0.1106 \, s$. After some tests the value of $N = 10$ is selected for a good response.

Simulations using the TrueTime toolbox and a sampling time of $Ts = 0.008 \, s$ are depicted in Fig. 7. The limitation of the optimal tuning is that it might take several iterations to find the local minimum.

E. A design of an optimal robust PID controller using the maximum sensitivity

A constrained optimisation problem is proposed and solved to find an optimal robust PID controller that guarantees the robustness of the system subject to time-varying delays. The robustness is studied using the maximum sensitivity of the system.

1) Constrained optimisation: The work in [4] proposes a tuning of discrete-time PID controllers in which the parameters are found by solving an optimisation problem where the desired gain and phase margin are set as constraints of the problem. A simil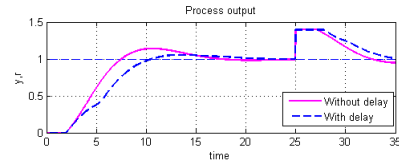ar method is followed in this study, but the maximum sensitivity value is used since the complexity of the computation is reduced significantly.

The maximum sensitivity is given by:

$$M_s = max_\omega |S(j\omega)| = max_\omega |1/(1 + G_{ol}(j\omega))| \qquad (19)$$

This equation defines the sensitivity of the system S and by limiting its maximum value, good robustness of the system can be achieved. The lower the value of Ms, the better the robustness. The PID controller is implemented using (15). The constrained optimisation problem is formulated as:

$$\min \ f(x) = \int_0^\infty t|y_r(t) - y(t - \tau(t), x)|dt$$

$$\text{s.t.} \qquad g(x) = \varnothing, \qquad h(x) = \begin{cases} -x + \varepsilon \le 0 \\ M_s - 1.4 - \varepsilon \le 0 \end{cases} \qquad (20)$$

$$x = [K_p \quad K_i \quad K_d]^T \in \mathbb{R}^n$$

The minimization cost criterion is chosen to be the ITAE. Considering constraints, firstly, the PID controller parameters have to be positive. Secondly, the robustness is guaranteed if there is at least a maximum sensitivity of $Ms = 1.4$ [5]. Defining a small positive value ε the inequality constraints are arranged in the general formulation form.

To simulate the effect of the time delay a Gaussian distributed random delay with mean $\mu = 1$ and variance $\sigma^2 = 0.1$ is chosen. Every sampling time, the optimisation algorithm evaluates the cost function subject to the constraints calculated with the random delays.

SQP is selected to solve the problem and find the controller parameters. The Optimization Toolbox of Matlab is used. A block diagram of the closed-loop system is implemented in Simulink. In particular, the function *fmincon* was used to find the minimum of the cost function.

2) Numerical example: For the process shown in (2), an optimal PI controller is studied for a random delay. N is selected with a constant value 10. The optimal results for a $Ts = 0.03 \ s$ are: $Ti = 3.7217$ and $Kp = 0.3201$. The tests for the TrueTime are performed now. Fig. 8 shows that the responses have a small overshoot for the simulator with or without delays.

According to Table I, the optimal PI controller showed a good performance and robustness to the time-varying delay and dropouts. However, the optimisation tuning requires several iterations to find the optimal parameters.

F. A design of a jitter-aware PID for NCS with time-varying delays

To address the adverse effect of time-varying delays in NCS, a robust PID controller has been implemented for a first order
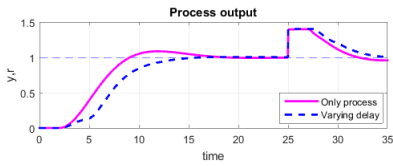
Fig. 9. System outputs for jitter-aware PID

system. A method in [6] has been used in this section where the AMIGO tuning rules are combined with the maximum time delay that the system can tolerate. A set of tuning equations gives the PID parameters that guarantee the robustness of the system. Consider the following version of the PID:

$$u = k\left(by_r - y_f\right) + k_i \int_0^t (y_r - y_f)\,d\tau + k_d\left(c\frac{dy_r}{dt} - \frac{dy_f}{dt}\right) \quad (21)$$

where b and c are the set-point weighting factors. Y_f is the output after the measurement filter.

1) Jitter margin: The variance of the time delay is studied under the concept of jitter margin. It can be defined as the maximum time-varying delay that can be increased in the system without causing instability. Consider a linear time invariant system with process $P(s)$ and controller $C(s)$. The control system is perturbed by an uncertain time-varying delay Δ in the feedback loop. The system is stable for any time varying delay defined by:

$$\Delta(v) = v(t - \delta(t)), \quad 0 \le \delta(t) \le \delta_{\max} \quad if \quad (22)$$

$$\left|\frac{P(j\omega)C(j\omega)}{1 + P(j\omega)C(j\omega)}\right| < \frac{1}{\delta_{\max}\,\omega}, \quad \forall\,\omega \in [0,\infty[\quad (23)$$

where $\delta_{\max}\,\omega$ is the maximum jitter margin.

2) New tuning rules: The PID parameters for the jitter-aware controller are found by solving an optimisation problem where the robustness of the AMIGO rules and the jitter margin are maximised. The resultant tuning rules for a FOPDT process are proposed as follows:

$$k = \frac{1}{K_p}\left[\frac{0.4T - 0.04}{L} + 0.16\right]$$

$$k_i = \frac{1}{100K_p}\left[\frac{-0.11T^3 + 1.5T^2 - 1.5}{L^2} + \frac{0.35T^2 + 4T + 50}{L}\right] \quad (24)$$

$$k_d = \frac{1}{100K_p}(0.4T^2 + 11T)$$

where L is the dead time and T is the time constant of the process. The controller gains will be set using these rules, and the remaining parameters will use the AMIGO rules.

3) Numerical example: Consider the first order system given in (2). A time constant $T_f = 0.2\,s$ is used for the filter, b = 0 and c = 0. Applying (24), the controller parameters are $k = 0.54$, $k_i = 0.3061$ and $k_d = 0.236$. The implementation of the system using the TrueTime Simulator follows the same configuration than previous tests. The first test has been made for the process without time-varying delays. Fig. 9 shows the good performance of the system with a small overshoot.

The response for the time-varying delays shows a slower and damped response.

G. A design of an optimal immune PID controller for NCS

In this section, an optimal immune PID controller is applied to a NCS subject to dropouts and time-varying delays. The immune feedback law proposed in [7] is used in this work combined with an optimisation problem to find the parameters of the immune PID controller.

The immune control system is a physiological action that produces antibodies to combat antigens. The primary components of this system are the recognition cells and the killing cells. When the antigens arrive, recognition cells begin to multiply themselves at the same time they activate the helper T cells (TH). Then, the helper T cells activate B cells, which secrete the antibodies. APC can also activate the suppressor T cells (TS), which can suppress the secretion of the helper T cells and the B cells. It can be generalised that the immune feedback algorithm is mainly based on the feedback regulating principle of T cell. The principle is as follows: $\varepsilon(k)$ is the amount of antigens at the kth generation and it is defined by:

$$\varepsilon(k) = \gamma\varepsilon(k - 1) - u_{kill}(k - d) \quad (25)$$

where ε is the antigen concentration, u_{kill} is the concentration of the B cells and d is the postmortem interval or the delay time of immune response. The concentration of the B cells can be expressed as:

$$u_{kill}(k) = TH(k) - TS(k) \quad (26)$$

$TH(k)$ is the output from TH stimulated by the antigens:

$$TH(k) = K_1\varepsilon(k) \quad (27)$$

where K_1 is the stimulating factor of TH $TS(k)$ the effect of TS cells on the B cells. The action from restraining B cell using T cell is given as:

$$TS(k) = K_2 f[\Delta u_{kill}(k)]\varepsilon(k) \quad (28)$$

where K_2 is a suppression factor of TS cell and $\Delta u_{kill}(k) = u_{kill}(k - d) - u_{kill}(k - d - 1)$. This is the concentration change of the B cells. Finally, $f(\cdot)$ is a non-linear function. Then, mathematical representation of the concentration of B cells is expressed as:

$$u_{kill} = K_1\varepsilon(k) - K_2\{f[\Delta u_{kill}(k)]\}\varepsilon(k) \quad (29)$$

By selecting the amount of the antigens, $\varepsilon(k)$ as the control error, $e(k)$ and the total stimulation received by B cells, u_{kill} as the control input $u(k)$, the immune feedback law can be describe as this:

$$u(k) = k\{(1 - \eta f[\Delta u(k)])\}e(k) \quad (30)$$

where $k = K_1$, $\eta = K_2/K_1$ Therefore, the immune PID control algorithm can be described by (15), where the proportional gain is: $K_{pl} = K(1 - \eta f[\Delta u(k)])\,K_p$. The parameter K is used to control the response speed, and the parameter η is used to control the stabilization effect. The function $f(\cdot)$ is selected as:

$$f[\Delta u(k)] = 1 - 2/(e^{-a\,\Delta(u)} + e^{a\,\Delta(u)}), \quad a > 0 \quad (31)$$

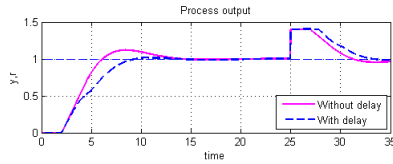where a is the factor of antibodies concentration.

Fig. 10. System outputs for optimal immune PID

TABLE I
PERFORMANCE EVALUATION

Control Algorithm	J_r	J_d	RT (s)	OS (%)	Ms	GM (dB)	PM (∘)
1. Jitter-aware PID	32	59	7	1	5	3	20
2. Optimal robust PI	25	65	6	5	1	11	77
3. Optimal PID	29	66	6	0	4	3	22
4. Optimal immune PID	15	54	4	2	3	4	29
5. Smith predictor (PI)	39	62	5	0	1	25	62
6. Adaptive IMC	65	115	9	3	1	18	Inf.
7. Robust PID	22	64	5	1	3	4	33

1) Numerical example: Consider the first order system with time delay process in (2). To determine the values of η, a, K and the PID parameters, a constrained optimisation problem has been solved using *fmincon* to find the minimum value for the function J. The cost function J has been selected as the ITAE criterion. The time-varying delay has been approximated by a Gaussian distributed random signal, with mean one and variance 0.1. $N = 10$ and sample time $Ts = 0.015\,s$.

The optimal results are: $K_p = 0.0812, T_i = 0.4105\,s, T_d = 0.9716\,s$, $a = 0.0567$, $\eta = 3.1767$ and $K = 6.7486$. The system is tested using the TrueTime simulator.

In Fig. 10 the closed-loop response for the system when no time delays are presented. The simulation shows that the control action of the optimal immune PID brings the system smoothly to the set point with no overshoot. It rejects the disturbance applied at time $t = 25$. The results of the experiment with time delays showed a slower response compared with the output without time delays. This is because the controller had to compensate the lack of information. In general terms, there is a good disturbance rejection and good enough set point tracking despite the presence of time delays and dropouts.

II. CONCLUSION

Seven control methods for high dropouts and time-varying delays are studied in this work. Fig. 11 shows the comparison of these methods. Results also shows performance evaluation based on ITAE cost function values for servo (J_r) and regulatory control J_d. Rising time (RT) and maximum overshoot (OS) are also given. The steady state errors are small except for the Smith predictor and the Adaptive IMC which values are 0.024 and 0.027, respectively. Considering robustness, the following values are presented: the maximum sensitivity (Ms), gain and phase margin. All these robustness indices are summarised in Table I.

The methodology of the optimal immune PID offers the best performance and also a good robustness. Moreover, the
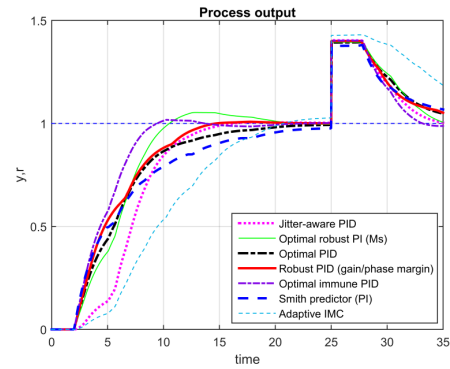

Fig. 11. Comparison all methods

difficulty of choosing the adequate values has been overcome with optimisation method.

The jitter-aware PID and the robust PID shows less robustness than the other robust methods studied here. The only limitation of the robust PID is that robustness of the proposed tuning method is not sustained for long time-varying delays.

From the optimal controllers, the optimal PID has one of the best performances. It is also robust when optimally tuned only in performance sense. On the other hand, the optimal robust PI presented good margins of robustness and guaranteed $Ms = 1.4$.

From the model based controllers, the Smith predictor gave the best performance with faster rising time and lower overshoot. It presents a good robustness too. Therefore, there is a good prediction for slow changes in the network. On the contrary, the adaptive IMC presented the poorest performance and robustness. This is a result of a sluggish control signal which does not yield good control action under the adverse network conditions.

REFERENCES

[1] K. J. Åström and T. Hägglund, "Revisiting the Ziegler-Nichols step response method for PID control," *Journal of Process Control*, vol. 14, pp. 635–650, 2004.
[2] Y. J. Huang and Y.-J. Wang, "Robust PID controller design for non-minimum phase time delay systems," *ISA Transactions*, vol. 40, pp. 31–39, 2001.
[3] H. K. 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)*. Paris, France: International Institute of Electrical and Electronics Engineers (IEEE), 2006, p. 6.
[4] H. N. K. 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)*. Espoo, Finland: International Institute of Electrical and Electronics Engineers (IEEE), 2005, p. 6.
[5] V. M. Alfaro, R. Vilanova, and O. Arrieta, "Maximum sensitivity based robust tuning for two-degree-of-freedom proportionalintegral controllers," *Industrial & Engineering Chemistry Research*, vol. 49, no. 11, pp. 5415–5423, 2010.
[6] L. Eriksson and M. Johansson, "Pid controller tuning rules for varying time-delay systems," in *American Control Conference, 2007. ACC '07*, July 2007, pp. 619–625.
[7] P. Yang, H. Zhao, Y. Zhou, and Z. Liu, "Study of Immune PID-PI Controller for FG-3000 Temperature Control System," pp. 343–347, 2007.