Online Tuning of PID Controller for Linear SISO System with Random Communication Delay by Using Genetic Algorithms

Adrian-Bogdan Hanchevici, Ioan Dumitrache

Faculty of Automatic Control and Computers, University "Politehnica" of Bucharest
313 Splaiul Indepdendentei, 060042 Bucharest, Romania

(e-mail: adrian.hanchevici@upb.ro, idumitrache@ics.pub.ro)

Abstract: In this paper we propose a control strategy for linear SISO systems in the presence of random communication delay. We have included the communication delay in the process model. This control strategy makes use of the several process models and based on the differences between each of them and real process, the PID controller is tuned. The tuning is made online, while the system is operational and for this are used the genetic algorithms. Simulation and implementation results are presented. The results show great performance improvement when the online tuning of the PID controller is used. The authors present the implementation for an axis of a 3D crane and using the Ethernet communication network.

Keywords: networked control systems, genetic algorithms, PID controller, linear systems, 3D crane.

1. INTRODUCTION

Networked Control Systems (NCS) are a new type of hybrid systems. They are feedback control systems, where the information from the sensors about the controlled parameters and the commands for the actuators are exchanged through a communication network like Ethernet, CAN, wireless.

In the next figure is presented the general architecture of a NCS (Aubrun et al. (2010)).

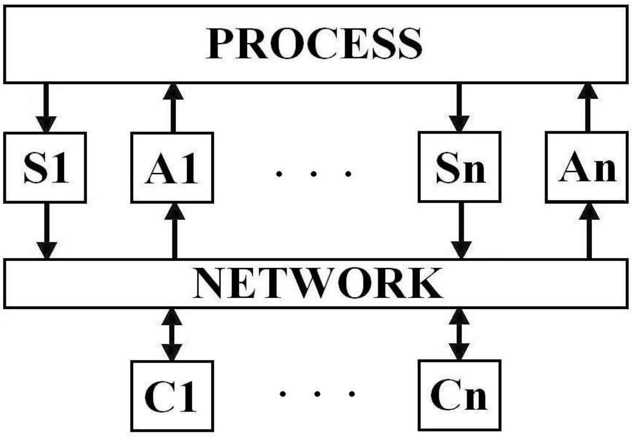

Fig. 1. The architecture of a Networked Control System.

In the above figure, S1 to Sn represents the sensors, A1 to An are the actuators and C1 to Cn the controllers. There is not compulsory to be as many controllers as actuators or sensors. The main problems of this hybrid systems are the random communication delays and package lose (information lose).

In Mazo et Tabuada (2011) and Lehmann et Lunze (2011) are presented methods to reduce the traffic through the network and so it drops the possibility to appear communication delays. For this purpose, event-based control is used, which means that the control signals are not modified until a condition is met.

Other methods are using an adaptation of the control strategy, also known as QoC (quality of control) according to the communication delay. Also, can be adapted the parameters of the network (QoS – quality of service) by analyzing the performances of all closed-loop systems that are sharing the network (Aubrun et al. (2010)).

When we want to control a process with a variant model we can use the multi-model control strategy. This solution has the advantage of breaking a difficult problem into several easy to solve problems (Athans et al. (2005); Dumitrache (2010c); Patrascu et Hanchevici (2011); Toscano (2007)).

The adaptive control systems represent another strategy that is used when the process has structural and parametric modifications. In this case, the controller has the capability to identify the modifications of the process and to tune his parameters accordingly (Aström et al. (1989); Dumitrache (2010a,b,c); Gang Tao (2003); Iserman (1991); Slotine (1993); Thomas et al. (2000)).

Recently were developed new concepts which are using a combination of multi-model and adaptive control. Viable solutions were proposed for the online control of complex processes characterized by nonlinear models (Athans et al. (2006); Athans et al. (2007)).

The paper is organized as follows. The authors start by presenting the proposed control strategy using online genetic algorithms (Section 2). In Section 3, the authors present the experimental setup and the model of the process. In Section 4, the authors present the case study discussed in this paper. The results from simulation are compared with the experimental ones. Finally, Section 5 contains the conclusions and the directions for future research.

2. PROPOSED CONTROL STRATEGY

The control strategy presented in this paper is proposed for linear SISO systems which have random communication delay because of the communication network.

The control strategy is optimized based on genetic algorithms. This optimization approach is used to find the tuning parameters for the PID controller.

In the below figure (Fig. 2) is presented the structure of the control strategy using online genetic algorithms, where: y_k represents the controlled variable, $\hat{y}_{k1}$ is the output provided by the process model 1, $\hat{y}_{k2}$ is the output provided by the process model 2, $\hat{y}_{k3}$ represents the output provided by the process model 3, $\hat{e}_{k1}$, $\hat{e}_{k2}$, $\hat{e}_{k3}$ are the errors between the real output signal and the output of each process model, r_k is the set-point for the controlled variable, u_k is the control signal, $\hat{\Theta}_k$ represents the estimated parameters of the process, $\hat{\Pi}_k$ is the set of the tuning parameters for the PID controller, and J is the performance index to be minimized by the GENETIC ALGORITHM PID TUNER (GAPT) module.

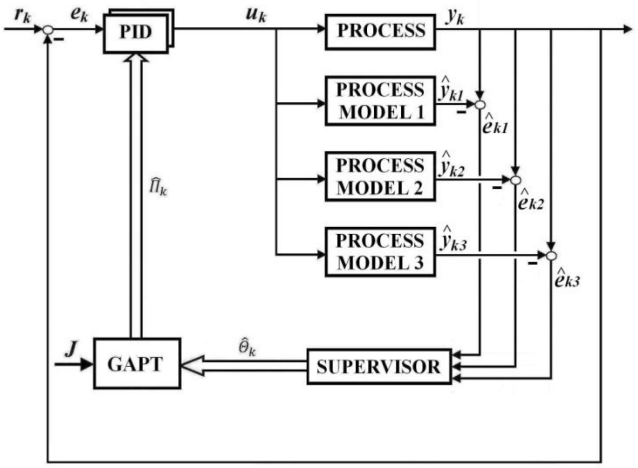

Fig. 2. The control strategy.

The GAPT module has one input variable (estimated parameters of the process), and one output variable (tuning parameters for the PID controller). Also, it uses a performance index to be minimized, which will allow GAPT to stop. GAPT is using the genetic algorithms optimization approach. The purpose is to find the tuning parameters for the PID controller. GAPT uses his performance index to know if the current parameters for the PID controller are convenient. The parameters are considered convenient after the performance index of them is compared with GAPT's performance index. If this value is lower than GAPT's performance index, GAPT will consider the current parameters for the PID controller convenient. It is set offline the way that GAPT determines the performance index of the current estimated parameters. The value for GAPT's performance index is also set offline.

During every sampling time GAPT finds the tuning parameters for the PID controller. This means that there are different parameters for the PID controller every sampling time. By using GAPT during every sampling time, is achieved an online adaptation of the PID controller according to the parameters' variations of the system.

In the first sampling time, the command for the linear SISO system will be provided by an a priori experimental determined PID controller.

Genetic algorithms are based on the evolutionary "survival of the fittest" concept. The problem's possible solutions are thus coded into a population, which is then allowed to "evolve". The passing from one generation to another is obtained by means of natural selection.

Genetic algorithms perform a search on multiple directions in order to find the solution amongst a given population, by evaluation the fitness of each individual. The fitness is a function modeling the objective of the procedure. Its description is problem-specific, but its purpose is to eliminate unfit/unwanted solutions from the gene pool. The fitness function returns a minimization performance index which in turn will be used for the selection of the new generation. The lower the performance index, the higher the fitness of an individual, the closer it is to the solution. The performance index of the entire procedure is given by the fitness of the best population, or, should the case be, by the fitness of the solution.

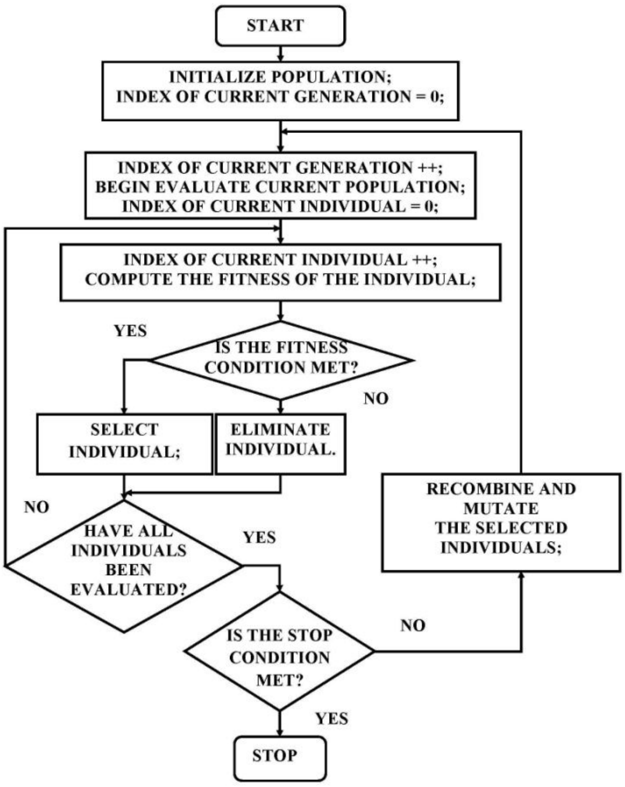

Fig. 3. The structure of genetic algorithms (Dumitrache et Buiu (1999).

In Patrascu, Hanchevici et Dumitrache (2011) is presented a genetic algorithm control of a non-linear MIMO system. The main issue solved in this paper is finding a suitable set of controllers for the considered non-linear system. The results show considerable improvement of the system's performances (settling time, steady stare error, overshoot) after implementing the GA offline module.

In Fig. 3 is presented, more detailed, the structures of GAPT. The structure is presented as a logic diagram. For this module, the solutions are coded as arrays with real values. The length of the array is four. The length of GAPT solution can be four because the solutions represents a PID controller and the four elements of the array can represent the proportional component, the integrative component and the derivative one, and finally the performance index of this solution.

The main steps of the implemented genetic algorithm are presented as follows. The initialization of the algorithm performs a random generation of the first solution pool, taking into consideration the constraint requirements for each element of the array. For each generation, an evaluation is performed, taking into account the fitness of each individual. When an individual is considered unfit (the solution does not meet the controller design performance requirements), it is eliminated from the population. The selected individuals are then recombined into a new generation. When the stop condition is met (for example: reach of a user defined number of generations), the algorithm provides a solution to the problem. Based on a GA performance index, the solution is considered viable or not.

3. EXPERIMENTAL SETUP AND CRANE MODEL

The non-linear SISO system taken into study in this paper is the *Oy* axis of a 3-dimensional (3D) Crane. This electromechanical system has a complex dynamic behaviour, being controlled from a process computer. The 3D Crane setup consists of a payload (lifted and lowered in the *Oz* direction by a motor mounted on a cart) hanging on a pendulum-like lift-line. The cart is mounted on a rail, giving the system capability of horizontal motion in the *Ox* and *Oy* directions. Thus, the payload can move in 3 dimensions. . As actuators, the system makes use of three RH158.24.75 DC motors. Data acquisition is performed by means of five encoders that measure five process variables: the three coordinated of the payload in space and two deviation angles of the lift-line, with a resolution of 4096 pulses per rotation (ppr) for the spatial coordinates and 0.0015 rad for the deviation angles. A break-out box contains an interface module that amplifies the control signals which are transmitted from the PC to the DC motors (InTeCo (2000)).

The 3D Crane is a MIMO system, having three input variables (control signals for the three DC motors) and five output variables (encoder signals). The non-linear characteristic is given by the pendulum-like motion of the payload in space, with a variable line length. Figure 4 (InTeCo (2000)) presents the crane system, where: x_w (not represented) is the distance of the rail with the cart from the center of the construction frame; y_w (not represented) is the distance of the cart from the center of the rail; R is the length of the lift-line; α represents the angle between the *Oy* axis and the lift-line; β represents the angle between the negative direction on the *Oz* axis and the projection of the lift-line onto the *Oxz* plane; m_c is the mass of the payload; m_w is the mass of the cart; m_s is the mass of the moving rail; x_c, y_c, z_c are the coordinates of the payload; $S = F_R - T_R$ represents the reaction force in the lift-line acting on the cart; F_x is the force driving the rail with cart; F_y is the force driving the cart along the rail; F_R is the force controlling the length of the lift-line; T_x, T_y, T_R are friction forces.

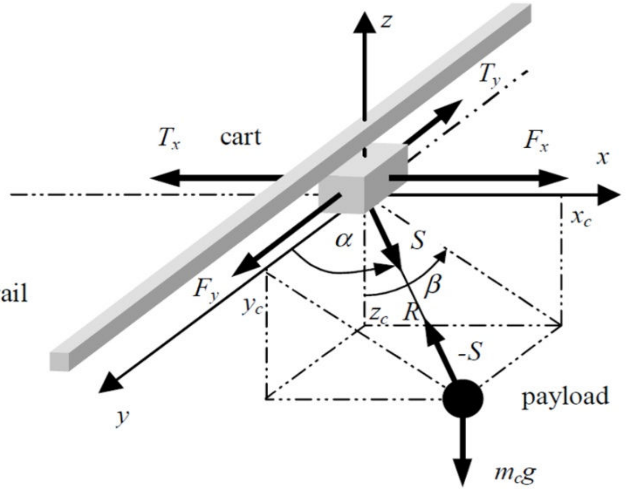

Fig. 4. The crane system (InTeCo (2000)).

If we define the state variables and the relations of them we get a mathematical model as:

$$\begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & s_5^2\dot{\beta}^2 + \dot{\alpha}^2 + Z & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \\ + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & \mu_1 s_5 s_7 \\ 0 & 0 & 0 \\ 1 & 0 & \mu_1 c_5 \\ 0 & 0 & 0 \\ -c_5 & -s_5 c_7 & -(1+\mu_1 c_5^2 + \mu_1 s_5^2 s_7^2) \end{bmatrix} \cdot \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix} \\ \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & s_5 s_7 & 0 \\ 0 & 0 & 1 & 0 & c_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & -s_5 c_7 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \end{cases}$$

where the following notations have been used:

$x_1 = x_w$; $x_2 = \dot{x}_1$; $x_3 = y_w$; $x_4 = \dot{x}_3$; $x_5 = R$; $x_6 = \dot{x}_5$; $\mu_1 = m_c / m_w$; $\mu_2 = m_c /(m_w + m_s)$; $N_1 = (F_y - T_y)/ m_w$; $N_2 = (F_x - T_x)/(m_w + m_c)$; $N_3 = (F_R - T_R)/ m_c$; $s_5 = \sin\alpha$; $c_5 = \cos\alpha$; $s_7 = \sin\beta$; $c_7 = \cos\beta$; g is the gravitational acceleration; $Z = (g s_5 c_7)/R$ is a nonlinear function.

A simulation model of the system was implemented. Details of this model can be found in InTeCo (2000).

4. CASE STUDY

The non-linear SISO system taken into study in this paper can be approximated by the rational s-transfer function:

$$H_P(s) = \frac{B(s)}{A(s)} = \frac{K_P}{s(T_P \cdot s + 1)} \tag{1}$$

where K_P represents the gain, and T_P describes the dynamics of the process. The communication delay is included in the process model and $H_P(s)$ becomes:

$$H_p^{'}(s) = \frac{B(s)}{A(s)} e^{-\tau \cdot s} = \frac{K_P}{s(T_P \cdot s + 1)} e^{-\tau \cdot s} \tag{2}$$

where τ is variant.

We have considered the linear approximation of the process, as presented above. This approximation was made experimentally. After making this linearization, we have tuned the PID controller, by using the genetic algorithms, for the case when there is no communication delay (τ = 0 sec). The tuning was made offline, in a simulated environment, and the performance index was computed according to an a priori defined function:

$$f(t_t, \sigma, e) \tag{3}$$

where t_t is the settling time, σ represents the overshoot of the controlled variable and e is the steady state error. The value of the performance index was also set offline.

After the offline tuning was achieved, the PID controller has been tested, in simulation. We have continued, and considered the communication delay τ = 0.8 seconds and τ = 1.2 seconds. Below, in figure 5, are presented the simulated responses of the system using the same PID controller and different communication delay. By analyzing the three responses, we can see that there is a difference between them in terms of settling time and overshoot, while there is no difference in terms of steady state error.

This mismatch is generated by the fact that when the communication delay appears, the PID controller designed for the case when there is no communication delay, is not able to perform as requested and the closed-loop system's performances are getting worse as the communication delay is getting higher values.

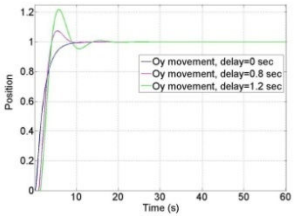

Fig. 5. Simulated responses using the same PID controller and different communication delay.

In order to prevent the decrease of performances for the closed-loop system, we have adopted the control strategy presented in figure 2.

We have considered that the minimum value for the delay is 0 seconds and the highest value is 1.2 seconds and we assumed that we the process can be approximated by three models. The three process models are considered for the case when there is no delay, when the delay is 0.8 seconds and when τ = 1.2 seconds. The process models are:

$$H_{p1}(s) = \frac{K_P}{s(T_P \cdot s + 1)}; \qquad H_{p2}(s) = \frac{K_P}{s(T_P \cdot s + 1)} e^{-0.8s};$$

$$H_{p3}(s) = \frac{K_P}{s(T_P \cdot s + 1)} e^{-1.2s} \tag{4}$$

On every sampling time, the SUPERVISOR (Fig. 2) compares the output of each of the three process models with the output of the real process and by evaluating the errors between the real output signal and the output of each process model, he decides which process model gives the best approximation of the process and the parameters of this process model are passed to the GAPT module for tuning the PID controller.

For evaluation, the SUPERVISOR is using this criterion:

$$I = \min\left(\hat{e}_{k1}^{2}\ ; \hat{e}_{k2}^{2}\ ; \hat{e}_{k3}^{2} \right) \tag{5}$$

The system works with a sampling time of 0.2 seconds.

The process computer used has the following hardware configuration: Intel(R) Core(TM)2 Duo CPU E7200 @ 2.53GHz 2.53GHz, 2.00 GB of RAM. As operating system, it was used Microsoft Windows XP Professional Version 2002 Service Pack 3.

In Table 1 and Table 2 are presented the cases analyzed in this paper. AG Offline S, E represent the performances of the system obtained in simulation and experimentally after the offline tuning of the PID controller was achieved and when was applied a different communication delay.

AG Online (fixed delay) represents the performances of the system when was used the control strategy presented in Fig. 2 and a communication delay appears and remains fixed all the time. We present simulated (S) and experimental (E) results for different values of the communication delay.

Table 1. Simulation study analysis

		Settling Time (s)	Overshoot (%)	Steady State Error (%)
		Oy	*Oy*	*Oy*
AG Offline	τ=0 sec (S)	6.0	0	0
	τ=0.8 sec (S)	6.8	7.3	0
	τ=1.2 sec (S)	8.4	21.7	0
AG Online (fixed delay)	τ=0 sec (S)	6.0	0	0
	τ=0.2 sec (S)	8.4	0	0
	τ=0.4 sec (S)	8.0	0	0
	τ=0.6 sec (S)	7.2	0	0
	τ=0.8 sec (S)	9.4	0	0
	τ=1 sec (S)	8.8	0	0
	τ=1.2 sec (S)	8.2	0.8	0
AG Online (random delay)	S1	8.6	0	0
	S2	7.2	0	0
	S3	9.8	0	0
	S4	7.6	2.1	0

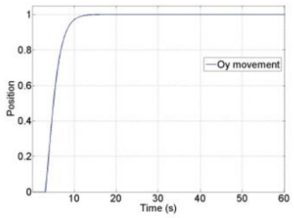

Fig. 6. Simulated response for τ=1 sec (fixed delay) when is performed the online tuning of PID.

AG Online (random delay) represents the performances of the system when was used the control strategy presented in Fig. 2 and the communication delay is changing after every second. Also, we present simulated (S) and experimental (E) results for four cases. In every case, the distribution of the communication delay is different.

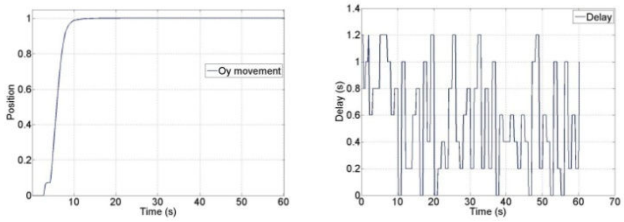

Fig. 7. Simulated response when is performed the online tuning of PID and the random generated delay (for case S1).

In Fig. 6 is presented the simulated response of the system when is performed the online tuning of the PID controller and the communication delay has a fixed value of 1 second.

In Fig. 7 is described the simulated response of the system when is performed the online tuning of the PID controller and the communication delay is changing every second and is presented the random generated communication delay for the case S1.

For the experimental study we have considered the Ethernet communication network. We assumed that the delay is not higher than 0.6 seconds and the three process models are considered for the case when there is no delay, when the delay is 0.4 seconds and when τ = 0.6 seconds. The process models are:

$$H'_{p1}(s)=\frac{K_P}{s(T_P\cdot s+1)}; \qquad H'_{p2}(s)=\frac{K_P}{s(T_P\cdot s+1)}e^{-0.4s}; \tag{9}$$

$$H'_{p3}(s)=\frac{K_P}{s(T_P\cdot s+1)}e^{-0.6s} \tag{6}$$

Table 2. Experimental study analysis

		Settling Time (s)	Overshoot (%)	Steady State Error (%)
		Oy	*Oy*	*Oy*
AG Offline	τ=0 sec (E)	2.4	0	0
	τ=0.4 sec (E)	3.2	8.2	0
	τ=0.6 sec (E)	5.4	18.5	0
AG Online (fixed delay)	τ=0 sec (E)	2.4	0	0
	τ=0.2 sec (E)	2.6	0	0
	τ=0.4 sec (E)	3.2	0	0
	τ=0.6 sec (E)	3.2	0	0
AG Online (random delay)	E1	2.8	0	0
	E2	2.2	0	0
	E3	2.6	4.8	0
	E4	2.6	4.4	0

As in simulation, also in experimental environment, the PID controller designed for the case when there is no communication delay, is not able to perform as requested and the closed-loop system's performances are getting worse as the communication delay is getting higher values. The experimental results are presented in Fig. 8.

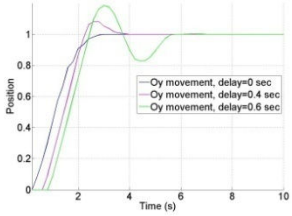

Fig. 8. Experimental responses using the same PID controller and different communication delay.

In Fig. 9 is presented the experimental response of the system when is performed the online tuning of the PID controller and the communication delay has a fixed value of 0.4 seconds.

In Fig. 10 is described the experimental response of the system when is performed the online tuning of the PID controller and the communication delay is changing every second and is presented the random generated communication delay for the case E1.

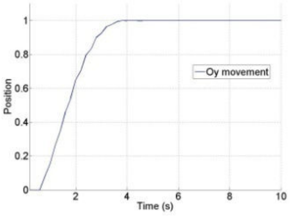

Fig. 9. Experimental response for τ=0.4 sec (fixed delay) when is performed the online tuning of PID.

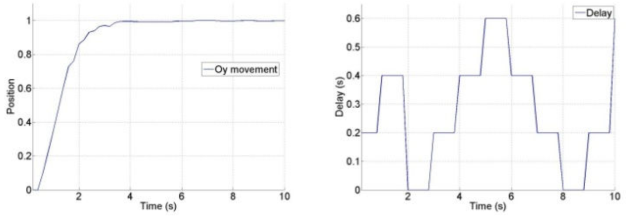

Fig. 10. Experimental response when is performed the online tuning of PID and the random generated delay (for case E1).

By analyzing the responses presented in Fig. 5, Fig. 6, Fig. 7 and in Fig. 8, Fig. 9, Fig. 10 we can see that the performances of the closed-loop system are improved when is performed the online tuning of the PID controller, in terms of overshoot, while there is no difference in terms of steady state error.

5. CONCLUSIONS

In this paper we have considered a control strategy for linear SISO systems with random communication delay. We presented how genetic algorithms are used to tune the parameters of the PID controller on every sampling time. This control strategy was implemented using the Java programming language, the process was considered an axe of a 3D Crane and we used the Ethernet communication network.

The experimental and implementation results show that the best performances were obtained when we used the online tuning of the PID controller, rather than using the same PID controller all the time. This means that the genetic algorithms are a very good method to achieve the online adaptation for the PID controller.

This control strategy is different from all other existing techniques of this type by the simple and elegant way that the PID controller is made suitable to control a linear process with random communication delay and the increased speed of the adaptation processes.

In the future work we will address the problem of online adaptation of multiples PID controllers for a MIMO system with random communication delays and to increase the upper bound of the delay. For study we will consider all the three axes of the 3D Crane. To achieve this goal we will reconsider the genetic algorithms from GAPT module in order to perform faster and to make a more complex search. Also, we will make a comparative study between the control strategy presented in this paper and the robust PID design.

ACKNOWLEDGEMENTS

The work has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/88/1.5/S/61178.

Fig. 4 and the non-linear crane model are presented with the courtesy of INTECO.

REFERENCES

Aström K. J., Wittenmark B. (1989). *Adaptive Control*. Addison-Wesley, Reading, Mass.

Athans, M., Fekri, S., Pascoal, A. (2005). Issues on Robust Adaptive Feedback Control. *Proceedings of 16th IFAC World Congress,* Prague, Czech Republic, pp. 9–39.

Aubrun, C., Simon, D., Song, Y. (2010). *Co-design Approaches for Dependable Networked Control Systems*, John Wiley&Sons Inc, USA.

Dumitrache, I. (2010a). *Ingineria Reglării Automate. Volumul 1.* Editura Politehnica Press, Romania

Dumitrache, I. (2010b). *Ingineria Reglării Automate. Volumul 2.* Editura Politehnica Press, Romania.

Dumitrache, I. (2010c). *Automatica. Volumul 1*. Editura Academiei Romane, Bucuresti, Romania.

Dumitrache, I., Buiu, C. (1999). *Algoritmi genetici*. Editura Nemira, Cluj-Napoca, Romania.

Fekri, S., Athans, M., Pascoal, A. (2006). Issues, Progress and New Results in Robust Adaptive Control. *International Journal of Adaptive Control and Signal Processing*, Vol. 20, pp. 519-579.

Fekri, S., Athans, M., Pascoal, A. (2007). Robust Multiple Model Adaptive Control (RMMAC): A Case Study. *International Journal of Adaptive Control and Signal Processing*, Vol. 21, pp. 1-30.

Gang Tao (2003). *Adaptive Control Design and Analysis*, Wiley-Interscience, John Wiley&Sons Inc..

InTeCo (2000). *3D Crane*. InTeCo Ltd., Krakow, Poland.

Iserman, R. (1991). *Digital Control Systems. Volume 1I.* Springer-Verlag Berlin Heidelberg, Germany.

Lehmann, D., Lunze, J. (2011). Event-based control with communication delays. *Proceedings of 18th IFAC World Congress*, Milano, Italia, pp. 3262-3267

Mazo, M., Tabuada, P. (2011). Decentralized Event-triggered Control Over Wireless Sensor/Actuator Networks, *IEEE Transactions on Automatic Control*, pp. 2456-2461.

Patrascu, M., Hanchevici, A. B., (2011). *Sisteme Avansate de Conducere. Indrumar de laborator*, Editura Politehnica Press, Bucuresti, Romania.

Patrascu, M., Hanchevici, A. B., Dumitrache, I. (2011). Tuning of PID Controllers for Non-Linear MIMO Systems Using Genetic Algorithms*, Proceedings of the 18th IFAC Congress*, Milano, Italy, pp. 12644-12649.

Slotine J. J. E.(1993). *Nonlinear Control Systems*, Prentice Hall.

Thomas L. V., Grantham J. W. (2000). *Nonlinear and Optimal Control Systems*, John Wiley&Sons Inc..

Toscano, R. (2007). Robust synthesis of a PID controller by uncertain multimodel approach, *Information Sciences*, 177, 1441–1451.