

CAN- based networked control systems: A compensation for communication time delays

Nguyen Trong Cac¹, Nguyen Xuan Hung², Nguyen Van Khang¹

¹School of Electronics and Telecommunications, Hanoi University of Science and Technology, Hanoi, Vietnam

²R&D Department, UINT Company, Saint Aubin, France

Email address:

cac.nguyentrong@hust.edu.vn (N. T. Cac), xuanhung.nguyen84@gmail.com (N. X. Hung),

khang.nguyenvan1@hust.edu.vn (N. V. Khang).

To cite this article:

Nguyen Trong Cac, Nguyen Xuan Hung, Nguyen Van Khang. CAN- Based Networked Control Systems: A Compensation for Communication Time Delays. *American Journal of Embedded Systems and Applications*. Vol. 2, No. 3, 2014, pp. 13-20.

doi: 10.11648/j.ajesa.20140203.11

Abstract: In the context of Networked Control Systems (NCS), communication time delays strongly influence on the Quality of Control (QoC) of process control applications. The goal of this paper is to propose a way to calculate the closed-loop communication time delay and we compensate this time delay using the pole placement design method in order to improve the QoC for CAN-based NCSs. We consider several process control applications implemented on a CAN network. Then we show the interest of the proposed method by comparing the QoC obtained with and without time delay compensation.

Keywords: Communication Time Delay, Pole Placement Design, CAN Network, Networked Control Systems

1. Introduction

Networked Control Systems (NCSs) are feedback control systems in which controllers, sensors and actuators are interconnected using a common bus network. The NCSs are more and more important and becoming an alternative to point-to-point architecture because of many advantages such as reduction of installation and maintenance cost, more easiness in diagnostic and configurability. However, time delays are induced due to the transmission of frames from controllers to actuators (feed-forward channel) and from sensors to controllers (feedback channel). These time delays degrade the QoC of the control system.

Several works [1-4] considered the compensation for time delays. The works [1] and [2] only consider time delay compensation for feed-forward channels. In contrast, in the works [3] and [4], NCS is modeled as an asynchronous dynamical system (ADS) with a rate constraint on communication time delays in feedback channels. The compensation schema is implemented by using the approximation of the plant explicit model and the time stamped transmitted data packet.

The goal of this paper is to analyze and propose a compensation method for closed-loop communication time delays (for both feed-forward and feedback channels) using

pole-placement design method. We consider the implementation of several DC-servo control applications on a CAN network and we show the good QoC obtained by the proposed method. We use the tool TrueTime [5] for the simulation and the QoC evaluation. Truetime is a toolbox based on Matlab/Simulink which allows simulating real-time network-based control systems.

The rest of the paper is organized as follows: the section 2 presents the context of the study; section 3 the proposal for computation and compensation of time delays; section 4 presents the computation of the parameters of controller; section 5 represents the implementation of process control applications on a CAN network and the QoC evaluation; the conclusions are given in section 6 of the paper.

2. Context of the Study

2.1. Process Control Application

The model of the considered process control application (using the Laplace transform) is given on Figure 1.

The process to control $G(s)=1000/s(2s+1)$; the PD controller $PD=K(1+sT_d)$; the input reference $R(s) = 1/s$

(unity position step) and the output $Y(s)$. The phase margin of $G(s)$ is 1.3° at the crossover frequency $\omega_c = 22.4$ rad/s (*i.e.* the frequency such that $|G(j\omega_c)| = 1$) [6].

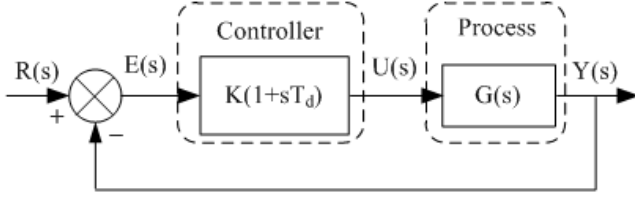


Figure 1. Continuous control system.

This phase margin is too small (the phase margin should be from 45° to 60° [7]). Here we consider that, initially, the PD controller must compensate a positive phase φ_c in order to have a system phase margin of 50° .

To do that, it must be:

$$\begin{cases} \tan(\varphi_c) = T_d \omega_c \\ 20|\log(K(1 + j\omega_c T_d))| = 0 \end{cases} \quad (1)$$

We have $K = 0.6598$ and $T_d = 0.0509$ s.

The closed-loop transfer function is:

$$F(s) = \frac{500K(1 + T_d s)}{s^2 + (0.5 + 500KT_d)s + 500K} \quad (2)$$

$$F(s) = \frac{\omega_n^2(1 + T_d s)}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (3)$$

Where ω_n is the natural pulsation (rad/s) and ζ is the damping coefficient.

We have $\omega_n^2 = 500K$ and $2\zeta\omega_n = 0.5 + 500KT_d$ which give $\omega_n = 18.16$ rad/s and $\zeta = 0.48$, the two poles $p_{1,2}$ are $p_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$, *i.e.* $p_{1,2} = -8.72 \pm j15.93$. The settling time at 2% is $t_s = 382$ ms, the Overshoot is $O = 38.17\%$. The time response $y(t)$ is represented in Figure 2.

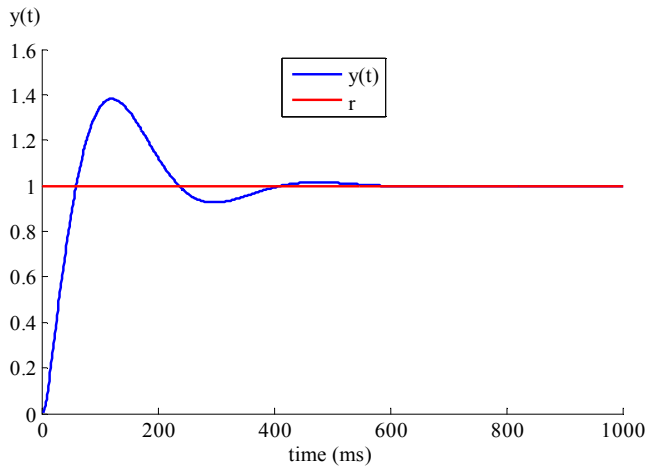


Figure 2. Time response $y(t)$.

2.2. Implementation of a Process Control Application on a CAN Network

The general model of the implementation of a process control application through a network is shown on Figure 3.

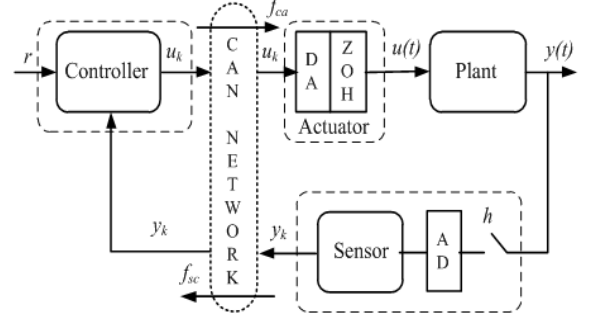


Figure 3. Implementation of a process control application on a CAN network.

The sensor receives the sampled output y_k provided by the Analog Digital (AD) converter and sends it to the controller via the communication network.

The controller receives y_k from the sensor, then calculates the control signal u_k and sends u_k to the actuator via the communication network. The actuator receives u_k , converts u_k into analog signal $u(t)$ using the Digital Analog (DA) converter and then directly applies $u(t)$ to plant. The Zero Order Hold (ZOH) keeps the value $u(t)$ until the new sampling time.

We have two flows of frames: the Sensor-Controller flow which is concerned the frames going from the sensor to the controller (noted f_{sc} flow and call “ f_{sc} frame” a frame of this flow); the Controller-Actuator flow which is concerned the frames going from the controller to the actuator (noted f_{ca} flow and call “ f_{ca} frame” a frame of this flow).

The sensor task is Time-Triggered (*i.e.* controlled by the sampler clock). The controller and actuator tasks are Event-Triggered (the events are the receptions of the f_{sc} frames and f_{ca} frames, respectively).

We call h the sampling period [8] which is defined by the following formula:

$$0.1 \leq \omega_n h \leq 0.6 \quad (4)$$

Where ω_n is the natural pulsation (rad/s). We choose the sampling period $h = 10$ ms.

2.3. Communication Time Delay

Communication time delays in each sampling period consist of 2 components:

- The time delay is due to transferring f_{sc} frames (noted τ_{sc}) which is the duration between the sampling instant ($t_k, k = 0, 1, 2, \dots$) and the reception instant of this frame by the controller. The time delay τ_{sc} includes the waiting time for medium access (in the case where the medium is busy when the frame is ready to send) and the transmission time of a f_{sc} frame (noted D_{sc}).

- The time delay due to transferring f_{ca} frames (noted τ_{ca}) elapsed from the ready-to-send instant of the f_{ca} frame till the reception instant of this frame by the actuator. The time delay τ_{ca} includes the waiting time for medium access and the transmission time of a f_{ca} frame (noted D_{ca}).

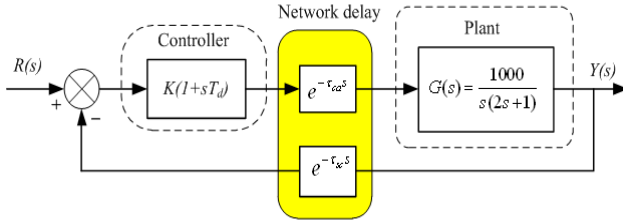


Figure 4. Control system with communication time delays.

Note that the durations D_{sc} and D_{ca} can be easily calculated by knowing the frame length and transmission speed.

Therefore, the communication time delay of a closed-loop control system is:

$$\tau = \tau_{sc} + \tau_{ca} \quad (5)$$

In this paper, we consider the following hypotheses:

- Communication time delay $\tau < h$.
- Computational time in the controller, sensors is neglected.
- There is no data loss.
- Clocks of the sensor and the controller are synchronized, *i.e.* the controller recognizes the sampling instant t_k .

The model of the implementation of a process control application on a network can be represented by the continuous model given on Figure 4 where a time delay τ is represented as $e^{-\tau s}$.

The transfer function $F(s)$ is now:

$$F(s) = \frac{K(1+sT_d)e^{-\tau_{ca}s}G(s)}{1+K(1+sT_d)e^{-\tau s}G(s)} \quad (6)$$

The exponential function can be replaced with the Padé first order approximation *i.e.*

$$e^{-\tau s} \approx \frac{-s + \frac{2}{\tau}}{s + \frac{2}{\tau}} = \frac{-s + a}{s + a}, \quad e^{-\tau_{ca}s} \approx \frac{-s + \frac{2}{\tau_{ca}}}{s + \frac{2}{\tau_{ca}}} = \frac{-s + b}{s + b}$$

Where $a = 2/\tau, b = 2/\tau_{ca}$

We get finally the transfer function as follows:

$$F(s) = \frac{500Ka(1+sT_d)(1+s/a)(1-s/b)}{[f_3(s)](1+s/b)} \quad (7)$$

Where: $f_3(s) = s^3 + (0.5 + a - 500KT_d)s^2 + (500KT_d + 0.5a$

$- 500K)s + 500Ka$.

From Equation (7), we have 4 poles (3 poles p_1, p_2, p_3 of the polynomial $f_3(s)$, $p_4 = -2/\tau_{ca}$) and 3 zeros ($z_1 = -1/T_d$, $z_2 = -2/\tau$, $z_3 = 2/\tau_{ca}$).

2.4. Medium Access Priority

As the MAC protocol of the bus CAN is the priority type. That means that each node in the network has a unique medium access priority. When several nodes have frames to transmit at the same time, they will do the competition by comparing their priorities and the node which has the highest priority is the only winner which can send its frame.

Here we consider the conclusion shown in [9]: The priority of the f_{ca} flow should be higher than the priority of the f_{sc} flow in order to get the best results.

Considering n process control applications (P_1, P_2, \dots, P_n), each process has one f_{sc} flow and one f_{ca} flow. We call $Prio_f_{ca_i}$ and $Prio_f_{sc_i}$ the priorities of the f_{sc} and f_{ca} flows of the process P_i ($i = 1, 2, \dots, n$), respectively.

The priorities of $2n$ data flows are arranged in the following order:

$$Prio_f_{ca_1} > Prio_f_{ca_2} > \dots > Prio_f_{ca_n} > Prio_f_{sc_1} > Prio_f_{sc_2} > \dots > Prio_f_{sc_n}$$

i.e. the controller has a higher priority than the sensor and during a sampling period, the order of medium access is arranged as P_1, P_2, \dots, P_n .

3. Proposal for Computation and Compensation of Time Delays

3.1. Ideas

The QoS, which is considered here, is the closed-loop communication time delay τ during each sampling period. With the knowledge of τ , the controller can compensate this time delay by modifying the parameters K and T_d in such a way to maintain the same type of transient behavior for the process control application as before the implementation on the network (*i.e.* characterized by the two poles of the transfer function of Equation (3)).

As the transfer function of the system implemented on the network (Equation (7)) has 4 poles, the modification of K and T_d , according to the pole placement design method, must keep the main role for the 2 poles of Equation (3) (*i.e.* poles $p_{1,2} = R \pm jI$, with $R = -8.72$ and $I = 15.93$, which are called the dominant poles) and integrate the conditions which give an insignificant role to the poles p_3, p_4 (called insignificant poles).

In order to be insignificant, the poles p_3 and p_4 must have their real part very smaller than that of the dominant poles.

Note that, if the pole placement design method maintains the transient behavior type, however it cannot maintain the overshoot value.

Actually, the numerator of Equation (7) has now three zeros. Thus we have to evaluate the influence of these zeros

on the overshoot.

The pole placement design method is characterized by two important points: The obtaining of the closed-loop communication time delay by the controller and the computations in the controller for the maintenance of the dominant poles.

3.2. Computation of Closed-Loop Communication Time Delays

The computation of closed-loop communication time delays is done by the controller in each sampling period. Concerning the time delay τ_{sc} , as the controller has the knowledge of sampling instants t_k ($k = 0, 1, \dots$), it can easily deduce the value of τ_{sc} by the time difference between the reception instant of f_{sc} frames and t_k . Concerning the time delay τ_{ca} , it cannot be calculated because the f_{ca} frames have not been transmitted yet. However, due to the hypotheses in Section 2.4 (*i.e.* the priority of the controller is higher than this of the sensor; there is no competition between the controllers; there is no data lost), the controller can immediately send its frame. *Therefore, τ_{ca} is equal to the duration of a frame transmission (D_{ca}).* The closed-loop time delay will be computed by the controller as followed:

$$\tau = \tau_{sc} + D_{ca} \quad (8)$$

3.3. Time Delay Compensation Steps

The compensation for time delays done by the controller in each sampling period has the following steps:

- Step 1: Identifying expected poles including the dominant poles and the other poles which are selected equally to the real part of the dominant pole divided α , with $\alpha = 2 \div 10$ [8].
- Step 2: Computing closed-loop communication time delay.
- Step 3: Computing the controller parameters according to the time delay value in order to maintain the position or the value of the expected poles.
- Step 4: Computing the control signal based on the new control parameters calculated in the previous step.

4. Computation of Controller Parameters

4.1. The Computations in the Controller for the Maintenance of the Dominant Poles

Consider the polynomial $f_3(s)$ in the denominator of Equation (7), this polynomial concerns the poles p_1, p_2, p_3 and can be written as $(s - p_1)(s - p_2)(s - p_3)$, which can be re-written by considering the values of $p_{1,2} = R \pm jI$:

$$\begin{aligned} (s - p_1)(s - p_2)(s - p_3) &= \\ &= s^3 - (2R + p_3)s^2 + (2Rp_3 + R^2 + I^2)s - (R^2 + I^2)p_3 \end{aligned} \quad (9)$$

By identifying $f_3(s)$ with Equation (9), we get the relations

which allow determining the values of p_3, K and T_d :

$$\begin{cases} 0.5 + a - 500KT_d = -(2R + p_3) \\ 500KT_d a + 0.5a - 500K = 2Rp_3 + R^2 + I^2 \\ 500Ka = -(R^2 + I^2)p_3 \end{cases} \Rightarrow \begin{cases} p_3 = -\frac{a^3 + (1 + 2R)a^2 - (R^2 + I^2)a}{a^2 - 2Ra + R^2 + I^2} \\ K = -\frac{(R^2 + I^2)p_3}{500a} \\ T_d = \frac{0.5 + a + p_3 + 2R}{500K} \end{cases} \quad (10)$$

We replace the value of K in Equation (7) by this one found in Equation (11) and taking into account for the relation $R^2 + I^2 = \omega_n^2$, we have now the transfer function $F(s)$:

$$F(s) = \frac{\omega_n^2(1 + sT_d)(1 + s/a)(1 - s/b)}{(s^2 + 2\zeta\omega_n s + \omega_n^2)(1 - s/p_3)(1 + s/b)} \quad (11)$$

Note that p_3 and $p_4 = -2/\tau_{ca}$ are real poles.

4.2. Conditions for the Insignificance of the Poles p_3 and p_4

We take the conditions expressed by [10] *i.e.* the real part is five times smaller than the real part of the dominant poles, then:

- $p_3 \leq 5R$ which need $\tau_{sc} + \tau_{ca} < 26.08$ ms.
- $p_4 \leq 5R$ need $\tau_{ca} < 45.87$ ms.

These conditions are always satisfied as the sampling period h is 10 ms and $\tau_{sc} + \tau_{ca}$ is always smaller than h .

4.3. Effect of the Zeros

“When a zero gets closer to the origin, the overshoot increases” [11], [12].

For a negative zero (call z_0 the absolute value of this zero), in [11], the zero has a little effect and can be neglected if $|z_0|/\zeta\omega_n \geq 5 \Rightarrow |z_0| \geq 5\zeta\omega_n = 43.58$.

For a positive zero (call z_0 the value of this zero), in [12], the overshoot is evaluated as follows:

$$O = e^{-\frac{\zeta(\Phi + \pi)}{\sqrt{1 - \zeta^2}}} \sqrt{1 + \frac{2\zeta\omega_n}{z_0} + \left(\frac{\omega_n}{z_0}\right)^2} \quad (12)$$

$$\text{Where } \Phi = \tan^{-1} \frac{\sqrt{1 - \zeta^2}}{z_0 + \zeta\omega_n}$$

We see that if $z_0/2\zeta\omega_n \gg 1$ and $(z_0/\omega_n)^2 \gg 1$, the

overshoot can be re-written as $O = e^{-\frac{\zeta\pi}{\sqrt{1 - \zeta^2}}}$ *i.e.* we have the overshoot of the second order system without zero. We

can thus consider that if $z_0/2\zeta\omega_n > 5$ and $(z_0/\omega_n)^2 > 5$ (these conditions induce that $z_0 > 87.17$ (with $\omega_n = 18.16$, $\zeta = 0.48$)).

We consider now the three zeros:

- The negative zero $z_2 = -2/\tau$: as $\tau = \tau_{sc} + \tau_{ca}$ is smaller than h , z_2 is smaller than $-2/0.01 = -200$. We see that $|z_2| > 43.58$. Thus the zero z_2 can be neglected.
- The positive zero $z_3 = 2/\tau_{ca}$: because of τ_{ca} is smaller than h , z_3 is bigger than $2/0.01 = 200$, thus $z_3 > 87.17$. Hence the zero z_3 can be neglected.
- The negative zero $z_1 = -1/T_d$: note that this zero is based on a parameter of the controller (T_d). The effect of z_1 depends on the value of T_d ; the higher the value of T_d is, the more closed the zero is to the origin, thus the zero has the stronger effect.

Remark about the transfer function after the compensation of the time delay: Considering the insignificance of the poles p_3 , p_4 and the negligible effect of the zeros z_2 and z_3 , the transfer function in Equation (11) can be written as Equation (13).

$$F(s) = \frac{\omega_n^2(1 + sT_d)}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (13)$$

We see that we have the same form as the transfer function of the system without delays (Equation (3)) but the value of T_d is now changed.

4.4. Validation of the Pole Placement Design Method

Table 1 shows the results obtained by considering different loop delays.

We see that the poles are maintained resulting from the action on K and T_d (decrease of K and increase of T_d). We also see the increase of the overshoot (mainly due to the increase of the zero $z_1 = -1/T_d$; the role of the zeros z_2 , z_3 and of the poles p_3 , p_4 is negligible).

5. Implementation of Process Control Applications on a CAN Network

5.1. Context

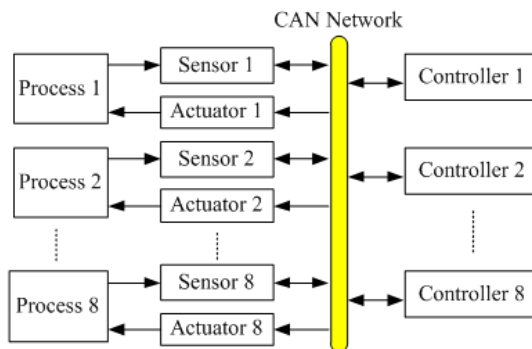


Figure 5. Implementation of 8 process control applications on the CAN network.

We implement 8 identical process control applications (noted P_1, P_2, \dots, P_8) through a CAN network (Figure 5). So we have 24 different nodes connected through a CAN network and 16 data flows ($8f_{sc}$ and $8f_{ca}$ flows) sharing the network. We consider other parameters and conditions as followed:

- Bit rate = 250 Kbit/s.
- Data field length of a frame = 8 bytes. Thus, $D_{sc} = D_{ca} = 150$ bit [13].
- The sensor tasks are synchronous and have the same sampling period.

5.2. Criteria of the QoC Evaluation

The QoC is evaluated through a cost function ITSE (Integral of Time-weighted Square Error) [11]:

$$J = \int_0^T te^2(t) dt \quad (14)$$

with $T > t_s$ in order to cover the transient regime duration.

We consider $T = 1000$ ms and we get the value of J for the control system without network (Section 2.1) is $J_0 = 0.001962$ which is considered as the reference value.

The QoC criteria is represented by the term:

$$\frac{\Delta J}{J_0} \% = \frac{J - J_0}{J_0} \% \quad (15)$$

The smaller the value $\Delta J/J_0$ is, the better the QoC is.

We will also consider the time response for the QoC evaluation.

5.3. Results

We want to show here the interest of the pole placement design method (which is based on an adaptive controller *i.e.* the parameters of the controller are modified according to the closed-loop communication time delays) in comparison with the case where we do not use this method (*i.e.* we have a fixed controller).

The QoC is represented in Table 2 ($\Delta J/J_0$ %), Figure 6 & 7. The comparison between the case delay compensation and the case without delay compensation is presented on Table 2 and on Figure 6 (we use the criteria ITSE) and on Figure 7 (presentation of the output time responses).

Remind that the priority of the process P_i is higher than this of the process P_j . We see in Table 2 for the both cases that the QoC follows the order of the priority, it is reasonable because the higher the priority is, the lower the time delay is and so the better the QoC is.

In Figure 6, we can see the improvement of the QoC for the proposal method (delay compensation) compared with the case of non-compensation and lack compensation expressed through smaller values of $\Delta J/J_0$ %.

From the results of the proposal method (Figure 7-d): it improves the results which are less dispersed (the influence of the priorities is reduced); however we cannot have identical performances for all the process control

applications because of the effect of the zero $-1/T_d$ (as the applications which have less priority have higher delays, the compensation requires higher values of T_d which induces that the zero $-1/T_d$ moves closer to the origin).

We note also the effect of the zero $-1/T_d$ by the increase of O (Figure 7-d) compared with the O (Figure 2) of the control system without network in Section 2.1.

Table 1. Validation of the pole-placement design method.

a) $\tau_{ca} = 1$ ms.

τ_{sc} (ms)	τ (ms)	K	T_d (s)	$p_{1,2}$	p_3	p_4	O (%)	z_1	z_2	z_3
1	2	0.6372	0.0533	$-8.72 \pm j15.93$	-966	-2000	32.15	-18.7	-1000	2000
2	3	0.6260	0.0544	$-8.72 \pm j15.93$	-633	-2000	32.74	-18.4	-667	2000
3	4	0.6148	0.0554	$-8.72 \pm j15.93$	-466	-2000	33.35	-18.1	-500	2000
4	5	0.6036	0.0565	$-8.72 \pm j15.93$	-366	-2000	33.99	-17.7	-400	2000

b) $\tau_{sc} = 1$ ms.

τ_{ca} (ms)	τ (ms)	K	T_d (s)	$p_{1,2}$	p_3	p_4	O (%)	z_1	z_2	z_3
1	2	0.6372	0.0533	$-8.72 \pm j15.93$	-966	-2000	32.15	-18.7	-1000	2000
2	3	0.6260	0.0544	$-8.72 \pm j15.93$	-633	-1000	33.73	-18.4	-667	1000
3	4	0.6148	0.0554	$-8.72 \pm j15.93$	-466	-667	35.46	-18.1	-500	667
4	5	0.6036	0.0565	$-8.72 \pm j15.93$	-366	-500	37.38	-17.7	-400	500

Table 2. QoC ($\Delta J/J_0$ %).

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
Non-Compensation	7.37	15.77	25.40	36.52	49.44	64.55	82.36	103.60
Compensation for forward channel	4.56	12.59	21.80	32.41	44.71	59.07	75.97	96.03
Compensation for feedback channel	4.56	11.06	18.21	26.08	34.76	44.32	54.87	66.51
Compensation for both forward and feedback channels	3.92	10.35	17.42	25.19	33.75	43.17	53.54	64.98

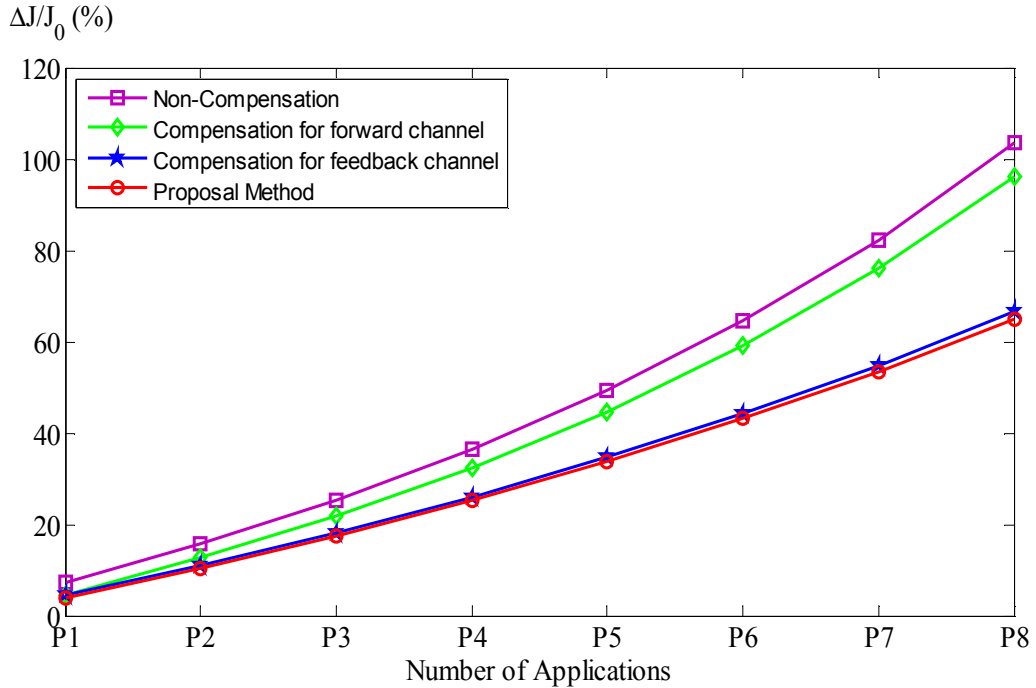


Figure 6. Graphic representation of the QoC ($\Delta J/J_0$ %) of the 8 applications.

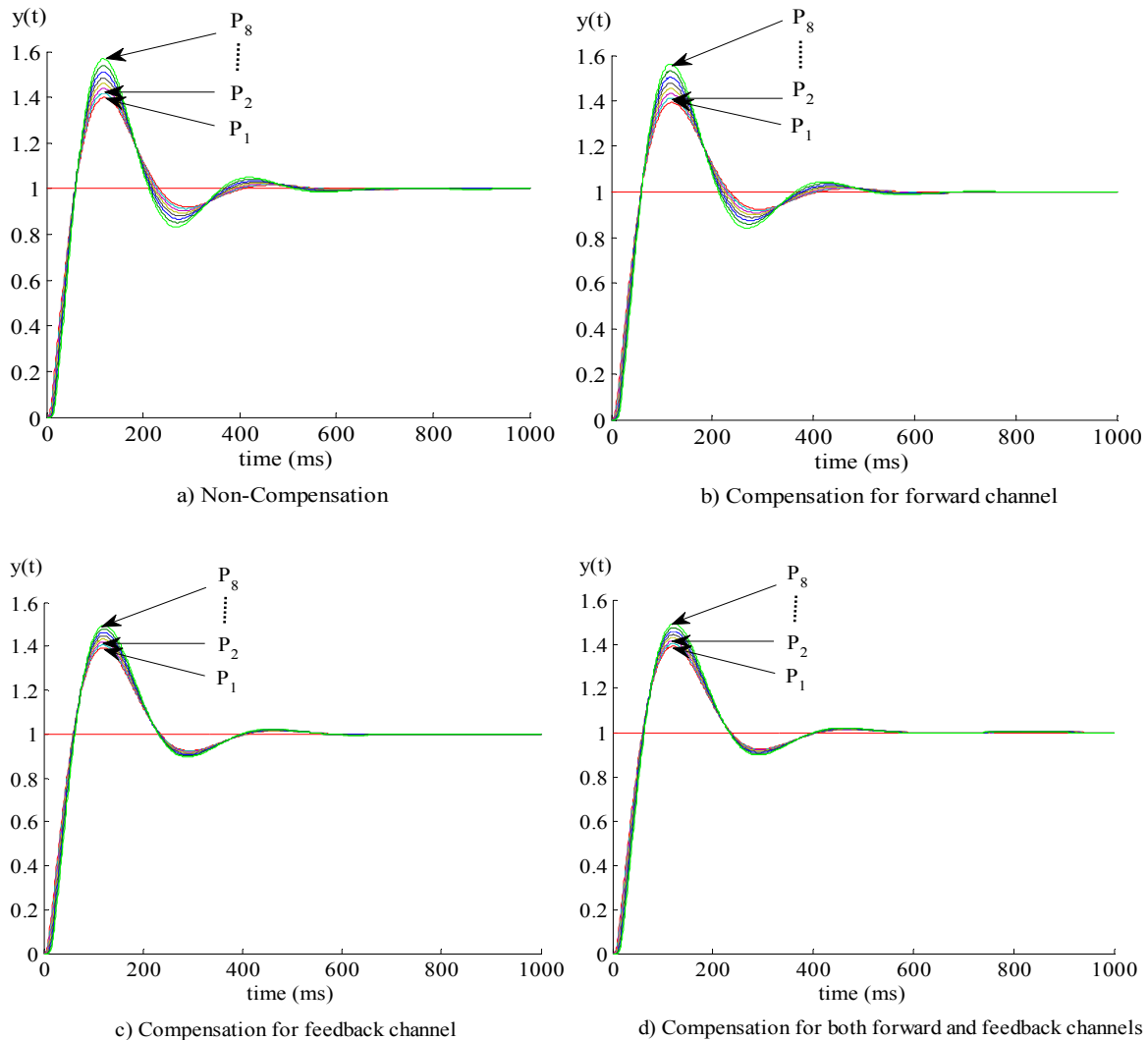


Figure 7. Time responses of the 8 applications.

6. Conclusion

In this paper, we have proposed a way to calculate the closed loop communication time delay and have considered the compensation for this time delay by using the pole placement design method for CAN-based NCSs. We have implemented 8 process control applications (DC – Servo) on a CAN network and we have shown the interest of our proposition through the obtained QoC improvement.

The future work should be the consideration of a co-design between controller designs (time delay compensation) and the message scheduling using hybrid priorities [9] in order to improve simultaneously QoS and QoC, and to obtain a more efficient NCS.

References

- [1] Wenshan Hu, Guoping Liu, and David Rees, “Event-Driven Networked Predictive Control”, IEEE Transactions on Industrial Electronics, Vol. 54, No. 3, pp. 1603-1613, 2007.
- [2] Yafeng Guo and Shaoyuan Li, “A new networked predictive control approach for systems with random network delay in the forward channel”, International Journal of Systems Science, Vol. 41, No. 5, pp. 511–520, 2010.
- [3] Bienvenu A. Soglo and Xianhui Yang, “Compensation for Network-induced Delays and Packet Dropping in Control Systems”, European Journal of Control, Vol. 12, pp. 296 – 306, 2006.
- [4] Bienvenu A. Soglo, Xianhui Yang, “Networked Control System Compensator Design and Stability Analysis”, International Conference on Control and Automation, June, Hungary, Vol. 2, pp. 715-719, 2005.
- [5] Martin Ohlin, Dan Henriksson and Anton Cervin, “TrueTime 1.5 - Reference Manual”, Lund Institute of Technology, Sweden, 2007.
- [6] Benjamin C. Kuo, Farid Golnaraghi, “Automatic Control Systems”, 8th Edition, John Wiley & Sons, INC, page 236-245, 2003.
- [7] John J. D’Azzo, C. H. Houpis, “Linear Control System Analysis and Design: Conventional and Modern”, 4th Edition, McGraw-Hill, New York, 1995.

- [8] Karl J. Åström and B. Wittenmark, "Computer controlled systems: theory and design", 3th Edition, Prentice Hall, 1997.
- [9] G. Juanole, G. Mouney, and C. Calmettes, "On different priority schemes for the message scheduling in Networked Control Systems", 16th Mediterranean Conference on Control and Automation, June, France, pp. 1106-1111, 2008.
- [10] Benjamin C. Kuo, Farid Golnaraghi, "Automatic Control Systems," 8th Edition, John Wiley & Sons, INC, page 236-245, 2003.
- [11] Richard. Dorf, Robert H. Bishop, "Modern control systems," 10th Edition, Pearson Prentice Hall, 2005.
- [12] Byung-Moon Kwon, Hee-Seob Ryu, and Oh-Kyu Kwo, "Transient response analysis and compensation of the second order system with one RHP real zero", Transactions on Control, Automation and Systems Engineering, Vol. 2, No. 4, pp. 262-267, 2000.
- [13] Salem Hasnaoui, Oussema Kallel, Ridha Kbaier, Samir Ben Ahmed, "An implementation of a proposed modification of CAN protocol on CAN fieldbus controller component for supporting a dynamic priority policy", 38th Annual Meeting of the Industrial Applications, Oct, Vol. 1, pp. 23-31, 2003.