



**HAL**  
open science

# Networked control systems quality of service aware quality of control adaptation

Xuan Hung Nguyen

► **To cite this version:**

Xuan Hung Nguyen. Networked control systems quality of service aware quality of control adaptation. 16th IEEE International Conference on Emerging Technologies & Factory Automation (ETFAs), Sep 2011, TOULOUSE, France. pp.4, 10.1109/ETFAs.2011.6059154 . hal-00656657

**HAL Id: hal-00656657**

**<https://hal.science/hal-00656657>**

Submitted on 4 Jan 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Networked Control Systems

## Quality of Service aware Quality of Control adaptation

Xuan Hung Nguyen<sup>1,2</sup>

<sup>1</sup>CNRS ; LAAS ; 7 avenue du Colonel Roche,  
F-31077 Toulouse, FRANCE

<sup>2</sup>Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS :  
F-31077 Toulouse, FRANCE

Email: xhnguyen@laas.fr

### Abstract

We consider the implementation of simple process control applications through a CAN network. The goal of this paper is to study modifications on-line of controller parameters according to the global time delay in the control loop. Two methods that we use are the phase margin method and the dominant pole one. We evaluate the performances of the two methods in terms of the time response.

## 1. Introduction

Networked Control Systems (NCSs) are a very important research area because of their multidisciplinary aspect: Automatic control, computer science (task scheduling), communication network (message scheduling, message transfer). Here we consider two aspects: automatic control and message scheduling. A realistic design, in this framework, is to have a co-design approach between the controller law and the message scheduling, *i.e.* with a bi-directional relation between the Quality of Control (QoC) provided by the controller and the Quality of Service (QoS) provided by the scheduling of messages (QoC  $\rightleftharpoons$  QoS). The idea of the link (QoC  $\rightarrow$  QoS) is to re-allocate the implementation system resources on-line to guarantee an expected QoS level required by the control application. In [6], we have in particular shown the interest of this link (QoC  $\rightarrow$  QoS) by means of the concept of dynamic priority schemes for messages in the MAC layer of a CAN network [5].

In this paper, we want to show the interest of the link (QoS  $\rightarrow$  QoC) *i.e.* where the controller is aware on-line of the time delay in the control loop and adapts, in consequence, its parameters. We consider a simple continuous linear process control application (one variable) implemented on a CAN network with a Proportional-Derivative (PD) controller.

Two methods for controlling the consequences of

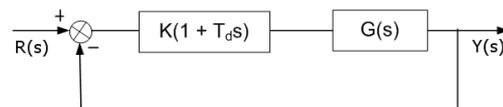


Figure 1. Continuous control system.

the time delay in the control loop will be the phase margin and the dominant poles methods.

This paper includes the following sections: the section 2 presents the study context; the section 3 concerns the proposal for the implementation on-line of the link (QoS  $\rightarrow$  QoC); the section 4 concerns an application of this proposal; the conclusion is represented in the last section 5.

## 2. Study context

### 2.1. Considered continuous control system

The model of the considered process control application is given on the figure 1: the process to control  $G(s) = \frac{1000}{s(s+1)}$ , the PD controller  $K(1 + T_d s)$ , the step input  $R(s)$  and the output  $Y(s)$ .

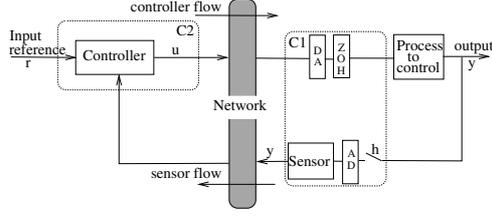
The phase margin ( $Pm$ ) of  $G(s)$  is  $Pm_0 = 1.8^\circ$  at the crossover frequency  $\omega_c = 31.6 \text{rad/s}$  (*i.e.*  $\omega_c$  such that  $|G(j\omega_c)| = 1$ ). This phase margin ( $Pm_0$ ) is not enough and thus initially we want to use a PD controller to compensate the phase and to have a phase margin  $Pm$  of  $45^\circ$  still at  $\omega_c$ .

Then the controller must bring, at  $\omega_c$ , a positive phase  $\varphi_c = 45^\circ - Pm_0$ . To do this, it must be:  $tg(\varphi_c) = T_d \omega_c$  and  $20|\log(K(1 + j\omega_c T_d))| = 0$ .

Thus we have the formula (1) which allows to get the values  $T_d = 0.0297 \text{s}$  and  $K = 0.7291$ .

$$\begin{cases} T_d = \frac{tg(\varphi_c)}{\omega_c} \\ K = \frac{1}{\sqrt{1 + (\omega_c T_d)^2}} \end{cases} \quad (1)$$

In such condition, the main characteristics of the continuous control system are:  $Pm = 45^\circ$ ; poles  $p_{1,2} = -11 \pm j24.5$ ; rise time  $t_r = 37.5 \text{ms}$ , steady state error  $e_{ss} = \lim_{s \rightarrow 0} s \frac{1}{1 + K(1 + T_d s)G(s)} R(s) = 0$ .



**Figure 2. Implementation of a process control application through a network.**

## 2.2. Implementation of a process control application on a network

The general model of the implementation of a process control application through a network is shown on the figure 2. DA and AD are the Digital-to-Analog and Analog-to-Digital converters. ZOH (Zero Order Hold) is to hold each sample value for one sample interval. The computer C1 includes the sensor and actuator tasks which are associated to the process to control. The computer C2 includes the controller task. The sensor task is time-triggered with a sampling period  $h$ . The controller task and the actuator task are event-triggered. We have two message flows: the sensor flow  $f_{sc}$  (*i.e.* messages sent from the sensor to the controller) and the controller flow  $f_{ca}$  (*i.e.* messages sent from the controller to the actuator).

The running of the process control application is characterized by several time delays:

- The computational and functional delays in the execution of the computers C1, C2.
- The communication network delays in the communications of the sensor flow (denoted  $\tau_{sc}$ ) and the controller flow (denoted  $\tau_{ca}$ ).
- The time delay induced by the ZOH (denoted  $\tau_{ZOH}$ ) which can be seen as a communication time delay of  $\frac{h}{2}$  [4].

In this work, we only consider the communication delays  $\tau_{sc}$ ,  $\tau_{ca}$  and  $\tau_{ZOH}$ . The global time delay  $\tau$  is evaluated as  $\tau_{sc} + \tau_{ca} + \tau_{ZOH}$ .

The considered network is the CAN network which has a bit rate of 125Kbit/s in the physical layer. The data frames of the  $f_{sc}$  and  $f_{ca}$  flows have a length of 80 bits, *i.e.* durations  $D_{sc}$  and  $D_{ca}$  of 0.64ms.

The choice of  $h$  is based on the formula  $\frac{t_r}{10} \leq h \leq \frac{t_r}{4}$  [1]. We take the round number of the upper bound  $h = 10$ ms.

## 2.3. Simulation tools

In this work, we use the simulator TrueTime [7], a toolbox based on Matlab/Simulink which allows to simulate real-time distributed control systems.

## 3. Proposal for the implementation on-line of the link (QoS $\rightarrow$ QoC)

### 3.1. Main ideas

As the sensor task is time-triggered (at every sampling instant  $kh$  with  $k = (0, 1, 2, \dots)$ ) and as this task is included in the same computer C1 as the actuator task, the computer C1 can evaluate the global time delay ( $\tau$ ) in the control loop of each sampling period. First, C1 evaluates the value ( $\tau_{sc} + \tau_{ca}$ ) by means of the time elapsed from the beginning of the sampling instant till the reception instant of the  $f_{ca}$  message by the actuator task. Then C1 deduces the value of the global time delay ( $\tau_k = \tau_{sc} + \tau_{ca} + \tau_{ZOH}$ ) in the control loop.

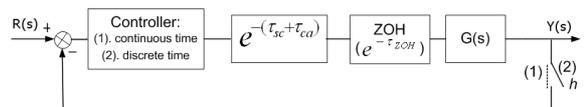
Concerning the  $f_{sc}$  message sent by the sensor task at the sampling instant  $kh$  (*i.e.* at the beginning of the  $(k+1)^{th}$  period), this message includes the sampled output  $y_k$  and the global time delay  $\tau_k$  evaluated during the previous period (the  $k^{th}$  period). Then when the controller receives this message, it will have the knowledge of these two informations ( $y_k, \tau_k$ ):

- Based on  $\tau_k$ , the controller computes on-line the new parameters  $K$  and  $T_d$  of the controller.
- After that, based on  $y_k$  and new parameters  $K, T_d$ , the controller computes the control signal  $u_{k+1}$  and sends immediately this  $u_{k+1}$  to the actuator.

Note that this action of the controller does not depend on the instant of the  $f_{sc}$  message reception in the present period. Hence the model of the closed loop system may be considered with a unity feedback chain and all the time delays in the direct chain as the model represented on the figure 3.

Concerning the computations of  $K, T_d$  and  $u_k$  for the implementation on-line of the link (QoS  $\rightarrow$  QoC), we want to give two points of clarification:

- The computation of  $K$  and  $T_d$  of the controller is done by considering a continuous controller represented by the mark (1) on Fig.3.
- The computation of  $u_k$  is done by considering the discrete time controller (got after the discretization of the continuous controller) represented by the mark (2) on Fig.3.



**Figure 3. System with time delays.**

**Remark:** at time 0 (*i.e.* the 1<sup>st</sup> period), the sensor has no information about the  $\tau_{sc}$  and  $\tau_{ca}$ , hence it takes the global time delay equal to  $\tau_{ZOH} = \frac{h}{2}$ .

### 3.2. Computation of the parameters $K$ and $T_d$

We present now the two methods which have been used: Phase margin and Dominant pole.

#### A/Phase margin method

The purpose of the phase margin method is to compensate the time delay of each sampling period to maintain an expected phase margin.

A time delay  $\tau$  induces a reduction of the phase margin of  $\Delta\varphi = \tau\omega_c$  [4]. Hence in order to have  $Pm = 45^\circ$ , the controller must provide a positive phase  $\varphi_c = 45^\circ - Pm_0 + \Delta\varphi$ . The new values of  $T_d$  and  $K$  are obtained by using the formula (1) in Section 2.1 with the new value of  $\varphi_c$ .

Note that the maximal positive phase that the PD controller can bring is  $90^\circ$ , the maximal phase  $\varphi_{cmax}$  that the controller can bring is  $90^\circ - 45^\circ - Pm_0 = 43.2^\circ$  corresponding to  $\tau = \frac{\varphi_{cmax}}{\omega_c} = 23.8ms$ . This means that we can only compensate to have  $Pm = 45^\circ$  if the global time delay  $\tau \leq 23.8ms$ .

#### B/Dominant pole method

The closed loop transfer function of the control system is  $F(s) = \frac{K(1+T_d s)e^{-\tau s}G(s)}{1+K(1+T_d s)e^{-\tau s}G(s)}$ . By using an approximation Pade of the 1<sup>st</sup> order  $e^{-\tau s} \approx \frac{-s+2/\tau}{s+2/\tau}$  and call  $a = \frac{2}{\tau}$ , we have:

$$F(s) = \frac{1000K(1+T_d s)(-s+a)}{b_0 s^3 + b_1 s^2 + b_2 s + b_3} \quad (2)$$

where  $b_0 = 1$ ;  $b_1 = 1 + a - 1000KT_d$ ;  $b_2 = 1000Ka + a - 1000K$  and  $b_3 = 1000Ka$ .

The denominator is a third order function with polynomials, thus we will have three poles. Note that there are two zeros ( $-\frac{1}{T_d}, \frac{2}{\tau}$ ) that get closer to the origin when the time delay increases.

The objective of the method is to make to appear in the denominator the two poles of the continuous process control application  $p_{1,2} = R \pm jI$  (with  $R = -11$  and  $I = 24.5$ ) found in the subsection 2.1 and to see the condition which gives a negligible role for the third pole  $p_3$ . Poles  $p_1$  and  $p_2$  are called the dominant poles.

The denominator can be written:

$$(s - p_1)(s - p_2)(s - p_3) = s^3 - (2R + p_3)s^2 + (2Rp_3 + R^2 + I^2)s - (R^2 + I^2)p_3$$

By identifying with the denominator of the equation (2), we obtain the values of  $p_3$ ,  $K$  and  $T_d$ :

$$\begin{cases} p_3 = -\frac{a^3 + (2+2R)a^2 - (R^2 + I^2)a}{a^2 - 2Ra + R^2 + I^2} \\ K = -\frac{(R^2 + I^2)p_3}{1000a} \\ T_d = \frac{1000a}{1000K + a + p_3 + 2R} \end{cases} \quad (3)$$

The condition for an insignificant pole  $p_3$ : the pole is considered "insignificant" if the magnitude of the real part of this pole is at least 5 to 10 times that

of a dominant pole (see [3], page 281). Therefore  $p_3 = -\frac{a^3 + (2+2R)a^2 - (R^2 + I^2)a}{a^2 - 2Ra + R^2 + I^2} \leq 5R \Rightarrow \tau < 40ms$ .

#### Remark:

- The compensable maximum delays of the two methods (23.8ms and 40ms) are conditions to have expected specifications and will change if we want other specifications. They depend on the used method and on the specifications we want, thus they are different.
- The two methods require to change on-line the proportional gain  $K$  (formulas (1), (3)), that will change the static gain of the controller. However the steady state error  $e_{ss} = 0 \forall K$  (Sec.2.1), hence we have no problem about the static gain when the gain  $K$  is modified.

### 3.3. Computation of the control signal $u_k$

The controller computes the control signal  $u_k$  (which includes the Proportional component  $P_k$  and the Derivate component  $D_k$ ) by using the formula (4):

$$\begin{cases} P_k = Ke_k \\ D_k = \frac{T_d}{Nh+T_d}D_{k-1} + \frac{NKT_d}{Nh+T_d}(e_k - e_{k-1}) \\ u_k = P_k + D_k \end{cases} \quad (4)$$

where  $e_k = r_k - y_k$ ,  $N$  is a constant ([1], page 307).

## 4. Application of the proposal

### 4.1 Context

We consider an implementation of four identical process control applications (denoted  $P1, P2, P3, P4$ ) implemented on the same CAN network. The processes have different priorities, so they have different time delays. The goal of the consideration of several processes instead of only one process is to well see the impact of the proposal in Sec.3 on performances of processes which have different time delays.

The sensor tasks of the 4 processes are periodic ( $h = 10$  ms) and synchronous. Call  $P_{sci}$  and  $P_{cai}$  the priorities of the flows  $f_{sc}$  and  $f_{ca}$  of  $Pi$ . The static priorities associated to the message flows of 4 processes are as follows:  $P_{ca1} > P_{ca2} > P_{ca3} > P_{ca4} > P_{sc1} > P_{sc2} > P_{sc3} > P_{sc4}$ . That means that  $Pi$  is considered more important than  $Pj$  with  $i < j$ .

Two cases of controller will be considered in order to well see the importance of the link (QoS  $\rightarrow$  QoC): either the controller is not changed with time delays, we say Fixed Controller (denoted Fix-Controller) or the controller is changed and adapted according to time delays, we say Adaptative Controller (denoted Adapt-Controller); this last case shows the interest of the link (QoS  $\rightarrow$  QoC).

**Remark:** the global time delay of the less important process ( $P4$ ) is  $4 \times (D_{sc} + D_{ca}) + \tau_{ZOH} = 10.12ms$ , it is less than the maximal compensable time delays of the two methods (23.8ms and 40ms). Thus we can have the expected phase margin and poles.

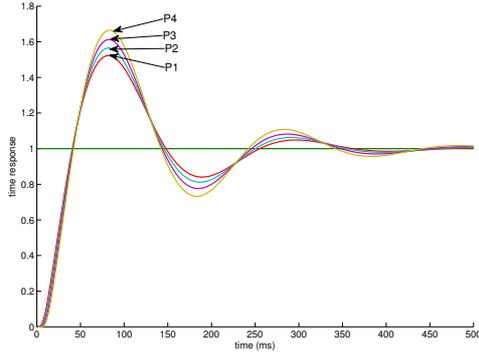


Figure 4. Fix-Controller.

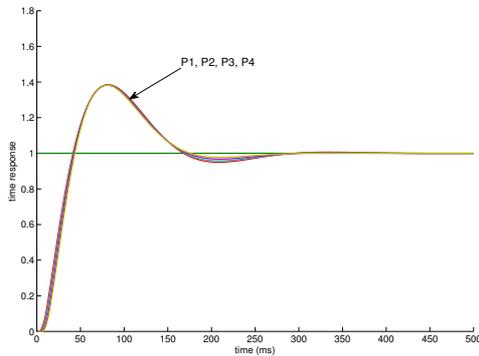


Figure 5. Adapt-Controller: phase margin.

## 4.2 Results

The time responses are represented on Fig.4 for the Fix-Controller and on Figs.5 and 6 for the Adapt-Controller respectively based on the phase margin method and the pole dominant method.

We see on Fig.4 the impact of the differences in the priorities which induce different time delays and then different overshoots. The process that has a higher priority has a better performance.  $P_4$  is the less important, so it has the biggest time delay and that's why it has the highest overshoot.

We see on Fig.5 that the phase margin method compensates very well the time delays and then we have identical time responses for the 4 processes.

On Fig.6 we have not the identical time responses for the 4 processes with the dominant pole method.  $P_i$  is less good than  $P_j$  with  $i > j$ . That results from the effect of the zeros of the closed transfer function (Equ.2). The zeros get closer to the origin with time delays so that they induce a higher overshoot ([2], page 226). But we can see that the result in terms of overshoot is better than this of the Fix-Controller.

The phase margin method gives better results than the dominant pole design since it takes exactly into account for the phase shift of all poles and the zeros of the open loop transfer function.

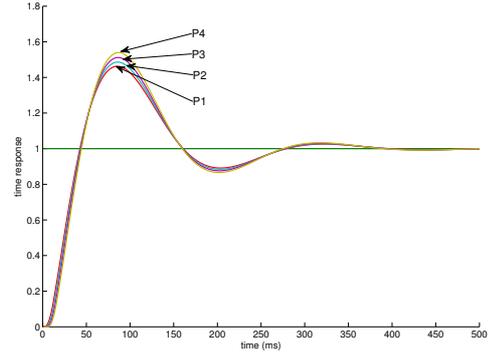


Figure 6. Adapt-Controller: dominant pole.

## 5. Conclusion

This paper has shown the interest that the awareness on-line of the network QoS by the controller in a NCS can bring in terms of performances for process control applications.

Two methods for the modification on-line of the controller parameters, on the basis of the knowledge of the global time delay in the control loop, have been considered. The phase margin method gives excellent results showing its ability to compensate perfectly the time delays. The dominant pole method is less powerful than the phase margin method in this particular application because of the effect of zeros in the closed loop transfer function.

This work presents, in our opinion, an interesting and practical contribution to the link (QoS  $\rightarrow$  QoC). Future works will consider the bidirectional link (QoS  $\rightleftharpoons$  QoC) in order to achieve a more efficient NCS design.

## References

- [1] K. J. Astrom and B. Wittenmark. *Computer-controlled systems – Theory and design*. International Edition – Prentice Hall, 1997.
- [2] R. C.Dorf and R. H.Bishop. *Modern Control Systems*. 7th edition – Addison-Wesley, 1995.
- [3] B. C.Kuo and F. Golnaraghi. *Automatic Control System*. International Edition – Willey, 2003.
- [4] F.-L. Lian, J. Moyne, and D. Tilbury. Network design consideration for distributed control system. *IEEE Transactions On Control Systems Technology*, >10(2):297–307, 2002.
- [5] Bosch GmbH. CAN specification 2.0 (A), 1991.
- [6] X. H. Nguyen, G. Juanole, G. Mouney, and C. Calmettes. Networked control system (ncs) on a network can: on the quality of service (qos) and quality of control (qoc) provided by different message scheduling schemes based on hybrid priorities. In *Proceedings of International Workshop on Factory Communication Systems (WFCS 2010)*, Nancy, France, May 2010.
- [7] M. Ohlin, D. Henriksson, and A. Cervin. *TrueTime 1.5 – Reference Manual*. Lund Institute of Technology, Sweden, January 2007.