



Design of Networked Control Systems (NCSs) on the basis of interplays between Quality of Control and Quality of Service

Xuan Hung Nguyen, Guy Juanole

► To cite this version:

Xuan Hung Nguyen, Guy Juanole. Design of Networked Control Systems (NCSs) on the basis of interplays between Quality of Control and Quality of Service. 7th IEEE International Symposium on Industrial Embedded Systems (SIES'12), Jun 2012, Karlsruhe, Germany. pp.9. hal-00668491v2

HAL Id: hal-00668491

<https://hal.science/hal-00668491v2>

Submitted on 6 Mar 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.

Design of Networked Control Systems (NCSs) on the basis of interplays between Quality of Control and Quality of Service

Xuan Hung Nguyen^{1,2}, Guy Juanole^{1,2}

¹CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse, France

²Univ de Toulouse, UPS, INSA, INP, ISAE, LAAS, F-31400 Toulouse, France

xhnguyen@laas.fr, juanole@laas.fr

Abstract—The goal of this paper is to consider a co-design approach between the controller of a process control application and the frame scheduling of a Local Area Network (LAN). More precisely we present a bi-directional relation between the Quality of Control (QoC) provided by the controller and the Quality of Service (QoS) provided by the LAN (relation noted QoS=QoC). We present, first, the implementation of the relation QoS→QoC on the basis of a compensation method for time delays called dominant pole method, and second, the implementation of the relation QoC→QoS on the basis of the hybrid priorities for the frame scheduling. The final objective is the implementation of the bidirectional relation QoS=QoC which is the combination of the both relations QoS→QoC and QoC→QoS in order to have a more efficient NCSs design.

I. INTRODUCTION

The study and design of Networked Control Systems (NCSs) is a very important research area today because of its multidisciplinary aspect (Automatic Control, Computer Science, Communication Network). The current objective of NCS design today is to consider a co-design in order to have an efficient control system [1], [2].

Several works [3], [4], [5] have considered the aspects Automatic Control and Computer Science (Task scheduling) and have shown the interest of a co-design approach.

Here we consider the aspects Automatic Control and Local Area Network (LAN) by focusing on the frame scheduling implemented in the MAC layer (the messages of the control-command applications are encapsulated in frames). The aim is to show the interest of a co-design approach, of the frame scheduling in the LAN and of the controller of a process control application, on the basis of a bidirectional relation between the Quality of Control (QoC) provided by the controller and the Quality of Service (QoS) provided by the LAN (QoC=QoS), *i.e.* we have both QoC→QoS (QoS is QoC driven *i.e.* Application performance aware dynamic QoS adaptation) and QoS→QoC (QoC is QoS driven *i.e.* Network performance aware dynamic adaptation).

Some works [6], [7], [8] have shown the interest of the relation QoC→QoS. In particular, in our works [7], [8], we have shown the interest of the implementation of this relation on the basis of the concept of hybrid priority for the frame scheduling in the MAC layer.

In [9], we have shown the interest of the relation QoS→QoC where the controller is aware on-line of the time delay in the control loop and adapts, in consequence, its parameters.

In this paper, we want to show the interest to associate the relation QoC→QoS with a relation QoS→QoC. We consider several simple continuous linear process control applications (DC motor) implemented on a CAN network [10] with a Proportional-Derivative (PD) controller.

This paper includes the following sections: the section 2 presents the general context of the study; the section 3 presents, by summarizing results obtained in [9], the implementation of the relation QoS→QoC; the section 4 presents the main ideas [7] and the main results of the implementation of the relation QoC→QoS; the section 5 presents the implementation of the bidirectional relation QoS=QoC and the recapitulation of all the obtained results; the conclusion is represented in the section 6.

II. CONTEXT OF THE STUDY

A. Process control application which is considered

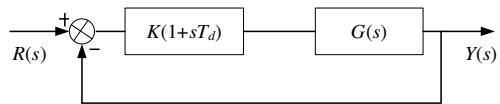


Fig. 1. Continuous control system.

The considered process control application is a continuous linear application, the model of which is given on Fig. 1. The process to control has the transfer function $G(s) = \frac{1000}{s(s+1)}$. We have a Proportional Derivative (PD) controller in order to have a phase margin of 45° which imposes the following values: $K = 0.7291$; $T_d = 0.0297$ s. The input reference is a unity position step $R(s) = 1/s$. The output is $Y(s)$.

The transfer function $F(s)$ of the closed loop system is

$$F(s) = \frac{K(1 + T_d s)G(s)}{1 + K(1 + T_d s)G(s)} \quad (1)$$

$$F(s) = \frac{1000K(1 + T_d s)}{s^2 + (1 + 1000KT_d)s + 1000K} \quad (2)$$

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

where ω_n is the natural pulsation and ζ is the damping coefficient ($\omega_n = 27$ rad/s; $\zeta = 0.4$); the two poles $p_{1,2}$ are $-\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$, i.e. $p_{1,2} = -11 \pm j24.5$.

The performances are: settling time (at 2%) $t_s = 284$ ms; overshoot $O = 33.8\%$. The time response is represented in Fig. 2.

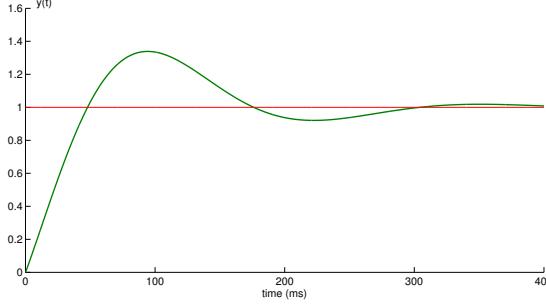


Fig. 2. Time response.

B. On the implementation of a process control application through a network

The implementation through a network which requires the sampling of the output y is represented on Fig. 3.

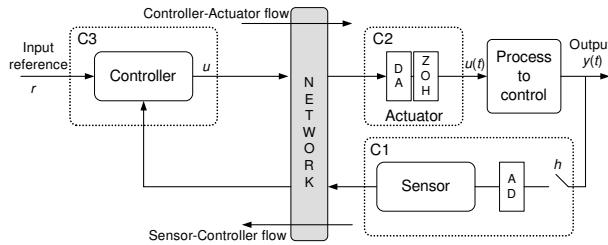


Fig. 3. Implementation of a process control application through a network.

We call h the sampling period which is defined by considering the following formula $\omega_n h \in [0.1; 0.6]$ [11].

We have three computers: the computer C1 where we have the sensor task which receives the numerical information provided by the Analog Digital (AD) Converter after the sampler; the computer C2 where we have the controller task which works out the control signal u from the difference between the reference r and the output y ; the computer C3 where we have the actuator task which provides the control signal u in a numerical way to the Digital Analog (AD) converter; the DA converter then transmits it in an analog way to the Zero Order Hold (ZOH).

We have two flows of frames: the Sensor-Controller flow which concerns 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 concerns 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 task is Event-Triggered (the

events are the receptions of the f_{sc} frames). The actuator task is also Event-Triggered (the events are the receptions of the f_{ca} frames).

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

- The computational and functional time delays in the execution of the tasks in the computers C1, C2, C3.
- The communication network delays in the communications of the f_{sc} frame (denoted τ_{sc}) and the f_{ca} frame (denoted τ_{ca}). During each sampling period, the time delay τ_{sc} is the time difference between the sampling instant and the reception instant of the f_{sc} frame by the controller; the time delay τ_{ca} is the time elapsed from the sending instant of the f_{ca} frame by the controller till the reception instant of this frame by the actuator.
- The time delay induced by the block ZOH can be seen as a communication time delay $\tau_{ZOH} \approx \frac{h}{2}$ [12].

In this work, we only consider the time delays τ_{sc} , τ_{ca} and τ_{ZOH} . The computational and functional delays are neglected. The time delay of the closed loop in each sampling period is $\tau = \tau_{sc} + \tau_{ca} + \tau_{ZOH}$.

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

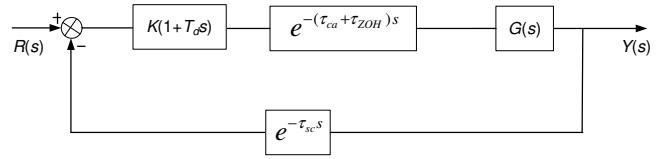


Fig. 4. Control system with time delays.

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

$$F(s) = \frac{K(1+T_d s)e^{-(\tau_{ca}+\tau_{ZOH})s}G(s)}{1+K(1+T_d s)e^{-\tau_{sc}s}G(s)} \quad (4)$$

The exponential function can be replaced with the Padé first order approximation i.e. $e^{-\tau_{sc}} \approx \frac{-s+2/\tau}{s+2/\tau}$ and $e^{-(\tau_{ca}+\tau_{ZOH})s} \approx \frac{-s+2/(\tau_{ca}+\tau_{ZOH})}{s+2/(\tau_{ca}+\tau_{ZOH})}$. By calling $a = 2/\tau$ and $b = 2/(\tau_{ca} + \tau_{ZOH})$, we get finally the transfer function as follows:

$$F(s) = \frac{1000Ka(1+T_d s)(1+s/a)(1-s/b)}{f_3(s)(1+s/b)} \quad (5)$$

where $f_3(s) = s^3 + (1+a-1000KT_d)s^2 + (1000KT_da + a - 1000K)s + 1000Ka$.

From Equation 5, we have 4 poles (3 poles p_1 , p_2 , p_3 of the polynomial $f_3(s)$, $p_4 = -2/(\tau_{ca} + \tau_{ZOH})$) and 3 zeros ($z_1 = -1/T_d$, $z_2 = -2/\tau$, $z_3 = 2/(\tau_{ca} + \tau_{ZOH})$).

C. Global control system which is considered

We now present the system which will be analyzed in this work by means of the simulator TrueTime [13], a toolbox based on Matlab/Simulink which allows to simulate real-time distributed control systems.

1) *General considerations:* We consider the implementation of 4 process control applications (noted P₁, P₂, P₃, P₄) like the one specified in the subsection II-A through a CAN network. The controller tasks, the sensor tasks and the actuator tasks of the 4 process control applications are all in different computers. Then we have 12 computers connected to the network and we have 4 f_{sc} flows and 4 f_{ca} flows which share the network.

The sampling period is $h = 10$ ms for the 4 applications and we suppose also that the sampling instants are the same for the 4 applications (synchronous samplings).

The CAN network has a throughput in the physical layer of 125 Kbits/s and the lengths of the data field of the f_{sc} frames (sensor data) and of the f_{ca} frames (command data) are two bytes (16 bits) which give frames of 75 bits [14]. But as we used the old version of TrueTime (which considers frames with lengths multiples of a byte), we have taken frames of 80 bits (*i.e.* 10 bytes).

Then the durations of the f_{sc} frames (noted D_{sc}) and the f_{ca} frames (noted D_{ca}) are identical and equal to 0.64 ms.

The scheduling of the frames is done in the MAC layer by means of priorities which are represented by the Identifier (ID) field of the frame. The ID field is at the beginning of the frames.

Generally, the priorities are static priorities, *i.e.* each flow has an unique priority (specified *a priori* out of line) and all the frames of this flow have the same priority.

Concerning the priorities, we will consider here either static priorities (*i.e.* each flow has an unique priority specified out of line) or hybrid priorities (as we said in the introduction) *i.e.* with two priority levels. One level represents the flow priority which is a static priority. The other level represents the frame transmission urgency. The urgency can be the same for all the frames of the flow and, in this case, the transmission urgency is also a static priority. The urgency can vary (for example, if the conditions of the application, which uses the flow, change) and, in this case, the transmission urgency is a dynamic priority. This concept has a great interest during the transient behavior of systems [15], [16], [17], [18].

The consideration of hybrid priorities requires to structure the field ID in two levels (Fig. 5) where the level 1 represents the flow priority and the level 2 represents the urgency priority.

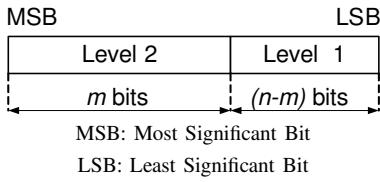


Fig. 5. ID field structure (hybrid priority).

In the context of the competition based on these hybrid priorities, the frame scheduling is executed by comparing first the bits of the level 2 (urgency predominance). If the urgencies are identical, the level 1 (static priorities which have the uniqueness properties) resolves the competition.

2) *Static priorities associated to the f_{sc} and f_{ca} flows:* Call P_{sc} and P_{ca} the priorities of respectively the f_{sc} flow and the f_{ca} flow of the process control application noted P. It has been shown in a previous work [19] that we must have P_{ca} > P_{sc}. In the context of a network shared by several applications, this condition allows the controller to act faster after the reception of a f_{sc} frame. In this study, we consider this result.

Concerning the 4 process control applications P₁, P₂, P₃, P₄, for each process P_i, the priorities of the f_{ca1} and f_{sci} flows are P_{ca1} > P_{sci} and the priorities of the flows of the 4 processes are arranged in the following order:

$$P_{ca1} > P_{ca2} > P_{ca3} > P_{ca4} > P_{sc1} > P_{sc2} > P_{sc3} > P_{sc4} \quad (6)$$

i.e. the process P_i is considered more important than the process P_j with i < j.

3) *Criteria of the performance evaluation:* The Quality of Control (QoC) is evaluated, in particular, with a cost function ITSE (Integral of Time-weighted Square Error) noted J with $J = \int_0^T t(r(t) - y(t))^2 dt$ with T > t_s in order to cover the transient regime duration. We consider T = 500 ms and we get the value of J for the system in Section II-A (noted J₀) is J₀ = 9.4562.10⁻⁴. J₀ will be considered as the reference value for the study done along the paper.

When we consider the process control applications implemented on the network, the performance criteria is represented by the term $\frac{J-J_0}{J_0}\%$ = $\frac{\Delta J}{J_0}\%$. The higher the value $\frac{\Delta J}{J_0}\%$ is, the more degraded the QoC is.

We will also consider the overshoot (O%) and the time response (y(t)) in order to characterize QoC.

4) *Control law:* 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 (7).

$$\begin{cases} P_k = Ke_k \\ D_k = \frac{T_d}{Nh+T_d}D_{k-1} + \frac{NKT_d}{Nh+T_d}(e_k - e_{k-1}) \\ u_k = P_k + D_k \end{cases} \quad (7)$$

where e_k = r_k - y_k, N is a constant ([11], page 307).

III. RELATION QoS→QoC

The implementation of the relation QoS→QoC is on the basis of a compensation method for time delays. Here we consider the dominant pole method.

The strict study of this relation must be based on the use of static priorities (in this way QoS is not linked to QoC).

A. Main ideas

The QoS, which is considered here, is the time delay in the loop ($\tau = \tau_{sc} + \tau_{ca} + \tau_{ZOH}$) 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 5) has 4 poles (p_1, p_2, p_3, p_4), the modification of K and T_d , according to the dominant pole method, must keep the main role for the 2 poles of Equation 3 (*i.e.* poles $p_{1,2} = R \pm jI$ (with $R = -11$ and $I = 24.5$) 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 dominant pole method maintains the transient behavior type, however it cannot maintain the overshoot value.

Actually, the numerator of Equation 5 has now three zeros: $z_1 = -1/T_d$, $z_2 = -2/\tau$, $z_3 = 2/(\tau_{ca} + \tau_{ZOH})$. Thus we have to evaluate the influence of these zeros on the overshoot.

The dominant pole method is characterized by two important points: The obtaining of the loop delay by the controller; The computations in the controller for the maintenance of the dominant poles.

1) The obtaining of the loop delay by the controller: We consider the following hypotheses:

- The computer, where the sensor task is, knows the static priority of the f_{ca} flow.
- The CAN network is not overloaded *i.e.* each process control application can send its two frames (f_{sc} and f_{ca} frames) during each sampling period. Then the value $(\tau_{ca} + \tau_{ZOH})$ is the same for each process control application in each sampling period. Considering the priorities which have been specified in the formula 6, we have $(\tau_{sc} + \tau_{ca})$ of P_i which is smaller than $(\tau_{sc} + \tau_{ca})$ of P_j with $i < j$.

Each sensor task can evaluate the time delay $(\tau_{sc} + \tau_{ca})$ of the concerned process control application (by noting the time difference between the sampling instant $t_k = kh$ (with $k = 0, 1, 2, \dots$) where a f_{sc} frame must be sent and the instant of the reception of the corresponding f_{ca} frame). Then, at the next sampling instant t_{k+1} , the sensor task will put in the data field of the f_{sc} frame this value $(\tau_{sc} + \tau_{ca})$ which will be used by the controller.

From the knowledge of $(\tau_{sc} + \tau_{ca})$, the controller gets, by adding the value $\tau_{ZOH} = h/2$, the value of the loop time delay τ characterizing the period starting at t_k (call τ_k this loop delay).

Remark: the loop time delay, used by the controller during each sampling period, is the loop time delay evaluated at the previous sampling period (but it is not important as the time delay $(\tau_{sc} + \tau_{ca})$ is the same whatever the sampling period may be).

In brief, the controller, in the sampling period starting at t_k , receives the f_{sc} frame including the value $(\tau_{sc} + \tau_{ca})$ and the sampled output value y_k (sampled at t_k):

- With the knowledge of $(\tau_{sc} + \tau_{ca})$, the controller computes the loop time delay τ_k and then computes on-line the new parameters K and T_d using the dominant pole method.

- With the knowledge of these new parameters K and T_d and the value of y_k , the controller computes the control signal u_k and send it in a f_{ca} frame.

Remark about the first sampling period: at the first sampling instant $t_0 = 0$, as the sensor has no information about τ_{sc} and τ_{ca} , the controller will not get such information and then will use only τ_{ZOH} *i.e.* the loop time delay $\tau_0 = \tau_{ZOH}$.

2) The computations in the controller for the maintenance of the dominant poles: Consider the polynomial $f_3(s)$ in the denominator of Equation 5, 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 $p_{1,2} = R \pm jI$:

$$(s - p_1)(s - p_2)(s - p_3) = s^3 - (2R + p_3)s^2 + (2Rp_3 + R^2 + I^2)s - (R^2 + I^2)p_3 \quad (8)$$

By identifying $f_3(s)$ with Equation 8, we get the relations which allow to determine the values of p_3, K and T_d :

$$\left\{ \begin{array}{l} p_3 = -\frac{a^3 + (2 + 2R)a^2 - (R^2 + I^2)a}{a^2 - 2Ra + R^2 + I^2} \\ K = -\frac{(R^2 + I^2)p_3}{1000a} \\ T_d = \frac{1 + a + p_3 + 2R}{1000K} \end{array} \right. \quad (9)$$

We replace the value of K in Equation 5 by this one found in Equation 9 and taking into account for the relation $R^2 + I^2 = \omega_n^2$, we have now the transfer function $F(s)$:

$$F(s) = \frac{\omega_n^2(1 + T_{as})(1 + s/a)(1 - s/b)}{(s^2 + 2\zeta\omega_n s + \omega_n^2)(1 - s/p_3)(1 + s/b)} \quad (10)$$

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

3) Conditions for the insignificance of the poles p_3 and p_4 : We take the conditions expressed by [20] (page 281) *i.e.* the real part is five times smaller than the real part of the dominant poles, then:

- $p_3 \leq 5R$ which gives $\tau < 40$ ms and as $\tau_{ZOH} = 5$ ms we need $\tau_{sc} + \tau_{ca} < 35$ ms.
- $p_4 \leq 5R$ which gives $\tau_{ca} + \tau_{ZOH} < 36$ ms and as $\tau_{ZOH} = 5$ ms we need $\tau_{ca} < 31$ 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) Effect of the zeros: “When a zero gets closer to the origin, the overshoot increases” [21], [22].

For a negative zero (call z_0 the absolute value of this zero), in [21] (page 226), the zero has a little effect and can be neglected if $z_0/\zeta\omega_n \geq 5$.

For a positive zero (call z_0 the value of this zero), in [22], the overshoot is evaluated as follows:

$$O = e^{-\frac{\zeta(\Phi+\pi)}{\sqrt{1-\zeta^2}}} \sqrt{1 + \frac{2\zeta\omega_n}{z_0} + \left(\frac{\omega_n}{z_0}\right)^2} \text{ where } \Phi = \tan^{-1} \frac{\sqrt{1-\zeta^2}}{z_0 + \zeta\omega_n}. \text{ We see that if } \frac{z_0}{2\zeta\omega_n} \gg 1 \text{ and } \left(\frac{z_0}{\omega_n}\right)^2 \gg 1, \text{ the overshoot can be rewritten as } O = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \text{ i.e. we have the overshoot of }$$

τ_{sc} (ms)	τ (ms)	K	T_d (s)	$p_{1,2}$	p_3	p_4	O (%)	z_1	z_2	z_3
1	7	0.6119	0.0365	$-11 \pm j24.5$	-242	-333	40.66	-27.3	-285	333
2	8	0.5959	0.0377	$-11 \pm j24.5$	-206	-333	41.74	-26.5	-250	333
3	9	0.5799	0.0389	$-11 \pm j24.5$	-178	-333	42.84	-25.7	-222	333

TABLE I
VALIDATION OF THE DOMINANT POLE METHOD ($\tau_{ca} = 1$ ms, $\tau_{ZOH} = 5$ ms).

the second order system without zero. We can thus consider that if $\frac{z_0}{2\zeta\omega_n} > 5$ and $(\frac{z_0}{\omega_n})^2 > 5$ (these conditions induce that $z_0 > 110$ (with $\omega_n = 27$, $\zeta = 0.4$)), the positive zero has a little effect and can be neglected.

We consider now the three zeros:

- The negative zero $z_2 = -2/\tau$: as $\tau = \tau_{sc} + \tau_{ca} + \tau_{ZOH}$ is smaller than $h + h/2 = 15$ ms, z_2 is smaller than $\frac{-2}{0.015} = -133$. We see that $|z_2|/\zeta\omega_n > 5$. Thus the zero z_2 can be neglected.
- The positive zero $z_3 = 2/(\tau_{ca} + \tau_{ZOH})$: because of $\tau_{ca} + \tau_{ZOH} < 15$ ms, z_3 is bigger than $\frac{2}{0.015} = 133$, thus $z_3 > 110$. 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 10 can be written as Equation 11.

$$F(s) = \frac{\omega_n^2(1+T_d s)}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (11)$$

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

B. Validation of the dominant pole method

Table I shows the results obtained by considering different loop delays τ . We see the maintain of the dominant poles 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).

C. Considering the implementation of the four process control applications on CAN network

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

The comparison between the case “adaptive controller” (noted Adapt-Controller) and the case “fixed controller” (noted Fix-Controller) is presented on Table II (we use the criteria

	P ₁	P ₂	P ₃	P ₄
Fix-Controller	75.3	107	150	208
Adapt-Controller (relation QoS→QoC)	47.6	62.8	80.6	101

TABLE II
QoC: $\Delta J/J_0\%$.

ITSE in the subsection II-A) and on Figures 6 and 7 (presentation of the output time responses).

From the results of the case “fixed controller”: the performances follow the order of the priorities which is normal (lower is the priority, then higher is the time delay and less good is the performance).

From the results of the case “adaptive controller”: it improves the results (what was waited for !) 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 (Fig. 7) compared with the O (Fig. 2) of the control system without network in Section II-A.

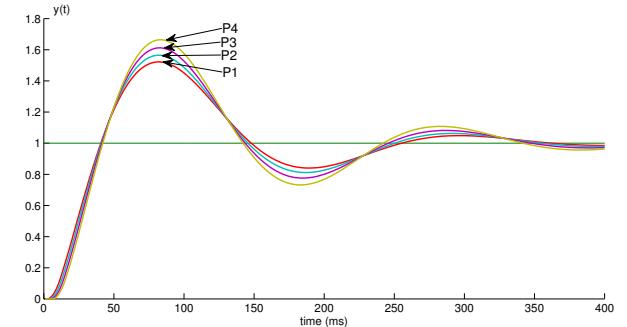


Fig. 6. Fix-Controller: time responses.

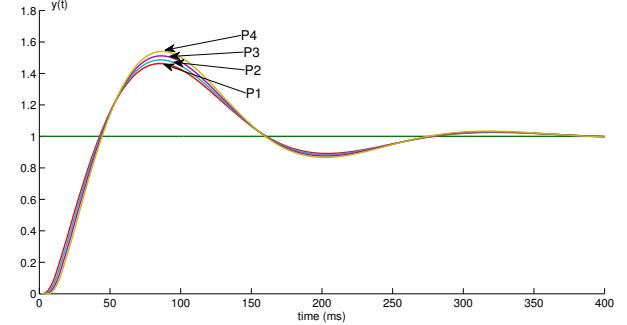


Fig. 7. Adapt-Controller: time responses.

IV. RELATION QOC→QOS

The implementation of this relation is based on the use of hybrid priorities for the scheduling of the frames. The controller is a fixed controller (the parameters of the controller are presented in Section II-A)

A. On the dynamic priority part (level 2 of the identifier field) used in hybrid priorities

The specification of a dynamic priority requires, at first, to choose a variable of the process control application, which is relevant for characterizing the frame transmission urgency, and then to express a way (how and when) for translating this variable into a priority.

Once this priority has been obtained, we have to finally specify when and how this priority is implemented in the frames of the f_{sc} and f_{ca} flows.

1) Dynamic priority specification: The variable that we consider here is the control signal u [7], the value of which is representative of the strength of the action which is requested to the actuator (higher is this value that means that higher is the error ($e = r - y$) and then more quickly the action of the actuator must be done). Then the control signal u is appropriate to obtain from it the dynamic priority.

The control signal u is translated into a priority on the basic of an increasing function of $|u|$ (Fig. 8, Equation 12) characterized by a saturation for a value $|u|$ which is $2/3|u|_{\max}$ i.e. less than $|u|_{\max}$. The idea is to be very reactive before the need becomes very important. The choice $2/3|u|_{\max}$ is an arbitrary choice.

The dynamic priority is computed by the controller each time a frame of the f_{sc} flow is received.

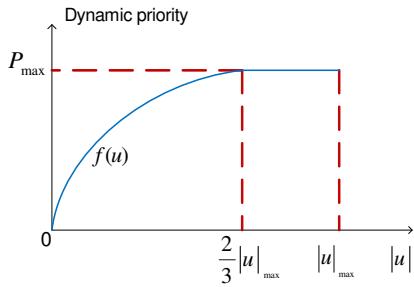


Fig. 8. Function of u .

$$f(u) = \begin{cases} P_{\max} \sqrt{\frac{|u|}{\frac{2}{3}|u|_{\max}}}, & 0 \leq |u| \leq \frac{2}{3}|u|_{\max} \\ P_{\max}, & |u| > \frac{2}{3}|u|_{\max} \end{cases} \quad (12)$$

2) Implementation of the dynamic priority in the frames of the f_{sc} and f_{ca} flows: Recall us that, as CAN is a bus, any computer can read the frames which are sent. We furthermore consider here that the computer, where the sensor task of a process control application is, knows the static priority of the f_{ca} flow (level 1 of the ID field of each frame of the f_{ca} flow) of this process control application.

The presented implementation scheme has two objectives:

- 1) The sensor task sends, at the beginning of each sampling period (i.e. at $t_k = kh$), a f_{sc} frame with a dynamic priority which has been computed by the controller during the previous sampling period.
- 2) The sequence “sending of a f_{sc} frame by the sensor task; reception of this frame by the controller; sending of the f_{ca} frame by the controller; reception of this frame by the actuator” must be an atomic action.

Considering the objective 2, the controller puts in the level 2 of the ID field the maximum priority P_{\max} [23] (in this way and remember that the f_{ca} frames have static priorities higher than those of the f_{sc} frames, the frame of the f_{ca} flow is sent immediately). Considering the objective 1, the controller puts, in the data field of the f_{ca} frame, the dynamic priority which has been computed (in this way, the sensor task, as it knows the static priority of the f_{ca} flow, learns the dynamic priority, and it will use this dynamic priority in the next sampling period). *Remark:* consequently to this behavior, at the first sampling instant ($t_0 = 0$), the sensor has no knowledge about a dynamic priority to use. So we propose to use the maximum value P_{\max} .

B. Interest of the hybrid priority compared with the static priority

We want here to compare, at first, the QoS which is provided by the different types of priorities and then to show the consequences on the QoC which is obtained.

1) Quality of Service: The QoS is here characterized by the mean value of the communication delay ($\tau_{sc} + \tau_{ca}$). We note \bar{D} this mean value. Call D_i the time delay ($\tau_{sc} + \tau_{ca}$) during the period starting at the instant t_i (i.e. period $[t_i, t_{i+1}]$) and call n the number of the sampling period during the simulation. We have $\bar{D} = \frac{\sum_{i=1}^n D_i}{n}$.

We present on Table III the QoS in terms of \bar{D} of the 4 processes provided by the static and hybrid priorities. For the static priorities, the higher the priority is, the smaller the time delay is, which is logic. For the hybrid priorities, we obtain time delays more balanced than these with static priorities. This is the result of the predominant role of the parts “dynamic priorities” compared with the parts “static priorities” (as all the processes have the same type of variable urgencies because they have the same transfer function, that explains the balanced aspect).

	P ₁	P ₂	P ₃	P ₄
Static priority	1.28	2.56	3.84	5.12
Hybrid priority (relation QoC→QoS)	2.71	3.14	3.40	3.53

TABLE III
QUALITY OF SERVICE: \bar{D} MS.

2) Quality of Control: The QoC is represented on Table IV ($\frac{\Delta t}{J_0}\%$) and on Fig. 9 ($y(t)$). The balance of QoS induces a balance of QoC. This is an interesting result (compared with the static priority use) in the perspective of implementing more applications by considering a performance threshold.

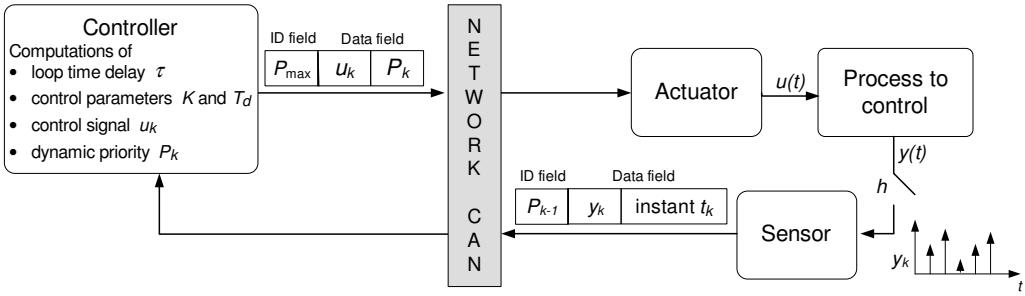


Fig. 10. Principle of the implementation of the relation $\text{QoS}=\text{QoC}$.

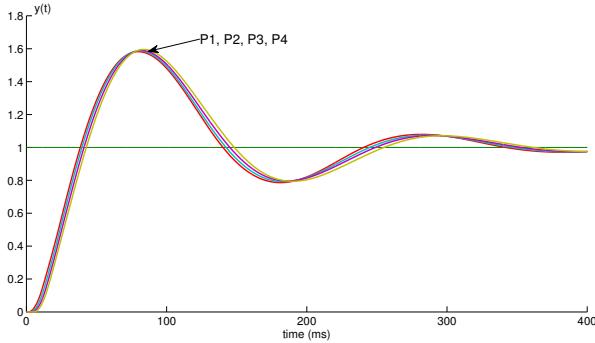


Fig. 9. Time responses (hybrid priority).

	P ₁	P ₂	P ₃	P ₄
Static priority	75.3	107	150	208
Hybrid priority (relation $\text{QoC}\rightarrow\text{QoS}$)	122	125	131	145

TABLE IV
QUALITY OF CONTROL: $\Delta J/J_0\%$.

V. RELATION $\text{QoS}=\text{QoC}$

A. Ideas

The idea is to combine the frame scheduling scheme based on the hybrid priority (*i.e.* the relation $\text{QoC}\rightarrow\text{QoS}$) and the method of compensation of time delay (*i.e.* the relation $\text{QoS}\rightarrow\text{QoC}$) in order to have a more efficient control system.

However, concerning the loop time delay compensation, in the sampling period k , we cannot consider here that the controller can use the value of the loop time delay of the sampling period $(k-1)$ because now, taking into account for the dynamic priority used by the sensor task, the time delay $(\tau_{sc} + \tau_{ca})$ changes every period.

Then the controller must make the delay compensation in the sampling period k by knowing the loop time delay of this sampling period k . We explain now this implementation.

B. Principle of the implementation of the relation $\text{QoS}=\text{QoC}$

This principle, relatively to the sampling period starting at t_k , is represented on Fig. 10 where we indicate the content of the f_{sc} and f_{ca} frames and the computations done by the controller.

At the instant t_k , the sensor task generates the f_{sc} frame which includes in the ID field the dynamic priority P_{k-1} *i.e.*

dynamic priority computed by the controller in the previous period (this expresses the implementation of the relation $\text{QoC}\rightarrow\text{QoS}$). The data field of this frame includes the value of the instant t_k and the sampled value of the output y_k (these values represent the contribution of the sensor task to the relation $\text{QoS}\rightarrow\text{QoC}$).

When the controller task receives this frame, it undertakes the following actions (we suppose that these actions are instantaneous):

- With the value of t_k , it deduces (by comparison to its local clock; we suppose that the clocks are well synchronized) the value of the time delay τ_{sc} ; furthermore, as the level 2 of the ID field of the f_{ca} frame includes P_{max} , the f_{ca} frame is sent immediately without waiting time (then $\tau_{ca} = D_{ca}$); and as the value of τ_{ZOH} is known ($\tau_{ZOH} = h/2$), the controller has the knowledge of the loop time delay τ in this sampling period and can then compute the new parameters K and T_d . This compensation expresses the implementation of the relation $\text{QoS}\rightarrow\text{QoC}$.
- Then, with the value y_k and the new parameters K and T_d , the controller computes the value of the control signal u_k . With this value u_k , the controller computes the dynamic priority P_k which will be used in the next sampling period by the sensor task (this characterizes the setting up of the relation $\text{QoC}\rightarrow\text{QoS}$ for the next sampling period).

C. Performance evaluation and summary of obtained results

We still consider the implementation of 4 applications (P₁, P₂, P₃, P₄) studied in the previous sections.

The results ($\Delta J/J_0\%$) of the implementation of the relation $\text{QoS}=\text{QoC}$ and the other relations (static priority + Fix-Controller, $\text{QoS}\rightarrow\text{QoC}$, $\text{QoC}\rightarrow\text{QoS}$) are represented on Table V. These results which are represented graphically on Fig. 11 obviously show the balanced performances provided by the hybrid priority ($\text{QoC}\rightarrow\text{QoS}$) compared with the static priority and by the bidirectional relation $\text{QoS}=\text{QoC}$ compared with the relation $\text{QoS}\rightarrow\text{QoC}$.

From these results, we can say that, if we have a constraint of performance which cannot be exceeded (not too small), the hybrid priority allows to implement more applications than the static one and, extensionally, the bidirectional relation $\text{QoS}=\text{QoC}$ allows to implement more applications than the relation $\text{QoS}\rightarrow\text{QoC}$.

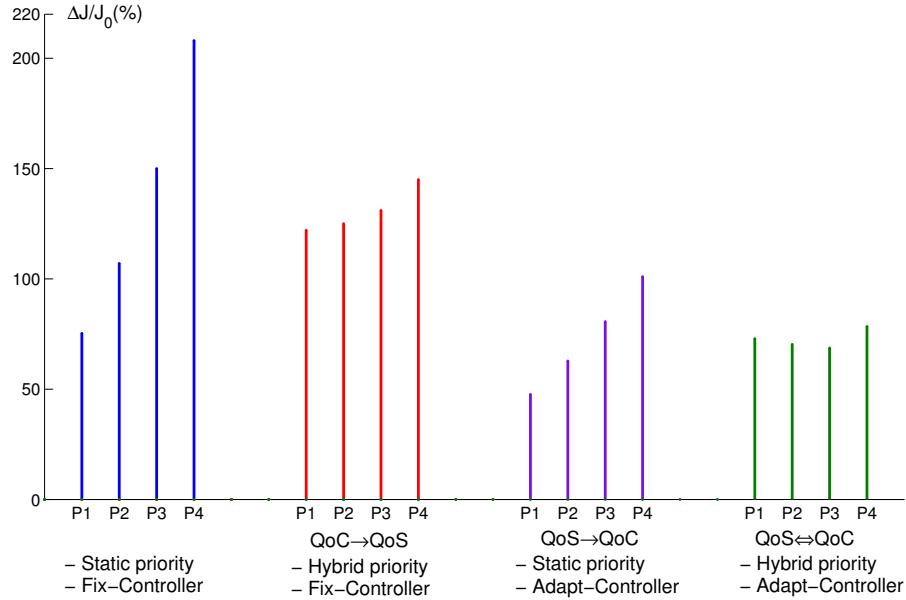


Fig. 11. Graphic representation of the QoC ($\Delta J/J_0\%$) of the 4 applications.

	P ₁	P ₂	P ₃	P ₄
Static priority + Fix-Controller	75.3	107	150	208
QoS→QoS (Hybrid priority + Fix-Controller)	122	125	131	145
QoS→QoC (Static priority + Adapt-Controller)	47.6	62.8	80.6	101
QoS=QoC (Hybrid priority + Adapt-Controller)	72.9	70.3	68.7	78.4

TABLE V
QOC SUMMARY ($\Delta J/J_0\%$).

The visualizations of the time responses provided by the different relations are represented on Fig. 12.

VI. CONCLUSION

In this paper, we have presented three following points:

- The first is the implementation of the relation $QoS \rightarrow QoC$ on the basis of the dominant pole method and of static priorities for the frame scheduling. This relation $QoS \rightarrow QoC$ shows the interest of the delay compensation to improve the QoC performances provided by static priorities only.
- The second is the implementation of the relation $QoC \rightarrow QoS$ on the basis of hybrid priorities for the frame scheduling. The hybrid priority is characterized by, on one hand, a static part representing the uniqueness of the flow, and on the other hand, a dynamic part that represents the urgency of transmission (this dynamic part is expressed from a function of the control signal).

This relation $QoC \rightarrow QoS$ (*i.e.* hybrid priority) gives, in comparison to the static priority case, balanced performances for the different process control applications.

- The final is the implementation of the relation $QoS = QoC$. This implementation can be considered as a combination of the relations $QoS \rightarrow QoC$ and $QoC \rightarrow QoS$.

The results, which are obtained, show the interest of the joint action of the hybrid priorities and the delay compensation (by the hybrid priorities, we introduce the balance aspect compared with the static priority; by the delay compensation, we maintain the balance aspect while improving the QoC and then we can consider the possibility to implement more applications than with the relation $QoS \rightarrow QoC$).

The relation $QoS = QoC$ is a co-design approach for Networked Control Systems.

The further work should be the following points: the utilization of other compensation methods for time delays for the relation $QoS \rightarrow QoC$ (for example, maintenance of the phase margin); the consideration of other types of controller (PID for example) and the consideration of other types of process control applications.

Still furthermore, the study of this relation ($QoS = QoC$) might be also improved by the consideration of theoretical problems (in particulars, stability conditions when the online control law parameters change from sampling period to sampling period).

ACKNOWLEDGEMENTS

This work has been partly funded by the project OSEO AMIC-TCP (Architecture Multiplexage Informatique Communication pour Transport en Commun de Personnes). We want to express here our sincere thanks.

REFERENCES

- [1] M. S.Branicky, V.Liberatore, and S. M.Phillips, "Networked Control Systems Co-Simulation for Co-Design," in *Proc. 2003 American Control Conference (ACC'03)*, vol. 4, Debver, USA, June 2003, pp. 3341–3346.
- [2] Y.-Q. Song, "Networked Control Systems: From Independent Designs of the Network QoS and the Control to the Co-design," in *Proc. 8th IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems (FeT 2009)*, Hanyang University, Republic of Korea, May 2009.

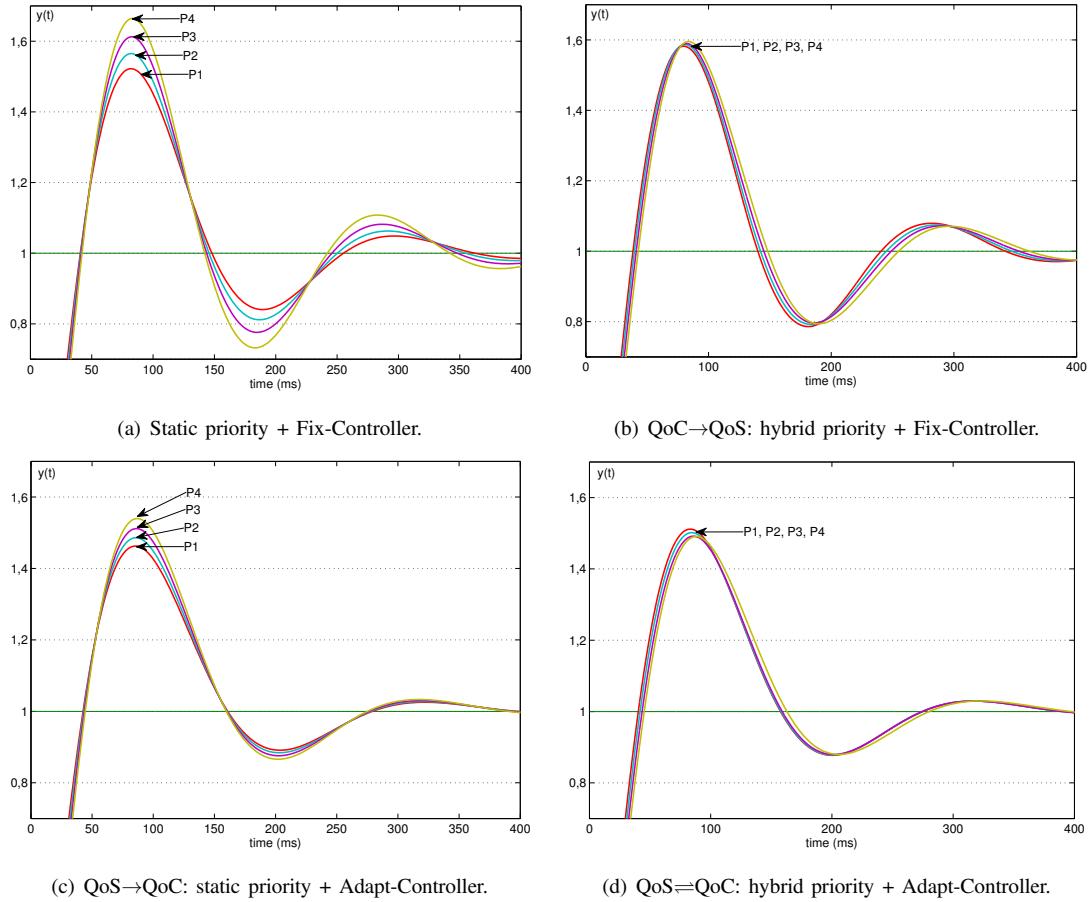


Fig. 12. Time responses.

- [3] P. Martí, "Analysis and design of real-time control systems with flexible timing constraints," Ph.D. dissertation, Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial, Univ Politècnica de Catalunya, Barcelona, Spain, June 2002.
- [4] M. S.Branicky, S. M.Phillips, and W. Zhang, "Scheduling and Feedback Co-Design for Networked Control Systems," in *Proc. 41st IEEE Conference on Decision and Control (CDC'02)*, Las Vegas, Nevada, USA, Dec 2002.
- [5] A. Cervin, "Integrated control and real-time scheduling," Ph.D. dissertation, Department of Automatic Control, Lund Institute of Technology, Sweden, April 2003.
- [6] M. Velasco, P. Martí, R. Castaño, J. Guardia, and J. Fuertes, "A CAN Application Profile for Control Optimization in Networked Embedded Systems," in *Proc. 32nd Annual Conference of the IEEE Industrial Electronics Society (IECON'06)*, Paris, France, Nov 2006, pp. 4638 – 4643.
- [7] G. Juanole and G. Mouney, "Networked Control Systems: Definition and Analysis of a Hybrid Priority Scheme for the Message Scheduling," in *Proc. 13th IEEE conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'07)*, Daegu, Korea, Aug 2007.
- [8] H. Nguyen Xuan, 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 *Proc. 8th IEEE International Workshop on Factory Communication Systems (WFCS 2010)*, Nancy, France, May 2010.
- [9] H. Nguyen Xuan, "Networked Control Systems Quality of Service aware Quality of Control adaptation," in *Proc. 16th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2011)*, Toulouse, France, Sep 2011.
- [10] B. GmbH. (1991) Can specification 2.0 (part a). [Online]. Available: www.can-cia.de/fileadmin/cia/specifications/CAN20A.pdf
- [11] K. J. Åström and B. Wittenmark, *Computer-controlled systems: Theory and design*, 3rd ed. Prentice Hall, 1997.
- [12] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 4th ed. Prentice Hall, 2002.
- [13] M. Ohlin, D. Henriksson, and A. Cervin, *TrueTime 1.5 – Reference Manual*, Lund Institute of Technology, Sweden, January 2007.
- [14] N. Navet, "Evaluation de performances temporelles et optimisation de l'ordonnancement de tâches et messages," Ph.D. dissertation, Institut National Polytechnique de Lorraine, Nancy, France, Nov 1999.
- [15] K. Zuberi and K. Shin, "Scheduling messages on controller area network for real-time CIM applications," *IEEE Transactions On Robotics And Automation*, vol. 13, no. 2, pp. 310–314, 1997.
- [16] K. Zuberi and K. Shin, "Design and Implementation of Efficient Message Scheduling for Controller Area Network," *IEEE Transactions on Computers*, vol. 49, no. 2, pp. 182–188, 2000.
- [17] G. Walsh and H. Ye, "Scheduling of Networked Control Systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 57–65, Feb 2001.
- [18] J. Yepez, P. Martí, and J. Fuertes, "The large error first (lef) scheduling policy for real-time control systems," in *Proc. 24th IEEE Real-Time Systems Symposium (RTSS'03)*, Cancun, Mexico, Dec 2003.
- [19] G. Juanole, G. Mouney, C. Calmettes, and M. Peca, "Fundamental Considerations for Implementing Control Systems on a CAN Network," in *Proc. 6th IFAC International Conference on Fieldbus Systems and their Applications (FET 2005)*, Puebla, Mexico, Nov 2005.
- [20] B. C. Kuo and F. Golnaraghi, *Automatic Control System*, 8th ed. John Wiley & Sons, 2003.
- [21] R. C.Dorf and R. H.Bishop, *Modern Control Systems*, 7th ed. Addison-Wesley, 1995.
- [22] B.-M. Kwoon, H.-S. Ryu, and O.-K. 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, Dec 2000.
- [23] H. Nguyen Xuan, "Réseaux de communication et applications de contrôle-commande," Ph.D. dissertation, Univ de Toulouse, INSA, LAAS, Toulouse, France, Dec 2011.