



**HAL**  
open science

# On interplays between the Quality of Service and the Quality of Control for a co-design approach in Wireless Networked Control Systems (WNCSs)

Xuan Hung Nguyen, Guy Juanole

► **To cite this version:**

Xuan Hung Nguyen, Guy Juanole. On interplays between the Quality of Service and the Quality of Control for a co-design approach in Wireless Networked Control Systems (WNCSs). 2012. hal-00668746v1

**HAL Id: hal-00668746**

**<https://hal.science/hal-00668746v1>**

Submitted on 10 Feb 2012 (v1), last revised 27 Apr 2013 (v4)

**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.

# On interplays between the Quality of Service and the Quality of Control for a Wireless Networked Control Systems (WNCSs) co-design approach

Xuan Hung Nguyen<sup>1,2</sup>, Guy Juanole<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  
xhnguyen@laas.fr, juanole@laas.fr

## Abstract

*We consider the implementation of process control applications on a Wireless Local Area Network (WLAN) with a collision-free CSMA MAC protocol called CANlike protocol (adaptation of the MAC protocol of the CAN bus to the wireless context). The co-design approach is based on links between the Quality of Control (QoC) provided by the controller and the Quality of Service (QoS) provided by the frame scheduling in the MAC layer of the WLAN. We present, first, the implementation of the relation  $QoS \rightarrow QoC$  on the base of the delay compensation method called dominant pole method, and second, the implementation of the relation  $QoC \rightarrow QoS$  on the base of the hybrid priorities for the frame scheduling. Finally, we show the interest of the relation  $QoS \rightleftharpoons QoC$  i.e. the joint action of the delay compensation and the role of the hybrid priority in order to have a more efficient WNCS design.*

## 1. 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 [17], [18].

Several works [9, 16, 2] have considered the aspects Automatic Control and Computer Science (Task scheduling) and have shown the interest of a co-design approach.

Other works have shown the co-design interest by considering the aspects Automatic Control and Communication Network [12, 19, 5, 13]. But the aspect communication network has mainly concerned the wired networks.

However today, we see more and more the use of wireless networks in many areas and, in particular we see also the development of WNCSs then it becomes very important to work on the co-design of WNCSs.

We want in this paper, make such a study by considering WNCSs where each node is in the transmission range of the other nodes (one hop communication). In the WLANs context, the MAC protocol has a basic role as it

implements the scheduling of the frames of the two flows of each distributed process control application (flow  $f_{sc}$  between the sensor task and the controller task; flow  $f_{ca}$  between the controller task and the actuator task). However we consider a MAC protocol of the CSMA type (the CSMA type is more flexible than the TDMA type with respect to changes in a network i.e. adding or withdrawing a node or an application). Furthermore the CSMA type is also more suitable for sporadic traffic.

However, the big drawback of the protocols based on the CSMA technique is that collisions can occur (an example is the protocol IEEE802.11 DCF [3] still called DCF-WiFi) and then we cannot get QoS guarantees which are necessary for applications (like Networked Control System) which have time constraints. Obtaining QoS guarantees with CSMA type protocols is possible [11] by associating priorities to the frames of the flows (the role of the priorities is to transform what would be a “collision situation” with a strict CSMA type protocol into a “winner-looser(s) situation”; the winner is the frame which has the highest priority among the frames trying to access the channel). In this study, we consider a protocol that we have defined in previous works [11], [7] and which is called CANlike protocol (it is based on an adaptation, defined in [20], [15], of the CAN protocol, used in the CAN bus [4], to the wireless context).

The final aim of this paper is to show the interest of a co-design of the frame scheduling in the WLAN and of the controller of the process control application on the basis of a bidirectional relation between the QoC provided by the controller, and the QoS provided by the scheduling of the frames of the WLAN (relation  $QoS \rightleftharpoons QoC$ ) i.e. we have both relation  $QoC \rightarrow QoS$  (QoS is QoC driven i.e. Application performances aware dynamic QoS adaptation) and relation  $QoS \rightarrow QoC$  (QoC is QoS driven i.e. network performances aware dynamic QoC adaptation). We have already done such works in the context on the wired CAN bus [5], [13], [12], [10]: firstly, on the interest of the relation  $QoC \rightarrow QoS$  on the basis of the use of hybrid priorities for the frame scheduling, secondly, on the interest of the relation  $QoS \rightarrow QoC$  where the controller is aware on-line of the time delay in the loop of a process control applica-

tion and adapts, in consequence, its parameters by using a dominant pole method, and thirdly, on the benefit to combine these two relations *i.e.* the relation  $QoS \rightleftharpoons QoC$ .

We want to present in this paper a study of the same type for the context WLAN.

This study is done by using the simulator TrueTime [14], a tool box based on Matlab/Simulink which allows to simulate real-time distributed control systems.

This paper includes three following sections: the section 2 presents the context of the study; the section 3 presents the implementation of the relation  $QoS \rightarrow QoC$ ; the section 4 presents the implementation of the relation  $QoC \rightarrow QoS$ ; the section 5 presents the implementation of the bidirectional relation  $QoS \rightleftharpoons QoC$ ; the section 6 presents the conclusion.

## 2 Context of the study

### 2.1 Process control application which is considered

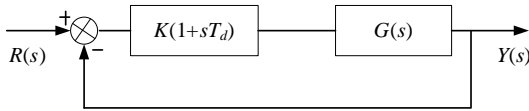


Figure 1. Continuous control system.

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  $45^\circ$  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

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

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

where  $\omega_n$  is the natural pulsation and  $\zeta$  is the damping coefficient ( $\omega_n^2 = 1000K$ ;  $2\zeta\omega_n = 1 + 1000KT_d$ ).

We have:  $\omega_n = 27$  rad/s;  $\zeta = 0.4$ ; the two poles  $p_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$ , *i.e.*  $p_{1,2} = -11 \pm j24.5$ ; the overshoot  $O = 33.8\%$ ; the settling time (at 2%)  $t_s = 284$  ms; the time response (represented on Fig. 2).

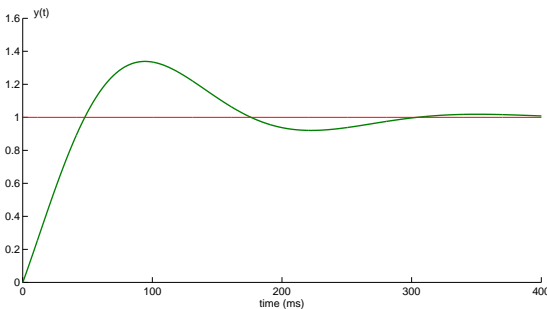
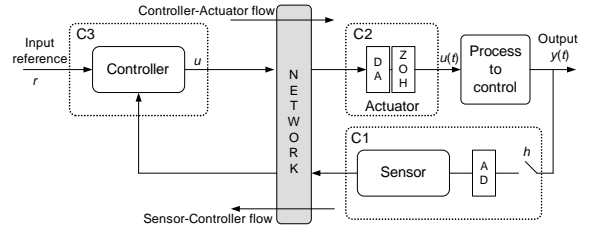


Figure 2. Time response  $y(t)$ .

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

The implementation on a network which requires the sampling of the output  $y(t)$  is represented on Fig. 3.



(DA: Digital Analog Converter; AD: Analog Digital Converter; ZOH: Zero Order Hold)

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

We call  $h$  the sampling period (which is defined by considering the following formula  $\omega_n h \in [0.1; 0.6]$  [1]) and  $t_k$  the sampling instant ( $t_k = kh$  with  $k = 0, 1, 2, \dots$ ) which is the start of the period  $k$ . Here we take  $h = 10$  ms.

We have three computers: the computer C1 where is located the sensor task which samples the output signal and sends the sampled output on the network; the computer C2 where is located the controller task which works out the command signal  $u$  from difference between the input reference  $r$  and the sampled outputs and sends it on the network; the computer C3 where is located the actuator task which receives the command signal  $u$  from the network and applies this  $u$  to the process to be controlled. The sensor task is time-triggered while the controller task and the actuator task are event-triggered.

The implementation on a network produces two frame flows: the Sensor-Controller flow (noted the  $f_{sc}$  flow) and the Controller-Actuator flow (noted  $f_{ca}$  flow).

The carrying out of the process control application is characterized by several delays: computational delays in the running of the tasks (sensor, controller, actuator) in their computers; communication delays in the transmission of the  $f_{sc}$  frames (noted  $\tau_{sc}$ ) and the  $f_{ca}$  frames (noted  $\tau_{ca}$ ). Note furthermore that the ZOH behavior can be seen as a pure delay  $\tau_{ZOH} = h/2$  [8].

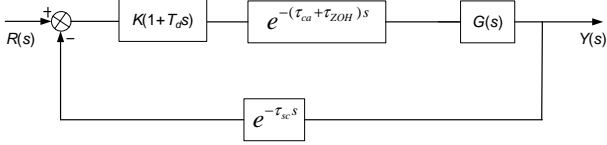
In this work, we only consider the time delays  $\tau_{sc}$ ,  $\tau_{ca}$  and  $\tau_{ZOH}$ . The computational and functional delays are neglected. The time delay of the closed loop in each sampling period is  $\tau = \tau_{sc} + \tau_{ca} + \tau_{ZOH}$ .

### 2.3 Model of the implementation of a process control application on a network

This model can be represented by the continuous model given on Fig. 4 where a time delay  $\tau$  is represented by the transfer function  $e^{-\tau s}$ .

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

$$F(s) = \frac{K(1 + T_d s)e^{-(\tau_{ca} + \tau_{ZOH})s}G(s)}{1 + K(1 + T_d s)e^{-\tau s}G(s)} \quad (2)$$



**Figure 4. Control system with time delays.**

The exponential function can be replaced with the Padé first order approximation *i.e.*  $e^{-\tau s} \approx \frac{-s+2/\tau}{s+2/\tau}$  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:

$$F(s) = \frac{1000Ka(1+T_d s)(1+s/a)(1-s/b)}{f_3(s)(1+s/b)} \quad (3)$$

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

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

## 2.4 CANlike protocol

This protocol has been totally explained and presented in [11], [7]. Here we only present a summary.

### Main ideas

We have, at the beginning of each frame, an ID (identifier) field which carries the priority of the frame and which allows to implement bit by bit a tournament phase between the different frames which are contending for the channel access. As, in a wireless context, the transceivers cannot transmit and receive at the same time, we consider the proposal, which has been done in [20], [15], for the bits of the ID field: a dominant bit consists, during its duration, in the sending of a carrier wave; a recessive bit consists, during its duration, in the sensing of the channel.

So, in each MAC entity, the tournament on each bit has the following characteristics:

- The MAC entity has a dominant bit: this entity wins by definition the tournament related to this bit and then continues the tournament on the next bit.
- The MAC entity has a recessive bit: either it senses a carrier wave, then it loses the tournament related to this bit and stops the tournament phase, or it senses nothing (that means that there is no dominant bit sent by another MAC entity) and then it can continue the tournament on the next bit.

The winner, at the end of the analysis of all the bits, can then send its frame.

Three other points are important [11], [7]:

- A MAC entity needs, before to start a tournament, to observe that the channel is idle during some time. The length of this time (called observation time of the idle channel) must be such that we cannot have intrusions during the progress of a tournament.

- The starting of a tournament by a MAC entity requires that this MAC entity sends a synchronization bit (still a carrier wave) the role of which is to inform the other MAC entities, which participate in the tournament, of the arrival of the ID bits and then to constitute a time reference for the analysis of the ID bits. A MAC entity, which receives a synchronization bit, but has not sent itself a synchronization bit, does not participate in the tournament.
- Taking into account for the asynchronism and the distance between the MAC entities which participate to a tournament, we need a guard time which is added at the end of the synchronization bit and at the end of each bit of the ID field.

### The parameters of the CANlike protocol

Call  $\tau_{TT}$ ,  $\tau_{ST}$ ,  $\tau_{PT}$  and  $n$  respectively the turnaround time of a transceiver, the sensing time of a transceiver, the propagation time between the two more remote MAC entities and the number of bits of the ID field.

The parameters of the CANlike protocol, *i.e.*  $l_b, l_s, t_g$  and TOBS respectively the length of bit of the ID field, the length of the synchronization bit, the length of the guard time (of the synchronization bit and an ID bit) and the observation time of the idle channel, are given on Tab. 1.

$l_b$	$2\tau_{PT} + \tau_{TT} + \tau_{ST}$
$l_s$	
$t_g$	$2\tau_{PT} + \tau_{TT}$
TOBS	$(n+1)(4\tau_{PT} + 2\tau_{TT} + \tau_{ST})$

**Table 1. Parameters of the CANlike protocol.**

The duration of TOBS is obtained by considering the case where we have an ID field with only recessive bits (which gives the maximum time when the channel is idle during the competition based on the ID field ( $n(l_b + t_g)$ ) and then the time TOBS must be higher than this value. We take:

$$\text{TOBS} = (n+1)(l_b + t_g) = (n+1)(4\tau_{PT} + 2\tau_{TT} + \tau_{ST})$$

Finally, we can obtain the duration linked to the implementation of a tournament:

$$\text{TOBS} + (l_s + t_g) + n(l_b + t_g) = 2(n+1)(4\tau_{PT} + 2\tau_{TT} + \tau_{ST})$$

*Remark:* The bits (synchronization; ID field) have a duration which is considerable longer than a data bit (bit of a frame). The winner transmits the frame at the bit rate allowed by the radio transceiver.

## 2.5 On the priorities associated to the frames

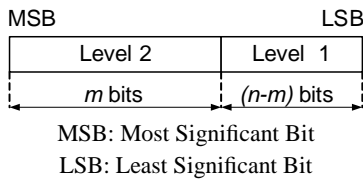
### Generalities

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

However, we can have also hybrid priorities *i.e.* priorities 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 of dynamic priority has a great interest for systems where we have transient behaviors which require a good reactivity. That is particularly true for the process control applications.

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 [5].



**Figure 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.

We will here consider the two cases (static priority, hybrid priority) for the scheduling of the frames of the flows of a process control application.

### On the priorities for the frames of the $f_{sc}$ and $f_{ca}$ flows of a process control application

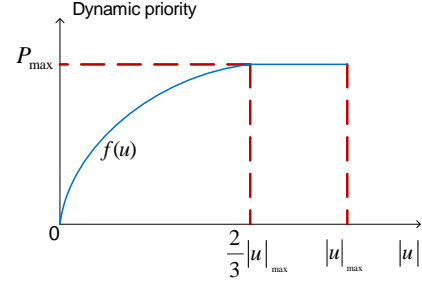
#### Case of the static priorities

Call  $P_{sc}$  and  $P_{ca}$  the priorities of respectively the  $f_{sc}$  flow and the  $f_{ca}$  flow of a process control application. It has been shown in a previous work [6] that we must have  $P_{ca} > P_{sc}$ . In the context of a shared network, this condition allows the controller to act faster after the reception of a  $f_{sc}$  frame. In this study, we consider this result.

#### Case of the hybrid priorities

Concerning the level 1 *i.e.* the static priority, we consider the result that we have just presented. Concerning the level 2, *i.e.* the dynamic priority, its specification requires, at first, to choose a variable of the process control application which is relevant for characterizing the transmission urgency and, then, to express a way for translating this variable into a priority.

The variable that we consider here is the control signal  $u$  [5], the value of which is representative of the strength of the action which is requested to the actuator (the higher this value is, the higher the error ( $e = r - y$ ) is and then the more quickly the action of the actuator must be done).



**Figure 6. Dynamic priority (function of  $u$ ).**

$$f(u) = \begin{cases} P_{\max} \sqrt{\frac{|u|}{\frac{2}{3}|u_{\max}}}}, & 0 \leq |u| \leq \frac{2}{3}|u_{\max} \\ P_{\max}, & |u| > \frac{2}{3}|u_{\max} \end{cases} \quad (4)$$

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 basis of an increasing function of  $|u|$  (Fig. 6, Equa. 4) 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 it receives a frame of the  $f_{sc}$  flow.

## 2.6 On the dominant poles method

### Main ideas

Consider that the controller has the knowledge of the loop time delay  $\tau$ . With this knowledge, the controller has to 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 Equa. 1). As the transfer function of the system implemented on the network (Equa. 3) 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 Equa. 1 (*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 small with respect to 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, we have three zeros:  $z_1 = -1/T_d$ ,  $z_2 = -2/\tau$ ,  $z_3 = 2/(\tau_{ca} + \tau_{ZOH})$  (cf. Equa. 3), thus we have to evaluate the influence of these zeros on the performance.

## The computations in the controller for the maintenance of the dominant poles

Consider the polynomial  $f_3(s)$  in the denominator of Equa. 3, 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  $p_{1,2} = R \pm jI$ :

$$\begin{aligned} & (s - p_1)(s - p_2)(s - p_3) \\ &= s^3 - (2R + p_3)s^2 + (2Rp_3 + R^2 + I^2)s - (R^2 + I^2)p_3 \end{aligned} \quad (5)$$

By identifying  $f_3(s)$  with Equa. 5, we get the relations which allow to determine the value of  $p_3, K$  and  $T_d$ :

$$\begin{cases} p_3 = -\frac{a^3 + (2 + 2R)a^2 - (R^2 + I^2)a}{a^2 - 2Ra + R^2 + I^2} \\ K = -\frac{(R^2 + I^2)p_3}{1000a} \\ T_d = \frac{1 + a + p_3 + 2R}{1000K} \end{cases} \quad (6)$$

We replace the value of  $K$  in Equa. 3 by this one found in Equa. 6 and taking into account for the relation  $R^2 + I^2 = \omega_n^2$ , we have now the transfer function:

$$F(s) = \frac{\omega_n^2(1 + T_d s)(1 - s/z_2)(1 - s/z_3)}{(s^2 + 2\zeta\omega_n s + \omega_n^2)(1 - s/p_3)(1 - s/p_4)} \quad (7)$$

In the context of an application, we have to check that the poles  $p_3$  and  $p_4$  are insignificant and the influence of the three zeros ( $-1/T_d, z_2$  and  $z_3$ ) on the overshoot.

## 2.7 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 following formula:

$$\begin{cases} P_k = K e_k \\ D_k = \frac{T_d}{Nh + T_d} D_{k-1} + \frac{NK T_d}{Nh + T_d} (e_k - e_{k-1}) \\ u_k = P_k + D_k \end{cases} \quad (8)$$

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

## 2.8 Global system which will be analyzed

We will consider the implementation of four process control applications ( $P_1, P_2, P_3, P_4$ ) on a WLAN. These processes  $P_i$  ( $i = 1, 2, 3, 4$ ) are identical to the one which has been presented in Sect. 2.1 and are synchronous (the sampling instants are identical). We suppose that the data field of the  $f_{sc}$  and  $f_{ca}$  frames are of 16 bits. The controller tasks, the sensor tasks and the actuator tasks of the 4 processes 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 sharing the network.

Note that, the static priorities of the  $f_{cai}$  and  $f_{sci}$  flows of each process  $P_i$  are such that  $P_{cai} > P_{sci}$  (Sect. 2.5.2) and the static priorities of the flows of the 4 processes are arranged in the following order:

$$P_{ca1} > P_{ca2} > P_{ca3} > P_{ca4} > P_{sc1} > P_{sc2} > P_{sc3} > P_{sc4}$$

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

Concerning the WLAN, which will be limited to the MAC layer, we will consider as the base of our analysis the protocol IEEE802.11 DCF (DCF-WiFi) with a bit rate of 1 Mbits/s in the physical layer and we will also use the frame structure of this protocol for the considered CANlike protocol. So, the characteristics of the considered protocols are as follows:

### Protocol DCF-WiFi:

The control informations are 192 bits for the physical layer + 272 bits for the MAC layer then the length of a frame:  $192 + 272 + 16 = 480$  bits; DIFS =  $50 \mu s$ ; Contention window =  $[620 \mu s; 20460 \mu s]$ ;  $\tau_{TT} = 5 \mu s$ ;  $\tau_{ST} = 15 \mu s$ ;  $\tau_{PT}$  is neglected.

### Protocol CANlike:

We will consider both static priorities (call CANlike-sta this protocol CANlike) and hybrid priorities (call CANlike-hyb this protocol CANlike).

The length of the ID field is  $n = 8$  bits. For CANlike-hyb, we consider the ID field of 4 MSB bits for the level 2 and 4 LSB bits for the level 1 (Fig. 5).

The length of a frame is 480 bits, then, as the bit rate is 1 Mbits/s, the duration of a frame is  $480 \mu s$ .

$l_b = l_s = 20 \mu s$ ,  $t_g = 5 \mu s$ , then, the duration linked to the tournament (cf. Sect. 2.4.2) is  $450 \mu s$ .

The total duration for a winner to send a  $f_{sc}$  frame (noted  $D_{sc}$ ) or a  $f_{ca}$  frame (noted  $D_{ca}$ ):  $450 + 480 = 930 \mu s$ . So, as we have to transmit 8 frames during each period of  $h = 10$  ms, this can be done ( $8 \times 930 = 7440 \mu s < h$ ) and then the network is not overloaded.

## 2.9 Criteria of the QoC evaluation

The 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 Sect. 2.1 (noted  $J_0$ ) is  $J_0 = 9.4562 \cdot 10^{-4}$ .  $J_0$  will be considered as the reference value for the study done along the paper.

When we will consider the process control applications implemented on the network, the performance criteria will be 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.

Note that we will also consider the overshoot ( $O\%$ ) and visualize the time response ( $y(t)$ ).

## 2.10 Results to have in mind

We have done the evaluation of QoC ( $\Delta J/J_0 \%$ ) of the 4 processes ( $P_1, P_2, P_3, P_4$ ) by considering the protocols DCF-WiFi and CANlike-sta. The results are on Tab. 2.

For DCF-WiFi, we have done 20 simulations and we give the mean value (the results of the different process control applications are balanced) and the maximum gap existing between the results of one simulation and the

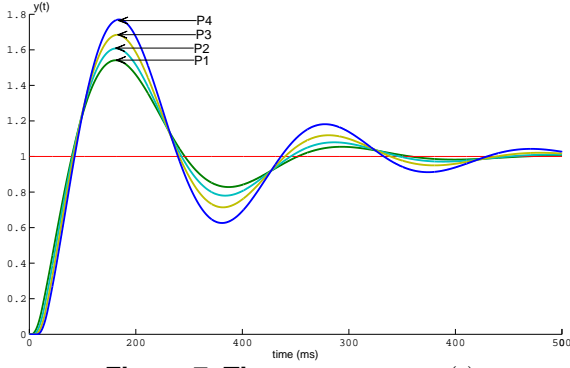


Figure 7. Time responses  $y(t)$ .

	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>
DCF-WiFi: -mean value	139	136	134	161
-max gap	132	84	156	140
CANlike-sta	88.9	145	234	382

Table 2. DCF-WiFi and CANlike-sta:  $\frac{\Delta J}{J_0}\%$ .

mean value. The balance between the mean values is normal because the process control applications are not differentiated but the possibility of maximum gaps very high indicates that the protocol DCF-WiFi cannot be used for such real time applications.

For CANlike-sta, we see that obviously the priority discriminates the processes ( $P_i$  is better than  $P_j$  with  $i < j$ , cf. Tab. 2 and Fig. 7) and so the higher the priority is, the less the time delay is, then the better the performance is.

### 3 Relation QoS $\rightarrow$ QoC

The strict study of this relation is based on the use of CANlike-sta protocol (in this way, using static priorities, the QoS is independent of the QoC). We then use the dominant pole method for the delay compensation.

#### 3.1 Essential problem

The implementation of this relation, by the controller of a process control application during each sampling period  $k$ , requires that the controller gets the communication time delay ( $\tau_{sc} + \tau_{ca}$ ) of this period  $k$ . But it is not possible ( $\tau_{sc}$  can be known but  $\tau_{ca}$  is unknown). However, with static priorities and taking into account that the network is not overloaded (Sect. 2.8), the delay ( $\tau_{sc} + \tau_{ca}$ ), for each process, is the same whatever the period may be. This aspect suggests to the controller that it should use the value ( $\tau_{sc} + \tau_{ca}$ ) of the previous period ( $k - 1$ ). The sensor task allows the controller to get this knowledge. Actually a sensor task can evaluate, in any sampling period noted ( $k - 1$ ), the value ( $\tau_{sc} + \tau_{ca}$ ) by noting the time difference between the sampling instant  $t_{k-1}$  (when the  $f_{sc}$  frame must be sent) and the reception instant of the corresponding  $f_{ca}$  frame by the actuator task (that supposes that the computer where the sensor task is located, knows the static priority of the  $f_{ca}$  flow and then can read the corresponding  $f_{ca}$  frame; and note furthermore that we

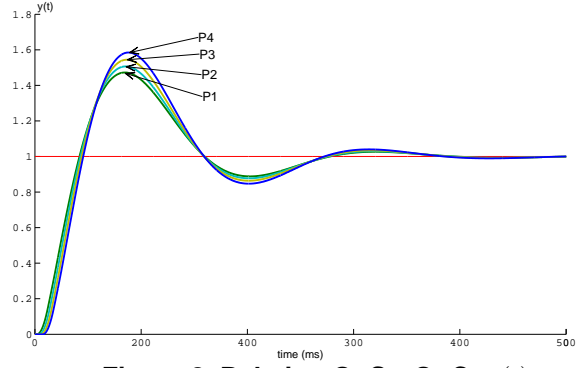


Figure 8. Relation QoS  $\rightarrow$  QoC:  $y(t)$ .

P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>
54.0	77.6	106	142

Table 3. Relation QoS  $\rightarrow$  QoC:  $\Delta J/J_0\%$ .

consider that the propagation time can be neglected).

Then, at the next sampling instant  $t_k$ , the sensor task will put in the data field of the  $f_{sc}$  frame, the values of the sampled output  $y_k$  and the time delay ( $\tau_{sc} + \tau_{ca}$ ) which will be used by the controller. In brief, about the controller:

- With the knowledge of ( $\tau_{sc} + \tau_{ca}$ ), it computes the loop delay ( $\tau_{sc} + \tau_{ca} + \tau_{ZOH}$ ) and then computes on-line the new parameters  $K$  and  $T_d$  (cf. Equa. 6).
- With the knowledge of these new parameters  $K$  and  $T_d$  and the value of  $y_k$ , it computes the control signal  $u_k$  (cf. Equa. 8) and sends it in a  $f_{ca}$  frame.

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

Concerning the dominant pole method (Sect. 2.6), we have shown [12], [11] that, in the context of this study, the poles  $p_3$  and  $p_4$  are insignificant and that the effect of the zeros  $z_2$  and  $z_3$  can be neglected. Only the zero  $z_1 = -1/T_d$  (as  $T_d$ , in order to allow the delay compensation, increases with the delay) must be considered. We have already shown that, when the delay increases, the zero  $z_1$  moves closer to the origin and it increases the overshoot.

*Important remark:* in DCF-WiFi, as we have collisions and the resolution of collisions is done by using a random time (backoff mechanism), the time ( $\tau_{sc} + \tau_{ca}$ ) is variable and unknown in each sampling period. Thus this relation cannot be implemented with the DCF-WiFi protocol.

#### 3.2 Results

The QoC ( $\Delta J/J_0\%$ ) and time responses for 4 processes are represented respectively on Tab. 3 and Fig. 8. We see that, this relation QoS  $\rightarrow$  QoC improves the results compared with the results on Tab. 2. This improvement is normal because here we have compensated the time delay. Note that, we have not the identical performances for 4 processes because of the zero  $-1/T_d$  (the lower the priority is, the higher the time delay to be compensated is, the higher the value of  $T_d$  is, and then the higher the  $O$  is).

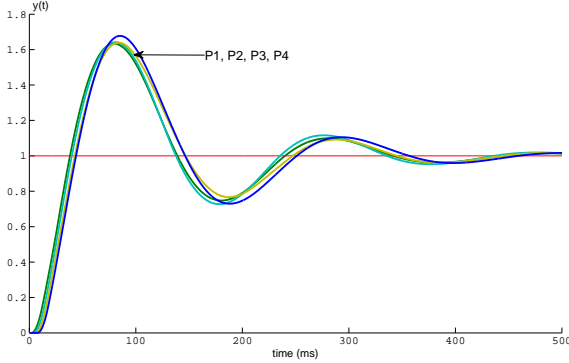


Figure 9. Relation QoC→QoS:  $y(t)$ .

P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>
169	198	185	233

Table 4. Relation QoC→QoS:  $\Delta J/J_0\%$ .

## 4 Relation QoC→QoS

This relation is implemented by means of the CANlike-hyb protocol where the part “dynamic priority” provides the property “Application aware network”. The controller is fixed (controller defined in Sect. 2.1)

### 4.1 Essential problem

We have to specify the value of the level 2 of the ID field of the  $f_{sc}$  and  $f_{ca}$  frames knowing that the controller computes a dynamic priority (using Equa. 4) each time it receives a  $f_{sc}$  frame.

The first proposal we did [5], [13] and which seems natural is that, during each sampling period, the sensor task uses the last dynamic priority it knows and the controller task uses the dynamic priority it has just computed.

So, by considering the sampling period  $k$  (period starting at the instant  $t_k$  in the sensor):

- Level 2 of the ID field of the  $f_{sc}$  frame: the value is the dynamic priority computed in the previous sampling period ( $k - 1$ ); call it  $P_{k-1}$ .
- Level 2 of the ID field of the  $f_{ca}$  frame: the value is the dynamic priority which has been computed (by the controller) in this sampling period  $k$ ; call it  $P_k$ .

However, with this scheme, the sequence (reception of the  $f_{sc}$  frame by the controller - sending of the  $f_{ca}$  frame by the controller) can be interrupted (because when the controller try to send the  $f_{ca}$  frame, there can be a frame of another process which has a bigger dynamic priority). So, in order to this sequence be not interrupted (*i.e.* atomic action), we put the value  $P_{max}$  in the level 2 of the ID field of the  $f_{ca}$  frame and we put the value  $P_k$  in the data field of this frame. In this way, the sensor task, by considering that it knows the static priority of the  $f_{ca}$  flow will read the concerned  $f_{ca}$  frame and will learn the dynamic priority  $P_k$  that it will use in the next sampling period ( $k + 1$ ).

## 4.2 Results

The QoC is represented on Tab. 4 ( $\Delta J/J_0\%$ ) and on Fig. 9 (time responses). We can see a balance in QoC between the 4 processes by comparing with the results done by the use of the static priorities on Tab. 2 and Fig. 7. This balance results from the predominant role of the part “dynamic priority” with respect to the part “static priority” (as all the processes have the same type of variable urgencies because they have the same transfer function, that explains the balanced aspect).

## 5 Relation QoS⇒QoC

### 5.1 Principle of the implementation

The objective is to combine the frame scheduling scheme based on the hybrid priority (*i.e.* the relation QoC→QoS) and the compensation method for time delays (*i.e.* the relation QoS→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 sampling 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 the principle of which is represented on Fig. 10.

At the instant  $t_k$ , the sensor task generates the  $f_{sc}$  frame which includes the dynamic priority  $P_{k-1}$  in the ID field *i.e.* the sensor task uses the dynamic priority computed by the controller in the previous period (this expresses the implementation of the relation QoC→QoS). The data field of this  $f_{sc}$  frame includes the value of the instant  $t_k$  and the output sampled value  $y_k$  (these values represent the contribution of the sensor task to the relation QoS→QoC).

When the controller task receives this frame, it undertakes the following actions (we suppose that these actions are instantaneous *i.e.* computational time is neglected):

- With the value of  $t_k$ , it deduces the value of the time delay  $\tau_{sc}$  (by comparison to its local clock; we suppose that the clocks are well synchronized); furthermore, as the level 2 of the ID field of the  $f_{ca}$  frame includes  $P_{max}$ , the  $f_{ca}$  frame is sent immediately, then  $\tau_{ca} = D_{ca}$ ; and as the value of  $\tau_{ZOH}$  is known ( $\tau_{ZOH} = h/2$ ), the controller has the knowledge of the loop time delay  $\tau$  in this sampling period and can then compute the new parameters  $K$  and  $T_d$  (Equa. 6). This compensation expresses the implementation of the relation QoS→QoC.
- Then, with the value  $y_k$  and the new parameters  $K$  and  $T_d$ , the controller computes the value of the control signal  $u_k$  (Equa. 8). With this value  $u_k$ , the controller computes the dynamic priority  $P_k$  (Equa. 4)



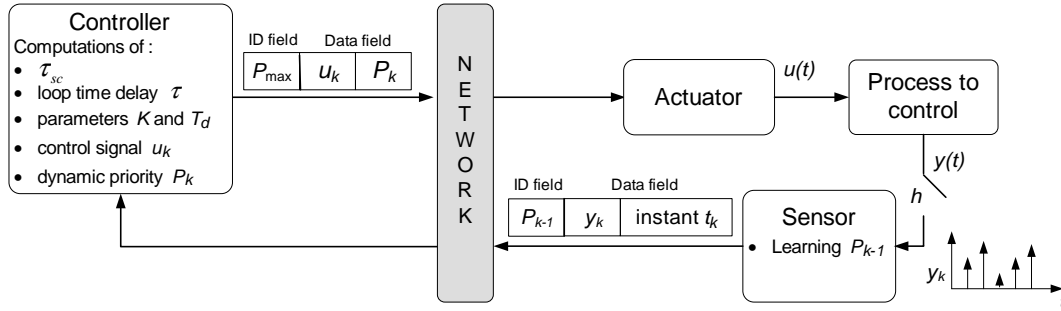


Figure 10. Principle of the implementation of the relation  $QoS=QoC$ .

which will be used in the next sampling period by the sensor task (this characterizes the setting up of the relation  $QoC \rightarrow QoS$  for the next sampling period).

## 5.2 Results

The  $QoC$  is represented on Tab. 5 ( $\Delta J/J_0\%$ ) and on Fig. 11 (time responses).

Comparing with the results relative to the relation  $QoC \rightarrow QoS$  (Tab. 4; Fig. 9): we still maintain the balanced performances for processes and by adding the relation  $QoS \rightarrow QoC$  (*i.e.* delay compensation), the relation  $QoS=QoC$  improves the  $QoC$  (we see the smaller overshoots on Fig. 11).

$P_1$	$P_2$	$P_3$	$P_4$
99.3	76.6	107	112

Table 5. Relation  $QoS=QoC$ :  $\Delta J/J_0\%$ .

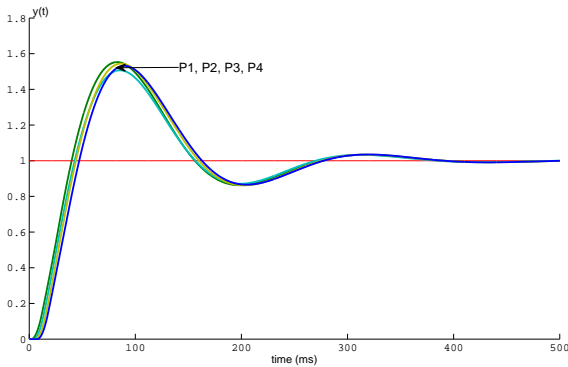


Figure 11. Relation  $QoS=QoC$ :  $y(t)$ .

## 6 Conclusion

We give on Fig. 12 a graphic representation of the  $QoC$  ( $\Delta J/J_0\%$ ) which summarizes the study done in this paper (on Fig. 12: the dotted lines represent the maximum gap in DCF-WiFi with respect to the mean value on 20 simulations; “Adaptive-Controller” is the controller where we use the dominant pole method and “Fixed Controller” is the controller in the case where we do not use the compensation method for time delays).

A first point to mention is the interest of the implementation of priorities in the protocols of the type CSMA (we

see the improvement of the performance which can be obtained with respect to DCF-WiFi for example).

The main point that we want to emphasize here is the interest of the relation  $QoS=QoC$  for an one-hop WNCS *i.e.* the joint action of the hybrid priorities and the delay compensation: by the dynamic part of the hybrid priorities (*i.e.* we capture the frame transmission urgency), we introduce the balance aspect with respect to the case of 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.

## Acknowledgments

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] K. J. Åström and B. Wittenmark. *Computer-controlled systems : Theory and design*. Prentice Hall, 3rd edition, 1997.
- [2] A. Cervin. *Integrated Control and Real-Time Scheduling*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, Sweden, April 2003.
- [3] B. Crow, W. I., K. L.G., and S. P.T. IEEE 802.11 Wireless Local Area Networks. *IEEE Communications Magazine*, 35:116–126, September 1997.
- [4] B. GmbH. Can specification 2.0 (part a), 1991.
- [5] G. Juanole and G. Mouney. Networked Control Systems: Definition and Analysis of a Hybrid Priority Scheme for the Message Scheduling. In *13<sup>th</sup> IEEE conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2007)*, Daegu, Korea, August 2007.
- [6] G. Juanole, G. Mouney, C. Calmettes, and M. Peca. Fundamental Considerations for Implementing Control Systems on a CAN Network. In *6<sup>th</sup> IFAC International conference on Fieldbus Systems and their Applications (FET 2005)*, Puebla, Mexico, November 2005.
- [7] G. Juanole, H. Nguyen Xuan, and G. Mouney. Specification of Collision-Free CSMA MAC Protocols for Wireless LANs. In *Submitted to 17<sup>th</sup> IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2012)*, Kraków, Poland, September 2012.

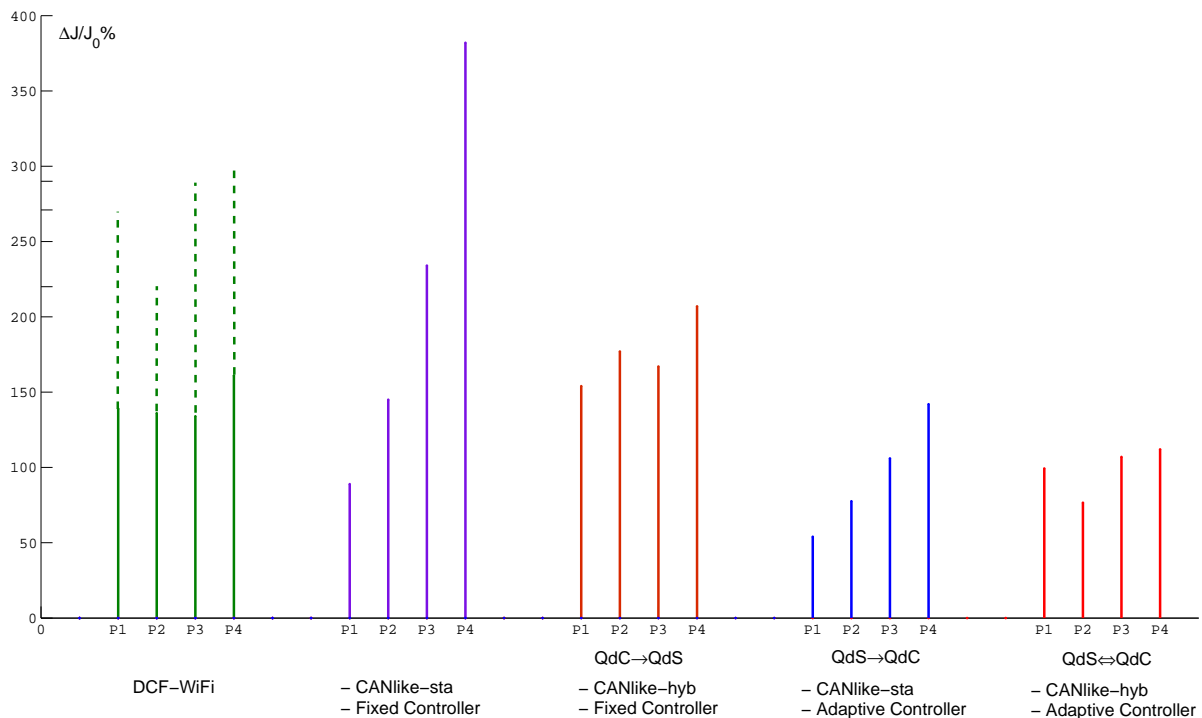


Figure 12. Graphic representation of the QoC ( $\Delta J/J_0\%$ ).

- [8] F.-L. Lian, J. Moyne, and D. Tilbury. Network Design Consideration for Distributed Control Systems. *IEEE Transactions on Control Systems Technology*, 10(2):297–307, March 2002.
- [9] P. Marti. *Analysis and design of Real-Time Control Systems with Flexible Timing Constraints*. PhD thesis, Departament d’Enginyeria de Sistemes, Automàtica i Informàtica Industrial, Universitat Politècnica de Catalunya, Barcelona, Spain, June 2002.
- [10] H. Nguyen Xuan. Networked Control Systems Quality of Service aware Quality of Control adaptation. In *Proc. 16<sup>th</sup> IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2011)*, Toulouse, France, September 2011.
- [11] H. Nguyen Xuan. *Réseaux de Communication et Applications de Contrôle-Commande*. PhD thesis, Institut National des Sciences Appliquées (INSA) de Toulouse, Toulouse, France, December 2011.
- [12] H. Nguyen Xuan and G. Juanole. Networked Control Systems (NCSs): on the interest of a Co-design approach based on interplays between Quality of Control and Quality of Service. In *Submitted to 9<sup>th</sup> IEEE International Workshop on Factory Communication Systems (WFCS 2012)*, Lemgo/Detmold, Germany, May 2012.
- [13] 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. 8<sup>th</sup> IEEE International Workshop on Factory Communication Systems (WFCS 2010)*, Nancy, France, May 2010.
- [14] M. Ohlin, D. Henriksson, and A. Cervin. *TrueTime 1.5 – Reference Manual*. Lund Institute of Technology, Sweden, January 2007.
- [15] N. Pereira, B. Andersson, and E. Tovar. WiDom: A Dominance Protocol for Wireless Medium Access. *IEEE Transactions on Industrial Informatics*, 3(2):120–130, May 2007.
- [16] M. S.Branicky, S. M.Phillips, and W. Zhang. Scheduling and Feedback Co-Design for Networked Control Systems. In *41st IEEE Conference on Decision and Control, Las Vegas, Nevada USA*, Dec 2002.
- [17] M. S.Branicky, V.Liberatore, and S. M.Phillips. Networked Control Systems Co-Simulation for Co-Design. In *American control conference, Denver, USA*, volume 4, pages 3341–3346, June 2003.
- [18] Y.-Q. Song. Networked Control Systems: From Independent Designs of the Network QoS and the Control to the Co-design. In G. Juanole and S. Ho Hong, editors, *8<sup>th</sup> IFAC international conference on Fieldbuses and networks in industrial and embedded systems (FeT 2009)*, Hanyang University, Republic of Korea, May 2009.
- [19] M. Velasco, P. Marti, R. Castane, J. Guardia, and J. Fuentès. A CAN Application Profile for Control Optimization in Networked Embedded Systems. In *IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on*, pages 4638–4643, November 2006.
- [20] T. You, C.-H. Yeh, and H. Hassanein. A New Class of Collision Prevention MAC Protocols for Wireless Ad Hoc Networks. In *IEEE International Conference on Communications (ICC’03)*, pages 1135–1140, 2003.