



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

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. 2012. hal-00668746v3

HAL Id: hal-00668746

<https://hal.science/hal-00668746v3>

Submitted on 5 Jul 2012 (v3), 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 Co-design approach in Wireless Networked Control Systems

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

We consider the implementation of process control applications on a Wireless LAN with a collision-free CSMA MAC protocol called CANlike (adaptation of the MAC protocol of the CAN bus to the wireless context). The goal of this paper is to show a co-design approach which is based on relations 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 Wireless LAN. We present, first, the relation $QoS \rightarrow QoC$ on the basis of a compensation method for time delays called dominant pole method, and second, the relation $QoC \rightarrow QoS$ based on the hybrid priorities for the frame scheduling. Finally, we show the interest of the joint action of the delay compensation and the role of the hybrid priority (i.e. relation $QoS \rightleftharpoons QoC$) in order to have a more efficient Wireless Networked Control System 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 [21], [23], [2].

Several works [11], [20], [4] have shown the interest of a co-design between Automatic Control and Computer Science (Task scheduling).

Some other works have shown the co-design interest by considering Automatic Control and Communication Network [13], [24], [7], [14]. But the aspect Communication Network mainly concerns the wired networks.

However today, we see more and more the use of wireless networks in the development of NCSs, then it is very important to work on the co-design of Wireless NCSs.

In the Wireless LAN (WLAN) context, the MAC protocol plays a basic role on the scheduling of the frames. Two main types of Wireless MAC protocol are CSMA (Carrier Sense Multiple Access) and TDMA (Time Division Multiple Access) types. CSMA is more flexible than

TDMA with respect to changes in a network i.e. adding or withdrawing a node or an application. Furthermore, CSMA is also more suitable for sporadic traffic. In this work, we consider MAC protocols of the type CSMA for single-channel and single-hop WLANs.

However, the big drawback of the CSMA protocols is the occurrence of collisions (an example is IEEE 802.11 DCF [5]) and then we cannot get QoS guarantees (for example, time delay upper bound) for real-time applications. Obtaining QoS guarantees with CSMA protocols is possible by associating priorities to the frames (a rich bibliography is presented in [26], [25]). The priorities can be defined in CSMA protocols by different sizes of the inter-frame space and/or contention window [6], [10], but these approaches cannot fully avoid the collision. One collision-free approach is to use the BlackBurst technique [22], [18], [15]. The idea is to let contending nodes send jamming signals of length according to the priority, the winner is the one that has the longest signal (i.e. the highest priority). Another collision-free approach is to adapt the MAC protocol of the wired CAN bus [3] to the wireless context [27], [19]. In this study, we consider the last approach and use a collision-free CSMA MAC protocol called CANlike that we have defined in [16].

The goal of this paper is to show the interest of a co-design of the QoS provided by the frame scheduling of the WLAN, and the QoC provided by the controller (i.e. relation $QoS \rightleftharpoons QoC$). We have then both the relation $QoS \rightarrow QoC$ (QoC is QoS driven i.e. *network performances aware dynamic QoC adaptation*) and the relation $QoC \rightarrow QoS$ (QoS is QoC driven i.e. *application performances aware dynamic QoS adaptation*).

We have already done such works in the context of the wired CAN bus [13]. In this paper, we want to present a study of the same type as in [13], but for the wireless context using CANlike. Firstly, we present the interest of the relation $QoS \rightarrow QoC$ where the controller is aware on-line of the loop time delay of a process control application and adapts its parameters by using a dominant pole method (compensation for time delays). Here we use CANlike with the static priorities for the frame scheduling. Secondly, we present the interest of the relation

QoC \rightarrow QoS based on the use of hybrid priorities for the frame scheduling. Here we use CANlike with the hybrid priorities and do not use any compensation method for time delays. Finally, we present the benefit from combining these two relations *i.e.* the relation QoS \Rightarrow QoC. We have thus a co-design on the basis of the joint action of the delay compensation and the hybrid priority.

This study is done by using the simulator TrueTime [17], a tool box based on Matlab/Simulink which allows to simulate real-time distributed control systems.

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

2 Context of the study

2.1 Process control application which is considered

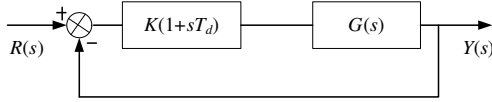


Figure 1. Continuous control system.

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 use 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 $R(s) = 1/s$ and the output is $Y(s)$.

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

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

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

where ω_n is the natural pulsation and ζ 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).

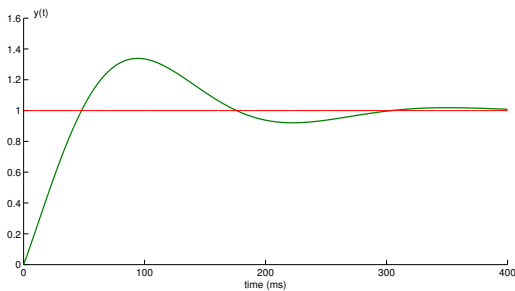
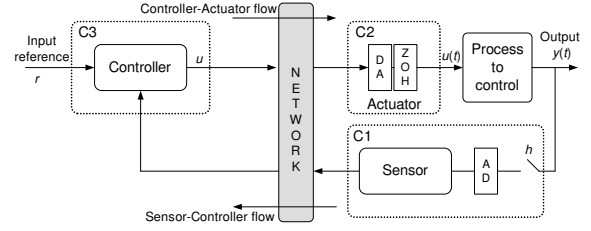


Figure 2. Time response $y(t)$.



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

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

2.2 Implementation of the process control application through a network

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

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

We have three computers C1, C2, C3 in which we have the sensor task, the controller task and the actuator task respectively. The sensor task samples the output signal and sends it to the controller task. The controller task computes the control signal u and sends it to the actuator task. The actuator task receives the control signal u and applies it to the controlled process.

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 including the Sensor-Controller flow (noted the f_{sc} flow) and the Controller-Actuator flow (noted the f_{ca} flow).

The running of the control system is characterized by several time delays: computational time delays in the running of the sensor, controller and actuator tasks in their computers; communication time delays in the transmission of the f_{sc} frames (noted τ_{sc}) and the f_{ca} frames (noted τ_{ca}). Note that the ZOH behavior can be seen as a pure communication time delay $\tau_{ZOH} = h/2$ [9].

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:*

$$\tau = \tau_{sc} + \tau_{ca} + \tau_{ZOH} \quad (2)$$

2.3 Model of the implementation of a process control application through a network

This model can be represented by the continuous model given on Fig. 4 where the time delays ($\tau_{ca} + \tau_{ZOH}$) and τ_{sc} are represented respectively by the transfer functions $e^{-(\tau_{ca} + \tau_{ZOH})s}$ and $e^{-\tau_{sc}s}$.

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

$$F(s) = \frac{K(1 + T_d s)e^{-(\tau_{ca} + \tau_{ZOH})s}G(s)}{1 + K(1 + T_d s)e^{-\tau_{sc}s}G(s)} \quad (3)$$

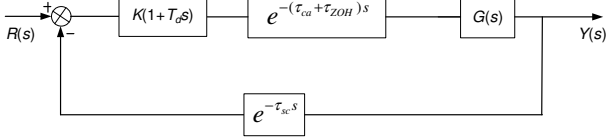


Figure 4. Control system with time delays.

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 finally get the transfer function as follows:

$$F(s) = \frac{1000Ka(1 + T_d s)(1 + s/a)(1 - s/b)}{f_3(s)(1 + s/b)} \quad (4)$$

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

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 Dominant pole method

This method has been represented in [12], [13]. We will present here some main points.

Main ideas

Consider that the controller has the knowledge of the loop time delay τ (Equa. 2). 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. 4) 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 smaller than that of the dominant poles.

Note that as the equation 4 has three zeros, we also have to evaluate the influence of these zeros on the overshoot of the time response $y(t)$.

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

Consider the polynomial $f_3(s)$ in the denominator of Equa. 4, 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 $p_{1,2} = R \pm jI$:

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

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. 4 by this one found in Equa. 6 and taking into account the relation $R^2 + I^2 = \omega_n^2$, we have now the transfer function:

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

Remark

If the network is not overloaded ($\tau_{sc} + \tau_{ca} < h$), we have shown in [13] that, 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 $z_1 = -1/T_d$ must be considered. In these conditions, the transfer function in Equ. 7 can be rewritten as follows:

$$F(s) = \frac{\omega_n^2(1 + T_d s)}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (8)$$

We see that we have the same form as the transfer function of the system without delay (Equa. 1) but now the value of T_d is changed. *We have already shown in [13] that, when the delay increases, the zero z_1 moves closer to the origin and it increases the overshoot.*

2.5 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:

$$\begin{cases} P_k = Ke_k \\ D_k = \frac{T_d}{Nh + T_d} D_{k-1} + \frac{NKT_d}{Nh + T_d} (e_k - e_{k-1}) \\ u_k = P_k + D_k \end{cases} \quad (9)$$

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

2.6 Criteria for 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 ($y(t)$).

2.7 On the priorities associated to the frames

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

However, we can also have hybrid priorities *i.e.* priorities with two priority levels. One level represents the flow priority which is a static priority and the other represents the frame transmission urgency. 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) in which the level 1 represents the flow priority and the level 2 represents the urgency priority [7].

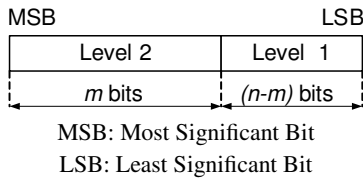


Figure 5. ID field structure (hybrid priority).

In the context of the competition based on these hybrid priorities, the competition 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 both two cases (static priority and hybrid priority) for the scheduling of the frames of the flows f_{sc} and f_{ca} of process control applications.

Case of the static priorities

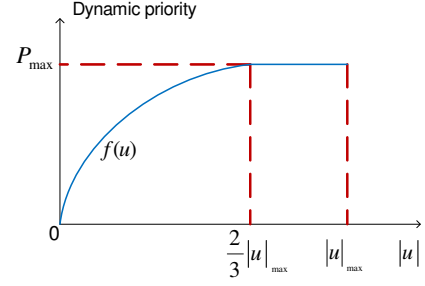
Call P_{sc} and P_{ca} the priorities of the f_{sc} and f_{ca} flow, respectively, of a process control application. It has been shown in a previous work [8] that we must have $P_{ca} > P_{sc}$. In the context of a shared network, this condition allows the controller to act faster after the reception 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. While with respect to 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 is the control signal u [7], the value of which is representative of the strength of the action which is requested to the actuator.

The control signal u is translated into a priority on the basis of an increasing function of $|u|$ (Fig. 6, Equa. 10) 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.



P_{\max} is the maximum dynamic priority
 u_{\max} is the maximum value of the control signal u

Figure 6. Dynamic priority (function of u).

$$f(u) = \begin{cases} P_{\max} \sqrt{\frac{|u|}{\frac{2}{3}|u|_{\max}}}, & 0 \leq |u| \leq \frac{2}{3}|u|_{\max} \\ P_{\max}, & |u| > \frac{2}{3}|u|_{\max} \end{cases} \quad (10)$$

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

Remark

We can have other possible choices of the variable to represent the transmission urgency, for example, the error $e = r - y$. We have done a comparison in terms of QoC between the use of u and e for this particular considered application. In this case, u gives better results than e . This is why u is used. But a deeper study has to be done considering different transfer functions and controllers.

2.8 CANlike protocol

This protocol has been totally explained and presented in [16]. Here we only present a summary.

Main ideas

As introduced, CANlike is an adaptation of the MAC protocol of the wired bus CAN to the wireless network.

In the wired bus CAN, MAC entities can send bits and listen to the channel simultaneously. Each MAC entity has a unique ID (identifier) field placed at the beginning of the frame. The ID represents the priority and allows to do the channel access tournament. The tournament is done by a comparison bit by bit between the IDs of the frames trying to access the channel. In one bit-by-bit comparison, a bit 0 called a dominant bit overwrites a bit 1 which is a recessive bit. Because of the uniqueness of the ID, the MAC entity which has the highest priority will be the only one winner after the tournament and it will send its frame.

In the wireless context, the bus CAN protocol cannot be directly implemented with wireless transceivers since the transceivers cannot transmit and receive simultaneously in the same channel, so we consider the proposal, which has been done in [27], [19]: one slot time (duration) is provided for each ID bit, a dominant bit consists in the sending of a carrier wave during its duration while a recessive bit consists in the sensing/listening of the channel during its duration.

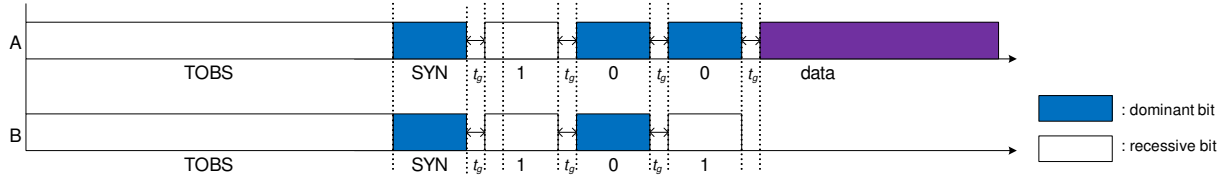


Figure 7. Example of a time diagram of the CANlike protocol.

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

- The MAC entity has a dominant bit: this entity sends a carrier wave and at the end of the sending, it wins by definition the tournament related to this bit and then, it continues the tournament on the next bit.
- The MAC entity has a recessive bit: either it senses a carrier wave on the channel, 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.

At the end of the tournament, there will be only one winner which will send its frame.

We want to note here three important points:

- A MAC entity needs, before to start a tournament, to observe that the channel is idle during some time (called *observation time* of the idle channel). The role is to avoid intrusions in a tournament in progress. This time must be higher than the maximum time when the channel is idle during the tournament.
- The start of a tournament requires the MAC entity to send a synchronization (SYN) bit (which is still a carrier wave). The role 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 SYN bit, but has not sent its SYN bit, does not participate in the tournament.
- Because of the distance among MAC entities and the asynchronism of SYN bit transmissions, a MAC entity may finish its SYN bit transmission before the end of the propagation of the SYN bit transmission of other MAC entities. We thus need to add a *guard time* at the end of the SYN bit and each ID bit in order to ensure that there is no overlap of bits of different rank. During the guard time, the MAC entity is blind (*i.e.* no sending, no sensing) and after this time, there is no residue of the previous bit tournament.

Parameters of the CANlike protocol

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

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 parameters of CANlike are as follows: l_b is the length of an ID bit, l_s is the length of the SYN bit, t_g is the length of the guard time and TOBS is the observation time of the idle channel. The values of these parameters are given on Tab. 1.

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 the value of TOBS:

$$\begin{aligned} \text{TOBS} &= (n + 1)(l_b + t_g) \\ &= (n + 1)(4\tau_{PT} + 2\tau_{TT} + \tau_{ST}) \end{aligned} \quad (11)$$

Finally, we obtain the tournament duration of a winner:

$$\begin{aligned} &\text{TOBS} + (l_s + t_g) + n(l_b + t_g) \\ &= 2(n + 1)(4\tau_{PT} + 2\tau_{TT} + \tau_{ST}) \end{aligned} \quad (12)$$

Remark

- The duration (slot time) of the SYN bit and the ID bit is considerably longer than this of a data bit (bit of a frame). The winner transmits the frame at the bit rate allowed by the radio transceiver.
- We have implemented this protocol in the simulator TrueTime.

Example of a time diagram of the CANlike protocol

We present a simple example of the tournament of this protocol on Fig. 7. We consider 2 nodes A and B which start the tournament at the same time and which have the ID fields of 3 bits (1; 0; 0) and (1; 0; 1) respectively, *i.e.* A has a higher priority than B. At the 1st ID bit, the two nodes, which have recessive bits, find the channel idle and continue the next bit. At the 2nd ID bit, the two nodes have dominant bits so they continue the next bit. At the last ID bit, A has a dominant bit so it is the winner by definition while B, which has a recessive bit, finds the channel busy and then it stops the tournament. A then sends its data.

2.9 Global system which will be analyzed

We will consider the implementation of 4 processes (P_1, P_2, P_3, P_4) on a WLAN which are identical to the one presented in Sect. 2.1. The sensor tasks of which are synchronous (*i.e.* same sampling instants). The data field of the f_{sc} and f_{ca} frames are of 16 bits. The controller, sensor and actuator tasks of the 4 processes are all in different computers. Then we have 12 computers connected to the network and we have 8 frame flows including 4 f_{sc} flows and 4 f_{ca} flows.

Note that the static priorities of the f_{cai} and f_{sci} flows of each process P_i ($i = 1, 2, 3, 4$) are such that $P_{cai} > P_{sci}$ (Sect. 2.7) and these 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. P_i is considered more important than P_j with $i < j$.

Concerning the WLAN, we consider two MAC protocols: IEEE 802.11 DCF and CANlike. *IEEE 802.11 DCF is considered as the base of our analysis. CANlike has the same structure of the frame of IEEE 802.11 DCF.* The bit rate in the Physical layer is 1 Mbits/s. Other characteristics of the two MAC protocols are as follows:

Protocol IEEE 802.11 DCF

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

Protocol CANlike

- We 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 bits for the level 2 and 4 bits for the level 1 (Fig. 5).
- From Tab. 1, we get $l_b = l_s = 20 \mu s$, $t_g = 5 \mu s$, $TOBS = 225 \mu s$ and the tournament duration of a winner (Equa. 12) is $450 \mu s$.
- The length of a frame is 480 bits, then, as the bit rate is 1 Mbits/s, the duration of a frame is $480 \mu s$.
- The total duration for a winner to send a f_{sc} frame or a f_{ca} frame is: tournament duration ($450 \mu s$) + frame duration ($480 \mu s$) = $930 \mu s$. So, as we have to transmit 8 frames during each period of $h = 10$ ms, this can be done because $8 \times 930 = 7440 \mu s < h$ and then the network is not overloaded.
- *Remark:* as the network is not overloaded, the conditions for insignificances of the poles p_3, p_4 and neglected roles of the zeros z_2, z_3 in the remark in Sect. 2.4 are validated. The proposed dominant pole method can be used without problem.

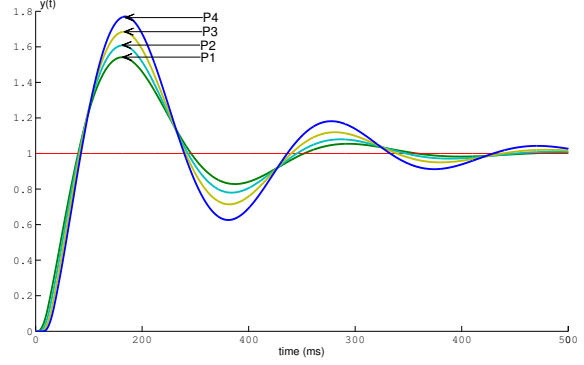


Figure 8. CANlike-sta: time responses $y(t)$.

	P_1	P_2	P_3	P_4
IEEE 802.11 DCF:				
-mean value	139	136	134	161
-max gap	132	84	156	140
CANlike-sta	88.9	145	234	382

Table 2. IEEE 802.11 DCF and CANlike-sta: QoS evaluation ($\Delta J/J_0\%$).

Remark about the implementation in the simulator

IEEE 802.11 DCF had already implemented in the simulator TrueTime, we have only implemented CANlike.

2.10 Reference results

We have done the evaluation of QoS ($\Delta J/J_0\%$) of the 4 processes (P_1, P_2, P_3, P_4) by considering the protocols IEEE 802.11 DCF and CANlike-sta. The results which are on Tab. 2 are considered as the reference to compare.

For IEEE 802.11 DCF, as we have collisions and the collision resolution is done by using a random time (back-off mechanism), each run simulation gives a different result. Thus, we have done 20 simulations and we give the mean value and the maximum gap existing between the results of one simulation and the mean value. We can see that the results of the different processes are pretty balanced. The balance between the mean values is normal because the processes are not differentiated. If we do more simulations, we will have the more balance. But the very high possibility of maximum gaps and the non-deterministic characteristic indicate that IEEE 802.11 DCF cannot be used for real-time applications.

For CANlike-sta, we see that obviously the priority discriminates the processes, P_i is better than P_j with $i < j$ (Tab. 2 and Fig. 8) 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 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 because τ_{ca} is unknown (τ_{sc} can be known). However, with static priorities and taking into account that the network is not overloaded (Sect. 2.9), the delay ($\tau_{sc} + \tau_{ca}$), for each process, is the same whatever the period may be. Thus the controller can 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} and the reception instant of the corresponding f_{ca} frame by the actuator task (we suppose that the propagation time is neglected and the sensor task knows the address of the f_{ca} frame, so the instant when the sensor task reads the f_{ca} frame is exactly the reception instant of the f_{ca} frame by the actuator task).

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

In brief, concerning the controller task:

- It gets ($\tau_{sc} + \tau_{ca}$) from the f_{sc} frame, and then computes the loop delay $\tau = \tau_{sc} + \tau_{ca} + \tau_{ZOH}$ and computes 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. 9) 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 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 $\tau_0 = \tau_{ZOH}$.

Important remark

In IEEE 802.11 DCF, the time ($\tau_{sc} + \tau_{ca}$) is variable and unknown in each sampling period because of the using of a random time (backoff mechanism) for collision resolutions. Thus this relation $QoS \rightarrow QoC$ cannot be implemented with the IEEE 802.11 DCF protocol.

3.2 Results

The QoC ($\Delta J/J_0\%$) and time responses for 4 processes are represented respectively on Tab. 3 and Fig. 9.

We see that, this relation $QoS \rightarrow QoC$ improves the results compared with the results on Tab. 2 and on Fig. 8. This improvement is normal because here we have compensated the time delays.

Note that although we have compensated the time delays, we still do not have identical performances for the 4 processes (we see different overshoots on Fig. 9) due to the effect of the zero $-1/T_d$ (cf. Remark in Sect. 2.4). The lower the priority is, the higher the time delay to be compensated is, the higher the value of T_d is, the zero gets closer to the origin and then the higher the value of O is).

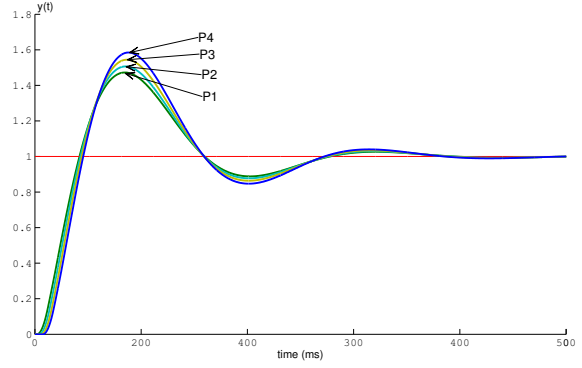


Figure 9. Relation $QoS \rightarrow QoC$: $y(t)$.

P ₁	P ₂	P ₃	P ₄
54.0	77.6	106	142

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

4 Relation QoC \rightarrow 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 and we do not use the delay compensation method).

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. Reminding that the controller task computes a dynamic priority (using Equa. 10) each time it receives a f_{sc} frame.

The first proposal we did [7], [14] and which seems natural is that, during each sampling period k (i.e. period starting at the sampling instant t_k in the sensor task), the sensor task uses the last dynamic priority computed by the controller in the previous sampling period $k-1$ (call it P_{k-1}) and the controller task uses the dynamic priority it has just computed (call it P_k).

Thus the value of the level 2 of the ID field of the f_{sc} frame is P_{k-1} while this one of the f_{ca} frame is 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 tries to send the f_{ca} frame, there can be a frame of another process which has a higher 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 (i.e. the controller task uses the maximum dynamic priority) while the value P_k which has just been computed is put in the data field of this frame in order to send this P_k to the sensor task. In this way, the controller task can send its frame immediately after the reception of the f_{sc} frame (computational times are neglected). In the next sampling period ($k+1$), the sensor task will learn P_k after having read the concerned f_{ca} frame and will put it in the level 2 of the ID field of its frame.

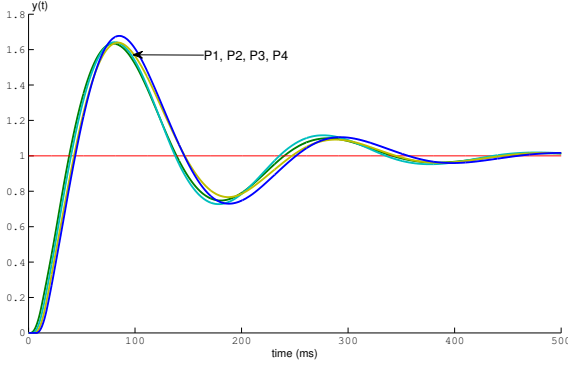


Figure 10. Relation QoC \rightarrow QoS: $y(t)$.

P ₁	P ₂	P ₃	P ₄
169	198	185	233

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

4.2 Results

The QoC is represented on Tab. 4 ($\Delta J/J_0\%$) and on Fig. 10 (time responses).

We can see a QoC balance between the 4 processes by comparing with the results done by the use of the static priorities on Tab. 2 and Fig. 8. This balance results from the predominant role of the part “dynamic priority” in comparison with 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 \rightleftharpoons QoC

5.1 Principle of the implementation

The objective is to combine the frame scheduling based on the hybrid priority (*i.e.* the relation QoC \rightarrow QoS) and the compensation method for time delays (*i.e.* the relation QoS \rightarrow 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 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. 12.

At the instant t_k , the sensor task generates a 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 \rightarrow 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 \rightarrow QoC).

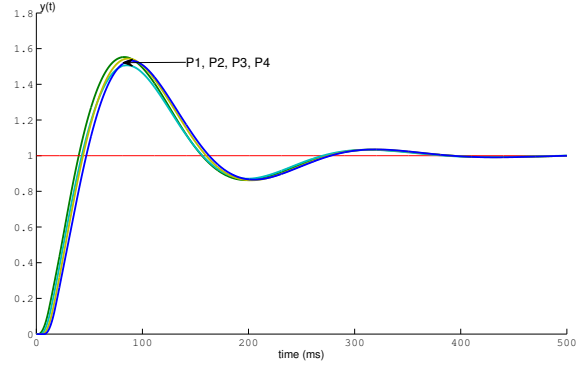


Figure 11. Relation QoS \rightleftharpoons QoC: $y(t)$.

P ₁	P ₂	P ₃	P ₄
99.3	76.6	107	112

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

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 τ_{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 τ_{ca} = frame duration (480 μ s); 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 (Equa. 6). This compensation expresses the implementation of the relation QoS \rightarrow 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. 9). With this value u_k , it computes the dynamic priority P_k (Equa. 10) 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. 10): we still maintain the balanced performances for processes and by adding the relation QoS \rightarrow QoC (*i.e.* delay compensation), the relation QoS \rightleftharpoons QoC improves the QoC (we see the smaller overshoots on Fig. 11).

6 Conclusion

We introduce on Fig. 13 a graphic representation of the QoC ($\Delta J/J_0\%$) which summarizes the study done in this paper (on Fig. 13: the dotted lines represent the maximum gap in IEEE 802.11 DCF in comparison with the

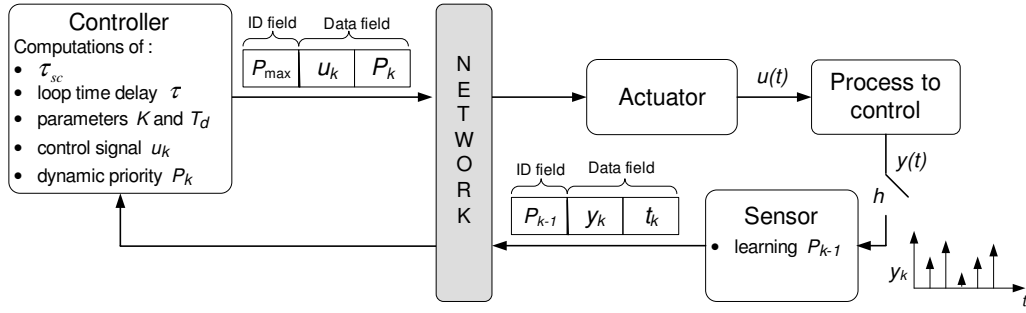


Figure 12. Principle of the implementation of the relation $QoS \Rightarrow QoC$.

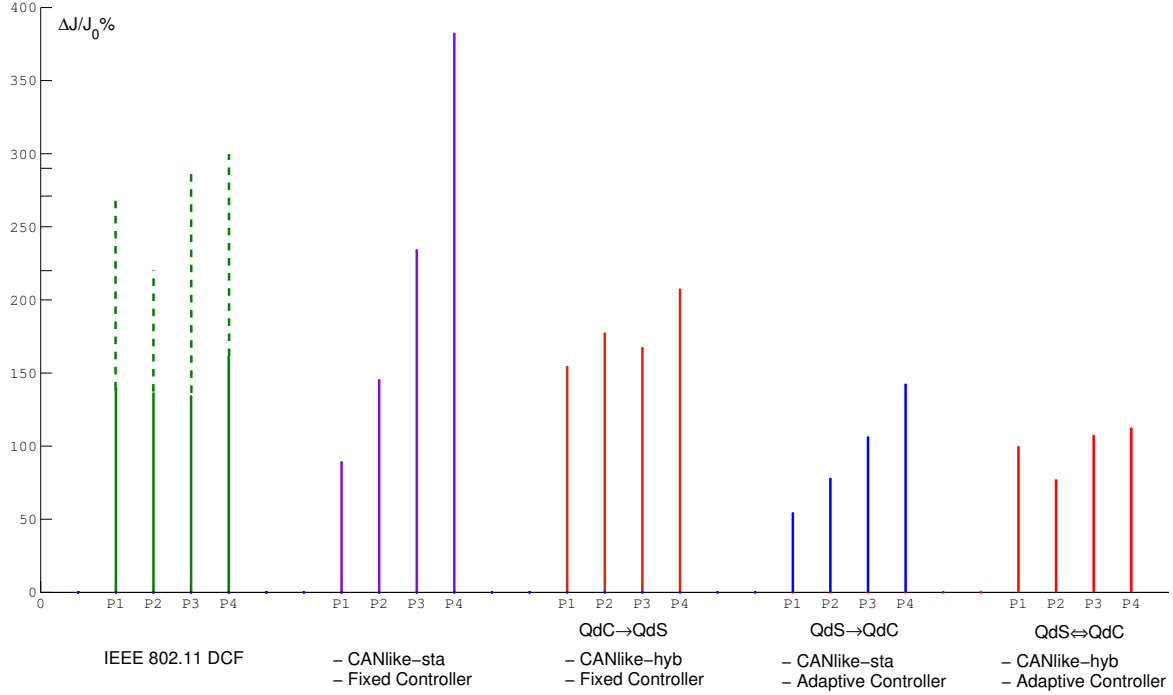


Figure 13. Graphic representation of the QoC ($\Delta J / J_0 \%$).

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 delay compensation method).

The first point to mention is the interest of the implementation of priorities in the CSMA protocols. We see the improvement of the performance obtained by the priority-based CSMA MAC protocols (CANlike) in comparison with IEEE 802.11 DCF for example. This study shows that IEEE 802.11 DCF in which collisions can occur cannot get QoS guarantees and then cannot be used for real-time applications.

The main point that we want to emphasize here is the interest of the relation $QoS \Rightarrow QoC$ which is the combination of the relation $QoS \rightarrow QoC$ and the relation $QoC \rightarrow QoS$ *i.e.* the joint action of the delay compensation and the hybrid priorities:

- By the delay compensation, we improve the QoC for all process control applications compared with the case we do not use the delay compensation.
- By the hybrid priorities (role of the part “dynamic

priority”), we introduce the QoC balance for different process control applications compared with the case of the static priorities.

- By the joint action, the relation $QoS \Rightarrow QoC$ allows to improve QoC while maintaining the balanced aspect. And then we can consider the possibility to implement more applications.

The further work should be the following points:

- From the automatic control point of view: to study stability conditions when the on-line control law parameters change from sampling period to sampling period; to consider other delay compensation methods, other types of controller (PID for example), other transfer functions.
- From the communication network point of view: to evaluate the difference of the performances between CAN and CANlike (influence of the tournament duration in CANlike); to specify collision-free CSMA MAC protocols for multi-hop communication.

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] C. Aubrun, D. Simon, and Y.-Q. Song. *Co-design Approaches for Dependable Networked Control Systems*. ISTE/John Wiley, 2010.
- [3] G. Bosch. Can specification 2.0, 1991. [online]. Available: www.can-cia.de/fileadmin/cia/specifications/can20a.pdf.
- [4] A. Cervin. *Integrated Control and Real-Time Scheduling*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, Sweden, April 2003.
- [5] 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.
- [6] D.-J. Deng and R.-S. Chang. A priority scheme for ieee 802.11 dcf access method. *IEICE Trans Commun (Inst Electron Inf Commun Eng)*, E82-B(1):96–102, 1999.
- [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] 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.
- [9] 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.
- [10] S. Mangold, S. Choi, G. Hiertz, O. Klein, and B. Walke. Analysis of IEEE 802.11e for qos support in wireless lans. *IEEE Wireless Communications*, 35:40–50, December 2003.
- [11] 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, Univ Politècnica de Catalunya, Barcelona, Spain, June 2002.
- [12] 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.
- [13] H. Nguyen Xuan and G. Juanole. Design of Networked Control Systems (NCSs) on the basis of interplays between Quality of Control and Quality of Service. *Will be appeared in Proc. 7th IEEE International Symposium on Industrial Embedded Systems (SIES 2012)*. [Online]. Available: <http://hal.archives-ouvertes.fr/hal-00668491>.
- [14] 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.
- [15] H. Nguyen Xuan, G. Juanole, G. Mouney, and C. Calmettes. Wireless communication networks and process control applications: studying the influence of MAC protocols. In *Proc. IEEE Globecom 2010, Workshop SaCoNAS*, Miami, Florida, USA, Dec 2010.
- [16] H. Nguyen Xuan and G. Mouney. Spécification du protocole « CANlike » pour la couche MAC des réseaux locaux sans fil. In *École d'Été Temps Réel 2011*, pages 195–198, Brest, France, Sep 2011.
- [17] M. Ohlin, D. Henriksson, and A. Cervin. *TrueTime 1.5 – Reference Manual*. Lund Institute of Technology, Sweden, January 2007.
- [18] A. Pal, A. Dogan, and F. Özgüner. Mac layer protocols for real-time traffic in ad-hoc wireless networks. In *Proceedings of the 2002 International Conference on Parallel Processing (ICPP'02)*, pages 539–546, Washington, DC, USA, 2002. IEEE Computer Society.
- [19] 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.
- [20] 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.
- [21] 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)*, volume 4, pages 3341–3346, Denver, USA, June 2003.
- [22] J. Sobrinho and A. Krishnakumar. Real-time traffic over the ieee 802.11 medium access control layer. *Bell Labs Technical Journal*, 10(1):172–187, 1996.
- [23] 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.
- [24] M. Velasco, P. Marti, R. Castane, 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)*, pages 4638–4643, Paris, France, Nov 2006.
- [25] A. Willig. Recent and emerging topics in wireless industrial communications: A selection. *IEEE Transactions on Industrial Informatics*, 4(2):102–124, 2008.
- [26] A. Willig, K. Matheus, and A. Wolisz. Wireless technology in industrial networks. *Proceedings of the IEEE*, 93:1130–1151, June 2005.
- [27] T. You, C.-H. Yeh, and H. Hassanein. A New Class of Collision Prevention MAC Protocols for Wireless Ad Hoc Networks. In *Proc. IEEE International Conference on Communications (ICC'03)*, pages 1135–1140, Anchorage, Alaska, USA, 2003.