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

Xuan Hung Nguyen, Guy Juanole, Gérard Mouney

On interplays between the Quality of Service and the Quality of Control for a co-design approach of a Wireless Networked Control System (WNCS)

Xuan Hung Nguyen[1,2], Guy Juanole[1,2], Gérard Mouney[1,2]
[1]CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse, FRANCE
[2]Univ de Toulouse, UPS, LAAS, F-31400 Toulouse, FRANCE
xhnguyen@laas.fr, juanole@laas.fr, mouney@laas.fr

Abstract—**We consider, by means of a study based on a simulation, 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→QoC on the base of the delay compensation method called dominant pole method, and second, the implementation of the relation QoC→QoS on the base of hybrid priorities for the frame scheduling. Finally, we show the interest of the relation QoC⇌QoS 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.**

I. Introduction

The study and design of Networked Control Systems (NCSs) is a very important research area today because of its multidisciplinary aspect (Automatic Control, Computer Science, Communication Network). The current objective of NCS design today is to consider a co-design in order to have an efficient control system. These works concern mainly a co-design on the aspects Automatic Control and Task Scheduling or Communication Network (wired) [1], [2], [3], [4]. 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 an WNCS based on a WLAN where each node is in the transmission range of the other nodes (one hop communication). In this 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). We consider here a MAC protocol of the CSMA type (an example is the protocol IEEE802.11 DCF [5] still called DCF-WiFi that is a CSMA CA protocol). However, the big drawback of such protocols is that collisions can occur and 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 [6] 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 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 [6] and which is called CANlike protocol (it is based on an adaptation, defined in [7], of the CAN protocol, used in the CAN bus [8], 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 QoC⇌QoS) *i.e.* we have both relation QoC→QoS (QoS is QoC driven *i.e.* Application performances aware dynamic QoS adaptation) and relation QoC→QoS (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 [4], [9], [10]. 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 [11], 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→QoC; the section 4 presents the implementation of the relation QoC→QoS; the section 5 presents the implementation of the bidirectional relation QoC⇌QoS; the section 6 presents the conclusion.

II. Context of the study

A. Process control application

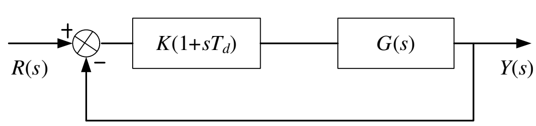


Fig. 1. Continuous control system.

The considered process control application is a continuous linear application, the model of which (using the Laplace transform) [12] is given on Fig. 1. The process to control has the transfer function $G(s) = \frac{1000}{s(s+1)}$. We have a Proportional Derivative (PD) controller in order to have a phase margin of 45° which imposes the following values: $K = 0.7291$; $T_d = 0.0297$ s. The input reference is a unity position step $R(s) = 1/s$. The output is $Y(s)$.

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

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

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).

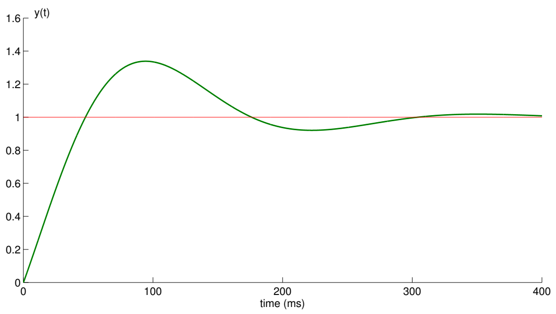
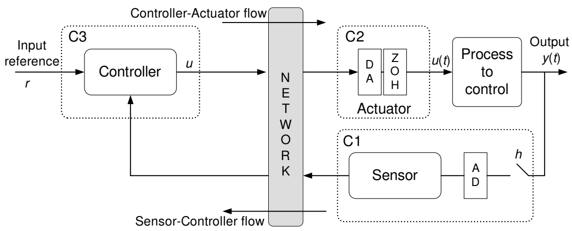


Fig. 2. Time response $y(t)$.

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)

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

We call h the sampling period (which is defined by considering the following formula $\omega_n h \in [0.1;\ 0.6]$ [12]) and t_k the sampling instants ($t_k = kh$ with $k = 0,1,2\ldots$). Here we take $h = 10$ ms.

The sensor task, in the computer C1, samples the output $y(t)$ and sends the output samples in the f_{sc} frames. The controller task, in the computer C3, works out, after the reception of the output samples, the command signal u from the difference between the input reference r and the received output samples and sends these command signal u in the f_{ca} frames. The actuator task, in the computer C2, uses the command signal u from the received f_{ca} frames and commands the process. The sensor task is time-triggered while the controller task and the actuator task are event-triggered. 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 τ_{sc}) and the f_{ca} frames (noted τ_{ca}). Note furthermore that the ZOH behavior can be seen as a pure delay $\tau_{ZOH} = h/2$. In this work, we only consider the time delays τ_{sc}, τ_{ca} and τ_{ZOH}. The computational delays are neglected. The time delay of the closed loop in each sampling period is $\tau = \tau_{sc} + \tau_{ca} + \tau_{ZOH}$.

B. *Model of the implementation on a network*

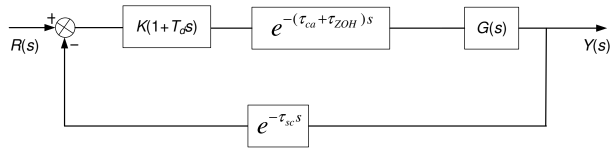


Fig. 4. Control system with time delays.

This model is represented on Fig. 4. The transfer function $F(s)$ is now:

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

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

$$f_3(s) = s^3+(1+a-1000KT_d)s^2+ (1000KT_d a+a-1000K)s+1000Ka \quad (4)$$

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 ($z_1 = -\frac{1}{T_d}$, $z_2 = -a = -\frac{2}{\tau}$, $z_3 = b = \frac{2}{\tau_{ca}+\tau_{ZOH}}$).

C. *CANlike protocol*

The ID (IDentifier) field at the beginning of each frame carries the priority which allows to implement bit by bit a tournament phase.

Main ideas: As, in a wireless context, the transceivers cannot transmit and receive at the same time, we consider the proposal, which has been done in [7], 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. A MAC entity which has a dominant bit wins by definition the tournament related to this bit and then continues the tournament on the next bit. A MAC entity which 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 ID field bits, can then send its frame.

Three other points are important [6]: *1)* A MAC entity needs, before to start a tournament, to observe that the channel is idle during some time (TOBS). 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. *2)* The starting of a tournament by a MAC entity requires that this MAC entity sends a synchronization bit (duration l_s) 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 (duration l_b). A MAC entity, which receives a synchronization bit, but has not sent itself a synchronization bit, does not participate in the tournament. *3)* Taking into account for the asynchronism and the distance between the MAC entities which participate to a tournament, a guard time (t_g) is added at the end of each bit (synchronization bit, ID field bit).

The parameters of the CANlike protocol [6], *i.e.* l_b, l_s, t_g and TOBS, given on Tab. I, are expressed by means of the parameters of the physical layer (propagation time τ_{PT}, sensing time τ_{ST}, turnaround time τ_{TT}) and the length of the ID field (n).

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 I. PARAMETERS OF THE CANLIKE PROTOCOL.

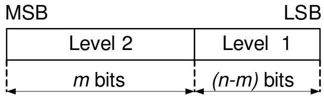
The duration linked to the implementation of a tournament is:

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

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

D. On the priorities associated to the frames

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, in the context of applications which have a transient behaviour which requires a good reactivity (like process control applications), it can be interesting to use hybrid priorities *i.e.* priorities with two levels: one level represents the flow priority (static priority); the other level represents the frame transmission urgency which can vary (dynamic priority). The working out of the dynamic priority, in the context of process control application has been done by means of a function of the control signal u (non linear square root function with a saturation [4]).
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 [4]. With this priority type, the competition for the frame scheduling is executed by comparing first the bits of the level 2 (urgency predominance). The highest urgency wins the competition. If the urgencies are identical, the level 1 (static priorities which have the uniqueness properties) resolves the competition.
We will consider here the two cases (static priorities, hybrid



MSB: Most Significant Bit, LSB: Least Significant Bit

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

priorities) for the scheduling of the frames of the flows of a process control application.

E. The global system which will be studied

We 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. II-A 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 $4f_{sc}$ flows and $4f_{ca}$ 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 $P_{cai} > P_{sci}$ 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, it is limited to the MAC layer and we consider for CANlike, the control part of the frame of the standard DCF-WiFi (480 bits) with a bit rate of 1 Mbits/s. We add to this part the ID field (8 bits with $l_b = l_s = 20$ μs and $t_g = 5$ μs. Then the duration linked to the tournament is 450 μs. The total duration for a winner to send a f_{sc} frame (noted D_{sc}) or a f_{ca} frame (noted D_{ca}) : $450+480+16 = 946$ μs. So, as we have to transmit 8 frames during each period of $h = 10$ ms, this can be done ($8 \times 946 = 7568$ $\mu s < h$) and then the network is not overloaded.

F. Reference result

We consider the protocol with static priorities (called CANlike-sta) and we give on the Fig. 6 the output $y(t)$ for the 4 process control application ($y(t)$ gives an idea of the QoC). Note that the performances are less good than when there is no the network. That is normal as we have now the delays in the network.

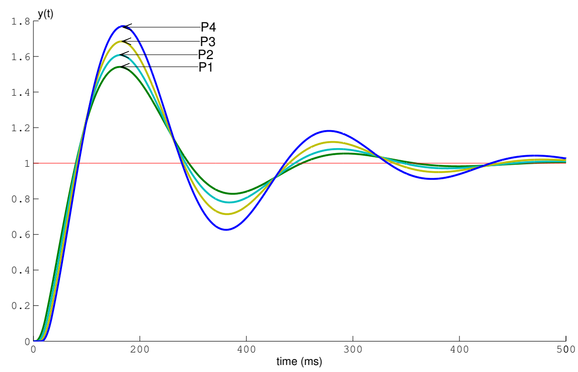


Fig. 6. CANlike-sta : time responses $y(t)$.

We see that obviously the priority discriminates the processes ,P_i is better than P_j with $i < j$ and so the higher the priority is, the less the time delay is, then the better the performance is. This result will be taken as reference for the analysis of the benefits given by the implementation of the relations QoS→QoC, QoC→QoS, and QoC⇌QoS.

III. Relation QoS→QoC

A. Main ideas

This relation is based on the use of the CANlike-sta protocol (using static priorities, the QoS is independant of the QoC) and on the actions of the controller which, at each period k, must make the compensation of the loop delay τ and compute the control signal. The controller learns, in each period, the delay $(\tau_{sc} + \tau_{ca})$ which is transmitted by the sensor in a f_{sc} frame (more precisely during the period k, the controller uses the delay $(\tau_{sc} + \tau_{ca})$ evaluated by the sensor during the period $k-1$); but with static priorities and, as the network is not overloaded, the delay $(\tau_{sc} + \tau_{ca})$ for each process is the same whatever the period may be. With the knowledge of $(\tau_{sc} + \tau_{ca})$, the controller computes, at first, the loop delay $(\tau_{sc} + \tau_{ca} + \tau_{ZOH})$ and then the new parameters K and Td using the dominant pole method (we have an adaptive controller). The goal of this method is to modify the parameters K and Td in such a way to maintain the same type of transient behaviour for the process control application, as before the implementation on the network, *i.e.* characterized by the two poles of the transfer function (1). Then, with the knowledge of the new parameters K and Td, the controller computes the control signal 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 $\tau_0 = \tau_{ZOH}$.

B. Dominant pole method

As the transfer function of the system implemented on the network (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 (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 (3) 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 are concerned by the polynomial $f_3(s)$ in the denominator of (3). This polynomial (poles p_1, p_2 and p_3 can be rewritten $s-p_1$, $s-p_2$, $s-p_3$ by considering the values

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

By identifying $f_3(s)$ in (4) with (5), we get the relations which allow to determine the value of p_3, K and T_d:

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

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

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

Remark As the network is not overloaded ($\tau_{sc} + \tau_{ca} < h$), we have shown [6] 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 (7) can be rewritten as follows :

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

We see that we have the same form as the transfer function of the system without delay (1) but now the value of T_d increases with the delay [6]. Then the zero z_1 moves closer to the origin what increases the overshoot.

C. Control law

At each period k the controller computes the control signal u_k (which includes the Proportional component P_k and the Derivate component D_k) by using the following formula:

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

where $e_k = r_k - y_k$, N is a constant ([12], page 307).

D. Results

We see Fig. 7 that, this relation QoS→QoC improves the results compared with the results on Fig. 6. This improvement is normal because here we have compensated the time delay. Note that although we have compensated the time delay, we still do not have identical performances for the 4 processes (we see differents overshoots due to the effect 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 value of the overshoot O is).

IV. Relation QoC→QoS

A. Main ideas

This relation is implemented by using a protocol CANlike with hybrid priorities (called CANlike-hyb). We have two main ideas : the first is that the controller (which is fixed *i.e.* it is the controller defined in section II-A) computes, at each period when it receives the f_{sc} frame, the dynamic priority which will be used in the next period by the sensor task(the value of the dynamic priority, which is transmitted in the data field of the

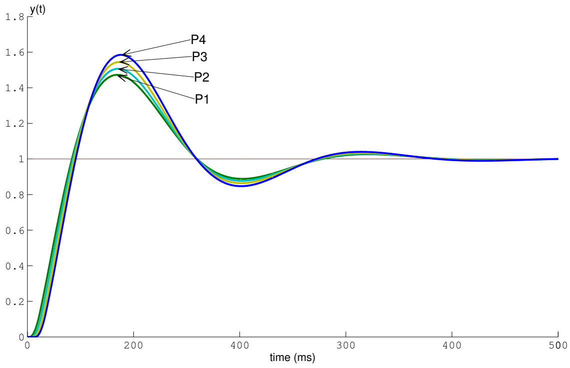


Fig. 7. Relation QoS→QoC: $y(t)$.

f_{ca} frame, provides the property "Application aware network"); the second, is that we want that the sequence of actions during a period (sending of the f_{sc} frame, sending of the f_{ca} frame) be atomic (*i.e.* no interruption) and then the controller uses the value Pmax for the dynamic priority of the f_{ca} frame).
Remark: At the initialization, the sensor has no information from the controller for the dynamic priority to use. So we propose that it uses also the value Pmax in the first f_{sc} frame.

B. Results

We give on the Fig. 8 the output $y(t)$ for the 4 process control applications. We can see, with respect to the results obtained with static priorities (Fig. 6) a balance in the performances of the 4 process control applications. 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).

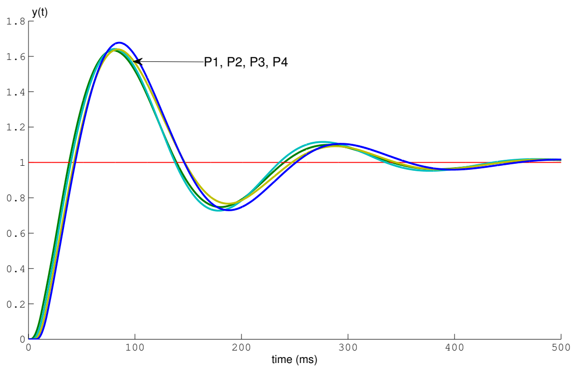


Fig. 8. Relation QoC→QoS: $y(t)$.

V. Relation QoC⇌QoS

A. Main ideas

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 $(\tau_{sc}+\tau_{ca})$, during the transient behaviour, 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. The principle is represented on Fig. 9.

At the instant t_k, the sensor task generates the f_{sc} frame with 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 the f_{sc} frame, it undertakes the computations indicated on Fig. 9 and sends the f_{ca} frame (as in the case of the relation QoC→QoS).

B. Results

We give on Fig. 10 the output $y(t)$ for the 4 process control applications. Comparing with the results relative to the relation QoC→QoS (Fig. 8): we still maintain the balanced performances for processes and by adding the relation QoS→QoC (*i.e.* delay compensation), the relation QoS⇌QoC improves the QoC (we see smaller overshoots)

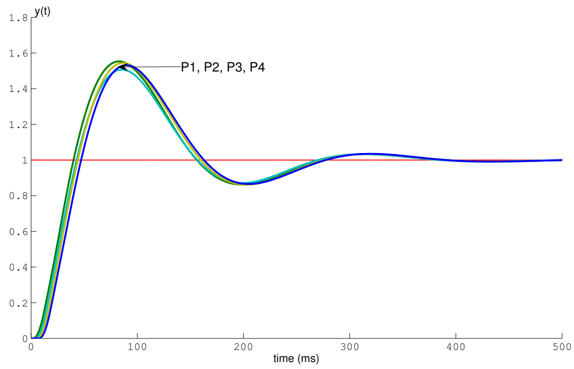


Fig. 10. Relation QoC⇌QoS: $y(t)$.

VI. Conclusion

In order to give a global view of the study, we evaluate the QoC with a cost function ITSE (Integral of Time-weighted Square Error) noted J with $J=\int_0^T t(r(t)-y(t))^2dt$ with $T>t_s$ (settling time of the process control application without the network) in order to cover the transient regime duration. We consider $T=500$ ms and we evaluate, at first, the value of J for the process control application without the network (we call J_0) this value; $J_0=9.4562.10^{-4}$). J_0 is considered as the reference value for the study of the implementation of the different relations with the network. The performance criteria, for the different relations with the network, is $\frac{J-J_0}{J_0}\%=\frac{\Delta J}{J_0}\%$. The higher the value $\frac{\Delta J}{J_0}\%$ is, the more degraded the QoC is.
Note still that we also evaluate the performances that we would get by using the protocol IEEE 802.11 DCF. We give on Fig. 11 a graphic representation of the QoC ($\Delta J/J_0\%$) which summarizes the study done in this paper (the dotted lines represent the maximum gap in DCF-WiFi with respect to the mean value on 20 simulations).
The first point to mention is the interest of priorities in the

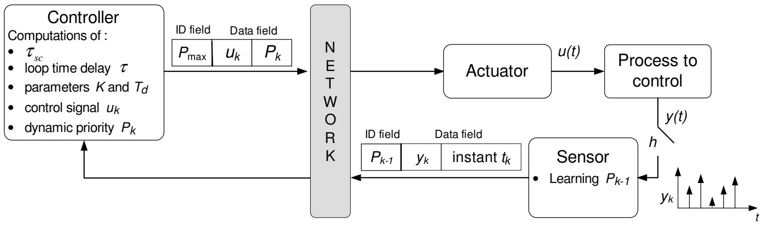


Fig. 9. Principle of the implementation of the relation QoC⇌QoS.

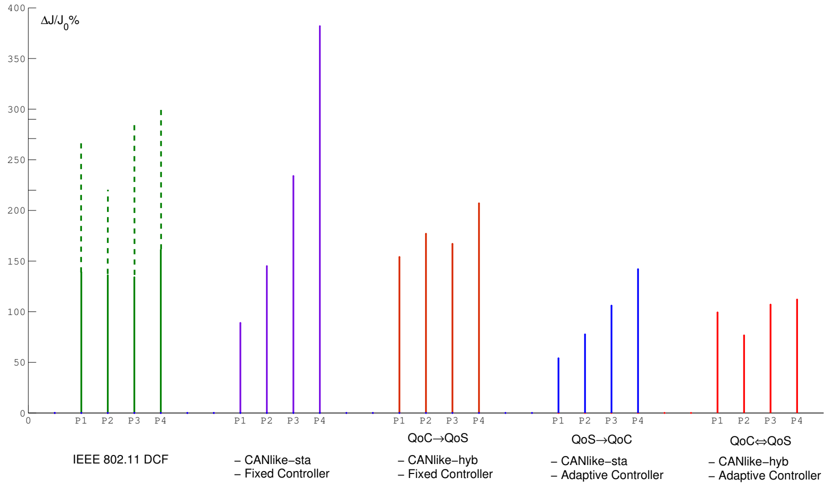


Fig. 11. Graphic representation of the QoC ($\Delta J/J_0\%$).

CSMA protocols. We see the improvement of the performance obtained by the priority-based CSMA MAC protocols (CAN-like) in comparison with IEEE 802.11 DCF. 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 QoC⇌QoS which is the combination of the relation QoC→QoS and the relation QoS→QoC *i.e.* the joint action of the hybrid priorities and of the delay compensation: *i/* 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; *ii/* by the delay compensation, we improve the QoC for all proces control applications compared with the case we do not use the delay compensation; *iii/* by the joint action, the relation QoC⇌QoS allows to improve QoC while maintaining the balanced aspect. And then we can consider the possibility of more applications which satisfy a given performance criteria.

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] 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.

[2] 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.

[3] A. Cervin, "Integrated control and real-time scheduling," Ph.D. dissertation, Department of Automatic Control, Lund Institute of Technology, Sweden, April 2003.

[4] 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.

[5] B. Crow, W. I., K. L.G., and S. P.T., "IEEE 802.11 Wireless Local Area Networks," *IEEE Communications Magazine*, vol. 35, pp. 116–126, September 1997.

[6] H. Nguyen Xuan, "Réseaux de communication et applications de contrôle-commande," Ph.D. dissertation, Univ de Toulouse, INSA, LAAS, Toulouse, France, Dec 2011.

[7] N. Pereira, B. Andersson, and E. Tovar, "WiDom: A Dominance Protocol for Wireless Medium Access," *IEEE Transactions on Industrial Informatics*, vol. 3, no. 2, pp. 120–130, May 2007.

[8] B. GmbH. (1991) Can specification 2.0 (part a). [Online]. Available: www.can-cia.de/fileadmin/cia/specifications/CAN20A.pdf

[9] 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.

[10] 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.

[11] M. Ohlin, D. Henriksson, and A. Cervin, *TrueTime 1.5 – Reference Manual*, Lund Institute of Technology, Sweden, Jan 2007.

[12] K. J. Åström and B. Wittenmark, *Computer-controlled systems: Theory and design*, 3rd ed. Prentice Hall, 1997.