

# A Control Approach to Bandwidth Management in Networked Control Systems

Manel Velasco\*, Josep M. Fuertes\*, Caixue Lin<sup>†</sup>, Pau Martí<sup>†</sup> and Scott Brandt<sup>†</sup>

\*Automatic Control Department, Technical University of Catalonia

Pau Gargallo 5, 08028 Barcelona, Spain

Email: {manel.velasco,josep.m.fuertes}@upc.es

<sup>†</sup>Department of Computer Science, University of California at Santa Cruz

1156 High Street, Santa Cruz, CA 95064, USA

Email: {lcx,pmarti,sbrandt}@cs.ucsc.edu

**Abstract**— Bandwidth allocation techniques for control loops closed over communication networks are based on static strategies that ensure average control performance at the expenses of permanently occupy the available bandwidth. In this paper we present a dynamic approach to bandwidth management in networked control systems that allow control loops to consume bandwidth according to the dynamics of the controlled process meanwhile attempting to optimize overall control performance. By augmenting the original state-space representation of each controlled process with a new state variable that describes the network dynamics, 1) the allocation of bandwidth to control loops can be done locally at run time according to the state of each controlled process without incurring in overload situations and 2) control laws can be designed to account for the variations on the assigned bandwidth preventing the unexpected control performance degradation and even destabilization that would otherwise occur. Experimental data show that the approach we present permits to save bandwidth and even to improve control performance with respect to the static strategy.

## I. INTRODUCTION

Nowadays, embedded systems are expected to provide more functionalities and better application performance within the available resources (battery, processor, bandwidth, etc) to avoid increasing costs. And this is true for networked control systems, a special class of embedded systems, where sensors, controllers and actuators are interconnected using a common bus network architecture.

Traditionally, bandwidth allocation and scheduling techniques for networked control systems are based on static strategies that ensure average control performance at the expenses of permanently occupy the available bandwidth. However, static techniques may not be efficient when changing conditions occur at the control application level or at the network level, because pre-assigned resources may be underutilized. And those could be made available to other applications to provide new functionalities or improve performance.

In this paper we present an approach to dynamic bandwidth management in networked control systems that allow control loops to consume (or alternatively save) bandwidth according to the dynamics of the controlled process meanwhile attempting to optimize overall control performance. By augmenting the original state-space representation of each controlled process with a new state variable that describes the network dynamics, the technique we present provides the following two advantages.

First, the allocation of bandwidth to control loops can be done locally at run time according to the state of each controlled process without incurring in overload situations. For real-time systems with multiple control loops, resource allocation techniques based on state feedback pursuing control performance optimization have been presented in [12]. However, the successful implementation of these techniques require to make the state of each control loop available to a central entity (resource manager in the operating system, see for example [2]). For networked control systems, this would require to have a master node permanently receiving messages on the state of all networked control loops, which may not be feasible in terms of network traffic and not practical for networks driven by other protocols than master-slave. The control approach to bandwidth management we present allows for a distributed and cooperative management of the available resources among all the control loops on the network.

Second, control laws can be designed to account for the variations on the assigned bandwidth preventing the unexpected control performance degradation and even destabilization that would otherwise occur. As it can be concluded from the survey on control methodologies in networked control systems presented in [15], sophisticated results in control theory are required in order to cope with the complexity derived from inserting the network into control loops. In our approach, by customizing the design of the control law according to the augmented state space representation (that includes a description of the network dynamics), we can design each controller directly applying classical control design techniques, which assure stability and meeting the required control performance specifications.

Experimental data show that the approach we present permits to save bandwidth and even to improve control performance with respect to the traditional static strategy to bandwidth allocation in networked control systems.

## II. RELATED WORK

While co-design of control and processor allocation/scheduling has received considerable amount of attention in the literature, allocation/scheduling on networked control systems is still an area that need to be further explored.

Static scheduling strategies for networked control systems can be found in [5], [6], [3] and [9]. A scheduling algorithm is presented in [5] to determine data sampling times, so

that the performance requirement of each control loop is satisfied as well as the utilization of network resources is considerably increased. An extension of this algorithm to the bandwidth allocation of CAN can be found in [6]. We focus on minimizing the network utilization while improving control performance. In [3] a static scheduling optimization approach based on scheduling and control co-design is presented. In [9] a method for obtaining specifications on real-time schedulers that assure overall feedback system performance is presented. The technique we present targets similar control performance optimization problems but for the case of dynamic bandwidth allocation.

Dynamic strategies to scheduling and control co-design in networked control systems can be found in [16], [8], [14], [13] and [17]. In [16] a dynamic arbitration technique to grant network access to the control loop with highest error is presented. Our approach is similar on the baseline idea, but focusing on bandwidth allocation rather than in the priority assignment. The approach presented in [8] focuses on the middleware scheduling of network access through a polling scheme. Our resource allocation architecture does not require a master arbitrating the network access. The scheduling method presented in [14] allows for sampling period adjustment to allocate bandwidth to other type of messages. Although the sampling adjustment is considered on the control analysis, it is not performed according to the dynamics of the control loop as we do. In [13], traffic for networked control systems is reduced using deadbands, approach that demands to trade off traffic and control performance. We aim at improving control performance while reducing bandwidth consumption, following the scheduling strategy presented in [17] but in terms of bandwidth allocation.

### III. SYSTEM ARCHITECTURE AND PROBLEM DESCRIPTION

The system we consider is a networked control system with  $n$  control loops  $c_1, c_2, \dots, c_n$ , each one in charge of controlling a plant (or controlled process). Each plant in closed loop  $c_i$  can be specified by the following linear discrete-time system model [1] (where we assume full state availability)<sup>1</sup>

$$\vec{x}_{i,n+1} = \Phi_i(h_i)\vec{x}_{i,n} + \Gamma_i(h_i)u_{i,n} \quad (1)$$

If the input  $u_{i,n}$  is given by a state feedback control law designed according to (1), the new system dynamics (including the process and the control law, i.e., closed-loop system) are given by the following state-space representation

$$\vec{x}_{i,n+1} = \Phi_i(h_i)\vec{x}_{i,n} + \Gamma_i(h_i)L_i(h_i)\vec{x}_{i,n} \quad (2)$$

where  $L_i(h_i)$  is the control law. In (1) or (2),  $h_i$  is the sampling period, which is obtained from the bandwidth  $b_i$  to be assigned to each control loop according to (3),

$$b_i = \frac{m_i}{h_i} \quad (3)$$

where  $m_i$  is the time spent on the messaging required to perform each closed loop operation (which may include

<sup>1</sup>For the sake of simplicity, we shall omit the  $i$  subscript when not required.

data exchange from sensor to controller, and from controller to actuator). By assuming this time constant, any change on  $b_i$  will directly imply a change on  $h_i$  (and viceversa). Therefore, henceforth, either  $b_i$  or  $h_i$  will be used to denote bandwidth (or sampling period).

For each control loop, we assume that the required messaging to carry out all sensor-controller-actuator transactions is done within each sampling period (i.e., deadline is equal to the sampling period). And we assume a real-time network that guarantees all the deadlines upon any bandwidth allocation among all control loops.

Figure 1 illustrates the architecture we consider, where the allocation problem to be distributedly solved is how to assign bandwidth  $b_i$  to each control loop taking into account the state of each plant  $\vec{x}_i$ , obeying the following rationale: when a controlled process is in equilibrium, the control loop requires little bandwidth, but when the controlled process is affected by a perturbation, an increase on the assigned bandwidth will hasten system recovery, improving control performance.

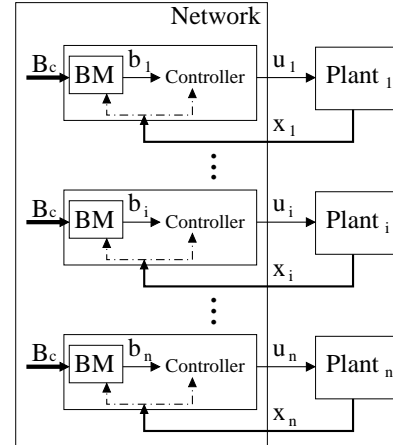


Fig. 1. Networked control system architecture

To solve the problem, the only *global* knowledge each control loop configuration has is the current network bandwidth utilization  $B_c$ . Therefore, by using  $B_c$  and  $\vec{x}_i$ , each node must infer the appropriate sampling period  $h_i$  (or bandwidth to be consumed,  $b_i$ ) to improve the overall control performance (which is represented in Figure ?? by *BM*, bandwidth management). We assume that not all closed loops can simultaneously execute at the highest sampling frequency, providing the best possible control performance equivalent to what they would provide if each one had a dedicated network.

### IV. LOCAL BANDWIDTH MANAGEMENT

In this section we introduce the approach of extending the state-space model of each networked plant in such a way that the bandwidth management can be done locally, at the network node, thus solving the problem posed earlier.

#### A. Extending state-space models

If we extend the state vector of each system (1) with a new state variable  $b$  aimed at describing the bandwidth management dynamics, the new process can be modeled by

equation (4).

$$\begin{bmatrix} \vec{x}_{n+1} \\ b_{n+1} \end{bmatrix} = \begin{bmatrix} \Phi(h) & \alpha \\ \beta & \gamma \end{bmatrix} \begin{bmatrix} \vec{x}_n \\ b_n \end{bmatrix} + \begin{bmatrix} \Gamma(h) \\ \delta \end{bmatrix} u_n \quad (4)$$

In (4),  $\alpha$ ,  $\beta$ ,  $\gamma$  and  $\delta$  are matrices of suitable dimension to be specified, which will determine the relation of the bandwidth management dynamics  $b_n$  with the original state variables  $\vec{x}_n$ , as further detailed in (5) and (6) respectively.

$$\vec{x}_{n+1} = \Phi(h)\vec{x}_n + \alpha b_n + \Gamma(h)u_n \quad (5)$$

$$b_{n+1} = \beta \vec{x}_n + \gamma b_n + \delta u_n \quad (6)$$

If input  $u_n$  is now given by a new state feedback gain  $L = [l_{\vec{x}}, l_b]$  (as in (7)) designed according to (4),

$$u_n = [l_{\vec{x}}, l_b] \begin{bmatrix} \vec{x}_n \\ b_n \end{bmatrix} \quad (7)$$

the extended model closed-loop system dynamics can be specified by (4) and (7). Note that the gain  $L$  in (7), is designed considering both process to be controlled ( $\vec{x}_n$ ) and bandwidth dynamics ( $b_n$ ). Therefore, the evolution of both type of variables (recall (5) and (6)) is determined by a combination of the two of them given by the controller gain.

### B. Model customization

The previous model (4) will allow each control loop configuration to obtain at each execution  $b_{n+1}$ , its next bandwidth requirements (or sampling period). In order to specify the desired behaviour on the bandwidth management dynamics according to the original state variables, matrices  $\alpha$ ,  $\beta$ ,  $\gamma$  and  $\delta$  in (4) have to be adequately customized. Let us discuss some properties of the extended model, depending on these matrices.

- If  $\alpha \neq 0$ , the dynamics of the original system are being modified (compare (1) and (5)), which is *not* a real choice. Consequently,  $\alpha$  should be 0. However, this does not mean that the bandwidth dynamics is not influencing the process evolution. Note that the control law designed using the extended model will be considering process and bandwidth, and the excitation input  $u_n$  in (7) depends on the control law and both variables  $\vec{x}_n$  and  $b_n$ .
- If  $\delta \neq 0$ , the input  $u_n$ , which in closed-loop configuration will be given by the state feedback control, will affect the evolution of  $b_n$  (as it can be seen in (6)). However, since the control law is, *by default*, used to control only the process (original system (1)), we set  $\delta = 0$ . However, this may be reconsidered if we want the control law to act on the new state variable.

In the following, assuming  $\alpha = 0$  and  $\delta = 0$

- If  $\beta = 0$  and  $\gamma = I$ , then, for each  $n$ ,  $b_{n+1} = b_n$  (see (6)). That is, no dynamics are specified on the network bandwidth.
- If  $\beta = 0$  and  $\rho(\gamma) < 1$ , where  $\rho(\cdot)$  is the spectral radius, bandwidth will decrease at each iteration, tending towards 0, thus leading to a system with no bandwidth, which has no sense.
- If  $\beta = 0$  and  $\rho(\gamma) > 1$ , bandwidth will increase at each iteration, tending towards  $\infty$ , thus leading to a system with infinite bandwidth, which is not realistic.

- If  $\beta \neq 0$  and  $\gamma = 0$ , bandwidth evolution will depend only on the original state variables. Consequently, the more quickly these variables move, the faster will be the bandwidth dynamics, introducing abrupt and too frequent changes between consecutive bandwidth allocations, which may be undesirable for control systems (introducing performance degradation and destabilization).
- If  $\beta \neq 0$  and  $\gamma \neq 0$ , the evolution of each control loop in the network (in term of original process evolution and bandwidth management) will depend on the specific chosen values for  $\beta$  and  $\gamma$ , which is the most desired choice. Recall that care must be taken when specifying  $\beta$  and  $\gamma$  in order to enforce smooth transitions between successive bandwidth allocations.

Therefore, the model we will consider is

$$\begin{bmatrix} \vec{x}_{n+1} \\ b_{n+1} \end{bmatrix} = \begin{bmatrix} \Phi(h) & 0 \\ \beta & \gamma \end{bmatrix} \begin{bmatrix} \vec{x}_n \\ b_n \end{bmatrix} + \begin{bmatrix} \Gamma(h) \\ 0 \end{bmatrix} u_n \quad (8)$$

Although the extended model (8) is flexible enough to specify different types of dynamics, if any linear combination of  $\beta$  and  $\gamma$  does not satisfy the desired dynamics for the bandwidth management (new state variable), we can use nonlinear relations at the expenses of slightly increasing the complexity of the design of the control law, as we explain next.

### C. Specification of the Bandwidth Management Dynamics

The bandwidth management mechanism has to obey the following specifications:

- For each control loop, the bigger the error (defined as the norm of its state variables (9)) is,

$$e_i = |\vec{x}_{i,n}| \quad (9)$$

the higher the bandwidth  $b_{i,n}$  (or the shorter the sampling period  $h_{i,n}$ ) to be allocated.

- For the set of networked control loops, a given constraint on the bandwidth utilization factor must be kept,

$$\sum_{i=1}^n b_{i,n} \leq B_g \quad (10)$$

where  $B_g$ , the global bandwidth utilization factor (or network capacity) can be either the total available bandwidth or a desired portion.

Note that both specifications may be conflicting because the first attempts to locally have higher bandwidth allocations meanwhile the second is globally restricting the overall bandwidth.

To express these specifications using the linear relation given by matrices  $\beta$  and  $\gamma$  in (8) is rather difficult. Instead, we define an heuristic function that mathematically behaves as required by our specifications, to be included in (8).

For each control loop, we define that at each closed loop operation, the next sampling period  $h_{n+1}$  (or bandwidth allocation) 1) is a function of the exponential of the original state variables  $\vec{x}_n$  (in order to ensure positive values with smooth transitions), and 2) takes into account the current and global bandwidth utilization factors,  $B_c$  and  $B_g$  (in order to fulfil the bandwidth utilizations constraint defined by (10)). Note that by knowing these factors, the available

(not occupied) bandwidth  $B_a$  can be calculated as follows (recall (3)).

$$B_a = B_g - (B_c - \frac{m}{h_n}) = B_g - (B_c - b_n) \quad (11)$$

Using (11), that is, taking into account the available bandwidth, the shortest possible sampling period  $h_{short}$  that can be assigned to a control loop is

$$h_{short} = \frac{m_n}{B_a} \quad (12)$$

Therefore, the next sampling period can be calculated follows:

$$h_{n+1} = (h_{long} - h_{short}) e^{-c|\bar{x}_n|} + h_{short} \quad (13)$$

where  $h_{long}$  and  $c$  are the longest possible period and the criticalness, both to be defined for each control loop. The criticalness determines how quick a control loop will increase or decrease its period according to error. Higher values for  $c$  will imply more abrupt changes on the sampling period. Therefore, looking at (13), the following behaviour has been achieved. If there is no error ( $|\bar{x}_n| = 0$ ), then  $h_{n+1} = h_{long}$ . And if the error increases, the sampling period decreases (and viceversa). Note that calculating  $h_{n+1}$  implies substituting (11) and (12) in (13).

Substituting in the extended model (8) the linear relation that would specify  $b_{n+1}$  (through  $\beta$  and  $\gamma$ ) by the new equation specifying the bandwidth management dynamics (13) in terms of the sampling period, we obtain the following model

$$\begin{bmatrix} \bar{x}_{n+1} \\ h_{n+1} \end{bmatrix} = \begin{bmatrix} \Phi(h) \cdot \bar{x}_n \\ (h_l - h_s) e^{-c|\bar{x}_n|} + h_s \end{bmatrix} + \begin{bmatrix} \Gamma(h) \\ 0 \end{bmatrix} u_n \quad (14)$$

where  $h_l$  stands for  $h_{long}$  and  $h_s$  for  $h_{short}$ . Note that (14) is nonlinear. Therefore, the application of the extended model will require the use of non-linear control design techniques. This is illustrated next.

## V. APPLICATION EXAMPLE

In this section we present an instance of the extended model (14) for the *Ball & Beam* process [1], which can be represented by the following linear discrete-time invariant state-space model:

$$\begin{bmatrix} x_{n+1}^1 \\ x_{n+1}^2 \end{bmatrix} = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_n^1 \\ x_n^2 \end{bmatrix} + \begin{bmatrix} \frac{1}{2} h^2 \\ h \end{bmatrix} \cdot u_n \quad (15)$$

In equation (15),  $x_n^1$  and  $x_n^2$  are the *original* state variables (that is,  $\bar{x}_n = [x_n^1, x_n^2]^T$  in (4)), and  $h$  is the sampling period. The sampling period, which appears on the system and input matrices as a result of the discretization process ((15) is a discrete-time model obtained via discretization of the continuous-time model) has no index, because it is assumed constant. However, if we add the sampling period  $h_n$  as a new state variable to obtain the extended model,  $h_n$  will vary. Therefore, it will be necessary to assign the appropriate  $n$ -index to the  $h$ , which is  $n + 1$ .

### A. Extended Model

For the *ball and beam* process and taking into account a) its original model of the (15), b) the generic extended model (14) and c) the bandwidth management dynamics specified by (13), its extended model is as follows:

$$\begin{bmatrix} x_{n+1}^1 \\ x_{n+1}^2 \\ h_{n+1} \end{bmatrix} = \begin{bmatrix} x_n^1 + x_n^2 ((h_l - h_s) e^{-c|\bar{x}_n|} + h_s) \\ x_n^2 \\ (h_l - h_s) e^{-c|\bar{x}_n|} + h_s \end{bmatrix} + \begin{bmatrix} \frac{1}{2} ((h_l - h_s) e^{-c|\bar{x}_n|} + h_s)^2 \\ (h_l - h_s) e^{-c|\bar{x}_n|} + h_s \\ 0 \end{bmatrix} \cdot u_n \quad (16)$$

To simplify the notation in next section, we compact (16) as follows

$$\begin{bmatrix} x_{n+1}^1 \\ x_{n+1}^2 \\ h_{n+1} \end{bmatrix} = \begin{bmatrix} f_1(x^1, x^2, h) \\ f_2(x^1, x^2, h) \\ f_3(x^1, x^2, h) \end{bmatrix} + \begin{bmatrix} g_1(x^1, x^2, h, u) \\ g_2(x^1, x^2, h, u) \\ g_3(x^1, x^2, h, u) \end{bmatrix} \quad (17)$$

The nonlinear characteristics of the extended model of the *Ball & Beam* (16) do no impair the design of an appropriate control law, as we detail next.

### B. Controller design

In this section we summarize the design of the control law [7], based on linearization techniques.

#### 1) Linearization with respect to the equilibrium point:

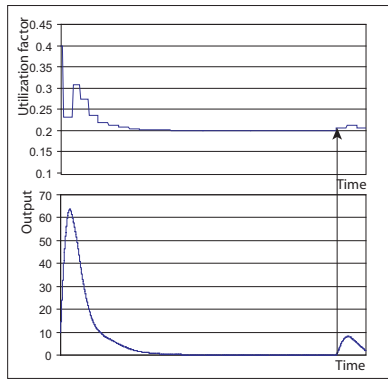
In a small neighborhood of the equilibrium point, we approximate the nonlinear system by its linearization about the equilibrium point. Although this is the simplest technique, it does not guarantee that the system will be stable outside of the neighborhood. The controller design process is as follows

- We obtain the linearized state space model about the equilibrium point  $x^1 = 0$ ,  $x^2 = 0$  and  $h = h_l$ , by constructing the Jacobian matrices of the system and input matrices of the nonlinear model (17) and evaluating them at the equilibrium point.

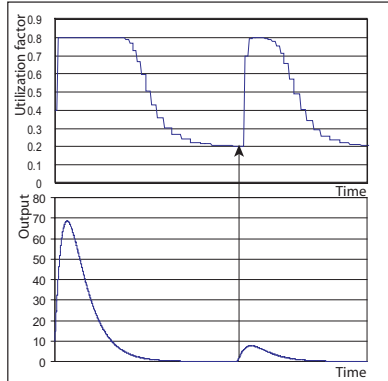
$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x^1} & \frac{\partial f_1}{\partial x^2} & \frac{\partial f_1}{\partial h} \\ \frac{\partial f_2}{\partial x^1} & \frac{\partial f_2}{\partial x^2} & \frac{\partial f_2}{\partial h} \\ \frac{\partial f_3}{\partial x^1} & \frac{\partial f_3}{\partial x^2} & \frac{\partial f_3}{\partial h} \end{bmatrix}_{(0,0,h_l)} = \begin{bmatrix} 1 & h_l & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{\partial g_1}{\partial u} \\ \frac{\partial g_2}{\partial x^1} \\ \frac{\partial g_3}{\partial x^2} \end{bmatrix}_{(0,0,h_l)} = \begin{bmatrix} \frac{1}{2} h_d^2 \\ h_d \\ 0 \end{bmatrix}$$

- Using the linearized matrices  $A$  and  $B$ , we design a control law using classical techniques such as pole placement (using a the default longest period  $h_l$ ), and verify that the control law applied to the nonlinear system keeps the system stable in the desired neighborhood of the equilibrium point.



(a) Scenario 1: Non critical control loop (small  $c$ )



(b) Scenario 2: critical control loop (bigger  $c$ )

Fig. 2. Bandwidth management as a function of the error (output) using a controller designed via linearization techniques

### C. Controller performance analysis

The behaviour of the *Ball & Beam* process controlled in closed loop by a controller implementing the control law obtained using this approach can be seen in Figure 2. We show two scenarios depending on the parameter criticalness ( $c$  in (13)). Figure 2 (a) corresponds to a non critical control loop, with a small value of  $c$ , and Figure 2 (b) corresponds to a critical control loop, with a higher value of  $c$ . In each scenario, a control loop executing in isolation on a network varies its execution period (labeled as *utilization factor*) according to the changes in the controlled process (labeled as *Output*), without violating the global bandwidth utilization factor,  $B_g = 0.8$ . Since in each scenario there is only one control loop, the bandwidth utilization factor directly reflects the sampling period variation (recall equation (3)).

In each scenario in Figure 2 each *Ball & Beam* is affected by two perturbations (that deviates the *outputs* from zero), a big one at the beginning and a small one later on. Upon each perturbation, the control loop (which had a default longest period  $h_l$  equivalent to a bandwidth utilization factor of  $b = 0.2$  consumes more bandwidth (shortens its period), proportionally to the *Ball & Beam* output, according to the criticalness parameter. More abrupt changes are made by the control loop with higher criticality, which at the same time, affect the evolution of the process output.

For Figure 2 it is important to point out that although the execution frequency of the control loop implementing the control law varies, the control law was designed according to  $h_l$ , which is constant. Such application of the linearization

technique ensures system stability, but does not guarantees *good* performance within the available resources.

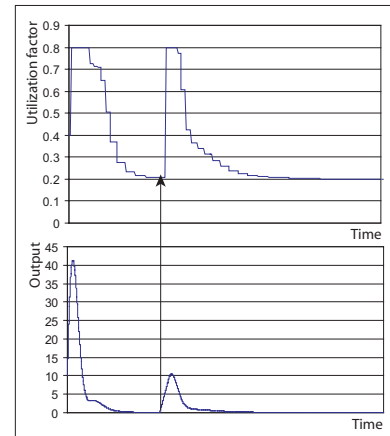


Fig. 3. Control loop executing a set of control laws obtained via linearization techniques (bigger  $c$ )

To improve control performance (to shorten the deviation of the *Ball & Beam* and fasten system recovery with respect to zero), we also applied the linearization technique not only for  $h_l$  but for the set of possible periods that may appear at run time, and designed a set of control law accordingly. At run time, depending on the control loop period, the corresponding control law is executed, adaptively. Note that for such system the stability can be analysed as in [4]. In Figure 3 we show the *Ball & Beam* output and network utilization for the case of the control loop executing the set of control laws, designed with the same criticalness than the single control law of scenario 2 (Figure 2 (b)). As it can be seen comparing 3 and 2 (b), the performance when executing the set of controllers is improved in terms of both control performance (i.e., smaller deviation) and network bandwidth consumption (i.e., smaller pressure on the network).

## VI. RESULTS

In this section we evaluate the effectiveness of the control approach to bandwidth management in networked control systems. We do so by using the following configuration: we simulated three control loops over a network that adaptively implement the set of control laws obtained via linearization techniques for the *Ball & Beam* process (see section V-B.1).

The desired global bandwidth utilization factor for the three control loops is  $B_g = 97\%$  (thus reserving a percentage of bandwidth capacity of  $3\%$ ). Each control loop

- is in charge of controlling a *Ball & Beam* modelled as in (16)
- is assigned different criticalness values ( $c = 0.1, 1, 10$ ) for different runs
- requires a messaging (and calculations) for each sensor-controller-actuator operation ( $m_i$ ) of  $0.135s$ .
- is allowed to choose sampling periods within  $0.3s$  and  $0.5s$ .

To provide a direct comparison with traditional static allocations techniques for networked control systems, we implemented a baseline policy in which all controllers

always share the available resources equally, that is, each control loop is given  $\frac{97\%}{3}$  of the bandwidth.

For the allocation policy we have presented and for the traditional one, we ran the three control loops for 1 hour and randomly generated perturbations for each *Ball & Beam* with different average perturbations intervals. The distance between two consecutive perturbations intervals on the same system varies in such a way that the system may be continuously perturbed or almost never perturbed (thus capturing all possible scenarios).

Figure 4 shows the analysis of control performance and bandwidth consumption of the dynamic allocation technique we have presented compared to the static, for the different criticalness values, and for the different perturbation intervals and. The data of Figure 4 is in percentage, and any data above 0 means either improving control performance or reducing bandwidth consumption. The numbers above the bars are the different criticalness values ( $c = 10$  implies an aggressive bandwidth management and  $c = 0.1$  implies a timid management).

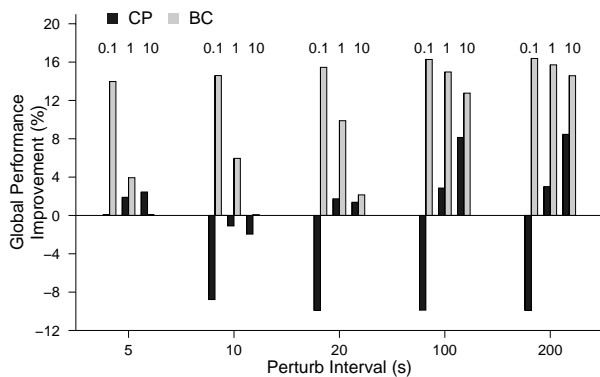


Fig. 4. Analysis of control performance (CP) and bandwidth consumption (BC) of the dynamic allocation technique compared to the static, for different perturbation intervals and for different criticalness values.

First of all, it is worth differentiating two scenarios in Figure 4. Up to perturbation interval 20s, intervals are short enough so that all the available network bandwidth could be allocated to the control loops for a big value of  $c$  (e.g., 10). From perturbation interval 20s, intervals are long enough so that even with a big value of  $c$ , a portion of bandwidth is saved by the control loops when there is no error. Clearly, it can be seen in Figure 4 that in this second scenario is when the improvement on control performance is more significant. Second, it is also worth mentioning that for example, although with  $c = 0.1$  there is no control performance improvement (rather a slight degradation), the bandwidth saving is important. Therefore, adequately trading off control performance and bandwidth consumption is crucial. And finally, from Figure 4 we see that whatever is the perturbation interval and the value of the criticalness parameter, the dynamic approach to bandwidth management always permits to save bandwidth.

## VII. CONCLUSIONS

In this paper we have presented a dynamic control-based approach to bandwidth management in networked control

systems. The key assumption for the presented technique is that the information needed to decide each allocation is not centrally located, which impairs the application of known dynamic allocation techniques for processor time in real-time control. Rather, each allocation is decided locally, for each closed loop configuration, taking into account both the local plant dynamics and the global network dynamics.

We have shown experimental results that demonstrate that the application of our dynamic bandwidth allocation technique permits to save bandwidth and even to improve control performance with respect to the static technique traditionally used in networked control systems.

## REFERENCES

- [1] K. J. Astrom and B. Wittenmark, "Computer-controlled systems", 1997
- [2] S. A. Brandt, S. Banachowski, C. Lin and T. Bisson, "Dynamic Integrated Scheduling of Hard Real-Time, Soft Real-Time and Non-Real-Time Processes," IEEE Real-Time Systems Symposium, pp 396-407, Cancun, Mexico, December 2003
- [3] M. S. Branicky, S. M. Phillips, and W. Zhang, "Scheduling and feedback co-design for networked control systems", *IEEE Conference on Decision and Control*, Las Vegas, December 2002
- [4] M. Dogruel and U. Zgner, "Stability of a Set of Matrices: A Control Theoretic Approach", in *34th Conference of Decision and Control*, September 1995
- [5] S. H. Hong, "Scheduling algorithm of data sampling times in the integrated communication and control systems," *IEEE Transactions on Control Systems Technology*, 3(2), 225-230, June 1995.
- [6] S. H. Hong, and W-H. Kim, "Bandwidth allocation scheme in CAN protocol," *IEE Proceedings Control Theory and Applications*, 147(1), 37-44, January 2000.
- [7] Isidori, A. 'Nonlinear Control Systems'. Spring Verlag, New York, 1989
- [8] V. Liberatore, "Scheduling of Network Access for Feedback-Based Embedded Systems," *Quality of Service over Next Generation Internet, SPIE ITCOM*, 2002.
- [9] Q. Ling, and M.D. Lemmon, "Soft real-time scheduling of networked control systems with dropouts governed by a markov chain," *American Control Conference*, Denver, Colorado, June 2003.
- [10] P. Marti, G. Fohler, K. Ramamritham, and J.M. Fuertes, "Jitter Compensation for Real-time Control Systems," *Proc. of the 22rd IEEE Real-Time System Symposium*, London, UK, December 2001.
- [11] P. Marti, G. Fohler, K. Ramamritham, and J.M. Fuertes, "Improving Quality-of-Control using Flexible Timing Constraints: Metric and Scheduling Issues," *Proc. of the 23rd IEEE Real-Time System Symposium*, Austin, TX, USA, December 2002.
- [12] P. Marti, C. Lin, Scott Brandt, M. Velasco, J.M. Fuertes, "Optimal State Feedback Based Resource Allocation for Resource-Constrained Control Tasks," *Proc. of the 25th IEEE Real-Time System Symposium*, Lisbon, Portugal, December 2004.
- [13] P. Otanez, J. Moyne, and D. Tilbury, "Using Deadbands to Reduce communication in Networked Control Systems," *American Control Conference*, 2002.
- [14] H.S.Park, Y.H.Kim, D-S.Kim, and W.H.Kwon, "A Scheduling Method for Network-based Control Systems," *IEEE Transaction on Control System Technology*, Vol.10, No.3, pp. 318-330, May, 2002.
- [15] Y. Tipsuwan, and M-Y. Chow, "Control methodologies in networked control systems," *Control Engineering Practice*, 11, 1099-1111, 2003.
- [16] G.C. Walsh, and H. Ye, "Scheduling of Networked Control Systems," *IEEE Control Systems Magazine*, 21(1), pp 57-65 February 2001.
- [17] J. Yépez, P. Martí and J. M. Fuertes, "Control Loop Scheduling Paradigm in Distributed Control Systems," *29th Annual Conference of the IEEE Industrial Electronics Society*, Roanoke, USA, November, 2003.