Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Optimal Integrated Control and Scheduling of Systems with Communication Constraints

Mohamed El Mongi Ben Gaid, Arben Çela and Yskandar Hamam

Abstract— **This paper addresses the problem of the optimal control and scheduling of Networked Control Systems over limited bandwidth time-slotted networks. Multivariable linear systems subject to communication constraints are modeled in the Mixed Logical Dynamical (MLD) framework. The translation of the MLD model into the Mixed Integer Quadratic Programming (MIQP) formulation is described. This formulation allows the solving of the optimal control and scheduling problem using efficient branch and bound algorithms. Advantages and drawbacks of on-line and off-line scheduling algorithms are discussed. Based on this discussion, an efficient on-line scheduling algorithm, which can be seen as a compromise, is presented and its performance is evaluated.**

I. Introduction

Shared communication networks are increasingly being used to support the information exchange in distributed control systems. Using a control network has many advantages, such as a higher reliability, an easier deployment and maintenance. However, many communication networks are subject to bandwidth constraints (for example Underwater Acoustic Networks [1] or Wireless Networks [2]). Reasons behind these resource constraints are multiple. Guaranteeing deterministic real-time communications induces important restrictions on the available bandwidth, especially when the communication channel is noisy. In other situations, some nodes of the communication network may be autonomous and battery-powered. The lifetime of the batteries limits the energy used in the transmission and thus limits the bandwidth of the communications. On the other hand, in the automotive industry, an increasing number of applications are being developed in order to improve the driving safety and comfort. These applications need to share and to exchange an important amount of information. As a consequence, the local bus is becoming more and more loaded. Using a more expensive technology can solve these problems, but components price is strongly balanced by production cost requirements.

Many research works addressed the problems stemming from the use of communication networks in control loops from different points of view, see for example [3]–[6].

Recently, it was shown that considering jointly control and real-time scheduling leads to an improvement of the control performance [7], [8]. In [9], the problem of the optimal off-line control and scheduling over TDMA networks is addressed. Network scheduling is modeled using the notion of communication sequence [10]. This modeling leads to a complex combinatorial optimization problem, whose solution is obtained by a heuristic search. In [11], a model predictive controller is used to calculate on-line both the control and the scheduling of a time-slotted network, assuming that control commands that cannot be updated are set to zero. Using this assumption, as well as a cost function based on the infinity norm, the optimization problem was shown to be equivalent to the Generalized Linear Complementarity Problem.

In this paper, the problem of the distributed control over deterministic real-time networks is addressed. The corresponding optimal control and scheduling problem is formulated. An efficient approach for the solving of this problem is proposed. An on-line scheduling algorithm, which can be seen as a compromise between off-line and on-line scheduling algorithms, is presented and its performance illustrated.

This paper is organized as follows. In section II, the modeling of a multivariable control system with communication constraints in the Mixed Logical Dynamical framework is described. Section III addresses the translation of this model into the Mixed Integer Quadratic Programming Formulation. The finite-time optimal control and scheduling problem based on this formulation is then illustrated. Finally, in section IV, an efficient on-line scheduling algorithm (the Optimal Pointer Placement scheduling algorithm) is proposed and its effectiveness is shown in simulation examples.

II. Problem formulation

Consider a collection of $\mathcal{N}$ continuous-time LTI systems $(S^j)_{1\leq j\leq \mathcal{N}}$ described by the state equations:

$$\dot{x}^j(t) = A_c^j x^j(t) + B_c^j u^j(t) \tag{1}$$

where $u^j(t)$ is a column vector with m_j entries, $1 \leq j \leq \mathcal{N}$. Assume that all the systems are controllable and that their state vectors are measurable. Suppose that system S^j contains m_j distinct distributed actuators which are connected to a main controller through a bus with a limited bandwidth. Let $A_c = Diag\left(A_c^1,\ldots,A_c^{\mathcal{N}}\right)$ and $B_c = Diag\left(B_c^1,\ldots,B_c^{\mathcal{N}}\right)$, then systems $(S^j)_{1\leq j\leq \mathcal{N}}$ can be described using the state equation:

$$\dot{x}(t) = A_c x(t) + B_c u(t) \tag{2}$$

where $x(t) = \begin{bmatrix} x^1(t) \\ \vdots \\ x^{\mathcal{N}}(t) \end{bmatrix}$ and $u(t) = \begin{bmatrix} u^1(t) \\ \vdots \\ u^{\mathcal{N}}(t) \end{bmatrix}$.

M-M. Ben Gaid and A. Çela are with the COSI Lab., ESIEE, Cité Descartes, BP 99, 93162 Noisy-Le-Grand Cedex, FRANCE (e-mail: {bengaidm,celaa}@esiee.fr)

Y. Hamam is with the A^2SI Lab., ESIEE, Cité Descartes, BP 99, 93162 Noisy-Le-Grand Cedex, FRANCE (e-mail: hamamy@esiee.fr)

The bus is scheduled using a Time Division Multiple Access (TDMA) scheme. At each slot, only one control command can be sent to a specified actuator. The duration T_s of the time slot, which depends on the bit rate and on the used communication protocol, defines the elementary time unit of the model. A similar model was considered in [9] and [11]. Let m be the total number of actuators of the global system:

$$m = \sum_{j=1}^{\mathcal{N}} m_j \tag{3}$$

Network scheduling can be described by m binary functions δ_i verifying:

$$\delta_i(k) = 1 \Longleftrightarrow \text{the } k^{th} \text{ slot is used to update the } i^{th} \text{ actuator} \tag{4}$$

At the k^{th} time slot, only one message can transmitted on the bus. This constraint can be modeled by the inequality:

$$\sum_{i=1}^{m} \delta_i(k) \leq 1 \tag{5}$$

In order to derive a digital control law, a discrete-time model of system (2), at the sampling period T_s is considered:

$$x(k+1) = Ax(k) + Bu(k) \tag{6}$$

The digital-to-analog converters, which are located at the actuators, use zero-order-holders to maintain the last received control commands constant until new control values are received. Consequently, if a control command is not updated at the k^{th} slot, then it is held constant. This assertion can be modeled by the logic formula:

$$\delta_i(k) = 0 \Longrightarrow u_i(k) = u_i(k-1) \tag{7}$$

where $u(k) = [u_1(k) \ \cdots \ u_m(k)]^T$. Finally, suppose that control signals are limited:

$$L_i \leq u_i \leq U_i \tag{8}$$

and that $L_i < 0$ and $U_i > 0$.

Equations (5), (6), (7) and (8) constitute the considered model of a networked control system with limited communication resources. This model can be addressed using the Mixed Logical Dynamical (MLD) systems framework [12].

III. Finite-time optimal control and scheduling problem

The finite-time optimal control and scheduling problem can be formalized as follows : Find the optimal control sequence $u^N = (u(0), \dots, u(N-1))$ and the optimal communication sequence $\delta^N = (\delta(0), \dots, \delta(N-1))$ which are solutions of the following optimization problem:

$$\begin{cases} \min\limits_{u^N, \delta^N} x^T(N)Sx(N) + \sum\limits_{k=0}^{N-1} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}^T Q \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \\ \text{subject to:} \\ x(k+1) = Ax(k) + Bu(k), \ \forall k \in \{0, \dots, N-1\} \\ \sum\limits_{i=1}^{m} \delta_i(k) \leq 1, \ \forall k \in \{0, \dots, N-1\} \\ \delta_i(k) = 0 \Longrightarrow u_i(k) = u_i(k-1), \ \forall k \in \{0, \dots, N-1\} \\ x(0) \text{ given} \end{cases} \tag{9}$$

where Q and S are definite positive matrices.

Problem (9) is a constrained control problem. In fact, the use of TDMA scheduling imposes to $m-1$ components of $u(k)$ to remain equal to the corresponding components of $u(k-1)$. The only component of $u(k)$ whose value can change at the k^{th} slot is that whose index i satisfies $\delta_i(k) = 1$.

In order to solve this problem, it is necessary to translate the logical formula (7) into linear inequalities. The connective "$\Longrightarrow$" can be eliminated if (7) is rewritten in the equivalent form:

$$u_i(k) - u_i(k-1) = \delta_i(k)u_i(k) - \delta_i(k)u_i(k-1) \tag{10}$$

However, equation (10) contains terms which are the product of logical variables and continuous variables. The use of the procedure described in [12] allows the translation of this product into an equivalent conjunction of linear inequalities. For example, let :

$$z_i(k) = \delta_i(k)u_i(k). \tag{11}$$

Then (10) can be rewritten in the equivalent form:

$$\begin{array}{l} z_i(k) \leq U_i \delta_i(k) \\ z_i(k) \geq L_i \delta_i(k) \\ z_i(k) \leq u_i(k) - L_i(1 - \delta_i(k)) \\ z_i(k) \geq u_i(k) - U_i(1 - \delta_i(k)) \end{array} \tag{12}$$

Note that the same procedure can be applied to $w_i(k) = \delta_i(k)u_i(k-1)$.

Let $\Delta = \begin{bmatrix} \delta(0) \\ \vdots \\ \delta(N-1) \end{bmatrix}$, $U = \begin{bmatrix} u(0) \\ \vdots \\ u(N-1) \end{bmatrix}$, $X = \begin{bmatrix} x(0) \\ \vdots \\ x(N) \end{bmatrix}$, $Z = \begin{bmatrix} z(0) \\ \vdots \\ z(N-1) \end{bmatrix}$, $W = \begin{bmatrix} w(0) \\ \vdots \\ w(N-1) \end{bmatrix}$ and $\mathscr{V} = \begin{bmatrix} \Delta \\ U \\ X \\ Z \\ W \end{bmatrix}$, then problem (9) can be written :

$$\begin{cases} \min\limits_{\mathscr{V}} \frac{1}{2} \mathscr{V}^T H \mathscr{V} + f^T \mathscr{V} \\ \mathscr{A}\mathscr{V} \leq \mathscr{B} \end{cases} \tag{13}$$

where H, f, $\mathscr{A}$, and $\mathscr{B}$ can be easily deduced form the discussion above. Problem (13) is a Mixed-Integer Quadratic Program. The advantage of this formulation is the existence of many efficient academic and commercial solvers, based on the branch and bound algorithm.

Example 1: Consider the collection of 3 continuous-time LTI systems defined by :

$$A_c^1 = \begin{pmatrix} 0 & 130 \\ -800 & 10 \end{pmatrix} \qquad B_c^1 = \begin{pmatrix} 0 \\ 224 \end{pmatrix} \qquad (S^1)$$

$$A_c^2 = \begin{pmatrix} 0 & 14 \\ -250 & -200 \end{pmatrix} \qquad B_c^2 = \begin{pmatrix} 0 \\ 620 \end{pmatrix} \qquad (S^2)$$

$$A_c^3 = \begin{pmatrix} 0 & 0 & 0 & 100 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & -10 & 0 \\ 11.6 & 0 & 1.184 & 0 \end{pmatrix} \qquad B_c^3 = \begin{pmatrix} 0 \\ 0 \\ 10 \\ 10.18 \end{pmatrix} \qquad (S^3)$$

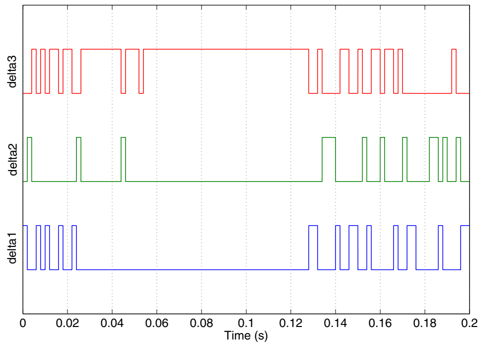

Fig. 1. Optimal schedule

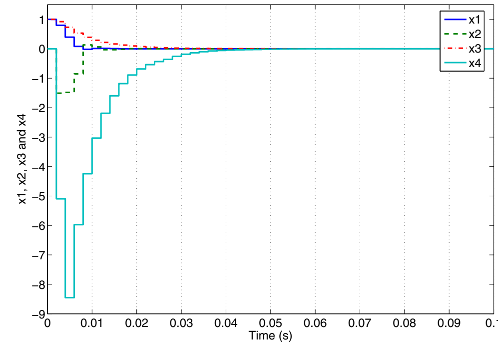

Fig. 2. Global system response - states x_1, x_2 (system S^1) and x_3, x_4 (system S^2) from $t = 0$ s to $t = 0.1$ s

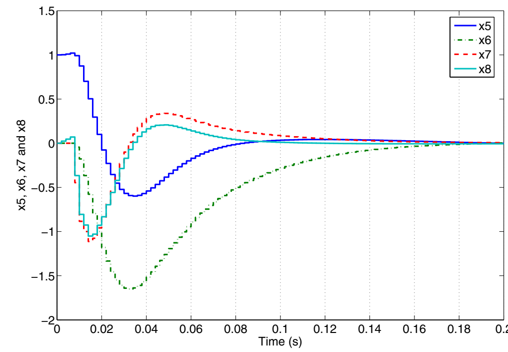

Fig. 3. Global system response - states $x_5 \cdots x_8$ (system S^3)

Systems S^1 and S^3 are open-loop unstable in opposition to system S^2 which is open-loop stable. The control commands are sent to the actuators through a bus with a slot time of 2 ms. A discrete-time model of the global system (composed of S^1, S^2 and S^3) at the sampling period of 2 ms is considered. Design criteria for each system are defined by matrices $Q_1 = Diag(100, 10)$, $R_1 = 1$, $Q_2 = Diag(1000, 10)$, $R_2 = 1$, $Q_3 = Diag(1000, 1000, 1, 1)$ and $R_3 = 1$. Desired closed loop specifications for the global system are described by the closed loop weighting matrices Q :

$$Q = Diag\left(c_1 Q_1, c_2 Q_2, c_3 Q_3, c_1 R_1, c_2 R_2, c_3 R_3\right)$$

and S :

$$S = Diag\left(c_1 Q_1, c_2 Q_2, c_3 Q_3\right)$$

where c_1, c_2 and c_3 are the weighting coefficients. In this example, c_1, c_2 and c_3 were chosen equal to the inverse of steady state performance index of each separate system controlled through a bus having an infinite bandwidth ($c_1 = 2.9$, $c_2 = 0.13$ and $c_3 = 0.016$).

The length of the optimal control and scheduling sequences is $N = 100$. An optimal solution with an error bound of 1×10^{-5} was required. The global system is started from the initial state $x(0) = [1 \;\; 0 \;\; 1 \;\; 0 \;\; 1 \;\; 0 \;\; 0 \;\; 0]^T$. The optimal schedule is depicted in figure 1. In this schedule, $\delta_i = 1$ means that the bus is dedicated to the transmission of control signal u_i. The response of the global system is illustrated in figures 2 and 3.

The first network slots are mainly dedicated to system S^1 until it is stabilized. It can be observed that system S^2, which is open-loop stable and whose response time is bigger than S^1, needs only three time slots to be stabilized. After the stabilization of systems S^1 and S^2, network resources are entirely dedicated to the stabilization of system S^3. When system S^3 is close to the equilibrium state (from $t = 0.13$ s), then its control signals changes are minor. Consequently, the scheduling has no significant impact on control, because control signals of the three systems are relatively constant, which explains the shape of the scheduling diagram, after $t = 0.13$ s. However, this optimal schedule is dependent from initial conditions. If the initial condition $x(0)$ is modified, then the optimal control and schedule would be different. It is clear that such an open-loop schedule, which is generated off-line, cannot be applied at run time as static schedule. In fact, assume that system S^1 is disturbed at $t = 0.024$ s. Observing the schedule, no slots are allocated to the transmission of the messages of system S^1 between $t = 0.024$ s and $t = 0.128$ s, which would induce performance degradations.

These observations show that in the same way as the optimal control, the optimal scheduling depends on the current state of the system. Consequently, fixed schedules may not be optimal if the system is started from another initial state or if it is disturbed at runtime. However, from a computer science point of view, off-line scheduling has many advantages, essentially because it consumes few computing resources and does not induce execution overheads. In order to obtain static schedules which are optimal from a certain point of view, it is necessary to use performance criteria which depend on the intrinsic characteristics of the system, not on a particular evolution, for example, based on the H_2 or H_∞ performance indices [13].

IV. Optimal Pointer Placement Scheduling

Open-loop optimization problems constitute the cornerstone of a successful control method: the model predictive control (MPC). MPC has strong theoretical foundations, and many interesting properties which make it suitable to address constrained control problems. However, its main drawback is that it requires very important computing resources, which make it only applicable to slow systems, like chemical processes. The use of the MPC technique leads to an on-line solving of an open-loop optimization problem. For the particular MLD model described in equations (5)(6)(7)(8), the application of the MPC technique requires the finding of the optimal control sequence $v^N = (v(0), \ldots, v(N-1))$ and the optimal bus allocation sequence $\sigma^N = (\sigma(0), \ldots, \sigma(N-1))$ which are solutions of the following optimization problem:

$$\begin{cases} \min\limits_{v^N, \sigma^N} y^T(N) S y(N) + \sum\limits_{h=0}^{N-1} \begin{bmatrix} y(h) \\ v(h) \end{bmatrix}^T Q \begin{bmatrix} y(h) \\ v(h) \end{bmatrix} \\ \text{subject to:} \\ y(0) = x(k) \\ y(h+1) = Ay(h) + Bv(h) \ , \ h \in \{0, \cdots, N-1\} \\ \sum\limits_{i=1}^{m} \sigma_i(h) \leq 1 \ , \ h \in \{0, \cdots, N-1\} \\ \sigma_i(0) = 0 \Longrightarrow v_i(0) = u_i(k-1) \\ \sigma_i(h) = 0 \Longrightarrow v_i(h) = v_i(h-1) \ , \ h \in \{1, \cdots, N-1\} \end{cases} \tag{14}$$

The solution of this problem is based on the predicted evolution of the system over a horizon N and depends on the current state $x(k)$. The variables $y(h)$, $h \in [0,N]$ represent the predicted values of system states $x(k+h)$, based on the nominal model and knowing $x(k)$. Sequences $(v(0), \ldots, v(N-1))$ and $(\sigma(0), \ldots, \sigma(N-1))$ are respectively called *virtual control sequence* and *virtual bus allocation sequence* because they are used to predict a possible evolution of the system, based on $x(k)$. Let $(v^*(0), \ldots, v^*(N-1))$ and $(\sigma^*(0), \ldots, \sigma^*(N-1))$ be the optimal solutions of this problem. Then the actual control command is obtained by setting $u(k) = v^*(0)$ and only sending the control command $u_i(k)$ whose index verifies $\delta_i(k) = 1$ where $\delta(k) = \sigma^*(0)$. At the next sampling period, the whole procedure is reiterated, based on $x(k+1)$.

Although the application of this technique gives very good resulats [14], its major drawback is that it requires very important computational resources, which makes its application to fast systems impracticable. The motivation behind the Optimal Pointer Placement (OPP) scheduling algorithm presented in this section is to be a compromise between the advantages of the on-line scheduling (control performance) and those of the off-line scheduling (a very limited usage of computing resources).

Assume that a periodic off-line controller as well as a periodic off-line schedule (both of period $\mathscr{P}$) guaranteeing the asymptotic stability of the system exist. The periodic controller is defined by a periodic sequence of state feedback control gains (corresponding to the global system) $K^{\mathscr{P}} = (K(0), \ldots, K(\mathscr{P}-1))$ and the schedule by the periodic communication sequence $\gamma^{\mathscr{P}} = (\gamma(0), \ldots, \gamma(\mathscr{P}-1))$. Note that optimal $K^{\mathscr{P}}$ and $\gamma^{\mathscr{P}}$ selection, considering worst case initial conditions, is described in [9].

At runtime, the execution of the periodic off-line controller and schedule can be described using the notion of pointer. The pointer can be seen as a variable which contains the index of the control gain to use and the actuator to update. The pointer is incremented at each slot. If it reaches the end of the sequence, its position is reset. More formally, if the pointer is started at the position p ($0 \leq p < \mathscr{P}$) its expression $I_p(k)$ at the k^{th} slot is :

$$I_p(k) = (k+p) \ mod \ \mathscr{P} \tag{15}$$

The idea behind the OPP scheduling heuristic is that instead of finding an optimal solution to problem (14), the search is restricted to a sub-optimal solution, based on the off-line schedule, according to the following problem:

$$\begin{cases} \min\limits_{p} J(\tilde{x}(k), p) = y^T(N) S y(N) + \sum\limits_{h=0}^{N-1} \begin{bmatrix} y(h) \\ v(h) \end{bmatrix}^T Q \begin{bmatrix} y(h) \\ v(h) \end{bmatrix} \\ \text{subject to:} \\ y(0) = x(k) \\ v(-1) = u(k-1) \\ \text{and for all } h \in \{0, \cdots, N-1\} \\ v(h) = -\Gamma(I_p(h)) K(I_p(h)) y(h) + (I_m - \Gamma(I_p(h)) v(h-1) \\ y(h+1) = Ay(h) + Bv(h) \end{cases} \tag{16}$$

where $\Gamma(l) = Diag\left(\gamma_1(l), \ldots, \gamma_m(l)\right)$, I_m is the $m \times m$ identity matrix and $\tilde{x}(k) = \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix}$. The horizon N is constrained to be a multiple of $\mathscr{P}$. The cost function is calculated according to a prediction of the future evolution of the system (described by $y(h)$). This evolution is calculated assuming that sequences $K^{\mathscr{P}}$ and $\gamma^{\mathscr{P}}$ are started from the position p. The solution of problem (16) is the pointer's position p^* which minimizes the cost function $J(\tilde{x}(k), p)$. The control command $u_i(k) = v_i(0)$ is sent to the actuator verifying $\gamma_i(p^*) = 1$.

When the horizon N is infinite, a formal proof of the stability of the OPP scheduling algorithm can be given. This result is stated in the following theorem:

Theorem 1: If N is infinite, and if the asymptotic stability of system (5)(6)(7) is guarantied by the off-line control and scheduling using the control gains sequence $K^{\mathscr{P}}$ and the network scheduling sequence $\gamma^{\mathscr{P}}$, than it is also ensured by the Optimal Pointer Placement scheduling algorithm.

Proof: Let $x(0)$ be an arbitrary initial condition and p_0 an arbitrary initial pointer position for the static scheduling algorithm. Let x^{ss} and u^{ss} (respectively x^{opp} and u^{opp}) be the state trajectory and the control of the system when scheduled using the static scheduling algorithm (respectively the OPP algorithm). Let $J^{ss}(\tilde{x}(i), i, f, p) = \sum\limits_{h=i}^{f} \begin{bmatrix} x^{ss}(h) \\ u^{ss}(h) \end{bmatrix}^T Q \begin{bmatrix} x^{ss}(h) \\ u^{ss}(h) \end{bmatrix}$ be be the cost function corresponding to an evolution from instant $h = i$ to instant $h = f$ starting from the extended state $\tilde{x}(i) = \begin{bmatrix} x(i) \\ u(i-1) \end{bmatrix}$ where the static scheduling algorithm is applied and where the pointer at instant i is placed at position p. Let $J^{opp}(\tilde{x}(i), i, f) = \sum\limits_{h=i}^{f} \begin{bmatrix} x^{opp}(h) \\ u^{opp}(h) \end{bmatrix}^T Q \begin{bmatrix} x^{opp}(h) \\ u^{opp}(h) \end{bmatrix}$ be the cost function corresponding to an evolution from instant $k = i$ to instant $k = f$ starting from the state $\tilde{x}(i)$ where the OPP algorithm is applied. According to the OPP strategy, the pointer position $p^*(l)$ at instant l is chosen such that:

$$p^*(l) = \underset{p}{\operatorname{argmin}} J^{ss}(\tilde{x}^{opp}(l), l, +\infty, p)$$

For $l = 0$, $\min_p J^{ss}(\tilde{x}(0), 0, +\infty, p) \leq J^{ss}(\tilde{x}(0), 0, +\infty, p_0)$ and at each step $l > 1$, the relation $J^{opp}(\tilde{x}(0), 0, l-1) + \min_p J^{ss}(\tilde{x}^{opp}(l), l, +\infty, p) \leq J^{ss}(\tilde{x}(0), 0, +\infty, p_0)$ holds. When $l \to +\infty$, the last relation reduces to:

$$J^{opp}(\tilde{x}(0), 0, +\infty) \leq J^{ss}(\tilde{x}(0), 0, +\infty, p_0) \tag{17}$$

Q is definite positive. Consequently, $J^{ss}(\tilde{x}(0), 0, +\infty, p_0)$ (respectively $J^{opp}(\tilde{x}(0), 0, +\infty)$) is finite if and only if the system is asymptotically stable. Knowing the asymptotic stability of the system scheduled using the static scheduling algorithm and using relation (17), the theorem is proved. ■

Example 2: In order to illustrate the effectiveness of the OPP scheduling algorithm, example 1 of section III is reconsidered. The control performance corresponding to the use of a static scheduling (SS) algorithm, an OPP algorithm and the MPC algorithm is compared. Static scheduling and OPP algorithms use the communication sequence:

$$\gamma^6 = \left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right)$$

which together with the control gains sequence $K^6 = (K_1, K_2, K_1, K_2, K_1, K_3)$ guarantees the asymptotic stability of the global system, where:

$$K_1 = (-1.14 \quad -1.04 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0)$$

$$K_2 = (0 \quad 0 \quad 4.12 \quad 0.36 \quad 0 \quad 0 \quad 0 \quad 0)$$

$$K_3 = (0 \quad 0 \quad 0 \quad 0 \quad 17.80 \quad -6.28 \quad -33.28 \quad 48.91)$$

The period $\mathscr{P}$ of the schedule is equal to 6 and the horizon N of the OPP and MPC algorithms is equal to 60. A sub-optimal solution with a relative error of 1×10^{-5} was required for the MPC algorithm. System responses corresponding to state variables x_1, x_3, x_5 and x_6 (the positions) are depicted in figures 4 and 5. The continuous-time cost functions corresponding to these responses are illustrated in figure 6.

The global system is started from the initial state $[1 \ \ 0 \ \ 1 \ \ 0 \ \ 1 \ \ 0 \ \ 0 \ \ 0]^T$. The three systems S^1, S^2 and S^3 converge progressively to the steady state. At $t = 0.14$ s, the system S^1 is disturbed. This deviation is quickly corrected. The best response is achieved by the MPC. However, this algorithm cannot be implemented in practice, because the required computation time is too long (a few seconds at each step in this example). For $N = 60$, the number of feasible communication sequences is $3^{60} = 4.23 \times 10^{28}$. The fact that the used branch and bound algorithm finishes in few seconds shows its efficiency to address this particular problem. The OPP algorithm significantly improves the control performance with respect to the static scheduling algorithm, requiring fewer computing resources than the MPC. In fact, using the OPP scheduling algorithm, the maximum number of possible communication sequences is equal to $\mathscr{P} = 6$.

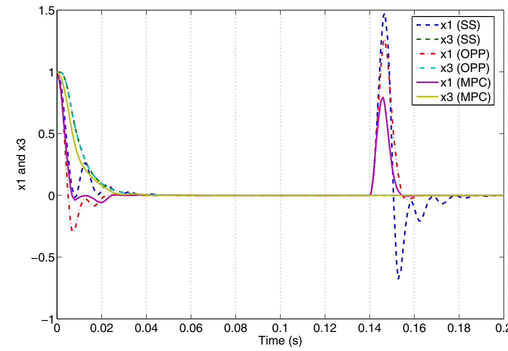

Fig. 4. System responses - states x_1 and x_3

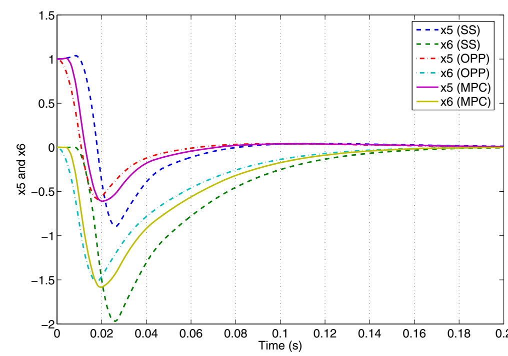

Fig. 5. System responses - states x_5 and x_6

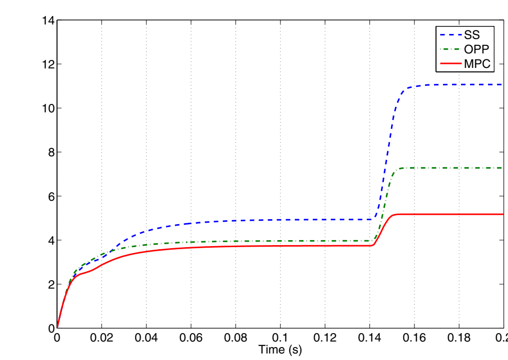

Fig. 6. Continuous time cost functions

The schedule obtained by the OPP and MPC algorithms are respectively depicted in figures 7 and 8. At $t = 0$ s, the OPP scheduling algorithm chooses to reserve the first slot to system S^3. This choice contributes to the improvement of its performance and that of the global system (shown in figure 5). MPC and OPP have the ability to change on-line the sampling period of each system. MPC can compensate for these changes. Consequently, the scheduling pattern is irregular. Network slots are allocated in order to improve the control performance. When a system is closer to the

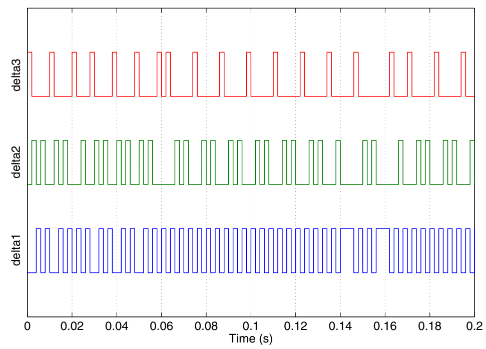

Fig. 7. OPP schedule

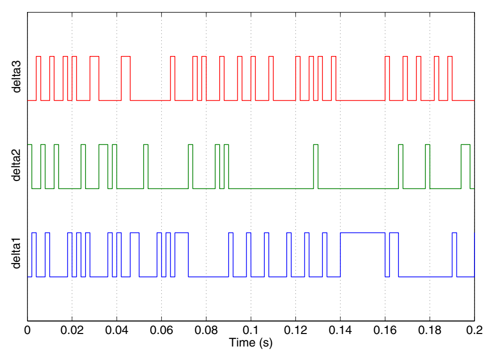

Fig. 8. MPC schedule

equilibrium, then its control commands remains constant and they may be sent less frequently over the network. That's why the MPC algorithm does not allocate network slots to systems in the equilibrium, when other systems are "far" form the steady state. At $t = 0.14$ s, when the system S^1 is disturbed, OPP and MPC algorithms allocates the network slots mainly for the transmission of the control command of system S^1, allowing to react earlier and quicker to the disturbance. This contrasts with the static scheduling algorithm, where the allocation of the network resources is independent from the dynamical state of the systems.

Finally, the three systems are disturbed with a band limited white noise characterized by a noise power of 0.1 and a correlation time of 1×10^{-5}. Simulation results are depicted in figure 9. Performance improvements using the OPP and the MPC algorithms are similar to those observed in the previous simulations.

V. Conclusions

In this paper, the problem of the integrated optimal control and scheduling was addressed theoretically. A model of a control system and its limited communication network was described, allowing control and scheduling to be tightly coupled in order to improve the control performance. A new on-line scheduling algorithm for networked control systems was introduced. Stability conditions for the systems scheduled using this algorithm were stated. Simulation results

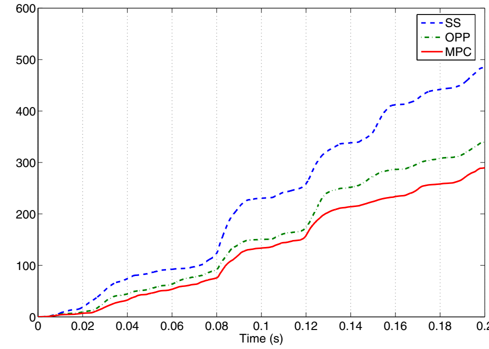

Fig. 9. Cumulative continuous time cost functions

show that performance improvements are significant and approach those obtained using the model predictive control algorithm, which can be seen as an optimal on-line control and scheduling algorithm. These improvements are due to a more efficient use of the available network resources, which are dynamically allocated as a function of the states variables of the different systems.

References

[1] E. Sozer, M. Stojanovic, and J. Proakis, "Underwater acoustic networks," *IEEE Journal of Oceanic Engineering*, vol. 25, no. 1, pp. 72–83, January 2000.

[2] L. Xiangheng and A. Goldsmith, "Wireless medium access control in networked control systems," in *Proceedings of the 23rd American Control Conference*, Boston, USA, June-July 2004.

[3] G. Walsh and H. Ye, "Scheduling of networked control systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 57–65, February 2001.

[4] M. Branicky, S. Phillips, and W. Zhang, "Scheduling and feedback co-design for networked control systems," in *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, USA, December 2002.

[5] J. P. Hespanha and Y. Xu, "Optimal communication logics for networked control systems," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 2004.

[6] D. Hristu and K. Morgansen, "Limited communication control," *System and Control Letters*, vol. 37, no. 4, pp. 193–205, July 1999.

[7] A. Cervin, J. Eker, B. Bernhardsson, and K.-E. Årzén, "Feedback-feedforward scheduling of control tasks," *Real-Time Systems*, vol. 23, no. 1, pp. 25–53, July 2002.

[8] D. Simon, D. Robert, and O. Sename, "Robust control/scheduling co-design: application to robot control," in *Proceedings of the 11th IEEE Real-Time and Embedded Technology and Applications Symposium*, San Francisco, USA, March 2005.

[9] H. Rehbinder and M. Sanfridson, "Scheduling of a limited communication channel for optimal control," *Automatica*, vol. 40, no. 3, pp. 491–500, March 2004.

[10] R. Brockett, "Stabilization of motor networks," in *Proceedings of the 34th IEEE Conference on Decision and Control*, New Orleans, Los Angeles, USA, December 1995.

[11] L. Palopoli, A. Bicchi, and A. Sangiovanni Vincentelli, "Numerically efficient control of systems with communication constraints," in *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, USA, December 2002.

[12] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[13] L. Lu, L. Xie, and M. Fu, "Optimal control of networked systems with limited communication: a combined heuristic and convex optimization approach," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Hawaii, USA, December 2003.

[14] M.-M. Ben Gaid and A. Çela, "Model predictive control of systems with communication constraints," in *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, July 2005.