



HAL
open science

Optimal Integrated Control and Scheduling of Networked Control Systems with Communication Constraints: Application to a Car Suspension System

Mohamed El Mongi Ben Gaid, Arben Cela, Yskandar Hamam

► **To cite this version:**

Mohamed El Mongi Ben Gaid, Arben Cela, Yskandar Hamam. Optimal Integrated Control and Scheduling of Networked Control Systems with Communication Constraints: Application to a Car Suspension System. IEEE Transactions on Control Systems Technology, 2006, 14 (4), pp.776 - 787. 10.1109/TCST.2006.872504 . inria-00466829

HAL Id: inria-00466829

<https://inria.hal.science/inria-00466829>

Submitted on 24 Mar 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimal Integrated Control and Scheduling of Networked Control Systems with Communication Constraints: Application to a Car Suspension System

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 deterministic 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, a computationally efficient on-line scheduling algorithm, which can be seen as a compromise, is presented and its performance is evaluated. Finally, this algorithm, called Optimal Pointer Placement (OPP) scheduling algorithm, is applied to the control and scheduling of a car suspension system.

Index Terms—Networked Control Systems, Limited Communication Control, Hybrid Systems, Control and Scheduling Co-design, Real-Time Scheduling, Active Suspension Control

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 approaches were proposed to address the problems stemming from the use of communication networks in control loops. Recently, it was shown that considering jointly control and communication/computation resource allocation leads to an improvement of the control performance with respect to the classical approaches, given the same resources ([3]–[11]).

The experimental study of communication networks characteristics was performed in [12]. It was shown that the transmission time of a message in the most used networks can be neglected and that the delays occurring in networked control loops are mainly due to the contention between the different messages sent by the nodes of the network. The most efficient way of reduction of these delays is through the design of appropriate message scheduling strategies.

Scheduling algorithms for networked control applications can be classified according to the scheduling policy, which may be either off-line or on-line. In off-line scheduling algorithms, the order of the different messages is specified at design time. In on-line scheduling algorithms, the access to the shared resource (the network) is determined at runtime, based on the information related to the priorities of the messages or to the controlled system state. Scheduling algorithms can also be designed to tackle the scheduling of sensor measures that are sent to the controller ([6], [13]) or to address the scheduling of control commands to the actuators ([7]–[9]).

The problem of the optimal control and off-line scheduling was studied in [7]. In the proposed model, the commands are sent to the actuators through a shared TDMA bus. At each slot, only one control command can be sent, the remaining commands for the other actuators are held constant. The choice of which actuator to update at each slot is handled using the notion of communication sequence [14]. Only periodic communication sequences were considered. Control commands and periodic communication sequences are obtained through the solving of a complex combinatorial optimization problem, which aims at optimizing a quadratic cost function. This problem can be treated using the approach developed in [15], which has the advantage of applying the dynamic programming method leading to a more efficient search heuristics.

On-line scheduling of control commands to the actuators

This work was supported by the COSI Laboratory, ESIEE Paris.

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

Y. Hamam is with the A2SI Laboratory, ESIEE Paris, Cité Descartes, BP 99, 93162 Noisy-Le-Grand Cedex, FRANCE and with the LIRIS Laboratory, UVSQ, 10/12 Avenue de l'Europe, 78140 Velizy, FRANCE (e-mail: hamamy@esiee.fr)

was studied in [8]. In the proposed model, it is assumed that every slot, only one command vector can be sent to an actuator group, the other control vectors are set to zero. The stabilization is achieved using a Model Predictive Controller, which calculates on-line the appropriate control law and the allocation of the shared bus. The cost function used by the MPC calculates a weighted sum of the infinity norms of the states and the control commands over a specified horizon. The optimization problem solved at each step by the MPC algorithm was proven to be equivalent to the Generalized Linear Complementarity Problem [16]. However, setting control commands to zero can degrade the dynamics of the controlled continuous time systems, that's why this approach is mainly dedicated to discrete-time systems, rather than sampled-data systems. The receding horizon approach was also addressed in [9]. The control commands were assumed to belong to a finite set of values. The solving of the optimization problem involves the search of the Nearest Neighbor Vector Quantizer, which may be very complex as the size of the finite set of control values increases.

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. A model predictive controller based on this problem formulation is studied and evaluated. In opposition to [8], control signals that could not be updated are held constant, a quadratic cost function is used to evaluate the control performance and the ability of the adaptive scheduling to improve the performance of sampled-data systems is demonstrated. The on-line solving of the optimization algorithm is very costly, that's why a computationally efficient on-line scheduling algorithm, which can be seen as a compromise between off-line and on-line scheduling algorithms, is designed and applied to a typical example of a distributed control system: the active suspension of car.

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 solved. In section IV, a Model Predictive Controller based on the formulated optimization problem is designed. In section V, a computationally efficient algorithm (the Optimal Pointer Placement (OPP) scheduling algorithm) is proposed and compared to the Model Predictive Controller. Finally, in section VI, the OPP algorithm is applied to an active suspension system.

II. PROBLEM FORMULATION

Consider the continuous-time LTI plant described by:

$$\dot{x}_c(t) = A_c x_c(t) + B_c u_c(t) \quad (1)$$

where $x_c(t) \in \mathbb{R}^n$ and $u_c(t) \in \mathbb{R}^m$. The plant contains m distinct actuators which are spatially distributed. The actuators are connected to the main controller through a limited bandwidth communication network (figure 1).

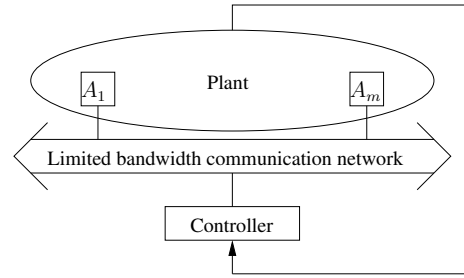


Fig. 1. A networked control system with communication constraints affecting the transmission of control commands to the actuators

In order to derive a digital control law, a discrete-time representation of system (1) at the sampling period T_s is considered:

$$x(k+1) = Ax(k) + Bu(k) \quad (2)$$

where $x(k) = x_c(kT_s)$ and $u(k) = u_c(kT_s)$.

In this article, it is assumed that the pair (A, B) is reachable and that the full state vector $x(k)$ is available to the controller at each sampling period.

The communication resource is limited in the sense that, at each sampling interval, it can carry at most b control commands, where $b \leq m$ [4]. In contrast to [7], [9], where only a control signal can be updated at each slot time (which is equal to the sampling period), the adopted modeling allows to specify separately the temporal parameters that are related to the dynamics of the control system (the sampling period) and those corresponding to the network capacity (the network bandwidth), and thus permits to achieve a maximal use of the available network resources.

The description of the constraints affecting the transmission of the control commands to the actuators can be performed using the notion of *scheduling function*. The scheduling function $\delta(k)$ at the sampling period k is defined by:

$$\begin{cases} \delta_i(k) = 1 & \text{if } u_i(k) \text{ is updated at instant } k \\ \delta_i(k) = 0 & \text{otherwise} \end{cases}$$

Bandwidth limitations can be specified by the following equality:

$$\sum_{i=1}^m \delta_i(k) = b \quad (3)$$

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} sampling period, then it is held constant. This assertion can be modeled by the logic formula:

$$\delta_i(k) = 0 \implies u_i(k) = u_i(k-1) \quad (4)$$

Let $v(k) \in \mathbb{R}^b$ be the vector of control commands that are sent to the actuators through the network at the k^{th} sampling period and $u^f(k)$ the vector containing the b "free" elements of $u(k)$ (i.e. the elements of $u(k)$ whose indices i satisfy $\delta_i(k) = 1$), arranged according to the increasing order of their indices.

The elements of $v(k)$ are mapped to the b free elements of $u(k)$ such that:

$$u^f(k) = v(k)$$

This mapping, together with constraint (4), can be described by:

$$\begin{cases} u_i(k) = v_j(k) & \text{if } \delta_i(k) = 1 \text{ and } \sum_{l=1}^i \delta_l(k) = j \\ u_i(k) = u_i(k-1) & \text{otherwise} \end{cases} \quad (5)$$

It can be easily verified that if v_{j_1} and v_{j_2} are mapped to respectively u_{i_1} and u_{i_2} , than $(j_1 < j_2)$ implies that $(i_1 < i_2)$.

The mapping (5) can be written in matrix form. Let $[D_\delta(k)]_{1 \leq i \leq m, 1 \leq j \leq b}$ the matrix defined by:

$$\begin{cases} [D_\delta(k)]_{ij} = 1 & \text{if } \delta_i(k) = 1 \text{ and } \sum_{l=1}^i \delta_l(k) = j \\ [D_\delta(k)]_{ij} = 0 & \text{otherwise} \end{cases}$$

and:

$$E_\delta(k) = \text{Diag}(1 - \delta_1(k), \dots, 1 - \delta_m(k))$$

then:

$$u(k) = D_\delta(k)v(k) + E_\delta(k)u(k-1)$$

Conversely, knowing the control input $u(k)$ and the scheduling decision $\delta(k)$, the vector of control commands $v(k)$ that were sent through the network can be determined. Let $[M_\delta(k)]_{1 \leq i \leq b, 1 \leq j \leq m}$ the matrix defined by:

$$\begin{cases} [M_\delta(k)]_{ij} = 1 & \text{if } \delta_j(k) = 1 \text{ and } \sum_{l=1}^j \delta_l(k) = i \\ [M_\delta(k)]_{ij} = 0 & \text{otherwise} \end{cases}$$

then:

$$v(k) = M_\delta(k)u(k) \quad (6)$$

The control system and its communication network can be seen as a hybrid system \mathcal{S} , having two types of inputs: continuous inputs $v(k)$ (control commands transmitted through the network) and logical inputs $\delta(k)$ (scheduling decisions) (figure 2). The considered model of system \mathcal{S} is composed of:

- A recurrent equation (2) describing the dynamics of the system.
- An equality constraint (3) expressing the limitations of the communication medium.
- Logic rules (5) modeling the zero-order holders and addressing the mapping of control signals v_i that are sent through the network to control system inputs u_i according to the scheduling decisions δ_i .

This model can be handled using the Mixed Logical Dynamical (MLD) framework [17].

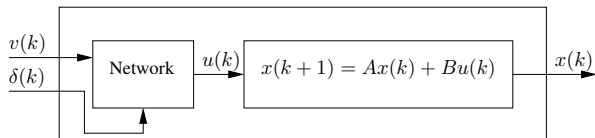


Fig. 2. Hybrid model of the networked control system \mathcal{S}

III. FINITE-TIME OPTIMAL CONTROL AND SCHEDULING

In this paragraph, the problem of the finite-time optimal control and scheduling is formulated and translated into the Mixed Integer Quadratic Programming (MIQP) formulation. It is assumed that $u(k) = 0$ and $v(k) = 0$ for $k < 0$, and that control commands $u(k)$ and $v(k)$ are subject to saturation constraints:

$$L_i \leq u_i(k) \leq U_i \quad \text{and} \quad L_i \leq v_i(k) \leq U_i$$

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

A. Performance index definition

In order to quantify the ‘‘Quality’’ of the control and scheduling, a quadratic cost function is associated to system (1) :

$$J_c(x_c, u_c, 0, T_f) = \int_0^{T_f} (x_c^T(t)Q_c x_c(t) + u_c^T(t)R_c u_c(t)) dt + x_c^T(T_f)S_c x_c(T_f) \quad (7)$$

where $T_f = NT_s$ and Q_c , R_c and S_c are positive definite matrices. These matrices define the design specifications of the ideal controller. The sampled-data representation of the cost function $J_c(x_c, u_c, 0, T_f)$ at the sampling period T_s is:

$$J(x, u, 0, N) = \sum_{k=0}^{N-1} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}^T \begin{bmatrix} Q_1 & Q_{12} \\ Q_{12}^T & Q_2 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} + x^T(N)Q_0 x(N) \quad (8)$$

The expressions of Q_1 , Q_2 , Q_{12} and Q_0 can be found in ([18], pp. 411–412). Note that this representation does not involve any approximation and is exact. In the following, it is assumed that Q , Q_2 and Q_0 are positive definite matrices, where:

$$Q = \begin{bmatrix} Q_1 & Q_{12} \\ Q_{12}^T & Q_2 \end{bmatrix}$$

B. Formalization and solving of the finite-time optimal integrated control and scheduling problem

The finite-time optimal control and scheduling problem can be formalized as follows:

Problem formulation 1: Given an initial state $x(0)$ and a final time N , find the optimal control sequence $v^{N-1} = (v(0), \dots, v(N-1))$ and the optimal scheduling sequence $\delta^{N-1} = (\delta(0), \dots, \delta(N-1))$ which minimize the performance index:

$$J(x, u, 0, N) = \sum_{k=0}^{N-1} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}^T Q \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} + x^T(N)Q_0 x(N)$$

subject to:

$$x(k+1) = Ax(k) + Bu(k)$$

$$\sum_{i=1}^m \delta_i(k) = b$$

$$u(k) = D_\delta(k)v(k) + E_\delta(k)u(k-1)$$

$$L_i \leq v_i(k) \leq U_i$$

System \mathcal{S} is time varying and the problem of finding the optimal control sequence v^{N-1} for a given fixed scheduling sequence δ^{N-1} is a quadratic programming (QP) problem. The number of possible scheduling sequences is finite. The resolution of problem 1 is reduced to the exploration of all the feasible scheduling sequences and the solving of a QP problem for each fixed scheduling sequence. However, in practice, the number of feasible scheduling sequences grows exponentially with N , which means that exhaustive search cannot be applied to problems with large values of N .

The solution of problem 1 can be obtained through the solving of a simpler optimization problem, which can be seen as a constrained control problem, where the variables v^{N-1} are eliminated and the constraint (5) is replaced by (4). Let $u^{N-1} = (u(0), \dots, u(N-1))$, this problem can be stated as follows:

Problem formulation 2: Given an initial state $x(0)$ and a final time N , find the optimal control sequence u^{N-1} and the optimal scheduling sequence δ^{N-1} which minimize the performance index:

$$J(x, u, 0, N) = \sum_{k=0}^{N-1} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}^T Q \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} + x^T(N)Q_0x(N)$$

subject to:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ \sum_{i=1}^m \delta_i(k) &= b \\ \delta_i(k) = 0 &\implies u_i(k) = u_i(k-1) \\ L_i &\leq v_i(k) \leq U_i \end{aligned}$$

In order to solve this problem, it is necessary to translate the logical formula (4) into linear inequalities. The connective “ \implies ” can be eliminated if (4) is rewritten in the equivalent form:

$$u_i(k) - u_i(k-1) = \delta_i(k)u_i(k) - \delta_i(k)u_i(k-1) \quad (10)$$

However, equation (10) contains terms which are the product of logical variables and continuous variables. The use of the procedure described in [17] allows the translation of this product into an equivalent conjunction of linear inequalities. For example, let:

$$z_i(k) = \delta_i(k)u_i(k) \quad (11)$$

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

$$\begin{aligned} z_i(k) &\leq U_i\delta_i(k) \\ z_i(k) &\geq L_i\delta_i(k) \\ z_i(k) &\leq u_i(k) - L_i(1 - \delta_i(k)) \\ z_i(k) &\geq u_i(k) - U_i(1 - \delta_i(k)) \end{aligned} \quad (12)$$

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

$$\text{Let } \Delta = \begin{bmatrix} \delta(0) \\ \vdots \\ \delta(N-1) \end{bmatrix}, U = \begin{bmatrix} u(0) \\ \vdots \\ u(N-1) \end{bmatrix}, X = \begin{bmatrix} x(0) \\ \vdots \\ x(N) \end{bmatrix},$$

$$Z = \begin{bmatrix} z(0) \\ \vdots \\ z(N-1) \end{bmatrix}, W = \begin{bmatrix} w(0) \\ \vdots \\ w(N-1) \end{bmatrix} \text{ and } \mathcal{V} = \begin{bmatrix} \Delta \\ U \\ X \\ Z \\ W \end{bmatrix},$$

then problem 2 can be written:

$$\begin{cases} \min_{\mathcal{V}} \frac{1}{2} \mathcal{V}^T H \mathcal{V} + f^T \mathcal{V} \\ \mathcal{A} \mathcal{V} \leq \mathcal{B} \end{cases} \quad (13)$$

where H , f , \mathcal{A} , and \mathcal{B} can be easily deduced from 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.

Problem 1 is identical to problem 2 augmented with the additional constraint: $v(k) = u^f(k)$. As a consequence, the optimal solutions of problem 1 can be deduced from the optimal solutions of problem 2 using the mapping (6).

IV. MODEL PREDICTIVE CONTROL

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 expensive computing resources, which make it only applicable to slow systems, like chemical processes. Model predictive control is the standard approach to control MLD systems. Its application to this particular problem was motivated by:

- The need to optimize simultaneously control actions and network scheduling, in order to achieve a better quality of control than the static network allocation schemes.
- The need for a control law that changes on-line the sampling period in order to improve the quality of control. This requires that these variations are taken into account by the control law [19].

Using Model Predictive Control, an optimal control problem is solved on-line at each sampling period. It aims at finding the optimal control values sequence $\hat{u}^{N-1} = (\hat{u}(0), \dots, \hat{u}(N-1))$ and the optimal network allocation sequence $\hat{\delta}^{N-1} = (\hat{\delta}(0), \dots, \hat{\delta}(N-1))$ which are solutions of the following optimization problem:

$$\begin{cases} \min_{\hat{u}^{N-1}, \hat{\delta}^{N-1}} \sum_{h=0}^{N-1} \begin{bmatrix} \hat{x}(h) \\ \hat{u}(h) \end{bmatrix}^T Q \begin{bmatrix} \hat{x}(h) \\ \hat{u}(h) \end{bmatrix} + \hat{x}^T(N)Q_0\hat{x}(N) \\ \text{subject to:} \\ \hat{x}(0) = x(0) \\ \hat{x}(h+1) = A\hat{x}(h) + B\hat{u}(h), h \in [0, N-1] \\ \sum_{i=1}^m \hat{\delta}_i(h) = b, h \in [0, N-1] \\ \hat{\delta}_i(0) = 0 \implies \hat{u}_i(0) = u_i(0) \\ \hat{\delta}_i(h) = 0 \implies \hat{u}_i(h) = \hat{u}_i(h-1), h \in [1, N-1] \end{cases} \quad (14)$$

The solution of this problem is based on the prediction of the future evolution of the system over a horizon of N sampling periods. This predicted evolution is calculated according to the model of the plant, knowing the current state $x(k)$ of the

system and the previously applied control input $u(k-1)$. The variables $\hat{x}(h)$, $h \in [0, N]$ represent the predicted values of system states $x(k+h)$. The sequences $(\hat{u}(0), \dots, \hat{u}(N-1))$ (virtual control sequence) and $(\hat{\delta}(0), \dots, \hat{\delta}(N-1))$ (virtual network allocation sequence) are called virtual sequences, because they are based on the predicted evolution of the system. The resolution of this problem aims at finding the optimal virtual control sequence $(\hat{u}^*(0), \dots, \hat{u}^*(N-1))$ and the optimal virtual network allocation $(\hat{\delta}^*(0), \dots, \hat{\delta}^*(N-1))$ which minimize a quadratic cost function over a finite horizon of N sampling periods. Assuming that the optimal virtual sequences exist, the actual control commands are obtained by setting:

$$v(k) = M_{\hat{\delta}^*}(0)\hat{u}^*(0) \quad (15)$$

and:

$$\delta(k) = \hat{\delta}^*(0) \quad (16)$$

and disregarding the remaining elements of the sequences $(\hat{u}^*(1), \dots, \hat{u}^*(N-1))$ and $(\hat{\delta}^*(1), \dots, \hat{\delta}^*(N-1))$. At the next sampling period (step $k+1$), the whole optimization procedure is repeated, based on $x(k+1)$.

A important issue concerns the stability of the proposed Model Predictive Controller. If the following constraint is added to problem (14):

$$\hat{x}(N) = 0 \quad (17)$$

the following result is obtained:

Theorem 1: If at $k = 0$, a feasible solution exists for the problem (14) augmented with the additional constraint (17), then $\forall Q = Q^T > 0$, the MPC law (14)(17) stabilizes the system \mathcal{S} such that:

$$\lim_{k \rightarrow \infty} x(k) = x_e = 0 \text{ and } \lim_{k \rightarrow \infty} u(k) = u_e = 0$$

Proof: The proof can be easily performed following the same ideas of the proof of the sufficient stability conditions for the Model Predictive Control of MLD systems stated in [17]. ■

Although the application of this technique gives very good results (as it will be illustrated in the next section), its major drawback is that it requires very expensive computational resources, which makes its application to fast systems impracticable. A more efficient heuristics are needed.

V. OPTIMAL POINTER PLACEMENT SCHEDULING

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).

A. Algorithm description

First, a time varying state representation of system \mathcal{S} is derived. Let $\xi(k) = u(k-1)$ and $\tilde{x}(k) = \begin{bmatrix} x(k) \\ \xi(k) \end{bmatrix}$. For a fixed scheduling function δ verifying the constraint (3), the system \mathcal{S} can be represented by the time-varying state equation:

$$\tilde{x}(k+1) = \tilde{A}(k)\tilde{x}(k) + \tilde{B}(k)v(k) \quad (18)$$

where:

$$\tilde{A}(k) = \begin{bmatrix} A & BE_{\delta}(k) \\ 0 & E_{\delta}(k) \end{bmatrix} \text{ and } \tilde{B}(k) = \begin{bmatrix} BD_{\delta}(k) \\ D_{\delta}(k) \end{bmatrix}$$

Assume that a periodic off-line controller as well as a periodic off-line schedule (both of period T) guaranteeing the asymptotic stability of the system exist. The periodic controller is defined by a periodic sequence of state feedback control gains $K^{T-1} = (K(0), \dots, K(T-1))$ and the schedule by the periodic communication sequence $\gamma^{T-1} = (\gamma(0), \dots, \gamma(T-1))$. Note that optimal K^{T-1} and γ^{T-1} selection is described in [7] (considering worst case initial conditions) and in [15] (according to LQG arguments).

At runtime, the execution of the periodic off-line controller and scheduler 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 scheduling to apply. The pointer is incremented at each sampling period. If it reaches the end of the sequence, its position is reset. More formally, if the pointer is started at position p ($0 \leq p < T$) its expression $I_p(k)$ at the k^{th} sampling period is:

$$I_p(k) = (k+p) \bmod T \quad (19)$$

According to the off-line strategy, the control commands that are sent and the scheduling decisions are chosen such that:

$$v(k) = K(I_p(k))\tilde{x}(k) \text{ and } \delta(k) = \gamma(I_p(k))$$

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

$$\begin{cases} \min_p \sum_{h=0}^{N-1} \begin{bmatrix} \hat{x}(h) \\ \hat{u}(h) \end{bmatrix}^T Q \begin{bmatrix} \hat{x}(h) \\ \hat{u}(h) \end{bmatrix} + \hat{x}^T(N)Q_0\hat{x}(N) \\ \text{subject to:} \\ \hat{x}(0) = x(k) \\ \hat{u}(-1) = u(k-1) \\ \text{and for all } h \in [0, N-1] \\ \hat{v}(h) = K(I_p(h)) \begin{bmatrix} \hat{x}(h) \\ \hat{u}(h-1) \end{bmatrix} \\ \hat{u}(h) = D_{\gamma}(I_p(h))\hat{v}(h) + E_{\gamma}(I_p(h))\hat{u}(h-1) \\ \hat{x}(h+1) = A\hat{x}(h) + B\hat{u}(h) \end{cases} \quad (20)$$

The cost function is calculated according to a prediction of the future evolution of the system (described by $\hat{x}(h)$). This evolution is calculated assuming that sequences K^{T-1} and γ^{T-1} are started from position p . The solution of problem (20) is the pointer's position p^* which minimizes the cost function $\hat{J}(\tilde{x}(k), p)$, subject to the constraints expressed above, where:

$$\hat{J}(\tilde{x}(k), p) = \sum_{h=0}^{N-1} \begin{bmatrix} \hat{x}(h) \\ \hat{u}(h) \end{bmatrix}^T Q \begin{bmatrix} \hat{x}(h) \\ \hat{u}(h) \end{bmatrix} + \hat{x}^T(N)Q_0\hat{x}(N)$$

The control command $v(k) = K(p^*)\tilde{x}(k)$ is sent according to the scheduling function $\delta(k) = \gamma(p^*)$.

B. A numerical example

In order to illustrate the proposed approach and to study the interdependency between control and scheduling, especially the relationship between the state vector of the plant and the network allocation, consider the continuous-time LTI system described by the state and input matrices:

$$A_c = \begin{bmatrix} A_{c_1} & 0 \\ 0 & A_{c_2} \end{bmatrix}, B_c = \begin{bmatrix} B_{c_1} & 0 \\ 0 & B_{c_2} \end{bmatrix} \quad \text{where:}$$

$$A_{c_1} = A_{c_2} = \begin{bmatrix} 0 & 130 \\ -800 & 10 \end{bmatrix} \quad \text{and} \quad B_{c_1} = B_{c_2} = \begin{bmatrix} 0 \\ 230 \end{bmatrix}$$

The system consists of two identical and independent subsystems (S^1 and S^2), which are open-loop unstable. A scalar control input is used in order to stabilize each subsystem. The control commands are sent to the actuators through a bus characterized by a slot time of 2 ms. Design criteria of the closed loop system are defined by matrices $Q_c = S_c = \text{Diag}(100, 10, 100, 10)$ and $R_c = \text{Diag}(1, 1)$. In fact, in opposition to the ideal case (corresponding to an infinite network bandwidth), the limited bandwidth of the network introduces a coupling between the different systems. This makes the impact of scheduling decisions on control performance very significant. The control performance corresponding to the use of a static scheduling (SS) algorithm, an OPP scheduling algorithm and a model predictive controller is compared. Static scheduling and OPP algorithms use the communication sequence γ^1 which together with the controller $K^1 = (K, K)$ guarantees the asymptotic stability of the system, where:

$$\gamma^1 = \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \quad \text{and} \quad K = [-1.1688 \quad 1.0609]$$

The period T of the schedule is equal to 2 and the horizon N of the OPP and MPC algorithms is equal to 14. 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 and x_3 (i.e. systems outputs) are depicted in figures 3 and 4. The continuous-time cost functions corresponding to these responses are illustrated in figure 5.

The global system is started from the initial state $[1 \ 0 \ -0.2 \ 0]^T$. Subsystems S^1 and S^2 converge progressively to the steady state. At $t = 20$ ms, system S^1 is disturbed. This deviation is quickly corrected. The best response is achieved by the model predictive controller. However, this algorithm cannot be implemented in practice, because the required computation time is too long (hundreds of milliseconds at each step in this example). For $N = 14$, the number of feasible communication sequences is $2^{14} = 16384$. The fact that the used branch and bound algorithm finishes in less than one second 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 OPP, the maximum number of possible communication decisions is equal to $T = 2$.

The schedule obtained by the OPP and MPC algorithms are respectively depicted in figures 6 and 7. The MPC has the

ability to change on-line the sampling period of each subsystem, and to 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 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" from the steady state. At $t = 20$ ms, when the subsystem S^1 is disturbed, OPP and MPC algorithms allocates the network slots mainly for the transmission of the control command u_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 two subsystems 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 8. Performance improvements using the OPP and the MPC algorithms are similar to those observed in the previous simulations.

C. Optimal infinite horizon control given a periodic scheduling function

In a wide category of networked control applications, especially those constructed for time-triggered networks (like TT-CAN, TTP/C,...), off-line schedules are periodic. An interesting property which is induced by the periodicity is that the solution of the Riccati equation associated with an infinite horizon optimal control problem will converge to a periodic solution if the controllability of system (18) is fulfilled [7]. This observation leads to an interesting simplification of the OPP scheduling algorithm. In the following we describe how the periodic optimal controller can be obtained given a fixed periodic communication sequence, guaranteeing the controllability of the system. A method for selecting communication sequences that guarantee the controllability was proposed in [20].

Assume that the scheduling function is periodic and verifies $\delta(k+T) = \delta(k)$, where T is period of the schedule. In addition, suppose that $\delta \in \mathcal{S}^c$, where \mathcal{S}^c is the set of scheduling functions such that the system (18) is controllable. A periodic scheduling function δ can be described by the communication sequence $\delta^{T-1} = (\delta(0), \dots, \delta(T-1))$. As a consequence of the periodicity of the scheduling function, the networked control system S is periodic and the matrices \tilde{A} and \tilde{B} verify $\tilde{A}(k+T) = \tilde{A}(k)$ and $\tilde{B}(k+T) = \tilde{B}(k)$. The cost function $J(x, u, 0, N)$ can be written:

$$J(x, u, 0, N) = J(\tilde{x}, v, 0, N) = \tilde{x}^T(N) \tilde{Q}_0 \tilde{x}(N) + \sum_{k=0}^{N-1} \begin{bmatrix} \tilde{x}(k) \\ v(k) \end{bmatrix}^T \tilde{Q}(k) \begin{bmatrix} \tilde{x}(k) \\ v(k) \end{bmatrix}$$

where:

$$\tilde{Q}(k) = \begin{bmatrix} Q_1 & Q_{12}E_\delta(k) & Q_{12}D_\delta(k) \\ E_\delta^T(k)Q_{12}^T & E_\delta^T(k)Q_2E_\delta^T(k) & E_\delta^T(k)Q_2D_\delta(k) \\ D_\delta^T(k)Q_{12}^T & D_\delta^T(k)Q_2E_\delta(k) & D_\delta^T(k)Q_2D_\delta(k) \end{bmatrix}$$

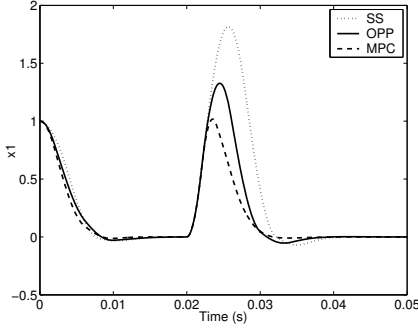
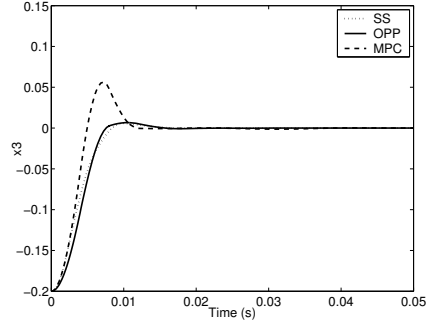
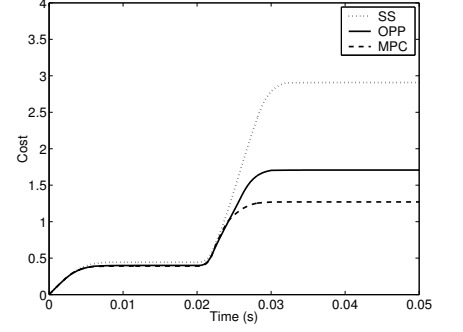
Fig. 3. System responses – state x_1 Fig. 4. System responses – state x_3 

Fig. 5. Continuous time cost functions

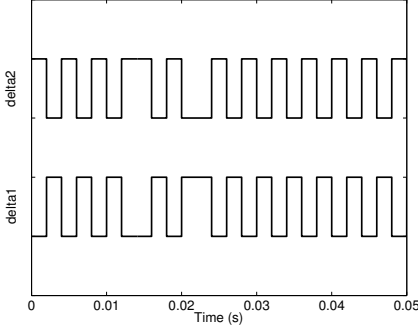


Fig. 6. OPP schedule

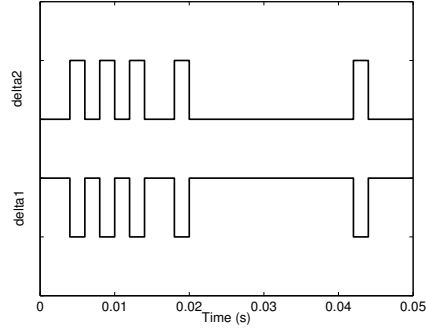


Fig. 7. MPC schedule

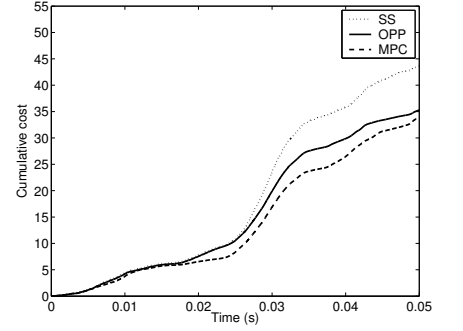


Fig. 8. Cumulative continuous time cost functions

and:

$$\tilde{Q}_0 = \begin{bmatrix} Q_0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Let τ be a discrete time instant, assume that $N = \tau + HT$ and consider the optimal control problem:

$$\begin{cases} \min_v J(\tilde{x}, v, \tau, N) = \sum_{k=\tau}^{N-1} \begin{bmatrix} \tilde{x}(k) \\ v(k) \end{bmatrix}^T \tilde{Q}(k) \begin{bmatrix} \tilde{x}(k) \\ v(k) \end{bmatrix} \\ \quad + \tilde{x}^T(N) \tilde{Q}_0 \tilde{x}(N) \\ \text{Subject to:} \\ \tilde{x}(k+1) = \tilde{A}(k) \tilde{x}(k) + \tilde{B}(k) v(k) \end{cases} \quad (21)$$

As illustrated in [21], a time invariant reformulation of the optimal control problem (21) can be obtained using the lifting technique. The time invariant reformulation can be seen as a down sampled representation of system (18) with periodicity T , having an augmented input vector. In the following, the formulation of the time invariant representation is described.

Let Φ be the transition matrix associated with the state matrix \tilde{A} . Φ is defined by:

$$\begin{cases} \Phi(r, s) = \tilde{A}(r-1) \tilde{A}(r-2) \cdots \tilde{A}(s) & \text{if } r > s \\ \Phi(r, r) = I \end{cases}$$

and Γ the matrix defined by:

$$\begin{cases} \Gamma(r, s) = [\Phi(r, s+1) \tilde{B}(s) \quad \Phi(r, s+2) \tilde{B}(s+1) \quad \cdots \\ \Phi(r, r) \tilde{B}(r-1) \quad \underbrace{0 \quad \cdots \quad 0}_{T-r-s}] & \text{for } s < r \leq s+T \\ \Gamma(s, s) = [0 \quad \cdots \quad 0] \end{cases}$$

$$\text{Let } \bar{x}_\tau(q) = \tilde{x}(\tau + qT) \text{ and } \bar{v}_\tau(q) = \begin{bmatrix} v(\tau + qT) \\ \vdots \\ v(\tau + (q+1)T - 1) \end{bmatrix},$$

then for $0 \leq i \leq T$:

$$\tilde{x}(\tau + qT + i) = \Phi(\tau + i, \tau) \bar{x}_\tau(q) + \Gamma(\tau + i, \tau) \bar{v}_\tau(q)$$

In particular, let $\bar{A}_\tau = \Phi(\tau + T, \tau)$ and $\bar{B}_\tau = \Gamma(\tau + T, \tau)$, then the following relation is obtained:

$$\bar{x}_\tau(q+1) = \bar{A}_\tau \bar{x}_\tau(q) + \bar{B}_\tau \bar{v}_\tau(q)$$

Let $\Lambda(i)$ the matrix defined for $0 \leq i < T$ by:

$$\Lambda(i) = \underbrace{[0 \quad \cdots \quad 0]}_i I_b \underbrace{[\cdots \quad 0]}_{T-i-1}$$

then the cost function can be written:

$$\begin{aligned} J(\tilde{x}, v, \tau, N) &= J(\bar{x}_\tau, \bar{v}_\tau, 0, H) = \bar{x}_\tau^T(H) (\bar{Q}_\tau)_0 \bar{x}_\tau(H) \\ &\quad + \sum_{q=0}^{H-1} \begin{bmatrix} \bar{x}_\tau(q) \\ \bar{v}_\tau(q) \end{bmatrix}^T \bar{Q}_\tau \begin{bmatrix} \bar{x}_\tau(q) \\ \bar{v}_\tau(q) \end{bmatrix} \end{aligned}$$

where:

$$\begin{aligned} \bar{Q}_\tau &= \sum_{i=0}^{T-1} F^T(i) \tilde{Q}(\tau + i) F(i) \\ F(i) &= \begin{bmatrix} \Phi(\tau + i, \tau) & \Gamma(\tau + i, \tau) \\ 0 & \Lambda(i) \end{bmatrix} \end{aligned}$$

and $(\bar{Q}\tau)_0 = \bar{Q}_0$. Finally, the following optimal control problem is obtained:

$$\begin{cases} \min_{\bar{v}_\tau} J(\bar{x}_\tau, \bar{v}_\tau, 0, H) = \bar{x}_\tau^T(H)(\bar{Q}\tau)_0\bar{x}_\tau(H) \\ \quad + \sum_{q=0}^{H-1} \begin{bmatrix} \bar{x}_\tau(q) \\ \bar{v}_\tau(q) \end{bmatrix}^T \bar{Q}_\tau \begin{bmatrix} \bar{x}_\tau(q) \\ \bar{v}_\tau(q) \end{bmatrix} \\ \text{subject to:} \\ \bar{x}_\tau(q+1) = \bar{A}_\tau\bar{x}_\tau(q) + \bar{B}_\tau\bar{v}_\tau(q) \end{cases} \quad (22)$$

Let \bar{S}_τ the solution of the Riccati equation associated with problem (22). Optimal control problems (21) and (22) are coincident. Consequently, as pointed in [21], the performance indices must be the same, which implies that the solutions of the Riccati equation associated with the two problems must be equal, thus $\tilde{S}(\tau + qT) = \bar{S}_\tau(q)$, for $0 \leq q \leq H$. As a result, when $H \rightarrow +\infty$, $\bar{S}_\tau(q)$ will converge to a constant solution \bar{S}_τ , consequently, $\tilde{S}(k)$ will converge to a periodic solution. This periodic solution can be obtained by solving the algebraic Riccati equation associated with problem (22) when $H \rightarrow +\infty$. The optimal periodic control gains can be described by the sequence $(\tilde{K}(0), \dots, \tilde{K}(T-1))$.

D. OPP scheduling over an infinite horizon

If the horizon N is infinite, the OPP scheduling algorithm presents interesting properties : its implementation becomes simpler and a formal proof of its stability can be given. Moreover, it can also be proven that OPP always improves the performance with respect to the static scheduling algorithm. Knowing that $\hat{J}(\tilde{x}(k), p) = \tilde{x}^T(k)\tilde{S}(p)\tilde{x}(k)$, the implementation of the OPP scheduling algorithm is simpler and can be described as follows:

$$\begin{aligned} \text{Find } p^* &= \underset{p}{\operatorname{argmin}} \tilde{x}^T(k)\tilde{S}(p)\tilde{x}(k) \\ v(k) &= \tilde{K}(p^*)\tilde{x}(k) \text{ and } \delta(k) = \gamma(p^*) \end{aligned}$$

For $N = +\infty$, the stability of the OPP scheduling algorithm is stated in the following theorem:

Theorem 2: If the asymptotic stability of system \mathcal{S} is guaranteed by the off-line control and scheduling using the control gains sequence \tilde{K}^{T-1} and the network scheduling sequence γ^{T-1} , then it is also ensured by the Optimal Pointer Placement scheduling algorithm.

Proof: The proof is based on the comparison of the trajectory of the system scheduled by the static scheduling algorithm (denoted by \tilde{x}^{ss}) and that of the system scheduled by the OPP algorithm (denoted by \tilde{x}^{opp}), starting from the same arbitrary initial condition $\tilde{x}^{ss}(0) = \tilde{x}^{opp}(0) = \tilde{x}(0)$. Let $J^{ss}(\tilde{x}(i), i, f, p)$ 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 static scheduling algorithm is applied and where the pointer at instant i is placed at position p .

$$J^{ss}(\tilde{x}(i), i, f, p) = \sum_{k=i}^f \begin{bmatrix} x^{ss}(k) \\ u^{ss}(k) \end{bmatrix}^T Q \begin{bmatrix} x^{ss}(k) \\ u^{ss}(k) \end{bmatrix}$$

Let $J^{opp}(\tilde{x}(i), i, f)$ 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.

$$J^{opp}(\tilde{x}(i), i, f) = \sum_{k=i}^f \begin{bmatrix} x^{opp}(k) \\ u^{opp}(k) \end{bmatrix}^T Q \begin{bmatrix} x^{opp}(k) \\ u^{opp}(k) \end{bmatrix}$$

In order to prove the stability of the system scheduled using the OPP algorithm, we must prove that for all $p_0 \in \{0, \dots, T-1\}$

$$J^{opp}(\tilde{x}(0), 0, +\infty) \leq J^{ss}(\tilde{x}(0), 0, +\infty, p_0) \quad (23)$$

In fact, Q is definite positive. As a consequence, $J^{ss}(\tilde{x}(0), 0, +\infty, p_0)$ (resp. $J^{opp}(\tilde{x}(0), 0, +\infty)$) is finite if and only if the system scheduled using the static scheduling algorithm (resp. the OPP algorithm) is asymptotically stable. Let $J^{opp-ss}(\tilde{x}(0), l)$ be the cost function corresponding to an evolution starting from the initial state $\tilde{x}(0)$ where OPP is applied from $k = 0$ to $k = l$ and then followed by the static scheduling algorithm (which is applied from $k = l + 1$ to $k = +\infty$). Then it is easy to see that:

$$J^{opp}(\tilde{x}(0), 0, +\infty) = \lim_{l \rightarrow +\infty} J^{opp-ss}(\tilde{x}(0), l) \quad (24)$$

Consequently, to prove (23), it is sufficient to verify that for all $p_0 \in \{0, \dots, T-1\}$ and for all $l \in [0, +\infty)$:

$$J^{opp-ss}(\tilde{x}(0), l) \leq J^{ss}(\tilde{x}(0), 0, \infty, p_0) \quad (25)$$

This proof can be done by recurrence on l . Let $p_0 \in \{0, \dots, T-1\}$ an arbitrary start position of the static scheduling algorithm.

At the stage $l = 0$, the OPP scheduling algorithm will choose the pointer position such that:

$$p^*(0) = \underset{p}{\operatorname{argmin}} J^{ss}(\tilde{x}(0), 0, +\infty, p) \quad (26)$$

Knowing that $J^{ss}(\tilde{x}(0), 0, +\infty, p^*(0)) = J^{opp-ss}(\tilde{x}(0), 0)$ implies that $J^{opp-ss}(\tilde{x}(0), 0) \leq J^{ss}(\tilde{x}(0), 0, +\infty, p_0)$. The property is then valid at stage 0.

Assume that (25) is valid at stage $l - 1$ with $l > 0$. We have to prove that (25) is also valid at stage l .

At stage l , according to the OPP strategy, the pointer position $p^*(l)$ will be chosen such that:

$$p^*(l) = \underset{p}{\operatorname{argmin}} J^{ss}(\tilde{x}^{opp}(l), l, +\infty, p) \quad (27)$$

Consequently:

$$J^{ss}(\tilde{x}^{opp}(l), l, +\infty, p^*(l)) \leq J^{ss}(\tilde{x}^{opp}(l), l, +\infty, I_{p^*(l-1)}(1)) \quad (28)$$

where $p^*(l-1)$ is the optimal pointer position at instant $l-1$. Adding $J^{opp}(\tilde{x}(0), 0, l-1)$ to both left and right terms of the previous inequality, we get:

$$J^{opp-ss}(\tilde{x}(0), l) \leq J^{opp-ss}(\tilde{x}(0), l-1) \quad (29)$$

Using the recurrence assumption:

$$J^{opp-ss}(\tilde{x}(0), l-1) \leq J^{ss}(\tilde{x}(0), 0, \infty, p_0) \quad (30)$$

and the inequality (29), the theorem is proved. \blacksquare

VI. OPTIMAL POINTER PLACEMENT SCHEDULING: APPLICATION TO A CAR SUSPENSION SYSTEM

In this section, the OPP scheduling algorithm is applied to an active suspension controller. The considered controller is based on a full-vehicle model and is implemented on a central processor. The controller sends the control commands to four hydraulic actuators located on the vehicle's corners through a bus subject to bandwidth limitations. In the following, the considered active suspension model is described, the control design methodology is illustrated and finally the OPP scheduling strategy is evaluated and compared to a fair static scheduling strategy.

A. The suspension control system

The simulated model (figure 9) was adopted from [22]. It consists of a seven degree-of-freedom system. In this model, the car body, or sprung mass, is free to heave, roll and pitch. In order to obtain a linear model, roll and pitch angles are assumed to be small. The suspension system connects the sprung mass to the four unsprung masses (front-left, front-right, rear-left and rear-right wheels), which are free to bounce vertically with respect to the sprung mass. The suspension system consists of a spring, a shock absorber and a hydraulic actuator at each corner. The shock absorbers are modeled as linear viscous dampers, and the tires are modeled as linear springs in parallel to linear dampers.

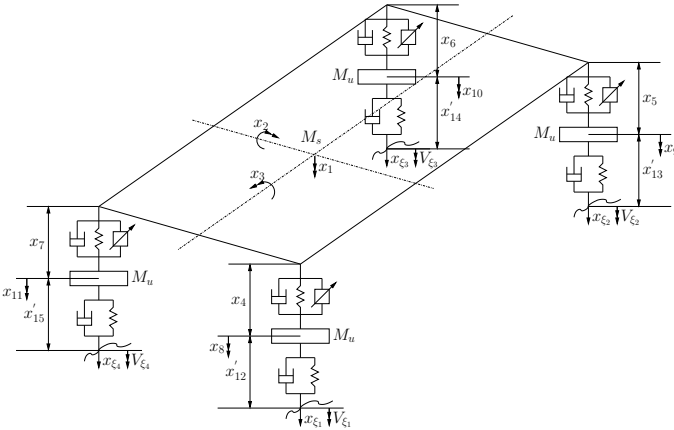


Fig. 9. Full vehicle model

In order to describe this system, fifteen variables need to be considered. These variables are the heave, pitch and roll velocities of the sprung mass (x_1, x_2, x_3), suspension deflections (x_4, x_5, x_6, x_7), unsprung mass velocities (x_8, x_9, x_{10}, x_{11}) and tire deflections (x_{12}, x_{13}, x_{14} and x_{15}). Road disturbances acting on the four wheels consist of height displacement inputs ($x_{\xi_1}, x_{\xi_2}, x_{\xi_3}, x_{\xi_4}$) and height velocity inputs ($V_{\xi_1}, V_{\xi_2}, V_{\xi_3}, V_{\xi_4}$) defined with respect to an inertial reference frame. The suspension model has seven degrees of freedom. Consequently, only fourteen state variables are needed to describe it. The extra variable can be eliminated if the wheel deflections are expressed as a function of three state variables x_{12}, x_{13} and x_{14} and of the road disturbances $x_{\xi_1}, x_{\xi_2}, x_{\xi_3}, x_{\xi_4}$ as illustrated in [22].

Applying a force-balance analysis to the model in figure 9, the state space equations can be derived from the equations of motion and are given by:

$$\dot{x}(t) = Ax(t) + Bu(t) + Ld(t) \quad (31)$$

where $x(t)$ is the state vector (14 variables), $u(t)$ is the control vector ($u(t) = [u_1(t) \ u_2(t) \ u_3(t) \ u_4(t)]^T$). u_1, u_2, u_3 and u_4 represent the control forces applied respectively by the front-left, rear-left, rear-right and front-right hydraulic actuators. $d(t)$ is the vector of road disturbances $d(t) = [x_{\xi_1}(t) \ x_{\xi_2}(t) \ x_{\xi_3}(t) \ x_{\xi_4}(t) \ V_{\xi_1}(t) \ V_{\xi_2}(t) \ V_{\xi_3}(t) \ V_{\xi_4}(t)]^T$

B. Active suspension control law

The control design for a vehicle's active suspension aims to maximize the driving comfort (as measured by sprung mass accelerations) and the safety (as measured by tire load variations) under packaging constraints (as measured by suspension deflections). However, comfort and safety are two conflicting criteria. In this paper, the control design methodology of [22] was adopted. The used controller is a linear quadratic regulator which aims at minimizing the sprung mass accelerations (heave, roll and pitch accelerations) as well as suspension deflections, unsprung mass velocities and the wrap torque acting on the sprung mass. The details of this control design method are clearly exposed in [22].

C. Simulation setup and results

The communication network connecting the controller to the actuators is subject to communication constraints: only a control command can be sent to an actuator every 10 ms. A simple approach to tackle this problem is to send the control commands alternately according to the periodic communication sequence:

$$\gamma^3 = \left(\begin{array}{c|c|c|c} \left[\begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \end{array} \right] & \left[\begin{array}{c} 0 \\ 1 \\ 0 \\ 0 \end{array} \right] & \left[\begin{array}{c} 0 \\ 0 \\ 1 \\ 0 \end{array} \right] & \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 1 \end{array} \right] \end{array} \right)$$

Using this communication sequence, the discretized model of the controlled car suspension system becomes periodic. Applying the methodology described in section V, paragraph C, the optimal periodic control gains ($\tilde{K}(0), \tilde{K}(1), \tilde{K}(2), \tilde{K}(3)$) can be derived.

The suspension system is evaluated by subjecting the left side of the vehicle to a "chuck hole" discrete road disturbance [22], in which the road elevation falls linearly by 5 cm over a distance of 76 cm, remains at that elevation for the next 76 cm, and then returns back linearly to the original level over the next 76 cm. The vehicle speed is equal to 40 km/h. First, the performance of the designed active suspension controller is evaluated and compared to the passive suspension. Then, the performance of the OPP algorithm is compared to that obtained by the application of the static scheduling algorithm. OPP and the static scheduling algorithm are both based on the communication sequence and control gains described above. Heave, roll and pitch velocity responses are illustrated in figures 10, 11 and 12.

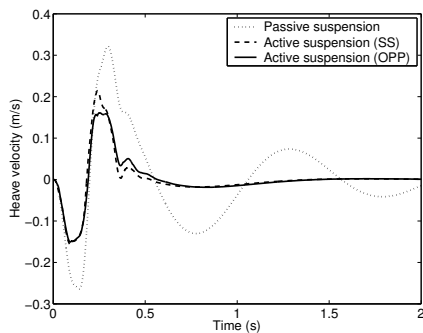


Fig. 10. Heave velocity of the passive and active suspension (controlled with the static scheduling and the OPP algorithms)

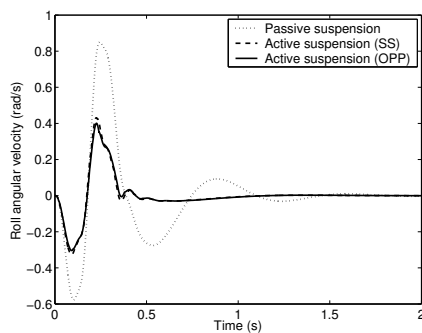


Fig. 11. Roll velocity of the passive and active suspension (controlled with the static scheduling and the OPP algorithms)

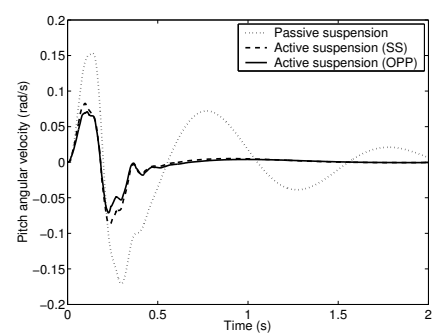


Fig. 12. Pitch velocity of the passive and active suspension (controlled with the static scheduling and the OPP algorithms)

From these simulation results, it can be seen that the active suspension induces an important improvement of the ride performance compared to the passive suspension (smaller and better damped velocities and thus accelerations). The responses using the OPP algorithm show a slight improvement with respect to the static scheduling (SS) algorithm. The improvements in ride comfort shown by the active suspension are obtained with suspension and tire deflection levels which are close to those obtained with the passive suspension. Finally, the quadratic cost functions corresponding to the ideal continuous time LQR controller, the static scheduling algorithm and the OPP scheduling algorithm are evaluated. The steady state cost function values corresponding to the static scheduling, OPP and to the ideal implementation are respectively equal to 4459, 4122 and 3290. Consequently, the improvements in terms of quality of control that were achieved by the OPP algorithm are equal to 28.8%. These improvements are significant, but not as “spectacular” as those observed in the previous example because the different components of the suspension system are tightly coupled: a disturbance affecting a single wheel influences all the state variables of the system.

D. Real-time implementation aspects of the OPP over infinite horizon algorithm

The OPP over infinite horizon algorithm requires the on-line evaluation of T quadratic functions of the extended state vector \tilde{x} . The computational complexity is linear with respect to the length of the sequence, quadratic with respect to the extended state (like a classical state-feedback control law) and independent of the size of the horizon. In the case of the suspension example, the additional computational requirements of OPP are approximately 4.3 times those required by the state feedback operation $u(k) = K(p)\tilde{x}(k)$. Using OPP follows the idea of trading additional computations for a more efficient use of network resources [23]. Further reducing the computational requirements is a both a difficult and interesting research issue. The application of multi-parametric programming techniques to the optimal control of hybrid systems have been recently considered [24]. In [25], an approximate solution to the model predictive control of linear systems with input and state constraints was proposed. This method is based on the partitioning the state space onto hypercubes which may

be further partitioned in order to meet on the cost function approximation error bounds and constraints violations bounds. By imposing an orthogonal search tree on the partition, the on-line computational requirements are significantly reduced with respect to the true optimal explicit MPC law. The search method is logarithmic with respect to the number of regions, but this number may augment exponentially with respect to the state vector size. The memory requirements, which are needed to the storage of this partitioning information and of the state-feedback control law parameters, are also tightly dependent on this number of regions. This problem is tractable for low dimensional systems but it may be problematic for problems like that treated in this paper. The difference between this approach and that of [25] resides in the fact that we explore the set of pointer positions which is in general and particularly in this application less complex than the state space. This considerably reduces the computational complexity.

VII. CONCLUSIONS

In this paper, the problem of the optimal integrated control and scheduling was addressed theoretically. A hybrid model of a control system and its limited communication network was described. In this modeling, control and scheduling are tightly coupled in order to improve control performance. A new on-line scheduling algorithm for networked control systems was introduced. Using this computationally efficient algorithm, *communication resources* are allocated according to the “needs” of the controlled *dynamical systems*. Stability conditions for the systems scheduled using this algorithm were stated. Simulation results 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 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 2004 American Control Conference*, Boston, USA, June 2004.

- [3] K.-E. Årzén, A. Cervin, J. Eker, and L. Sha, "An introduction to control and real-time scheduling co-design," in *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia, December 2000.
- [4] D. Hristu, "Optimal control with limited communication," Ph.D. dissertation, Division of Engineering and Applied Sciences, Harvard University, June 1999.
- [5] D. Hristu and K. Morgansen, "Limited communication control," *System and Control Letters*, vol. 37, no. 4, pp. 193–205, July 1999.
- [6] G. Walsh and H. Ye, "Scheduling of networked control systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 57–65, February 2001.
- [7] H. Rehbinder and M. Sanfridson, "Scheduling of a limited communication channel for optimal control," *Automatica*, vol. 40, no. 3, pp. 491–500, March 2004.
- [8] 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.
- [9] G. C. Goodwin, H. Haimovich, D. E. Quevedo, and J. S. Welsh, "A moving horizon approach to networked control systems design," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1562–1572, September 2004.
- [10] M.-M. Ben Gaid and A. Çela, "Model predictive control of systems with communication constraints," in *Proceedings of the 16th IFAC World Congress on Automatic Control*, Prague, Czech Republic, July 2005.
- [11] M.-M. Ben Gaid, A. Çela, and Y. Hamam, "Optimal integrated control and scheduling of systems with communication constraints," in *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference*, Seville, Spain, December 2005.
- [12] F.-L. Lian, J. Moyne, and D. Tilbury, "Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 66–83, February 2001.
- [13] 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.
- [14] R. Brockett, "Stabilization of motor networks," in *Proceedings of the 34th IEEE Conference on Decision and Control*, New Orleans, Los Angeles, USA, December 1995.
- [15] B. Lincoln and B. Bernhardsson, "LQR Optimization of linear system switching," *IEEE Transactions on Automatic Control*, vol. 47, no. 10, pp. 1701–1705, October 2002.
- [16] Y. Ye, "A fully polynomial-time approximation algorithm for computing a stationary point of the general linear complementarity problem," *Mathematics of Operations Research*, vol. 18, no. 2, pp. 334–345, 1993.
- [17] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.
- [18] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems : Theory and Design*. Prentice Hall, 1997.
- [19] P. Martí, J. Fuertes, K. Ramamritham, and G. Fohler, "Jitter compensation for real-time control systems," in *Proceedings of the 22nd IEEE Real-Time Systems Symposium*, London, England, December 2001.
- [20] C. Ionete and A. Çela, "Structural properties and stabilization of NCS with medium access constraints," Submitted to the *European Journal of Control*, 2006.
- [21] S. Bittanti, A. J. Laub, and J. Willems, Eds., *The Riccati Equation*. Springer-Verlag, 1991, ch. The periodic Riccati Equation.
- [22] R. M. Chalasani, "Ride performance potential of active suspension systems Part II: Comprehensive analysis based on a full-car model," in *Proceedings of the Symposium on Simulation and Control of Ground Vehicles and Transportation Systems*, ASME AMD, Anaheim, CA, USA, December 1986.
- [23] J. Yook, D. Tilbury, and N. Soparkar, "Trading computation for bandwidth: Reducing communication in distributed control systems using state estimators," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 4, pp. 503–518, July 2002.
- [24] F. Borrelli, M. Baotić, A. Bemporad, and M. Morari, "Dynamic programming for constrained optimal control of discrete-time linear hybrid systems," *Automatica*, vol. 41, no. 10, pp. 1709–1721, October 2005.
- [25] T. A. Johansen and A. Grancharova, "Approximate explicit model predictive control implemented via orthogonal search tree partitioning," in *Proceedings of the 15th IFAC World Congress*, Barcelona, Spain, July 2002.