

Optimal Scheduling of Control Tasks with State Feedback Resource Allocation

Mohamed El Mongi Ben Gaid, Arben Çela, Yskandar Hamam, and Cosmin Ionete

Abstract—In a large category of embedded systems, computing resources are limited. Consequently, they need to be exploited as efficiently as possible. Recently, many research works have demonstrated that considering jointly the problems of control and scheduling leads to a better control performance, given the same computing resources. In this paper, the problem of the optimal integrated control and non-preemptive off-line scheduling of control tasks in the sense of the H_2 performance criterion is addressed. It is shown that this problem can be decomposed into two sub-problems which can be solved separately. The first sub-problem aims at finding the optimal non-preemptive off-line schedule, and is solved using efficient Branch and Bound algorithms. The second sub-problem uses the lifting technique to determine the optimal control gains, based on the solution of the first sub-problem. Finally, in order to improve the control performance by dynamically allocating the computational resources, an efficient on-line scheduling heuristic is proposed.

I. INTRODUCTION

Embedded micro-computers are increasingly being used in modern control applications in order to perform the computations of the control laws. The development cycle of these applications often involves two different steps, which are usually performed in isolation. The first step is the control design, which is performed by the control engineer, assuming an ideal implementation. The following step is the software design. In this step, the functionalities of the application, specified by the control engineer, are decomposed into tasks. Usually, hard real-time constraints are associated to the tasks. This separation of concerns simplifies the design, especially when the applications are complex. However, for small applications, with a reduced number of tasks, and designated to a small footprint hardware target, the integration of control design and real-time scheduling design leads to a better control performance, using the same hardware resources.

The problem of the co-design of the control and the real-time scheduling was first introduced in [1]. In [2], the elastic task model [3] was applied to the scheduling of control tasks with variable execution times. In [4], the optimal off-line scheduling of control tasks in the sense of LQG was considered, assuming that all the control tasks have the

same execution time. The resolution of this problem was performed using the exhaustive search method, which limits the application of this approach to applications with a limited number of tasks. The idea of the dynamical allocation of processor resources as a function of the plant state was proposed in [5]. The problem of the optimal integrated control and scheduling was investigated in [6]. It was demonstrated that the optimal scheduling policy in the sense of an LQ performance index is very dependant from the plant state. A significant improvement in the control performance can be obtained by rapidly reacting to the disturbances [7]. This rapid reaction can be performed by plant state based on-line scheduling algorithms. The problem of the optimal on-line sampling period assignment was studied in [8]. A sub-optimal periodic sampling period assignment heuristic was proposed. In apposition to this approach, the on-line scheduling heuristic proposed in this paper, and called Reactive Pointer Placement (RPP) scheduling, changes the sampling period in reaction to the disturbances, and not according to a periodic triggering.

In this paper, a solution of the optimal control and non-preemptive off-line scheduling according to the H_2 performance criterion is proposed. Instead of using exhaustive search as in [4], a decomposition of the problem into two independent sub-problems is performed. The first sub-problem aims at the finding of the optimal static cycle schedule and is solved using the Branch and Bound method. The second sub-problem uses the result of the first sub-problem to determine the optimal control gains, applying the well known lifting technique. Finally, an efficient on-line scheduling heuristic, based on the pre-calculated off-line schedule is proposed, in order to improve the responsiveness to unpredictable disturbances that can occur at runtime.

II. OPTIMAL OFF-LINE SCHEDULING

A. Problem formulation

Consider a collection of N continuous-time LTI systems $(S^{(j)})_{1 \leq j \leq N}$. Assume that each system $S^{(j)}$ is controlled by a task $\tau^{(j)}$, characterized by its execution time $d^{(j)}$. The execution times $d^{(1)}, \dots, d^{(N)}$ of the tasks are assumed to be integers and expressed as multiples of an elementary time unit T_p . Task scheduling can be described by associating scheduling functions $y^{(j)}$ to control tasks $\tau^{(j)}$ such that

$$y^{(j)}(k) = \begin{cases} 1 & \text{if task } \tau^{(j)} \text{ finishes at instant } kT_p \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

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

C. Ionete is with The university of Craiova, Faculty of Automation, Computers and Electronics and now with the *COSI Lab.*, ESIEE (e-mail: cosmin@automation.ucv.ro)

$y^{(j)}(k)$ is called the *execution end indicator* of the jobs of task $\tau^{(j)}$. Due to the use of a non-preemptive scheduling, the $d^{(j)}$ slots preceding the end of a job of task $\tau^{(j)}$ are allocated to its execution. Using this observation, the processor utilization can be described by

$$e^{(j)}(k) = \sum_{l=k}^{k+d^{(j)}-1} y^{(j)}(l) \quad (2)$$

$e^{(j)}(k)$ is the *task execution indicator* corresponding to the jobs of task $\tau^{(j)}$ and verifies

$$e^{(j)}(k) = 1 \iff \begin{array}{l} \text{the processor is allocated to task } \tau^{(j)} \\ \text{during interval } [(k-1)T_p, kT_p) \end{array} \quad (3)$$

During interval $[(k-1)T_p, kT_p)$, the processor can only execute one task. This constraint can be modeled by the following inequality

$$\sum_{j=1}^N e^{(j)}(k) \leq 1 \quad (4)$$

Each system $S^{(j)}$ is characterized by its discrete-time model, derived at the sampling period T_p , and described by

$$x^{(j)}(k+1) = A^{(j)}x^{(j)}(k) + B_1^{(j)}w^{(j)}(k) + B_2^{(j)}u^{(j)}(k) \quad (5a)$$

$$z^{(j)}(k) = C_1^{(j)}x^{(j)}(k) + D_{12}^{(j)}u^{(j)}(k) \quad (5b)$$

where $x^{(j)}(k) \in \mathbb{R}^{n_j}$ is the state vector, $w^{(j)}(k) \in \mathbb{R}^{p_j}$ is the disturbance input, $u^{(j)}(k) \in \mathbb{R}^{m_j}$ is the control input and $z^{(j)}(k) \in \mathbb{R}^{q_j}$ is the controlled output. We assume that

- 1) The pair $(A^{(j)}, B_2^{(j)})$ is controllable
- 2) $R = \begin{bmatrix} C_1^{(j)} \\ D_{12}^{(j)} \end{bmatrix} \begin{bmatrix} (C_1^{(j)})^T & (D_{12}^{(j)})^T \end{bmatrix} = \begin{bmatrix} R_{xx}^{(j)} & R_{xu}^{(j)} \\ (R_{xu}^{(j)})^T & R_{uu}^{(j)} \end{bmatrix} \geq 0$
with $R_{uu}^{(j)} > 0$

Using straightforward algebraic manipulations, systems $(S^{(j)})_{1 \leq j \leq N}$ can be described using an extended state model representing a global system \mathcal{S} described by

$$x(k+1) = Ax(k) + B_1w(k) + B_2u(k) \quad (6a)$$

$$z(k) = C_1x(k) + D_{12}u(k) \quad (6b)$$

In the considered modeling, when a control task finishes its execution, then it immediately updates the plant, which means that

$$u^{(j)}(k) \text{ is updated at instant } kT_p \iff y^{(j)}(k) = 1 \quad (7)$$

The digital-to-analog converters, use zero-order-holders to maintain the last received control inputs constant until new control values are updated. This assertion can be modeled by

$$y^{(j)}(k) = 0 \implies u^{(j)}(k) = u^{(j)}(k-1) \quad (8)$$

In order to formulate the joint problem of optimal control and scheduling, in addition to the modeling of the tasks and the representation of the system's dynamics, it is necessary to choose an adequate criterion of performance. The previous studies which were carried out on the joint problem of

the optimal control and scheduling, starting from a given initial condition, have shown that the optimal schedule is extremely dependant from the chosen initial conditions [6]. This dependence can be exploited by the on-line scheduling algorithms in order to improve the control performance. But when only a fixed schedule is desired, it is necessary to use performance criteria which depend on the intrinsic characteristics of the system, not on a particular evolution or initial state. The use of the well known H_2 performance criterion provides an interesting solution to this problem. In fact, using this performance index, the obtained off-line schedules will be independent from any initial condition. Moreover, the results can be easily transposed to an LQG context [9].

Let T be the period of the static schedule, then scheduling functions $y^{(j)}(k)$ and $e^{(j)}(k)$ will be periodic with period T . In order to determine the H_2 norm of the system, we adopt a definition of this norm which is based on the impulsive response of a linear discrete-time periodic system [10]. This definition generalizes the well known definition of the H_2 norm for the discrete LTI systems. Let $(e_i)_{1 \leq i \leq p}$ be the canonical basis vectors in \mathbb{R}^p ($p = \sum_{j=1}^N p_j$) and δ_k the Dirac impulse applied at instant k . Using these notations, $\delta_k e_i$ is a Dirac impulse applied to the i^{th} disturbance input at instant k . Let g_{ik} be the resulting controlled output of the system assuming zero initial conditions. The H_2 norm of the periodic system \mathcal{S} is defined as

$$\|\mathcal{S}\|_2 = \sqrt{\frac{1}{T} \sum_{k=0}^{T-1} \sum_{i=1}^p \|g_{ik}\|_2^2} \quad (9)$$

Using this definition to calculate the H_2 norm involves the calculation of $\|g_{ik}\|_2$, which requires the observation of the system's response over an infinite time horizon. In this work, a very close approximation of $\|g_{ik}\|_2$ is obtained through a finite horizon H from the instant where the impulse is applied. It is necessary to choose H greater than the response time of the slowest system.

B. Solving of the optimal scheduling sub-problem

In this section, the translation of the optimal scheduling sub-problem in the sense of H_2 into the Mixed Integer Quadratic Formulation is described. This translation requires the transformation of all the involved constraints into linear equalities and/or inequalities. Constraints (8) can be translated into equivalent linear inequalities and equalities if extra variables are introduced, as illustrated in [11]. Remarking that (8) is equivalent to

$$u^{(j)}(k) - u^{(j)}(k-1) = y^{(j)}(k)u^{(j)}(k) - y^{(j)}(k)u^{(j)}(k-1) \quad (10)$$

and introducing the extra variables

$$v^{(j)}(k) = y^{(j)}(k)u^{(j)}(k) \quad (11)$$

then (8) can be rewritten in the equivalent form

$$\begin{aligned} v^{(j)}(k) &\leq U^{(j)}y^{(j)}(k) \\ v^{(j)}(k) &\geq L^{(j)}y^{(j)}(k) \\ v^{(j)}(k) &\leq u^{(j)}(k) - L^{(j)}(1 - y^{(j)}(k)) \\ v^{(j)}(k) &\geq u^{(j)}(k) - U^{(j)}(1 - y^{(j)}(k)) \end{aligned} \quad (12)$$

where $U^{(j)}$ and $L^{(j)}$ are respectively the upper and the lower bounds of the control commands $u^{(j)}$ of system $S^{(j)}$. The product $w^{(j)}(k) = y^{(j)}(k)u^{(j)}(k-1)$ can also be translated using the same procedure.

The constraints involved in this problem can be classified into two groups. The first group is related to the scheduling constraints (2) and (4). Let

$$\bar{Y}^{(j)} = \begin{bmatrix} y^{(j)}(0) \\ \vdots \\ y^{(j)}(H-1) \end{bmatrix} \quad \text{and} \quad \bar{Y} = \begin{bmatrix} \bar{Y}^{(1)} \\ \vdots \\ \bar{Y}^{(N)} \end{bmatrix}$$

then the constraints belonging to this group can be described by

$$\mathcal{A}_s \bar{Y} \leq \mathcal{B}_s \quad (13)$$

The second group is related to the calculation of the impulsive responses g_{ik} , for $0 \leq k \leq T-1$ and $1 \leq i \leq p$, over the horizon H . Let u^{ik} , x^{ik} , z^{ik} , v^{ik} and w^{ik} be respectively the values of the control, the state, the controlled output and the auxiliary variables corresponding to a Dirac impulse applied at instant k to the i^{th} disturbance input of the global system. Let S^{ik} be the set of the involved constraints, for a given response g_{ik} . S^{ik} includes the state model (6), the Dirac impulse verifying $w_r^{ik}(k) = 1$ and $w_r^{ik}(l) = 0$ for $r \neq i$ and $l \neq k$, in addition to the constraints that must be added to the problem to ensure the causality of the response ($u^{ik}(l) = 0$ for $l < k$).

$$\begin{aligned} \text{Let } \bar{U}^{ik} &= \begin{bmatrix} u^{ik}(0) \\ \vdots \\ u^{ik}(H-1) \end{bmatrix}, \quad \bar{X}^{ik} = \begin{bmatrix} x^{ik}(0) \\ \vdots \\ x^{ik}(H-1) \end{bmatrix}, \quad \bar{Z}^{ik} = \\ &\begin{bmatrix} z^{ik}(0) \\ \vdots \\ z^{ik}(H-1) \end{bmatrix}, \quad \bar{V}^{ik} = \begin{bmatrix} v^{ik}(0) \\ \vdots \\ v^{ik}(H-1) \end{bmatrix}, \quad \bar{W}^{ik} = \begin{bmatrix} w^{ik}(0) \\ \vdots \\ w^{ik}(H-1) \end{bmatrix} \\ \text{and } \mathcal{V}^{ik} &= \begin{bmatrix} \bar{U}^{ik} \\ \bar{X}^{ik} \\ \bar{Z}^{ik} \\ \bar{V}^{ik} \\ \bar{W}^{ik} \end{bmatrix}, \text{ then the set of constraints } S^{ik} \text{ can be} \\ \text{described by} & \end{aligned}$$

$$\mathcal{A}^{ik} \begin{bmatrix} \bar{Y} \\ \mathcal{V}^{ik} \end{bmatrix} \leq \mathcal{B}^{ik} \quad (14)$$

Consequently, the optimal scheduling problem in the sense of the H_2 norm can be written in the form

$$(POH_2) \begin{cases} \min_{\bar{Y}, (\mathcal{V}^{ik})_{1 \leq i \leq p, 0 \leq k \leq T-1}} \sum_{k=0}^{T-1} \sum_{i=1}^p (\mathcal{V}^{ik})^T \mathcal{H} \mathcal{V}^{ik} \\ \mathcal{A}_s \bar{Y} \leq \mathcal{B}_s \\ \mathcal{A}^{ik} \begin{bmatrix} \bar{Y} \\ \mathcal{V}^{ik} \end{bmatrix} \leq \mathcal{B}^{ik}, \text{ for } 1 \leq i \leq p, 0 \leq k \leq T-1 \end{cases}$$

Problem (POH_2) is a Mixed Integer Quadratic Problem (MIQP). The resolution of this problem gives the optimal off-line schedule \bar{Y}^* .

C. Solving of the optimal control sub-problem

Knowing the optimal off-line schedule \bar{Y}^* , which is a solution of the optimization problem (POH_2) , it is then possible to derive the optimal control gains, according to the H_2 performance criterion. In opposition to problem (POH_2) , the determination of the control gains can be performed on each system separately.

It is easy to see that knowing $(\bar{Y}^{(j)})^*$, the time-varying model of system $S^{(j)}$ is given by

$$\begin{aligned} \tilde{x}^{(j)}(k+1) &= \tilde{A}^{(j)}(k)\tilde{x}^{(j)}(k) + \tilde{B}_1^{(j)}w^{(j)}(k) + \tilde{B}_2^{(j)}(k)\tilde{u}^{(j)}(k) \\ \tilde{z}^{(j)}(k) &= \tilde{C}_1^{(j)}(k)\tilde{x}^{(j)}(k) + \tilde{D}_{12}^{(j)}(k)\tilde{u}^{(j)}(k) \end{aligned} \quad (16)$$

$$\text{where } \tilde{x}^{(j)}(k) = \begin{bmatrix} x^{(j)}(k) \\ u^{(j)}(k-1) \end{bmatrix}, \quad \tilde{z}^{(j)}(k) = z^{(j)}(k)$$

$$\tilde{A}^{(j)}(k) = \begin{bmatrix} A^{(j)} & B_2^{(j)}(1 - y^{(j)}(k)) \\ 0 & (1 - y^{(j)}(k)) \end{bmatrix}$$

$$\tilde{B}_1^{(j)} = \begin{bmatrix} B_1^{(j)} \\ 0 \end{bmatrix}, \quad \tilde{B}_2^{(j)}(k) = \begin{bmatrix} B_2^{(j)}y^{(j)}(k) \\ y^{(j)}(k) \end{bmatrix}$$

$$\tilde{C}_1^{(j)}(k) = \begin{bmatrix} C_1^{(j)} \\ D_{12}^{(j)}(1 - y^{(j)}(k)) \end{bmatrix} \quad \text{and} \quad \tilde{D}_{12}^{(j)}(k) = D_{12}^{(j)}y^{(j)}(k).$$

The control signal $\tilde{u}^{(j)}(k)$ can be seen as the command that could be fed-back to the plant at each sampling period, if the implementation was ideal (i.e $y^{(j)}(k) = 1$ for all k). However, due to the resource limitations, only the control $u^{(j)}(k)$ is effectively applied, where

$$u^{(j)}(k) = y^{(j)}(k)\tilde{u}^{(j)}(k) + (1 - y^{(j)}(k))u^{(j)}(k-1)$$

The binary function $y^{(j)}(k)$ is T-periodic. Consequently, matrices $\tilde{A}^{(j)}(k)$, $\tilde{B}_2^{(j)}(k)$, $\tilde{C}_1^{(j)}(k)$ and $\tilde{D}_{12}^{(j)}(k)$ are T-periodic. An equivalent time-invariant representation of system (16) can be established using the lifting technique [12]. Then, using the method in [13], the optimal H_2 controller of the lifted LTI system may be obtained. Consequently, the optimal H_2 controller of system (16) can be deduced.

D. A numerical example

Consider the collection of 2 open-loop unstable sampled-data LTI systems defined by

$$A^1 = \begin{pmatrix} 0.9967 & 0.0266 \\ -0.2500 & 0.9987 \end{pmatrix} \quad B_1^1 = \begin{pmatrix} 0.0133 \\ 0.9999 \end{pmatrix}$$

$$C_1^1 = \begin{pmatrix} 89.4427 & 0 \\ 0 & 3.1623 \\ 0 & 0 \end{pmatrix} \quad D_{11}^1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$B_2^1 = B_2^1 \quad D_{12}^1 = 0$$

$$A^2 = \begin{pmatrix} 1.1180 & 0 & 0.0025 & 0.5531 \\ 0 & 1 & 0.2129 & 0 \\ 0 & 0 & 0.7613 & 0 \\ 0.4518 & 0 & 0.0093 & 1.1180 \end{pmatrix} \quad B_1^2 = 10^{-4} \begin{pmatrix} 28 \\ 11 \\ 88 \\ 106 \end{pmatrix}$$

$$C_1^2 = \begin{pmatrix} 10 & 0 & 0 & 0 \\ 0 & 31.62 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad D_{11}^2 = 10^{-4} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$B_2^2 = B_2^2 \quad D_{12}^2 = 0$$

The optimal solutions (for T ranging from 3 to 12) are illustrated in the table below. The relative optimality gap of the used Branch and Bound algorithm is equal to 10^{-5} . The execution times of the corresponding tasks are equal to 1. The elementary time unit is $T_p = 0.001$ s. The computations were performed on a PC equipped with a 2.2 GHz Intel Celeron processor and 256 MB of RAM. In this particular implementation of the optimization algorithm, H must be a multiple of T . It is sufficient to choose H bigger than 25.

TABLE I
OPTIMAL H_2 NORM AS A FUNCTION OF THE PERIOD T

Period T	H	H_2 norm	Optimal Schedule	CPU Time (s)
3	30	3.7257	112...	12
4	32	4.4154	2121...	25
5	30	3.9725	21121...	35
6	30	3.7257	112112...	44
7	35	4.0556	1211211...	61
8	32	3.8813	21121211...	163
9	27	3.7257	112112112...	124
10	30	3.9590	1121112112...	292
11	33	3.8395	11212112112...	395
12	36	3.7257	112112112112...	787

The results for this particular example are given in table I. They indicate that the minimal optimal schedule is of length $T = 3$. The length of the optimal schedules which gives the best H_2 norm ($H_2 = 3.7257$) is a multiple of 3. The resource allocation depends on the rapidity of the systems and on their sensitivity to the disturbances.

III. ON-LINE SCHEDULING HEURISTIC

Using an on-line scheduling algorithm, which takes into account the state of the plant, significant improvements in the control performance can be achieved, as illustrated in [6], [8]. In order to be "implementable", the execution overhead of an on-line scheduling algorithm must be small compared to the execution load of the other control tasks. The proposed heuristic, called Reactive Pointer Placement (RPP) scheduling, is inspired from the Optimal Pointer Placement (OPP) scheduling algorithm proposed in [6], and aims to adapt this heuristic to the context of monoprocessor scheduling by reducing its computational requirements.

A. Algorithm description

Given the optimal off-line schedule, the ordered execution of the control tasks during the period $[0, T \times T_p)$ can be described by the sequence $(s(0), \dots, s(\mathcal{T} - 1))$, where \mathcal{T} is the number of tasks which are executed during the period $[0, T \times T_p)$. For example, the sequence $(1, 2, 2, 3)$ indicates that during this period, the processor begins by executing task $\tau^{(1)}$, followed by two consecutive executions of task $\tau^{(2)}$, which are followed by the execution of task $\tau^{(3)}$. At runtime, the execution of the periodic off-line schedule can be described using the notion of *pointer*. The pointer can be seen as a variable p which contains the index of the control task to execute. The pointer is incremented after each execution. If it reaches the end of the sequence, its position is reset. After each task execution, the position of the pointer is updated according to

$$p := (p + 1) \bmod \mathcal{T} \quad (17)$$

Let $\tilde{x}(k) = [\tilde{x}^{(1)}(k)^T \dots \tilde{x}^{(N)}(k)^T]^T$. If the pointer is placed at position p at instant k , then the cost function corresponding to an evolution of system $S^{(j)}$ (resp. the global system \mathcal{S}) over an infinite horizon starting from the state $\tilde{x}^{(j)}(k)$ (resp. $\tilde{x}(k)$) at instant k and using the static scheduling algorithm is $J^{(j)}(k, p) = \sum_{i=0}^{\infty} (z^{(j)})^T(k+i)z^{(j)}(k+i)$ (resp.

$J(k, p) = \sum_{j=1}^N J^{(j)}(k, p)$). Note that if $\tilde{M}^{(j)}(k)$ is the periodic solution of the Riccati equation associated with the optimal control problem (in the sense of LQR) of system (16), then $J^{(j)}(k, p) = \tilde{x}^{(j)}(k)^T \tilde{M}^{(j)}(p) \tilde{x}^{(j)}(k)$.

The RPP algorithm relies on two basic key points

1) *A quicker detection of the disturbances affecting a given system:* In order to improve the control performance with respect to the off-line scheduling algorithm, this detection have to be quicker than that allowed by the static algorithm. To this end, when the scheduling pointer is at position p ($0 \leq p < \mathcal{T}$) the scheduling algorithm first reads the outputs of system $S^{s(p)}$, and also the outputs of another system $S^{o(p)}$, given by the *detection sequence* o . The detection sequence is calculated in order to maximize the responsiveness of the RPP algorithm, according to a well defined optimization problem which will be described below.

Let Π be the set of all the permutations of the \mathcal{T} -uplet $(0, 1, \dots, \mathcal{T} - 1)$. Let $\pi \in \Pi$ and o_π the sequence defined by

$$o_\pi = (s(\pi(0)), s(\pi(1)), \dots, s(\pi(\mathcal{T} - 1))) \quad (19)$$

The *binary detection indicators* associated to each system $S^{(j)}$ are defined by

$$\begin{cases} D_\pi^{(j)}(p) = 1 & \text{if } o_\pi(p) = j \text{ or } s(p) = j \\ D_\pi^{(j)}(p) = 0 & \text{otherwise} \end{cases} \quad (20)$$

The binary detection indicators indicate whether or not the outputs of system $S^{(j)}$ are read, for a given position p of the sequence pointer. Let $\bar{D}_\pi^{(j)}(k) = D_\pi^{(j)}(k \bmod \mathcal{T})$. The detection sequence $o = o_{\pi^*}$ is determined by the solution π^* of the following optimization problem (POS).

$$\pi^* = \min_{\pi \in \Pi} \sum_{j=1}^N \max_{k_1, k_2, k_1 \leq k_2} \left\{ k_2 - k_1, \text{ such that: } \bar{D}_{\pi}^{(j)}(k_1) = 1, \bar{D}_{\pi}^{(j)}(k_2) = 1 \text{ and for all } k_1 < k < k_2, \bar{D}_{\pi}^{(j)}(k) = 0 \right\} \quad (\text{POS})$$

This optimization problems aims at minimizing the sum over j of the “maximal distances” between two successive output reading of system $S^{(j)}$.

2) *A plant state based feedback scheduling policy:* In order to describe the dynamical state of each system $S^{(j)}$, and how much it is close to the equilibrium, positive constants $\varepsilon_x^{(j)}$ are introduced. $\varepsilon_x^{(j)}$ have to be chosen small enough to consider that when $\|x^{(j)}(k)\|_{\infty} \leq \varepsilon_x^{(j)}$, then system $S^{(j)}$ is considered at the equilibrium. The RPP uses a data structure $E \in \{0, 1\}^N$ to memorize the dynamical states of all the controlled systems, according to the relation

$$\begin{cases} E_j = 1 & \text{if at the last execution of } \tau^{(j)}, \text{ at instant } k_l, \\ & \text{the state verifies } \|x^{(j)}(k_l)\|_{\infty} \leq \varepsilon_x^{(j)} \\ E_j = 0 & \text{otherwise} \end{cases} \quad (21)$$

Each task $\tau^{(j)}$ is responsible of updating the variable E_j when it is executed. The RPP scheduling algorithm operates like the static scheduling algorithm (i.e. updating the pointer according to (17)), except when the three conditions listed below are fulfilled. In this case, a *reactive pointer change* (i.e. the pointer is placed at position $\pi^*(p)$ instead of p) is performed. The conditions are

- 1) All the systems were considered at the equilibrium at the last execution of their control tasks (i.e. $\|E\|_2^2 = N$).
- 2) Placing the pointer at position $\pi^*(p)$ gives a better predicted cost function than the use of the static scheduling (i.e. $J(k, \pi^*(p)) < J(k, p)$).
- 3) An entire cycle is performed since the last reactive pointer change.

Condition 3) was added in order to guarantee a maximum time interval between two plant output readings. In fact, as indicated previously, RPP is responsible of plant output readings (of systems $S^{(p)}$ and $S^{o(p)}$). Conditions 1) and 3) were included in order to reduce the overhead of the feedback scheduler. These assumptions can be disregarded (i.e. RPP is triggered at each sampling period without any restriction) if the resulting overhead of the algorithm remains negligible. In some situations, in order to ensure reduced computational requirements of the RPP scheduling algorithm, a less complex test than 2) is required. In fact, condition 2) is based on the knowledge the plant states of the N systems. In the following, we propose a sufficient condition to guarantee condition 2), which is based on the reading of the outputs of only two systems. Let $R^{(i)}$ the equilibrium region of a given system $S^{(i)}$ (i.e. $R^{(i)} = \left\{ \tilde{x}^{(i)}(k) / \left\| \tilde{x}^{(i)}(k) \right\|_{\infty} \leq \varepsilon_x^{(i)} \right\}$). For a given system $S^{(j)}$, let

$$\bar{J}_{min}^{(j)}(p) = \min_{\tilde{x}^{(i)}(k) \in R^{(i)}, i \neq j} \left(\sum_{i \neq j} J^{(i)}(k, p) \right)$$

$$\bar{J}_{max}^{(j)}(\pi^*(p)) = \max_{\tilde{x}^{(i)}(k) \in R^{(i)}, i \neq j} \left(\sum_{i \neq j} J^{(i)}(k, \pi^*(p)) \right).$$

The following proposition states a sufficient condition which guarantees that the reactive pointer change improves the control performance. This sufficient condition considerably reduces the computations compared to the necessary and sufficient condition described in 2).

Proposition 1: For a given system $S^{(j)}$ and pointer position p , if $\forall i, 1 \leq i \leq N$ and $i \neq j$, $\left\| \tilde{x}^{(i)}(k) \right\|_{\infty} \leq \varepsilon_x^{(i)}$ and $J^{(j)}(k, p) + \bar{J}_{min}^{(j)}(p) > J^{(j)}(k, \pi^*(p)) + \bar{J}_{max}^{(j)}(\pi^*(p))$ then $J(k, \pi^*(p)) < J(k, p)$

Proof: This result directly follows from the fact that $J(k, p) = J^{(j)}(k, p) + \sum_{i \neq j} J^{(i)}(k, p) \geq J^{(j)}(k, p) + \bar{J}_{min}^{(j)}(p)$ and

$$\begin{aligned} J(k, \pi^*(p)) &= J^{(j)}(k, \pi^*(p)) + \sum_{i \neq j} J^{(i)}(k, \pi^*(p)) \\ &\leq J^{(j)}(k, \pi^*(p)) + \bar{J}_{max}^{(j)}(\pi^*(p)). \end{aligned}$$

Constants $\bar{J}_{min}^{(j)}(p)$ and $\bar{J}_{max}^{(j)}(\pi^*(p))$ can be easily pre-computed off-line using a QP solver. Note that $J^{(j)}(k, p)$ can be replaced by an upper bound and $J^{(j)}(k, \pi^*(p))$ by a lower bound, for control laws where the exact calculation of the cost may be impracticable or computationally complex.

Let $J^{rpp}(\tilde{x}(i), i, f)$ be the cost function corresponding to an evolution from instant $k = i$ to instant $k = f$ starting from the extended state $\tilde{x}(i)$ where the RPP scheduling algorithm is applied. The performance improvements of RPP are stated in the following theorem

Theorem 1: Let $\tilde{x}(0)$ be a given initial extended state of the global system \mathcal{S} (composed of systems $(S^{(j)})_{1 \leq j \leq N}$) and p_0 an initial pointer position of the static scheduling algorithm. Then

$$J^{rpp}(\tilde{x}(0), 0, \infty) \leq J^{ss}(\tilde{x}(0), 0, \infty, p_0)$$

Proof: Let \tilde{x}^{rpp} be the extended state trajectory of system \mathcal{S} when scheduled using the RPP algorithm, $p(l)$ the pointer position at the l^{th} execution of the RPP algorithm and k_l the instant corresponding to the end of this l^{th} execution. According to the RPP strategy, if the pointer position at the l^{th} execution is set to $p(l) = \pi^*((p(l-1) + 1) \bmod \mathcal{T})$ instead of $(p(l-1) + 1) \bmod \mathcal{T}$ then necessarily

$$\begin{aligned} J^{ss}(\tilde{x}(k_l), k_l, +\infty, \pi^*((p(l-1) + 1) \bmod \mathcal{T})) \\ \leq J^{ss}(\tilde{x}(k_l), k_l, +\infty, (p(l-1) + 1) \bmod \mathcal{T}) \end{aligned}$$

Consequently, at instant k_l , the relation $J^{rpp}(\tilde{x}(0), 0, k_{l-1}) + J^{ss}(\tilde{x}^{rpp}(k_l), k_l, +\infty, p(l)) \leq J^{ss}(\tilde{x}(0), 0, +\infty, p_0)$ holds. When $l \rightarrow +\infty$, the last relation reduces to

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

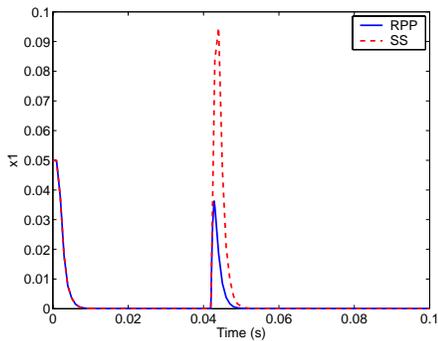


Fig. 1. System responses – State x_1

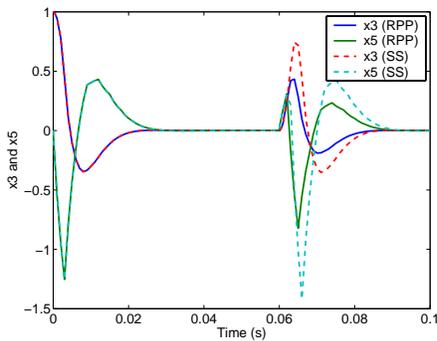


Fig. 2. System responses – States x_3 and x_5

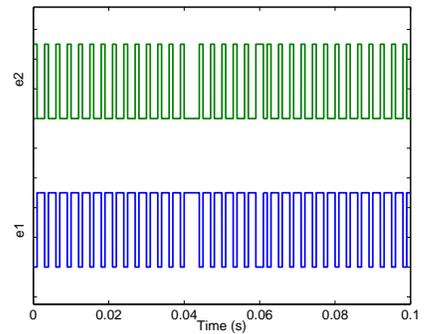


Fig. 3. RPP schedule

Theorem 1 demonstrates that the RPP strategy guarantees the performance improvements with respect to the static scheduling algorithm. The stability of the RPP scheduling algorithm directly follows from *Theorem 1* and is given the following corollary

Corollary 1: If R is positive definite and if the asymptotic stability of the global system \mathcal{S} (composed of systems $(S^{(j)})_{1 \leq j \leq N}$) is guaranteed by the static scheduling algorithm, then it is also ensured by the RPP scheduling algorithm.

Proof: When R is positive definite, then $J^{ss}(\tilde{x}(0), 0, +\infty, p_0)$ (respectively $J^{rpp}(\tilde{x}(0), 0, +\infty)$) is finite if and only if system \mathcal{S} is asymptotically stable. Knowing the asymptotic stability of the system scheduled using the static scheduling algorithm and using relation (23), the corollary is proved. ■

B. A numerical example

In order to illustrate the effectiveness of the RPP scheduling algorithm, the preceding example is considered. The constants $\varepsilon_x^{(1)} = 0.001$ and $\varepsilon_x^{(2)} = 0.01$ were chosen. The execution order sequence is $s = (s(0), s(1), s(2)) = (1, 1, 2)$, the optimal permutation is $\pi^* = (\pi^*(0), \pi^*(1), \pi^*(2)) = (2, 0, 1)$ and $o = o_{\pi^*} = (s(\pi^*(0)), s(\pi^*(1)), s(\pi^*(2))) = (2, 1, 1)$.

The global system responses corresponding states x_1 , x_3 and x_5 are respectively depicted in figures 1 and 2. The global system is started from the initial state $[0.05 \ 0 \ 1 \ 0 \ 0 \ 0]^T$. The two systems $S^{(1)}$ and $S^{(2)}$ “reach the equilibrium” respectively at $t = 0.012 \text{ s}$ and $t = 0.034 \text{ s}$. Consequently, the equilibrium indicators are set to 1. At $t = 0.419 \text{ s}$, system $S^{(1)}$ is severely disturbed. The conditions listed in line 2 of the listing of the RPP scheduling algorithm are fulfilled. The RPP algorithm reacts then to execute task $\tau^{(1)}$ instead of task $\tau^{(2)}$ at $t = 0.42 \text{ s}$, allowing to react better to this disturbance and to improve the control quality (as illustrated in figure 3). The same reactive pointer placement is performed at $t = 0.060 \text{ s}$, when the system $S^{(2)}$ is disturbed.

IV. CONCLUSIONS

In this paper, a new formulation of the problem of the optimal off-line scheduling of control tasks is proposed. This formulation is based on the H_2 performance criterion to statically allocate the computing resources according the intrinsic characteristics of the controlled systems. Using this

approach, the “sampling period” of each control task is automatically chosen. A plant state based feedback scheduling mechanism is proposed, enabling to enhance the control performance with respect to the optimal off-line scheduling algorithm.

REFERENCES

- [1] D. Seto, J. P. Lehoczky, L. Sha, and K. G. Shin, “On task schedulability in real-time control systems,” in *Proceedings of the 17th IEEE Real-Time Systems Symposium*, New York, USA, December 1996.
- [2] X. Liu, L. Sha, M. Caccamo, and G. Buttazzo, “Online control optimization using load driven scheduling,” in *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia, December 2000.
- [3] G. C. Buttazzo, G. Lipari, M. Caccamo, and L. Abeni, “Elastic scheduling for flexible workload management,” *IEEE Transactions on Computers*, vol. 51, no. 3, pp. 289–302, March 2002.
- [4] H. Rehbinder and M. Sanfridson, “Integration of off-line scheduling and optimal control,” in *Proceedings of the 12th Euromicro Conference on Real-Time Systems*, Stockholm, Sweden, June 2000.
- [5] P. Martí, C. Lin, S. Brandt, M. Velasco, and J. Fuertes, “Optimal state feedback based resource allocation for resource-constrained control tasks,” in *Proceedings of the 25th IEEE Real-Time Systems Symposium*, Lisbon, Portugal, December 2004.
- [6] 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.
- [7] 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.
- [8] D. Henriksson and A. Cervin, “Optimal on-line sampling period assignment for real-time control tasks based on plant state information,” in *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference*, Seville, Spain, December 2005.
- [9] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. Francis, “State-space solutions to the standard H_2 and H_∞ control problems,” *IEEE Transactions on Automatic Control*, vol. 34, no. 8, pp. 831–847, August 1989.
- [10] L. Xie, H. Zhou, and C. Zhang, “ H_2 optimal deconvolution of periodic channels: an LMI approach,” in *Proceedings of the 6th International Symposium on Signal Processing and its Applications*, Kuala-Lumpur, Malaysia, August 2001.
- [11] A. Bemporad and M. Morari, “Control of systems integrating logic, dynamics, and constraints,” *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.
- [12] P. P. Khargonekar, K. Poolla, and A. Tannenbaum, “robust control of linear time-invariant plants using periodic compensation,” *IEEE Transactions on Automatic Control*, vol. 30, no. 41, pp. 1088–1096, November 1985.
- [13] H. Shu and T. Chen, “State-space approach to discrete-time \mathcal{H}_2 -optimal control with a causality constraint,” *Systems and Control Letters*, vol. 26, no. 1, pp. 69–77, September 1995.