Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

A Design Methodology for Weakly-Hard Real-Time Control ⋆

Mongi Ben Gaid * **Daniel Simon** ** **Olivier Sename** ***

* *Institut Français du Pétrole, Rueil-Malmaison, France (e-mail: mongi.ben-gaid@ifp.fr)*
** *NeCS Project-Team, INRIA Rhône-Alpes, Montbonnot Saint Martin, France (e-mail: Daniel.Simon@inrialpes.fr)*
*** *Control Systems Department, Gipsa-Lab, INPG-CNRS UMR 5216 Saint Martin d'Hères, France (e-mail: olivier.sename@inpg.fr)*

Abstract: The problem of the integrated control and weakly-hard real-time scheduling is addressed. First, an abstract model of control tasks execution is introduced, allowing the establishment of a formal relationship linking control performance to deadline misses. Then, the notion of accelerable control task is introduced. An accelerable control task has the property that more executions are performed, better is the control performance. Thanks to this latter property, it becomes straightforward to design the control laws according to the average execution times of control tasks, and guaranteeing that in the worst-case scenario, the minimal allowable performance will be achieved. Based on Bellman optimality principle, sufficient conditions for a given control task to be accelerable are stated. A design method of optimal control laws for the weakly-hard execution model is then proposed.

Keywords: Control under computation constraints, optimal control, real-time scheduling

1. INTRODUCTION

During more than three decades, the computing power of embedded, general purpose and high performance processors has spectacularly progressed, principally, due to the ability to integrate an increasing number of transistors in a single ship, according to "Moore's law". However, this progress has been mainly obtained at the expense of an increase of the non-determinism. By non-determinism, we mean the variations of computing performance, measured in instructions per cycle, for a given processors family. For example, for a given task with constant and identical operations at each invocation, this non-determinism manifests itself by a significant gap between the average execution time and the worst-case execution time (WCET) of the task. This non-determinism mainly comes from the fact that the operation of modern processors heavily relies on predictions (which instructions have to be executed next, which data have to be accessed...), in order to achieve the maximal computing performance. When a prediction (for example, a prediction supplied by a branch predictor) is faulty, the throughput of the execution pipeline is dramatically reduced, lowering processor performance. When an instruction or a data that has to be loaded is not present in the cache, loading it from the main memory takes a much longer time. Modeling the interactions between caches, branch predictors, to cite a few elements, is very difficult (Lundqvist and Stenström (1999)). For that reason, the estimations of the WCET are becoming more and more pessimistic and difficult to obtain (Ferdinand et al. (2001)).

⋆ This work was supported by the French National Research Agency (ANR) project Safe-NeCS under grant No. ANR-ARA SSIA-NV-15

On the other side, state of the practice approaches (i.e. which are currently used in industry) for the real-time scheduling design of control tasks mainly rely on these WCET estimates for computing resources dimensioning. In fact, in these approaches, control tasks are considered as hard real-time tasks, characterized by a constant period, a WCET and a relative deadline (which is generally chosen equal to the period). However, in reality, control systems can tolerate a certain amount of deadline misses, if they occur occasionally, or if these misses are well characterized mathematically and taken into account in control design. For those reasons, control tasks may be situated in a category between hard real-time and soft real-time tasks. These observations motivated a surge of research in the frontier between control theory and real-time scheduling theory (Seto et al. (1996); Martí (2002); Lemmon et al. (2007)), leading to the introduction of new task models (Shih and Liu (1995)), schedulability constraints (Ramanathan and Hamdaoui (1995); Koren and Shasha (1995)) and scheduling algorithms (Eker et al. (2000); Simon et al. (2005); Ben Gaid et al. (2006)). In particular, the conceptual framework of weakly-hard real-time scheduling was introduced in (Bernat et al. (2001)). This framework encapsulates many previously introduced task models and schedulability constraints such as (m, k)-firm (Ramanathan and Hamdaoui (1995)) and skip-over (Koren and Shasha (1995)). It allows handling tasks that can tolerate a clearly specified number of missed deadlines during a window of time. For those reasons, this framework appears as a suitable approach allowing the design and implementation of control tasks based on their average execution times. In order to exploit this framework in a rigorous way, new control design approaches have to

be proposed. The design of such strategies is among the objectives of this paper.

Although the weakly-hard real-time scheduling paradigm received a significant interest, few works were dedicated to the problem of the control design under weakly-hard scheduling constraints. Ramanathan (1999) proposed the use of the (m, k)-firm scheduling concept to achieve a graceful degradation in a situation of processor overload. In this approach, the different jobs of a control task are classified into mandatory and optional. Mandatory jobs are guaranteed to complete before their deadlines. Optional jobs may either meet or miss their deadlines. The proposed associated control design methodology, which relies on the optimal linear quadratic periodic control theory, modifies the gains of the mandatory jobs in order to minimize the degradation which may result from executing the original control law in an overloaded processor. However, no control design methodology was proposed to compute the control law of the optional jobs (which simply maintain constant the previous controls). For that reason, this design method does not allow to exploit the successful executions of the optional instances. The problem of the weakly-hard real-time control may be related to the problems of control under unreliable communication links. Considering a linear quadratic Gaussian setting, Jia et al. (2007) addressed the problem of the joint controller design and packet drop pattern selection in order to optimize the control performance in a situation of network overload. Sinopoli et al. (2004) studied the problem of the optimal filtering with intermittent observations, assuming that packet losses verify a Bernoulli distribution. The dual control problem was studied in (Imer et al. (2006); Schenato et al. (2007)). In opposite to these approaches, the method developed in this paper does not require any assumption of the probabilistic distribution of the deadline misses or packet losses, and aims at achieving a guaranteed stability and performance at the worst case-situation.

The objective of this paper is to introduce a control design method allowing exploiting the advantages of the weakly-hard real-time scheduling concept. First, an abstract model, linking the plant dynamics and performance to its control task execution is proposed. Then the notion of accelerable task is introduced. Based on Bellman optimality principle, a generic control design methodology of accelerable control laws is then proposed. The specialization of this methodology to linear systems with quadratic cost is developed and illustrated.

The remaining of the paper organized as follows. Section 2 introduces the basic definitions, the considered control tasks execution model and the notion of accelerable control. In Section 3, a general method for designing accelerable control tasks is described. The application of these design principals to linear systems with quadratic performance index is investigated and illustrated in Section 4.

2. PROBLEM SETTING

2.1 Plant and performance index definition

Consider the control system described by the following difference equation

$$x(k+1) = f(k, x(k), u(k)), \tag{1}$$

where $x(k) \in \mathbb{R}^n$ represents the state, $u(k) \in \mathbb{R}^m$ represents the control input and f has an equilibrium point at the origin ($f(0,0) = 0$).

In order to simplify the notations, for any discrete instants k_a and k_b such that $k_a \leq k_b$, notations $x(k_a, k_b)$ and $u(k_a, k_b)$ represent respectively the states and input sequences $(x(k_a), \ldots, x(k_b))$ and $(u(k_a), \ldots, u(k_b))$. For any given instant k_i, the future values of the state $x(k)$ ($k > k_i$) are uniquely determined knowing k_i, k, $x(k_i)$ and $u(k_i, k-1)$. For that reason, and in order to simplify the notation, for any discrete instants k_a and k_b such that $k_b \geq k_a + 1$, $f(k_a, x(k_a), u(k_a, k_b))$ denotes the state reached by system (1) at instant $k_b + 1$, consecutive to the application of the control sequence $u(k_a, k_b)$ from instant k_a, and knowing the state $x(k_a)$ at instant k_a. A cost functional J is associated to system (1), and is defined by

$$J(x(0), u(0, \infty)) \triangleq \sum_{k=0}^{\infty} q(k, x(k), u(k)). \tag{2}$$

The expression of q will be specialized in Section 4. We will assume throughout this paper that q is chosen to ensure that when $J(x(0), u(0, \infty))$ is finite, then the controlled system (1) is uniformly asymptotically stable. In the following, $0_{n,m}$ denotes the $n \times m$ matrix whose elements are equal to zero and I_n the $n \times n$ identity matrix. Notations $\lfloor x \rfloor$ and $\lceil x \rceil$ represent respectively the floor and the ceiling functions of a x. We will denote by $J(x(k_a), u(k_a, k_b))$ the cost function corresponding to an evolution starting from state $x(k_a)$ at instant k_a to instant k_b.

$$J(x(k_a), u(k_a, k_b)) \triangleq \sum_{k=k_a}^{k_b} q(k, x(k), u(k)).$$

2.2 Control task execution

System (1) is controlled by a periodic control task τ, whose execution period is assigned according to average utilization considerations [1] . To simplify the discussions, it is assumed that this period is identical to the time period separating two consecutive discrete instants. However, due to the variations of its own execution time (up to its WCET), and also to the variations of the processor load, the jobs of task τ are not ensured to complete by their deadlines; the deadline of a job is defined as the release time of its subsequent job. Therefore, in order to characterize the jobs that complete by their deadlines, and the others that will miss their deadlines, and will be consequently aborted, the notion of execution sequence is introduced.

Definition 1. An execution sequence σ is an infinite sequence of elements of $\{0, 1\}$.

According to this definition, execution sequences are elements of $\{0, 1\}^{\mathbb{N}}$. An execution sequence σ is associated to each realization of task τ, and defined by

$$\begin{cases} \sigma(k) = 1 & \text{if the job activated at instant } k \text{ completes} \\ & \text{its execution by its deadline,} \\ \sigma(k) = 0 & \text{otherwise.} \end{cases}$$

[1] This may be performed using, for example, a feedback scheduling algorithm

Let $\mathcal{E}$ be the set valued function that associates to each execution sequence the invocation count (i.e. the discrete instant of activation) of the jobs that finish before their deadlines. Formally, $\mathcal{E}$ is defined by

$$\mathcal{E}(\sigma) \triangleq \{k \in \mathbb{N} \text{ such that } \sigma(k) = 1\}.$$

Fortunately, using a priority-based scheduling, and knowing the WCET of all the tasks that have priority over τ, it is possible to guarantee that selected jobs of τ will always meet their deadlines. This may be ensured using weakly-hard schedulability analysis techniques developed in Ramanathan (1999) and Bernat et al. (2001). In the following, it is assumed that task τ guarantees a (μ, κ)-constraint where μ and κ are two integers such that $\mu \leq \kappa$ (i.e. the deadlines of μ out of any κ consecutive jobs of τ are met). It is also assumed that this constraint is met by guaranteeing that the jobs whose invocation count k verifies

$$k = \left\lfloor \left\lceil \frac{k\mu}{\kappa} \right\rceil \frac{\kappa}{\mu} \right\rfloor \tag{3}$$

will always meet their deadlines. These jobs are called *mandatory* jobs. The other jobs, which are not guaranteed to complete by their deadlines, are called *optional* jobs. It has been proven in Ramanathan (1999) that when the jobs of a given task are classified according to (3), then the pattern of mandatory jobs will be κ-periodic. This means that if the job activated at instant k is mandatory, than any job activated at instant $k + i\kappa$, $(i \in \mathbb{N})$ is also mandatory. Consequently, when only these mandatory jobs are guaranteed to complete before their deadlines, the worst-case execution sequence γ that may be associated to τ is defined by

$$\begin{cases} \gamma(k) = 1 & \text{if } k = \left\lfloor \left\lceil \frac{k\mu}{\kappa} \right\rceil \frac{\kappa}{\mu} \right\rfloor, \\ \gamma(k) = 0 & \text{otherwise.} \end{cases} \tag{4}$$

When relation (3) is applied to impose the worst-case execution pattern of task τ, the corresponding worst-case execution sequence γ is guaranteed to be a κ-periodic execution sequence, verifying $\gamma(k) = \gamma(k + \kappa)$. When the respect of this worst-case execution pattern γ of task τ is ensured, the set of all possible executions of τ is denoted by $\mathcal{T}(\gamma)$ and defined as

$$\mathcal{T}(\gamma) \triangleq \left\{\sigma \in \{0,1\}^{\mathbb{N}} \text{ such that } \mathcal{E}(\sigma) \supseteq \mathcal{E}(\gamma)\right\}.$$

Let σ be an execution sequence and ℓ a time instant. We denote by $\sigma \bullet \ell$ the execution sequence defined by

$$\begin{cases} (\sigma \bullet \ell)(k) = \sigma(k) & \text{if } k \in \mathbb{N} \text{ and } k \neq \ell, \\ (\sigma \bullet \ell)(\ell) = 1 & \text{otherwise.} \end{cases}$$

The map $\bullet$ may be viewed as the generator of the set $\mathcal{T}(\gamma)$, i.e., for each sequence $\sigma \in \mathcal{T}(\gamma)$, there exists integers $k_1, k_2, \ldots, k_r$ such that $\sigma = (((\gamma \bullet k_1) \bullet k_2) \bullet \ldots) \ldots \bullet k_r$.

2.3 Notion of accelerable control

For any given control law $\xi(k)$ defined over $\mathbb{N}$, $\xi_\sigma(k)$ denotes the control input to the plant, taking into account execution constraints. Therefore, $\xi_\sigma(k)$ is defined by

$$\begin{cases} \xi_\sigma(k) = \xi(k) & \text{if } \sigma(k) = 1, \\ \xi_\sigma(k) = \xi_\sigma(k-1) & \text{otherwise.} \end{cases}$$

We are now ready to introduce the notion of accelerable control.

Definition 2. Assume that a control law $\xi(k)$ was defined, such that for all $x_0 \in \mathbb{R}^n$, the cost function $J(x_0, \xi_\gamma(0, \infty))$ corresponding to the worst-case execution sequence γ is finite (which ensures, according to the preliminary assumptions, the global asymptotic stability of system (1)). The control law $\xi(k)$ is called accelerable according to performance index (2) and worst-case execution sequence γ, if for all execution sequences σ_1 and σ_2 such that $\mathcal{E}(\gamma) \subseteq \mathcal{E}(\sigma_2) \subseteq \mathcal{E}(\sigma_1)$, for all $x_0 \in \mathbb{R}^n$, the associated cost functions satisfy $J(x_0, \xi_{\sigma_1}(0, \infty)) \leq J(x_0, \xi_{\sigma_2}(0, \infty))$.

A control task executing an accelerable control law will be called accelerable control task. An accelerable control task has the property that more executions are performed, better is the control performance. When used in conjunction with weakly-hard real-time scheduling design, an accelerable control task allows taking advantage of the extra computational resources that may be allocated to it, and to improve the control performance with respect to worst-case design methods. In practice, however, control laws designed using standard sampled-data control design methods are not necessarily accelerable. The following example illustrates this point.

Example 1. Consider the linearized model of a second order pendulum, described by

$$\dot{x}_c(t) = Ax_c(t) + Bu_c(t), \tag{5}$$

with $A = \begin{bmatrix} 0 & 1 \\ s\frac{g}{l} & -\frac{f_v}{ml^2} \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix}$ and $x_c(t) = \begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \end{bmatrix}$, where $\theta(t)$ represents the pendulum angle, l its length, m its weight, f_v the viscous friction coefficient, g the gravitational acceleration and $u(t)$ the control input. $s = -1$ (resp. $s = 1$) for the stable pendulum (resp. unstable pendulum). The numerical values of these parameters are $l = 1$, $m = 1$, $f_v = 1$, and $g = 9.81$.

The plant is controlled by a task τ. The estimated WCET of task τ is twice its average execution time. For that reason, the $(1, 2)$-firm constraint is associated to task τ. Using relation (4) determines the corresponding worst-case execution sequence $\gamma = (\gamma(0), \gamma(1), \gamma(2), \gamma(3), \ldots) = (1, 0, 1, 0, \ldots)$. The invocation count of mandatory jobs belongs to the set $\mathcal{E}(\gamma) = \{0, 2, 4, 6, 8, \ldots\}$.

An optimal sampled-data controller, corresponding to this worst-case execution sequence, and minimizing the cost function (2), was designed at the sampling period T_s=100 ms, based on the discretized cost function

$$q(k, x(k), u(k)) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ u(k) \end{bmatrix}^T Q \begin{bmatrix} x_1(k) \\ x_2(k) \\ u(k) \end{bmatrix},$$

where $Q = \begin{bmatrix} 9.9545 & 0.0857 & -0.0108 \\ 0.0857 & 0.7561 & 0.0371 \\ -0.0108 & 0.0371 & 0.0527 \end{bmatrix}$, $x(k) = x_c(k\frac{T_s}{2})$ and $u(k) = u_c(k\frac{T_s}{2})$ (assuming zero-order hold). Matrix Q was obtained through the discretization of a continuous cost function representing the design specification of the ideal controller. Taking into account the execution constraints defined by γ, the optimal control law u_γ that will be applied according to a worst-case hard-real time scheduling approach is defined by

$$\begin{cases} u_\gamma(k) = Kx(k) & \text{if } \gamma(k) = 1, \\ u_\gamma(k) = u_\gamma(k-1) & \text{otherwise.} \end{cases}$$

where $K = -[15.6596 \;\; 3.2416]$.

Starting from a given initial condition $x(0)$, we are interested in determining whether performing a single optional execution at instant $k = 1$ (i.e. by applying the execution sequence $\sigma = \gamma \bullet 1 = (1, 1, 1, 0, 1, 0, \ldots)$), based on the same invariant control gain K (i.e. by applying the control update $u(1) = Kx(1)$), will improve the control performance. Figures 1 and 2 provide an answer to this question. They show that performance improvements are state dependant. If the plant state $x(0)$ lies in the colored region of the state-space, then performing the execution of the optional job of instant $k = 1$ (where $\gamma(1) = 0$ but $\sigma(1) = 1$) leads to performance improvement. Otherwise, performing this single optional execution leads to performance degradation. For an accelerable control task, all the state space will be colored.

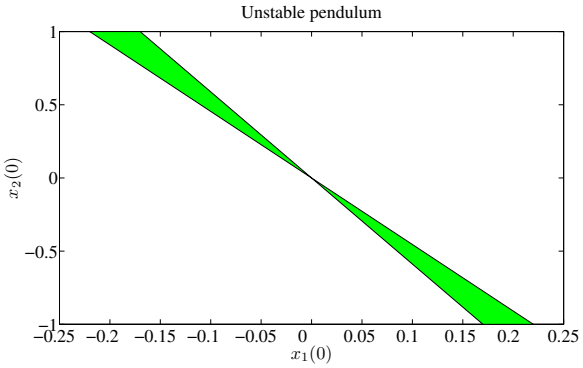

Fig. 1. Unstable pendulum: The green colored regions correspond to the states where performing a single optional execution at instant $k = 1$ lead to performance improvements

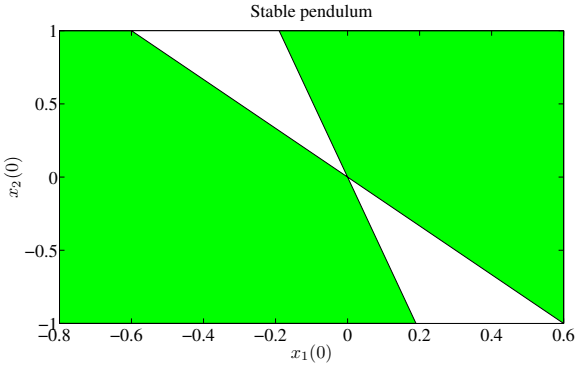

Fig. 2. Stable pendulum: The green colored regions correspond to the states where performing a single optional execution at instant $k = 1$ lead to performance improvements

3. APPLICATION OF BELLMAN OPTIMALITY PRINCIPLE TO THE DESIGN OF ACCELERABLE CONTROL TASKS

In this section, a general method for constructing accelerable control laws is presented. Let γ be a worst-case execution sequence determined according to (3). Let k_a and k_b two discrete instants such that $k_a \leq k_b$. Let $\mathcal{U}_\rho(k_a, k_b)$ be the set of admissible control inputs, defined between instants k_a and k_b, taking into account the resource constraints that are modeled by the execution sequence ρ. The set $\mathcal{U}_\rho(k_a, k_b)$ is formally defined as $\{u(k_a, k_b)$, such that $u(k) = u(k-1)$ if $\rho(k) = 0\}$.

In the remaining of this paper, we shall make the following assumption.

Assumption 1. For all $k \in \mathbb{N}$ and $x \in \mathbb{R}^n$,

$$\arg \min_{u(k,\infty) \in \mathcal{U}_{(\gamma \bullet k)}(k,\infty)} J(x, u(k, \infty)) \text{ exist.}$$

For any given $k \in \mathbb{N}$ and $x(k) \in \mathbb{R}^n$, an optimal control sequence corresponding to an evolution over an infinite horizon, starting at instant k from state $x(k)$, and taking into account the computation constraints defined by the execution sequence $\gamma \bullet k$, will be denoted as

$$u^*_{(\gamma \bullet k)}(k, \infty) \triangleq \arg \min_{u(k,\infty) \in \mathcal{U}_{(\gamma \bullet k)}(k,\infty)} J(x(k), u(k, \infty)). \tag{6}$$

An optimal solution of problem (6) is a control sequence $u^*_{(\gamma \bullet k)}(k, \infty) = (u^*_{(\gamma \bullet k)}(k), u^*_{(\gamma \bullet k)}(k+1), u^*_{(\gamma \bullet k)}(k+2), \ldots)$ that minimizes the cost function J, corresponding to an evolution over an infinite horizon, starting from state $x(k)$ at instant k, and assuming that:

- the job (mandatory or optional) activated at instant k will meet its deadline and update the plant,
- the subsequent computation constraints (from instant $k+1$ to ∞) are described following the worst-case execution sequence.

Based on the solutions of optimization problems (6), it is possible to construct, in a simple way, an accelerable control law. Let $u^\bullet(k)$ be the first element of the optimal control sequence $u^*_{(\gamma \bullet k)}(k, \infty)$:

$$u^\bullet(k) = u^*_{(\gamma \bullet k)}(k). \tag{7}$$

Strategy $u^\bullet(k)$ may be seen as a "robust control" approach against execution uncertainties satisfying the introduced weakly-hard model. It allows minimizing the cost function J for the "worst-case uncertainty" from the implementation. Under Assumption 1, strategy (7) provides a general method for constructing accelerable control laws. The following Theorem states the accelerability properties of strategy (7).

Theorem 1. Let γ be a worst-case execution sequence. Under Assumption 1, control law $u^\bullet(k)$, as defined in (7), is accelerable in accordance to (2) and γ.

Proof. The proof is made by construction. Let $\underline{\sigma}$ and $\overline{\sigma}$ two execution sequences such that $\mathcal{E}(\gamma) \subseteq \mathcal{E}(\underline{\sigma}) \subseteq \mathcal{E}(\overline{\sigma})$. Let $x_0 \in \mathbb{R}^n$. We have to prove that:

$$J(x_0, u^\bullet_{\overline{\sigma}}(0, \infty)) \leq J(x_0, u^\bullet_{\underline{\sigma}}(0, \infty)). \tag{8}$$

Let $(k_\ell)_{\ell \in \mathbb{N}^*}$ be the sequence of time instants where $\underline{\sigma}(k_\ell) = 0$ and $\overline{\sigma}(k_\ell) = 1$, arranged in increasing order. These discrete instants represent the invocation counts of optional jobs that complete successfully (i.e. before their deadlines), in the situation where their execution is described by $\overline{\sigma}$. Let $k_0 = -1$. For all $\ell \in \mathbb{N}^*$, k_ℓ is formally defined by

$$k_\ell \triangleq \left\{ \min_{k \in \mathbb{N}, k > k_{\ell-1}} k \text{ such that } \underline{\sigma}(k) = 0, \overline{\sigma}(k) = 1 \right\}. \tag{9}$$

Let $(\sigma_\ell)_{\ell\in\mathbb{N}}$ the sequence of execution sequences such that $\sigma_0 = \underline{\sigma}$ and for all $\ell \in \mathbb{N}^*$

$$\sigma_\ell \triangleq \sigma_{\ell-1} \bullet k_\ell. \tag{10}$$

For any given k_ℓ, let k_ℓ^+ the time instant corresponding to the subsequent execution of σ_ℓ, and defined as

$$k_\ell^+ \triangleq \left\{ \min_{k\in\mathbb{N},k>k_\ell} k \text{ such that } \sigma_\ell(k) = 1 \right\}.$$

An illustration of the introduced variables is given in Figure 3.

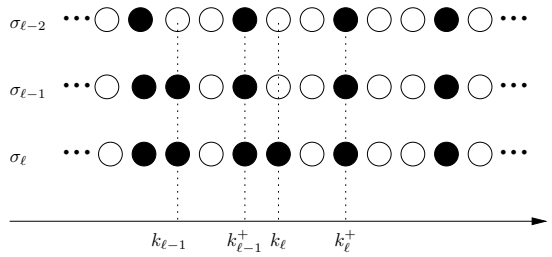

Fig. 3. "Convergence" of sequence $(\sigma_\ell)_{\ell\in\mathbb{N}}$ from $\underline{\sigma}$ to $\overline{\sigma}$. The filled bullets represent instants where control variables may be updated, in opposite to the empty bullets

Remarking that $\underline{\sigma} = \sigma_0$ and that $\overline{\sigma} = \lim_{\ell\to\infty} \sigma_\ell$, then the proof of (8) is reduced to showing that

$$J(x_0, u^\bullet_{\sigma_\ell}(0,\infty)) \leq J(x_0, u^\bullet_{\sigma_{\ell-1}}(0,\infty)). \tag{11}$$

However, examining execution sequences σ_ℓ and $\sigma_{\ell-1}$, we remark that the only difference is located at instant k_ℓ. For that reason, $J(x_0, u^\bullet_{\sigma_{\ell-1}}(0, k_\ell - 1)) = J(x_0, u^\bullet_{\sigma_\ell}(0, k_\ell - 1))$, and the proof of (11) is reduced to establishing that for all $x \in \mathbb{R}^n$,

$$J(x, u^\bullet_{\sigma_\ell}(k_\ell,\infty)) \leq J(x, u^\bullet_{\sigma_{\ell-1}}(k_\ell,\infty)). \tag{12}$$

To simplify the notation, define for all $\rho \in \mathcal{T}(\gamma)$,

$$V^*(x,k,\rho) \triangleq \min_{u(k,\infty)\in\mathcal{U}_\rho(k,\infty)} J(x, u(k,\infty)).$$

From the definition of $u^\bullet$, and remarking $u^*_{\sigma_\ell}(k_\ell,\infty) = u^\bullet(k_\ell,\infty)$, it is easy to see that

$$J(x, u^\bullet_{\sigma_\ell}(k_\ell,\infty)) = V^*(x,k_\ell,\sigma_\ell)$$

By definition,

$$V^*(x,k_\ell,\sigma_\ell) = \min_{u(k_\ell,\infty)\in\mathcal{U}_{\sigma_\ell}(k_\ell,\infty)} J(x, u(k_\ell,\infty)).$$

Using Bellman equation,

$$\begin{aligned} V^*(x,k_\ell,\sigma_\ell) = \min_{u(k_\ell,k_\ell^+-1)\in\mathcal{U}_{\sigma_\ell}(k_\ell,k_\ell^+-1)} \big\{ & J(x, u(k_\ell, k_\ell^+ - 1)) \\ & + V^*(f(x, u(k_\ell, k_\ell^+ - 1)), k_\ell^+, \sigma_\ell) \big\}. \end{aligned}$$

Consequently, for any control sequence $u(k_\ell, k_\ell^+ - 1) \in \mathcal{U}_{\sigma_\ell}(k_\ell, k_\ell^+ - 1)$,

$$\begin{aligned} V^*(x,k_\ell,\sigma_\ell) \leq & J(x, u(k_\ell, k_\ell^+ - 1)) \\ & + V^*(f(x, u(k_\ell, k_\ell^+ - 1)), k_\ell^+, \sigma_\ell). \end{aligned} \tag{13}$$

In particular, since $\mathcal{U}_{\sigma_{\ell-1}}(k_\ell, k_\ell^+ - 1) \subseteq \mathcal{U}_{\sigma_\ell}(k_\ell, k_\ell^+ - 1)$, applying the control sequence $u^\bullet_{\sigma_{\ell-1}}(k_\ell, k_\ell^+ - 1)$ in (13) gives

$$\begin{aligned} V^*(x,k_\ell,\sigma_\ell) \leq & J(x, u^\bullet_{\sigma_{\ell-1}}(k_\ell, k_\ell^+ - 1)) \\ & + V^*(f(x, u^\bullet_{\sigma_{\ell-1}}(k_\ell, k_\ell^+ - 1)), k_\ell^+, \sigma_\ell). \end{aligned} \tag{14}$$

Since for all $k > k_\ell$, $\sigma_\ell(k) = \sigma_{\ell-1}(k)$, then for all z, $V^*(z, k_\ell^+, \sigma_\ell) = V^*(z, k_\ell^+, \sigma_{\ell-1})$. In particular, for $z = \bar{z} = f(x, u^\bullet_{\sigma_{\ell-1}}(k_\ell, k_\ell^+ - 1))$,

$$V^*(\bar{z}, k_\ell^+, \sigma_\ell) = V^*(\bar{z}, k_\ell^+, \sigma_{\ell-1}). \tag{15}$$

Using (14) and (15), we get

$$\begin{aligned} V^*(x,k_\ell,\sigma_\ell) \leq & J(x, u^\bullet_{\sigma_{\ell-1}}(k_\ell, k_\ell^+ - 1)) \\ & + V^*(f(x, u^\bullet_{\sigma_{\ell-1}}(k_\ell, k_\ell^+ - 1)), k_\ell^+, \sigma_{\ell-1}). \end{aligned} \tag{16}$$

Finally, remarking that the right-hand side of the previous inequality is equal to $J(x, u^\bullet_{\sigma_{\ell-1}}(k_\ell,\infty))$, and that

$$J(x, u^\bullet_{\sigma_\ell}(k_\ell,\infty)) = V^*(x,k_\ell,\sigma_\ell),$$

then (16) reduces to $J(x, u^\bullet_{\sigma_\ell}(k_\ell,\infty)) \leq J(x, u^\bullet_{\sigma_{\ell-1}}(k_\ell,\infty))$. This inequality is a direct consequence of Bellman optimality principle. It shows that the cost corresponding to an evolution starting at state x from instant k_ℓ over an infinite horizon and using the execution sequence σ_ℓ is better than the cost obtained using $\sigma_{\ell-1}$. As shown in Figure 3, the only difference between execution sequences $\sigma_{\ell-1}$ and σ_ℓ is the possibility to update the value of the control input at instant k_ℓ using σ_ℓ. Adding the opportunity to update the control law at instant k_ℓ (together with using the control law $u^\bullet$) can only lead to an improvement of the cost function, since in worst-case, the optimal cost corresponding to the use of σ_ℓ will be equal to that of $\sigma_{\ell-1}$, by simply maintaining the previous control input constant. □

4. ACCELERABLE LQR DESIGN FOR LTI SYSTEMS

In this section, we restrict our attention to LTI plants. We illustrate how the previous general design principles may be applied to LTI systems. Under Assumption 1, which will be satisfied if reachability properties (as defined for linear time-varying systems) are fulfilled and when the cost functions are appropriately chosen, it becomes possible to compute off-line a closed form of $u^\bullet$, which will be a time-varying state feedback. In the remaining of this section, the following assumption is made.

Assumption 2.

$$f(k, x(k), u(k)) = Ax(k) + Bu(k),$$

$$q(k, x(k), u(k)) = \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}^T Q \begin{bmatrix} x(k) \\ u(k) \end{bmatrix},$$

where $Q = \begin{bmatrix} Q_1 & Q_{12} \\ Q_{12}^T & Q_2 \end{bmatrix} \geq 0$ and $Q_2 > 0$.

In the following, the methodology underlying the design of the accelerable control law $u^\bullet$ according to the design principles of the previous section is developed. The following definitions are first introduced. Let γ be a worst-case execution sequence defined according to (4). Let d_γ be the distance between the current job k and the next mandatory job in the execution sequence γ. Formally, d_γ is defined by

$$d_\gamma(k) \triangleq \min_{k_m} \{k_m - k, k_m > k \text{ and } \gamma(k_m) = 1\}. \tag{17}$$

Let

$$\Phi(i,k) \triangleq \begin{bmatrix} A^{i-k} & \sum_{j=0}^{i-k-1} A^j B \\ 0_{m,n} & I_m \end{bmatrix},$$

and consider the following "virtual model":

$$\tilde{A}_\gamma(k) \triangleq A^{d_\gamma(k)-k}, \tag{18}$$

$$\tilde{B}_\gamma(k) \triangleq \sum_{i=0}^{d_\gamma(k)-k-1} A^i B, \tag{19}$$

$$\tilde{Q}_\gamma(k) \triangleq \begin{cases} Q & \text{if } d_\gamma(k)-k=1, \\ Q + \sum_{i=k+1}^{d_\gamma(k)-1} \Phi(i,k)^T Q \Phi(i,k) & \text{if } d_\gamma(k)-k>1. \end{cases} \tag{20}$$

Finally, assume that $\tilde{Q}_\gamma(k)$ is partitioned as

$$\tilde{Q}_\gamma(k) = \begin{bmatrix} \tilde{Q}_{1_\gamma}(k) & \tilde{Q}_{12_\gamma}(k) \\ \tilde{Q}^T_{12_\gamma}(k) & \tilde{Q}_{2_\gamma}(k) \end{bmatrix}.$$

The virtual model (17)(18)(19)(20) captures the evolution of system (1) and its associated cost function (2) under Assumptions 1 and 2, when its control inputs over an infinite horizon starting at instant k are subject to the computation constraints defined by the execution sequence $\gamma \bullet k$. Let $k \in \mathbb{N}$ and $(q_i^k)_{i\in\mathbb{N}}$ be the sequence of mandatory instants following k, with $q_0^k = k$. The correspondence between the state and the cost function of model (1) with the virtual model is described by the following equations

$$x(q_{i+1}^k) = \tilde{A}_\gamma(q_i^k)x(q_i^k) + \tilde{B}_\gamma(q_i^k)u(q_i^k),$$

and

$$J(x(q_i^k), u_{\gamma\bullet q_0^k}(q_i^k, q_{i+1}^k - 1)) = \eta(q_i^k)^T \tilde{Q}_\gamma(q_i^k)\eta(q_i^k),$$

where

$$\eta(q_i^k) = \begin{bmatrix} x(q_i^k) \\ u_{\gamma\bullet q_0^k}(q_i^k) \end{bmatrix}.$$

The virtual model may be seen as a sub-sampled model of the plant (1) determining the value of the state at the different time instants (q_i^k) following instant k. It allows the easy computation of the accelerable control law $u^\bullet$ as shown in the following corollary.

Corollary 1. Let γ be a worst-case execution sequence defined according to (4). Under assumptions 1 and 2, the state feedback control law defined by

$$u^\bullet(k) = -L_\gamma(k)x(k) \tag{21}$$

is accelerable in accordance to (2) and γ, where

$$\begin{aligned} L_\gamma(k) = &\left(\tilde{Q}_{2_\gamma}(k) + \tilde{B}_\gamma^T(k)\tilde{S}_\gamma(d_\gamma(k))\tilde{B}_\gamma(k)\right)^{-1} \\ &\times \left(\tilde{B}_\gamma^T(k)\tilde{S}_\gamma(d_\gamma(k))\tilde{A}_\gamma(k) + \tilde{Q}^T_{12_\gamma}(k)\right). \end{aligned} \tag{22}$$

Matrices $\tilde{S}_\gamma(k)$ are defined for as the steady state solutions of the following Riccati equation

$$\begin{aligned} \tilde{S}_\gamma(k) = &\tilde{A}_\gamma^T(k)\tilde{S}_\gamma(d_\gamma(k))\tilde{A}_\gamma(k) + \tilde{Q}_{1_\gamma}(k) \\ &- \left(\tilde{A}_\gamma^T(k)\tilde{S}_\gamma(d_\gamma(k))\tilde{B}_\gamma(k) + \tilde{Q}_{12_\gamma}(k)\right) \\ &\times \left(\tilde{B}_\gamma^T(k)\tilde{S}_\gamma(d_\gamma(k))\tilde{B}_\gamma(k) + \tilde{Q}_{2_\gamma}(k)\right)^{-1} \\ &\times \left(\tilde{B}_\gamma^T(k)\tilde{S}_\gamma(d_\gamma(k))\tilde{A}_\gamma(k) + \tilde{Q}^T_{12_\gamma}(k)\right), \end{aligned} \tag{23}$$

where $d_\gamma(k)$, $\tilde{A}_\gamma(k)$, $\tilde{B}_\gamma(k)$ and $\tilde{Q}_\gamma(k)$ are respectively defined in (17), (18), (19) and (20).

Proof. This corollary is a direct application of Theorem 1. The obtained expression of $u^\bullet$ results from the fact that the virtual model captures a sub-sampled evolution of system (1) and its associated cost function (2), over an infinite horizon starting at instant k, when its control inputs are subject to the constraints defined by the execution sequence $\gamma \bullet k$. □

Corollary 1 provides a control design methodology allowing computing accelerable control laws using optimal control theory. In this approach, the gains of mandatory instances are designed using periodic optimal control theory, which ensures the stability and minimal acceptable performance in the worst-case situations. These gains, as well as those of the optional instances, are computed as a function of the steady-state solution of the periodic Riccati equation (23). Due to the periodicity of execution sequence γ, at most κ control gains need to be computed off-line in order to allow the implementation of the accelerable control strategy $u^\bullet$.

Remark 1. If plant model (1) includes a one sample delay, the control gains of the mandatory jobs will be identical to those obtained by Ramanathan (1999). The computation optional jobs gains is an extension of Ramanathan (1999). Therefore, the control law of Corollary 1 will be accelerable with respect to the method of Ramanathan (1999).

Example 2. Recall the unstable pendulum in Example 1 of Section 2.3. Assume that the (1,3)-firm constraint is associated to task τ. The corresponding worst-case execution sequence is described by the periodic execution sequence $\gamma = (1,0,0,1,0,0\ldots)$. An accelerable control design, based on the worst-case execution sequence γ was performed (according to Corollary 1). The accelerable control law $u^\bullet$ is defined by

$$u^\bullet(k) = -L_\gamma(k)x(k),$$

where

$$\begin{cases} L_\gamma(k) = [17.4441 \quad 3.6386] & \text{if } k \mod 3 = 0, \\ L_\gamma(k) = [19.8526 \quad 4.2192] & \text{if } k \mod 3 = 1, \\ L_\gamma(k) = [23.4618 \quad 5.1269] & \text{if } k \mod 3 = 2. \end{cases}$$

In this particular example, the influence of sampling period reduction manifests itself essentially on disturbance rejection abilities. Figure 4 illustrates this point. It compares the output of the system $y(t) = \theta(t)$ in three different situations:

- the situation where all the optional jobs are not executed (0 % hit), this corresponds to the state of the practice hard real-time scheduling design, where resources are dimensioned according to worst-case utilization situations,
- the situation where all optional jobs are triggered for execution (according to the weakly-hard real-time scheduling philosophy) and when 50 % of them meet their deadlines and update the control, according to a random Bernoulli probability distribution, with success probability 1/2 (50 % hit),
- the best-case situation where all the optional jobs are triggered and meet their deadlines (100 % hit).

The disturbance is a periodic rectangular pulse signal with amplitude 12 and whose pulse width is equal to 5 % of the period. Figure 4 shows that significant improvements in control performance result from the weakly-hard real-time design, with respect to the worst-case real-time design. The random disturbances are better rejected. These improvements are due to the fact that in the accelerable weakly-hard design, optional instances are executed when

possible, with conveniently computed and compensated control gains. Finally, Figure 5 shows the cumulative cost

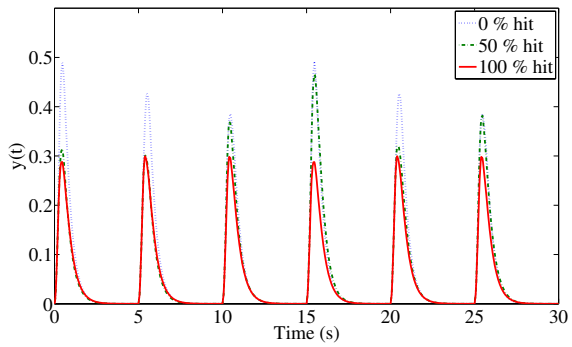

Fig. 4. Plant output for different optional jobs hit ratios

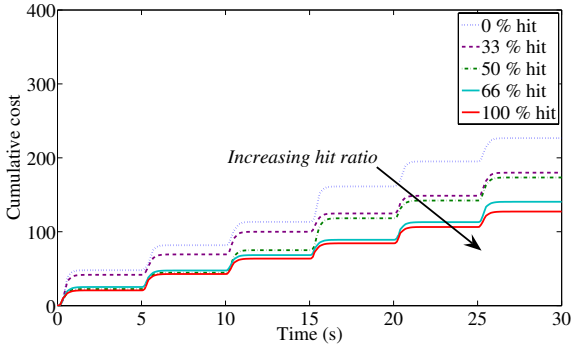

Fig. 5. Cumulative cost for different optional jobs hit ratios

functions that are associated to the previous simulations, including the additional situations where 33 % and 66 % of the optional jobs meet their deadlines (according to a random Bernoulli probability distribution, with respective success probabilities 1/3 and 2/3). They illustrate the intuitive notion behind accelerability: more optional instances are executed, better is the control performance.

5. CONCLUSION

A methodology for control and weakly-hard real-time scheduling co-design was proposed. It aims at achieving efficient resource utilization, through control and scheduling co-design according to average resources considerations, while guaranteeing worst-case performance requirements. Using this approach, the analysis and design of the worst-case situation (corresponding to the mandatory instants) is significantly simplified, and may be undertaken using state of the art methods. The design of the control law of the optional instants is undertaken subsequently, allowing performing an asynchronous control with guaranteed cost. A potential application of weakly-hard scheduling methods is their use with conjunction of feedback scheduling algorithms, where transient overruns may often occur. An interesting extension enabling this application is the development of both variable sampling period (Robert et al. (2007)) and accelerable controllers, taking into account the uncertainties resulting from a weakly-hard real-time implementation.

REFERENCES

M. Ben Gaid, A. Çela, Y. Hamam, and C. Ionete. Optimal scheduling of control tasks with state feedback resource allocation. In *Proc. 2006 American Control Conf.*, Minneapolis, USA, June 2006.

G. Bernat, A. Burns, and A. Llamosi. Weakly-hard real-time systems. *IEEE Tran. Computers*, 50(4), 2001.

J. Eker, P. Hagander, and K.-E. Årzén. A feedback scheduler for real-time control tasks. *Control Engineering Practice*, 8(12):1369–1378, 2000.

C. Ferdinand, R. Heckmann, M. Langenbach, F. Martin, M. Schmidt, H. Theiling, S. Thesing, and R. Wilhelm. Reliable and precise WCET determination for a real-life processor. In *Proc. First International Workshop on Embedded Software*, London, UK, Oct. 2001.

O. C. Imer, S. Yüskel, and Başar T. Optimal control of LTI systems over unreliable communication links. *Automatica*, 42(9):1429–1439, 2006.

N. Jia, Y.-Q. Song, and F. Simonot-Lion. Graceful degradation of the quality of control through data drop policy. In *Proc. Europ. Control Conf.*, Kos, Greece, July 2007.

G. Koren and D. Shasha. Skip-over: Algorithms and complexity for overloaded systems that allow skips. In *Proc. 16th IEEE Real-Time Sys. Symp.*, Pisa, Italy, Dec. 1995.

M. Lemmon, T. Chantem, X. Hu, and M. Zyskowski. On self-triggered full information h-infinity controllers. In *Proc. Hybrid Sys.: Computation and Control*, Apr. 2007.

T. Lundqvist and P. Stenström. Timing anomalies in dynamically scheduled microprocessors. In *Proc. 20th IEEE Real-Time Sys. Symp.*, Phoenix, USA, Dec. 1999.

P. Martí. *Analysis and Design of Real-Time Control Systems with Varying Control Timing Constraints*. PhD thesis, Technical University of Catalonia, 2002.

P. Ramanathan. Overload management in real-time control applications using (m, k)-firm guarantee. *IEEE Trans. Parallel and Distri. Sys.*, 10(6):549–559, 1999.

P. Ramanathan and M. Hamdaoui. A dynamic priority assignment technique for streams with (m, k)-firm deadlines. *IEEE Tran. Computers*, 44(12):1443–1451, 1995.

D. Robert, O. Sename, and D. Simon. A reduced polytopic LPV synthesis for a sampling varying controller : experimentation with a T inverted pendulum. In *Proc. Europ. Control Conference*, Kos, Greece, July 2007.

L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S.S. Sastry. Foundations of control and estimation over lossy networks. *Proc. IEEE*, 95(1):163–187, 2007.

D. Seto, J. P. Lehoczky, L. Sha, and K. G. Shin. On task schedulability in real-time control systems. In *Proc.17th IEEE Real-Time Sys. Sym.*, New York, USA, Dec. 1996.

W. K. Shih and J. W.-S. Liu. Algorithms for scheduling imprecise computations with timing constraints to minimize maximum error. *IEEE Tran. Computers*, 44(3): 466–471, 1995.

D. Simon, D. Robert, and O. Sename. Robust control/scheduling co-design: application to robot control. In *Proc. 11th IEEE Real-Time and Embedded Technology and Applica. Symp.*, San Francisco, USA, Mar. 2005.

B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry. Kalman filtering with intermittent observations. *IEEE Tran. Automatic Control*, 49(9):1453–1464, 2004.