Design and Stability Analysis for Anytime Control via Stochastic Scheduling

Luca Greco, *Member, IEEE*, Daniele Fontanelli, *Member, IEEE*, and Antonio Bicchi, *Fellow, IEEE*

***Abstract*—In this paper, we consider the problem of designing controllers for linear plants to be implemented in embedded platforms under stringent real-time constraints. These include preemptive scheduling schemes, under which the execution time allowed for control software tasks is uncertain. In a conservative Hard Real-Time (HRT) design approach, only a control algorithm that (in the worst case) is executable within the minimum time slot guaranteed by the scheduler would be employed. In the spirit of modern Soft Real-Time (SRT) approaches, we consider here an "anytime control" design technique, based on a hierarchy of controllers for the same plant. Higher controllers in the hierarchy provide better closed-loop performance, while typically requiring longer execution time. Stochastic models of the scheduler and of algorithm execution times are used to infer probabilities that controllers of different complexity can be executed at different periods. We propose a strategy for choosing among executable controllers, maximizing the usage of higher controllers, which affords better exploitation of the computational platform than the HRT design while guaranteeing stability (in a suitable stochastic sense).**

Results on the robustness with respect to uncertainties affecting the scheduler model, and on bumpless transfer for tracking problems are also reported. Simulation results on the control of two prototypical mechanical systems show that performance is substantially enhanced by our anytime control technique w.r.t. worst case-based scheduling.

***Index Terms*—Anytime algorithms, embedded control, stochastic scheduling, switched systems.**

I. Introduction

A general tendency can be observed in embedded systems towards implementation of a great variety of concurrent real-time tasks on the same platform, thus reducing the overall HW cost and development time. Among such tasks, those implementing control algorithms are usually highly time critical, and have traditionally imposed very conservative scheduling approaches, whereby execution time is allotted statically. This makes the overall architecture extremely rigid, hardly reconfigurable for additions or changes of components, and often underperforming.

Modern multitasking Real-Time Operating Systems (RTOSs), running, e.g., on embedded Electronic Control Units (ECUs) in the automotive domain, schedule their tasks dynamically, adapting to varying load conditions and Quality of Service (QoS) requirements. Real-time preemptive algorithms, such as, e.g., Rate Monotonic (RM) and Earliest Deadline First (EDF) [1]–[3] can suspend the execution of a task in the presence of requests by other, higher priority tasks.

To make a given set of tasks schedulable and to avoid deadline misses, conservative assumptions are made in so-called Hard Real-Time systems [1], including Worst Case Execution Time (WCET) estimates for tasks. These entail underexploitation of the computational platform and, ultimately, cost inefficiencies. Conversely, when the computational power budget is given and fixed (as common in industrial practice), then control algorithms may have to be drastically simplified to be computable within the allotted time. This clearly reflects in a degradation of the overall performance of the ensuing closed-loop system.

Substantial performance improvement would be gained if less conservative usage could be made of the platform. Indeed, assuming that the RTOS guarantees a minimum time $\tau_{\min}$ for the execution of the control task, it is often the case that, for most of the CPU cycles, a time $\tau > \tau_{\min}$ could be made available for control.

A current trend in embedded system design is to relax hard schedulability constraints and introduce "softer" models of computation (viz. "resource reservations" [4], "weakly hard" [5] and "firm" [6] RTOSs). Here, occasional deadline misses are tolerated for tasks whose execution time may largely vary (due e.g., to data-driven branching and/or the use of caches and pipelines). Instead of using gross WCET bounds, the random distribution pattern and average of such misses are quantified by QoS metrics, in terms of which design constraints are typically set [7].

In this paper, we propose a strategy to design and schedule linear controllers in a soft RT environment, so that the limits of current practice are overcome and better exploitation can be obtained of the same resources.

A. Anytime Control Algorithms

The key idea is to design controllers which can be implemented so that a useful result is guaranteed whenever the algorithm is run for at least $\tau_{\min}$; however, better results can be provided if longer times are allowed.

Manuscript received August 19, 2008; revised May 06, 2009; accepted June 14, 2010. Date of publication July 15, 2010; date of current version March 09, 2011. This work was supported in part by the European Commission under FP7 with Contract CHAT – Control of Heterogeneous Automation Systems: Technologies for scalability, reconfigurability and security IST FP7-2007-2-224428 and Contract HYCON2 Network of Excellence, Highly-complex and networked control systemsICT FP7–2009–5–257462. Recommended by Associate Editor J. Lygeros.

L. Greco is with the DIIMA, University of Salerno, Fisciano (SA) 84084, Italy and with the LSS – Supélec, Gif sur Yvette 91192, France (e-mail lucagrc@gmail.com; lgreco@ieee.org).

D. Fontanelli is with the Interdepartmental Research Center "E. Piaggio," University of Pisa, Pisa, Italy and with the Department of Information Engineering and Computer Science (DISI), University of Trento, Povo (TN), Trento 38100, Italy, (e-mail: fontanelli@disi.unitn.it).

A. Bicchi is with the Interdepartmental Research Center "E. Piaggio", University of Pisa, Pisa 56100, Italy and with the IIT – Istituto Italiano di Tecnologia, Genova, Genova, Italy (e-mail: bicchi@ing.unipi.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2010.2058497

The idea of anytime algorithms is borrowed from the field of *imprecise computation*, that has been proposed in the real-time systems literature [8], [9]. The characteristic of anytime algorithms is to always return an answer on demand; however, the longer they are allowed to compute, the better (e.g., more precise) an answer they will return. Thus, an anytime algorithm can be interrupted prematurely, still providing a valid result and improving the output accuracy as the available time increases. A periodic task is split in a *mandatory* part and one or more *optional* parts. Functionally critical subtasks are considered in the mandatory part. If all mandatory parts of a set of tasks are schedulable, the *feasible mandatory constraint* is satisfied [8].

In digital filter design [10], this philosophy has been pursued by decomposing the full-order filter in a series of lower order filters whose execution is prioritized. Execution of code implementing the first block is always guaranteed within $\tau_{\min}$; code for blocks in the cascade is then executed sequentially, until a deadline event takes over. The latest computed block output is used as the anytime filter output. The overall performance of the filter was shown in [10] to be superior to the conservative solution of always using only the first filter block.

To adopt the anytime approach in the control domain, a classical monolithic control task should be replaced by a hierarchy of control tasks of increasing complexity, each providing a correspondingly increasing performance of the controlled system. For instance, the simplest control task in the hierarchy, could be designed to guarantee only stability of the closed loop system, while whenever the scheduler provides "surplus" time, other more sophisticated control algorithms could be executed to obtain better quality of control.

However, application of the anytime algorithm idea to control is much more challenging than it may superficially appear. The main conceptual roadblock is that, as opposed to most anytime computation and filtering algorithms, anytime controllers interact in feedback with dynamic systems, which fact entails issues such as:

- *Switched System Performance:* unpredictable preemption events introduce stochastic switching among different closed-loop systems, which can subvert naïve expectations—e.g., switching between stabilizing controllers may well result in overall instability. More generally, closed-loop performance is strongly influenced by switching;
- *Practicality:* implementation of both control and scheduling algorithms must be numerically accurate, yet very simple and noninvasive, not to contradict the very nature of the limited-resource, embedded control problem;
- *Hierarchical Design:* the design of a set of controllers as progressive approximations towards a given target design does not typically provide the desired performance hierarchy. Indeed, performance of closed-loop systems are not trivially related to how close controller approximations are to the target, as it is e.g., in filter design;
- *Modularity:* the computational structure of control algorithms should be inherited through the hierarchy levels, so that the computation of higher controllers in the hierarchy exploits results of computations executed for lower controllers. Although this property is not strictly required, it can greatly enhance effectiveness of anytime control.

We will discuss these issues further in the rest of the paper.

B. Prior Work and Outline

A first attempt to apply the anytime computation paradigm to control is reported in [11], [12], where standard system reduction methods (balanced truncation or modal decomposition) are used to decompose a target controller in simpler ones. If the simplified controllers are individually stabilizing, stability under switching is guaranteed by [11], [12] only under the implicit assumption that a long enough (but unquantified) dwelling time is allowed by the scheduler between switches. Very recently a specular approach, using a sequence of increasingly more accurate models of the open loop system, has been presented in [13].

On the other hand, the substantial literature on *switching system* stability (see e.g., [14]–[17] and references therein) provides much inspiration and ideas for the problem at hand, but few results can be used directly. For instance, application of the important results of [18] would provide state-space realizations of different stabilizing controllers such that the overall closed-loop systems would remain stable under *any* switching law. Unfortunately, however, the method is thought for a different application, and assumes that all controllers are designed by the internal-model approach, thus having the same (rather heavy) computational complexity. Most importantly, at each switching instant, a state-space transformation has to be applied, whose complexity is comparable to that of the controllers themselves. By the same practicality argument, algorithms for *switched system* stabilization (such as e.g., [19]–[22]) requiring the computation of complex functions of the state to ascertain which subsystem can be activated next time, are not applicable to the anytime control problem as we consider it here.

The thread of work closest to ours is probably that related to Firm Real Time Systems (FRTSs) [5], [6]. In a series of papers [7], [23]–[25], Lemmon and co-workers consider performance of Networked Control Systems (NCSs) in a FRTS framework, introduce a stochastic model to describe the task dropout process, and provide a general QoS constraint encompassing classical metrics such as average- or window-based dropout measures.

Probabilistic modeling of real-time systems is by now a widely accepted approach to avoid overconservatism of deterministic (WCET-based) models, to which an ample and growing literature [26]–[28] is devoted. Within stochastic models of RT systems, the use of Markov chains (cf. e.g., [25], [29], [30]) is one of the most promising avenues to accurately compute the response time distribution (and deadline miss probabilities) of different tasks in systems ranging from fixed-priority (e.g., RM) to dynamic-priority (e.g., EDF).

In this paper, which builds upon [31], we also adopt a Markov Chain model to describe the sequence of time slots allotted by a scheduler for the execution of a control task (for methods to infer the parameters of the stochastic model of the scheduler, see e.g., [28], [29]). Our model differs from the one used in the FRTSs literature cited above, as we define our probabilities on the space of execution times rather than on deadline misses. A more substantial difference, however, is that we regard the scheduler characteristics to be a given in our problem, rather than a design objective. In our anytime control setup, different control subroutines can be alternatively activated in different periods to control the same plant, and we develop a switching policy that affords better exploitation of the computational platform while guaranteeing overall closed-loop stability. As the system resulting from switching under stochastic constraints is a Markov Jump Linear System (MJLS), stability is studied in the probabilistic sense of "Almost Sure" (AS) stability [32], [33].

II. Background Material

Let $\Sigma = (A, B, C, D)$ be a given linear, discrete time, invariant SISO plant and let $\Gamma_i, i \in I \triangleq \{1, 2, \ldots, n\}$, be a family of controllers for Σ such that the feedback connection of Σ with Γ_i is asymptotically stable for all i. Let the closed-loop system thus obtained be denoted as Σ_i and let its dynamics be

$$x(t+1) = \widehat{A}_i x(t). \tag{1}$$

The set of controllers is assumed to provide a hierarchy of performance and complexity. In other terms, we assume that application of controller i provides better closed-loop performance than controller j if $i > j$, and that its software implementation typically requires longer execution time. Which controller will be actually executed at each period depends on the time allotted by the RTOS scheduler and on the execution time of different controllers, both nondeterministic quantities. The main tool used in this paper to address stability of the switching system ensuing from probabilistic scheduling models is the theory of Almost Sure stability.

A. Almost Sure Stability

In this section, a brief review of results on Almost Sure stability (AS-stability) for discrete-time systems is reported following [33]. Consider the discrete-time Markov Jump Linear System (MJLS)

$$x(t+1) = A_{\sigma(t)} x(t) \tag{2}$$

where $x \in \mathbb{R}^N$, $A_i \in \mathbb{R}^{N \times N}$, $i \in I = \{1, \ldots, n\}$, and $\sigma(t)$ is a finite-state discrete-time homogeneous irreducible aperiodic (FSHIA) Markov chain taking values in the finite state space $L_\sigma = I$, with transition probability matrix $S = (s_{ij})_{n \times n}$, $s_{ij} \triangleq \Pr\{\sigma(t+1) = j \mid \sigma(t) = i\}$, and with initial probability measure $\pi_\sigma(0)$. The evolution of the probability distribution $\pi_\sigma(t)$ of the process σ at time t is given by

$$\pi_\sigma(t+1) = \pi_\sigma(t) S. \tag{3}$$

We will denote by $\overline{\pi}_\sigma$ the unique invariant probability distribution (i.p.d.) of the irreducible and aperiodic Markov chain, corresponding to the steady-state probability distribution for the process σ (i.e., $\lim_{t \to \infty} \pi_\sigma(t) = \overline{\pi}_\sigma$ for any $\pi_\sigma(0)$).

Definition 1: The MJLS (2) is said exponentially almost surely stable (AS-stable) if there exists $\mu > 0$ such that, for any $x_0 \in \mathbb{R}^N$ and any initial distribution $\pi_\sigma(0)$, the following condition holds:

$$\Pr\left\{\limsup_{t \to \infty} \frac{1}{t} \ln \|x(t)\| \leq -\mu\right\} = 1.$$

The following sufficient condition for AS-stability was proven in [32]:

Theorem 1 (1–Step Average Contractivity): [32] If there exists a matrix norm $\|\cdot\|$, such that

$$\xi_1 = \prod_{i \in I} \|A_i\|^{\overline{\pi}_{\sigma_i}} < 1 \tag{4}$$

then the MJLS (2) is AS-stable.

Inequality (4) can be interpreted as an average contractivity of the state norm over a one-step horizon. A condition less restrictive than (4), and involving the average contractivity over a multi-step horizon has been presented in [33]. Namely, a new MJLS, called "lifting of period m", is associated to the MJLS (2). Such a system represents the sampling of the original one at time instants hm, $h \in \mathbb{N}$, and its stability properties mirror those of the original system. More precisely, the lifted version of period m of system (2) is defined by

$$\widetilde{x}_{h+1} = \widetilde{A}_{\widetilde{\sigma}(h)} \widetilde{x}_h$$

with $\widetilde{x}_h = x_{mh}$, $\widetilde{\sigma}(h) = [\sigma(mh), \ldots, \sigma(mh+m-1)]$, $\widetilde{A}_{\widetilde{\sigma}(h)} = A_{\sigma(mh+m-1)} A_{\sigma(mh+m-2)} \cdots A_{\sigma(mh)}$. Moreover, $\widetilde{\sigma} = \{\widetilde{\sigma}(h)\}_{h \in \mathbb{N}}$ is a Markov process taking values in $\widetilde{I} = I^m$ and characterized as follows: for $\widetilde{\imath} = (i_1, i_2, \ldots, i_m) \in \widetilde{I}$ and $\widetilde{\jmath} = (j_1, j_2, \ldots, j_m) \in \widetilde{I}$, the transition probability matrix $\widetilde{S}$ has elements $\widetilde{s}_{\widetilde{\imath}\widetilde{\jmath}} \triangleq \Pr\{\widetilde{\sigma}(h+1) = \widetilde{\jmath} \mid \widetilde{\sigma}(h) = \widetilde{\imath}\}$ given by $\widetilde{s}_{\widetilde{\imath}\widetilde{\jmath}} = s_{i_m j_1} \prod_{k=1}^{m-1} s_{j_k j_{k+1}}$. This process has i.p.d. given by $\widetilde{\pi}_{\widetilde{\sigma}_{\widetilde{\imath}}} = \prod_{k=1}^{m-1} s_{i_k i_{k+1}} \overline{\pi}_{\sigma_{i_1}}$.

The 1–step average contractivity condition applied to the lifted system yields

Theorem 2 (m–Step Average Contractivity): [33] If

$$\xi_m = \prod_{\widetilde{\imath} \in \widetilde{I}} \|\widetilde{A}_{\widetilde{\imath}}\|^{\widetilde{\pi}_{\widetilde{\sigma}_{\widetilde{\imath}}}} < 1 \tag{5}$$

then the MJLS (2) is AS-stable.

The importance of condition (5) is related to the fact that for increasing values of m it provides a sequence of sufficient conditions and, most importantly, to the following result

Theorem 3: [33] System (2) is AS-stable if and only if $\exists m \in \mathbb{N}$ such that condition (5) holds.

III. Stochastic Modeling of the Scheduling Process

We consider a single-processor platform with a multitasking RTOS, and a periodic control task with period T_g. We further assume that control inputs and outputs are time-triggered and synchronized, i.e., measurements are acquired at the beginning of each period and control inputs are released at the end. As a consequence, the control task is not affected by jitter, while the constant unit delay can be easily accounted for directly in the design of the individual controllers [34]. Assume the period T_g to be divided in $l < \infty$ time slices of length ΔT, and that each controller Γ_j is implemented by a homonymous software subroutine. Both the time allotted by the scheduler to the control task, and the execution times of any subroutine, are finite multiples of ΔT. Let $L_\eta \triangleq \{\eta_1, \ldots, \eta_l\}$, with $\eta_i = l_i \Delta T$, $l_i \in \{l_{\min}, \ldots, l_{\max}\}$ and $\eta_1 = l_{\min} \Delta T = \tau_{\min}$.

Let the time allotted to the control task during the t-th sampling period be described by the discrete random variable $\gamma(t)$ taking on values in the set L_η. A simple stochastic description of the random sequence $\{\gamma(t)\}_{t\in\mathbb{N}}$ can be given in terms of an independently and identically distributed (i.i.d.) process, with probability distribution denoted by $\overline{\pi}_\gamma = [\overline{\pi}_{\gamma_1}, \ldots, \overline{\pi}_{\gamma_l}]$. Motivated by the stochastic model of RT systems in [29], [30], we adopt in what follows a more general model based on Markov chains. Accordingly, $\gamma(t)$ will denote a FSHIA Markov chain taking values in L_η, with transition probability matrix P and i.p.d. $\overline{\pi}_\gamma$.

Since it is assumed that no deterministic information is available in advance on the effective time allowance for the control task, subroutines Γ_j must be computed sequentially, i.e., the computation of Γ_i cannot start until the computation of Γ_{i-1} has terminated. Therefore, the execution time from Γ_1 to Γ_j is modeled as a discrete random variable T^j defined on L_η, with probability distribution $\Pr\left\{T^j = \eta_i\right\} \triangleq \overline{\pi}_{T_i^j}$, with $0 \le \overline{\pi}_{T_i^j} \le 1$ and $\sum_i \overline{\pi}_{T_i^j} = 1$. The (time-independent) probability distribution of T^j is described by $\overline{\pi}_{T^j} = [\overline{\pi}_{T_1^j}, \ldots, \overline{\pi}_{T_l^j}]$.

We now combine the probabilistic models above into a stochastic description of the highest-index executable controller in each period. This will be referred to as the (unconditioned) scheduler process τ. Indeed, τ represents the highest index of a schedulable controller, i.e., the controller such that its execution time is the largest among those shorter than the available time. Assume first that the simplest controller Γ_1 (whose worst-case execution time is $WCET_1$) is considered to be mandatory in the anytime scheme, and that the *feasible mandatory constraint* $WCET_1 \le \tau_{\min} = \eta_1$ is satisfied. Let the random process $\tau(t)$ taking on values in $L_\tau \triangleq \{\tau_1, \ldots, \tau_n\}$ be defined as "$\tau(t) = \tau_i$ iff in the t-th period all controllers $\Gamma_j, j \le i$ but no controller $\Gamma_k, k > i$ can be executed." Because controllers are computed sequentially, this is equivalent to stating that "$\tau(t) = \tau_i$ iff in the t-th period Γ_i can be executed but Γ_{i+1} can not.".

Let $\pi_{\tau_i}(t) \triangleq \Pr\{\tau(t) = \tau_i\}$. From $\pi_{\tau_i}(t) = \sum_j \Pr\{\tau(t) = \tau_i \mid \gamma(t) = \eta_j\}\, \pi_{\gamma_j}(t)$, it follows:

$$\pi_{\tau_i}(t) = \sum_j \Pr\left\{T^i \le \eta_j \cap T^{i+1} > \eta_j\right\} \pi_{\gamma_j}(t).$$

To express $\Pr\left\{T^i \le \eta_j \cap T^{i+1} > \eta_j\right\}$ in terms of distributions $\overline{\pi}_{T^i}$, let us write

$$\begin{aligned}&\Pr\left\{T^i \le \eta_j\right\}\\ &= \Pr\left\{\left(T^i \le \eta_j \cap T^{i+1} \le \eta_j\right) \cup \left(T^i \le \eta_j \cap T^{i+1} > \eta_j\right)\right\}\\ &= \Pr\left\{T^i \le \eta_j \cap T^{i+1} \le \eta_j\right\} + \Pr\left\{T^i \le \eta_j \cap T^{i+1} > \eta_j\right\}\end{aligned}$$

because the two events are mutually exclusive. Recalling that controllers are computed sequentially, $T^{i+1} \le \eta_j$ implies $T^i \le \eta_j$, hence $\Pr\left\{T^i \le \eta_j \cap T^{i+1} \le \eta_j\right\} = \Pr\left\{T^{i+1} \le \eta_j\right\}$ and $\Pr\left\{T^i \le \eta_j \cap T^{i+1} > \eta_j\right\} = \Pr\left\{T^i \le \eta_j\right\} - \Pr\left\{T^{i+1} \le \eta_j\right\}$. Defining the cumulative distribution of T^i as $\kappa_{T^i} = [\overline{\pi}_{T_1^i}, \overline{\pi}_{T_1^i} + \overline{\pi}_{T_2^i}, \ldots, \sum_{k=1}^{l} \overline{\pi}_{T_k^i}]$, we can write $\Pr\left\{T^i \le \eta_j \cap T^{i+1} > \eta_j\right\} = \left(\kappa_{T^i} - \kappa_{T^{i+1}}\right)_j$. The relation between the stochastic processes γ and τ is then given by

$$\pi_\tau(t) = \pi_\gamma(t) T_{\gamma\tau} \tag{6}$$

with

$$T_{\gamma\tau} = \begin{bmatrix} \kappa_{T^1} - \kappa_{T^2} \\ \vdots \\ \kappa_{T^{n-1}} - \kappa_{T^n} \\ \kappa_{T^n} \end{bmatrix}^T$$

a stochastic matrix. If $\gamma(t)$ is a Markov chain, then $\pi_\tau(t)$ is time dependent, though not necessarily a Markov chain.

Remark 1: The assumption $WCET_1 \le \eta_1$ can be easily dropped. Indeed, if $WCET_1 > \eta_1$, it is sufficient to add to L_τ the event τ_0, with the meaning "$\tau(t) = \tau_0$ iff in the t-th period no controllers can be executed. " In this case, matrix $T_{\gamma\tau}^T$ is modified by adding on its top a row $\kappa_{T^0} - \kappa_{T^1}$, where $\kappa_{T^0} = [1, \ldots, 1]$ is the cumulative distribution that no controller can be executed.

Remark 2: In our previous work [31], a deterministic model of the controller execution time was considered. This can be regarded as a particular case of the present treatment, where $\overline{\pi}_{T^i}$ is chosen as a vector having a 1 in j-th position if $\eta_j = WCET_i$, and 0 elsewhere. Moreover, if L_η is limited to the set of all the WCETs, namely $L_\eta = \{WCET_1, \ldots, WCET_n\}$, then $T_{\gamma\tau}$ becomes the identity matrix and $\tau(t)$ coincides with the FSHIA Markov chain $\gamma(t)$.

IV. Problem Formulation

Given a set of controllers Γ_i, the associated closed loop dynamics $\widehat{A}_i$, and a probabilistic description of the process $\tau(t)$ modeling the maximum schedulable controller, a degree of freedom is left to the control task designer to make an explicit choice of the controller to be actually executed in each period. We will refer to such choice as the *switching policy*, which is defined as a map $s : \mathbb{N} \to I, t \mapsto s(t)$ determining the upper bound $i \le s(t)$ to the index i of the controller to be executed at time t. In other terms, at time tT_g, the control task computes controllers $\Gamma_1, \Gamma_2, \ldots$ until Γ_s, unless a deadline event occurs forcing it to provide only Γ_τ, the highest controller computed before the deadline.

Application of a switching policy s to a set of feedback systems Σ_i, $i \in I$ under a given scheduler τ generates a switched linear system (Σ_i, τ, s) which, under general hypotheses, is a stochastic JLS with a conditioned i.p.d. $\overline{\pi}_{\tau|s}$. As an example, the most conservative policy is to set $s(t) \equiv 1$, i.e., forcing always the execution of the simplest controller Γ_1, regardless of the probable availability of more computational time ($\overline{\pi}_{\tau|s} = [1, 0, \ldots, 0]$). If the feasible mandatory constraint $WCET_1 \le \eta_1$ is satisfied, this (non-switching) policy guarantees stability of the resulting closed loop system.

On the opposite, a "greedy" strategy would set $s(t) \equiv n$, which leads to computing Γ_τ for all t (hence $\overline{\pi}_{\tau|s} = \overline{\pi}_\tau$). Although this policy attempts at maximizing the utilization of the most performing controller, it is well known that switching arbitrarily among asymptotically stable systems Σ_i may easily result in an unstable behavior [35]. A sufficient condition for the greedy switching policy to provide an AS-stable system is provided by Theorem 1, where $A_i = \widehat{A}_i$ and $\overline{\pi}_\sigma = \overline{\pi}_\tau$. Notice that here we assume that all closed-loop systems have the same number of states, which coincide with the sum of the number of states of the plant and of the largest controller (the actual arrangement of the state vectors for closed-loop systems is illustrated in detail in the implementation Section VII).

If this condition is not verified, it is important to investigate whether there exist scheduling policies which can ensure AS-stability other than the over-conservative non-switching choice. To do so, assume that the set

$$\overline{\Pi} = \left\{ \overline{\pi} \mid \prod_{i \in I} \|\widehat{A}_i\|^{\overline{\pi}_i} < 1 \right\}$$

is not empty for some matrix norm. Given a scheduler i.p.d. $\overline{\pi}_\tau$, in order for a compatible and stabilizing conditioned i.p.d. $\overline{\pi}_{\tau|s} \in \overline{\Pi}$ to exist, further constraints have to be satisfied. Indeed, let $\overline{\pi}_d \in \overline{\Pi}$. For a switching policy to exist which can alter a given scheduler probability distribution $\overline{\pi}_\tau$ into $\overline{\pi}_{\tau|s} = \overline{\pi}_d$, the following must hold:

$$\overline{\pi}_{d_n} \leq \overline{\pi}_{\tau_n} \quad (C.1)$$

$$\overline{\pi}_{d_{n-1}} \leq \overline{\pi}_{\tau_{n-1}} + (\overline{\pi}_{\tau_n} - \overline{\pi}_{d_n}) \quad (C.2)$$

$$\vdots$$

$$\overline{\pi}_{d_1} \leq \overline{\pi}_{\tau_1} + (\overline{\pi}_{\tau_2} - \overline{\pi}_{d_2}) + \cdots + (\overline{\pi}_{\tau_n} - \overline{\pi}_{d_n}) \quad (C.n)$$

where $\overline{\pi}_\tau = [\overline{\pi}_{\tau_1}, \ldots, \overline{\pi}_{\tau_n}]$, and $\overline{\pi}_d = [\overline{\pi}_{d_1}, \ldots, \overline{\pi}_{d_n}]$.

Inequalities $(C.1)$–$(C.n)$ take into account the fact that no switching law can alter the scheduler so as to give more computational time to control tasks than it is made available by the scheduler. Furthermore, constraints $(C.2)$–$(C.n)$ model the fact that the probability $\overline{\pi}_{d_i}$ of the i-th controller can be increased only at the expenses of a reduction of the probabilities $\overline{\pi}_{d_j}, j > i$ of more complex controllers.

Definition 2: Given a set of matrices $\widehat{A}_i$, $i \in I$, and a scheduler i.p.d. $\overline{\pi}_\tau$, the set of feasible AS-stabilizing scheduling probabilities is defined as $\overline{\Pi}_d = \{\overline{\pi}_d\}$, with

$$D2.1) \quad \prod_{i=1}^{n} \left\| \widehat{A}_i \right\|^{\overline{\pi}_{d_i}} < 1$$

$$D2.2) \quad 0 \leq \overline{\pi}_{d_i} \leq 1$$

$$D2.3) \quad \sum_{i=1}^{n} \overline{\pi}_{d_i} = 1$$

$$D2.4) \quad \overline{\pi}_{d_i} \leq \overline{\pi}_{\tau_i} + \sum_{j=i+1}^{n} \overline{\pi}_{\tau_j} - \sum_{j=i+1}^{n} \overline{\pi}_{d_j}.$$

Assuming that $\overline{\Pi}_d \neq \emptyset$, it is natural to consider a Quality of Control (QoC) metric within the set of feasible stabilizing schedules. Based on the hypothesis that controllers are hierarchically ordered by their closed-loop performance, so that use of controller Γ_i provides better results than Γ_j, $j < i$, a natural QoC metric for the problem at hand can be simply given as

$$J(\overline{\pi}_d) = \sum_{i=1}^{n} d_i^2 \overline{\pi}_{d_i} \quad (7)$$

i.e., the second moment of a random variable $d \in \{d_1, \ldots, d_n\}$ with i.p.d. $\overline{\pi}_d$, where d_i is the performance index associated with Γ_i, $0 < d_i < d_j$, $i < j$.

In conclusion, we state the following problem

Problem 1 (Optimal Switching Policy (OSP) Problem): For a given plant Σ, a set of controllers Γ_i, with associated performance index d_i and probabilistic execution time T^i, and a probabilistic scheduler process τ, find a switching policy $s(t)$ that maximizes $J(\pi_{\tau|s})$, subject to $\pi_{\tau|s} \in \Pi_d$.

V. Independent Stochastic Switching Policy

In this section, we tackle problem 1 above by introducing a switching law which is itself stochastic, and is based on the concept of a *conditioning* Markov chain. The stochastic properties of the scheduler process and conditioning chain interact to produce a resulting switched system. To study such interaction, we will make use of the operations of *merging* and *aggregating* stochastic processes.

The merging of two finite state Markov chains $\alpha(t)$ and $\beta(t)$, defined on the state spaces L_α and L_β, respectively, is the stochastic process $\alpha\beta(t)$ defined on $L_{\alpha\beta} \triangleq L_\alpha \times L_\beta$ such that $\alpha\beta(t) = (\alpha(t), \beta(t))$. The following holds:

Theorem 4: For two independent, FSHIA Markov chains $\alpha(t)$ and $\beta(t)$ with transition probability matrices $P_\alpha = ({}^\alpha p_{ij})_{n \times n}$ and $P_\beta = ({}^\beta p_{ij})_{l \times l}$, and initial probability distributions $\pi_\alpha(0)$ and $\pi_\beta(0)$, let $\overline{\pi}_\alpha$ and $\overline{\pi}_\beta$ denote their respective (unique) invariant probability distributions. Then, for the merging $\alpha\beta(t)$, it holds

i) $\alpha\beta(t)$ is a FSHIA Markov chain whose statistics are given by the transition probability matrix $P_{\alpha\beta} = ({}^{\alpha\beta} p_{ij})_{nl \times nl} = P_\alpha \otimes P_\beta$ and by the initial probability distribution $\pi_{\alpha\beta}(0) = \pi_\alpha(0) \otimes \pi_\beta(0)$ ($\otimes$ denotes the Kronecker product);

ii) the evolution of the chain $\alpha\beta(t)$ is given by

$$\pi_{\alpha\beta}(t) = \pi_\alpha(t) \otimes \pi_\beta(t) = (\pi_\alpha(0) \otimes \pi_\beta(0)) \, P_{\alpha\beta}^t \quad (8)$$

with $t \in \mathbb{N}$. Moreover, $\pi_{\alpha\beta}(t)$ converges to the unique i.p.d.

$$\overline{\pi}_{\alpha\beta} = \overline{\pi}_\alpha \otimes \overline{\pi}_\beta \quad (9)$$

for any initial distribution $\pi_{\alpha\beta}(0)$.

Proof: See the Appendix. ■

Consider further a stochastic process $\rho(t)$ defined on the finite state space L_ρ and a function $g : L_\rho \to G$ mapping L_ρ in another finite state space G. The *aggregation* of ρ with respect to g is the stochastic process $\rho^\star$ defined on the quotient space of L_ρ by the equivalence relation $\rho_i \equiv \rho_j$ iff $g(\rho_i) = g(\rho_j)$. Notice that the aggregated process is not necessarily Markovian even if the original process is.

Coming back to the stochastic processes describing the scheduling and execution of controllers, consider an independent FSHIA Markov chain $\sigma(t)$ defined on a finite state space $L_\sigma = \{\sigma_1, \ldots, \sigma_n\}$. We will associate a switching law to the stochastic process $\sigma(t)$ in the following explicit sense:

Independent Stochastic Switching Policy (ISSP).

At the t–th period, the control task produces the results of the k-th controller Γ_k iff

1) $\tau(t) \geq \tau_k$ and $\sigma(t) = \sigma_k$; or
2) $\sigma(t) \geq \sigma_k$ and $\tau(t) = \tau_k$.

The fact that process $\tau(t)$ may not be a Markov chain implies that the merged process $\tau\sigma(t) = (\tau(t), \sigma(t))$ is also not a Markov chain in general. However, the probability distribution $\pi_{\tau\sigma}(t)$ of $\tau\sigma(t)$ is linearly related to the distribution $\pi_{\gamma\sigma}(t)$ of the merged Markov chain $\gamma\sigma(t)$. Indeed, from the independence of the random variables $\tau(t)$ and $\sigma(t)$ $\forall t \in \mathbb{N}$, by the mixed product rule we can write

$$\begin{aligned}\pi_{\tau\sigma}(t) &= \pi_\tau(t) \otimes \pi_\sigma(t) = (\pi_\gamma(t) T_{\gamma\tau}) \otimes \pi_\sigma(t)\\ &= (\pi_\gamma(t) \otimes \pi_\sigma(t))\,(T_{\gamma\tau} \otimes I_n)\\ &= \pi_{\gamma\sigma}(t)\,(T_{\gamma\tau} \otimes I_n)\end{aligned} \tag{10}$$

with I_n the identity matrix.

We identify states of $\tau\sigma$ resulting in the execution of the same controller by introducing the aggregating function $g : L_{\tau\sigma} \to I$ with $g(\tau_i, \sigma_j) \triangleq \min\{i, j\}$. The switching law choice above amounts to setting the conditioned probability distribution $\pi_{\tau|\sigma}$ equal to $\pi^\star(t)$, which is the probability distribution of the aggregated process $(\tau\sigma)^\star$. It can be easily verified that the evolution of $\pi^\star(t) = [\pi_1^\star(t), \ldots, \pi_n^\star(t)]$ is given by

$$\pi^\star(t) = (\pi_\tau(t) \otimes \pi_\sigma(t))\, H \tag{11}$$

with $H \in \{0,1\}^{n^2 \times n}$ such that $H = [H_1^T, H_2^T, \ldots, H_n^T]^T$ and

$$H_i^T = \left[\begin{array}{c|c} I_{i,i} & \begin{matrix}0_{i-1,n-i}\\ 1_{1,n-i}\end{matrix} \\ \hline 0_{n-i,i} & 0_{n-i,n-i}\end{array}\right].$$

Remark 3: Because of the linear mappings (10) and (11) the process $(\tau\sigma)^\star$ admits an i.p.d. to which each initial distribution of type $(\pi_\gamma(0) \otimes \pi_\sigma(0))\,(T_{\gamma\tau} \otimes I_n)\,H$ converges. In particular, we have

$$\overline{\pi}^\star = (\overline{\pi}_\tau \otimes \overline{\pi}_\sigma)\, H \tag{12}$$

with $\overline{\pi}_\tau = \overline{\pi}_\gamma T_{\gamma\tau}$. Hence the contractivity condition of the AS-stability analysis applies to $\overline{\pi}^\star$.

Remark 4: It can be observed that the choice of a switching law by a Markov chain $\sigma(t)$ independent from the scheduler process $\tau(t)$ is not the most general possible choice. However, besides simplifying the analysis considerably, this choice has the advantage of not requiring on-line computations. Indeed, an arbitrary realization of the process $\sigma(t)$ can be computed off-line and used as a switching law $s(t)$, which fact is crucial from a practicality point of view.

A. One-Step Average Contractivity Condition

Based on results of the previous section, we seek a solution of Problem 1 with a structure such as in (12). The conditioned i.p.d. $\overline{\pi}_{\tau|\sigma}$, which for notational simplicity we will denote henceforth as $\overline{\pi}^\star = \overline{\pi}_{\tau|\sigma}$, can be written more explicitly as

$$\overline{\pi}_i^\star = \sum_{(\tau_h,\sigma_k)\in\chi_i} \overline{\pi}_{\tau_h}\overline{\pi}_{\sigma_k} \tag{13}$$

where $\chi_i = \{(\tau_h, \sigma_k) \in L_{\tau\sigma} \mid g(\tau_h, \sigma_k) = i\}$.

It actually turns out that the structure of $\overline{\pi}^\star$ described in (13), resulting from the choice of an independent conditioning chain, simplifies the formulation of the synthesis problem substantially. Indeed, it can be proven by simple if lengthy arguments (see Lemma 2 in the Appendix), that constraints $D2.2$, $D2.3$, and $D2.4$ in Definition 2 are automatically satisfied by an i.p.d. $\overline{\pi}^\star$ as in (13). Furthermore, for such $\overline{\pi}^\star$, constraint $D2.1$ can be rewritten (with the proviso that $\|\widehat{A}_i\| \neq 0\ \forall i$) as

$$\begin{aligned}\ln\left(\prod_{i=1}^n \left\|\widehat{A}_i\right\|^{\overline{\pi}_i^\star}\right) &= \ln\left(\prod_{(\tau_h,\sigma_k)\in L_{\tau\sigma}} \left\|\widehat{A}_{g(\tau_h,\sigma_k)}\right\|^{\overline{\pi}_{\tau_h}\overline{\pi}_{\sigma_k}}\right)\\ &= \sum_{k=1}^n \overline{\pi}_{\sigma_k} \sum_{h=1}^n \overline{\pi}_{\tau_h} \ln\left(\left\|\widehat{A}_{g(\tau_h,\sigma_k)}\right\|\right) < 0.\end{aligned}$$

By introducing the QoC vector $c_d \in \mathbb{R}^n$, $c_{d_i} = d_i^2$, such that $J(\overline{\pi}^\star) = \sum_{i=1}^n c_{d_i}\overline{\pi}_i^\star = c_d\overline{\pi}^{\star T}$, using (13) and rearranging terms, the OSP problem restricted to the ISSP class can be stated as a classical linear programming problem in the unknown $\overline{\pi}_\sigma$:

OSP Problem

$$\begin{aligned}&\max_{\overline{\pi}_\sigma}\ J(\overline{\pi}^\star) = \max_{\overline{\pi}_\sigma}\ \overline{\pi}_\tau M_d\, \overline{\pi}_\sigma^T\\ &1)\quad \overline{\pi}_\tau M_c \overline{\pi}_\sigma^T \le -\varepsilon < 0\\ &2)\quad 0 \le \overline{\pi}_{\sigma_i} \le 1\\ &3)\quad \sum_{i=1}^n \overline{\pi}_{\sigma_i} = 1\end{aligned}$$

where $(M_c)_{ij} = \ln\left(\left\|\widehat{A}_{\min(i,j)}\right\|\right)$ and $(M_d)_{ij} = c_{d_{\min(i,j)}}$. Notice that $\varepsilon > 0$ indicates here a desired contractivity margin, while the use of non-strict inequalities in 2) implies that a non-irreducible Markov chain $\sigma(t)$ is accepted as a possible solution.

It is also worth noting that a necessary condition for OSP to be feasible is that $\exists \widehat{A}_i$ such that $\|\widehat{A}_i\| < 1$. This may easily not be the case for an arbitrary matrix norm. On the other hand, if a matrix norm for which $\|\widehat{A}_1\| < 1$ is chosen, and $WCET_1 \le \tau_{\min}$ (which is reasonable to assume), then feasibility is guaranteed. While finding such a norm is always possible if $\|\widehat{A}_1\|$ is Schur, the choice of such an $\widehat{A}_1$–adapted norm tends to bias solutions of OSP towards using the simplest controller more often. This in turn tends to produce low performance index J. Therefore, it can be useful to consider a more general position of the scheduling problem, which is described next.

B. Multi-Step Average Contractivity Condition

The advantage of using m-step contractivity is illustrated by the following Lemma:

Lemma 1: If $\exists i \in I$ such that $\widehat{A}_i$ is Schur, then for any given matrix norm $\|\cdot\|$, $\exists m \in \mathbb{N}$ and $\widetilde{\imath} \in \widetilde{I} = I^m$ such that the solution set is not empty, i.e.

$$\overline{\Pi} = \left\{\overline{\pi} \mid \prod_{\widetilde{\imath}\in\widetilde{I}} \|\widehat{A}_{\widetilde{\imath}}\|^{\overline{\pi}_{\widetilde{\imath}}} < 1\right\} \neq \emptyset.$$

Proof: If $\widehat{A}_i$ is Schur, then $\exists m \in \mathbb{N}$ such that $\|\widehat{A}_i^m\| < 1$. Let $\imath_i = \{i, \ldots, i\} \in \widetilde{I}$ and $\widehat{A}_{\widetilde{\imath}} = \widehat{A}_i^m$. Hence $\exists\,\overline{\epsilon} > 0$ such that, for all probability distributions with $\overline{\pi}_{\imath_i} = 1 - \epsilon$, $\sum_{\jmath\in\widetilde{I}}^{\jmath\neq\imath_i} \overline{\pi}_\jmath = \epsilon$, $\epsilon \le \overline{\epsilon}$, $\overline{\Pi} \neq \emptyset$. ■

This result guarantees that a m-step stabilizing switching policy exists, for large enough m: by the assumption that $WCET_1 \le \tau_{\min}$, it is indeed sufficient to choose $i = 1$ in the Lemma above. To exploit the more general m-step contractivity condition (5), we introduce a switching law such that some control patterns, i.e., sequences of symbols in I, are preferentially used with respect to others. This may imply associating to a matrix sequence a steady-state probability of occurrence different from the product of the probabilities of each matrix. Accordingly, an unconstrained chain $\widetilde{\sigma}$ will be used for conditioning (as opposed to using a lifted version of the one-step conditioning chain σ), which can be interpreted as suggesting the sequence of controllers to be executed in the next m steps.

Let $\widetilde{\gamma}$ denote the lifted Markov chain describing the time allotted to control task computation in the next m periods, whose states are sequences of the original symbols η_i defined in the new state space L_η^m with cardinality l^m. Also let $\widetilde{\tau}$ denote the lifted scheduler process and $\widetilde{\tau}_{\widetilde{\imath}} \in L_\tau^m$ its states ($\widetilde{\imath} \in I^m$).

To compute the distribution of the lifted process $\widetilde{\tau}$ defined on $L_{\widetilde{\tau}} \triangleq L_\tau^m$, consider the matrix $T_{\widetilde{\gamma}\widetilde{\tau}}$, whose $(\widetilde{\jmath},\widetilde{\imath})$ entry is $\Pr\{\widetilde{\tau}(h) = \widetilde{\tau}_{\widetilde{\imath}} \mid \widetilde{\gamma}(h) = \widetilde{\eta}_{\widetilde{\jmath}}\}$, with $\widetilde{\tau}_{\widetilde{\imath}}$ the sequence $\tau_{i_1}\cdots\tau_{i_m}$ and $\widetilde{\eta}_{\widetilde{\jmath}} = \eta_{j_1}\cdots\eta_{j_m}$. Recalling the meaning of the process τ, this probability can be written as

$$\Pr\left\{\bigcap_{k=1}^{m}(T^{i_k}(mh+k-1)\leq\eta_{j_k}\cap T^{i_k+1}(mh+k-1)>\eta_{j_k})\right\} = \Pr\left\{\bigcap_{k=1}^{m}\left(T^{i_k}\leq\eta_{j_k}\cap T^{i_k+1}>\eta_{j_k}\right)\right\}$$

where we used the stationarity of the random variables T^{i_k}. From the independence of the variables T^{i_k} and T^{i_r} $\forall i_k, i_r \in I$ $k \neq r$, we have

$$\Pr\left\{\bigcap_{k=1}^{m}\left(T^{i_k}\leq\eta_{j_k}\cap T^{i_k+1}>\eta_{j_k}\right)\right\} = \prod_{k=1}^{m}\Pr\left\{T^{i_k}\leq\eta_{j_k}\cap T^{i_k+1}>\eta_{j_k}\right\}$$

where $T_{\widetilde{\gamma}\widetilde{\tau}} = T_{\gamma\tau}\otimes\cdots\otimes T_{\gamma\tau}$, m times.

The distribution of $\widetilde{\tau}$ and $\widetilde{\gamma}$ are hence linearly related through $T_{\widetilde{\gamma}\widetilde{\tau}}$. Consider the conditioning Markov chain $\widetilde{\sigma}$ with n^m states taking values in a finite state space $L_{\widetilde{\sigma}}$. For notational simplicity, let $\widetilde{\sigma}_{\widetilde{\jmath}} \in L_{\widetilde{\sigma}}$ denote a sequence of symbols $\sigma_j \in L_\sigma$, so that we have $L_{\widetilde{\sigma}} \triangleq L_\sigma^m$. A m-step equivalent of Problem 1 is directly obtained by replacing $\widehat{A}_i$, $\overline{\pi}_i^\star$ and $\overline{\pi}_{\tau_i}$ with $\widehat{A}_{\widetilde{\imath}}$, $\widetilde{\overline{\pi}}_{\widetilde{\imath}}^\star$ and $\widetilde{\overline{\pi}}_{\widetilde{\tau}_{\widetilde{\imath}}}$, respectively.

If a set of steady-state probability distributions $\widetilde{\overline{\Pi}}_d$ exists solving the m-step version of Problem 1, the synthesis problem is again to find a steady-state probability distribution $\widetilde{\overline{\pi}}_\sigma$ for the chain $\widetilde{\sigma}$ such that the aggregated process $(\widetilde{\tau}\widetilde{\sigma})^\star$ has steady-state distribution $\widetilde{\overline{\pi}}^\star \in \widetilde{\overline{\Pi}}_d$, with the aggregation induced by the element-wise minimum function.

Finally, a QoC metric generalizing (7) is obtained by setting $J(\widetilde{\overline{\pi}}^\star) = \widetilde{c}_{\widetilde{d}}\widetilde{\overline{\pi}}^{\star T}$, with $\widetilde{c}_{\widetilde{d}} = c_d\otimes\cdots\otimes c_d$ (m times). Indeed, in this case any weight $\widetilde{c}_{\widetilde{d}_{\widetilde{\imath}}}$ is the product of weights related to the sequence $\widetilde{d}_{\widetilde{\imath}} = d_{i_1}\cdots d_{i_m}$. With these stipulations, an optimal multistep switching problem for fixed m can be formulated in the parameters $\widetilde{\overline{\pi}}_{\widetilde{\sigma}}$ in the same terms as the OSP.

VI. Robust Synthesis

We consider now the case that an exact stochastic model of the scheduler is not available, rather it is subject to Unknown But Bounded (UBB) uncertainties. For the sake of simplicity, we limit our robustness analysis to the one-step average contractivity condition.

Assume first that UBB uncertainties affect the steady-state probabilities that a time γ is allotted to the control task, but that the probabilistic computational model of the different controllers is known. In other terms, we assume that the matrix $T_{\gamma\tau}$ is given, while $\overline{\pi}_\gamma$ is only constrained to belong to a polytopic set with a finite number of vertices, i.e., $\overline{\pi}_\gamma \in \Pi_\gamma$, with $\Pi_\gamma = \operatorname{conv}\{\pi_\gamma^1,\ldots,\pi_\gamma^q\}$ (conv denotes the convex hull operator).

Due to the linearity of the mapping between $\overline{\pi}_\gamma$ and $\overline{\pi}_\tau$, Π_γ is mapped in the polytopic set $\Pi_\tau = T_{\gamma\tau}\Pi_\gamma = \operatorname{conv}\{\pi_\tau^1,\ldots,\pi_\tau^r\}$, $r \leq q$, where $\pi_\tau^i \in \{T_{\gamma\tau}\pi_\gamma^1,\ldots,T_{\gamma\tau}\pi_\gamma^q\}$, $i = 1,\ldots,r$.

This uncertainty description lends itself to a very simple robustness problem formulation. Indeed, being that problem OSP is linear, hence convex, w.r.t. $\overline{\pi}_\tau$, the set of $\overline{\pi}_\sigma$ solving OSP for every $\overline{\pi}_\tau \in \Pi_\tau$ can be characterized simply by replacing the first inequality in OSP with r inequalities, one for each vertex π_τ^i. Hence, the new solution set is the polytope

$$\Pi_\sigma = \begin{cases}(M_{\pi_\tau}M_c)\overline{\pi}_\sigma^T \leq -\varepsilon < 0\\ 0 \leq \overline{\pi}_{\sigma_i} \leq 1\\ \sum_{i=1}^{n}\overline{\pi}_{\sigma_i} = 1\end{cases}$$

with $M_{\pi_\tau} = [\,\pi_\tau^{1T}\quad \pi_\tau^{2T}\quad\cdots\quad\pi_\tau^{rT}\,]^T$. Exploiting the linearity of the index function in OSP w.r.t. both $\overline{\pi}_\tau$ and $\overline{\pi}_\sigma$, the search for a robust optimal switching policy can be cast as a bilinear programming problem with disjoint constraints:

ROSP Problem

$$\max_{\overline{\pi}_\sigma\in\Pi_\sigma}\ \min_{\overline{\pi}_\tau\in\Pi_\tau}\ \overline{\pi}_\tau M_d\overline{\pi}_\sigma^T. \tag{14}$$

Such a problem admits an optimal solution $(\widehat{\overline{\pi}}_\sigma,\widehat{\overline{\pi}}_\tau)$ where $\widehat{\overline{\pi}}_\sigma \in \operatorname{vert}\{\Pi_\sigma\}$ and $\widehat{\overline{\pi}}_\tau \in \operatorname{vert}\{\Pi_\tau\}$ [36] (with $\operatorname{vert}\{\Pi\}$ denoting the set of vertices of Π). Applying von Neumann's minimax theorem [37], an equivalent formulation of ROSP is given by

$$\min_{\overline{\pi}_\tau\in\Pi_\tau}\ \max_{\overline{\pi}_\sigma\in\Pi_\sigma}\ \overline{\pi}_\tau M_d\overline{\pi}_\sigma^T.$$

An optimal solution $\left(\widehat{\overline{\pi}}_\sigma,\widehat{\overline{\pi}}_\tau\right)$ to ROSP can then be found by the following algorithm:

1) $\forall\pi_\tau^i \in \operatorname{vert}\{\Pi_\tau\} = \{\pi_\tau^1,\ldots,\pi_\tau^r\}$, solve the linear program

$$\pi_\sigma^i = \arg\max_{\overline{\pi}_\sigma\in\Pi_\sigma}\pi_\tau^i M_d\overline{\pi}_\sigma^T$$

and build the set $\Pi_{\tau\sigma} = \left\{\left(\pi_\tau^i,\pi_\sigma^i\right), i = 1,\ldots,r\right\}$;

2) Exhaustively search $\Pi_{\tau\sigma}$ and find

$$\left(\widehat{\overline{\pi}}_\sigma,\widehat{\overline{\pi}}_\tau\right) = \arg\min_{(\pi_\tau,\pi_\sigma)\in\Pi_{\tau\sigma}}\ \pi_\tau^i M_d\left(\pi_\sigma^i\right)^T.$$

Consider now the more realistic case that the transition probability matrix P of the Markov chain γ is also affected by UBB uncertainties, described by a polytope of stochastic matrices $\mathcal{P} = \mathrm{conv}\{P_1, \ldots, P_m\}$ such that $P \in \mathcal{P}$, and let

$$\mathcal{S} = \left\{ \overline{\pi}_\gamma \in \mathbb{R}^l \mid 0 \le \overline{\pi}_{\gamma_i} \le 1\, \forall i,\ \sum_i \overline{\pi}_{\gamma_i} = 1 \right\}$$

denote the simplex of probability distributions. The set of steady-state distributions $\overline{\pi}_\gamma$ corresponding to $\mathcal{P}$, i.e.

$$\mathcal{L} = \{\overline{\pi}_\gamma \in \mathcal{S} \mid \overline{\pi}_\gamma (P - I) = 0,\, P \in \mathcal{P}\}$$

is unfortunately not convex in general. In fact, if $\overline{\pi}_\gamma^i$ denote the steady-state vectors corresponding to vertex matrices P_i, there is no guarantee that $\mathcal{L}$ is included in $\mathrm{conv}\{\overline{\pi}_\gamma^1, \ldots, \overline{\pi}_\gamma^m\}$.

A numerically tractable condition for robust AS-stability can be obtained by constructing a polytope $\Pi_\mathcal{P}$ providing an outer (conservative) approximation of $\mathcal{L}$, i.e., $\Pi_\mathcal{P} \supseteq \mathcal{L}$.

Consider the image under $\mathcal{P}$ of a set $W \subseteq \mathcal{S}$, denoted as $\mathcal{P}(W) = \mathrm{conv}\{WP_1, \ldots, WP_m\}$. As a consequence of the linearity of the $\mathcal{P}$ map, if W is a polytope $W = \mathrm{conv}\{w^1, \ldots, w^r\}$, then also $\mathcal{P}(W)$ is a polytope, described by $\mathcal{P}(W) = \mathrm{conv}_{\substack{i=1,\ldots,m \\ j=1,\ldots,r}} \{w^j P_i\}$. Because $\mathcal{L}$ is the set of all fixed points for the map $\mathcal{P}$, i.e., $\forall \overline{\pi}_\gamma \in \mathcal{L}$, $\exists P \in \mathcal{P}$ such that $\overline{\pi}_\gamma P = \overline{\pi}_\gamma$, it holds $\mathcal{P}(\mathcal{L}) \supseteq \mathcal{L}$. For the same reason, for $W \supseteq \mathcal{L}$, it holds $\mathcal{P}(W) \supseteq \mathcal{L}$. It is worth noting that in general $\mathcal{P}(W) \not\supseteq W$, but, being $W \supseteq \mathcal{L}$ and $\mathcal{P}(W) \supseteq \mathcal{L}$, clearly $\mathcal{P}(W) \cap W \supseteq \mathcal{L}$. This suggests the following iterative algorithm:

$$\Pi_\mathcal{P}(0) \triangleq \mathcal{S}$$
$$\Pi_\mathcal{P}(k+1) = \mathcal{P}(\Pi_\mathcal{P}(k)) \cap \Pi_\mathcal{P}(k)$$

providing a non-increasing sequence of outer polytopic approximations for $\mathcal{L}$ as desired. If UBB uncertainties also affect the mapping $T_{\gamma\tau}$, i.e., if we assume $T_{\gamma\tau} \in \mathcal{T} = \mathrm{conv}\{T_{\gamma\tau}^1, \ldots, T_{\gamma\tau}^p\}$, then the approximating polytope $\Pi_\mathcal{P}$ is mapped by means of all the mappings in $\mathcal{T}$ in the polytope $\mathcal{T}\Pi_\mathcal{P} = \mathrm{conv}_{\substack{i=1,\ldots,p \\ j=1,\ldots,q}} \{\pi_\gamma^j T_{\gamma\tau}^i\}$. The general ROSP problem is therefore restated by simply replacing the polytope Π_τ with $\mathcal{T}\Pi_\mathcal{P}$.

VII. Implementation of Anytime Control Systems

In this section we discuss the problem of designing an ordered set of feedback control algorithms for the given plant Σ, whose state $w \in \mathbb{R}^N$ and output $y \in \mathbb{R}$ evolve with the input $u \in \mathbb{R}$ as

$$\begin{aligned} w(t+1) &= Aw(t) + Bu(t) \\ y(t) &= Cw(t). \end{aligned} \tag{15}$$

A first, straightforward approach consists of designing a set of n independent controllers Γ_i

$$\begin{aligned} z_i(t+1) &= F_i z_i(t) + G_i y(t) \\ u(t) &= H_i z_i(t) + L_i y(t) \end{aligned}$$

where $z_i \in \mathbb{R}^{N_i}$. A hierarchical design, with $N_i < N_{i+1}$ and increasing performance, can be made e.g., by adopting a sequence of design procedures of increasing sophistication and/or by introducing successive refinements of design specifications. Controllers are then computed sequentially in the order $\Gamma_1, \Gamma_2, \ldots, \Gamma_\sigma$ according to the switching policy proposed above.

TABLE I
Computational Complexity (Considered as the Number of Multiplications Except by 0 or 1) and Numerical Reliability of Different State-Space Realizations of a Strictly Proper Transfer Function $G(z)$ With N Poles. The Generic Case Assumes No Particular Structure in the Systems Matrices

	Comp. Compl.	Num. Reliab.
Generic	$N\,(N+2)$	–
Companion	$2N$	low
Jordan	$2N$ to $3N$	high

For AS stability analysis purposes only, it is convenient to set up a compound state vector $x \in \mathbb{R}^{N_c}$, with $N_c = N + N_1 + \cdots + N_n$, which includes the states of the plant and those of each controller. The corresponding closed-loop matrices $\widehat{A}_i \in \mathbb{R}^{N_c \times N_c}$ are computed as follows:

$$\widehat{A}_i = \begin{bmatrix} A - BL_iC & -B\widetilde{H}_i \\ \widetilde{G}_iC & \widetilde{F}_i \end{bmatrix} \tag{16}$$

where, assuming that the "sleeping" states corresponding to inactive controllers Γ_i, $i \neq \sigma(t)$ are reset to zero, we have

$$\begin{aligned} \widetilde{H}_i &= [0, \ldots, 0, H_i, 0, \ldots, 0] \\ \widetilde{F}_i &= \mathrm{diag}\,(0, \ldots, 0, F_i, 0, \ldots, 0) \\ \widetilde{G}_i^T &= \left[0, \ldots, 0, G_i^T, 0, \ldots, 0\right] \end{aligned}$$

with the non-zero blocks in the i–th block positions. It should be noticed that this independent controller design approach is the simplest but does not provide any modularity: indeed, in such a mutually exclusive scheme, all computation results for $\Gamma_1, \ldots, \Gamma_{\sigma-1}$ are eventually discarded.

To obtain a modular design for anytime control, one could pursue a top-down approach, starting with the design of a complex, high-performance controller Γ_n (by, e.g., a H_∞ technique applied to the full set of performance requirements). Progressively simpler controllers Γ_i, $n-1 \ge i \ge 1$ may then be obtained by e.g., model reduction techniques. As already remarked, however, this approach does not systematically guarantee closed-loop performance under switching.

Moreover, most model reduction techniques require state-space realizations with full dynamic matrices F_i, which makes them impractical in real-time embedded applications. Indeed, practicality requirements imply careful consideration of algorithmic implementations of control laws [38]. Table I reports a comparison among three different state space representations for SISO systems.

We propose a simple, bottom-up design technique which is suitable for addressing the main requirements of anytime control algorithms. The method is based on classical cascade design. Consider the two design stages illustrated in Fig. 1, in which controllers are designed to ensure increasing performance by any classical synthesis technique. The scheme in Fig. 1 cannot be implemented as a modular anytime control, because after computation of the a) scheme, the input to the $F_1(z)$ block needs to be recomputed completely if the b) scheme is to be applied. However, by simple block manipulations, the scheme in Fig. 2 can be obtained, where we set $\widehat{K}_2(z) = F_1(z)K_2(z)$. The scheme in Fig. 2 is suitable

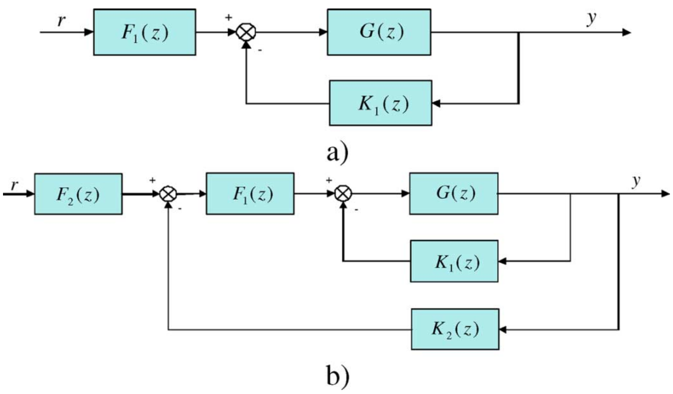


Fig. 1. Stages of a classical cascade design procedure.

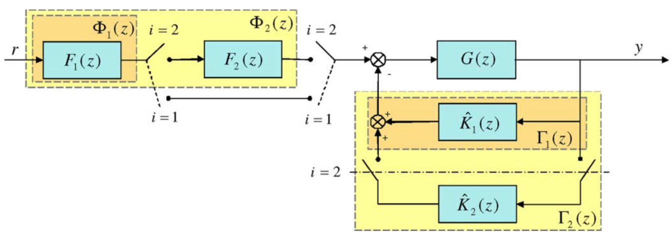


Fig. 2. Switched control scheme suitable for anytime control implementation. The scheme is equivalent to Fig. 1(a) when the switches are in the $i = 1$ position, and to Fig. 1(b) for $i = 2$.

for anytime implementation. Indeed the series of $F_1(z)$ and $F_2(z)$ is in open-loop (hence equivalent to an anytime filter $F_1(z)F_2(z)$), while the parallel connection in the feedback loop is simply obtained by summing the value computed in $\widehat{K}_2(z)$ to the value computed previously in $\widehat{K}_1(z) = K_1(z)$. Using Jordan form realizations of the blocks provides good numerical accuracy as well as low computational complexity. The cascade design method can be applied iteratively to provide a complete hierarchy of n controllers, satisfactorily addressing the issues of hierarchical design, practicality, and modularity.

For AS stability analysis, consider the sequence of feedback controllers $\Gamma_1(z) = \widehat{K}_1(z), \ldots, \Gamma_{i+1}(z) = \Gamma_i(z) + \widehat{K}_{i+1}(z)$. Let the dynamics of the component controllers $\widehat{K}_i$ be realized in a N_i–dimensional state space with $(A_{K_i}, B_{K_i}, C_{K_i}, D_{K_i})$. A state-space realization for Γ_i is then given by

$$\begin{aligned} F_i &= \mathrm{diag}\,(A_{K_1}, \ldots, A_{K_i}) \\ G_i^T &= [\, B_{K_1}^T \quad \cdots \quad B_{K_i}^T \,] \\ H_i &= [\, C_{K_1} \quad \cdots \quad C_{K_i} \,] \\ L_i &= D_{K_1} + \cdots + D_{K_i}. \end{aligned}$$

Accordingly, and assuming again that the states of inactive controllers are reset to zero, the corresponding closed-loop matrices $\widehat{A}_i \in \mathbb{R}^{N_c \times N_c}$ have the same structure as in (16) with the new block definitions

$$\widetilde{H}_i = [\, H_i \quad 0 \quad \cdots \quad 0 \,] \tag{17}$$

$$\widetilde{F}_i = \mathrm{diag}\,(F_i, 0, \ldots, 0). \tag{18}$$

$$\widetilde{G}_i^T = [\, G_i^T \quad 0 \quad \cdots \quad 0 \,]. \tag{19}$$

A. Tracking Control and Bumpless Transfer

The schedule conditioning technique of Section V is able to address the switched system performance issue satisfactorily when a regulation problem is considered. In reference-tracking tasks, however, performance can be severely impaired by switching between different controllers. Indeed, some of the controllers may remain idle for a certain number of periods, with their states in a "sleeping" condition. Sudden re-activation of sleeping states may produce "bumps" in the states and outputs, and degrade performance, if the active part of the system has gone through significant changes in the meantime.

The design of bumpless-control techniques has been extensively considered in the literature since at least the 80's ([39]), and is still an active area of research ([40]–[42]). However, most of the existing results are not meant for resource-limited applications as those targeted here, and do not comply with the practicality requirements which are at a premium in anytime control. We describe below a simple method for bumpless control that applies to the modular anytime structure in Fig. 2, and that generalizes it to tracking problems for set-point references $r(t)$ which vary slowly with respect to the scheduling switches.

We assume in what follows that the input reference is scaled, as customary in tracking problems, in the prefilter block Φ_i (see Fig. 2), by the steady-state gain of the corresponding closed loop system with controller Γ_i, $1 \leq i \leq n$. Let w denote the state of the controlled plant, and x_{K_i} be the state of the i–th controller component $\widehat{K}_i$. Suppose now that, at some instant in time $t^* T_g$, the i–th level controller is active and the system components have reached a steady-state equilibrium under a constant reference $\bar{r}$. Let $\bar{w}$, $\bar{x}_{K_i}$, and $\bar{y} = \bar{r}$ denote the corresponding equilibrium values of the states and output.

Consider the event that, at time $(t^* + 1)T_g$, the execution of the j–th level controller is imposed, by either the occurrence of a deadline or a conditioned schedule switch, while the reference remains unchanged from $\bar{r}$. If $j \leq i$ (high-to-low level switching), it can be easily verified that the active parts of the system state remain at $\bar{w}, \bar{x}_{K_1}, \ldots, \bar{x}_{K_j}$. If instead $j > i$ (low-to-high level switching), the evolution of the whole system for $t > t^*$ depends upon the values of the sleeping states at time t^*, i.e., $x_{K_k}(t^*)$, $i < k \leq j$. The dynamics of controller states during their idle periods is therefore an important degree-of-freedom in switching control design. Straightforward policies for managing sleeping states, such as e.g., keeping them constant during sleep, may be adequate for the regulation problem, but not for tracking (if the reference changes even slowly during the idle time of a controller component, output bumps will necessarily result at re-activation of that component). It is worthwhile noticing explicitly that zeroing the sleeping states, either instantaneously – as assumed in the previous section – or progressively with a simple and computationally inexpensive dynamics as e.g., in [11], effectively avoid bumps only for zero-regulation problems.

When the active part of the system is at steady state in $t = t^*$ under a constant reference $\bar{r}$, perfectly smooth low-to-high switching would be achieved if and only if the state of the k-th component controller ($i < k \leq j$) is re-initialized as

$$(I - A_{K_k})x_{K_k}(t^* + 1) = B_{K_k}\bar{y}. \tag{20}$$

TABLE II
SAMPLED-TIME TRANSFER FUNCTIONS FOR SYSTEMS IN FIGS. 3 AND 7, AND HIERARCHICAL CONTROLLERS USED IN SIMULATIONS

FURUTA PENDULUM SYSTEM

$G(z) = \frac{1.2\,10^{-3}(z+3.7)(z+0.3)}{(z-1)(z^2-1.7z+1)}$

$C_1(z) = \frac{31.6(z^2-1.8z+1.1)}{(z-0.1)(z-0.5)}$	$F_1(z) = 21.28$
$C_2(z) = \frac{136.5(z-1.3)(z-4.7\,10^{-3})(z^2-1.4z+0.7)}{(z-0.5)(z+0.3)(z-0.1)(z^2+0.6z+0.8)}$	$F_2(z) = 0.54$
$C_3(z) = \frac{1370.9(z-0.4)(z-0.6)(z-0.2)(z+0.3)(z-4.7\,10^{-3})(z^2-0.7z+0.2)(z^2-0.4z+0.3)}{(z-0.5)^2(z+0.3)^2(z-0.1)^2(z^2+0.6z+0.8)(z^2+3.4z+4.7)}$	$F_3(z) = 2.73$

TORA SYSTEM

$G(z) = \frac{0.27266(z+1)(z^2-1.967z+1)}{(z-1)^2(z^2-1.964z+1)}$

$C_1(z) = \frac{2.0895(z-0.75)}{(z+0.3761)}$	$F_1(z) = 0.38$
$C_2(z) = \frac{0.8(z-0.4)}{(z+0.6)}$	$F_2(z) = 1.79$
$C_3(z) = \frac{0.73(z^2-0.76z+0.2228)}{(z+0.3)^2}$	$F_3(z) = 1.29$

Observing that $y(t^*) = \bar{y}$ and setting $W_k \triangleq (I - A_{K_k})^{-1} B_{K_k}$, one can rewrite (20) as

$$x_{K_k}(t^*+1) = W_k y(t^*). \tag{21}$$

We replace this re-initialization formula with the following one:

$$x_{K_k}(t^*+1) = A_{K_k} W_k y(t^*-1) + B_{K_k} y(t^*) \tag{22}$$

which is equivalent to (20) provided that $y(t^*-1) = y(t^*) = \bar{y}$. This holds true under the hypothesis that the system is at steady-state before the switch. Indeed, using the definition of W_k and Woodbury's matrix inversion lemma, it is straightforward to prove that $(A_{K_k} W_k + B_{K_k}) = W_k$. If the system is not exactly at steady-state at t^*, but the reference is varying slowly, the difference $y(t^*+1) - y(t^*)$ is small and application of the two formulas will slightly differ.

From a computational viewpoint, this re-initialization scheme introduces negligible overhead, because matrices W_k can be easily pre-computed, and (22) needs to be evaluated only once when a low-to-high transition occurs for the reactivated controllers. However, for the sake of AS stability analysis, it is instrumental to consider an equivalent system where all sleeping states are re-initialized at every time instant as

$$x_{K_k}(t) = W_k y(t-1)$$

(the re-initialization of states clearly has no effect on the closed-loop system until they remain inactivated). Therefore, the closed-loop matrices $\widehat{A}_i \in \mathbb{R}^{N_c \times N_c}$ have the same structure as in (16), (17) and (18), but with (19) replaced by

$$\widetilde{G}_i^T = [\, G_i^T \quad W_{i+1}^T \quad \cdots \quad W_n^T \,]. \tag{23}$$

In conclusion, the solution of the OSP (ROSP) problem with $\widehat{A}_i$ matrices computed as in (16), (17), (18), and (23), provides the (robust) optimal switching policy for the intended controller hierarchy Γ_i, while also guaranteeing AS stability of the bumpless transfer technique (22).

VIII. EXAMPLES

The control of two mechanical systems will be used to illustrate the application of the proposed technique. In the first example we consider a regulation problem for both a nominal and an uncertain scheduler model. A conditioning Markov chain solving the OSP and ROSP problems, respectively, is obtained which satisfies the 1-step average contractivity condition. The second example illustrates application of m-step average contractive solutions to the OSP problem, and the effectiveness of the proposed bumpless transfer technique to track piecewise-constant references. In both cases, the unit delay between the output sampling instant and the application of the control action has been explicitly considered.

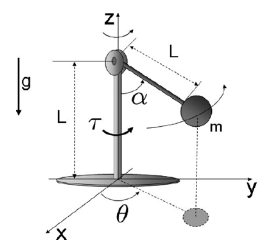


Fig. 3. Model of Furuta pendulum with zero offset ([43]).

For the sake of brevity, in the examples we consider a deterministic WCET model of controller execution time, so that, in the light of Remark 2, the process τ is described by a FSHIA Markov chain. The same transition probability matrix of the nominal scheduler process will be assumed in both examples, which is

$$P_\tau = \begin{bmatrix} 0.2744 & 0.342 & 0.3836 \\ 0.0881 & 0.3443 & 0.5676 \\ 0.0204 & 0.2097 & 0.7699 \end{bmatrix}. \tag{24}$$

The corresponding steady-state probability distribution is given by $\overline{\pi}_\tau = [1/20, 5/20, 14/20]$.

In the application of the contractivity condition for solving (R)OSP problems, a suitable choice of the adopted matrix norm can considerably help. In our examples we use the method proposed by [32] to choose a vector norm $|x| = |Tx|_2$ and induced matrix norm $\|A\| = \|TAT^{-1}\|_2$, where T is a non-singular matrix satisfying the condition that $\exists \widehat{A}_i$ such that $\|T\widehat{A}_i T^{-1}\|_2 < 1$ (this condition generalizes directly to the multistep problem).

Example 1 (One-Step Contractivity With Robustness): A model of Furuta pendulum with zero offset ([43]) is depicted in Fig. 3 and its sampled-time linearized dynamics is reported in Table II. With reference to the connection scheme in Fig. 2,

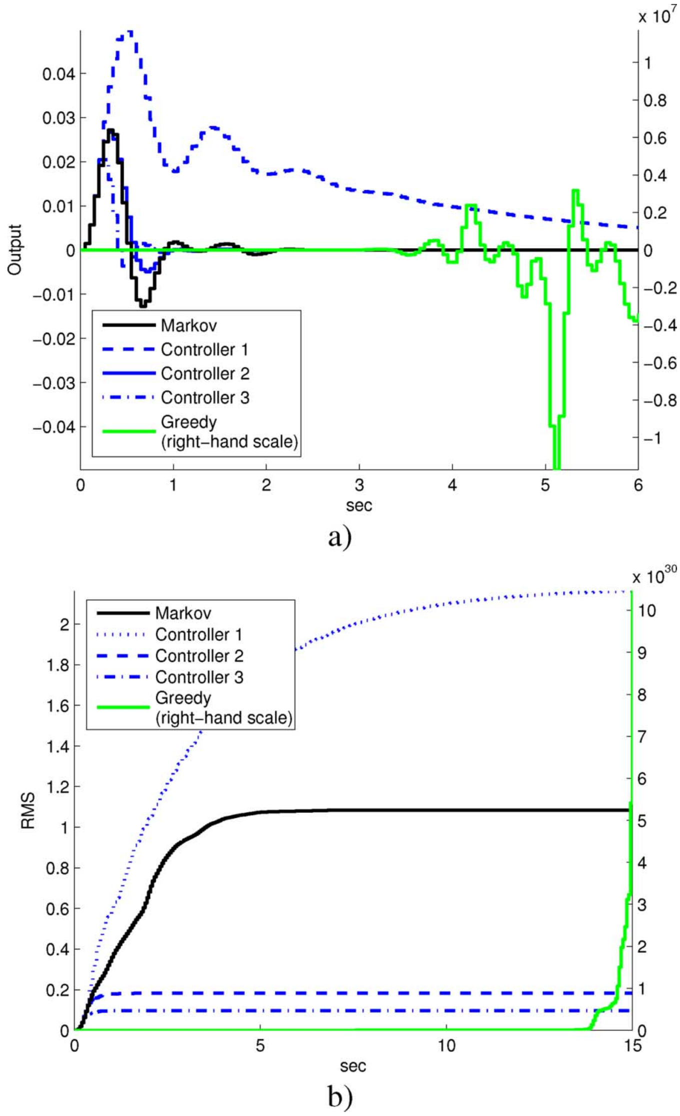


Fig. 4. Regulation results for the Furuta pendulum example: (a) output signals; (b) mean values of RMS errors (for one thousand runs) of the closed loop system with different control schedulings.

the first controller $K_1(z)$ has been designed to ensure the stability requirement, while the controllers $K_2(z)$ and $K_3(z)$ are obtained applying LQG design techniques. Prefilters $F_i(z)$ are constants used to adapt the steady-state gain and ensure static requirements. Notice that, in order to model the constant unit delay, an additional $1/z$ term should be added to the transfer function $G(z)$.

For the Furuta pendulum a solution to the 1-step OSP with QoC index vector $c_d = [1, 4, 9]$ is $\overline{\pi}_\sigma = [0.006, 0.972, 0.022]$, resulting in $\overline{\pi}_{\tau|\sigma} = [0.056, 0.929, 0.015]$ and $J(\overline{\pi}_{\tau|\sigma}) \approx 3.91$.

In Fig. 4(a), the output signals obtained by different controllers are shown, corresponding to regulation errors for initially perturbed angular velocity $\pi/10$ rad/s with respect to an equilibrium velocity of 3.71 rad/s, and an equilibrium pendulum displacement of $\pi/4$ rad from the vertical. It can be noticed in this case that the application of the greedy switching policy causes instability.

In Fig. 4(b), the *Root Mean Squares* (RMSs) of the regulation error are shown for different controllers corresponding to the same perturbation. Data plotted represent the average RMS error of one thousand simulations performed for different independent realizations of the scheduler and conditioning processes. Plots labeled Controller 1, 2, and 3, corresponding to results obtained without switching, are reported for reference. Notice the performance increase which would be obtained by the (unschedulable) higher-level controllers. The average behavior of the greedy switching policy is clearly unstable. On the same Fig. 4(b), the plot labeled "Markov" shows the average RMS error obtained by the stochastically conditioned scheduler. This example shows how the proposed stochastic switching policy ensures the AS-stability of the closed loop system (which is not guaranteed by the greedy policy), while it obtains a definite performance increase (of the order of 50%) with respect to the conservative scheduling policy consisting in using only Controller 1 [see Fig. 4(b)]. Consider now the case that the actual transition probability matrix of the scheduler is affected by UBB uncertainties. In practical cases, a rough description of the uncertainty may be available in terms of a bounding box centered in the nominal value P_τ and described by an additive uncertainty matrix ΔP_τ, i.e., $\tilde{\mathcal{P}} = \{\tilde{P}_\tau = P_\tau + \delta P_\tau \mid \delta P_\tau \in \Delta P_\tau\}$ where $\Delta P_\tau = (\rho_{ij})_{n\times n}$. Notice however that $\tilde{\mathcal{P}}$ is not in general a set of stochastic matrices, so that a polytopic description $\mathcal{P}$ of the possible perturbations is obtained as the intersection of $\tilde{\mathcal{P}}$ with the set of stochastic matrices (this can be done e.g., by using the Polytope Library of the Multi-Parametric Toolbox [44]). In our example, taking P_τ as in (24) and $\rho_{1j} \in [-0.025, 0.025]$, $\rho_{2j} \in [-0.02, 0.02]$, and $\rho_{3j} \in [-0.01, 0.01]$, a matrix polytope $\mathcal{P}$ is obtained which has 216 vertices (not reported here). Using the algorithm presented in Section VI, an approximating polytope of the steady-state probability set $\mathcal{L}$ with 30 vertices is obtained after 8 iterations (see Fig. 5) A solution to the ROSP problem for this example with QoC index vector $c_d = [1, 4, 9]$ is $\overline{\pi}_\sigma = [10^{-6}, 0.992, 0.007]$, resulting in $J(\overline{\pi}_{\tau|\sigma}) \approx 3.88$ (i.e., slightly worse than the QoC index in the nominal case).

In Fig. 6, the RMSs of the regulation error for different closed loop controllers is shown. The perturbed initial conditions are the same of the previous case (with known scheduler). The RMS regulation error is obtained as the mean of the RMS errors for 2500 simulations, whereby the scheduler stochastic description is obtained by randomly choosing 50 transition probability matrices $\tilde{P}_\tau \in \mathcal{P}$ and considering 50 different sample realizations for each matrix. The closed loop system driven by the greedy switching policy still appears unstable. The example shows results very close to the ones obtained for the nominal scheduler.

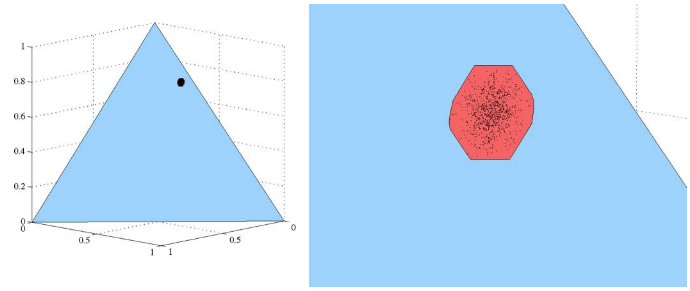


Fig. 5. Samples of steady-state probability distributions in $\mathcal{L}$ for $P \in \mathcal{P}$ and the outer approximating polytope $\Pi_{\mathcal{P}}(8)$. On the right, the approximating polytope is located on the canonical 3-D simplex.

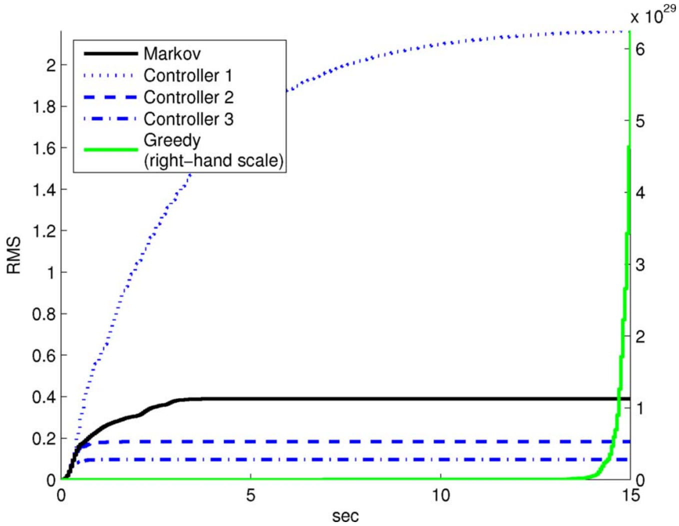


Fig. 6. Regulation results for the Furuta pendulum example and scheduler transition probability matrix affected by UBB uncertainties: mean value of the RMS errors (for 2500 runs) of the closed loop system with different schedules.

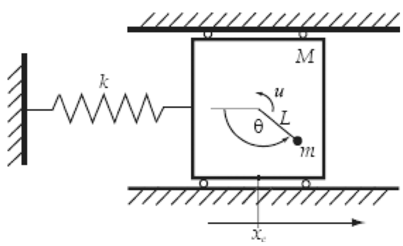


Fig. 7. Model of a Translational Oscillator/Rotational Actuator (TORA) system ([45]).

Example 2 (Multi-Step Contractivity and Tracking): A model of the Translational Oscillation Rotational Actuation (TORA) systems ([45]) is depicted in Fig. 7, while its sampled-time linearized dynamics is reported in Table II. In this case the first (simplest) controller ensures only the stability requirement, while other controllers are designed by hand to achieve performance enhancements with minimal complexity increase. Fig. 8(a) and (b) report plots to illustrate application of the proposed methodology to the TORA example. A 4-steps lifted version of the OSP problem with index vector $c_d = \otimes_{i=1}^{4}[1,4,9]$ admits a solution resulting in $J(\widetilde{\overline{\pi}}_{\widetilde{\tau}|\widetilde{\sigma}}) \approx 382.49$, according to which the conditioning sequence of controllers is a concatenation of the 81 possible combinations of length 4 of the three controllers. The steady state conditioning probability distribution $\widetilde{\overline{\pi}}_{\widetilde{\sigma}} \in (0,1)^{81}$ is not reported here; it is however worth noticing that the two controller sequences $\Gamma_2 - \Gamma_2 - \Gamma_2 - \Gamma_2$ and $\Gamma_2 - \Gamma_2 - \Gamma_2 - \Gamma_3$ turn out to be by far the most likely, being used in more than the 51.5% and the 40.5% of cases, respectively. Results of simulations of the different controllers and scheduling policies are reported in Fig. 8(a), for a regulation problem from perturbed initial angular velocity of 0.1 rad/s. The RMS performance plots in Fig. 8(a) show that the greedy policy (in this particular case) does not lead to divergence. In Fig. 8(b), sample realizations of scheduler, conditioning and conditioned processes are depicted: the prevalence of the preferred patterns is apparent.

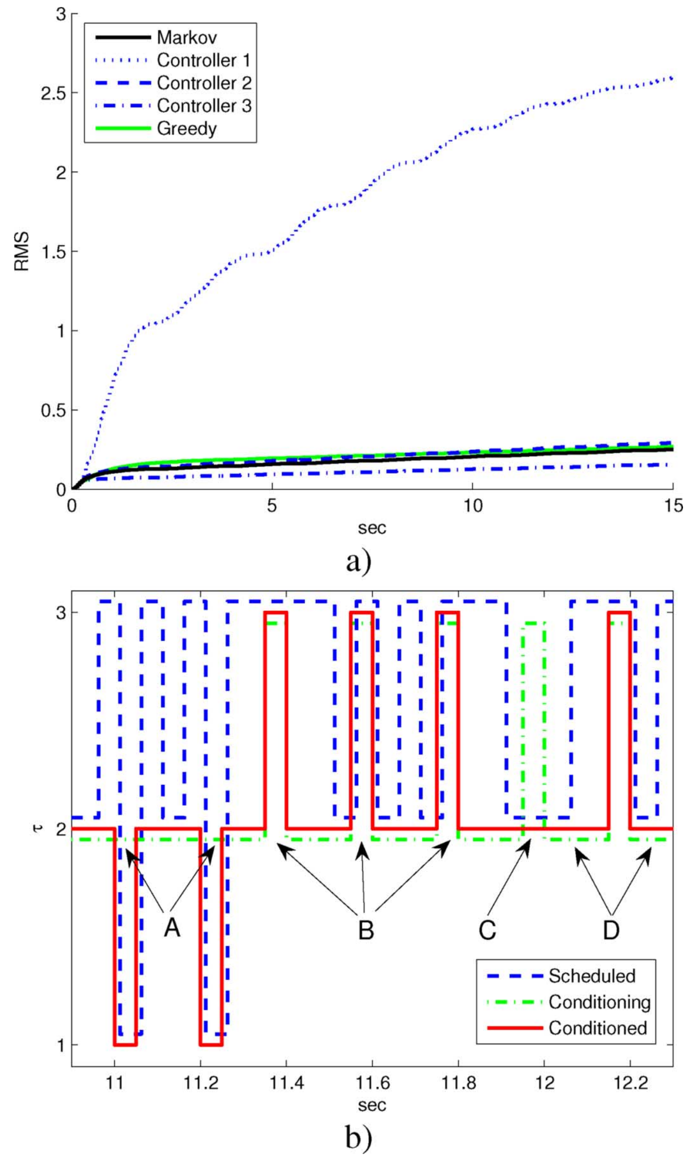
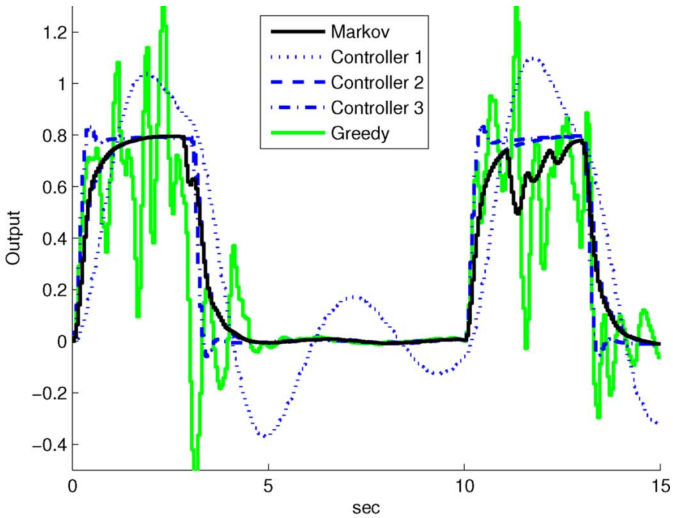
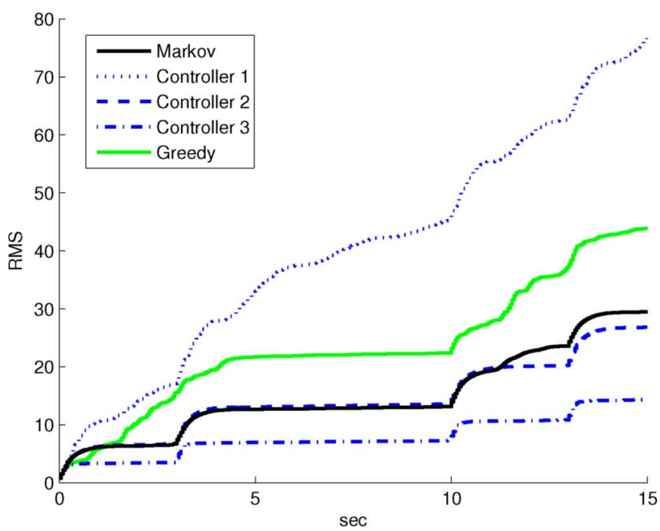


Fig. 8. (a) Regulation results for the TORA example: mean values of RMS errors (for one thousand runs) of the closed loop system with different control schedulings. (b) Sample realization of the scheduler, conditioning and conditioned stochastic processes for the TORA example. At instants labeled as A the conditioning process (i.e., the switching policy) calls for the execution of Controller 2, while the scheduler gives time only for the execution of Controller 1, hence the conditioned process executes Controller 1. Similarly, at instant C, the switching policy call for Controller 3 is overridden and Controller 2 is executed. At instants tagged as D, the opposite occurs, i.e., the conditioning process forces execution of Controller 2, while the scheduler could allow Controller 3 to be executed. Finally, at instants labeled as B, the switching policy and the scheduler agree on the execution of Controller 3.

Finally, results of application of the proposed technique for a tracking control problem for the TORA example are reported in Fig. 9. The reference to be tracked by the angular position is a piecewise constant signal of amplitude $\pi/4$ rad, period 10 s and pulse width of 30%. The comparison of RMS errors shows that the simple bumpless switching technique proposed in Section VII-A ensures good performance, as shown in both Fig. 9(a),(b). The RMS performance of the conditioned switching policy is better than both the conservative and the greedy approaches and is quite close to the results given by using always Controller 2, which is not a feasible choice.



a)



b)

Fig. 9. Tracking results for the TORA system: (a) output signals; (b) RMS errors (one thousand runs).

IX. Conclusion

We considered the problem of scheduling the execution of several control tasks on a multi-tasking, preemptive RTOS, with an anytime control approach. Given a stochastic model of the scheduler and the set of controllers, we formulated a linear programming problem whose solutions provide switching laws that condition the scheduler in such a way that the resulting switched system is stable in a probabilistic sense. Although a satisfactory solution for this problem is not guaranteed for one-step switching laws, we have shown that for any set of stabilizing anytime controllers it is possible to find a long enough step horizon m, such that a m-step switching law exists providing almost sure stability. The case of a scheduler model affected by unknown but bounded uncertainties has also been considered, and results on robustness have been reported for the one-step contractivity condition.

We also presented simple design techniques to synthesize anytime controllers, ensuring robustness to numerical errors as well as reduction of computational burden w.r.t. worst case execution time design. A bottom-up approach was described, where closed-loop performance are used to drive the practical design of a compositional hierarchy of controllers. Tracking problems related to the re-activation of controllers are treated using a bumpless-transfer approach, still guaranteeing AS stability and capable of significant performance improvements despite its simplicity and computational inexpensiveness.

Finally, the performance of the controlled system under switching have been illustrated by simulation. Two different mechanical systems are used as examples, showing how the proposed anytime control technique performs better than conventional, over-conservative control scheduling approaches.

Appendix

Lemma 2: Constraints $D2.2)$, $D2.3)$, and $D2.4)$ in Definition 2 are satisfied by any $\overline{\pi}_d = \overline{\pi}^\star = [\,\overline{\pi}_1^\star \quad \cdots \quad \overline{\pi}_n^\star\,]^T$ with

$$1) \qquad \overline{\pi}_i^\star = \sum_{(\tau_h,\sigma_k)\in\chi_i} \overline{\pi}_{\tau_h}\overline{\pi}_{\sigma_k}$$

$$2) \qquad 0 \le \overline{\pi}_{\sigma_i} \le 1$$

$$3) \qquad \sum_{i=1}^{n} \overline{\pi}_{\sigma_i} = 1.$$

Proof: Recall (12) and in particular

$$\overline{\pi}^\star = (\overline{\pi}_\gamma \otimes \overline{\pi}_\sigma)\,(T_{\gamma\tau} \otimes I_n)\,H.$$

The vector $\overline{\pi}_\gamma \otimes \overline{\pi}_\sigma$ is the unique invariant probability distribution of the merged chain $\gamma\sigma(t)$, hence satisfies constraints formally equal to $D2.2$ and $D2.3$ (provided that the rigth dimensions are specified). The fact that even $\overline{\pi}^\star$ satisfies $D2.2$ and $D2.3$, hence, is a straight consequence of $T_{\gamma\tau} \otimes I_n$ and H being stochastic matrices.

We prove now that $\overline{\pi}^\star$ satisfies constraint $D2.4$. Let us re-write $D2.4$ as

$$\sum_{j=i}^{n} \overline{\pi}_j^\star \le \sum_{j=i}^{n} \overline{\pi}_{\tau_j} \tag{25}$$

and $\overline{\pi}_j^\star$ as

$$\overline{\pi}_j^\star = \overline{\pi}_{\sigma_j} \sum_{k=j}^{n} \overline{\pi}_{\tau_k} + \overline{\pi}_{\tau_j} \sum_{k=j+1}^{n} \overline{\pi}_{\sigma_k}. \tag{26}$$

Substituting (26) in (25) we obtain

$$\sum_{j=i}^{n} \overline{\pi}_{\sigma_j} \sum_{k=j}^{n} \overline{\pi}_{\tau_k} + \sum_{j=i}^{n} \overline{\pi}_{\tau_j} \sum_{k=j+1}^{n} \overline{\pi}_{\sigma_k} \le \sum_{j=i}^{n} \overline{\pi}_{\tau_j}$$

from which, rearranging the first two sums and exchanging names to the indices, we have

$$\sum_{j=i}^{n} \overline{\pi}_{\tau_j} \sum_{k=i}^{j} \overline{\pi}_{\sigma_k} + \sum_{j=i}^{n} \overline{\pi}_{\tau_j} \sum_{k=j+1}^{n} \overline{\pi}_{\sigma_k} - \sum_{j=i}^{n} \overline{\pi}_{\tau_j} \le 0$$

$$\sum_{j=i}^{n} \overline{\pi}_{\tau_j} \left(\sum_{k=i}^{n} \overline{\pi}_{\sigma_k} - 1 \right) \le 0.$$

If $\sum_{j=i}^{n} \overline{\pi}_{\tau_j} = 0$ the previous inequality is trivially verified, else $\sum_{j=i}^{n} \overline{\pi}_{\tau_j} > 0$ can be removed. Thus $\sum_{k=i}^{n} \overline{\pi}_{\sigma_k} \leq 1$, which is true by virtue of 3). ∎

Before proving Theorem 4 we need some preliminary results on primitive matrices. Recall that the transition probability matrix of a FSHIA Markov chain is a time-invariant stochastic irreducible aperiodic matrix of finite dimension.

Definition 3 ([46, p.127]): A matrix is *primitive* if it is irreducible and aperiodic.

Theorem 5 ([46, p.128]): Let $A \geq 0$[1]. The following are equivalent:

1) A is primitive.
2) $A^m > 0$ for some $m \geq 1$.
3) $A^m > 0$ for all sufficiently large m.

Primitivity is preserved by the Kronecker product.

Lemma 3: Given $A \geq 0$ and $B \geq 0$ primitive matrices, then $A \otimes B$ is a primitive matrix.

Proof: From Theorem 5 we know that there exist $m_1, m_2 \geq 1$ such that $A^{m_1} > 0$ and $B^{m_2} > 0$. From point 3 of the same theorem we know that there exists $m \geq \max(m_1, m_2)$ such that $A^m > 0$ and $B^m > 0$. Recalling the definition of Kronecker product, it is apparent that $A^m \otimes B^m > 0$ and using the "mixed product rule" $A^m \otimes B^m = (A \otimes B)^m > 0$, hence $A \otimes B$ is primitive. ∎

Proof of Theorem 4:

i) From the independence of the random variables $\alpha(t)$ and $\beta(t)$ $\forall t \in \mathbb{N}$, we can write compactly the probability distribution of the merged random variable $\alpha\beta(t) = (\alpha(t), \beta(t))$ $\forall t \in \mathbb{N}$ as follows:

$$\pi_{\alpha\beta}(t) = \pi_\alpha(t) \otimes \pi_\beta(t). \tag{27}$$

Considering the previous relation in $t = 0$, yields the initial probability distribution for the process $\alpha\beta(t)$, while in $t + 1$ yields

$$\begin{aligned}\pi_{\alpha\beta}(t+1) &= \pi_\alpha(t+1) \otimes \pi_\beta(t+1)\\ &= (\pi_\alpha(t)P_\alpha) \otimes (\pi_\beta(t)P_\beta)\\ &= (\pi_\alpha(t) \otimes \pi_\beta(t))\,(P_\alpha \otimes P_\beta)\\ &= \pi_{\alpha\beta}(t)\,(P_\alpha \otimes P_\beta)\end{aligned}$$

where we used the "mixed product rule." To prove that $\alpha\beta(t)$ is a FSHIA Markov chain, we must show that the transition probabilty matrix $P_{\alpha\beta} = P_\alpha \otimes P_\beta$ is a time-invariant stochastic primitive matrix. The first two properties follow directly by the same properties of P_α and P_β and by the definition of Kronecker product; the third property is proved by Lemma 3.

ii) From (27) and the previous point, we obtain the evolution (8). Considering again (27), it is apparent that $\overline{\pi}_{\alpha\beta} = \lim_{t\to\infty} \pi_{\alpha\beta}(t) = \overline{\pi}_\alpha \otimes \overline{\pi}_\beta$ for any $\pi_{\alpha\beta}(0) = \pi_\alpha(0) \otimes \pi_\beta(0)$. To extend this property to any initial distribution $\pi_{\alpha\beta}(0)$, it is sufficient to recall that the steady-state probability distribution of a FSHIA Markov chain is unique. ∎

[1]With $M \geq 0$ ($M > 0$) we mean nonnegative (positive) matrices. Stochastic matrices are a subset of nonnegative matrices.

ACKNOWLEDGMENT

The authors wish to thank B. Picasso for useful comments and discussion on "Almost Sure" stability.

REFERENCES

[1] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *J. Assoc. Comput. Mach.*, vol. 20, no. 1, pp. 46–61, 1973.

[2] J. W. S. Liu, *Real-Time Systems*, T. Holm, Ed. Upper Saddle River, NJ: Prentice Hall, 2000.

[3] G. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Boston, MA: Kluwer Academic Publishers, 1997.

[4] L. Palopoli, L. Abeni, G. Bolognini, B. Allotta, and F. Conticelli, "Novel scheduling policies in real-time multithread control system design," *Control Eng. Prac.*, vol. 10, no. 10, pp. 1095–1110, Nov. 2002.

[5] G. Bernat, A. Burns, and A. Llamosi, "Weakly hard real-time systems," *IEEE Trans. Computers*, vol. 50, no. 4, pp. 308–321, Apr. 2001.

[6] P. Ramanathan, "Overload management in real-time control applications using (m, k)-firm guarantee," *IEEE Trans. Parallel Distrib. Syst.*, vol. 10, no. 6, pp. 549–559, 1999.

[7] D. Liu, X. S. Hu, M. D. Lemmon, and Q. Ling, "Firm real-time system scheduling based on a novel QoS constraint," *IEEE Trans. Computers*, vol. 55, no. 3, pp. 320–333, Mar. 2006.

[8] J. W. S. Liu, W.-K. Shih, K.-J. Lin, R. Bettati, and J.-Y. Chung, "Imprecise computation," *Proc. IEEE*, vol. 82, no. 1, pp. 83–93, Jan. 1994.

[9] J. W. S. Liu, K.-J. Lin, W.-K. Shih, A. C.-S. Yu, J.-Y. Chung, and W. Zhao, "Algorithms for scheduling imprecise computations," *Computer*, vol. 24, no. 5, pp. 58–68, 1991.

[10] N. Perrin and B. H. Ferri, K. E. Le Royal Meridien, Ed., "Digital filters with adaptive length for real-time applications," in *Proc. IEEE Real-Time Embedded Technol. Appl. Symp.*, Toronto, ON, Canada, May 2004 [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.133.7527

[11] R. Bhattacharya and G. J. Balas, "Implementation of control algorithms in an environment of dynamically scheduled CPU time using balanced truncation," in *Proc. AIAA Guid., Nav., Control Conf. Exhibit*, Monterey, CA, Aug. 2002, [CD ROM].

[12] R. Bhattacharya and G. J. Balas, "Anytime control algorithm: Model reduction approach," *J. Guid., Control, Dyn.*, vol. 27, no. 5, pp. 767–776, 2004.

[13] V. Gupta, "On an anytime algorithm for control," in *Proc. 48th IEEE Conf. Decision Control*, Shanghai, China, Dec. 2009, pp. 6218–6223.

[14] M. S. Branicky, "Stability of hybrid systems: State of the art," in *Proc. 36th IEEE Conf. Decision Control*, San Diego, CA, Dec. 1997, pp. 120–125.

[15] R. A. DeCarlo, M. S. Branicky, S. Pettersson, and B. Lennartson, "Perspectives and results on the stability and stabilizability of hybrid systems," *Proc. IEEE*, vol. 88, no. 7, pp. 1069–1082, Jul. 2000.

[16] D. Liberzon and A. S. Morse, "Basic problems in stability and design of switched systems," *IEEE Control Syst. Mag.*, vol. 19, no. 5, pp. 59–70, 1999.

[17] H. Ye, A. N. Michel, and L. Hou, "Stability theory for hybrid dynamical systems," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 461–474, Apr. 1998.

[18] J. P. Hespanha and A. S. Morse, "Switching between stabilizing controllers," *Automatica*, vol. 38, no. 11, pp. 1905–1917, 2002.

[19] M. A. Wicks, P. Peleties, and R. DeCarlo, "Construction of piecewise Lyapunov functions for stabilizing switched systems," in *Proc. 33rd IEEE Conf. Decision Control*, Lake Buena Vista, FL, Dec. 1994, pp. 3492–3497.

[20] M. Wicks and R. DeCarlo, "Solution of coupled Lyapunov equations for the stabilization of multimodal linear systems," in *Proc. Amer. Control Conf.*, Albuquerque, NM, Jun. 1997, pp. 1709–1713.

[21] S. Pettersson and B. Lennartson, "Stabilization of hybrid systems using a min-projection strategy," in *Proc. Amer. Control Conf.*, Arlington, VA, Jun. 2001, pp. 223–228.

[22] X. D. Koutsoukos and P. J. Antsaklis, "Design of stabilizing switching control laws for discrete- and contiunuous-time linear systems using piecewise-linear Lyapunov functions," *Int. J. Control*, vol. 75, no. 12, pp. 932–945, 2002.

[23] Q. Ling and M. D. Lemmon, "Robust performance of soft real-time networked control systems with data dropouts," in *Proc. IEEE Int. Conf. Decision Control*, Dec. 10–13, 2002, vol. 2, pp. 1225–1230.

[24] Q. Ling and M. D. Lemmon, "Soft real-time scheduling of networked control systems with dropouts governed by a Markov chain," in *Proc. Amer. Control Conf.*, Jun. 4–6, 2003, vol. 6, pp. 4845–4850.

[25] D. Liu, X. S. Hu, M. Lemmon, and Q. Ling, "Scheduling tasks with Markov-chain based constraints," in *Proc. 17th Euromicro Conf. Real-Time Syst.*, Jul. 6–8, 2005, pp. 157–166.

[26] S. Manolache, P. Eles, and Z. Peng, "Memory and time-efficient schedulability analysis of task sets with stochastic execution time," in *Proc. 13th Euromicro Conf. Real-Time Syst.*, 2001, pp. 19–26.

[27] G. Bernat, A. Colin, and S. M. Petters, "WCET analysis of probabilistic hard real-time systems," in *Proc. 23rd IEEE Real-Time Systems Symp. (RTSS'02)*, 2002, pp. 279–288.

[28] G. Bernat, A. Colin, and S. M. Petters, "pWCET: A Tool for Probabilistic Worst-Case Execution Time Analysis of Real-Time Systems," Dept. Comp. Sci., Univ. of York, Tech. Rep. YCS353, 2003.

[29] J. L. Diaz, D. F. Garcia, K. Kim, C.-G. Lee, L. L. Bello, J. M. Lopez, S. L. Min, and O. Mirabella, "Stochastic analysis of periodic real-time systems," in *Proc. 23rd IEEE Real-Time Syst. Symp. (RTSS'02)*, 2002, pp. 289–300.

[30] K. Kim, J. L. Diaz, L. L. Bello, J. M. Lopez, C.-G. Lee, and S. L. Min, "An exact stochastic analysis of priority-driven periodic real-time systems and its approximations," *IEEE Trans. Computers*, vol. 54, no. 11, pp. 1460–1466, Nov. 2005.

[31] L. Greco, D. Fontanelli, and A. Bicchi, "Almost sure stability of anytime controllers via stochastic scheduling," in *Proc. IEEE Int. Conf. Decision Control*, New Orleans, LA, Dec. 2007, pp. 5640–5645.

[32] Y. Fang, K. Loparo, and X. Feng, "Almost sure and δ-moment stability of jump linear systems," *Int. J. Control*, vol. 59, no. 5, pp. 1281–1307, 1994.

[33] P. Bolzern, P. Colaneri, and G. D. Nicolao, "On almost sure stability of discrete-time Markov jump linear systems," in *Proc. 43rd IEEE Conf. Decision Control*, 2004, vol. 3, pp. 3204–3208.

[34] A. Cervin, B. Lincoln, J. Eker, K.-E. Årzén, and G. Buttazzo, "The jitter margin and its application in the design of real-time control systems," in *Proc. 10th Int. Conf. Real-Time Embedded Comp. Syst. Appl.*, Gothenburg, Sweden, Aug. 2004, [CD ROM].

[35] D. Liberzon, J. P. Hespanha, and A. S. Morse, "Stability of switched systems: A Lie-algebraic condition," *Syst. Control Lett.*, vol. 37, no. 3, pp. 117–122, 1999.

[36] H. Konno, "A cutting plane algorithm for solving bilinear programs," *Math. Programming*, vol. 11, no. 1, pp. 14–27, 1976.

[37] J. von Neumann, "Über einökonomisches gleichungs-system und eine verallgemeinerung des brouwerschen fixpunktsatzes," in *Proc. Ergebnisse eines Mathmatischen Kolloquiums*, 1937, vol. 8, pp. 1945–1946.

[38] K.Åström and B. Wittenmark, *Computer Controlled Systems*. Englewood Cliffs, NJ: Prentice Hall, 1996.

[39] R. Hanus, M. Kinnaert, and J. L. Henrotte, "Conditioning technique, A general anti-windup and bumpless transfer method," *Automatica*, vol. 23, no. 6, pp. 729–739, 1987.

[40] A. B. Arehart and W. A. Wolovich, "Bumpless switching controllers," in *Proc. IEEE Int. Conf. Decision Control*, Kobe, Japan, Dec. 1996, vol. 2, pp. 1654–1655.

[41] L. Zaccarian and A. R. Teel, "The $\mathcal{L}_2$ (l_2) bumpless transfer problem: Its definition and solution," in *Proc. IEEE Int. Conf. Decision Control*, Atlantis, Paradise Island, Bahamas, Dec. 2004, vol. 5, pp. 5505–5510.

[42] J. P. Hespanha, P. Santesso, and G. Stewart, "Optimal controller initialization for switching between stabilizing controllers," in *Proc. IEEE Int. Conf. Decision Control*, New Orleans, LA, Dec. 2007, pp. 5634–5639.

[43] K. Furuta, M. Yamakita, and S. Kobayashi, "Swing-up control of inverted pendulum using pseudo-state feedback," *Proc. Inst. Mech. Eng. I. J. Syst. Control Eng.*, vol. 206, no. I4, pp. 263–269, 1992.

[44] M. Kvasnica, P. Grieder, and M. Baotić, "Multi-Parametric Toolbox (MPT)," 2004 [Online]. Available: http://control.ee.ethz.ch/ mpt/

[45] R. Bupp, D. Bernstein, and V. Coppola, "A benchmark problem for nonlinear control design: Problem statement, experimental testbed and passive, nonlinear compensation," in *Proc. Amer. Control Conf.*, 1995, pp. 4363–4367.

[46] D. A. Lind and B. Marcus, *An Introduction to Symbolic Dynamics and Coding*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

Luca Greco (M'05) received the M.S. degree in information engineering and the Ph.D. degree in automation, robotics and bioengineering from the University of Pisa, Pisa, Italy, in 2001 and 2005, respectively.

From 2005 to 2007, he was an Associate Researcher at the Interdepartment Research Center "E. Piaggio", University of Pisa. From 2007 to 2009, he was an Associate Researcher at the DIIMA, University of Salerno, Salerno, Italy. He is presently an Associate Researcher at LSS – Supélec, Gif-sur-Yvette, France. His main research interests are in hybrid and switching systems, embedded system control, networked control systems.

Daniele Fotanelli (M'10) received the M.S. degree in information engineering and the Ph.D. degree in automation, robotics and bioengineering from the University of Pisa, Pisa, Italy, in 2001 and 2006, respectively.

He was a Visiting Scientist with the Vision Lab, University of California at Los Angeles, from 2006 to 2007. From 2007 to 2008, he was an Associate Researcher with the Interdepartment Research Center "E. Piaggio," University of Pisa. In 2008, he joined as an Associate Researcher the Department of Information Engineering and Computer Science, University of Trento, Trento, Italy. His research interests include robotics and visual servoing, embedded system control, wireless sensor networks, networked and distributed control.

Antonio Bicchi (F'06) received the "Laurea" degree in mechanical engineering from the University of Pisa, Pisa, Italy, in 1984, and the Doctoral degree from the University of Bologna, Bologna, Italy, in 1989.

After a Post-Doctoral Fellowship at the Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, he joined the Faculty of Engineering, University of Pisa, in 1990. He is Professor of Systems Theory and Robotics in the Department of Electrical Systems and Automation (DSEA), University of Pisa, and the Director of the Interdepartmental Research Center "E. Piaggio" of the University of Pisa, where he has been leading the Automation and Robotics group since 1990. He has published more than 250 papers on international journals, books, and refereed conferences. His main research interests are in Theory and control of nonlinear systems, in particular hybrid (logic/dynamic, symbol/signal) systems; dynamics, kinematics and control of complex mechanical systems, including robots, autonomous vehicles, and automotive systems; haptics, and dexterous manipulation.