Adaptive Reservations for Feedback Control

D. Fontanelli*, L. Greco† and L. Palopoli*

Abstract— **In this paper, we start from an assigned control law with known probability distributions of its execution time. Our goal is to identify an optimised scheduling policy that allows us to strike an acceptable tradeoff between control performance and consumption of computation resource. The methodology we advocate is based on a resource reservation scheduler that allows us to control the budget of CPU time allocated to the task and to construct a dynamic model for the delays introduced by the computation. This model allows us to construct a feedback scheduler (adaptive reservations) that dynamically changes the bandwidth to fulfil the desired application goals minimising the computation resource requirements.**

I. Introduction

In modern embedded control systems computation and communication resources are increasingly shared among different tasks. This choice raises issues of non-trivial complexity. First and foremost, the real-time tasks implementing the feedback controllers undergo scheduling interference from the other tasks in the system introducing time varying delays in the feedback loop. This situation is not typically addressed in standard design procedures, where the delay is typically assumed null or negligible.

A reasonable way to deal with the issue is by the adoption of the so called time triggered paradigm [1]. The idea is to set a deadline for the computation (large enough to account for all possible scheduling interferences) and to constrain the communication between plant and controller to take place exactly on the deadline. This way the jitter is virtually nullified and the control law can be designed compensating for a fixed and known delay. Despite the evident simplicity of this approach, there can be a high price to pay in terms of performance. Indeed, if the CPU workload is subject to wide changes, the permanent and artificial introduction of the worst case delay can be too conservative a choice. On the other hand, several studies reveal that there can be a sensible performance gain in aggressive choices for the scheduling parameters (e.g., the sampling period) even in the face of a moderate occurrence of deadline misses [2], [3].

This viewpoint, though, is hardly defensible in industrial contexts in the absence of a "certifiable" level of performance guarantees for design techniques departing from the traditional setting of digital control design. To compensate for this lack, an intense research activity has been started in the last few years. Different authors have investigated on *how to make the design robust* against an irregular timing behaviour of the software implementation, focusing on such effects as packet dropout [4], [5], jitter in computation [6], [7] and time varying delays [8]. Another important thread of work has aimed at modifying the scheduling behaviour in overload conditions to accommodate the needs of the control tasks. Very significant in this class is the work of Marti [9], who propose to re-modulate the task periods in response to an overload condition. In the same direction goes the work of Lemmon and co-workers [10], who develop a Markov Chain model to model the possible activations of the control task that can be skipped without compromising stability. This model is built upon to develop a heuristic scheduling strategy.

As in some of the cited work, in this paper, we start from an *assigned* control law. Then we construct a Markov model that describes the timing behaviour of the scheduler, given the probability distributions of the computation time of the task. This model allows us to derive a stochastic model for the delays and can then be combined with the model of the plant, resulting in the definition of a Jump Linear System [11]. Notably, this model is parametric with respect to the scheduling parameters. Thereby, we can formalise the design as an optimisation problem (a Markov decision problem) in which the use of computation resources is minimised under the constraint that the closed loop system results asymptotically stable (for a suitable definition of stability). Since we assume the presence of a monitoring component that reports to the run-time execution environment the current state of the task (e.g., in terms of accumulated delays), our scheduling mechanism is an adaptive one: the scheduling parameters are dynamically adjusted. With this regard, our work bears a close resemblance with the adaptive reservation mechanism proposed in the multimedia community [12]. Crucial in this construction is the role of the scheduling algorithm. The adoption of a resource based scheduling algorithm [13] enables an exact modelling and control of the temporal behaviour of the task. Therefore, contrary to previous work [10], our scheduling proposal cannot be classified as a heuristic.

The paper is organized as follows. In Section II we describe the scheduling model and the model of computation adopted in the paper, and discuss the dynamic model describing the timing behaviour of the task implementing the controller. In Section III, we formalize the control problem considered in this paper. In Section IV we derive a stochastic model for the evolution of the closed loop system, connecting

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7 under grant agreement n° IST-2008-224428 "CHAT - Control of Heterogeneous Automation Systems"

*Department of Engineering and Information Science (DISI), University of Trento, Via Sommarive 14, 38050 Trento (TN), Italy. Phone: +390461883967. Fax:+390461882093. fontanelli,palopoli@disi.unitn.it

†LSS - Supélec, 3, rue Joliot-Curie, 91192 Gif sur Yvette, France lgreco@ieee.org.

the timing behaviour of the control task with the closed loop dynamics of the system. In Section V we formalize the optimization problem producing the design of the scheduler. In Section VI, we show the application of our methodology to a realistic case-study, comparing its performance with the one of the classic hard real-time policy. Finally, in Section VII we state our conclusions and announce our future work.

II. Platform model

In this paper, we consider a set of real-time tasks $\{\tau_i\}$ sharing a *processing unit* (CPU).A real-time task τ_i consists of a stream of jobs $J_{i,j}$. Each job $J_{i,j}$ arrives (becomes executable) at time $r_{i,j}$, and finishes at time $f_{i,j}$ after executing for a time $c_{i,\,j}$. Job $J_{i,j}$ is also characterized by a deadline $d_{i,j}$, that is respected if $f_{i,j} \leq d_{i,j}$, and is missed if $f_{i,j} > d_{i,j}$.

We focus on *periodic* tasks, where arrival times are spaced out by a *task period* T_i, i.e., $r_{i,j+1} = r_{i,j} + T_i$, and each activation time is also the deadline of the previous instance $d_{i,j} = r_{i,j} + T_i = r_{i,j+1}$. We assume that the period T_i is divided in $l < \infty$ time slices of length Δt, which is an integer multiple of the processor's clock period.

A. The scheduling algorithm

As multiple real-time tasks may be concurrently active at the same time, a scheduling mechanism is used to properly schedule the CPU. To this purpose, we advocate the use of *resource reservations*. Each task τ_i is associated a reservation (Q_i, R_i), with the meaning that τ_i is allowed to execute for Q_i (*budget*) time slices in every interval of length R_i (*reservation period*). Hence, R_i and Q_i are both integer multiple of the time slice Δt and $Q_i \in \{0, \ldots, R_i/\Delta t\}$. The bandwidth allocated to the task is $B_i = Q_i/R_i$ and it can be thought of as the fraction of CPU time allocated to the task. It is important not to confuse the reservation period R_i with the task period T_i: although $R_i = T_i$ is a perfectly reasonable assignment, it is often useful to set the reservation period so that $T_i = N_i R_i,\ N_i \in \mathbb{N}$. The particular implementation of the Resource Reservations approach that we consider is the Constant Bandwidth Server (CBS) [13]. In CBS, reservations are implemented by means of an Earliest Deadline First (EDF) scheduler which schedules tasks $\{\tau_i\}$ based on their *scheduling deadlines* $\{d_i^s\}$, dynamically managed by the CBS algorithm .

B. Dynamic model for the scheduler

A CPU reservation can be considered as a discrete-event dynamic system whose evolution is observed at the termination of each job $J_{i,\,j}$. In this work, we propose to use budget Q_i as a control input to control the temporal evolution of the task. Therefore, we will use the notation $Q_{i,\,j}$ to denote the budget allocated to the j^{th} job of the i^{th} task. The reservation period R_i is held constant and it can be used to decide the granularity of the CPU allocation (a smaller value for R_i corresponds to a more fluid allocation but to a greater overhead). As discussed next, by making this choice we can build a dynamic model describing the evolution of the QoS as a function of the control input $Q_{i,\,j}$ and of an exogenous disturbance term given by the allotted computation time $c_{i,\,j}$.

Instrumental to this construction is the definition of the *scheduling error* as the difference between the server scheduling deadline $d^s_{i,\,j}$ (evaluated at the finishing-time of each job) and the soft deadline of the task:

$$\epsilon_{i,\,j} \triangleq d^s_{i,j} - d_{i,j} = d^s_{i,j} - r_{i,j} - T_i. \qquad (1)$$

The server deadline is aligned with the end of the last server period used for the execution of job $J_{i,\,j}$. Therefore a positive value for $\epsilon_{i,\,j}$ means that $J_{i,\,j}$ finished in a reservation period beyond the deadline (it received) less bandwidth than it needed. Conversely, a negative value means that it finished in a reservation period preceding its deadline (the assigned bandwidth was greater than the task needed). A null value corresponds to a perfect match between the computation requirement of the task and the resource assignment. An approximation for the dynamic evolution of the scheduling error is given by [14]:

$$\epsilon_{i,\,j+1} = S(\epsilon_{i,\,j}) + \left\lceil \frac{c_{i,j+1}}{Q_{i,j+1}} \right\rceil R_i - T_i. \qquad (2)$$

where the function $S(\cdot)$ defined as:

$$S(x) \triangleq \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

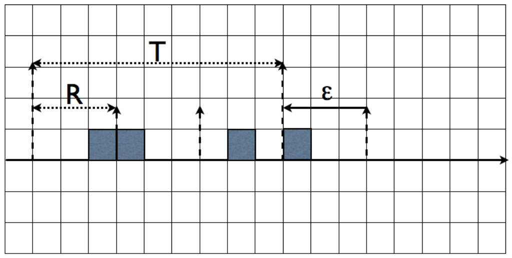
expresses the scheduling error due to the previous jobs which affect the present job.

a) Example: Consider a periodic task τ with period $T = 9\Delta t$ and computation time $c = 4$. Let it be scheduled through a reservation with $R = 3\Delta t$ and $Q = \Delta t$. A possible schedule for the first job is shown in Figure 1.(a). Since the task utilization $C/T = 4/9$ is larger than the reserved bandwidth $Q/R = 1/3$, the job terminates beyond the deadline. The scheduling deadline (i.e., the end of the reservation period) when the job finishes is equal to 12. Therefore, the job experiences a scheduling error equal to 3 (i.e., one reservation period).

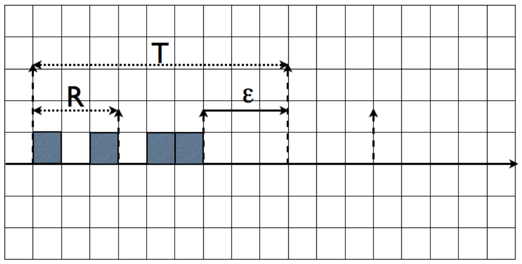
In Figure 1.(b), we consider the same situation, but with a reserved budget equal to 2. In this case, the reserved bandwidth (2/3) is greater than the task utilization and the job finishes one reservation period before the deadline experiencing a negative scheduling error equal to -3.

C. Model of Computation for Embedded Controllers

In the following discussion, we refer to a single control task τ managed by the CBS, hence we drop the i subscript in all the quantities related to τ. As a customary choice in classical digital control, in our setting the embedded controller is implemented as a real-time periodic task activated with a period $T = NR$. Every job J_j reads an input y_j from the plant and produces a control value u_j according to the following rules: 1) the input and output operations take place only at the start/termination of a server period, 2) although the activation of the jobs is periodic, the input y_j is sampled



(a)



(b)

Fig. 1. Schedule of the first job of the task in the example. (a) the task is scheduled through a reservation $(Q, R) = (1, 3)$. (b) the task is scheduled through a reservation $(Q, R) = (2, 3)$

at the end job J_{j-1}, 3) if the job finishes before the deadline (positive scheduling error) the release of the output u_j is deferred to the end of the period, 4) the output release can be delayed by an amount $\Delta_j = D_j R$ (an integer multiple of the server period), but if this delay becomes greater than the period T, the job J_j is dropped and a new job is activated. The formulation of these rules is partly due to the need for controlling the jitter (if τ finishes all of its jobs before its deadline, the input and output operations are exactly spaced by N server periods). Another reason for the formulation of rule 3 and 4 is to simplify the formulation of the model. Indeed, with this rule, the integer delay D_j can change for each job but it alway ranges in the finite set $\{0, \ldots, N\}$. Recalling equation (2) and considering the drop event, we can provide the following expression for D_j:

$$D_{j+1} \triangleq \frac{\epsilon_{j+1}}{R} = S^\star(D_j) + \left\lceil \frac{c_{j+1}}{Q_{j+1}} \right\rceil - N, \tag{3}$$

where

$$S^\star(x) \triangleq \begin{cases} N & \text{if } x \geq N \\ x & \text{if } N > x > 0 \\ 0 & \text{otherwise} \end{cases} .$$

III. The control problem

Let us consider a continuous-time, strictly proper, linear system and its discrete time equivalent (sampled with the reservation period R) given by:

$$\begin{aligned} \dot{x} &= \overline{A}x + \overline{B}u \\ y &= \overline{C}x \end{aligned} \quad \text{and} \quad \begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned} , \tag{4}$$

with $A \triangleq \mathrm{e}^{\overline{A}R}$, $B \triangleq \int_0^R \mathrm{e}^{\overline{A}(R-s)}\overline{B}\, ds$ and $C \triangleq \overline{C}$. The system is controlled by a task implementing a linear controller:

$$\begin{aligned} z_{j+1} &= A_c z_j + B_c y_j \\ u_j &= C_c z_j + D_c y_j, \end{aligned} \tag{5}$$

where j is the index of the j^{th} job. Such a controller is designed assuming an ideal periodic implementation with period $T = NR$.

Let us denote by $F_j \in \{0, \ldots, 2N\}$ the integer variable describing the number of server periods during which the control value u_{j-1} is held constant. In ideal conditions, where the control task has no delay, the control value is held constant for a whole period: $F_j = N$ (due to the unit delay between the output y_j sampling and the control u_j availability). If, instead, the $(j-1)^{th}$ job finished later with a delay $D_{j-1} > 0$, the control u_{j-1} (computed by the $(j-1)^{th}$ job) is held constant for $N - D_{j-1}$ server periods in the present task period ($N - D_{j-1}$ is the number of server periods remaining till the task period expires). If the j^{th} job has a delay $D_j > 0$, then u_{j-1} is held constant also for D_j server periods in the next task period. Therefore, $F_j = N - D_{j-1} + D_j$. According to this definition, we can write the controlled system dynamics in a job-based fashion instead of a time-based one as :

$$\begin{aligned} x_{j+1} &= A^{F_j} x_j + \sum_{t=0}^{F_j - 1} A^{F_j - t - 1} B u_{j-1} \\ y_j &= C x_j. \end{aligned} \tag{6}$$

In case of maximum delay ($D_j = N$), if the input u_{j-1} is available just at the end of the task period, then both the dynamics of the system (6) and of the controller (5) do not change. If, instead, the delay is greater than N, then a drop event takes place limiting the delay to N. The drop event can be managed either holding the previous control value (drop and hold), or zeroing it (drop and zero). In both cases we consider the controller state z_j to be held ($z_j = z_{j-1}$). Therefore, in the drop case, the dynamics of the controller (5) has to be modified, while the dynamics (6) is still unchanged.

A. Closed Loop Model

In order to properly describe the constant jitter delay (of N server periods) and the drop event, we introduce the state variable ξ_j and re-write the controller dynamics as follows:

$$\begin{aligned} z_{j+1} &= A_c z_j + B_c y_j \\ \xi_{j+1} &= C_c z_j + D_c y_j \\ u_j &= C_c z_j + D_c y_j \end{aligned} \quad \text{and} \quad \begin{aligned} z_{j+1} &= z_j \\ \xi_{j+1} &= \xi_j, \\ u_j &= \xi_j \end{aligned}$$

for constant jitter and drop and hold respectively. The drop and zero approach can be derived by the previous dynamics simply assuming $\xi_{j+1} = u_j = 0$.

Therefore, the closed loop system is represented by the following switching system:

$$\begin{aligned} w_{j+1} &= \tilde{A}_{\phi(j)} w_j \\ y_j &= \tilde{C} w_j \end{aligned} \tag{7}$$

where $w_j = [x_j^T, z_j^T, \xi_j]^T$ and $\tilde{C} = [C, 0, 0]$. The piecewise constant function $\phi : \mathbb{Z}_{\geq 0} \to \{0, \ldots, 3N+1\}$ rules the switchings among the different subsystems according to the delay evolution (3) and the drop policy. A rigorous description of such a function will be provided in the next sections. The structure of $\tilde{A}_{\phi(j)}$ can be easily derived replacing in (6) u_{j-1} with ξ_j. For $\phi(j) = 0, \ldots, 2N$, $2N+1$ closed loop matrices are thus obtained for the *regular* dynamics (no drop)

$$\tilde{A}_{\phi(j)} = \begin{bmatrix} A^{F_j} & 0 & \tilde{B}_{\phi(j)} \\ B_c C & A_c & 0 \\ D_c C & C_c & 0 \end{bmatrix}, \tag{8}$$

with $\tilde{B}_{\phi(j)} = \sum_{t=0}^{F_j-1} A^{F_j-t-1} B$ and $F_j = 0, \ldots, 2N$.

Additionally, $N+1$ dynamics after a drop and hold event are given by indices $\phi(j) = 2N+1, \ldots, 3N+1$ that generate

$$\tilde{A}_{\phi(j)} = \tilde{A}^{\mathrm{dh}}_{\phi(j)} = \begin{bmatrix} A^{F_j} & 0 & \tilde{B}_{\phi(j)} \\ 0 & I & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{9}$$

with $F_j = N, \ldots, 2N$ and the $\tilde{B}_{\phi(j)}$ is formally equal to the one given previously. The drop and zero matrix $\tilde{A}^{\mathrm{dz}}_{\phi(j)}$ is instead obtained by $\tilde{A}^{\mathrm{dh}}_{\phi(j)}$ zeroing the last element of the last row.

IV. Problem Formulation

Suppose the control law (5)) is given and suppose to have a stochastic description of the computation time c_j of the control task τ. Our objective is to define a reservation policy $\mathcal{P}(Q_j)$ which chooses the budget Q_j to achieve the following goals: 1) stability of the closed loop dynamics; 2) minimization of the bandwidth Q/R utilized by the system.

With the aim of fulfilling these goals, we provide a stochastic description of the delay D_j, and, hence, of the process $\phi(j)$ ruling the switchings. Then, a policy for the adaptive reservation technique is determined by solving an optimization problem.

A. Stochastic Description of the Delay

The sequence of the computation times of the control task τ is a stochastic process denoted as $\{c_j\}_{j\in\mathbb{Z}_{\geq 0}}$, where j is the job index. The discrete random variable c_j takes values in the set $L_c = \{1, \ldots, \bar{n}\}$, where $\bar{n}\Delta t$ is the worst-case execution time (WCET) of the task τ.

In order to derive a stochastic characterization of c_j, let us consider a generic continuous time probability density function (pdf) $f(\bar{c}(j))$, taking values in the continuous set $\bar{c}(j) \in [0, \bar{n}] \subset \mathbb{R}$. The discrete random variables c_j are thus given by a map $\mathcal{M} : [0, \bar{n}] \to L_c$, $\bar{c}(j) \mapsto c_j$, where $c_j = c$ if $\bar{c}(j) \in (c-1, c]$ with $c = 1, \ldots, \bar{n}$ and their distributions are inherited from the pdf $f(\bar{c}(j))$ and given by $\Pr\{c_j = c\} = \int_{c-1}^{c} f(x)dx$. Assuming that the continuous time process $\{\bar{c}(j)\}_{j\in\mathbb{Z}_{\geq 0}}$ is independent and identically distributed (i.i.d.), the process governing the discrete variables c_j is i.i.d. in its turn.

From (3) follows that $\{D_j\}_{j\in\mathbb{Z}_{\geq 0}}$ is itself a stochastic process that is related to the process $\{c_j\}_{j\in\mathbb{Z}_{\geq 0}}$, the budget Q_j and the value of the previous delay D_{j-1}. Our objective is to design a reservation policy $\mathcal{P}(Q_j)$, that is to define how to grant the budget to achieve the two aforementioned goals. We look for a time invariant policy, which can be state dependent, i.e. $\mathcal{P}(Q_j) = \mathcal{P}(Q\{D_{j-1}\})$. With these assumptions, the process describing the evolution of D_j and also the drop events is a finite-state homogeneous, discrete-time Markov chain (FSH MC). Let us name $\sigma(j)$ such a MC. It takes on values in the set $L_\sigma = \{0, \ldots, N+1\}$, with the meaning that if $\sigma(j) = q$ $\forall q \in \{0, \ldots, N\}$ then $D_j = q$, if $\sigma(j) = N+1$ then $D_j = N$ and a drop event takes place. The stochastic characterization of $\sigma(j)$ is given by the transition probability matrix $P = (p_{q,g})_{(N+2)\times(N+2)}$, $p_{q,g} \triangleq \Pr\{\sigma(j+1) = g \mid \sigma(j) = q\}$ and by the initial probability measure $\pi_\sigma(0) \in S^{N+2}$, where $S^{N+2} \triangleq \left\{ s = [s_1, \ldots, s_{N+2}] \in [0,1]^{N+2} \mid \sum_{i=1}^{N+2} s_i = 1 \right\}$ is the $(N+2)$-dimensional canonical stochastic simplex. The evolution of the probability distribution $\pi_\sigma(j)$ of the MC σ is then given by $\pi_\sigma(j+1) = \pi_\sigma(j)P$.

Recalling (3), for every q and g such that $0 \leq q \leq N$ and $1 \leq g \leq N$, we have

$$\begin{aligned} p_{q,g} &= \Pr\left\{ g = q + \left\lceil \frac{c_{j+1}}{Q\{q\}} \right\rceil - N \right\} \\ &= \Pr\left\{ \underline{a}_{q,g} < c_{j+1} \leq \overline{a}_{q,g} \right\} = \Pr\left\{ c_{j+1} \in V(q,g) \right\}, \end{aligned}$$

where the set $V(q,g) \triangleq (\underline{a}_{q,g}, \overline{a}_{q,g}]$, and the values $\underline{a}_{q,g} = \min\left(Q\{q\}(g-q+N-1), \bar{n}\right)$ and $\overline{a}_{q,g} = \min\left(Q\{q\}(g-q+N), \bar{n}\right)$. The set $V(q,g)$ can be empty if lower and upper bounds are both equal to $\bar{n}$. Moreover

$$p_{q,g} = \sum_{\forall c \in V(q,g)} \Pr\{c_{j+1} = c\} = \int_{V(q,g)} f(x)dx.$$

A special case is related to the probability to finish within the deadline ($g = D_{j+1} = 0$). In this case all the computation times from 0 to $Q\{q\}(N-q)$ for $0 \leq q \leq N$, describe the same event. Hence, $V(q,0) \triangleq (0,\ \min\left(Q\{q\}(N-q), \bar{n}\right)]$. Clearly, increasing the budget $Q\{q\}$ reduces the probability of terminating the job with some delay. Conversely, a high accumulated delay reduces probability to finish within the deadline. However, the dependence of the transition probability matrix from the budget is not immediate and it depends on the particular distribution of the computation times. Another important special case is related to the state $g = N+1$ (corresponding to a drop event, the cancellation of the job). Indeed

$$\begin{aligned} p_{q,N+1} &= \Pr\left\{ q + \left\lceil \frac{c_{j+1}}{Q\{q\}} \right\rceil - N > N \right\} \\ &= \Pr\left\{ c_{j+1} > Q\{q\}(2N-q) \right\}, \end{aligned}$$

therefore $V(q, N+1) \triangleq \left(\min\left(Q\{q\}(2N-q), \bar{n}\right), \bar{n}\right]$ for $0 \leq q \leq N$. Even the set $V(q, N+1)$ can be empty if $\min\left(Q\{q\}(2N-q), \bar{n}\right) = \bar{n}$. Finally, it is apparent that the expression for the probabilities $p_{N+1,g}$ for $0 \leq g \leq N+1$ after a drop event is formally equal to the corresponding $p_{N,g}$. The only difference is given by the values of the Q if they are state dependent (i.e. $Q\{N+1\} \neq Q\{N\}$).

$\bar{g} =$	0	g	$N+1$
$p_{q,\bar{g}}$	$Q\{q\}\frac{N-q}{\bar{n}}$	$Q\{q\}\frac{1}{\bar{n}}$	$1-Q\{q\}\frac{2N-q}{\bar{n}}$

TABLE I

TRANSITION PROBABILITY FOR THE FSH MC $\sigma(j)$ IN THE CASE OF UNIFORM PDF $f(\bar{c}(j))$.

If the FSH MC σ is also irreducible and aperiodic (thus it will be denoted by FSHIA MC), then there exists a unique invariant probability distribution (i.p.d.) $\bar{\pi}_\sigma$ corresponding to the steady-state probability distribution of the MC (i.e. $\lim_{h\to\infty}\pi_\sigma(h) = \bar{\pi}_\sigma$ for any $\pi_\sigma(0)$).

It can be easily verified that for σ to be a FSHIA MC it is sufficient to have no zero entries on the diagonal, on the first subdiagonal and on the first superdiagonal of the transition matrix P. Indeed, a tridiagonal non-negative matrix P with no zero elements on its diagonals is such that $P^m > 0$ for $m > n-1$ with n the dimension of P.

The previous condition (Frobenius's test for primitivity [15]) requires that the integration domains $V(q,q-1)$, $V(q,q)$ and $V(q,q+1)$ related to the entries on the three main diagonals of P are not empty, and that the pdf $f(\cdot)$ is not identically null on them. The latter requirement can be ensured by asking that $f(\cdot)$ is such that $\int_{c-1}^{c} f(x)dx \neq 0 \ \forall c \in \{1,\dots,\bar{n}\}$. To satisfy the former requirement we impose a lower bound on $V(q,q+1)$ and an upper bound on $V(q,q-1)$. That is $\min(Q\{q\}N,\bar{n}) < \bar{n}$ and $\min(Q\{q\}(N-1),\bar{n}) > 0$, which yield

$$\begin{aligned} &0 < Q\{q\} < \left\lfloor \frac{\bar{n}}{N} \right\rfloor,\ \forall q \in \{0,\dots,N\}, \\ &0 < Q\{N+1\} < \left\lfloor \frac{\bar{n}}{N-1} \right\rfloor. \end{aligned} \tag{10}$$

Example 1: We want to see how the budget Q_j influences the steady state probability of σ. Consider a uniform pdf $\mathcal{U}(0,\bar{n})$ for the i.i.d. computation times, whose transition probability matrix entries are computed using the previously depicted methodology and summarized in Table I. If the budget given to the platform is equal for each state and time invariant, i.e., $Q_j = Q$, and we define $p_0 = \frac{N(N-1)}{2}\frac{Q^2}{\bar{n}(\bar{n}-NQ)}$, we get $\bar{\pi}_\sigma(Q) = [p_0, \frac{Q}{\bar{n}}, \dots, \frac{Q}{\bar{n}}, \frac{\bar{n}-NQ}{\bar{n}} - p_0]^T$. On the other hand, if the budget is time invariant but state dependent, i.e., $Q_j = Q\{q\}$, $q \in \{1,\dots,N+1\}$ and we define $d_0 = d_1 + d_2 - d_3$, with $d_1 = Q\{0\}\left(\sum_{k=1}^{N} kQ\{k\} - N\bar{n} - N^2 Q\{N+1\}\right)$, $d_2 = Q\{N+1\}\left(\sum_{k=1}^{N-1}(N-k)Q\{k\}\right)$ and $d_3 = \bar{n}\left(\sum_{k=1}^{N} Q\{k\} - \bar{n} - NQ\{N+1\}\right)$, we get for $\bar{\pi}_\sigma(Q)^T$:

$$\frac{1}{d_0}\begin{bmatrix} Q\{N+1\}\left(\sum_{k=1}^{N-1}(N-k)Q\{k\}\right) \\ Q\{N+1\}(\bar{n}-NQ\{0\}) \\ \vdots \\ Q\{N+1\}(\bar{n}-NQ\{0\}) \\ Q\{0\}\left(\sum_{k=1}^{N} kQ\{k\} - N\bar{n}\right) - \bar{n}\left(\sum_{k=1}^{N} Q\{k\} - \bar{n}\right) \end{bmatrix}.$$

B. Stochastic Description of the Switching Process

In this section we derive a stochastic description for the process $\{\phi(j)\}_{j\in\mathbb{Z}_{\geq 0}}$ ruling the switchings of the closed loop system (7), which is thus a Stochastic Jump Linear System. The process $\{\phi(j)\}_{j\in\mathbb{Z}_{\geq 0}}$ takes values in the set $L_\phi = \{0,\dots,3N+1\}$ as follows:

1) $\phi(j) = N - \sigma(j-1) + \sigma(j)$ for $\sigma(j-1), \sigma(j) < N+1$;
2) $\phi(j) = 2N - \sigma(j-1)$ for $\sigma(j-1) < N+1$ and $\sigma(j) = N+1$;
3) $\phi(j) = 2N + 1 + \sigma(j)$ for $\sigma(j-1) = N+1$ and $\sigma(j) < N+1$;
4) $\phi(j) = 3N+1$ for $\sigma(j-1), \sigma(j) = N+1$.

According to this definition, we can define for each $i \in L_\phi$ a set of pair(s) $S_i = \{(a,b) \in L_\sigma \times L_\sigma \mid \sigma(j-1) = a,\ \sigma(j) = b \text{ s.t. } \phi(j) = i\}$. Hence we can write

$$\pi_{\phi_i}(j) = \Pr\{\phi(j) = i\} = \sum_{(a,b)\in S_i} \Pr\{\sigma(j) = b, \sigma(j-1) = a\}. \tag{11}$$

In order to provide the stochastic characterization of ϕ, we define another process describing the evolution of two consecutive steps of the MC σ $\hat{\sigma}(j) = (\sigma(j), \sigma(j-1))$ taking values in the set $L_{\hat{\sigma}} = L_\sigma \times L_\sigma$. Hence we have

$$\begin{aligned} \hat{\pi}_{ab}(j) &= \Pr\{\hat{\sigma}(j) = (a,b)\} = \Pr\{\sigma(j) = b,\ \sigma(j-1) = a\} \\ &= \Pr\{\sigma(j) = b \mid \sigma(j-1) = a\}\Pr\{\sigma(j-1) = a\} \\ &= p_{ab}\pi_{\sigma_a}(j-1). \end{aligned}$$

Recalling the definition of π_σ we have $\pi_{\sigma_a}(j-1) = \sum_{c=0}^{N+1} p_{ca}\pi_{\sigma_c}(j-2) = \sum_{c=0}^{N+1}\hat{\pi}_{ca}(j-1)$, and we can write $\hat{\pi}_{ab}(j) = p_{ab}\sum_{c=0}^{N+1}\hat{\pi}_{ca}(j-1) = \hat{\pi}(j-1)v_a p_{ab}$, with $\hat{\pi} = [\hat{\pi}_{00}, \hat{\pi}_{01}, \dots, \hat{\pi}_{N+1,N}, \hat{\pi}_{N+1,N+1}]$ and v_a is the a-th column of the matrix $V \in \{0,1\}^{(N+2)^2\times(N+2)}$ given by $V = [I_{N+2}, \dots, I_{N+2}]^T$. Hence, the generic $\hat{\pi}_{ab}(j)$ can be rewritten as $\hat{\pi}(j) = \hat{\pi}(j-1)\hat{P}$, with $\hat{P} = [v_0 p_{00}, v_0 p_{01}, \dots, v_{N+1}p_{N+1,N+1}]$, thus revealing that $\hat{\sigma}$ is a finite-state homogeneous MC. Unfortunately $\hat{\sigma}$ is not always irreducible even if σ is irreducible. This fact means that $\hat{\sigma}$ has some transient states whose steady state probability is zero. However, we can still work with $\hat{\sigma}$ in the light of the following Proposition.

Proposition 1: If σ is a FSHIA MC with unique steady state distribution $\bar{\pi}_\sigma = [\bar{\pi}_{\sigma_0}, \bar{\pi}_{\sigma_1}, \dots, \bar{\pi}_{\sigma_{N+1}}]$, then $\hat{\sigma}$ is a FSH MC with a unique steady state distribution $\bar{\hat{\pi}} = [\bar{\hat{\pi}}_{00}, \bar{\hat{\pi}}_{01}, \dots, \bar{\hat{\pi}}_{N+1,N}, \bar{\hat{\pi}}_{N+1,N+1}]$ with $\bar{\hat{\pi}}_{ab} = p_{ab}\bar{\pi}_{\sigma_a}$.

Proof: Let us define the following matrices, whose upper left entries have indices $(0,0)$ for semplicity of notation:

$$L_i = \begin{cases} v_i & i\text{-th column} \\ 0 & \text{otherwise} \end{cases},\quad E_i = \begin{cases} 1 & (i,i) \text{ entry} \\ 0 & \text{otherwise} \end{cases},$$

$$R_i = \begin{cases} I_{N+2} & [i(N+2)]\text{-th to} \\ & [(i-1)(N+2)-1]\text{-th column} \\ 0 & \text{otherwise} \end{cases},$$

where $L_i \in \{0,1\}^{(N+2)^2\times(N+2)}$, $R_i \in \{0,1\}^{(N+2)\times(N+2)^2}$ and $E_i \in \{0,1\}^{(N+2)\times(N+2)}$. It is easy to verify that

$$\hat{P} = \sum_{i=0}^{N+1} L_i P R_i. \tag{12}$$

At this point, the following Lemma is needed.

Lemma 1: For every $m > 0$, the $(m+1)$-th power of $\hat{P}$ is given by

$$\hat{P}^{m+1} = VP^m H$$

with $H = \sum_{j=0}^{N+1} E_j P R_j$.

Proof: Let us start computing

$$\hat{P}^2 = \sum_{i=0}^{N+1} L_i P R_i \sum_{j=0}^{N+1} L_j P R_j = \sum_{i=0}^{N+1} L_i P \sum_{j=0}^{N+1} E_j P R_j$$
$$= VPH,$$

where we used the facts that $R_i L_j = E_j \ \forall i, j$ and $\sum_{i=0}^{N+1} L_i = V$. Assuming that $\hat{P}^m = VP^{m-1}H$ and noting that $\sum_{j=0}^{N+1} E_j = I_{N+2}$ we can prove the thesis by induction. ■

In the light of the previous lemma and of the structure of H is easily to verify that if $p_{ab} \neq 0, \ \forall a, b$ then $\hat{P}$ is irreducible and aperiodic. Indeed, P is irreducible and aperiodic, hence there exists $m > 0$ such that $P^m > 0$ (i.e. P^m has no zero entries) and $\hat{P}^{m+1} > 0$. If some $p_{ab} = 0$, instead, $\hat{P}$ is reducible. In order to prove that the steady state distribution is unique, we must prove that there exists a unique ergodic class. Let us assume that there exist some $p_{ab} = 0$ but P is irreducible and aperiodic. There exists a permutation matrix U such that $U^T \hat{P} U = \begin{bmatrix} 0 & \hat{P}_{12} \\ 0 & \hat{P}_{22} \end{bmatrix}$. We must prove that $\hat{P}_{22}$ is irreducible and aperiodic to assert the existence of a unique ergodic class. Noting that $U^T U = I_{(N+2)^2}$, we have $\left(U^T \hat{P} U\right)^{m+1} = U^T \hat{P}^{m+1} U = \begin{bmatrix} 0 & * \\ 0 & \hat{P}_{22}^{m+1} \end{bmatrix}$. By direct inspection it can be verified that if $m > 0$ is such that $P^m > 0$, then $U^T V P^m > 0$ and $HU = \begin{bmatrix} 0 & H_2 \end{bmatrix}$ is such that $U^T V P^m H U$ gives $\hat{P}_{22}^{m+1} > 0$. Thus $\hat{P}_{22}$ is irreducible and aperiodic. The unique steady state distribution in thesis can be obtained in both cases ($\hat{P}$ irreducible and reducible) simply considering

$$\Gamma_{(N+2)^2}(\hat{\bar{\pi}}) = \lim_{m \to \infty} \hat{P}^{m+1} = V \lim_{m \to \infty} P^m H = V \Gamma_{N+2}(\bar{\pi}_\sigma) H,$$

whereas $\Gamma_a(b)$ is the matrix of a rows all equal to the row vector b. ■

Dealing with systems whose switchings are driven by stochastic processes, we turn on a stochastic definition of stability.

Definition 1: [AS-stability] Let us consider a FSHIA MC $\sigma(t)$ taking values in the set $\{1, \ldots, m\}$ and characterized by the transition probability matrix P and the initial distribution $\pi_\sigma(0)$. The Jump Linear System $x_{t+1} = A_{\sigma(t)} x_t$ with $x \in \mathbb{R}^M$, $A_i \in \mathbb{R}^{M \times M}$, $i \in \{1, \ldots, m\}$ is said almost surely (exponentially) stable if for any $x(0) \in \mathbb{R}^M$ and any initial distribution $\pi_\sigma(0)$ the following holds

$$\Pr\left\{\lim_{t \to \infty} \|x_t\| = 0\right\} = 1.$$

Theorem 2 (1–step average contractivity): [16] If there exists a matrix norm $\|\cdot\|$, such that

$$\xi_1 = \prod_{i \in \{1, \ldots, m\}} \|A_i\|^{\overline{\pi}_{\sigma_i}} < 1 \tag{13}$$

then the MJLS of Definition 1 is AS-stable.

The previous stability definition and criterion are given for a FSHIA MC and in terms of its unique i.p.d. The process ruling the switchings of our Jump Linear System (7) is not directly a FSHIA MC, but its distribution is linearly related to that of the FSH MC $\hat{\sigma}$. Indeed, recalling (11) we can write $\pi_\phi(j) = \hat{\pi}(j) W$ for a suitable matrix $W \in \{0, 1\}^{(3N+2) \times (N+2)^2}$. Hence, the steady state distribution needed by the almost sure stability criterion is given by

$$\bar{\pi}_\phi = \hat{\bar{\pi}} W. \tag{14}$$

V. Optimal Reservation Policy for System Stability

We are now in condition to formulate an optimal synthesis problem to solve the problem introduced at the beginning of Section IV. We want to find a reservation policy $\mathcal{P}(Q)$ ensuring the AS-stability of the closed loop system (7) and minimizing a cost index related to the bandwidth B.

A suitable cost index, which provides a stochastic average of the budget provided by the scheduler, is the long-run expected value of the budget itself. Let us define the stochastic process $\{q(j)\}_{j \in \mathbb{Z}_{\geq 0}}$ as the sequence of budgets assigned to the control task τ by the scheduler. Such a process takes values in the set $\{Q\{0\}, \ldots, Q\{N+1\}\}$ representing all the possible state dependent budget values. Collecting all these values in a vector $Q = [Q\{0\}, \ldots, Q\{N+1\}]^T$ and recalling the meaning of the FSHIA MC σ, the index cost can be written as $\lim_{j \to \infty} \mathrm{E}\{q(j)\} = \bar{\pi}_\sigma Q$. By definition $\bar{\pi}_\sigma$ is a function of the budget Q, hence the previous one is a nonlinear cost index (see for instance Example 1).

Regarding the stability requirement, we re-write the 1-step average contractivity condition (13) for the closed loop system (7) in terms of logarithms. To this end, we define $v = \left[\ln\left(\left\|\tilde{A}_0\right\|\right), \ldots, \ln\left(\left\|\tilde{A}_{3N+1}\right\|\right)\right]^T$, in order to have $\sum_{i=0}^{3N+1} \bar{\pi}_{\phi_i} \ln\left(\left\|\tilde{A}_i\right\|\right) = \bar{\pi}_\phi v = \hat{\bar{\pi}} W v < 0$. Even $\hat{\bar{\pi}}$ is a function of the budget Q by means of P and $\bar{\pi}_\sigma$, as clearly shown in Proposition 1. In order to complete the optimal problem formulation we must add the constraints (10) on the values of the budget. Assuming that a fraction of each server period R has to be reserved for the execution of other tasks than τ, then each $Q\{q\}$ must be less than a pre-specified $Q_{\max}$. Therefore, defining the vectors $\underline{Q} = [0, \ldots, 0]^T$ and $\overline{Q} = [a, \ldots, a, b]^T$, where $a = \max\left(\left\lfloor \frac{\bar{n}}{N} \right\rfloor, Q_{\max}\right)$ and $b = \max\left(\left\lfloor \frac{\bar{n}}{N-1} \right\rfloor, Q_{\max}\right)$, we can write the constraints on the budget as $\underline{Q} < Q < \overline{Q}$. Summarizing we have the following non linear optimization problem in the unknown vector Q:

$$\textit{ORP Problem} \ \min_Q \bar{\pi}_\sigma(Q) Q \ \text{ s.t.}$$
$$\hat{\bar{\pi}}(Q) W v < 0$$
$$\underline{Q} < Q < \overline{Q}$$

VI. Simulation Results

The effectiveness of the proposed approach is shown in this Section by simulations. The mechanical system chosen for the simulation results is a Furuta pendulum with zero

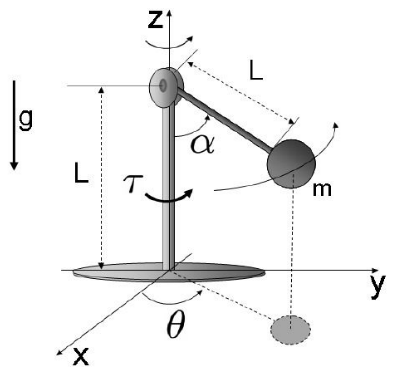


Fig. 2. The Furuta pendulum adopted for the simulations.

$G(s)$	$\frac{7.435}{s(s^2+34.63)}$
$C(z)$	$\frac{-35.7517z(z^2-1.99z+0.9919)}{(z-6\cdot 10^{-5})(z-0.8106)(z^2-1.85z+0.8811)}$

TABLE II

CONTINUOUS TRANSFER FUNCTION OF THE FURUTA PENDULUM WITH THE RELATIVE DISCRETE TIME CONTROLLER.

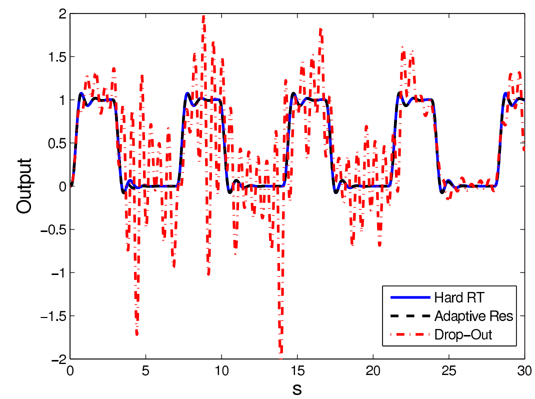


Fig. 3. Tracking output for the nominal (solid), adaptive reservation (dotted) and hard real–time (dashed) situations.

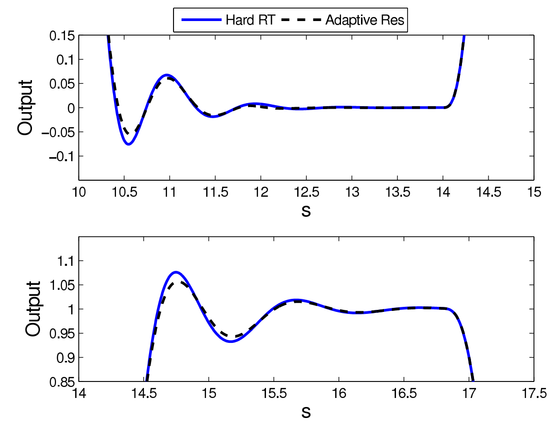


Fig. 4. Zoomed subsets of the tracking experiment shown in Figure 3.

offset [17] (see Figure 2). The continuous time transfer function $G(s)$ between the input torque τ and the output angle α and the discrete controller $C(z)$ are reported in Table II. The controller $C(z)$ is obtained using a systematic Linear Quadratic Gaussian design assuming a constant unit delay between the controller and the system. The system is sampled with a period equal to $T = 10^{-2}$ s.

For the platform, we refer to the Example 1. The task period T is divided in 4 server periods R, and each server period is divided in 20 time slices of length $\Delta T = 125$ μs, with a server period length equal to 2.5 ms. The computation times are given by i.i.d. processes with uniform distributions $c \sim \mathcal{U}(0, \bar{n})$, where $\bar{n} = 80$ time slices (i.e., equal to the period T). In this condition, a hard real–time scheduler based on the WCET of the control task, which reserves all of the 20 time slices for each server period, would guarantee the ideal behavior, i.e., the controller $C(z)$ is closed in loop with the plant $G(s)$ exactly every T seconds.

The ORP problem has been set up constraining the budget Q between $\underline{Q} = [0, \ldots, 0]^T$ and $\overline{Q} = [20, \ldots, 20]^T$ and allowing for a maximum delay equal to T. The solution has produced a final cost index value equal to 15.85 and a state dependent budget $Q = [16, 16, 15, 14, 13, 12]^T$. Interestingly, the optimal budget decreases as the delay increases and it is minimal after a drop event. This is because the adaptive reservation policy tries to keep the interval between two jobs constant in time.

The adaptive policy for the budget Q is then compared with the ideal behavior (in which the budget is greater than the WCET of the task) in a tracking problem in which the input signal is a square wave of period 7 s of unitary amplitude and duty cycle of 40% (Figure 3, dashed and solid lines respectively). A first remark, the adoption of a hard real–time scheme surely corresponds to a waste of computational resources since the proposed adaptive reservation policy saves almost 20% of the computational budget with a negligible difference in terms of closed loop performance (see the zoomed picture of Figure 4). Notably, such results are obtained for a uniform distribution of the computation times. We expect a much better performance of the adaptive policy in the more typical case in which the distribution of the computation times has a small variance and long tails.

Very interesting is also the comparison with a drop-out policy: the task is reserved a fixed budget of $\max(Q) = 16$ and the jobs are dropped whenever the deadline constraint is not satisfied. It is evident from Figure 3, dash dotted line, that our adaptive policy, in which control input are delayed rather than dropped, has a better performance. In the proposed example the closed loop system resulting from the drop-out policy resulted even unstable for some sequences of the computation times.

A third remark is related to the total computation time spent for the three examples: hard real–time, adaptive reservation and scheduler with drop–out. The time spent in the third case is lower than the other two approaches, since a higher number of drop events is generated (see Figure 5, dash–dotted line). The computation time spent for the hard

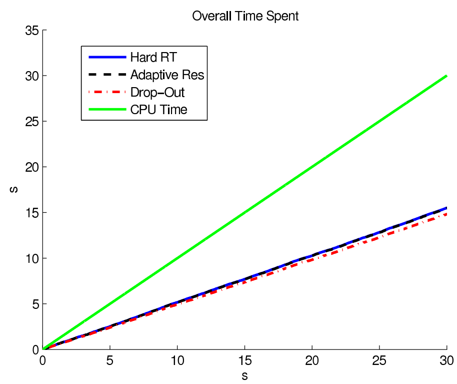


Fig. 5. Computation time spent for the reservation and hard real–time policies.

real–time scheduler and the adaptive reservation scheme is almost the same, despite of the difference in the budget. The remarkable characteristic of the adaptive reservation scheme here presented is to defer the computations to task periods in which the computation time is not demanding, tolerating delays and occasional computation drops but preserving the stability and the performance of the closed loop system.

VII. Conclusion

In this paper, we have considered the problem of scheduling a control task. Starting from an assigned control law and from a stochastic description of its execution time, we derive an adaptive scheduling policy that allows us to attain stability for the system and to minimise the consumption of computation resources.

Our paper is, in our evaluation, the first step in a promising direction. There are several issues reserved for future investigation, the most important being the analysis of alternative options for the desired system properties. We envisage that the use of second moment stability or of other properties related to the system performance could lead us to considerable resource savings. Another interesting direction is to consider multiple control loop that execute concurrently.

References

[1] H. Kopetz and G. Bauer, "The time-triggered architecture," *Proceedings of the IEEE*, vol. 91, no. 1, pp. 112–126, 2003.

[2] L. Palopoli, L. Abeni, G. Buttazzo, F. Conticelli, and M. Di Natale, "Real-time control system analysis: an integrated approach," 2000, pp. 131 –140.

[3] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Arzen, "How does control timing affect performance? analysis and simulation of timing using jitterbug and truetime," *Control Systems Magazine, IEEE*, vol. 23, no. 3, pp. 16 – 30, june 2003.

[4] J. Nilsson and B. Bernhardsson, "Analysis of real-time control systems with time delays," vol. 3, dec 1996, pp. 3173 –3178 vol.3.

[5] Q. Ling and M. Lemmon, "Robust performance of soft real-time networked control systems with data dropouts," vol. 2, dec. 2002, pp. 1225 – 1230 vol.2.

[6] P. Marti, J. Fuertes, G. Fohler, and K. Ramamritham, "Jitter compensation for real-time control systems," dec. 2001, pp. 39 – 48.

[7] B. Lincoln and A. Cervin, "Jitterbug: a tool for analysis of real-time control performance," vol. 2, dec. 2002, pp. 1319 – 1324 vol.2.

[8] C.-Y. Kao and A. Rantzer, "Stability analysis of systems with uncertain time-varying delays," *Automatica*, vol. 43, no. 6, pp. 959–970, Jun. 2007.

[9] G. Buttazzo, M. Velasco, and P. Marti, "Quality-of-control management in overloaded real-time systems," *Computers, IEEE Transactions on*, vol. 56, no. 2, pp. 253 –266, feb. 2007.

[10] T. Chantem, X. S. Hu, and M. Lemmon, "Generalized elastic scheduling for real-time tasks," *Computers, IEEE Transactions on*, vol. 58, no. 4, pp. 480 –495, april 2009.

[11] M. Mariton, *Jump linear systems in automatic control*. CRC, 1990.

[12] L. Palopoli, T. Cucinotta, and A. Bicchi, "Quality of service control in soft real-time applications," in *Proc. of the IEEE 2003 conference on decision and control (CDC02)*, Maui, Hawai, USA, December 2003.

[13] L. Abeni and G. Buttazzo, "Integrating multimedia applications in hard real-time systems," in *Proc. of the* 19^{th} *IEEE Real-Time Systems Symposium (RTSS 1998)*, Madrid, Spain, December 1998.

[14] L. Abeni, L. Palopoli, G. Lipari, and J. Walpole, "Analysis of a reservation-based feedback scheduler," in *Proc. of the Real-Time Systems Symposium*, Austin, Texas, November 2002.

[15] C. Meyer, *Matrix analysis and applied linear algebra*. Society for Industrial Mathematics, 2000.

[16] 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.

[17] K. Furuta, M. Yamakita, and S. Kobayashi, "Swing-up control of inverted pendulum using pseudo-state feedback," *Proceedings of the Institution of Mechanical Engineers. Pt.I. Journal of Systems and Control Engineering*, vol. 206, no. I4, pp. 263–269, 1992.