On the application of hybrid control to CPU Reservations

Luigi Palopoli, Luca Abeni, Giuseppe Lipari,
{palopoli, luca, lipari}@sssup.it ⋆

ReTiS Laboratory - Scuola Superiore S. Anna
Pisa, Italy

Abstract. An important class of soft real-time applications require dynamic allocation of computational resources in order to comply with their quality of service (QoS) requirements. These applications are characterised by large fluctuations in their computation time requirements. One of the biggest problems in such systems is how to assign the bandwidths to the software tasks so that every task meets its QoS requirements and computational resources are not wasted.
In this paper, we present a novel feedback scheduling controller based on a scheduling strategy called resource reservation. First, we model the scheduler as a discrete time switching system; then, we present hybrid control techniques for the design of the feedback scheduler; finally, we report simulation results that show the effectiveness of our approach.

1 Introduction

An emerging class of real-time systems requires the dynamic allocation of computational resources to time-sensitive applications realised by software tasks. Important examples are multimedia streaming programs video/audio players, software sound mixers, etc. Other examples are embedded systems used in data-intensive contexts, where relatively high volumes of sensor data are flowing and must be processed and analysed in real time (i.e. radar systems).

The amount of computation time, and more generally of hardware/software resources, required by this class of applications presents large fluctuations. Design approaches based on classical hard real-time techniques, being based on worst case assumption, can be overly conservative. Moreover, occasional failures in respecting the timing constraints attached to a task do not necessarily lead to catastrophic consequences. However, if failures are too frequent the Quality of Service (QoS) provided by the application degrades beyond acceptable limits. As an example, for a multimedia application it is not necessary to decode every frame within a fixed interval as long as fluctuations in the decoding rate do not overcome the threshold of human perception. Constraints of this type are referred to as QoS constraints.

When multiple real-time tasks of this type share the same CPU, guaranteeing simultaneous compliance with their QoS constraints is a challenging goal, for which a

⋆ Partial support from E.C. Project OCERA, IST-2001-35102

properly designed CPU scheduler can provide considerably better results than schedulers/resource allocators used in conventional operating systems. In this context, a fundamental requirement is the enforcement of *temporal protection*: i.e. each task has to be guaranteed a certain bandwidth independently of the fluctuations of CPU requirements of the other tasks in the system. If this property is respected, each task executes as if it were on a dedicated slower processor. The so called *resource reservation techniques*, first proposed in [12] for the CPU, proved to be very effective in providing temporal isolation, and have been implemented in a number of different systems using different scheduling algorithms [14, 1, 7, 15].

One of the biggest problems of CPU reservation systems is the selection of the fraction (or bandwidth) of the CPU assigned to each task. In presence of wide variations of the required computation time, one could either give the task too low or too high a bandwidth. In the former case the system experiences unacceptable degradations in the offered QoS, whereas in the latter case system resources are wasted. To cope with this problem, many authors proposed the use of feedback control mechanism inside the operating system. A first proposal of this kind for time sharing systems dates back to 1962 [5]. More recently, feedback control techniques have been applied to real-time scheduling [13, 16, 9, 11] and multimedia systems [17, 2]. However, little theoretical analysis of such mechanisms has been provided. One of the main problems that hinders this type of analysis is the lack of a realistic and analytically tractable model for "the plant", i.e. for a real-time scheduler. In [3], this gap has been filled in for CPU reservation schedulers: the authors proved that a CPU reservation scheduler for a task can realistically be modelled as a switching and parametric dynamic system (the varying parameter being the computation time of each activation of the task).

In this paper the problem of analytical design for the controller is tackled. As a first step toward a more general theory, and for the sake of simplicity, we consider here the case of a single periodically activated task with varying computation time. Also, we assume that the internal state of the scheduler is measurable. Under these assumptions, the resulting system is piecewise affine. The control design has to be carried out taking into account QoS requirements for the task and physical limitation of the control action (the assigned bandwidth cannot exceed the total power of the processor). As a solution to this problem, we propose a switching dynamic controller and offer techniques for synthesising its parameters and analysing its performance.

The QoS requirements and the physical constraints (input saturation) can be captured defining a polyhedral region in the state space of the closed loop system. The proposed analysis problem is aimed at the construction of an invariant set for the closed loop system that is entirely contained in the polyhedral "safe" set. The solution that we advocate is based on ellipsoidal reachability analysis. As far as the synthesis problem is concerned, we show that the problem is one of robust static output feedback synthesis. The application of the devised methodology to a general situation where multiple feedback control loops operate on the same CPU and the internal state of the scheduler is not accessible are discussed throughout the paper.

Notation. In the sequel, symbol $\succ$ ($\succeq$) denotes that a matrix is positive definite (semidefinite). Similarly, symbol $\prec$ ($\preceq$) denotes that a matrix is negative definite (semidefinite). Let $Q = Q^T \succ 0$, by $\mathcal{E}(Q, \hat{x})$ we will denote the ellipsoid $(x - \hat{x})^T Q^{-1} (x - \hat{x}) \leq 1$.

We recall that given a set of matrices $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_m$ with $\mathcal{A}_i = \mathcal{A}_i^T$ and a vector of parameters $x = [x_1, \ldots, x_m]^T$, the expression

$$\mathcal{A}_0 + x_1\mathcal{A}_1 + \ldots + x_m\mathcal{A}_m \preceq 0$$

is called a linear matrix inequality (LMI). Finding a feasible solution vector to a LMI is a problem that can be solved in polynomial time by means of convex optimisation [4]. In this paper, we will make frequent use of LMI based analysis techniques.

2 Problem Presentation

Definitions. Before developing a formal model for a reservation-based system, we need to introduce some definitions. In particular, we use the *real-time task model*, which associates a temporal constaint, called *deadline*, to each execution of a task. If these temporal constraints are violated, it means that the amount of resource associated with the task is insufficient, and the size of the reservation should be increased.

According to the real-time task model, a task $\mathcal{T}_i$ is a stream of jobs $J_i(k)$. Each job $J_i(k)$ arrives (becomes executable) at time $r_i(k)$, and finishes at time $f_i(k)$ after executing for a time $c_i(k)$. Moreover, $J_i(k)$ is characterised by a deadline $d_i(k)$, that is respected if $f_i(k) \leq d_i(k)$, and is missed if $f_i(k) > d_i(k)$.

For the sake of simplicity, we will only consider *periodic tasks*, in which $r_i(k+1) = r_i(k) + T_i$, where T_i is the *task period*. Moreover, we will assume that $d_i(k) = r_i(k) + T_i$; hence, $r_i(k+1) = d_i(k)$.

Our goal is to provide support for time-sensitive application in which a deadline miss can degrade the QoS of the task but does not have any catastrophic consequence. Therefore, we will consider *soft deadlines*, as opposed to *hard deadlines*: the goal of a soft real-time system is to control the number of deadline misses, whereas in a hard real-time system even a single deadline miss is not acceptable.

Reservation-Based Scheduling. A reservation is a pair (B_i, T_i^s), where B_i is the fraction of resource utilisation dedicated to task $\mathcal{T}_i$, and T_i^s is the *period* of the reservation: task $\mathcal{T}_i$ will be allowed to use the resource for $Q_i = B_i T_i^s$ units every period T_i^s. Q_i is also called *budget* or *capacity* of the reservation. It is important not to confuse the reservation period T_i^s with the task period T_i. Intuitively, the reservation period T_i^s can be seen as the *temporal granularity* of the reservation: the smaller is T_i^s, the more precise is the allocation of the resource to task $\mathcal{T}_i$ in the system. It is possible to define a reservation for all kind of resources. In this paper we will consider only CPU reservations: however, most of the results presented can easily be extended to other kind of resources (network, disk, etc.).

A reservation mechanism is often implemented on the top of a "host" scheduler enforcing a real-time policy (such as Rate monotonic or Earliest Deadline First [10]). If $\sum_i B_i \leq U_l$, with $U_l \leq 1$ depending on the "host" scheduling algorithm, then each task $\mathcal{T}_i$ attached to a reservation (B_i, T_i^s) is guaranteed to receive its reserved amount of time (i.e., Q_i time units over T_i^s).

Every reservation-based scheduler provides the *temporal isolation property*: the ability of a task to meet its deadline depends only on the reservation (B_i, T_i^s), and not on the presence of other tasks in the system. It can be easily proved [7] that, thanks to the isolation property, each task executes as it were on a dedicated processor of speed B_i.

For any reservation based system, we can define the *virtual finishing time* $\mathrm{VFT}_i(k) = \frac{c_i(k-1)}{B_i}$ of job $J_i(k-1)$ as the time it would finish in a dedicated processor of speed B_i. Intuitively, if $\mathrm{VFT}_i(k) > T_i$, then we need to allocate a larger reservation to task $\mathcal{T}_i$, (i.e. we need to speed-up the dedicated processor) in order to fulfil its requirements.

The quantity thus defined takes continuous values and it can be computed if the internal state of the scheduler is accessible (which is possible only for certain classes of scheduling algorithms and implementations). For other classes of systems it is useful to define the concept of *latest possible finishing time* for a job. The latest possible finishing time $\mathrm{LFT}_i(k)$ for job $J_i(k-1)$ is the end of the latest reservation period used by the job, minus the job arrival time: for example, if $J_i(k-1)$ has execution time $c_i(k-1) = 5$, it has been reserved a bandwidth $B_i = 0.5$, and the reservation period is $T_i^s = 4$, then it uses $\lceil \frac{5}{0.5 \cdot 4} \rceil = 3$ reservation periods, and its latest possible finishing time is 12, whereas its VFT is given by $5/0.5 = 10$. This quantity clearly carries a somewhat quantised information but it can easily be measured on any CPU reservation scheduler.

Dynamic Model of a Single Reservation. The reservation mechanism presented in the previous section can be modelled as a discrete time dynamic system. It is to stress that, due to the temporal isolation property, the model can be built for a single task regardless of other tasks present in the system. More precisely, the only coupling between the different dynamics is the constraint $\sum B_i \leq U_l$, which will be briefly discussed by the end of the section. Henceforth, we simplify the notation by removing the task index from all the quantities (e.g. we will use Q instead of Q_i and so for the other quantities). In this context we merely summarise results that are reported in detail in [3].

The evolution of the system is evaluated at the termination $f(k)$ of each job. We distinguish between two different situations: 1) the internal state of the scheduler is accessible, 2) the internal state of the scheduler is not accessible.

For the first case the most natural quantity that can be used as output variable is the scheduling error defined as the difference between VFT and relative deadline T: $\epsilon(k) = \mathrm{VFT}(k) - T$. If the scheduling error at the previous instance is less than 0, the virtual time can be easily calculated as: $\mathrm{VFT}(k) = \frac{c(k-1)}{B(k-1)}$. However, if the previous scheduling error is greater than 0, the virtual time at step k depends on the value of the previous virtual time: $\mathrm{VFT}(k) = V(k-1) - T + \frac{c(k-1)}{B(k-1)}$. By substituting, we can express the dynamic equation of the system as follows:

$$\epsilon(k+1) = \begin{cases} \epsilon(k) + \frac{c(k)}{B(k)} - T & \epsilon(k) \geq 0 \\ \frac{c(k)}{B(k)} - T & \epsilon(k) < 0 \end{cases} \quad (1)$$

When the internal state is not accessible, the scheduling error is more conveniently defined using the difference between the *latest possible finishing time* and the task period: $\epsilon(k) = \mathrm{LFT}(k) - T$. In this case, the scheduling error is a discrete variable and it is multiple of T^s. As shown in the cited paper, it is possible to write the dynamic of a conservative approximation $\tilde{\epsilon}(k)$ of the scheduling error as follows:

$$\tilde{\epsilon}(k+1) = \begin{cases} \tilde{\epsilon}(k) + \frac{c(k)}{B(k)} - T + q(k) & \tilde{\epsilon}(k) \geq T_s \\ \frac{c(k)}{B(k)} - T + q(k) & \tilde{\epsilon}_k < T_s \end{cases} \qquad (2)$$

where: $q(k) = \lceil \frac{c(k)}{B(k)T_s} \rceil - \frac{c(k)}{B(k)T_s}$. Equations (1) and (2) are similar, except for the switching point (which is 0 for Equation (1) and T^s for Equation (2)) and for the addition of a quantisation error term $q(k)$.

Feedback scheduling. It is now possible to formally define the problem of feedback control. The variable we want to control is the QoS provided by each task, which is represented by the $\epsilon(k)$ scheduling error. This variable, as we said, can be measured for each task upon the termination of a job and a feedback controller can adjust the bandwidth $B(k)$ reserved to the task. The purpose of the controller is to maintain this quantity below a specified level: $\epsilon(k) \leq E$. Moreover, if the computation time $c(k) = \overline{c}$ is constant or slowly varying, we require that the scheduling error be reduced exactly to zero and the bandwidth exactly to the value $\frac{\overline{c}}{T}$.

Considering the general situation of multiple tasks sharing the processor, a problem to cope with is that the termination of the jobs are asynchronous events. Therefore, designing a global controller for the entire task set is not a viable solution. However, the temporal isolation property allows us to adopt a decentralised strategy designing a separate controller for each task. The only global constraint is $\sum B_i \leq U_l$, which has to be enforced upon every control decision lest the correctness of the reservation algorithm be jeopardised. A possible way for handling this situation, which gave good results in real situations, is the adoption of a compression mechanism. Each decentralised controller can decide a bandwidth $B_i(k)$ constrained by $B_i(k) \leq B_0$. To achieve efficiency in handling the CPU, in the worst case the sum of all bandwidth is allowed to exceed the bound U_l: i.e. $nB_0 \geq U_l$. However, whenever a controller decides a bandwidth such that $\sum B_i$ would exceed the U_l bound, a compression mechanism comes into play that reduces the bandwidth within the bound: The compression function is given by:

$$B_i = \begin{cases} B_i & \text{if } \sum_i B_i \leq U_l \\ B_i w_i \frac{U_l}{\sum_j B_j w_j} & \text{otherwise} \end{cases}$$

where w_i are weights assigned to the different tasks. Effects of the compression mechanism can be considered as an additive disturbance ΔB_i on the bandwidths assigned to the tasks. It is not difficult to show that the maximum absolute value of ΔB_i is upper-bounded by $|\Delta B_i| \leq \frac{1 - w_i U_l}{W}$ where $W = \sum w_i$.

Summarising the discussion above, a control design methodology that applies to the case of *a single task with accessible internal state* dealing with possible additive noise is instrumental to the solution of the overall problem. A first contribution in this direction is offered in the next section. Constraints and requirements can be formally expressed as follows:

R.1 the maximum available bandwidth is upper bounded: $B(k) \leq B_0$; we'll find it convenient as command variable $u(k) = \frac{1}{B(k)}$ that has to be lower bounded by $u(k) \geq u_0 = \frac{1}{B_0}$.
R.2 the scheduling error $\epsilon(k)$ has to be upper bounded: $\epsilon(k) \leq E$
R.3 if the computation time is unknown but constant, it is required that $\epsilon(k) \to 0$ and the assigned bandwidth $B(k)$ tends to the equilibrium value $\frac{\bar{c}}{T}$. Moreover we require a guaranteed exponential rate of convergence, i.e.the existence of a real number $\gamma \in]0, 1[$ such that $|\epsilon(k)| \leq M\gamma^k|\epsilon_0|$ for some $M \in \mathbb{R}$.

3 Controller Design

In order to comply with requirement R.3 in case of constant computation time, we use a switching proportional integral (PI) policy:

$$u(k) = \begin{cases} u(k-1) + \alpha_1\epsilon(k) + \beta_1\epsilon(k-1) & \text{if } \epsilon(k) > 0 \\ u(k-1) + \alpha_2\epsilon(k) + \beta_2\epsilon(k-1) & \text{if } \epsilon(k) \leq 0 \end{cases} \tag{3}$$

The use of a two degrees of freedom controller for each mode might seem exaggerate for a first order system. However, as the synthesis procedure shown below is based on sufficient conditions, enlarging the design space enables an easier convergence.

The closed loop dynamics can be derived considering the two conditions $\epsilon_k \geq 0$ and $\epsilon_k \leq 0$. Let us introduce the state vector$x(k) = [x_1(k), x_2(k), x_3(k)]^T$ where $x_1(k) = \epsilon(k)$, $x_2(k) = \epsilon(k-1)$ and $x_3(k) = u(k-1) - \frac{T}{c(k-1)}$. The closed loop equations can be written as follows:

$$x(k+1) = \begin{cases} A_1(c(k))x + w(k) & \text{if } x_1(k) > 0 \\ A_2(c(k))x + w(k) & \text{if } x_1(k) \leq 0, \end{cases} \tag{4}$$

where

$$A_1(c(k)) = \begin{bmatrix} 1+\alpha_1 c(k) & c(k)\beta_1 & c(k) \\ 1 & 0 & 0 \\ \alpha_1 & \beta_1 & 1 \end{bmatrix},\ A_2(c(k)) = \begin{bmatrix} \alpha_2 c(k) & c(k)\beta_2 & c(k) \\ 1 & 0 & 0 \\ \alpha_2 & \beta_2 & 1 \end{bmatrix},$$
$$w(k) = \begin{bmatrix} 0 \\ 0 \\ \frac{T}{c(k-1)} - \frac{T}{c(k)} \end{bmatrix}.$$

The resulting system is piecewise affine and time varying. In the sequel, we will deal both with the problems of control analysis (i.e. is a given controller compliant with the requirements?) and with the control synthesis (i.e. find a controller able to stabilise the system). We will first provide results on the simpler case of constant computation times and then the more general case will be analysed.

3.1 Analysis

As a preliminary consideration, observe that requirements R.1 and R.2 are met if the system state evolves in the polyhedral region:

$$\mathcal{P} = \{x \text{ such that } x_1 \leq E \text{ and } x_3 \geq u_0 - \frac{T}{H}\} \tag{5}$$

In addition, for the case of constant $c(k)$, attractivity of the origin of the state space is required (requirements R.3). Thereby, the proposed analysis strategy is aimed at finding regions that are contained in $\mathcal{P}$ and that ensure [exponential] attractivity of the state space if $c(k)$ is constant. To this aim, it is useful to introduce the following definitions:

Definition 1. *Consider a time varying discrete time system* $x(k+1) = f_k(x(k)), x_k \in \mathbb{R}^n$ *and a set* $\mathcal{P} \subseteq \mathbb{R}^n$ *regarded as safe for the system's state. A set* $\mathcal{D}$ *is said:*

a $\mathcal{P}$-safe invariant set: *if* $\mathcal{D} \subseteq \mathcal{P}$ *and* $x(k) \in \mathcal{D} \forall k \in \mathbb{N},$ *and* $\forall x_0 \in \mathcal{D}$

a $\mathcal{P}$-safe exponential stability basin with decay rate γ: *for* $\gamma \in]0,1[$ *if* $\mathcal{D}$ *is a* $\mathcal{P}$*-safe invariant set and if there exist* $M \in \mathbb{R}^+$ *such that* $\forall k \in \mathbb{N},$ *and* $\forall x(0) \in \mathcal{D} itholds ||x(k)|| \leq M\gamma^k$.

Using this terminology we can define an analysis scheme in two steps. In the first step, we construct a set $\mathcal{D}$ that is $\mathcal{P}$-safe exponential stability basin in case of constant $c(k)$. This result has been inspired by the approach for dealing with control under input saturations shown in [8]. In the second one, we will provide conditions for $\mathcal{D}$ to be a $\mathcal{P}$-safe invariant set for variable $c(k)$.

Step 1: the case of constant $c(k)$ This step receives as inputs the gains $\alpha_1, \beta_1, \alpha_2, \beta_2$, the minimum and the maximum computation times h and H, the bounds E and u_0 for the scheduling error and for the control commands and the required exponential decay rate γ. The output is an ellipsoid $\mathcal{D}$ that is a $\mathcal{P}$-safe exponential stability basin.

The basic tool for this step is formally expressed by the following:

Theorem 1. *Assume that the following LMIs have a feasible solution:*

$$\begin{array}{ll} i) & \begin{bmatrix} \gamma^2 Q & (A_i(H)Q)^T \\ (A_i(h)Q) & Q \end{bmatrix} \succeq 0 \qquad i = 1,\ 2 \\ ii) & \begin{bmatrix} \gamma^2 Q & (A_i(h)Q)^T \\ (A_i(h)Q) & Q \end{bmatrix} \succeq 0 \qquad i = 1,\ 2 \\ iii) & Q_{1,\,1} \leq E^2, Q_{3,\,3} \leq (u_0 - \frac{T}{H})^2) \end{array} \tag{6}$$

then $\mathcal{E}(\overline{\alpha}^2 Q, 0)$*, with* $\overline{\alpha} = \min\{E, (u_0 - \frac{T}{H})\}$*, is a* $\mathcal{P}$*-safe exponential stability basin with respect to the set* $\mathcal{P}$ *in Equation (5) and with guaranteed decay rate* γ*.*

In order to prove the above, it is useful to recall some standard results. The following is derived from quadratic lyapunov stability theory:

Lemma 1. *Consider a discrete time linear and time varying system:*

$$x(k+1) = A(k)x(k), \tag{7}$$

where A_k *belongs to a set* $\mathcal{A}$ *for all* k *and assume that there exists a matrix* Q *such that:*

$$\begin{array}{l} Q^{-1} \succeq 0 \\ \gamma^2 Q^{-1} - A_s^T Q^{-1} A_s \preceq 0,\ \forall A_s \in \mathcal{A} \end{array} \tag{8}$$

then $\mathcal{E}(\alpha^2 Q, 0)$ *for any* $\alpha \in \mathbb{R}$ *is an exponential stability basin with a guaranteed decay rate* γ*.*

From easy considerations of convex analysis it is possible to prove the following:

Lemma 2. *Let $V(x) = x^T Q^{-1} x$ with $Q = Q^T \succeq 0$. The minimum of $V(x)$ on the hyperplane $cx = r$ is given by:*

$$\alpha_r^2 = \frac{r^2}{cQc^T}. \tag{9}$$

Proof (Theorem 1). For any instant k, 1) the system's state evolves according to one of the two closed loop dynamics $A_i(c(k))$, 2) matrices A_i depend affinely on $c(k)$, 3) $c(k)$ can be expressed as $\beta h + (1-\beta)H$ for some $\beta \in [0,\ 1]$. Thereby, $A_i(c(k)) = \beta A_i(h) + (1-\beta)A_i(H)$.

If conditions i), ii) hold in Equation (6) then their convex combination with coefficient β yields

$$\begin{bmatrix} \gamma^2 Q & (A_i(c(k))Q)^T \\ (A_i(c(k))Q) & Q \end{bmatrix} \succeq 0.$$

Using Schur complements lemma (see [4]) this condition can be written as:

$$\begin{aligned} & Q \succeq 0 \\ & \gamma^2 Q - (A_i(c(k))Q)^T Q^{-1} (A_i(c(k))Q) \geq 0. \end{aligned}$$

Pre-multiplying and post-multiplying both sides of the conditions by Q^{-1}, we get exactly the conditions of Lemma 1. In virtue of this result, $\mathcal{E}(\alpha^2 Q, 0)$ is an exponential stability basin. Clearly, $\alpha_1 \leq \alpha_2$ implies $\mathcal{E}(\alpha_1^2 Q, 0) \subseteq \mathcal{E}(\alpha_2^2 Q, 0)$. Therefore $\mathcal{E}(\alpha^2 Q, 0)$ is a subset of $\mathcal{P}$ if and only if lower than or equal to the minimum of $V(x) = x^T Q^{-1} x$ on $z_k = u_0 - \frac{T}{H}$ and $\epsilon_k = E$. Applying Lemma 2 this consideration is easily translated into condition iii), thus proving the claim.

Theorem 1 provides conditions for an ellipsoid to be an $\mathcal{P}-$safe exponential stability basin. Evidently, it is possible to direct the search toward ellipsoids that respect some notion of optimality. To this regard, very intuitive cost functions could be $\max \log det(P)$ or $\max trace(P)$. Both optimisation problems are convex. In the former case, the volume of the ellipsoid is maximised; in the latter maximisation is performed on the sum squared values of the semi-axes of the ellipsoid.

Step 2: accounting for variations of $c(k)$ Inputs to this step are: 1) the ellipsoid $\mathcal{D} = \mathcal{E}(Q, 0)$ produced at step 1, 2) the maximum and minimum computation times h and H, 3) the task's activation period T, 4) a real m such that $|\frac{T}{c(k)} - \frac{T}{c(k-1)}| \leq m, \forall k$. As an output of this step, if some sufficient conditions are met, we can conclude that $\mathcal{D}$ is an invariant set if the computation time varies as specified in the third input. Clearly, we cannot draw the opposite conclusion if the test fails.

It is interesting to briefly discuss the third input. If the computation time $c(k)$ is allowed to vary, the additive disturbance term $w(k)$ has to be accounted for in the system dynamics described by Equation (4). Both the matrices A_i of the switching systems and the additive disturbance w vary in time depending on the evolution of $c(k)$. However, we will not use this knowledge of the structure of $w(k)$. Namely, we will conduct a worst case study in which $w(k)$ is allowed to evolve freely respecting only the norm constraint

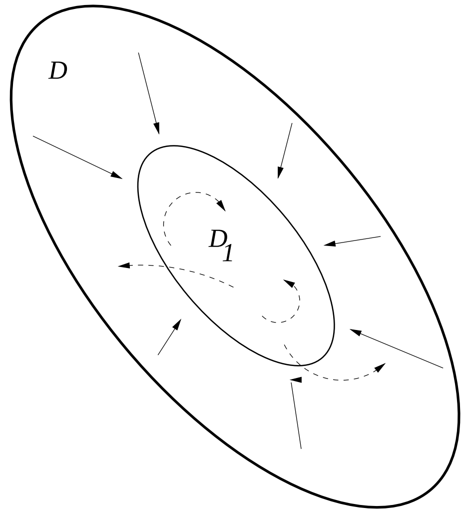

Fig. 1. The basic idea of the analysis for the invariance of $\mathcal{D}$: 1) $\mathcal{D}_1$ is attractive, 2) trajectories starting in $\mathcal{D}_1$ do not leave $\mathcal{D}$.

$|w(k)| \leq m$. This choice is conservative but it leads to a remarkable simplification in the analysis. A further advantage of this choice is that it provides a tool for handling also other additional noise terms as long as their maximum norm is bounded (and this is the case of the quantisation error for the case of non accessible internal state).

The following definition is useful for our purposes:

Definition 2. *Consider system (4) and two subsets $\mathcal{S}$ and $\mathcal{R}$ of the state space and let $\xi_{c(.),w(.)}(x_0, k)$ denote the state reached at step k starting from x_0 under the action of sequences $c(.)$ and $w(.)$.*

1. *$\mathcal{S}$ is attractive from a $\mathcal{R}$ if and only if for any vector $x_0 \in \mathcal{R}$ and for any sequence $\tilde{c}(.)$, $\tilde{w}(.)$ either there exists a step $\overline{k}$ such that $\xi_{\tilde{c}(.),\tilde{w}(.)}(x_0, \overline{k}) \in \mathcal{S}$ or $\lim_{k\to\infty}$ dist$(\xi_{\tilde{c}(.),\tilde{w}(.)}(x_0, k), \mathcal{S}) = 0$, where dist$(x, \mathcal{S})$ denotes the distance of x from the set $\mathcal{S}$.*
2. *$\mathcal{S}$ is n-steps invariant from $\mathcal{R}$ if $\mathcal{R} \subseteq \mathcal{S}$ and for any vector $x_0 \in \mathcal{R}$ and or any sequence $\tilde{c}(.)$, $\tilde{w}(.)$, it holds $\xi_{\tilde{c}(.),\tilde{w}(.)}(x_0, n) \in \mathcal{S}$.*

The idea used to decide whether ellipsoid $\mathcal{D}$ is invariant is very simple and it is illustrated in Figure 1. Consider a second ellipsoid $\mathcal{D}_1 = \mathcal{E}(r^2 Q, 0)$; clearly if $r \in]0, 1[$, then $\mathcal{D}_1$ is entirely contained in $\mathcal{D}$. We are able to conclude invariance of $\mathcal{D}$ if the following two conditions are satisfied: 1) $\mathcal{D}_1$ is attractive from $\mathcal{D} \backslash \mathcal{D}_1$, 2) $\mathcal{D}$ is 1-step invariant from $\mathcal{D}_1$. We deal separately with the two conditions.

As far as the fist condition is concerned, we can state the following:

Theorem 2. *Assume that there exists four reals $\tau_1(h), \tau_2(h), \tau_1(H), \tau_2(H)$ such that the following LMIs are satisfied:*

$$
\begin{array}{ll}
i) & \begin{bmatrix} Q^{-1} - A_i^T(h)Q^{-1}A_i(h) - \frac{\tau_i(h)}{r^2}Q^{-1} & -(Q^{-1}A_i(h)^T \\ -Q^{-1}A_i(h) & I\frac{\tau_i(h)}{m^2} - Q^{-1} \end{bmatrix} \succ 0 \quad i = 1,\ 2 \\
ii) & \begin{bmatrix} Q^{-1} - A_i^T(H)Q^{-1}A_i(H) - \frac{\tau_i(H)}{r^2}Q^{-1} & -(Q^{-1}A_i(H)^T \\ -Q^{-1}A_i(H) & I\frac{\tau_i(H)}{m^2} - Q^{-1} \end{bmatrix} \succ 0 \quad i = 1,\ 2 \\
ii) & \tau_i(h), \tau_i(H) > 0 \qquad i = 1,\ 2.
\end{array}
\tag{10}
$$

Then $\mathcal{D}_1$ is attractive from $\mathcal{D}\backslash\mathcal{D}_1$:

Proof. The claim is true if condition

$$x^T Q^{-1} x - (A_i(c(k))x + w)^T Q^{-1}(A_i(c(k))x + w) > 0, \tag{11}$$

holds whenever

$$x^T Q^{-1} x > r^2 \text{ and } w^T w \leq m^2. \tag{12}$$

Using the $\mathcal{S}$-procedure [18], the above implication is verified if there exist, for all $c(k)$ positive reals $\tau_{i,1}(c(k))$ and $\tau_{i,2}(c(k))$ such that $\forall x \in \mathbb{R}^3$ and $\forall w \in \mathbb{R}^3$:

$$
\begin{array}{l}
x^T Q^{-1} x - (A_i(c(k))x + w)^T Q^{-1}(A_i(c(k))x + w) + \\
-\tau_{i,1}(c(k))(x^T \frac{Q^{-1}}{r^2} x - 1) - \tau_{i,2}(c(k))(1 - \frac{w^T w}{m^2}) > 0
\end{array}
\tag{13}
$$

Since the above has to be verified also for $x = 0$ and $w = 0$, we derive the condition $\tau_{i,1}(c(k)) \geq \tau_{i,2}(c(k))$. Moreover, it is easy to see that if the inequality is verified for a $\tau_{i,1}^0(c(k)) > \tau_{i,2}(c(k))$ then it is verified also for $\tau_{i,1}(c(k)) = \tau_{i,2}(c(k))$. Hence, it is not restrictive to set $\tau_{i,1}(c(k)) = \tau_{i,2}(c(k)) = \tau_i(c(k))$. The resulting condition can be written as:

$$[x^T w^T] \begin{bmatrix} Q^{-1} - A_i^T(c(k))Q^{-1}A_i(c(k)) - \frac{\tau_i(c(k))}{r^2}Q^{-1} & -(Q^{-1}A_i(c(k))^T \\ -Q^{-1}A_i(c(k)) & I\frac{\tau_i(c(k))}{m^2} - Q^{-1} \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} > 0, \tag{14}$$

and it must hold $\forall x \in \mathbb{R}^3, \forall w \in \mathbb{R}^3$. For fixed r, the above is an infinite set of linear matrix inequalities (parametrised by $c(k)$). Following the same line of reasoning as in Theorem 1 it is possible to verify the LMI only at the four "vertexes" $A_i(h)$ and $A_i(H)$ and find by convex combination of $\tau_i(h)$ and $\tau_i(H)$ a a $\tau_i(c(k))$ that satisfies (14) for any $c(k)$.

It is easy to show the following corollary:

Corollary 1. *If condition (10) holds for a real $\overline{r}$ then it also holds for $r \geq \overline{r}$.*

Applying the above it is possible to find the minimum value r_1 in the interval $]0, 1[$ for which attractivity of $\mathcal{D}_1$ is ensured by a simple bi-section scheme.

The second condition can be enforced by the following:

Theorem 3. *Assume that there exists four reals* $\tau_1(h), \tau_2(h), \tau_1(H), \tau_2(H)$ *such that the following LMIs are satisfied:*

$$\begin{array}{ll} i) \begin{bmatrix} A_i^T(h)Q^{-1}A_i(h) + \frac{1-\tau_i(h)}{r^2} & -(Q^{-1}A_i(h)^T \\ -Q^{-1}A_i(h) & I\frac{\tau_i(h)}{m^2} - Q^{-1} \end{bmatrix} \succeq 0, & \\ ii) \begin{bmatrix} A_i^T(H)Q^{-1}A_i(H) + \frac{1-\tau_i(H)}{r^2} & -(Q^{-1}A_i(H)^T \\ -Q^{-1}A_i(H) & I\frac{\tau_i(H)}{m^2} - Q^{-1} \end{bmatrix} \succeq 0, & \\ ii)\tau_i(h), \tau_i(H) > 0 & i = 1,\ 2. \end{array} \tag{15}$$

Then for $\mathcal{D}$ *is* 1*-step invariant from* $\mathcal{D}_1$.

Proof. The claim is satisfied if

$$(A_i(c(k))x + w)^T Q^{-1}(A_i(c(k))x + w) \leq 1, \tag{16}$$

holds whenever

$$x^T Q^{-1} x \leq r^2 \text{ and } w^T w \leq m^2. \tag{17}$$

The proof is completed applying the $\mathcal{S}$-procedure and following the same arguments as in the proof of Theorem 2.

Also in this case, it is immediate to the following:

Corollary 2. *If condition (15) holds for a real* $\overline{r}$ *then it also holds for* $r \leq \overline{r}$.

A bisection scheme enables one to find, if possible, the maximum value r_2 in the interval for which the 1-step invariance of $\mathcal{D}$ from $\mathcal{D}_1$ is guaranteed.

Summing up, invariance of the set $\mathcal{D}$ can be concluded if $r_1 \leq r_2$ where r_1 and r_2 are found as shown above.

3.2 Synthesis

The analysis algorithms proposed in the previous section can be started if the designer comes up with a set of gains that, at least for constant $c(k) = c$, exponentially stabilise the system. The problem is essentially one of robust static output feedback. Indeed, matrices $A_i(c)$ can be written as:

$$A_i(c) = A_{i,0}(c) + b(c)[\alpha_i,\ \beta_i]C, \text{ where:}$$
$$A_{1,0}(c) = \begin{bmatrix} 1 & 0 & c \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ A_{2,0}(c) = \begin{bmatrix} 0 & 0 & c \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ b(c) = \begin{bmatrix} c \\ 0 \\ 1 \end{bmatrix},\ C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

The problem of finding a stabilising static output feedback is itself regarded as very hard (more precisely it is provably Np-hard if the range of possible values for the gains is limited). The synthesis can be performed requiring a common lyapunov function for the two different modes and for all possible values of the computation time c in the

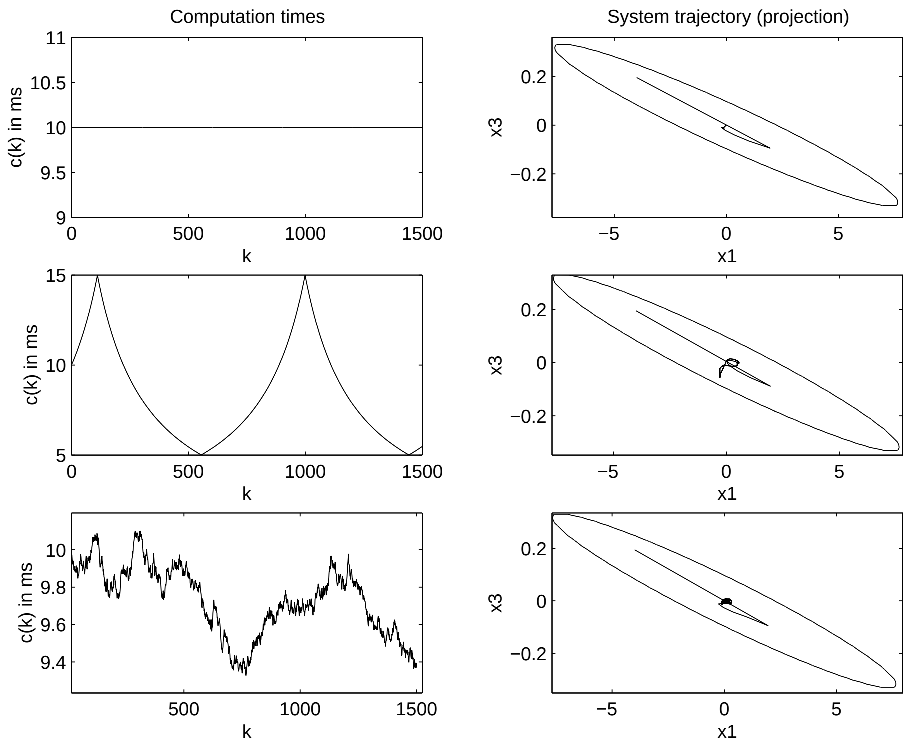

Fig. 2. Numerical example. First row: constant computation time. Second row, deterministically varying computation time. Third row, stochastically varying computation time.

admissible range $[h, H]$. Using Schur complements, a set of sufficient conditions for robust exponential stability (with decay rate γ) are as follows:

$$
\begin{array}{ll}
\text{i)} & \begin{bmatrix} \gamma^2 P & (A_{i,0}(h) + b(h)[\alpha_i,\ \beta_i]C)^T \\ A_{i,0}(h) + b(h)[\alpha_i,\ \beta_i]C & Q \end{bmatrix} \succeq 0 \quad i = 1,\ 2 \\
\text{ii)} & \begin{bmatrix} \gamma^2 P & (A_{i,0}(H) + b(H)[\alpha_i,\ \beta_i]C)^T \\ A_{i,0}(h) + b(H)[\alpha_i,\ \beta_i]C & Q \end{bmatrix} \succeq 0\ i = 1,\ 2 \\
\text{iii)} & PQ = I.
\end{array}
\qquad (18)
$$

Constraint iii) is the most troublesome, since it is nonlinear and non-convex. We attacked this problem using the cone complementarity approach, proposed in [6]. Although based on sufficient conditions, the method behaved acceptably well in our context. For the sake of brevity, we omit further details referring the reader to to the cited paper.

4 Numerical Examples

Extensive simulations were performed to validate the methodology. In this context we report an example consisting of a task with period $T = 20ms$, and computation time

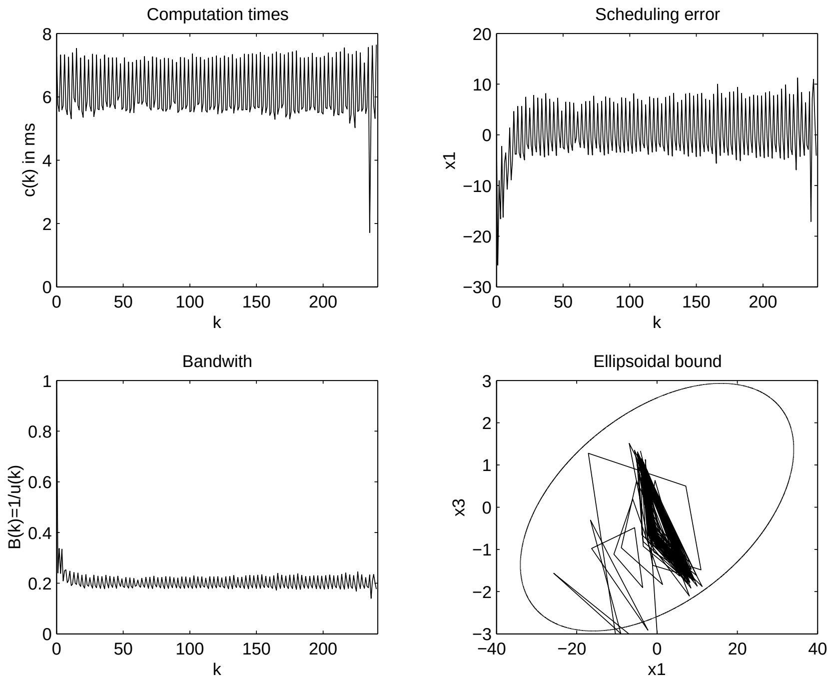

Fig. 3. Experiment on a real life application: MPEG decoding a trailer of Star Wars Episode 1.

varying in the range $[5ms, 15ms]$. The task is quite demanding (the 75% of the processor in the worst case).In order to give enough flexibility to the control action we put the saturation for the bandwidth at $B_0 = 0.92$. The synthesis procedure, requiring a guaranteed convergence rate $\gamma = 0.9$, produced $\alpha_1 = -0.057$, $\beta_1 = 0.044$, $\alpha_2 = -0.065$ and $\beta_2 = 0.0445$. The analysis was performed assuming $|\frac{T}{c(k)} - \frac{T}{c(k-1)}| \leq 0.006$ and it produced a $\mathcal{P}$-safe invariant set that is an exponential stability basin in case of constant $c(k)$. The projection of the ellipsoid onto the x_1-x_3 plane is reported, along with some simulation results in Figure 2. In particular in the first row, we show the results for constant computation time. As it is possible to see trajectories converge to the origin in exponential time. In the second row a simulation is reported for a deterministically varying computation time, where at each step $|\frac{T}{c(k)} - \frac{T}{c(k-1)}| = 0.006$. Finally, at the third row, the computation time was varied stochastically respecting the constraint $|\frac{T}{c(k)} - \frac{T}{c(k-1)}| \leq 0.006$ (random walk). As it is possible to see the safe ellipsoid contains, for both experiments the produced trajectories.

When the application presents sharp variations in computation time, the analysis proposed in Step 2 in the previous section need not produce acceptable results, being based on conservative conditions. This is the case of such real life application as MPEG players. As an example, we considered actual execution traces (profiled on an AMD Athlon 1.8 Ghz computer) for decoding a trailer of *Star Wars Episode 1*. Computation times for 250 samples are reported on the upper left plot contained in Figure 3. The maximum and minimum computation times are respectively, in the considered segment, $7.6ms$ and $1.71ms$; the activation period for standard MPEG decoding is $T = 33ms$. For the considered segment we recorded very strong variations in computation time: $\max \frac{|c(k)-c(k+1)|}{\max(c(k))-\min(c(k))} > 0.65$. We applied the synthesis procedure described above requiring $\gamma = 0.85$ and the produced gains were: $\alpha_1 = -0.0972$, $\alpha_2 = -0.1454$m $\beta_1 = \beta_2 = 0.0886$. By Step 1 of the analysis, we produced a $\mathcal{P}$-safe ellipsoid for constant computation time. Due to the strong variations in computation times, Step2 failed. However, as it is possible to see from Figure 3, the controller performs acceptably. Indeed, not only is the scheduling error contained within small bound, but also the assigned bandwidth does not change very much around 0.2 and the evolution in the $[e(k), u(k) - T/c(k)]$ plane lies almost always within the projection of the ellipsoid computed in Step 1.

5 Conclusions and future work

In this paper we considered the problem of allocating processor bandwidth to parallel software tasks hosted on a CPU so as to enforce compliance with soft real-time execution constraints. The adoption of resource reservation scheduling allows one to model the scheduler as a discrete-time switching system for which the assigned bandwidth is a control variable, the scheduling error is an output variable and the execution time of each job is a disturbance term. We showed analysis and synthesis techniques for a feedback controller suitable to this kind of application . The control scheme is entirely decentralised whereas occasional interactions between tasks due to the saturation of the total available bandwidth act as disturbance terms on the inputs. We showed the effectiveness of the proposed approach on a laboratory example and on data from a real application.

Many open problems are left for future research. Our first effort will be directed toward reducing conservativeness of the analysis technique for variable computation time, as shown in the MPEG player example above. Moreover, the problem of how to steer the system into the ellipsoidal safe set under the bandwidth saturation constraints is still open. Finally we want to develop mixed centralised/decentralised control to deal with possible interactions between tasks due to the saturation of the total bandwidth.

References

1. Luca Abeni and Giorgio Buttazzo. Integrating multimedia applications in hard real-time systems. In *Proceedings of the IEEE Real-Time Systems Symposium*, Madrid, Spain, December 1998.
2. Luca Abeni and Giorgio Buttazzo. Adaptive bandwidth reservation for multimedia computing. In *Proceedings of the IEEE Real Time Computing Systems and Applications*, Hong Kong, December 1999.
3. Luca Abeni, Luigi Palopoli, Giuseppe Lipari, and Jonathan Walpole. Analysis of a reservation-based feedback scheduler. In *Proc. of the Real-Time Systems Symposium*, Austin, Texas, November 2002.
4. S. Boyd and L.L. Vandenberghe. *Convex optimization*. Course lecture notes - Stanford University, 2002.

5. F. J. Corbato, M. Merwin-Dagget, and R. C. Daley. An experimental time-sharing system. In *Proceedings of the AFIPS Joint Computer Conference*, May 1962.
6. L. El Ghaoui, F. Oustry, and M. Ait Rami. A cone complementary linearization algorithm for static output-feedback and related problems. *IEEE Transaction and Automatic Control*, August 1997.
7. G.Lipari and S.K. Baruah. Greedy reclaimation of unused bandwidth in constant bandwidth servers. In *IEEE Proceedings of the 12th Euromicro Conference on Real-Time Systems*, Stokholm, Sweden, June 2000.
8. Haitham Hindi and Stephen Boyd. Analysis of linear systems with saturation using convex optimization. In *Proc. of the 37th IEEE Conference on decision adn Control (cdc 1998)*, Tampa, Florida, 1998.
9. B. Li and K. Nahrstedt. A control theoretical model for quality of service adaptations. In *Proceedings of Sixth International Workshop on Quality of Service*, 1998.
10. C. L. Liu and J. Layland. Scheduling alghorithms for multiprogramming in a hard real-time environment. *Journal of the ACM*, 20(1), 1973.
11. C. Lu, J. A. Stankovic, T. F. Abdelzaher, G. Tao, S. H. Son, and M. Marley. Performance specifications and metrics for adaptive real-time systems. In *Proceedings of the 21th IEEE Real-Time Systems Symposium*, Orlando, FL, December 2000.
12. Clifford W. Mercer, Stefan Savage, and Hideyuki Tokuda. Processor capacity reserves for multimedia operating systems. Technical Report CMU-CS-93-157, Carnegie Mellon University, Pittsburg, May 1993.
13. Tatsuo Nakajima. Resource reservation for adaptive qos mapping in real-time mach. In *Sixth International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS)*, April 1998.
14. Raj Rajkumar, Kanaka Juvva, Anastasio Molano, and Shuichi Oikawa. Resource kernels: A resource-centric approach to real-time and multimedia systems. In *Proceedings of the SPIE/ACM Conference on Multimedia Computing and Networking*, January 1998.
15. Dickson Reed and Robin Fairbairns (eds.). Nemesis, the kernel – overview, May 1997.
16. John Regehr and John A. Stankovic. Augmented CPU Reservations: Towards predictable execution on general-purpose operating systems. In *Proceedings of the IEEE Real-Time Technology and Applications Symposium (RTAS 2001)*, Taipei, Taiwan, May 2001.
17. David Steere, Ashvin Goel, Joshua Gruenberg, Dylan McNamee, Calton Pu, and Jonathan Walpole. A feedback-driven proportion allocator for real-rate scheduling. In *Proceedings of the Third usenix-osdi*. pub-usenix, feb 1999.
18. V. A. Yakubovich. *Vestnik Leningrad Uiversity*, chapter S-Procedure in nonlinear control theory, pages 66–77. 1971. (English translation in Vestnik Leningrad University 4:73-93,1977).