Maximizing the Stability Radius of a Set of Systems Under Real-Time Scheduling Constraints

Luigi Palopoli, Claudio Pinello, Antonio Bicchi, and Alberto Sangiovanni-Vincentelli

***Abstract*—We address the problem of synthesising real-time embedded controllers taking into account constraints deriving from the implementation platform, thus exploring the relation between the processor's time (or "attention") devoted to different control tasks and the overall robustness of the resulting design. Assuming a time-triggered model of computation for tasks controlling a set of independent systems and a real-time preemptive scheduling policy managing a single CPU processor board, we deal with two problems: 1) deciding whether a performance specification can be attained on a candidate platform, 2) optimising performance on a platform. The considered performance metric is the minimum stability radius attained over the different feedback loops.**

***Index Terms*—Control implementation, control with computation and communication constraints, real-time scheduling, robust control.**

I. Introduction

This note deals with the problem of designing concurrently both control laws and control implementation parameters. The control laws are implemented as software tasks running on a single processor board. In particular, we address the case when multiple independent control tasks share the same CPU (this situation is frequent, for example, in automotive applications) and each control task stabilises a linear time invariant system. When multiple tasks share computation and communication resources, an algorithm is required to schedule them. From the control engineering perspective, the presence of a processor in the loop and of other tasks poses restrictions on the sampling periods and introduces both computation and scheduling delays.

We can compute schedules either offline or on-line (i.e., while the tasks execute). A good example of the former approach is offered by time division multiplexing allocation (TDMA), in which a time quantum is periodically assigned to each task for execution/communication. The problem of scheduling/control codesign, when a TDMA scheduler is used, has been coped with in several works [6], [12], [21], [15], [20], [13].

Most real-time operating systems perform scheduling decisions online based on parameters (e.g., priorities) assigned to each task and allow suspension (preemption) of a task by a higher priority task. For real-time control applications, compliance with the set of timing constraints has to be the main concern [8] of the scheduling algorithms. Representative examples of such scheduling algorithms are rate monotonic (RM) and earliest deadline first (EDF) [16].

Manuscript received February 13, 2003; revised May 18, 2004 and May 3, 2005. Recommended by Associate Editor A. Bemporad. This work was supported by the European Commission under Contract IST2001-37170, by the Italian Research Center under Contract C00E714-001, by the Italian Ministry for Education University and Research (MIUR contract), by the National Science Foundation Chess project under Award 0225610, and by the GSRC DOD Advanced Research Projects Agency (DARPA) under Grant MDA972-02-1-0003.

L. Palopoli is with the Department of Information and Communication Technology, University of Trento, 14 38050 POVO (Trento), Italy (e-mail: palopoli@dit.unitn.it).

C. Pinello and A. Sangiovanni-Vincentelli are with the Department of Electrical Engineering and Computer Science, the University of California, Berkeley, CA 94720 USA (e-mail: pinello@eecs.berkeley.edu; alberto@eecs.berkeley.edu).

A. Bicchi is with the Centro Piaggio, Facoltá di Ingegneria, 56126 Pisa, Italy (e-mail: bicchi@ing.unipi.it).

Digital Object Identifier 10.1109/TAC.2005.858639

Preemptive schedulers may exhibit *scheduling jitter*. Stochastic execution times for the tasks may lead to stochastic scheduler-induced delays that are very difficult to model. This issue hinders the applicability of design techniques such as the one proposed in [17] for networked systems, where independent and Markov delays are assumed, and of the Lyapunov-based techniques in [26].

In this work, we aim at formulating the design problem as an optimization problem where scheduling parameters are the decision variables and the performance metric is related to the control performance. The control scheme is time-driven, and this assumption marks a remarkable difference from other works (e.g., [27]), where the authors assume an event driven strategy. More similar to our approach is the work reported in [22], where the authors considered a set of independent loops controlled by real-time periodic tasks and assessed the system's performance by a weighted sum of quadratic cost functions. However, they *assume* an exponential structure for the cost function with few control theoretical arguments supporting this assumption. The same model was adopted in [9], where an analytical optimization procedure can potentially be set up for systems having convex cost functions. We believe that quadratic cost functions, although an interesting metric, do not unambiguously capture the fundamental issue of controlling a system with limited information: the technological feasibility of the controller.

We introduce appropriate assumptions on the implementation, so that, we can derive an abstract, yet fully significant description of the ensemble Hardware + RTOS that we will refer to as *control platform*. This approach is inspired by the recent literature on *platform based design* for embedded systems [14]. The real-time scheduling theory allows us to condense platform performance constraints into a simple linear inequality. Moreover, the use of a time-triggered model of computation on top of the scheduler [23] nullifies the scheduling jitter—which is the main concern of other works such as [10].

The formalization of the control platform allows us to tackle two fundamental questions.

- Problem P.1) Can a candidate control platform sustain a desired performance specification?
- Problem P.2) If it qualifies, how should its potential be exploited to maximize performance?

We want to explore the tradeoff between the processor's time devoted to a control task—which can be related to the "attention" functional introduced in [7] – and the size of the family of stabilising controllers. This size can be quantified, for example, by the Chebychev radius of the region of stabilising control parameters (e.g., the feedback gains). For an open-loop unstable system, it is reasonable that such a region vanishes when the attention devoted to the task is too small. This intuition is confirmed by Fig. 1, which shows this region for a scalar system at different values of the control task activation period: For an open-loop stable system the region reduces to an asymptotic triangle for increasing values of the period, while it vanishes if an open loop unstable system is considered. Clearly, a large family of stabilising gains favors the technological feasibility of a controller. Moreover, the Chebychev radius is a generalization of the classical gain margin and it quantifies, for example, robustness with respect to fixed point approximations of the gains.

The note is organized as follows. In Section II, we characterize our abstraction of the control platform. In Section III, we formulate the design problem. In Section IV, we assess the stability radius of the closed loop system. In Section V, we formulate the optimization problem and its solution. Finally, we offer some computational results and in Section VI, we give conclusions and present our plan for extending this approach to a larger class of systems.

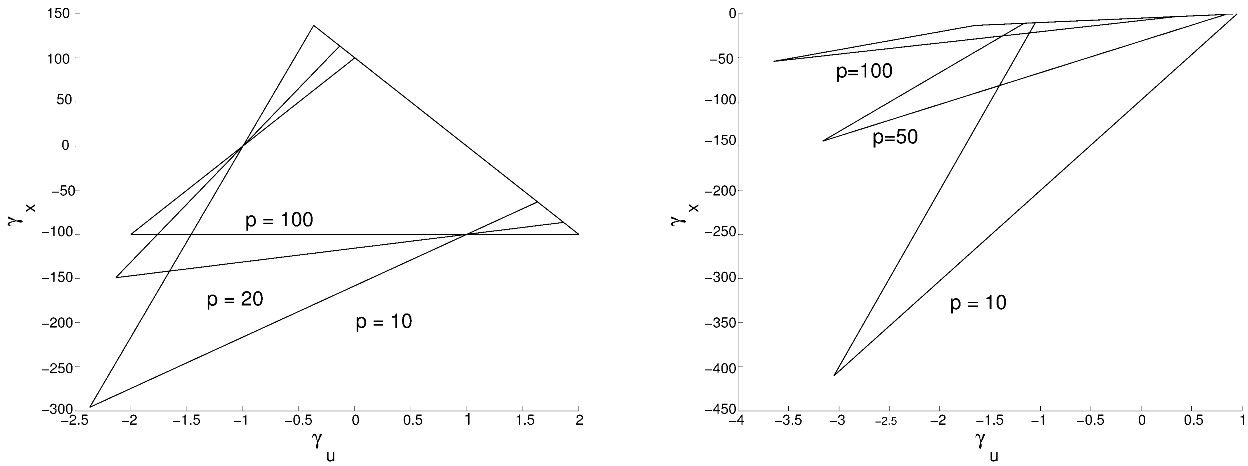

Fig. 1. Triangular regions describe sets of stabilising gains for a 2-dof controller at different activation periods for the task, as applied to an open-loop stable system (left) or to an open-loop unstable system (right). The output of the controller is given by $u(k) = \gamma_u\, u(k-1) + \gamma_x\, x(k)$, where $x(k)$ is the state of the plant.

II. Control Platform

In this section, we provide a formal characterization of an abstraction layer (called control platform) that we will use to gauge the effect of the hardware/software platform on control performance. We assume a set of computing tasks $\tau^{(i)}$. Each task has an input space, an output space and may have an internal state. Every time a new input is read, the task updates its internal state (if any) and produces its output. This sequence of actions will be called a *job*. Once an output is released, its value is sustained until the next reading (ZOH model).

A. Model of Computation

The time-triggered model of computation is based on the adoption of a common clock, having period $T_c \in \mathbb{R}$, as the basis for the activation of all tasks hosted on the CPU. Interactions between the computer and the environment take place at time instants hT_c with $h \in \mathbb{N}$. Task $\tau^{(i)}$ is associated to a temporal interface consisting of two sequences $I^{(i)}(h)$ and $O^{(i)}(h)$ with $O^{(i)}(h) > I^{(i)}(h)$. At instant $I^{(i)}(h)T_c$ inputs are sampled and execution of the hth job of $\tau^{(i)}$ started. Outputs of the h-th job are released at instant $O^{(i)}(h)T_c$. We assume $p^{(i)}$-periodic sequences for task $\tau^{(i)}$: $I^{(i)}(h) = hp^{(i)}$ and $O^{(i)}(h) = (h+1)p^{(i)}$ with $p^{(i)} \in \mathbb{N}$. The implementation of this model requires some technological support to ensure that interactions with the environment take place at the specified instants. It is possible to use specialized I/O hardware (release outputs and acquire inputs at programmed instants) or we can configure appropriately the real-time operating system (e.g., giving high priority to the interrupt handlers used for input/output of each task).

B. Real-Time Schedulability

When a class of hardware implementations, based on a single CPU, is considered for the platform, each task $\tau^{(i)}$ is associated to the worst case execution time $e^{(i)} \in \mathbb{R}$, expressed relative to the clock period, i.e., $w.c.e.t.(\tau^{(i)}) = e^{(i)} \cdot T_c$. In order for the task $\tau^{(i)}$ to honor its timing interface, the execution of its hth job must be completed by time $O^{(i)}(h)T_c$. A scheduling policy guarantees *schedulability* of a task set if all deadlines are respected. If there exists such a scheduling policy, the task set is said *schedulable*. For periodically activated tasks, the ratio $e^{(i)}/p^{(i)}$ gives the processor utilization per unit time. A necessary condition for schedulability is $\sum_{i=1}^{m}(e^{(i)}/p^{(i)}) \leq 1$. RM or EDF schedules have important properties [16].

Theorem 1: A sufficient schedulability condition for a set of m periodic tasks $\tau^{(i)}$, is

$$\sum_{i=1}^{m} \frac{e^{(i)}}{p^{(i)}} \leq U_l \tag{1}$$

where, $U_l = 1$ for EDF and $U_l = m(2^{1/m} - 1)(> 0.69)$ for RM.

C. Platform Definition

Given the choice of the time triggered MoC and of a class of real time schedulers, the implementation platform can be parameterized by

$$U_l, T_c, e^{(1)}, \ldots, e^{(m)}. \tag{2}$$

The control platform exposes periods $p^{(i)}$ as free design parameters. Acceptable choices for the free parameters are those respecting condition (1).

III. Problem Definition

A. The Control Problem

Consider a collection $\mathcal{S}$ of single input completely reachable systems $\mathcal{S}^{(i)}$ described by equations

$$\dot{x}^{(i)} = A^{(i)}x^{(i)} + b^{(i)}u^{(i)} \tag{3}$$

where $A^{(i)} \in \mathbb{R}^{n_i \times n_i}$, $b^{(i)} \in \mathbb{R}^{n_i}$ and $i = 1, \ldots, m$. A periodic task $\tau^{(i)}$ is used to control system $\mathcal{S}^{(i)}$ (words "controller" and "task" will be used interchangeably). The task samples the state variables every $p^{(i)}T_c$ time units. For notational simplicity, we will drop the (i) superscript, when it is not strictly needed.

Considering the ZoH and the input–output delay of pT_c due to the time-triggered MoC, we have $u(t) = u(hpT_c) = \tilde{u}(h-1), \forall t \in [hpT_c, (h+1)pT_c[$. Hence, we study the system considering the discrete-time sequence of samples at $t = hpT_c$: $\tilde{x}(h) = x(hpT_c)$. The one-period delay can be modeled introducing an additional state variable $\tilde{z}$

$$\begin{bmatrix} \tilde{x}(h+1) \\ \tilde{z}(h+1) \end{bmatrix} = \tilde{A} \begin{bmatrix} \tilde{x}(h) \\ \tilde{z}(h) \end{bmatrix} + \tilde{b}\tilde{u}(h)$$
$$\text{where } \tilde{A} = \begin{bmatrix} e^{A\,p\,T_c} & (\int_0^{pT_c} e^{As} ds)b \\ 0 & 0 \end{bmatrix}, \tilde{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{4}$$

We use a dynamic feedback scheme keeping memory of the past value used for control

$$\tilde{u}(h) = \gamma^{(x)}\tilde{x}(h) + \gamma^{(u)}\tilde{z}(h). \quad (5)$$

The closed-loop dynamics for the subsequence $\tilde{x}(h)$ is given by

$$\begin{bmatrix} \tilde{x}(h+1) \\ \tilde{z}(h+1) \end{bmatrix} = (\tilde{A} + \tilde{b}\gamma) \begin{bmatrix} \tilde{x}(h) \\ \tilde{z}(h) \end{bmatrix}, \text{ where } \gamma = [\gamma^{(x)}\ \gamma^{(u)}]. \quad (6)$$

B. Performance Metric and Problem Formulation

Based on the assumed control scheme, each controller $\tau^{(i)}$ has a vector of free parameters $\gamma^{(i)} \in \mathbb{R}^{d_i}$ (e.g., feedback gains). The stability radius is defined as follows.

Definition 1—Stability Center and Stability Radius: Let $\Gamma^{(i)} \subseteq \mathbb{R}^{d_i}$ be a set such that system $\mathcal{S}^{(i)}$ is asymptotically stabilised by controller $\tau^{(i)}$ if and only if $\gamma^{(i)} \in \Gamma^{(i)}$.

1) The stability center $\gamma_c^{(i)}$ and the stability radius $\mu^{(i)}$ for system $\mathcal{S}^{(i)}$ are respectively the Chebychev center and the Chebychev radius of $\Gamma^{(i)}$.
2) The stability radius $\mu^{(\mathcal{S})}$ of the collection $\mathcal{S}$ is defined as $\min_{i=1,\ldots,m} \mu^{(i)}$.

We recall that the Chebychev centre and radius of a set are, respectively, the center and the radius of the maximum-norm ball entirely contained in the set. In this note, we use the ∞-norm. Extensions are straightforward for the 1-norm and are possible, in most cases, for the 2-norm.

Remark 1: The stability radius μ is a generalization for single-input–multiple-output (SIMO) systems of the classic notion of gain margin and it quantifies robustness with respect to a variety of technological limitations of the implementation (e.g., actuator imprecision, truncation errors etc.). Indeed a very small region $\Gamma^{(i)}$ might require a design accuracy impossible to achieve. However, the stability radius does not capture *per se* important aspects of robustness, such as those deriving form uncertainties on the A matrix. A possible way to address these uncertainties is to consider the *robust* stability radius defined as follows. Let $\omega^{(i)}$ be any set of unknown plant parameters varying in a set $\Omega^{(i)}$ and let $\Gamma^{(i)}(\omega^{(i)})$ be the set of stabilising $\gamma^{(i)}$ parameters as in Definition 1. The *robust* stability centre and radius are respectively the Chebychev centre and radius of $\Gamma^{(i)}(\Omega^{(i)}) = \cap_{\omega^{(i)}\in\Omega^{(i)}}\Gamma^{(i)}(\omega^{(i)})$. Techniques for computing the robust stability radius can be derived from the ones shown below for the stability radius (using the robust counterpart of linear and semidefinite programming [4]) or randomised algorithms. However, we will not report them here for the sake of brevity.

The μ metric allows for a mathematical definition of **Problem P.1** and **Problem P.2**, presented in the introduction, as optimization problems on the integer periods $p^{(i)}$ as decision variables. Both problems are captured by the following definition:

$$\left\{\max_{\mu,\, p\in\mathbb{N}} J \middle| \sum_i \frac{e^{(i)}}{p^{(i)}} \leq U_l \wedge \mu^{(i)} \geq \mu \wedge \mu \geq \mu_0 .\right\} \quad \text{(P.1)}$$

where the cost function J is 0 if we want a feasible solution with $\mu \geq \mu_0$ (**Problem P.1**), and J is μ if we want to optimize the stability radius (**Problem P.2**).

Remark 2: The results presented below can be easily generalized considering a weighted version for the stability radius $\mu'^{(i)} = w^{(i)}\mu^{(i)}$. To avoid undesirable bias caused by strong authority actuators on some of the systems, it is possible to choose $w^{(i)} = |b^{(i)}|$. More generally, it can be useful to relate the weights to the controllability of the system.

IV. Computing the Stability Radius

The focus of this section is on the machinery for computing the stability radius of the collection. To this end, we need a mathematical description of the regions $\Gamma^{(i)}$. In Section IV-A, an exact closed form is derived for first order systems, while multidimensional systems are dealt with in Section IV-B.

Standard arguments show that exponential stability of the linear systems (3) under the action of the ZoH controllers (5) can be achieved if and only if exponential stability of systems (6) can. As controllers have access to the entire state of the augmented system (inclusive of $\tilde{x}(h)$ and $\tilde{z}(h)$), the latter problem has feasible solutions whenever the $\tilde{A}, \tilde{b}$ pair is reachable. Starting from reachable continuous time systems, reachability losses can occur only for isolated values of pT_c [1] and will not be considered in our discussion.[1] Moreover, the region Γ is given by vectors γ for which $\tilde{A} + \tilde{b}\gamma$ matrix is Schur-stable. Indeed, the coefficients of the characteristic polynomial are affine functions of the gains. Thereby, Γ can be computed *via* the Jury criterion resulting, in the general case, into an intersection of nonlinear polynomial inequalities.

A. First-Order Systems

For first order systems, an analytical computation of the stability radius is viable since the Jury criterion provides a set of linear inequalities. As a result, region Γ is a polyhedron.

The characteristic polynomial of the $\tilde{A} + \tilde{b}\gamma$ matrix is given by:

$$z^2 - (e^{\lambda_1 pT_c} + \gamma^{(u)})z + e^{\lambda_1 pT_c}\gamma^{(u)} - \left(\int_0^{pT_c} e^{\lambda_1 s} ds\right) b\gamma^{(x)}$$

where λ_1 is the eigenvalue of the continuous-time system. The Jury criterion yields

$$\begin{bmatrix} e^{\lambda_1 pT_c} & -(\int_0^{pT_c} e^{\lambda_1 s} ds)b \\ -(e^{\lambda_1 pT_c} - 1) & (\int_0^{pT_c} e^{\lambda_1 s} ds)b \\ -(e^{\lambda_1 pT_c} + 1) & +(\int_0^{pT_c} e^{\lambda_1 s} ds)b \end{bmatrix} \begin{bmatrix} \gamma^{(x)} \\ \gamma^{(u)} \end{bmatrix} < \begin{bmatrix} 1 \\ 1 - e^{\lambda_1 pT_c} \\ 1 + e^{\lambda_1 pT_c} \end{bmatrix}. \quad (7)$$

The resulting Γ is a triangle that, when p tends to infinity, collapses onto the point of coordinates $[\gamma^{(x)}, \gamma^{(u)}] = [-(e^{\lambda_1 pT_c}\lambda_1/b), -e^{\lambda_1 pT_c}]$ if $\lambda_1 \geq 0$, and tends to an asymptotic triangle if $\lambda_1 < 0$ (see Fig. 1). The expression for the stability radius is contained in the following.

Proposition 1: The stability radius of the system is given by

$$\mu = \begin{cases} \frac{\lambda_1}{e^{\lambda_1 pT_c}(\lambda_1 + |b|) - |b|}, & \text{if } \lambda_1 > 0 \\ \frac{2\lambda_1}{e^{\lambda_1 pT_c}(\lambda_1 + 2|b|) + \lambda_1 - 2|b|}, & \text{if } \lambda_1 < 0 \\ \frac{1}{1 + pT_c|b|}, & \text{if } \lambda_1 = 0 \end{cases}. \quad (8)$$

To prove the proposition, we will use this well-known result [5].

Lemma 1: Let $\mathcal{P}$ be a polyhedral set defined by a set of linear inequalities

$$h_i u \leq f_i, i = 1, \ldots, c.$$

The Chebychev centre of $\mathcal{P}$ in the norm $\|.\|$ is the solution to the following linear program:

$$\max\{\mu | h_i u + \mu\ \|h_i\|_* \leq f_i, \text{ for } i = 1, \ldots, c \wedge \mu \geq 0\}. \quad (9)$$

Proof: [Proposition 1]: The proof is given for the case $\lambda > 0$ (other cases follow similar arguments). With regard to the ∞ norm

[1]Should this problem occur, a different choice of T_c, or of p, would preserve reachability.

(whose dual is the 1-norm), the application of Lemma 1 to the set defined in (7) yields the following linear program:

$$\left\{\max_{\gamma^{(u)},\gamma^{(x)},\mu} \mu | H \begin{bmatrix}\gamma^{(u)}\\ \gamma^{(x)}\\ \mu\end{bmatrix} \le q, \mu \ge 0\right\}, \text{ where}$$

$$H=\begin{bmatrix} e^{\lambda_1 pT_c} & -\frac{e^{\lambda_1 pT_c}-1}{\lambda_1}b & e^{\lambda_1 pT_c}+\frac{e^{\lambda_1 pT_c}-1}{\lambda_1}|b| \\ 1-e^{\lambda_1 pT_c} & \frac{e^{\lambda_1 pT_c}-1}{\lambda_1}b & e^{\lambda_1 pT_c}-1+\frac{e^{\lambda_1 pT_c}-1}{\lambda_1}|b| \\ -1-e^{\lambda_1 pT_c} & \frac{e^{\lambda_1 pT_c}-1}{\lambda_1}b & e^{\lambda_1 pT_c}+1+\frac{e^{\lambda_1 pT_c}-1}{\lambda_1}|b| \end{bmatrix}$$

$$\text{and } q=\begin{bmatrix}1\\ 1-e^{\lambda_1 pT_c}\\ 1+e^{\lambda_1 pT_c}\end{bmatrix}. \tag{P.2}$$

The dual problem is given by

$$\left\{\min_y [q^T \ 0]y | \begin{bmatrix} H^T & \begin{matrix}0\\0\\-1\end{matrix}\end{bmatrix} y = \begin{bmatrix}0\\0\\1\end{bmatrix}, y \ge 0\right\}.$$

The solution $H^{-1}q$ is primal feasible. The complementary slackness solution

$$\begin{bmatrix} H^{-1^T}\begin{bmatrix}0\\0\\1\end{bmatrix} \\ 0\end{bmatrix}$$

is dual feasible. Hence, the two solutions are an optimal pair and the proof is then completed. ■

This result confirms the qualitative intuition that the stability radius is, for open-loop unstable systems, a decreasing function of p that tends to 0 as p tends to infinity.

B. Results for Multidimensional Systems

Even though in the general case it is not possible to easily compute the stability radius, it is possible to find lower and upper bounds that can be used in the design problem. With this respect, the use of a polyhedral norm in the definition of the stability radius turns out to be particularly useful, since it enables efficient numerical procedures for its computation.

An external polyhedral approximation of the stability region will be used to compute an upper bound in closed form. Lower bounds can be found numerically. We will show a technique based on the concept of quadratic stability and on the subsequent use of convex optimization.

1) An Upper Bound: An upper bound for the stability radius can be found using the following.

Lemma 2: A necessary condition for a matrix X to be Schur-stable is that $|det(X)| < 1$.

The computation of the determinant shown previously yields an affine expression in the gains

$$det(\tilde{A}+\tilde{b}\gamma) = \gamma^{(x)}q(p) + \gamma^{(u)}e^{\mathrm{tr}(A)pT_c} \tag{10}$$

where the expression for vector $q(p)$ is easy to find, as a function of p, using symbolic computation tools. Now, it is possible to state the following.

Proposition 2: An upper bound for the stability radius is given by

$$\overline{\mu} = \frac{1}{\|q(p)\|_1 + e^{\mathrm{tr}(A)pT_c}}. \tag{11}$$

Proof: The proof is not reported for the sake of brevity. It follows the same line of reasoning as the proof of Proposition 1 and it can be found in [19]. ■

As an immediate consequence of the above, it is possible to state the following.

Corollary 1: An upper bound of the stability radius is given by

$$\frac{1}{e^{\mathrm{tr}(A)pT_c}}. \tag{12}$$

Interestingly, when $\mathrm{tr}(A) > 0$ the bound is decreasing with exponential rate with respect to p.

2) Computing a Lower Bound: There are different techniques for computing lower bounds of the stability radius. In this note, we will show an approach based on the concept of *quadratic* stability, which is a good compromise between efficiency and accuracy. The system $x(k+1) = X(\theta)x(k)$, where θ is a vector of unknown parameters belonging to a set $\Theta \subseteq \mathbb{R}^l$, is said *quadratically stable* if there exists a matrix $P \succ 0$, such that $\forall \theta \in \Theta \ X(\theta)^T P X(\theta) - P \preceq 0$. Quadratic stability can be tested using the following lemma [2]:

Lemma 3: Let $X(\theta)$ be an affine function of θ and let the Θ set be a polytope. Then the system is quadratically stable if the Lyapunov condition $X(\theta)^T P X(\theta) - P \preceq 0$ is respected at the vertexes of Θ.

Using the above result, a lower bound of the stability radius that can be tested imposing the following set of Lyapunov conditions:

$$\begin{aligned} &\max_{\mu,\gamma^{(c)},P} \mu \text{ subj. to } \mu \ge 0 \wedge P \succ 0 \wedge \\ &\wedge (\tilde{A}+\tilde{b}(\gamma^{(c)}+\mu v_i))^T P (\tilde{A}+\tilde{b}(\gamma^{(c)}+\mu v_i)) - P \preceq 0 \\ &\text{for } i = 1, 2, \ldots \end{aligned} \tag{P.3}$$

where row vectors v_i denote the vertexes of the norm ball: $B_1(0) = \{\gamma \mid \|\gamma\|_\infty \le 1\}$. Premultiplying and postmultiplying each Lyapunov condition by $W = P^{-1}$ and applying Schur complements the problem can easily be reformulated as

$$\begin{aligned} &\max_{\mu,W,Y} \mu \\ &\mu \ge 0 \wedge W \succ 0 \\ &\begin{bmatrix} -W & (\tilde{A}W+\tilde{b}(Y+\mu v_i W))^T \\ (\tilde{A}W+\tilde{b}(Y+\mu v_i W)) & -W \end{bmatrix} \\ &\preceq 0 \ i = 1, \ldots \end{aligned} \tag{P.4}$$

where the $\gamma^{(c)}$ variable is "absorbed" into $Y = \gamma^{(c)}W$. Observe that, for fixed μ, the above problem reduces to a LMI based feasibility test. Hence, it is possible to apply a bisection scheme over μ to find a lower bound for the stability radius.[2] This algorithm is applicable also to more general classes of systems, such as piecewise linear systems.

3) Validation by Randomised Algorithms: In this section, we assess the quality of the bounds proposed in Sections IV-B.I and B.2 using randomized algorithms (R.A.) and statistical learning theory. Let $P(\gamma^{(c)}, \mu)$ denote the probability of getting a Schur-stable matrix in the set $\tilde{A} + \tilde{b}(\gamma^{(c)} + \gamma)$, where γ is a random vector varying in $B_\mu(0) = \{\gamma \ s.t. \|\gamma\|_\infty \le \mu\}$ (We can assume a uniform distribution of μ inside $B_\mu(0)$). For a given μ, we search for the centre ${\gamma^{(c)}}^*(\mu)$ that maximizes $P(\gamma^{(c)}, \mu)$. The optimal probability $P({\gamma^{(c)}}^*(\mu), \mu)$ is a non increasing function of μ. Clearly, for the lower bound $\underline{\mu}$ it holds $P({\gamma^{(c)}}^*(\underline{\mu}), \underline{\mu}) = 1$, while for the upper bound $\overline{\mu}$ it holds $P({\gamma^{(c)}}^*(\overline{\mu}), \overline{\mu}) \le 1$. The actual stability radius can be estimated as the minimum μ^+ such that $P({\gamma^{(c)}}^*(\mu^+), \mu^+) \le 1 - \xi$, where $\xi > 0$ is some numeric tolerance.

We use empirical probabilities to evaluate $P(\gamma^{(c)}, \mu)$ for a given $\gamma^{(c)}$ and μ. That is, we draw N samples of γ in $B_\mu(0)$ and estimate the empirical probability $P_N(\gamma^{(c)}, \mu)$ by counting the number of stable systems in the sample set. Then, we need to maximize $P_N(\gamma^{(c)}, \mu)$, with respect to $\gamma^{(c)}$. Given a candidate region $\hat{\Gamma} \supset \Gamma$, we draw M random samples $\gamma_1^{(c)}, \ldots, \gamma_M^{(c)}$, and return the sample that maximizes the empirical probability: $\gamma_{M,N}^{(c)} = arg\max_{i=1,\ldots,M} P_N(\gamma_i^{(c)}, \mu)$. The problem of selecting M and N to get an approximation of acceptable quality is addressed in [25] and [24]. In those papers, the authors compute—for a given triple of real numbers $(\epsilon_1, \epsilon_2, \delta)$ – two numbers

[2]A more efficient solution can be obtained formulating the problem as a Generalized Eigenvalue minimization problem (GEVP). Such formulation can exploit the quasiconvexity of the problem to take advantage of specialized algorithms [3].

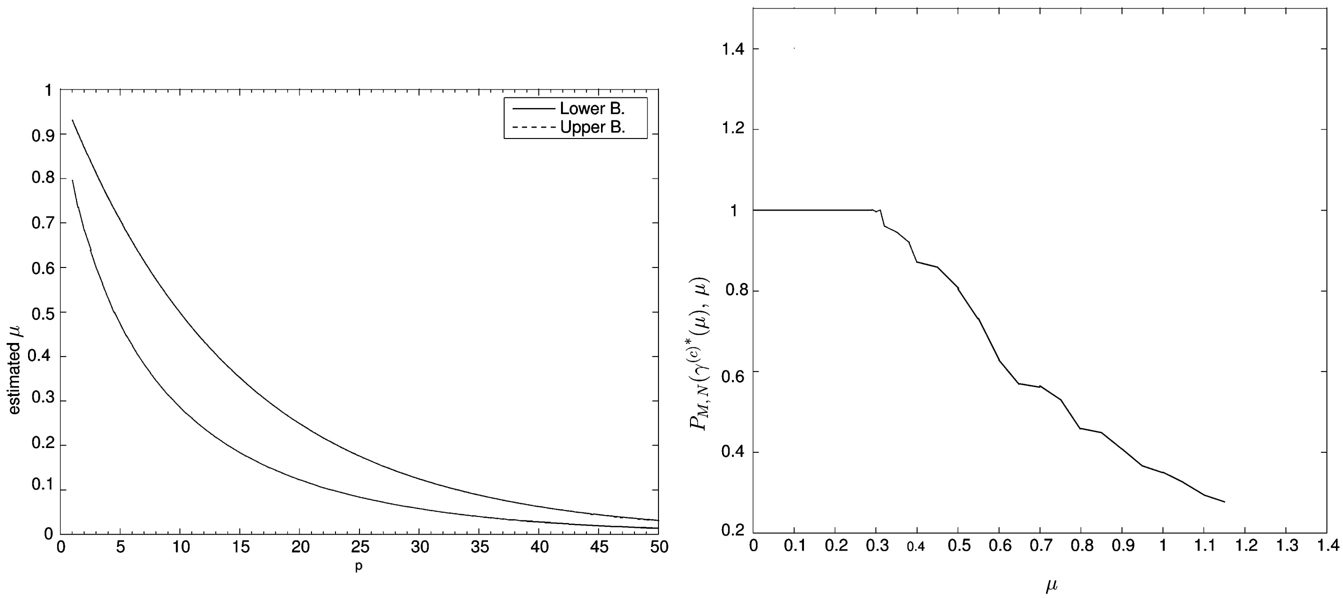

Fig. 2. (Left) Upper and lower bounds for the stability radius of the inverted pendulum. (Right) This plot shows (for period $p = 10$) an estimation of the probability of finding a norm ball of radius μ entirely contained in the stability region.

$\underline{M}$ and $\underline{N}$, such that for $M \geq \underline{M}$ and $N \geq \underline{N}$ it is possible to assert with confidence $1-\delta$ that $Pr\{p(\gamma^{(c)}, \mu) \leq P_N(\gamma^{(c)}_{M,N}, \mu)-\epsilon_1\} \leq \epsilon_2$.

We applied the technique on a real-life example: the angular stabilization a 2 *mm* micro-mechanical inverted pendulum, with a sampling period of $T_c = 1e-3$. The system state space is comprised of two variables: the angular position and the angular velocity. The upper bound was computed as shown in Proposition 2, while for the lower bound we applied the algorithm shown in Section IV-B.2. In Fig. 2 left, we compare the two bounds for different values of p. For $p = 10$, we got a considerable deviation between the two bounds: $\overline{\mu} = 0.49$ and $\underline{\mu} = 0.2844$. Therefore, we applied the randomised technique for $p = 10$. The levels of accuracy were chosen as $\epsilon_1 = 0.01$, $\epsilon_2 = 0.008$, $\delta = 0.008$. Theorem 10.3 of [24] recommends a number of samples $M \geq \underline{M} = 687$ and $N \geq \underline{N} = 63\,734$. The optimal probability $P_{M,N}(\gamma^{(c)^*}(\mu), \mu)$ was computed for different values of μ and is plotted on the right hand side of Fig. 2. At the lower bound, we got $P_{M,N}(\gamma^{(c)^*}(\underline{\mu}), \underline{\mu}) = 1$, while at the upper bound we got $P(\gamma^{(c)^*}(\overline{\mu}), \overline{\mu}) \approx 0.804$. We estimated the actual stability radius, searching the minimum μ^+ for which $P_{M,N}(\gamma^{(c)^*}(\mu^+), \mu^+) \leq 1-\epsilon_1$. This choice is reasonable because ϵ_1 is the maximum possible distance (with probability $1-\epsilon_2$) between the optimal probability $P(\gamma^{(c)^*}(\mu^+), \mu^+)$ and the its empirical estimate $P_{M,N}(\gamma^{(c)^*}(\mu^+), \mu^+)$. In our experiments, we found $\mu^+ \approx 0.315$ at $p = 10$. Therefore, the lower bound underestimates the stability radius by about 8%, while the upper bound overestimates the stability radius by about 55% This is, for our example, a worst case scenario as upper and lower bounds get closer to each other for $p > 10$. Consistent results were obtained for other examples of open loop unstable systems. Our conclusion is that the lower bound computed by the algorithm in Section IV-B.2 can be used as an approximation of the stability radius insofar as a maximum error of 10% is acceptable.

V. Design Optimization Problem

Based on the results of the previous sections, we propose a solution to the design problems presented in Section III. The stability radius $\mu^{(i)}$ is a function of $p^{(i)}$, parameterised by the dynamical parameters of system $\mathcal{S}^{(i)}$, that can be found analytically or otherwise estimated.

We tackle the problem by considering the relaxation to real values for the periods $p^{(i)}$. The integrality of the activation periods can be recovered by a Branch-and-bound scheme, which is not shown here for the sake of brevity (see [18] for a description).

A. The Continuous Relaxation

Introduce functions $\sigma^{(i)}(\mu)$ and $H(\mu)$ defined as follows:

$$\sigma^{(i)}(\mu) = \max\{p \in \mathbb{R} \text{ such that } p \geq 0, \mu^{(i)}(p) \geq \mu\}$$
$$\text{and } H(\mu) = U_l - \sum_i \frac{e^{(i)}}{\sigma^{(i)}(\mu)}. \quad (13)$$

Function $\sigma^{(i)}(\mu)$ is the maximum activation period of task $\tau^{(i)}$ that achieves a stability radius $\mu^{(i)} \geq \mu$. Clearly, it may be undefined for some values of μ. By construction, $\sigma^{(i)}(\mu)$ and $H(\mu)$ are decreasing functions of μ.

Fact 1: The following statements are true.

- **Problem P.1** is feasible, (i.e., a platform parameterised by $e^{(i)}, T_c, U_l$ can achieve μ_0), **iff**

$$H(\mu_0) \geq 0. \quad (14)$$

- If condition (14) holds, then $H(\cdot)$ has only one zero μ_* in the set $\mu \geq \mu_0$ that is optimal solution to **Problem P.2**. The optimal periods are given by $p^{(i)}_* = \sigma^{(i)}(\mu_*)$.

In this formulation of the feasibility and optimization problems, the complexity is hidden in the computation of $\sigma^{(i)}$. Observe that if $\mu^{(i)}(p^{(i)})$ is a decreasing function, then $\sigma^{(i)}(\mu)$ is its inverse.

1) First-Order Systems: For simplicity, we will assume that all the systems in $\mathcal{S}$ are open-loop unstable: $\lambda^{(i)}_1 > 0, \forall i = 1, \ldots, m$. Therefore, in view of (8), function $\mu^{(i)}$ is decreasing in $p^{(i)}$ and upper bounded by 1. Hence, $\sigma^{(i)}$ is defined only for $\mu \leq 1$ and it is given by: $\sigma^{(i)}(\mu) = (1/\lambda^{(i)}_1 T_c) \log((\lambda^{(i)}_1 + \mu|b^{(i)}|)/(\mu(\lambda^{(i)}_1 + |b^{(i)}|)))$. These expressions can be plugged into (13) to find the zero of $H(\mu)$ in the admissible range of μ.

2) Multidimensional Systems: For these systems, an analytical expression for $\mu^{(i)}$ and $H(\cdot)$ is not available. However, in the feasibility problem, we can use the lower bound $\underline{\mu}^{(i)}$ computed in Section IV-B.2. As discussed earlier, although conservative, this bound is regarded as a satisfactory approximation of the stability radius for most purposes. Functions $\sigma^{(i)}$ can be found numerically by interpolating the values of $\underline{\mu}^{(i)}$ evaluated on a grid of periods. Exponential interpolation turns out to be particularly convenient (the radius μ is upper-bounded by the exponential in (12), which is decreasing if $\mathrm{tr}(A) > 0$). The approximated $\sigma^{(i)}$ functions thus found can be used in the optimization problem yielding suboptimal solutions.

B. Numerical Example

For systems composed of a number of micro mechanical devices, sharing computation and communication resources is a very important issue. As an example of this scenario, we assumed the use of a shared platform to stabilise the angular dynamics of four inverted pendulums, similar to the one described in Section IV-B.VII. For the platform, we used an ARM 7 CPU board, operated at 50 MHz and endowed with 8 Kb of cache memory. Performance measures were taken using the Erika kernel [11] as RTOS and the RM scheduling policy ($U_l = 0.69$). The sampling clock period T_c, used for the time-triggered paradigm, was set to 1 *ms*.

The use of very small devices makes the stabilization problem nontrivial. The length of the different pendulums was chosen as: $l^{(1)} = 1.5 \text{ mm}, l^{(2)} = 2.5 \text{ mm}, l^{(3)} = 2 \text{ mm}$, and $l^{(4)} = 1.8 \text{ mm}$. The worst case execution time (inclusive of data filtering and computation of the feedback law) that we profiled on the platform for the control tasks was $e^{(i)}T_c \approx 499\ \mu\text{s}$. The stability radius was required to satisfy $\mu \geq \mu_0 = 0.55$.

Problem **P.1** has been tackled by computing the numeric inversion of the lower bound function evaluated at $\mu = 0.55$. This operation yields: $p^{(1)} = 3.879, p^{(2)} = 5.007, p^{(3)} = 4.479$ and $p^{(4)} = 4.249$. Even with truncation of the periods to the smaller integer (which preserves performance), we got $\sum (e^{(i)}/p^{(i)}) = 0.51 < 0.69$. Thereby, we concluded that the specification is attainable on the assumed platform. As far as Problem **P.2** is concerned, for each system we restricted the search to the set $[0, \overline{\sigma}_i(\mu)]$, where $\overline{\sigma}_i(\mu) = \max\{p \in \mathbb{R} s.t. \overline{\mu}_i(p) \geq 0.55\}$ and $\overline{\mu}_i(p)$ is the upper bound shown above. This is certainly a conservative choice, since if for a period p the specification is not attained for the upper bound of the stability radius, it is not attainable by the stability radius itself. We used the lower bound $\underline{\mu}_i(p)$ as an approximation of the stability radius. Namely, we computed this value on a grid of 50 points in the range $[0, \overline{\sigma}_i(\mu)]$, and interpolated by a sum of exponentials: $\underline{\mu}_i(p) = \sum_{j=1,\ldots,L} c_{i,j} e^{l_{i,j} p T_c}$. Parameters $l_{i,j}$ and $c_{i,j}$ were the result of a standard interpolation algorithm minimizing $\sum_h 0.5(\underline{\mu}_i(p_h) - \sum_{j=1,\ldots,L} c_{i,j} e^{l_{i,j} p_h T_c})^2$, where p_h are the period samples in the considered grid. By using two exponential functions, the error on the considered grid was of the order of 10^{-4} for all the systems in the considered set. Finally, we computed the optimum using the numeric procedure shown above getting $\mu_* = 0.598\,68$ with periods: $p^{(1)} = 2.58, p^{(2)} = 3.47, p^{(3)} = 2.9832$ and $p^{(4)} = 2.8307$. Among the $2^4 = 16$ choices of the closest integer periods, the one which is feasible and which maximizes μ should be selected.

VI. Conclusion

In this note, a design procedure was presented for taking into account the effects of limited computing power in the shared control platform. In particular, we considered a set of independent linear plants controlled by feedback loops and assumed that controllers are realized by means of periodic software tasks. The choice of the performance metric and a precise characterization of the control platform allowed for an effective analytical formulation of the design problems. In this context, we showed results for deciding whether a performance specification is attainable (Problem **P.1**) on the chosen platform. Moreover, if a platform qualifies, we proposed a methodology for finding optimal activation periods of tasks (Problem **P.2**).

This result is a first step toward a more general approach concerning mixed scheduling/control synthesis. Future work along the lines of this note includes formulations for the platform constraints, with two distinct purposes: 1) relaxing, if possible, the strong assumption inherent to the choice of the time-triggered approach, 2) extending the analysis to architectures with a higher degree of parallelism, such as multiprocessors and distributed architectures.

References

[1] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems*. Upper Saddle River, NJ: Prentice-Hall, 1997.

[2] V. Balakrishnan, "Linear matrix inequalities in robust control – A brief survey," in *Proc. 15th Int. Symp. Mathematical Theory of Networks and Systems (MTNS02)*, Notre Dame, IN, 2002.

[3] V. Balakrishnan and F. Wang, "Efficient computation of a guaranteed lower bound on the robust stability margin for a class of uncertain systems," *IEEE Trans. Autom. Control*, vol. 44, no. 11, pp. 2185–2190, Nov. 1999.

[4] A. Ben-Tal, L. El Ghaoui, and A. Nemirovskii, "Robust semidefinite programming," in *Handbook of Semidefinite Programming*, R. Saigal, L. Venberghe, and H. Wolkowic, Eds. Norwell, MA: Kluwer, 2000.

[5] S. Boyd and L. Vanderberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[6] R. Brockett, "Stabilization of motor networks," in *Proc. 34th IEEE Conf. Decision and Control*, 1995, pp. 1484–1488.

[7] ——, *Minimum Attention Control*, pp. 2628–2632, 1997.

[8] G. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Norwell, MA: Kluwer, 1997.

[9] A. Cervin, J. Eker, B. Bernhardsson, and K.-E. Årzén, "Feedback-feedforward scheduling of control tasks," Real Time Syst., 2002, to be published.

[10] P. M. Colom, "Analysis and design of real-time control systems with varying control timing constraints," Ph.D. dissertation, Automatic Control Dept., Tech. Univ. Catalonia, , Catalonia, Spain, 2002.

[11] P. Gai, G. Lipari, L. Abeni, M. di Natale, and E. Bini, "Architecture for a portable open source real-time kernel environment," in *Proc. 2nd Real-Time Linux Workshop and Hand's on Real-Time Linux Tutorial*, 2000.

[12] D. Hristu and K. Moransen, "Limited communication control," *Syst. Control Lett.*, vol. 37, no. 4, pp. 193–205, Jul. 1999.

[13] D. Hristu-Varsakelis, "Feedback control systems as users of a shared network: Communication sequences that guarantee stability," in *Proc. IEEE Conf. Decision and Control*, Dec. 2001.

[14] K. Kuetzer, S. Malik, R. Newton, J. M. Rabaey, and A. Sangiovanni-Vincentelli, "System-level design: Orthogonalization of concerns and platform-based design," *IEEE Trans. Computer-Aided Design Int. Circuits Syst.*, vol. 21, no. 7, pp. 1523–1543, Jul. 2000.

[15] B. Lincoln and A. Rantzer, "Optimizing linear system switching of switched linear systems," in *Proc. 40th IEEE Conf. Decision and Control*, Piscataway, NJ, Dec. 2001.

[16] 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, 1973.

[17] J. Nilsson, "Real-time control systems with delays," Ph.D. dissertation, Dept. Automatic Control, Lund Inst. Technol., Lund, Sweden, 1998.

[18] L. Palopoli, C. Pinello, A. L. Sangiovanni-Vincentelli, L. El-Ghaoui, and A. Bicchi, "Synthesis of robust control systems under resource constraints," in *Hybrid Systems: Computation and Control*, M. Greenstreet and C. Tomlin, Eds. Heidelberg, Germany: Springer-Verlag, 2002, vol. LNCS 2289, Lecture Notes in Computer Science, pp. 337–350.

[19] L. Palopoli, "Design of embedded control systems under real-time scheduling constraints," Ph.D. dissertation, ReTiS Lab—Scuola Supoeriore S. Anna, P.zza Martiri della Libertá, Pisa, Italy, 2002.

[20] L. Palopoli, A. Bicchi, and A. Sangiovanni Vincentelli, "Numerically efficient control of systems with communication constraints," in *Proc. IEEE 2002 Conf. Decision and Control* , Las Vegas, NV, Dec. 2002.

[21] H. Rehbinder and M. Sanfridson, "Scheduling of a limited communication channel for optimal control," in *Proc. 39th IEEE Conf. Decision and Control*, Sidney, Australia, Dec. 2000.

[22] D. Seto, J. P. Lehoczky, L. Sha, and K. G. Shin, "On task schedulability in real-time control systems," in *Proc. IEEE Real Time System Symp.*, Dec. 1996.

[23] C. M. Kirsch, T. Henzinger, and B. Horowitzm, "Embedded control systems development with giotto," in *Proc. ACM SIGPLAN 2001 Workshop on Languages, Compilers, and Tools for Embedded Systems (LCTES'2001)*, Jun. 2001.

[24] R. Tempo, G. Calafiore, and F. Dabbene, *Randomized Algorithms for Analysis and Control of Uncertain Systems*. New York: Springer-Verlag, 2004, ch. 10.

[25] M. Vidayagasar, "Randomized algorithms for robust controller synthesis using statistical learning theory," *Automatica*, vol. 37, pp. 1515–1528, 2001.

[26] W. Zhang, "Stability analysis of networked control system," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Case Western Reserve Univ., Cleveland, OH, 2001.

[27] Q. Zhao and D. Z. Zheng, "Stable real-time scheduling of a class of perturbed hybrid dynamic system," in *Proc. 14th IFAC World Congr.*, vol. J, Beijing, CHina, 1999, pp. 91–96.