Proceedings of the 45th IEEE Conference on Decision & Control
Manchester Grand Hyatt Hotel
San Diego, CA, USA, December 13-15, 2006



# Preliminary results on state-trigered scheduling of stabilizing control tasks

Paulo Tabuada and Xiaofeng Wang
Dept. of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556
E-mail: {ptabuada,xwang13}@nd.edu
http://www.nd.edu/∼ptabuada

*Abstract*— **In this paper we revisit the problem of scheduling stabilizing control tasks on embedded processors. We start from the paradigm that a real-time scheduler should be regarded as a feedback controller that decides which task is executed at any given instant. This controller has for objective guaranteeing that software tasks meet its deadlines and that stabilizing control tasks asymptotically stabilize the plant. According to this feedback paradigm, the decision of executing control tasks should not be based on release times and deadlines but rather on the state of the plant. We investigate the feasibility of a simple state triggered scheduler based on the state norm and provide some schedulability results.**

## I. Introduction

Small embedded microprocessors are quickly becoming an essential part of the most diverse applications. A particularly interesting example are physically distributed sensor/actuator networks responsible for collecting and processing information, and to react to this information through actuation. The embedded microprocessors forming the computational core of these networks are required to execute a variety of tasks comprising the relay of information packets in multi-hop communication schemes, monitoring physical quantities, and computation of feedback control laws. Since we are dealing with resource limited microprocessors it becomes important to assess to what extent we can increase the functionality of these embedded devices through novel real time scheduling algorithms based on state triggered rather than time triggered execution of control tasks.

We investigate a very simple state triggered scheduling algorithm that preempts running tasks to execute the control task whenever a certain error becomes large when compared with the state norm. This idea is an adaptation to the scheduling context of several techniques used to study problems of control under communication constraints [NE00], [BL00], [EM01]. We take explicitly into account the execution time of the control task and provide sufficient conditions for globally asymptotical stability under this scheduling policy. We also provide sufficient conditions for co-schedulability of the control task with other tasks competing for processor time. We illustrate the proposed approach through several computer simulations illustrating the tradeoff between rate of convergence and the execution frequency of the control task.

Real-time scheduling of control tasks has received renewed interest from the academic community in the past years [SLCB00], [CE00], [ACR+00], [CEBA02], [BA02], [CLS03], [MLB+04], [HC05], [GCHI06]. Common to all these approaches is the underlying principle that better control performance is achieved by providing more CPU time to control tasks. This can be accomplished in two different ways: letting control tasks run for longer amounts of time using anytime implementations or model predictive controllers; or by scheduling control tasks more frequently. In any case, the existence of a performance criterion for the control task is assumed, such as a cost function used to design an optimal linear quadratic regulator. Scheduling strategies are then obtained through on-line or off-line optimization algorithms seeking to determine schedules maximizing the performance criterion. The work presented in this paper does not resort to optimization and does not require a performance criterion. Instead, the decision to execute the control task is determined by a feedback mechanism based on the state of the plant.

Closer to the results presented in this paper is the work described in [PPV+02], [PPBSV05], where resource allocation and feedback control are designed in an integrated fashion. Several concurrent controllers described by scalar gains and activation rates of the corresponding processes are designed so as to ensure stability of the controlled processes as well as real-time schedulability. Close at the technical level, although addressing very different problems is the recent work on stabilization under communication constraints [NE00], [BL00], [EM01], [Lib03], [NT04], [LHLQ06], [BPZ02]. All these approaches are concerned with the stabilization of continuous systems under reduced communication and the employed techniques share with some of the techniques described in this paper, a common ancestor: the perturbation approach to stability analysis of control systems, described for example in [Kha96]. Finally, we would like to refer the reader to [AB02] where some advantages of event-driven control over time-driven control are presented in a stochastic setting.

This research was partially supported by the National Science Foundation EHS award 0509313.

## II. Sample-and-hold control

We start with a control system:

$$\dot{x} = f(x, u), \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m \tag{II.1}$$

for which a feedback control law:

$$u = k(x) \tag{II.2}$$

has been designed, rendering the closed loop system:

$$\dot{x} = f(x, k(x)) \tag{II.3}$$

globally asymptotically stable. Control law (II.2) is to be implemented in an embedded processor and we assume that its software implementation $C$ takes $\Delta$ units of time to be executed. This means that if $C$ is executed at time $\tau$, then the state of (II.1) is sampled at time $\tau$ and the actuators are updated at time $\tau+\Delta$ with the value $u = k(x(\tau))$. We further assume that the control value $u$ will be held constant until it is updated by another execution of the control task $C$. This means that to a sequence of execution times $\tau_1, \tau_2, \ldots, \tau_s$ of the control task $C$ there correspond a sequence actuator update times of $\tau_1+\Delta, \tau_2+\Delta, \ldots, \tau_s+\Delta$. When the actuator values are updated, there is already a discrepancy of $\Delta$ units of time between the current state $x(\tau+\Delta)$ and the state $x(\tau)$ used to compute control law (II.2). Furthermore, this discrepancy will continue to grow until the actuator values are updated by another execution of the control task. This discrepancy induces a state error defined by:

$$e(t) = x(\tau_i) - x(t), \qquad \tau_i + \Delta \le t < \tau_{i+1} + \Delta, \quad i \in \mathbb{N}$$

Taking this error into account, the result of closing the loop using a sample-and-hold software implementation $C$ of (II.2) can be described by:

$$\dot{x}(t) = g(x(t), e(t)) \tag{II.4}$$

$$\dot{e}(t) = -g(x(t), e(t)) \tag{II.5}$$

where $g(x(t), e(t)) = f(x(t), k(x(t) + e(t))) = f(x(t), k(x(\tau_i)))$ for $\tau_i + \Delta \le t < \tau_{i+1} + \Delta$. It will be useful to also introduce the error $\varepsilon$ defined as:

$$\varepsilon(t) = x(\tau_i) - x(t), \qquad \tau_i \le t < \tau_{i+1}, \quad i \in \mathbb{N}$$

## III. An academic example

To illustrate the results presented in this paper we will consider the following control system:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{III.1}$$

and a feedback control law rendering the closed loop asymptotically stable:

$$u = x_1 - 4x_2 \tag{III.2}$$

Associated with the closed loop is also a Lyapunov function $U = x^T P x$ with $P$ defined by:

$$P = \begin{bmatrix} 1 & \frac{1}{4} \\ \frac{1}{4} & 1 \end{bmatrix} \tag{III.3}$$

and with time derivative described by $\dot{U} = -x^T Q x$ where:

$$Q = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & \frac{3}{2} \end{bmatrix} \tag{III.4}$$

## IV. A stability abstraction

In oder to analyze stability of the closed loop system (II.4) and (II.5) under state-trigered scheduling policies we will need a model capturing the evolution of both $x$ and $e$. This model, which we shall call a stability abstraction, should be simple enough to allow for analysis and yet it should capture the interplay between the state evolution and the increase in the error caused by a sample-and-hold implementation of feedback. Furthermore, a stability abstraction should leverage on the knowledge of (II.3) that is obtained when designing (II.2). It is therefore without surprise that the proposed stability abstraction is an extension of the Lyapunov function used to prove stability of (II.3).

*Definition 4.1:* A stability abstraction for the closed loop system (II.4) and (II.5) consists of:

1) Four class[1] $\mathcal{K}_\infty$ functions $\alpha_1, \alpha_2, \alpha_3$ and $\alpha_4$;
2) Three smooth maps $V, E, \mathcal{E} : \mathbb{R}^n \to \mathbb{R}_0^+$ satisfying:
   a) $\alpha_1(\|x\|) < V(x) < \alpha_2(\|x\|)$;
   b) $\alpha_3(\|e\|) < E(e) < \alpha_4(\|e\|)$;
   c) $\mathcal{E} = E$;
   d) $-\underline{a}V(x) - \underline{b}E(e) \le \frac{\partial V}{\partial x} g(x, e) \le -\overline{a}V(x) + \overline{b}E(e)$;
   e) $-\frac{\partial E}{\partial e} g(x, e) \le \overline{c}V(x) + \overline{d}E(e)$;
   f) $-\frac{\partial \mathcal{E}}{\partial \varepsilon} g(x, e) \le \overline{c}V(x) + \overline{d}E(e)$,

where $\underline{a}, \overline{a}, \underline{b}, \overline{b}, \overline{c}, \overline{d} \in \mathbb{R}^+$.

As mentioned before, $V$ can be seen as the Lyapunov function guaranteeing stability of (II.3) provided that (II.3) is input-to-state stable (ISS) with respect to measurement errors. In addition, we require also a lower bound for $\dot{V}$ and similar ISS Lyapunov functions describing the evolution of the errors $e$ and $\varepsilon$. We shall discuss these assumptions in more detail in Section IX. For now, we will simply mention that linear control systems stabilized by linear feedback always admit a stability abstraction. In this case, if the closed loop is described by:

$$\dot{x} = Ax + BKx \tag{IV.1}$$

and if $U = x^T P x$ is a Lyapunov function for (IV.1) satisfying $\dot{U} = -x^T Q x$ with $P^T = P > 0$ and $Q^T = Q > 0$, then we can explicitly construct a stability abstraction by:

$$V = \sqrt{x^T P x} \qquad E = \sqrt{e^T P e} \qquad \mathcal{E} = \sqrt{\varepsilon^T P \varepsilon}$$

$$\underline{a} = \frac{\lambda_M(Q)}{\lambda_m(P)} \qquad \overline{a} = \frac{\lambda_m(Q)}{\lambda_M(P)}$$

$$\underline{b} = \frac{2\|PBK\|}{\lambda_m(P)} \qquad \overline{b} = \frac{2\|PBK\|}{\lambda_m(P)}$$

$$\overline{c} = \frac{\lambda_M(Q)}{\lambda_m(P)} \qquad \overline{d} = \frac{2\|PBK\|}{\lambda_m(P)}$$

*Example 4.2:* For the control system and feedback described in Section III, the corresponding stability abstraction

[1] A continuous function $\gamma : [0, a[ \to \mathbb{R}_0^+$, $a > 0$, is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\gamma(0) = 0$. It is said to belong to class $\mathcal{K}_\infty$ if $a = \infty$ and $\gamma(r) \to \infty$ as $r \to \infty$.

is characterized by the following parameters:

$$\underline{a} = \overline{c} = \frac{1}{3}(4+\sqrt{5}) \quad \overline{a} = \frac{1}{5}(4-\sqrt{5}) \quad \underline{b} = \overline{b} = \overline{d} = \frac{34}{3}$$

## V. An error control strategy

The strategy underlying the scheduling algorithm to be proposed is based on the following simple idea that has been used before to study stabilization problems under communication constraints [NE00], [BL00], [EM01]. Recall that $V$ satisfies the following differential inequality:

$$\dot{V} \leq -\overline{a}V + \overline{b}E \tag{V.1}$$

A simple stabilization strategy, based on (V.1), consists in keeping the error, as measured by $E$, small when compared with the state, as measured by $V$. More specifically, if the inequality:

$$E \leq \gamma V$$

is enforced for some $\gamma > 0$, (V.1) becomes:

$$\dot{V} \leq -(\overline{a} - \gamma\overline{b})V$$

implying that:

$$V(t) \leq e^{-(\overline{a}-\gamma\overline{b})t}V(0) \tag{V.2}$$

Stability is now guaranteed by $\overline{a} - \gamma\overline{b} > 0$ or equivalently by $\gamma < \overline{a}/\overline{b}$. Since executing the control task at time $\tau$ has the effect of reducing the state error to zero, an obvious scheduling strategy guaranteeing global asymptotical stability consists in scheduling the control task sufficiently often in order to enforce $E \leq \gamma V$. One of the advantages of such scheduling strategy is that asymptotical stability is automatic guaranteed by the inequality $\gamma < \overline{a}/\overline{b}$. The important question becomes to determine *when* to execute the control task. In order to address this question it is fundamental to understand the evolution of the ratio $E/V$. By making use of a stability abstraction we can obtain the following estimate of the time derivative of $E/V$:

$$\begin{aligned} \frac{d}{dt}\frac{E}{V} &= \frac{\dot{E}V - E\dot{V}}{V^2} \\ &\leq \frac{\overline{c}V^2 + \overline{d}EV + \underline{a}EV + \underline{b}EE}{V^2} \\ &= \overline{c} + (\underline{a}+\overline{d})\frac{E}{V} + \underline{b}\Big(\frac{E}{V}\Big)^2 \end{aligned} \tag{V.3}$$

providing:

$$\frac{E}{V}(t) \leq -\frac{1}{2\underline{b}}\left(\underline{a} + \overline{d} - \Theta\tan\left(\frac{1}{2}\Theta(t+\Psi)\right)\right) \tag{V.4}$$

where:

$$\Theta = \sqrt{4\underline{b}\overline{c} - (\underline{a}+\overline{d})^2}$$

and $\Psi$ is the constant satisfying:

$$\frac{E}{V}(0) = -\frac{1}{2\underline{b}}\left(\underline{a} + \overline{d} - \Theta\tan\left(\frac{1}{2}\Theta(\Psi)\right)\right)$$

Note that when $4\underline{b}\overline{c} - (\underline{a}+\overline{d})^2 < 0$, $\Theta$ is a pure imaginary number and we take:

$$\tan\left(\frac{1}{2}\Theta(t+\Psi)\right) = -i\tanh\left(-\frac{i}{2}\Theta(t+\Psi)\right)$$

so that $E/V$ is still a real number. For later use we will denote by $\rho(z,t)$ the upper bound (V.4), that is:

$$\rho(z,t) = -\frac{1}{2\underline{b}}\left(\underline{a} + \overline{d} - \Theta\tan\left(\frac{1}{2}\Theta(t+\Psi)\right)\right)$$

with $\Psi$ satisfying:

$$z = -\frac{1}{2\underline{b}}\left(\underline{a} + \overline{d} - \Theta\tan\left(\frac{1}{2}\Theta(\Psi)\right)\right)$$

## VI. Scheduling the control task

In an ideal computing system where $\Delta = 0$, the scheduling strategy described in the previous section could be implemented by executing the control task when $E/V = \gamma$. However, since $\Delta$ units of time elapse between the measurement of the state and the actuators update, a different value of $E/V$ has to be used. An estimate for such value can be immediately obtained by using (V.4).

*Proposition 6.1:* Consider a closed loop system of the form (II.4) and (II.5) equipped with a stability abstraction. Then, for any $\gamma \in \mathbb{R}$ if $\sigma \leq \rho(\gamma, -\Delta)$ the following implication holds:

$$E(\tau) = \sigma V(\tau) \quad \Longrightarrow \quad E(\tau+\delta) \leq \gamma V(\tau+\delta)$$

for any $\delta \in [0, \Delta]$.

This proposition shows that the bound $E \leq \gamma V$ can be enforced by executing the control task when $E = \sigma V$. However, there is no guarantee that $V(\tau+\delta) = \sigma\mathcal{E}(\tau+\delta)$ will not hold for some $0 \leq \delta \leq \Delta$ thereby triggering another execution of the control task before termination of the previous one. We thus need one additional constraint on $\sigma$.

*Proposition 6.2:* Consider a closed loop system of the form (II.4) and (II.5) equipped with a stability abstraction. Then, for any $\gamma \in \mathbb{R}$ if $\sigma \in \mathbb{R}$ satisfies:

$$\rho(0,\Delta) < \sigma \leq \rho(\gamma, -\Delta) \tag{VI.1}$$

the following holds:

$$\begin{aligned} &E(\tau) = \sigma V(\tau) \quad \Rightarrow \quad E(\tau+\delta) \leq \gamma V(\tau+\delta) \\ &\mathcal{E}(\tau+\delta) < \sigma V(\tau+\delta) \end{aligned}$$

for any $\delta \in [0, \Delta[$.

*Proof:* At $t = \tau$ sensors are read and this implies $\mathcal{E}(\tau) = 0$ which leads to $\big(\mathcal{E}/V\big)(\tau) = 0$. Since $\dot{E} = \dot{\mathcal{E}}$ we have:

$$\begin{aligned} \big(\mathcal{E}/V\big)(\tau+\Delta) &< \sigma \\ \rho\Big(\big(\mathcal{E}/V\big)(\tau), \Delta\Big) &< \sigma \\ \rho(0,\Delta) &< \sigma \end{aligned}$$

Furthermore, it follows from (V.3) that $\frac{d}{dt}\mathcal{E}/V > 0$ and thus $\rho(0,\Delta) < \sigma$ implies $\rho(0,\delta) < \sigma$ for any $\delta \in [0, \Delta[$. ■

| $\gamma$ | $\sigma=\rho(\gamma,-\Delta)$ | $\overline{a}-\gamma\overline{b}$ | $\tau^*$ |
|---|---|---|---|
| 0.03 | 0.0251 | 0.0128 | 0.0092 |
| 0.02 | 0.0154 | 0.1261 | 0.0050 |
| 0.01 | 0.0056 | 0.2395 | 0.0006 |

TABLE I

## VII. Schedulability of Control Tasks

One of the major difficulties with event based scheduling is time predictability. If no restrictions on the event occurrence times exist, then it is not possible to provide any schedulability guarantees. In the particular context of closed loop systems equipped with stability abstractions we can use (V.4) to estimate a lower bound for the time elapsed between executions of the control task.

*Proposition 7.1:* Consider a closed loop system of the form (II.4) and (II.5) equipped with a stability abstraction. Then, for any $\gamma \in \mathbb{R}$ and any $\sigma \in \mathbb{R}$ satisfying (VI.1) the time elapsed between interrupts triggering the control task is lower bounded by $\Delta+\tau^*$ with $\tau^*$ defined by the equation:

$$\rho(0,\Delta+\tau^*)=\sigma$$

*Proof:* If the control task is executed at $\tau$ it terminates its execution at $\tau+\Delta$ and as discussed in the proof of Proposition 6.2, $\big(\mathcal{E}/V\big)(\tau+\Delta)=\rho(0,\Delta)$. The next interrupt will occur when $E/V=\sigma$, that is, when $\rho(\rho(0,\Delta),\tau^*)=\rho(0,\Delta+\tau^*)=\sigma$. ∎

*Example 7.2:* For the control system and controller described in Section III and for an implementation with execution time $\Delta=0.002s$ we obtain the values reported in Table I. Note that in order to guarantee asymptotic stability $\gamma$ must be smaller than $\overline{a}/\overline{b}\approx 0.0311$. As intutively expected, we see from Table I that higher values of $\gamma$ guarantee a longer time between executions of the control task at the expense of a slower convergence measured by $\overline{a}-\gamma\overline{b}$ and described by (V.2). The value of $\gamma$ can thus be regarded as a design parameter capturing the tradeoff between performance (decay rate of $V$) and execution frequency of the control task. The effect of $\gamma$ will be further discussed in Section VIII where simulation results are reported.

The lower bound between execution times of the control task presented in Proposition 7.1 can be used to analyze schedulability of a set of tasks $T=\{T_i\}_{i\in I}$. We shall assume a preemptive scheduler in which the control task has the highest priority and thus cannot be preempted by any other task and is executed without delays when $V=\sigma E$. Note that timing overheads associated with context switching can be captured in the proposed framework by suitably enlarging $\Delta$. We regard execution of the control task as a timing overhead imposed on the tasks $T_i$ by the scheduler in the sense that $T_i$ may need to be interrupted to execute the control task. When a set of control tasks $T$ can be scheduled despite the overhead associated with the control task we say that $T$ is co-schedulable with the control task. Co-schedulability is now ensured by the following sufficient condition where $\lceil r\rceil$ denotes the smallest integer greater than $r\in\mathbb{R}$.

*Theorem 7.3:* Consider a closed loop system of the form (II.4) and (II.5) equipped with a stability abstraction and let $T=\{T_i\}_{i\in I}$ be a set of tasks with execution times $\{\Delta_i\}_{i\in I}$. If the set of tasks $T$ is schedulable with new execution times $\Delta_i'$ given by:

$$\Delta_i'=\Delta_i+\lceil\Delta_i/\tau^*\rceil\Delta$$

then $T$ is co-schedulable with the control task.

*Proof:* Since the time between executions of the control task is lower bounded by $\Delta+\tau^*$, each task $T_i$ can be interrupted at most $\lceil\Delta_i/\tau^*\rceil$ times. Each interruption will delay the execution of $T_i$ by $\Delta$ units of time resulting in a total execution time of $\Delta_i+\lceil\Delta_i/\tau^*\rceil\Delta$. ∎

Co-schedulability is therefore ensured by the existence of a schedule where the duration $\Delta_i$ of each task $T_i$ has been inflated to $\Delta_i'$ in order to accommodate the timing overhead imposed by the control task.

## VIII. Simulations

In this section we report on simulation results obtained for the control system and feedback controller described in Section III.

On figure 1 we present the evolution of $V$ for the three values of $\sigma$ reported on Table I. Before the first execution of the control task no control is being exerted on the plant and we thus see an increase in $V$. We also see that increasing the value of $\sigma$ has the effect of reducing the rate at which $V$ decreases to zero. Figure 2 illustrates the proposed state-

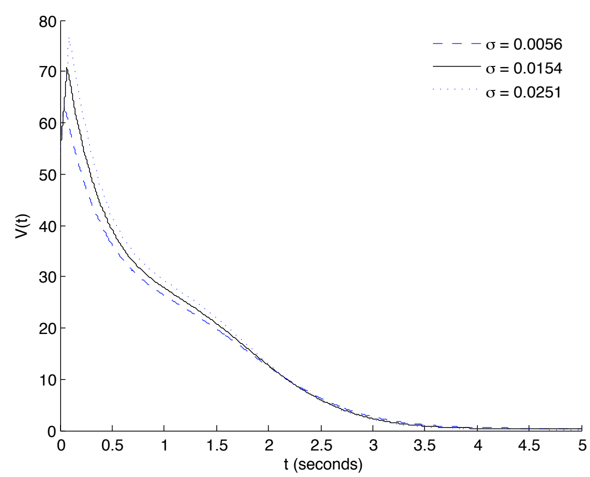


Fig. 1. Evolution of $V$ for different values of $\sigma$ and initial condition $(x_1(0),x_2(0))=(10,20)$.

trigered scheduling policy based on the evolution of $E(t)$ and $V(t)$. The control task is executed when $E=\sigma V$, however, since only $\Delta=0.002$ seconds later the actuators are updated with the result of the control task, we see that $E$ goes above the $\sigma V$ threshold, however it does not reach the $\gamma V$ threshold as guaranteed by Proposition 6.1. The maximum and minimum time between executions of the control task is reported on Table II. We see that the theoretical estimate for $\Delta+\tau^*$ reported on Table I is off by an order of magnitude. This reflects the fact that we are

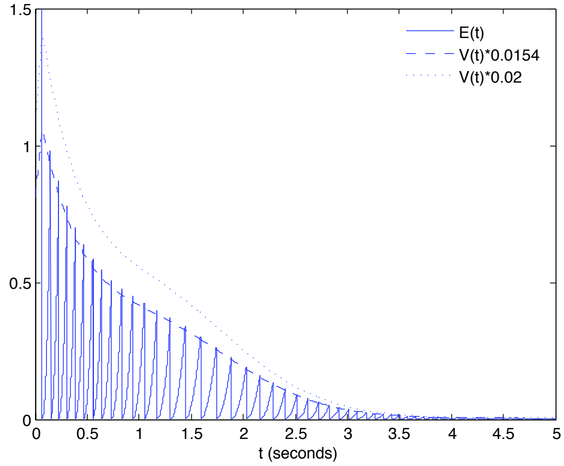


Fig. 2. Evolution of $E$ and $\sigma V$ for $\sigma = 0.0154$ and initial condition $(x_1(0), x_2(0)) = (10, 20)$.

| $\sigma$ | Maximum time | Minimum time | $\Delta + \tau^*$ |
|---|---|---|---|
| 0.0251 | 0.191 | 0.096 | 0.0112 |
| 0.0154 | 0.151 | 0.078 | 0.0070 |
| 0.0056 | 0.094 | 0.051 | 0.0026 |

TABLE II

useing very conservative estimates for $\dot{V}$ and $\dot{E}$ which can be confirmed by comparing the uperbound for $V(t)$ described by V.2 with the simulated evolution of $V(t)$ in Figure 3. Further evidence of the conservatiness of the used estimates is provided in Figure 4 where the estimated evolution of $E/V$ is compared with the simulated evolution. Finally, we

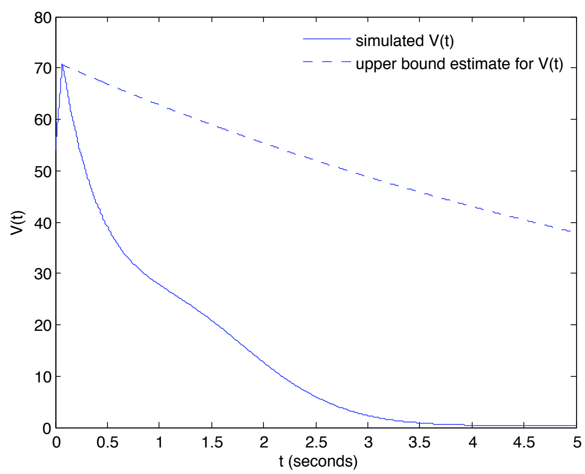


Fig. 3. Evolution of the upper bound on $V$ defined in (V.2) and the simulated value of $V$ for initial condition $(x_1(0), x_2(0)) = (10, 20)$.

show in Figure 5 the evolution of $V$ for a value of $\sigma = 0.1$ which is well above the estimated upper bound guaranteeing stability $\overline{a}/\overline{b} \approx 0.0311$. Even with such large value of $\sigma$ the plant is stabilized albeit the larger initial period where $V$ increases before the first execution of the control task.

These results suggest that although the theoretical results provide a good qualitative description of the closed loop behavior under the proposed state triggered scheduling strategy, further research is required to obtain better estimates as elaborated on Section IX.

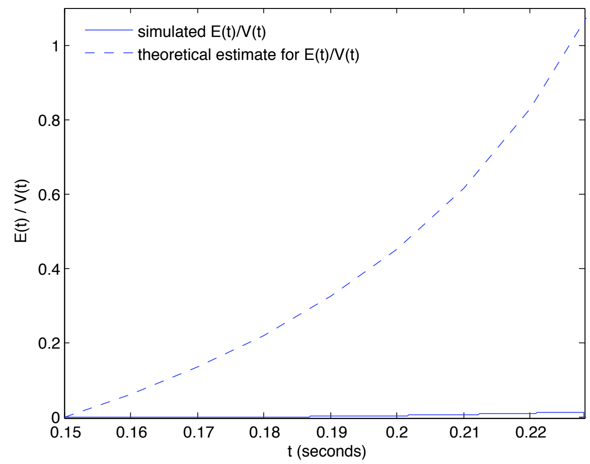


Fig. 4. Evolution of $E/V$ obtained by simulation and through expression (V.4) for $\sigma = 0.0154$.

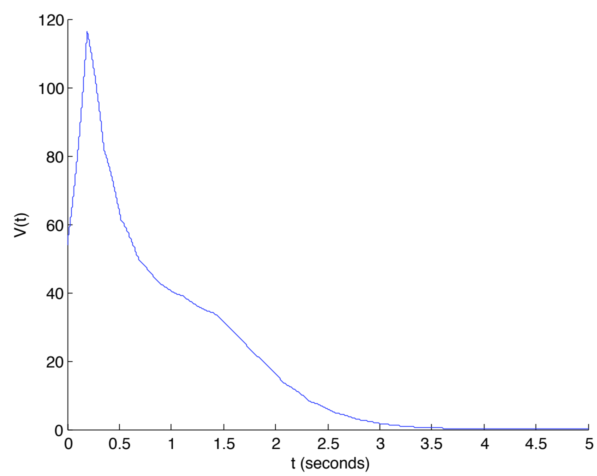


Fig. 5. Evolution of $V$ for $\sigma = 0.1$ and initial condition $(x_1(0), x_2(0)) = (10, 20)$.

## IX. Discussion

### A. Conservativeness and new stability abstractions

The simulation results reported on Section VIII show that stability abstractions, although providing a good qualitative description of the closed loop behavior, are very conservative. This is somewhat unavoidable since we are describing the stability properties of the system by using a single differential equation for $V$. Furthermore, since this differential equation is obtained when designing the stabilizing controller it seems disadvantageous not to incorporate it in a stability abstraction. Contrary to the use of $V$, the proposed model for $E$ does not seem to be so essential. In fact, the estimate of $E$ was only used to obtain the estimate of $E/V$ which suggests that instead of assuming the knowledge of a lower bound for $\dot{V}$ and an upper bound for $\dot{E}$ we can directly assume the existence of an upper bound for $\frac{d}{dt}(E/V)$. An estimate for $E/V$ has to be simple enough to admit a simple analysis and preferentially a closed form solution, but it must also be as less conservative as possible. Determining the exact form of such bound (linear, quadratic, etc) is the subject of on going research.

### B. On-line adaptation

The parameter $\sigma$ captures the existing tradeoffs between rate of convergence and usage of computational resources. In a dynamic environment where the processor load is changing over time, it makes sense to have a higher level of supervision responsible for regulating the value of $\sigma$. Similar control inspired scheduling strategies have already appeared in the literature [SLCB00], [CE00]. However, the existing methods do not provide guarantees that hard bounds such as (VI.1) are enforced. Current research is focusing of developing such feedback mechanisms responsible for regulating $\sigma$ as a function of processor load in order to improve performance.

### C. Hardware requirements for state-trigered scheduling

Even though we have not addressed how to implement the proposed scheduling strategy, it is clear that it requires special purpose hardware. The control task is triggered by the equality $V = \sigma E$ and testing this equality by a dedicated software task would only contribute to increase the processor load. Instead, the comparison $V < \sigma E$ can be converted into $\|x\| < \alpha_1^{-1}(\sigma\alpha_4(\|e\|))$ which in the linear case reduces to $\|x\| < \sigma l\|e\|$ for some constant $l > 0$. This comparison can now be easily implemented in hardware with a violation of the bound $\|x\| < \sigma l\|e\|$ triggering the execution of the control task. Note that replacing $\|x\| < \alpha_1^{-1}(\sigma\alpha_4(\|e\|))$ with $\|x\| < \sigma l\|e\|$ contributes to further increase the conservativeness of the approach. Ultimately, the advantages of the proposed scheduling policy can only be decided through the analysis of concrete examples.

## REFERENCES

[AB02] K. J. Astrom and B.M. Bernhardsson. Comparison of riemann and lebesgue sampling for first order stochastic systems. In *Proceedings of the 41st IEEE Conference on Decision and Control*, December 2002.

[ACR+00] P. Albertos, A. Crespo, I. Ripoll, M. Valles, and P. Balbastre. Rt control scheduling to reduce control performance degrading. In *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia, 2000.

[BA02] G. Buttazzo and L. Abeni. Adaptive workload management through elastic scheduling. *Real-Time Systems*, 23(1):7–24, 2002.

[BL00] R.W. Brockett and D. Liberzon. Quantized feedback stabilization of linear systems. *IEEE Transactions on Automatic Control*, 45(7):1279–1289, 2000.

[BPZ02] M. Branicky, S. Phillips, and W. Zhang. Scheduling and feedback co-design for networked control systems. In *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, NV, December 2002.

[CE00] Anton Cervin and Johan Eker. Feedback scheduling of control tasks. In *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia, 2000.

[CEBA02] A. Cervin, J. Eker, B. Bernhardsson, and K.-E. Arzen. Feedbackfeedforward scheduling of control tasks. *Real-Time Systems*, 23(1):25–53, 2002.

[CLS03] R. Chandra, X. Liu, and L. Sha. On the scheduling of flexible and reliable real-time control systems. *Real-Time Systems*, 24(2):153–169, 2003.

[EM01] N. Elia and S.K. Mitter. Stabilization of linear systems with limited information. *IEEE Transactions on Automatic Control*, 46(9):1384–1400, 2001.

[GCHI06] M. Ben Gaid, A. Cela, Y. Hamam, and C. Ionete. Optimal scheduling of control tasks with state feedback resource allocation. In *Proceedings of the 2006 American Control Conference*, Minneapolis, Minnesota, USA, June 2006.

[HC05] Dan Henriksson and Anton Cervin. Optimal on-line sampling period assignment for real-time control tasks based on plant state information. In *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference*, Seville, Spain, December 2005.

[Kha96] Hassan K. Khalil. *Nonlinear Systems*. Prentice Hall, New Jersey, second edition, 1996.

[LHLQ06] D. Liu, S. Hu, M. Lemmon, and L. Qiang. Firm real-time system scheduling based on a novel qos constraint. *IEEE Transactions on Computers*, 55(3):320–333, 2006.

[Lib03] Daniel. Liberzon. Hybrid feedback stabilization of systems with quantized signals. *Automatica*, 39(9):1543–1554, 2003.

[MLB+04] Pau Marti, Caixue Lin, Scott Brandt, Manel Velasco, and Josep Fuertes. Optimal state feedback resource allocation for resource-constrained control tasks. In *IEEE Real-Time Systems Symposium (RTSS 2004)*, page 161172, Lisbon, Portugal, December 2004.

[NE00] Girish N. Nair and Robin J. Evans. Stabilization with data-rate-limited feedback: Tightest attainable bounds. *Systems and Control Letters*, 41(1):49–56, 2000.

[NT04] D. Nesic and A.R. Teel. Input to state stability of networked control systems. *Automatica*, 40:2121–2128, 2004.

[PPBSV05] L. Palopoli, C. Pinello, A. Bicchi, and A. Sangiovanni-Vincentelli. Maximizing the stability radius of a set of systems under real-time scheduling constraints. *IEEE Transactions on Automatic Control*, 50(11):1790–1795, 2005.

[PPV+02] Luigi Palopoli, Claudio Pinello, Alberto Sangiovanni Vincentelli, Laurent Elghaoui, and Antonio Bicchi. Synthesis of robust control systems under resource constraints. In Claire Tomlin and Mark R. Greenstreet, editors, *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Sience, pages 337–350. Springer-Verlag, 2002.

[SLCB00] Lui Sha, Xue Liu, Marco Caccamo, and Giorgio Buttazzo. Online control optimization using load driven scheduling. In *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia, 2000.