On the minimum attention and anytime attention problems for nonlinear systems

Adolfo Anta and Paulo Tabuada
Dept. of Electrical Engineering, University of California, Los Angeles
E-mail: {adolfo,tabuada}@ee.ucla.edu

***Abstract*— The current convergence of control, communication and computation opens the door to new avenues for the applicability of control systems, but it also leads to new constraints and requirements. The traditional quadratic cost function has become too plain to represent the diverse performance indices of today's complex applications. For instance, the implementation cost of a control law has become a major factor, since resources are frequently shared among different tasks. Implementation costs of control laws depend on a wide range of factors: computation complexity, sensor accuracy, required bandwidth, actuation speed, or sampling rates, to name a few. In this paper we focus on the notion of attention, that is, how often the loop needs to be closed in order to achieve a desired performance. We propose two attention-aware control laws: first, a technique to construct minimum attention control laws that maximizes the open loop operation of a nonlinear control system; and second, a control algorithm that permits the system run in open loop for a pre-scheduled amount of time.**

I. Introduction

With the onset of advanced communication networks and microprocessors, control laws are usually no longer executed on dedicated platforms, but rather on shared devices. In this shared setup, numerous new problems arise, such as real-time scheduling, delay effects, packet dropouts, co-design of communication protocols and control laws, etc. One of such problems lies in how to reduce the amount of resources (computations, bandwidth) required by a control system. New policies have been recently developed to address these issues. Anytime control ([FGB08], [Gup09]) takes into account the time varying computation resources to decide which control law can be executed at each invocation. Sampling period selection ([SLSS96]), on the other hand, focuses in changing the execution periods according to the current state of the plant and the current load of the shared platform. Other researchers opt for dropping the periodicity assumption for control tasks and propose new paradigms such as event-triggered and self-triggered control ([Årz99], [VFM03], [Tab07], [AT10]). In all these analyses, the control laws are previously designed and the focus is on how to implement such algorithms in a resource-aware manner. Moreover, most of those strategies are aimed at linear systems. In this paper we propose a control strategy that minimizes the *attention* needed by a control loop. By attention we mean *how often* it is necessary to close the loop in a control system, *i.e.*, to sample the state, execute the control law and update the actuator. Hence, minimizing the attention of a control loop minimizes both the number of executions and the bandwidth that a control task requires.

The concept of attention of a control law was introduced in [Bro97]. In that seminal paper the attention index was a function of the variation of the input and the state with respect to time ($\frac{\partial \mathrm{u}}{\partial t}, \frac{\partial \mathrm{x}}{\partial t}$): control laws with small values of $||\frac{\partial \mathrm{u}}{\partial t}||$ and $||\frac{\partial \mathrm{x}}{\partial t}||$ imply less frequent updates. The problem, posed as the minimization of such attention index, leads to a set of partial differential equations, hard to solve even in the linear case. Instead, we propose several minimum attention control laws that attain suboptimal solutions at different complexity costs. In our path to answer this question, we propose a novel solution to the anytime control problem: assuming that the control input cannot be recomputed in the next T units of time, find a control input that guarantees a desired performance until the next execution time. This question naturally arises for instance in real-time scheduling problems, where a microprocessor is in charge of the execution of a set of tasks. In this context, a scheduler distributes the computational resources among all tasks, that is, it assigns online the execution times for the control tasks. The control input needs to be designed to let the control system run in open loop until the next prescheduled execution time.

A related problem was tackled in [Cha07], where an open-loop control law is designed in order to minimize the need of feedback. In this paper we assume that the control input is only updated whenever the state is measured, and it is kept constant between updates. Notice that the framework in [Cha07] corresponds to the case when computational resources are available at the actuator level, as in the case of smart actuators. On the other hand, the problem we address in this paper applies to systems where computations are costly or the rate of actuator changes is limited. Such limitations appear frequently in a wide range of applications, ranging from embedded control systems, where a single microprocessor is in charge of several tasks, to the control of stock portfolio, where a fee is charged for every transaction.

Our results leverage the self-triggered control framework, previously developed by the authors in ([AT09], [AT10]). Using this mathematical formulation, we derive a formula for the inter-execution times τ_i of a control law as a function of the last measurement of the state $\mathrm{x}(\tau_i)$ and the control input $\mathrm{u}(\tau_i)$:

$$\tau_{i+1} = \gamma(\mathrm{x}(\tau_i), \mathrm{u}(\tau_i))$$

The minimum attention control input $\mathrm{u}^*(\tau_i)$ is thus the one maximizing τ_i for a given state $\mathrm{x}(\tau_i)$. Depending on the expression of $\gamma(\mathrm{x}, \mathrm{u})$, the solution of this maximization problem might be computationally very expensive. For that matter, we propose different formulas leading to suboptimal control inputs at a low computational cost. Likewise, we resort to this expression of γ to compute the set of inputs that let the system run in open loop for a given duration of time, in the spirit of anytime control.

This research was partially supported by the National Science Foundation EHS award 0834771.

II. Notation and Problem statement

A. Notation

We shall use the notation $|x|$ to denote the Euclidean norm of an element $x \in \mathbb{R}^n$. We consider control systems of the form:

$$\dot{\mathrm{x}} = f(\mathrm{x}, \mathrm{u}), \qquad \mathrm{x}(t) \in \mathbb{R}^n,\ \mathrm{u}(t) \in \mathbb{R}^p, t \in \mathbb{R}_0^+ \tag{II.1}$$

We denote by $x \in \mathbb{R}^n$ the state of the control system, by x a solution of (II.1), and by u the input trajectory.

A function is said to be of class C^∞ or smooth if it can be differentiated infinitely many times. All the objects in this paper are considered to be smooth unless otherwise stated. We use $(\mathcal{L}_X h)(x)$ to denote the Lie derivative of a map $h : M \to \mathbb{R}$ evaluated at a point x along the flow of the vector field $X : M \to TM$ which, in local coordinates, we take as $\frac{\partial h}{\partial x} X(x)$. Likewise, $\mathcal{L}_X^k h$ represents the kth Lie derivative, defined by $\mathcal{L}_X^0 h = h$ and $\mathcal{L}_X^k h = \mathcal{L}_X(\mathcal{L}_X^{k-1} h)$.

B. Problem statement

Consider the control system described in (II.1), for which a control law needs to be designed to achieve a desired performance. The implementation of such control algorithm on a digital platform is typically done by sampling the state x at a time instant t_i, computing the control algorithm and updating the actuator with the new control input u. A sample-and-hold implementation is considered at the actuation level, *i.e.*:

$$t \in [t_i, t_{i+1}[\Rightarrow \mathrm{u}(t) = \mathrm{u}(t_i) \tag{II.2}$$

Under this implementation, both a sequence of control inputs $\{u_i\}_{i\in\mathbb{N}}$ and a sequence of time instants $\{t_i\}_{i\in\mathbb{N}}$ need to be computed. Traditionally, this sequence of times is chosen to be periodic, *i.e.*, $t_{i+1} - t_i = T$ for any $i \geq 0$ and period $T > 0$. However, this is not necessary and leads to a conservative usage of resources (microprocessor time, communication bandwidth, etc.).

In this paper the performance requirement of a control system will be determined by a desired rate of decay $h : \mathbb{R}^n \to \mathbb{R}$ of a given control Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}$ for the system (II.1):

$$\dot{V}(t) \leq h(\mathrm{x}(t)) \quad \forall t \in \mathbb{R}_0^+ \tag{II.3}$$

The sequence of inputs u_i and time instants t_i have to be chosen so that inequality (II.3) is satisfied at any time. We assume that there always exists an input achieving the desired performance:

$$\forall x \in \mathbb{R}^n \quad \exists u \in \mathbb{R}^p : \frac{\partial V}{\partial x} f(x, u) < h(x) \tag{II.4}$$

That is, V represents a control Lyapunov-like function. The problems we tackle in this paper are the following:

1) *Consider a control system of the form (II.1). Given a desired control performance h and a function V satisfying (II.4), how to compute a control input u_i^* that maximizes the time gap between execution times (i.e., requires the least attention), while satisfying the desired performance as defined by (II.3)?*

   In other words, the optimal control input u_i^* can be mathematically defined as follows:

$$u_i^* = \arg\max\{t_{i+1}(u_i) - t_i\} \tag{II.5}$$

$$t_{i+1} = \min\{t > t_i : \dot{V}(t) = h(\mathrm{x}(t))\} \tag{II.6}$$

   Furthermore, the input u_i^* needs to be selected so that the required performance is satisfied between executions:

$$\left.\frac{\partial V}{\partial x}\right|_{x=\mathrm{x}(t)} f(\mathrm{x}(t), u_i^*) < h(\mathrm{x}(t)) \quad \forall t \in [t_i, t_{i+1}[$$

2) *Consider a control system of the form (II.1). Given a desired control performance h and a function V satisfying (II.4), find a control input u_i so that the desired performance is satisfied from the current time instant t_i until a given next execution time t_{i+1}.*

   While in the previous problem the sequence of inputs u_i needs to be chosen so that inter-execution times are enlarged, now the sequence of times t_i is fixed, and the objective is to find at each time instant t_i the set of input values Ω that let the system run in open loop (while satisfying (II.3)) until the prescheduled next execution time t_{i+1}:

$$\Omega = \{u \in \mathbb{R}^p \,|\, \dot{V}(t) - h(\mathrm{x}(t)) < 0,\ t \in [t_i, t_{i+1}[\ \}$$

We first focus on the solution to the first problem, that could be posed as an optimization problem, since we are looking for an input $u \in \mathbb{R}^p$ that maximizes the next execution time. However, the flow of a nonlinear system is in general not known, and hence no closed form expression for the execution times is available. Even in the case of linear systems, the optimization problem has no easy solution that can be solved online at a low computational cost. To overcome this issue, we derive in the next section formulas that relate the next execution time t_{i+1} (implicitly defined in (II.6)) to the current state $\mathrm{x}(t_i)$ and input u. Once we have such formulas at our disposal, the optimal value of u^* will be the one maximizing the inter-execution times. In the next section we explain how to obtain formulas for t_{i+1} without explicit computation of the flow of the control system.

III. Computation of execution times

The time instants t_i at which the control input is updated are implicitly defined by inequality (II.6), in the spirit of event-triggered control [ÅB02]: the control law is executed when *something* relevant occurs. Our first step is to develop formulas that denote *explicitly* the relation between the sequence of times t_i and the control input u. We start studying the simpler problem where no input is present. This problem has been solved by the authors in ([AT09], [Tab07]), in the context of self-triggered control. We summarize here the main results. We consider a dynamical system of the form $\dot{\mathrm{x}} = f(\mathrm{x})$, and the goal is to find the time instants t_i at which a given condition $\phi(\mathrm{x}(t_i)) = 0$ is satisfied. Notice the similarities with problem 1, where $\phi(\mathrm{x}(t)) = \dot{V}(\mathrm{x}(t)) - h(\mathrm{x}(t))$. We provide two explicit bounds for such time instants:

Theorem 3.1: [AT09], [AT10] Given a dynamical system $\dot{\mathrm{x}} = f(\mathrm{x})$ with initial condition x_0, the time instant T at which $\phi(\mathrm{x}(T)) = 0$ is lower bounded by the following two expressions:

1)
$$T \geq \tau_1 = \left(\frac{|x_0|}{r}\right)^{-d} \tau_* \tag{III.1}$$

   where $d > 0$, $r > 0$ and $\tau_* > 0$ are parameters that are computed from $f(x)$ and $\phi(x)$.

2)
$$T \geq \tau_2 = \lambda t_* \tag{III.2}$$

where λ satisfies:

$$\sum_{i=0}^{n-1} \alpha_i (\mathcal{L}_f^i \phi)(x_0)\lambda^i = 0 \tag{III.3}$$

with α_i and t_* being coefficients that are computed from $f(x)$ and $\phi(x)$.

The parameter τ_* in (III.1) represents a lower bound for the time T on a small ball of radius r around the origin, while d defines how the times vary as the initial condition moves away from the origin. The design parameter $n > 1$ in (III.3) represents the tradeoff between the accuracy of the expression for τ_2 and its computational complexity (high values for n imply times τ_2 closer to T, but at the cost of a more involved algebraic equation). The coefficients α_i in (III.3) represent a bound for the dynamics of ϕ on a compact set. We refer the reader to ([AT09], [AT10]) for a deeper explanation of the roles played by d and α_i in equations (III.1) and (III.3). These papers also describe how to compute such parameters from the expression of $f(x)$ and $\phi(x)$. Notice that both (III.1) and (III.2) represent lower bounds for the time instants t_i. In general, equation (III.2) represents a tighter bound but at a higher computational cost, since equation (III.3) needs to be solved online in order to compute τ_2.

Example 3.2: Consider a basic model for tumor cell growth [AWLL03]:

$$\dot{\mathrm{x}} = \mathrm{x} - 0.01\mathrm{x}^2, \quad \mathrm{x}(t) \in \mathbb{R}_0^+$$

For a given initial number of cells $x_0 < L$, we want to know the time instant T when the number of tumor cells reaches L. In the framework of Theorem 3.1, the condition is $\phi(\mathrm{x}(t)) = \mathrm{x}(t) - L$. For $L = 30$, equation (III.1) becomes:

$$T \geq \tau_1 = \frac{1.7}{\sqrt{x_0^2 + 1}}$$

And equation (III.2) for $n = 2$ and $n = 3$ is given by:

$$T \geq \tau_2\big|_{n=2} = -0.17\frac{x_0 - 30}{x_0 - 0.01x_0^2}$$

$$T \geq \tau_2\big|_{n=3} = \frac{-0.30}{1 - \frac{1}{50}x_0} + \frac{\sqrt{-1.051x_0^2 + 0.019x_0^3 - 0.0001x_0^4 + 18.03x_0}}{(1 - \frac{1}{50}x_0)(x_0 - \frac{1}{100}x_0^2)}$$

Figure 1 depicts the evolution of times according to the different formulas and shows the tradeoff between complexity of the expression for the bound of τ and accuracy of the times. The exact times were computed via numerical simulations.

This set of formulas was developed for dynamical systems where no input was present. However, it is straightforward to generalize the formulas for the case when the system has inputs. Since u is assumed to be held constant during two consecutive updates (zero-order hold), we can extend the state space with u as a new state. The behaviour between two consecutive executions is governed by:

$$\begin{bmatrix} \dot{\mathrm{x}} \\ \dot{\mathrm{u}} \end{bmatrix} = \begin{bmatrix} f(\mathrm{x}, \mathrm{u}) \\ 0 \end{bmatrix}$$

with initial conditions $\mathrm{x}(t_i)$ and $\mathrm{u}(t_i)$. Hence, with this extended state space representation, formulas (III.1) and (III.2) relate the time instants τ_{i+1} with the control input $\mathrm{u}(t_i)$ and

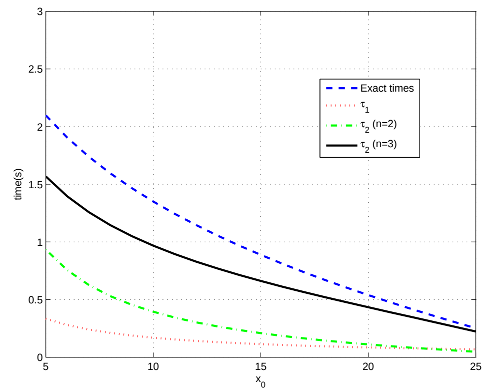

Fig. 1. Computation of times according to the different bounds.

the last measurement of the state of the plant $\mathrm{x}(t_i)$. Thus, the control input u^* maximizing τ_i will reduce the attention required by a system, as explained in the next section.

IV. Minimum attention control

The optimal input value that minimizes attention is given by the expression in (II.5). Since we do not have the exact relation between t_i and u, we resort instead to the approximate formulas (III.1) and (III.2). In this section we describe how to compute the input values that maximize the approximate times τ (while achieving the desired performance, $\dot{V} \leq h(x)$). Such input values $\tilde{u}^*$ represent a suboptimal solution to the problem of minimum attention. We start analyzing the first approximate formula (III.1).

Proposition 4.1: Consider a control system $\dot{\mathrm{x}} = f(\mathrm{x}, \mathrm{u})$ with initial condition $\mathrm{x}(t_i)$. The input value maximizing τ_1, as defined in (III.1), that satisfies the desired performance $\dot{V} \leq h(x)$ for $t \in [t_i, t_{i+1}[$ corresponds to the solution of the following problem:

$$\tilde{u}_1^* = \arg\min_u |u| \tag{IV.1}$$

$$\text{subject to } \left.\frac{\partial V}{\partial x}\right|_{x=\mathrm{x}(t_i)} f(\mathrm{x}(t_i), u) < h(\mathrm{x}(t_i)) \tag{IV.2}$$

Proof: Including the input as a state variable, equation (III.1) turns into:

$$\tau_1 = \left(\frac{\sqrt{|\mathrm{x}(t_i)|^2 + |u|^2}}{r}\right)^{-d} t_* \tag{IV.3}$$

Since the parameter d is greater than 0, the minimum value of $|u|$ maximizes τ regardless of the value of x. If such input guarantees the desired performance at t_i, by continuity we have:

$$\left.\frac{\partial V}{\partial x}\right|_{x=\mathrm{x}(t)} f(\mathrm{x}(t), u) < h(\mathrm{x}(t)) \quad \forall t \in [t_i, t_{i+1}[$$

since t_{i+1} is defined as the time $t > t_i$ where $\dot{V} = h(x)$ is satisfied for the first time. ■

Since the set defined by the constraint (IV.2) is open, a solution to the optimization problem might not exist if there is no maximum in this set. To guarantee the existence of solutions, we modify the open constraint with an tighter closed constraint:

$$\tilde{u}_1^* = \arg\min_u |u| \tag{IV.4}$$

$$\text{subject to } \left.\frac{\partial V}{\partial x}\right|_{x=\mathrm{x}(t_i)} f(\mathrm{x}(t_i), u) \le h(\mathrm{x}(t_i)) - \epsilon \tag{IV.5}$$

where $\epsilon > 0$ is as small as the implementation constraints permit. For the case of input-affine control systems, it is straightforward to solve the problem given in (IV.4), (IV.5) as summarized in the following proposition.

Proposition 4.2: Consider a control system of the form $\dot{\mathrm{x}} = f(\mathrm{x}) + G(\mathrm{x})\mathrm{u}$, $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$. The solution for the optimization problem posed in (IV.4), (IV.5) is given by:

$$\tilde{u}_1^* = \begin{cases} 0 & \text{if } \frac{\partial V}{\partial x} f(x) \le h(x) - \epsilon \\ -\frac{(\frac{\partial V}{\partial x} f(x) - h(x) + \epsilon) \frac{\partial V}{\partial x} G(x)}{|\frac{\partial V}{\partial x} G(x)|^2} & \text{if } \frac{\partial V}{\partial x} f(x) > h(x) - \epsilon \end{cases} \tag{IV.6}$$

Proof: If $\frac{\partial V}{\partial x} f(x) \le h(x) - \epsilon$ the constraint (IV.5) is not active and therefore the optimal $\tilde{u}^*$ is 0. If $\frac{\partial V}{\partial x} f(x) \ge h(x) - \epsilon$, $u = 0$ does not stabilize the system, and the stabilizing input with the minimum norm is in the boundary of the search space, given by the constraint (IV.5). Notice that (IV.6) is well defined since $\frac{\partial V}{\partial x} G(x) = 0$ and $\frac{\partial V}{\partial x} f(x) > h(x) - \epsilon$ would imply that the desired performance of the control system is not achievable, failing to fulfill assumption (II.4). ■

Regarding the second formula (III.2), the maximum value for τ_2 corresponds to the maximum value for λ. Hence the input value $\tilde{u}_2^*$ will be the solution of the following maximization problem:

$$\tilde{u}_2^* = \arg\min_u \lambda \tag{IV.7}$$

$$\text{subject to} \quad \phi(\mathrm{x}(t_i), u) \le 0 \tag{IV.8}$$

$$\sum_{i=0}^{n-1} \alpha_i (\mathcal{L}_f^i \phi)(\mathrm{x}(t_i), u) \lambda^i = 0 \tag{IV.9}$$

$$\text{where} \quad \phi(x, u) = \frac{\partial V}{\partial x} f(x, u) - h(x) - \epsilon$$

As before, ϵ is to be picked as small as the implementation constraints permit. Again, the design parameter n represents the tradeoff between accuracy and complex. For small values of n, the optimization problem is easy to solve in an analytical way, as shown in the example below.

Example 4.3: We extend the model in Example 3.2 to include an input:

$$\dot{\mathrm{x}} = \mathrm{x} - 0.01\mathrm{x}^2 + \mathrm{u}$$

The objective is to design the control input so that inter-execution times are enlarged, while preserving stability for the system. A Lyapunov function for the system is $V = \frac{1}{2}x^2$. The desired rate of decay of the Lyapunov function is given by $h(x) = -x^2$. We compute the input value u^* enlarging the inter-execution times given by formulas (III.1) and (III.2). Using (III.1), we obtain the following expression for u:

$$\tilde{\mathrm{u}}_1^*(t_i) = \begin{cases} 0 & \text{if } \mathrm{x}(t_i) > 200 \\ -2\mathrm{x}(t_i) + \frac{1}{100}\mathrm{x}(t_i)^2 - \epsilon & \text{if } 0 \le \mathrm{x}(t_i) \le 200 \end{cases}$$

And using (III.2), for $n = 2$ we obtain:

$$\tilde{\mathrm{u}}_2^*(t_i) = \begin{cases} -4\mathrm{x}(t_i) + \frac{3}{100}\mathrm{x}(t_i)^2 & \text{if } 0 \le \mathrm{x}(t_i) \le 100 \\ -2\mathrm{x}(t_i) + \frac{1}{100}\mathrm{x}(t_i)^2 - \sqrt{-2 + \frac{1}{50}\mathrm{x}(t_i)} & \\ & \text{if } \mathrm{x}(t_i) > 100 \end{cases}$$

In this case, it can be seen through simulations that the control input $\tilde{u}_1^*(t_i)$ leads to a better approximation of the optimal control input. However, it is unclear to the authors how to determine in general which model leads to better approximations given the structure of the control system and the desired performance.

Remark 4.4: In general, the optimization problem in (IV.7) might not have a solution, since there might be values of u that let the system run in open loop for all $t > t_i$. Indeed, the system might have an open loop stable manifold, where no input updates are needed to steer the system to the equilibrium point. However, from a practical point of view, a bound on the open loop operating time is needed to guarantee robustness on the system [MJT09]. Furthermore, the solution to (IV.7) might not be unique, especially for the case of multiple-input control systems.

V. ANYTIME CONTROL

In this section we tackle the second problem specified in section II-B: given an initial condition $\mathrm{x}(t_i)$ and a certain time interval δ_i, find the control input $\mathrm{u}(t_i)$ that allows the system run open loop until $t_{i+1} = t_i + \delta_i$. This problem naturally arises for instance in embedded systems, where a microprocessor is shared by different tasks and computation is a scarce resource, or in the case of shared actuators, where an actuator is available to the control loop at given instants of time. A related question was addressed in [FGB08] and [Gup09], where the focus was on the design of control laws whose computation times vary according to the available resources. In our case, times at which the control law is executed vary over time, as dictated by a scheduler.

To address this problem, we resort again to the formulas in Theorem 3.1 in order to derive expressions for the input value. In this case, the execution times are given and the equations (III.1) and (III.2) need to be solved to obtain the admissible control inputs. Since the formulas (III.1) and (III.2) represent approximations to the exact inter-execution times, the set of admissible inputs obtained through these formulas are approximations of the exact set of admissible inputs Ω.

For the first equation (III.1), we can derive the set of input values Ω_1 that let the system run in open loop for δ_i units of time:

$$\Omega_1 = \Bigg\{ \mathrm{u}(t_i) \in \mathbb{R}^p \Big| \mathrm{u}(t_i)^T \mathrm{u}(t_i) \le -\mathrm{x}(t_i)^T \mathrm{x}(t_i) + r^2 \Big(\frac{t_*}{\delta_i}\Big)^{\frac{d}{2}} \wedge \left.\frac{\partial V}{\partial x}\right|_{\mathrm{x}(t_i)} f(\mathrm{x}(t_i), \mathrm{u}(t_i)) < h(\mathrm{x}(t_i)) \Bigg\} \tag{V.1}$$

Hence, it can be guaranteed that any input in this set satisfies the desired performance:

$$\mathrm{u}(t_i) \in \Omega_1 \Rightarrow \dot{V}(t) < h(\mathrm{x}(t)) \quad \forall t \in [t_i, t_{i+1}[$$

The first inequality shows that $|\mathrm{u}(t_i)|$ is inversely proportional to the time δ_i. That is, the longer the system needs to run in open loop, the smaller $|\mathrm{u}(t_i)|$ needs to be. The second inequality guarantees that the input guarantees the desired performance during the interval. Likewise, in the case of equation (III.2) we obtain:

$$\Omega_2 = \Bigg\{ \mathrm{u}(t_i) \in \mathbb{R}^p \Big| \sum_{i=0}^{n-1} \alpha_i (\mathcal{L}_f^i \phi)(\mathrm{x}(t_i), \mathrm{u}(t_i)) \Big(\frac{t_*}{\delta_i}\Big)^i \le 0$$

$$\wedge \left.\frac{\partial V}{\partial x}\right|_{\mathrm{x}(t_i)} f(\mathrm{x}(t_i), \mathrm{u}(t_i)) < h(\mathrm{x}(t_i))\Bigg\} \qquad \text{(V.2)}$$

As before, it can be guaranteed that any input in this set satisfies the desired performance:

$$\mathrm{u}(t_i) \in \Omega_2 \Rightarrow \dot{V}(t) < h(\mathrm{x}(t)) \quad \forall t \in [t_i, t_{i+1}[$$

The choice of a particular approximation Ω_1 or Ω_2 should be done on a case by case basis according to the available resources and the required accuracy. Notice that, since both sets are just approximations, the sets might be empty even if there exist inputs that achieve the desired performance.

Example 5.1: Consider again the model in Example (4.3). Using expression (V.1), the set of inputs is defined by:

$$\begin{aligned}\Omega_1 \quad = \quad & \Big\{\mathrm{u}(t_i) \in \mathbb{R} \big| \mathrm{u}(t_i)^2 \leq \frac{9.1 \cdot 10^{-4}}{\delta_i^2} - (\mathrm{x}(t_i)^2 + 1) \\ & \wedge\ \mathrm{u}(t_i) < 0.01\mathrm{x}(t_i)^2 - \mathrm{x}(t_i)\Big\}\end{aligned}$$

Using expression (V.2), for $n = 2$, the set of inputs is given by:

$$\begin{aligned}\Omega_2 = \{&\mathrm{u}(t_i) \in \mathbb{R} \big| \delta_i u^2 + (\delta_i(-1/25\mathrm{x}(t_i)^2 + 5\mathrm{x}(t_i)) + \\ & \mathrm{x}(t_i))u + \delta_i(4\mathrm{x}(t_i)^2 - 7/100\mathrm{x}(t_i)^3 + 3/100^2\mathrm{x}(t_i)^4) \\ & +2\mathrm{x}(t_i)^2 - 1/100\mathrm{x}(t_i)^3 < 0 \\ & \wedge\ \mathrm{u}(t_i) < 0.01\mathrm{x}(t_i)^2 - \mathrm{x}(t_i)\}\end{aligned}$$

In this case, the set Ω_2 represents a better approximation of the exact set guaranteeing an open loop operation for δ_i units of time, but it implies a higher computational cost.

VI. Example

In order to compare the different solutions, we consider the control of a magnetic suspension system [TOS97] to illustrate the previous analysis. The state space representation of the system is:

$$\begin{aligned}\dot{\mathrm{x}}_1 &= \mathrm{x}_2 \\ \dot{\mathrm{x}}_2 &= -\frac{C}{M}\left(\frac{\mathrm{u}}{\mathrm{x}_1}\right)^2 + g_e - \frac{F_d}{M}\end{aligned} \qquad \text{(VI.1)}$$

where x_1 stands for the position of the plunger, x_2 is its speed, and the parameters take the following values: $M = 67, C = 4.43 \cdot 10^{-4}$, $g_e = 9.81 m/s^2$, and $F_d = \frac{1}{2} M g_e$. The coil current u is assumed to be the control input. The control objective is to stabilize the position of the plunger at $x_1^{\mathrm{ref}} = 0.1$m. A control law is to be designed in order to enlarge the inter-execution times while satisfying the performance requirement. For simplicity, we consider $h(x) = 0$, *i.e.*, the performance requirement is $\dot{V} \leq 0$. We will compare the proposed control strategies in Section IV against a standard control law design, given by Sontag's formula [Son89]:

$$\tilde{u}^* = \begin{cases} -\frac{\frac{\partial V}{\partial x} f + \sqrt{(\frac{\partial V}{\partial x} f)^2 + (\frac{\partial V}{\partial x} g)^4}}{\frac{\partial V}{\partial x} g} & \text{if } \frac{\partial V}{\partial x} g \neq 0 \\ 0 & \text{if } \frac{\partial V}{\partial x} g = 0 \end{cases} \qquad \text{(VI.2)}$$

A control Lyapunov function for this system is:

$$V = \frac{1}{2}\big((x_1 - x_1^{\mathrm{ref}} - x_2)^2 + (x_1 - x_1^{\mathrm{ref}})^2 + x_2^2\big) \qquad \text{(VI.3)}$$

Optimal	(VI.5)	(VI.6)	(VI.7)	Sontag's formula [Son89]
22	44	26	25	83

TABLE I

AVERAGE NUMBER OF EXECUTIONS FOR A SIMULATION TIME OF 2S.

In order to obtain formulas relating the inter-execution times and the input, we extend the state space model:

$$\begin{bmatrix} \dot{\mathrm{x}}_1 \\ \dot{\mathrm{x}}_2 \\ \dot{\mathrm{u}} \end{bmatrix} = f(\mathrm{x}, \mathrm{u}) = \begin{bmatrix} \mathrm{x}_2 \\ -\frac{C}{M}\left(\frac{\mathrm{u}}{\mathrm{x}_1}\right)^2 + g_e - \frac{F_d}{M} \\ 0 \end{bmatrix} \qquad \text{(VI.4)}$$

The formulas (III.1) and (III.2) relating time and input for a given state take the following values for this example:

$$\tau_1 = \frac{0.034}{\sqrt{\mathrm{x}_1(t_i)^2 + \mathrm{x}_2(t_i)^2 + \mathrm{u}(t_i)^2 + 10^{-4}}} \qquad \text{(VI.5)}$$

$$\tau_2\big|_{n=2} = -1.092 \frac{\dot{V}(\mathrm{x}(t_i), \mathrm{u}(t_i))}{(\mathcal{L}_f(\dot{V}))(\mathrm{x}(t_i), \mathrm{u}(t_i))} \qquad \text{(VI.6)}$$

$$\begin{gathered}\tau_2\big|_{n=3} = 10^{-3}\lambda \qquad \text{(VI.7)} \\ \textstyle\sum_{i=0}^{2} \alpha_i(\mathcal{L}_f^i(\dot{V}))(\mathrm{x}(t_i), \mathrm{u}(t_i))\lambda^i = 0, \\ \alpha_0 = 0.999 \quad \alpha_1 = 9.999 \cdot 10^{-4} \quad \alpha_2 = 5.686 \cdot 10^{-7}\end{gathered}$$

For all the control laws, the time instants at which the control inputs are updated are given by the following event:

$$t_{i+1} = \min\{t > t_i : \dot{V}(t) = 0\}$$

To compare the behaviour of the different techniques, several random initial conditions were considered. Table I depicts the average number of executions along trajectories for: the exact minimum attention strategy (computed numerically); the control laws derived from (VI.5), (VI.6), (VI.7); and Sontag's formula (VI.2), with V as defined in (VI.3). In this example we can see that both (VI.6) and (VI.7) generate approximately the same number of executions, hence there is barely any improvement by considering a more complex expression in this case. We can also observe how Sontag's formula requires at least 2 times as many executions as any of the proposed techniques in Section IV. The evolution of the Lyapunov function and the state trajectories are displayed in Figures 2 and 3 for a particular initial condition. Figure 2 includes as well the number of executions that each technique requires during this simulation time of 2 seconds. For this particular initial condition, both the models (VI.6) and (VI.7) generate the same number of executions. We can see that all of them meet the desired performance $\dot{V} \leq 0$ but at very different rates of decay. The model (VI.5) generates nearly the same number of executions as the optimal policy, albeit the decay of the Lyapunov function is much slower. In this sense, the control law derived from (VI.7) outperforms the controller obtained from Sontag's formula, since they achieve a faster rate of decay even with a smaller number of executions.

We also apply the results in Section V to this example. We assume that the magnetic suspension controller is being executed in a shared microprocessor, and inter-execution

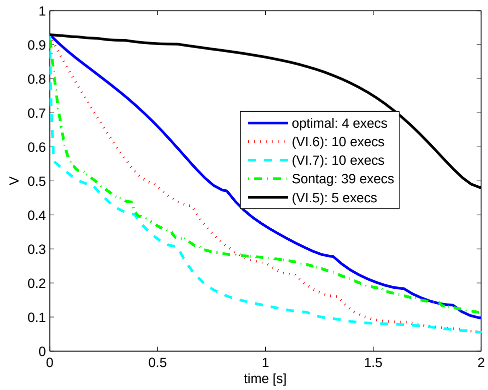

Fig. 2. Evolution of the Lyapunov function under the different minimum attention strategies.

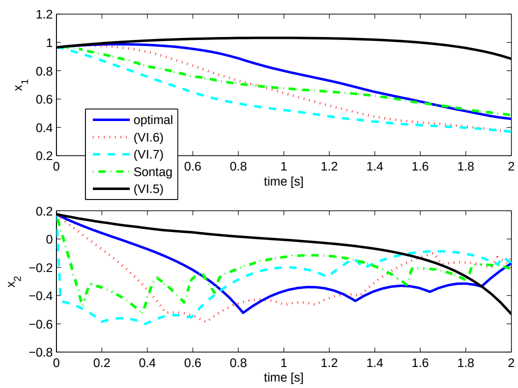

Fig. 3. Trajectories under the different minimum attention strategies.

times allocated to the different tasks vary as dictated by a scheduler at running time. Figure 4 shows the evolution of the Lyapunov function under the strategies derived in Section V and Sontag's formula as the execution times vary. The initial condition of the system is $x_0 = (0.6, -0.3)$. The dashed vertical lines mark the execution times. At every execution point t_i, the state is measured and the set Ω_2 (given by (V.2)) is computed, with $\delta = t_{i+1} - t_i$. For the anytime attention technique, the input to the control system is randomly selected from Ω_2. Figure 4 shows how the anytime technique is able to select the inputs in order to satisfy the desired requirement $\dot{V} \leq 0$, while the control law derived from Sontag's formula violates $\dot{V} \leq 0$ as the inter-execution time varies. In this example, due to its conservativeness, the technique defined in (V.1) leads to an empty set Ω_1, implying that there is no controller stabilizing the system with the given resources. Therefore, the more complex solution (V.2) is needed in this example to generate control inputs stabilizing the system for the allocated resources.

VII. Discussion

In this paper we have presented preliminary results on attention-aware control laws for nonlinear systems. Regarding the minimum attention control strategy, the control input was chosen in order to enlarge the next open loop operation time. This strategy represents a greedy approach, since a control input might steer the system to a region in the state space where more executions are needed to stabilize the system. A more long-term strategy would generate inputs

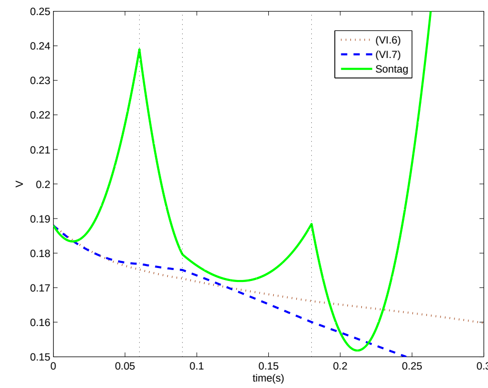

Fig. 4. Evolution of the Lyapunov function under the anytime strategies.

to steer the system to *slow* manifolds, *i.e.*, regions where the system requires little or no attention.

This analysis has also a direct application for real-time scheduling co-design problems. The derived formulas show the fundamental limitations imposed by the dynamics of the system to be controlled. Enough computational resources need to be allocated to a system so that the set Ω computed in section V is nonempty. Otherwise, there might not exist any controller that can stabilize the system with the preassigned resources. Thus, the set of formulas provided in this paper suggests one alternative approach to the minimum-bit rate problem (in the spirit of [NFZE07]) for the case of nonperiodic implementations.

References

[ÅB02] K.J. Åström and B.M. Bernhardsson. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. *41st IEEE Conference on Decision and Control*, 2, 2002.

[Årz99] K.E. Årzén. A simple event-based PID controller. *Preprints 14th World Congress of IFAC. Beijing, PR China*, 1999.

[AT09] A. Anta and P. Tabuada. Isochronous manifolds in self-triggered control. *48th IEEE Conference on Decision and Control*, 2009.

[AT10] A. Anta and P. Tabuada. To sample or not to sample: Self-triggered control for nonlinear systems. *To appear in IEEE Transactions on Automatic Control. arXiv:0806.0709*, 2010.

[AWLL03] B.Q. Ai, X.J. Wang, G.T. Liu, and L.G. Liu. Correlated noise in a logistic growth model. *Physical Review E*, 2003.

[Bro97] R.W. Brockett. Minimum attention control. In *36th IEEE Conference on Decision and Control*, volume 3, 1997.

[Bro01] R.W. Brockett. New issues in the mathematics of control. *Mathematics unlimited: 2001 and beyond*, 2001.

[Cha07] D. Chakraborty. *Need-based feedback: An optimization approach*. PhD thesis, University of Florida, Gainesville, 2007.

[FGB08] D. Fontanelli, L. Greco, and A. Bicchi. Anytime control algorithms for embedded real-time systems. *Hybrid Systems: Computation and Control*, 4981:158–171, 2008.

[Gup09] V. Gupta. On an Anytime Algorithm for Control. *48th IEEE Conference on Decision and Control*, 2009.

[MJT09] M. Mazo Jr and P. Tabuada. On the robustness of self-triggered control for sensor/actuator networks. *48th IEEE Conference on Decision and Control*, 2009.

[NFZE07] G.N. Nair, F. Fagnani, S. Zampieri, and R.J. Evans. Feedback control under data rate constraints: an overview. *Proceedings of the IEEE*, 95(1):108, 2007.

[SLSS96] D. Seto, J.P. Lehoczky, L. Sha, and K.G. Shin. On task schedulability in real-time control systems. *17th IEEE Real-Time Systems Symposium*, pages 13–21, 1996.

[Son89] E.D. Sontag. A 'universal' construction of Artstein's theorem on nonlinear stabilization. *Syst. Control Lett.*, 1989.

[Tab07] P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE TAC*, 52(9):1680–1685, 2007.

[TOS97] D.L. Trumper, S.M. Olson, and P.K. Subrahmanyan. Linearizing control of magnetic suspension systems. *IEEE Transactions on Control Systems Technology*, 5(4):427–438, 1997.

[VFM03] M. Velasco, J. Fuertes, and P. Marti. The self triggered task model for real-time control systems. *24th IEEE Real-Time Systems Symposium (work in progress)*, pages 67–70, 2003.