A PROBABILISTIC APPROACH TO THE STABILITY ANALYSIS OF REAL-TIME CONTROL SYSTEMS ⋆

Manel Velasco, Pau Martí, Ricard Villà, Josep M. Fuertes, Jordi Ayza, Miquel Monroig

Department of Automatic Control and Computer Engineering
Technical University of Catalonia
Pau Gargallo 5, 08028 Barcelona, Spain,
{manel.velasco, pau.marti, ricard.villa, josep.m.fuertes, jordi.ayza}@upc.edu

Abstract: In real-time multitasking systems, feasible periodic tasks execute within their periods. However, the exact time at which each task executes vary due to other task interferences. For control tasks, this produces irregular sampling and varying time delays, which may degrade system performance and bring the system to instability. In this paper we present stability conditions that allow us to evaluate if a closed-loop system designed to work at a nominal sampling period h_d with a nominal time delay τ_d will remain stable if the run-time sampling is not strictly periodic and time delays vary randomly. In the closed-loop system model, we consider the irregular sampling and the varying time delays as random variables with known expectation. With this model, the system evolution can be seen as a sequence of random state vectors generated by the system closed-loop matrix. Then, we derive stability conditions for the system in terms of convergence of a sequence of random variables.

Keywords: Real-time systems, Timing jitter, Random variables, Probabilistic models, Stability tests

1. INTRODUCTION

In real-time multitasking systems, feasible periodic tasks execute within their periods. However, the exact time at which each job executes vary due to other task interferences. That is, real-time systems allow jitter in task executions as far as feasibility constraints are met.

In such systems, controllers are often implemented as periodic tasks, where at each job execution, sampling, control algorithm computation and actuation are sequentially performed. In this context, variability in jobs execution produces sampling and latency jitter (Arzen *et al.*, 2000), which may degrade control system performance and even bring the system to instability (Martí *et al.*, 2001*b*).

Recently, several works combining control and scheduling co-design approaches have focused on the jitter problem (e.g., (Cervin, 1999) and (Martí *et al.*, 2001*a*)). Their main goal has been to minimize the degrading effects that jitter in control tasks introduces in the performance of control systems. This is achieved by computing and switching controllers according to the run-time jitters. In addition, several tools have been presented for simulation and performance analysis of real-time control systems (e.g. (Henriksson *et al.*, 2002) and (Lincoln and Cervin, 2002)). However, none of the previous works focused on control systems stability (although stability issues were considered).

⋆ This research has received support from the Spanish Ministerio de Ciencia y Tecnologia Project ref. DPI2002-01621.

Looking at stability and the jitter problem for control tasks, in (Martí *et al.*, 2001*c*), a sufficient stability condition was presented for analysing closed-loop systems where control tasks subject to jitter adapt their gains at run time (i.e., switching controllers). Necessary and sufficient stability conditions that can be also applied in these scenarios can be found in (Dogruel and Özgüner, 1995) or (Liberzon *et al.*, 1999). However, the application of these conditions requires previous knowledge of the exact jitter values that each control task will be subject to at run time. And for some application scenarios, this may not be known previously or could be impossible to predict due to the dynamics of the real-time multitasking system. In these cases, the application of these stability criteria fails and new criteria are required.

The application of the stability criteria we present does not require knowing the exact jitter values, but their distribution. Our approach is based on modelling the irregular sampling and varying time delay as random variables with known expectation. With this model, the evolution of the system can be seen as a sequence of random state vectors generated by the system closed-loop matrix. Then, we derive stability conditions in terms of convergence of a sequence of random variables. Results are illustrated using simulated examples.

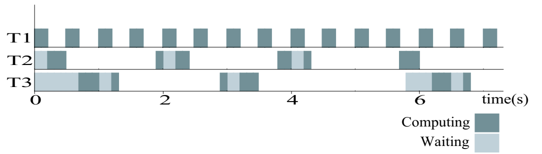
In addition, the stability criteria we present do not assume switching controllers at run-time. We let the system run a single discrete-time controller designed assuming a constant sampling period h_d and a constant (or zero) time delay τ_d. The stability tests we present can be used to analyse whether the closed-loop system will remain stable if the run-time sampling is not strictly periodic and time delays are not constant.

The rest of this paper is organized as follows. In Section 2 we review the jitter problem. In Section 3 we define the stability problem in terms of convergence of the sequence of random state vectors. Section 4 presents two criteria to test closed-loop system stability. Finally, we conclude and discuss future work in Section 5.

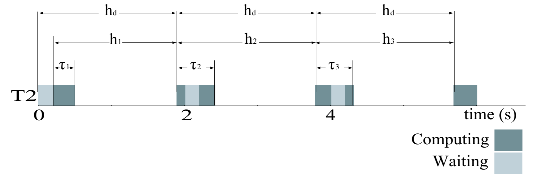
2. THE JITTER PROBLEM

In this section we briefly review the jitter problem that may arise in real-time multitasking systems, which may result in random sampling and varying time delays for control systems.

In real-time scheduling, controllers are usually implemented using the periodic task model. A periodic task is seen as a successive execution of jobs. The k^{th} job of a feasible periodic task fulfils the following constraints: it has to execute within its period, which starts at $(k-1)T$ and finishes at kT (where T is the task period), and has to complete before or at time $(k-1)T+D$, where D is its relative deadline (provided $C \leq D \leq T$, where C is the task worst-case execution time). This means that each job will start and finish its execution within an interval of D time units (usually $D=T$), but no assurances can be made on the exact start and completion time of each job execution because of the interference of other tasks executions.



(a) Three tasks partial EDF schedule



(b) Jitters on *task 2* execution

Fig. 1. Jitters on tasks jobs executions

A common way of coding classically designed controllers using the periodic task model is to set each control task period T equal to the sampling period h_d used in the controller design stage (with $D=T$). Sampling and actuation are assumed to take place when each job starts and completes its execution respectively. With this assumption, the allowed variability in jobs executions results in random sampling and varying time delays. This is problematic because control actions are calculated with respect to the assumptions on regular sampling and constant time delay that were made in the controller design stage.

Example 1 illustrates the jitter phenomena of real-time multitasking systems and its influence on the control signal for a generic controller.

***Example** 1.* Let us consider a real-time multitasking system with three control tasks, as specified in Table 1 (where periods P, deadlines D and worst-case computation times C are given in seconds).

Table 1. Task set.

	P	D	C
task 1	0.5	0.5	0.2
task 2	1.9	1.9	0.3
task 3	2.9	2.9	0.4

Figure 1 (a) shows a partial feasible schedule of the three tasks (during 7s, approximately) if the task set is scheduled using the optimal priority-based scheduling algorithm *earliest deadline first* (EDF) (Liu and Layland, 1973). For each task, *dark grey* symbolizes jobs executions, which may be blocked (symbolized in *light grey*) due to the interference of other tasks executions. These interferences, which are allowed as far as schedulability constraints are satisfied, cause jitter in jobs execution.

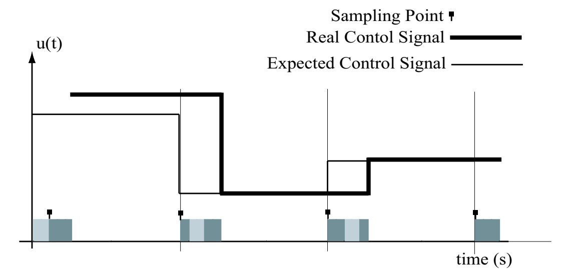


Fig. 2. *Task 2* expected and real control signal

Looking for example at the sequence of jobs of *task 2* (Figure 1 (b)), we have that the time interval between consecutive jobs start execution times, i.e., sampling period, takes different values (for example $h_1 = 1.7s$ and $h_2 = 1.9s$). And the time elapsed from each job start execution time to its completion time, i.e., latency or delay, also varies (for example $\tau_1 = 0.3s$ and $\tau_2 = 0.5s$). In fact, the task was assigned a period of $h_d = 1.9s$ (because this value was used in the controller design stage), but at run-time, the *real* sampling period varies in the vicinity of $1.9s$, taking values at $1.7s$, $1.9s$ and $2s$. In addition, in the controller design a delay of $0.3s$ was accounted for, but at run-time, the *real* latency varies, taking values of $0.3s$ but also of $0.5s$.

Figure 2 shows the evolution of the control signal for each job execution. In fact, it shows the evolution of the expected control signal (thin line, corresponding to a classically designed controller executing in isolation, that is, without jitters) and the *real* control signal (thick line) for the same controller if implemented in *task 2* in the multitasking real-time system .

In Figure 2, let us suppose for example that a perturbation affects the system controlled by *task 2* just before time 0. The first sampling that would detect that the controlled system is not in equilibrium (due to the perturbation) should occur at time 0. But due to a start time delay, sampling occurs $0.2s$ later. Therefore, the sampling will read a greater value (corresponding to a greater deviation of the controlled system with respect to its equilibrium point) than it should. Consequently the real control signal will be stronger (higher valued) than the expected one, because it will try to correct a greater deviation. Therefore, the evolution of the controlled system will not be the expected one. We don't know if the system will remain stable as it would do if the controller would execute in isolation, without suffering jitters.

By treating these jitters as random variables, this paper presents a new (to the authors knowledge) probabilistic approach to the stability analysis of control systems subject to sampling and latency jitter.

3. PROBLEM FORMULATION

In this section we present the system model we use to derive the stability criteria. We then define the problem to be analysed for the stability analysis, and without losing generality, we reduce the system model that we consider in order to simplify notation.

3.1 System model

A linear discrete-time control system with time delay (where the delay is less than or equal to the sampling period) can be described by equation (1), where x_n is the state vector, u_k and u_{k-1} are the current and past control signals, and $\Phi(h)$, $\Gamma_0(h,\tau)$ and $\Gamma_1(h,\tau)$ are the system and input matrices that depend on the sampling period h and time delay τ (Astrom and Wittenmark, 1997).

$$x_{n+1} = \Phi(h)x_n + \Gamma_0(h,\tau)u_n + \Gamma_1(h,\tau)u_{n-1} \quad (1)$$

Consider that the sampling period h and time delay τ are independent random variables with known expectation but the control signal u is given by a controller designed to work at a constant sampling period h_d with a constant time delay τ_d. The new closed-loop system dynamics can be described by (2),

$$\begin{bmatrix} x_{n+1} \\ u_n \end{bmatrix} = A(h,\tau,h_d,\tau_d) \begin{bmatrix} x_n \\ u_{n-1} \end{bmatrix} \quad (2)$$

where $A(h,\tau,h_d,\tau_d)$ is the closed-loop matrix next specified in (3). In (3), $L = (L(h_d,\tau_d), L_u(h_d,\tau_d))$ is the feedback gain.

$$\begin{bmatrix} \Phi(h)+\Gamma_0(h,\tau)L(h_d,\tau_d) & \Gamma_1(h,\tau)+\Gamma_0(h,\tau)L_u(h_d,\tau_d) \\ IL(h_d,\tau_d) & IL_u(h_d,\tau_d) \end{bmatrix} \quad (3)$$

To clarify the notation, the closed-loop matrix (3) can be further detailed as in (4).

$$\begin{bmatrix} a_{1,1}(h,\tau,h_d,\tau_d) & \dots & a_{1,n}(h,\tau,h_d,\tau_d) \\ \vdots & \ddots & \vdots \\ a_{n,1}(h,\tau,h_d,\tau_d) & \dots & a_{n,n}(h,\tau,h_d,\tau_d) \end{bmatrix} \quad (4)$$

3.2 Problem definition

The closed-loop system given by (2) may become instable because control signals are not appropriate according to the system dynamics. Control signals u are computed according to the feedback gain L, that was designed assuming a constant sampling period h_d and a constant time delay τ_d. However, the run-time system evolution is discretely driven by random variables h and τ , which take unexpected but bounded [1] values at each iteration.

[1] Although the modeling of the jitters is out of the scope of this paper, it is worth to mention that for a given feasible real-time periodic task, the jitter variation is bounded (for further details, see (Martí *et al.*, 2001*a*)).

To focus on the stability of the closed-loop system given by (2), we analyse the convergence of the sequence of state vectors that it generates. To do so we use known concepts of convergence of sequences of random variables (Grimmett and Stirzaker, 2001). Note that looking at (2) we have to study which conditions the generated sequence of random state vectors $x_1,...,x_n$ has to fulfil in order to converge towards a random vector $x = 0$.

Henceforth, convergence will refer to convergence in mean (recall that convergence in mean implies convergence in probability (Grimmett and Stirzaker, 2001)). We say that the sequence X_n converges in mean towards X, if $\mathbf{E}(|X_n|) < \infty$ for all n, and:

$$\lim_{n\to\infty} \mathbf{E}(|X_n - X|) = 0$$

where the operator $\mathbf{E}$ denotes the expectation.

Therefore, to establish stability conditions for the closed-loop system given by (2), we will focus on studying the convergence in mean of the sequence of state vectors x_n towards x. That is,

$$\lim_{n\to\infty} \mathbf{E}(|x_n - x|) = 0$$

In fact, if the equilibrium point is zero (without losing generality), then, we study

$$\lim_{n\to\infty} \mathbf{E}(|x_n|) = 0$$

In order to establish convergence criteria, we shall look at the sequence given by $\mathbf{E}(|x_n|)$. If such sequence converges towards zero, then, the sequence of state vectors will converge in mean. Therefore, the closed loop system specified in (2) will be stable.

We focus on sequences that are given by the expectation of the norm of each state vector. It is then important to point out that the stability criteria we present are given in terms of vector and corresponding (induced) matrix norms satisfying the sub-multiplicative property $\|A_1A_2\| \leq \|A_1\|\|A_2\|$ (such as *p-norms* do) (Lancaster and Tismenetsky, 1985). Henceforth, the operator $\|\cdot\|$ denotes the appropriate norm, and in the examples we will use as vector norm the *Euclidean norm*, with the *spectral norm* as associated matrix norm.

3.3 System Simplification

From now on, for the sake of clarity, we will present our results for a simplified version of (2). Also we will assume that the system we study depends on a single random variable, e.g., the sampling period. Therefore, the system that we study is given by

$$\begin{aligned} x_{n+1} &= \Phi(h)x_n + \Gamma(h)u(h_d) \\ &= (\Phi(h) + \Gamma(h)L(h_d))\,x_n \\ &= A(h,h_d)x_n \end{aligned} \tag{5}$$

where $L = (l_1(h_d),\ldots,l_k(h_d))$ is the feedback gain designed assuming a nominal sampling period h_d, and h is the random variable that describes the runtime sampling. Therefore, the system evolution can be described as

$$\begin{bmatrix} x^1_{n+1} \\ \vdots \\ x^k_{n+1} \end{bmatrix} = \begin{bmatrix} a_{1,1}(h,h_d) & \ldots & a_{1,n}(h,h_d) \\ \vdots & \ddots & \vdots \\ a_{n,1}(h,h_d) & \ldots & a_{n,n}(h,h_d) \end{bmatrix} \begin{bmatrix} x^1_n \\ \vdots \\ x^k_n \end{bmatrix} \tag{6}$$

This will allow us to better explain and illustrate our results. Note that with this simplification, we are not losing generality in the sense that the results we present also apply to the closed-loop system specified in (2) (characterized also by the random variable time delay). This holds because we consider the two random variables to be independent.

In example 2 we introduce the system that we use throughout the paper to illustrate the results we presented.

***Example** 2.* Let us consider the double integrator, process that can be modeled using the following continuous-time state space representation (Astrom and Wittenmark, 1997):

$$\frac{dx}{dt} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{7}$$

Sampling (7) with sampling period h (which we consider a random variable) gives:

$$x_{n+1} = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} x_n + \begin{bmatrix} \frac{h^2}{2} \\ h \end{bmatrix} u_n \tag{8}$$

Let us consider that the input u_n is given by $u_n = L(h_d)x_n$, where the feedback gain L is designed with a nominal sampling period h_d and obtained by pole placement, where $p1$ and $p2$ are the closed-loop poles.

$$L(h_d) = \left[\frac{1-p1-p2+p1p2}{{h_d}^2} \;\; \frac{3-p1-p2-p1p2}{2h_d} \right] \tag{9}$$

For this particular example with poles $(p1,p2) = (0,0.5)$, the close-loop system in (6) is:

$$\begin{bmatrix} x^1_{n+1} \\ x^2_{n+1} \end{bmatrix} = \begin{bmatrix} 1 - \frac{0.25h^2}{h_d^2} & h - \frac{0.625h^2}{h_d} \\ -\frac{0.5h}{h_d^2} & 1 - \frac{1.25h}{h_d} \end{bmatrix} \begin{bmatrix} x^1_n \\ x^2_n \end{bmatrix} \tag{10}$$

Recall that in (10), h is a random variable, whose probability distribution is characterized by a known expectation.

4. STABILITY CRITERIA

Two stability criteria to assess the stability of closed-loop systems modelled as in (5) are presented next.

4.1 First stability criterion

We study the sequence given by $\mathbf{E}(|x_n|)$. We look for conditions on this sequence that ensure its tendency towards zero. If it does, then, $\lim_{n\to\infty}\mathbf{E}(|x_n|)=0$ will hold, implying convergence in mean of the sequence of state vectors x_n, which implies stability of system (5).

Proposition 1. Let $A(h,h_d)$ be the closed-loop system matrix as defined in (5). If $\mathbf{E}(\|A(h,h_d)\|)\leq 1$, then the system $x_{n+1}=A(h,h_d)x_n$ is stable.

Observe the following inequality:

$$\begin{aligned}\mathbf{E}(\|x_{n+1}\|) &= \mathbf{E}(\|A(h,h_d)x_n\|)\\ &\leq \mathbf{E}(\|A(h,h_d)\|\|x_n\|)\\ &= \mathbf{E}(\|A(h,h_d)\|)\mathbf{E}(\|x_n\|)\end{aligned} \tag{11}$$

Therefore, if $\mathbf{E}(\|A(h,h_d)\|)\leq 1$ then $\lim_{n\to\infty}\mathbf{E}(\|X_n\|)=0$, implying system stability. Because proposition 1 is a sufficient (but not necessary) condition, we can not affirm anything about system stability if $\mathbf{E}(\|A(h,h_d)\|)>1$. This is illustrated in the following example.

***Example** 3.* Consider system (10) in example 2. Consider that the random sampling period h is a discrete variable that takes alternatively values $1.7s$, $1.9s$ and $2s$ (as if the controller was implemented in *task2* of example 1). For this example, the probability distribution of the sampling period for *task 2* can be described as follows: the probability of $h=1.7s$ is $f(1.7s)=0.2$, of $h=1.9s$ is $f(1.9s)=0.4$ and of $h=2.0s$ is $f(2.0s)=0.4$. Therefore, $\mathbf{E}(h)=1.7*0.2+1.9*0.4+2*0.4=1.9$. Recall that we will use here the *Euclidean norm* for vectors and the *spectral norm* for matrices. For this configuration, if the state feedback controller is designed with nominal sampling period $h_d=1.9$, then $\mathbf{E}(\|A(h,h_d)\|)=1.09$ (aprox.), which does not satisfy proposition 1. In fact, for any nominal sampling period $h_d\in[1.7s\ldots 3s]$, never holds that $\mathbf{E}(\|A(h,h_d)\|)<1$. This is shown in Fig. 3 (a), where we plot $\mathbf{E}(\|A(h,h_d)\|)$ (vertical axis) as a function of h_d (horizontal axis).

Therefore, for this example, by using this criterion, we can not ensure convergence for any of the tested h_d values, nor system stability. However, using the stability results presented in (Dogruel and Özgüner, 1995), the system we are studying happens to be stable. Note that this system can be viewed as a switching control system that randomly switches between three matrices $A(1.7,1.9)$, $A(1.9,1.9)$ and $A(2,1.9)$, where

$$A(1.7,1.9)=\begin{bmatrix}0.7998614959 & 0.7493421055\\ -0.2354570637 & -0.118421052\end{bmatrix},$$

$$A(1.9,1.9)=\begin{bmatrix}0.7500000000 & 0.712500000\\ -0.2631578947 & -0.250000000\end{bmatrix},$$

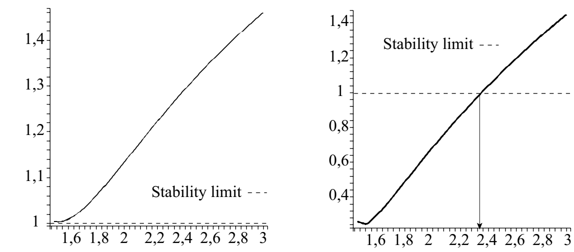


Fig. 3. Application of the stability criteria: (left) $\mathbf{E}(\|A(h,h_d)\|)$ (in vertical axis) vs. h_d (horizontal axis), (right) $\mathbf{E}(\|A(h_2,h_d)A(h_1,h_d)\|)$ (in vertical axis) vs. h_d (horizontal axis)

$$A(2.0,1.9)=\begin{bmatrix}0.7229916898 & 0.684210527\\ -0.2770083102 & -0.315789473\end{bmatrix}$$

For them we can find the following matrix $P>0$

$$P=\begin{bmatrix}46.58723044600783 & 19.81892544766115\\ 19.81892544766115 & 55.51306950193158\end{bmatrix}$$

that satisfies that $\forall A\in\Omega=\{A(1.7,1.9), A(1.9,1.9), A(2,1.9)\}$, $A^TPA-P<0$, which implies that the system is stable according to (Dogruel and Özgüner, 1995).

This example shows that the stability condition given in proposition 1 can be rather conservative. This is because the inequality in (11) $\mathbf{E}(\|A(h,h_d)x_n\|)\leq \mathbf{E}(\|A(h,h_d)\|\|x_n\|)$ may be too inaccurate, in the sense of being a loose upper bound. Note also that in the example, the same results are obtained regardless of which *p-norm* is used.

4.2 Second stability criterion

To improve the accuracy of the previous criterion, in this section we present a less conservative stability condition, which is based on the following observation. If in the system used in example 3 we consider a step further in the inequality (11), that is,

$$\begin{aligned}\mathbf{E}(\|x_{n+2}\|) &= \mathbf{E}(\|A(h_2,h_d)x_{n+1}\|)\\ &= \mathbf{E}(\|A(h_2,h_d)A(h_1,h_d)x_n\|)\\ &\leq \mathbf{E}(\|A(h_2,h_d)A(h_1,h_d)\|\|x_n\|)\\ &= \mathbf{E}(\|A(h_2,h_d)A(h_1,h_d)\|)\mathbf{E}(\|x_n\|),\end{aligned} \tag{12}$$

and plot $\mathbf{E}(\|A(h_2,h_d)A(h_1,h_d)\|)$ (vertical axis) as a function of h_d (horizontal axis), we can see in Fig. 3 (b) that for a set of h_d values less than $2.35s$ (aprox.), the expectation is less than 1. Therefore, for any controller designed with one of these h_d values with $\mathbf{E}(\|A(h_2,h_d)A(h_1,h_d)\|)<1$, the closed-loop system would have remained stable even if the controller would have been affected by the sampling jitter we modeled in the random variable h. Note that for these h_d values, the sequence of state vectors x_n will converge in mean towards zero, that is, system (5) will be stable. In (12), h_1 and h_2 denote two consecutive samples, which we consider to be two independent random variables.

Therefore, with that further step in inequality (11) and adequately applying proposition 1, we have been able to affirm stability for a range of h_d values. This observation leads to the following necessary and sufficient stability condition.

Proposition 2. (*General Criterion*) Let $A(h, h_d)$ be the closed-loop system matrix as defined in (5). Given h_d, the control system is stable *iff* exists $k \in \mathbb{N}$ such as

$$\mathbf{E}(\|A(h_k, h_d)\cdots A(h_2, h_d)A(h_1, h_d)\|) \leq 1$$

Proposition 2 is a direct consequence of the previous observation, which takes advantage of the submultiplicative property of the matrix norm operator. In fact, inequality (12) can be generalized to inequality specified in (13).

$$\mathbf{E}(\|x_{n+k}\|) \leq \mathbf{E}(\|A(h_k, h_d)\cdots A(h_2, h_d)A(h_1, h_d)\|)\mathbf{E}(\|x_n\|) \quad (13)$$

If $\mathbf{E}(\|A(h_k, h_d)\cdots A(h_2, h_d)A(h_1, h_d)\|) \leq 1$, then the sequence of state vectors converges in mean towards the equilibrium point ($\mathbf{E}(\|x_n\|) \to 0$), that is, the closed-loop system in (5) is stable).

It is worth to mention that proposition 2 with $k = 1$ is equivalent to proposition 1. The application of proposition 2 requires to recursively compute the expectation of the norm of a matrix obtained after multiplying k matrices, for $k = 1, 2, \ldots$. And if the system is stable, at some point, the inequality of the proposition will hold.

Proposition 2 can be used in two directions. Given a sampling period distribution, it can be used to find a value or set of values for h_d such that the system is stable (as we described above). Or, it also can be used to find, given a value for h_d, the maximum sampling variability that the closed-loop system admits before going to instability. The later can be used for designing resource management/scheduling techniques capable of dealing with overload conditions in real-time control systems.

5. CONCLUSIONS

We have discussed why in the application scenario of real-time control systems (where controllers are subject to scheduling induced jitters), existing stability criteria may not be applicable. To overcome this applicability problem, in the closed-loop system designed to work with regular sampling and assuming a constant time delay, we have modeled the jitter effects on the sampling and controller latency as bounded random variables. Then, we have derived stability conditions in terms of convergence of the random state vectors generated by this new model. Using this stability test has the advantage of obtaining a closer result to real situations.We have illustrated the application of these conditions using simple examples.

Future work will focus on using the stability conditions for the design of more flexible and adaptive schedulers.

REFERENCES

Arzen, Karl-Erik, Anton Cervin, Johan Eker and Lui Sha (2000). An introduction to control and scheduling co-design. In: *Proceedings of the 39th IEEE Conference of Decision and Control*. Sydney, Australia.

Astrom, Karl J. and Bjorn Wittenmark (1997). *Computer-Controlled Systems. Third Edition*. Prentice–Hall.

Cervin, Anton (1999). Improved scheduling of control tasks. In: *Proceedings of the 11th Euromicro Conference on Real-Time Systems*. York, UK.

Dogruel, M. and U. Özgüner (1995). Stability of a set of matrices: A control theoretic approach. In: *Proceedings of the 34th IEEE Conference of Decision and Control*.

Grimmett, Geoffrey and David Stirzaker (2001). *Probability and Random Processes. Third Edition*. Oxford University Press.

Henriksson, Dan, Anton Cervin and Karl-Erik Arzen (2002). Truetime: Simulation of control loops under shared computer resources. In: *15th IFAC World Congress on Automatic Control*. Barcelona, Spain.

Lancaster, Peter and Miron Tismenetsky (1985). *The Theory of Matrices. Second edition*. Academic Press.

Liberzon, D., J.P. Hespanha and A.S. Morse (1999). Stability of switched systems: a lie-algebraic condition. *Systems Control Letters* **37**, 117–122.

Lincoln, Bo and Anton Cervin (2002). Jitterbug: A tool for analysis of real-time control performance. In: *41st IEEE Conference on Decision and Control*. Las Vegas, NV.

Liu, C. L. and James W. Layland (1973). Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the Association for Computing Machinery* **20**(1), 46–61.

Martí, Pau, Josep M. Fuertes, Gerhard Fohler and Krithi Ramamritham (2001*a*). Jitter compensation for real-time control systems. In: *Proceedings of the 22nd IEEE Real-Time Systems Symposium (RTSS 2001)*. London, Uk.

Martí, Pau, Ricard Villà, Josep M. Fuertes and Gerhard Fohler (2001*b*). On real-time control tasks schedulability. In: *European Control Conference*. Porto, Portugal.

Martí, Pau, Ricard Villà, Josep M. Fuertes and Gerhard Fohler (2001*c*). Stability of on-line compensated real-time scheduled control tasks. In: *IFAC Conference on New Technologies for Computer Control*. Hong Kong, China.