

Real-time control systems: feedback, scheduling and robustness

Daniel Simon, Alexandre Seuret, Olivier Sename

► **To cite this version:**

Daniel Simon, Alexandre Seuret, Olivier Sename. Real-time control systems: feedback, scheduling and robustness. International Journal of Systems Science, Taylor & Francis, 2017, 48 (11), pp.2368-2378. 10.1080/00207721.2017.1316879 . lirmm-01515226

HAL Id: lirmm-01515226

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-01515226>

Submitted on 10 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



DRAFT

Real-time control systems : feedback, scheduling and robustness

Daniel Simon^{a*}, Alexandre Seuret^b and Olivier Sename^c

^a CAMIN team, INRIA/LIRMM, Université de Montpellier, 860 Rue Saint Priest - 34095 Montpellier Cedex 5, France; ^bMAC team, LAAS CNRS, Université de Toulouse, 7 avenue du colonel Roche, BP 54200, 31031 Toulouse cedex 4, France; ^cSLR team, Gipsa-lab, Université Grenoble Alpes, 11 rue des Mathématiques BP 46, 38 402 St Martin d'Hères Cedex, Grenoble, France

ARTICLE HISTORY

Compiled April 28, 2017

The Version of Record of this manuscript has been published and is available in International Journal of Systems Sciences, <http://www.tandfonline.com/>, April 26 2017, DOI 10.1080/00207721.2017.1316879.

Abstract

The efficient control of real-time distributed systems, where continuous components are governed through digital devices and communication networks, needs a careful examination of the constraints arising from the different involved domains inside co-design approaches. Thanks to the robustness of feedback control, both new control methodologies and slackened real-time scheduling schemes are proposed beyond the frontiers between these traditionally separated fields. A methodology to design robust aperiodic controllers is provided, where the sampling interval is considered as a control variable of the system. Promising experimental results are provided to show the feasibility and robustness of the approach.

KEYWORDS

Computer Control, Robust Control, Real-Time Systems, Linear Parameter Varying Systems, Slack real-time, Control aware scheduling, Varying sampling

1. Introduction

The design and implementation of real-time control systems distributed over networks need to combine methodologies and technologies from the domains of control, computing and networking. Effective control methodologies, based on the feedback paradigm, must be further designed and implemented to cope with the fast evolution of the underlying technologies (Murray, Åström, Boyd, Brockett, and Stein (2003)).

Feedback control systems are deployed, often silently, over various fields of industry and services. For example, they are used in the control of automotive systems, for the coordination of swarms of mobile robots and in the redundant and distributed flight controllers embedded in modern aircraft. Networks for the production, transport and distribution of energy, water and gas are monitored by such automatic systems, just as the regulation of railway and aerial services. Sensors networks based on wireless links can be enhanced by adding some actuation and control functions to their sensing capabilities, for example in smart automated buildings (Baillieul and Antsaklis (2007)). Internet of Things now integrate various objects in large Cyber Physical Systems for which some kind of distributed control is needed, e.g. see (Lee (2008)), (Rajkumar, Lee, Sha, and Stankovic (2010)) and (Khaitan and McCalley (2015)) for founding references and further surveys.

For a control scientist point of view, the problem to be solved is the achievement of a control objective mixing stability, performance and reliability requirements. On the other hand, the computing and communication resources used to implement the control actions are numerical systems with limited capabilities such as computing speed and networking bandwidth (Hespanha, Naghshtabrizi, and Xu (2007)), whose consumption must be jointly studied and eventually optimized (Cao, Cheng, Chen, and Sun (2013)).

Moreover, especially in the case of embedded, mobile and autonomous systems, the energy storage and consumption are also subject to tight constraints. Jointly taking into account these various constraint may lead to a Quality of

*Corresponding author. Email: daniel.simon@inria.fr

Service formulation of the control problem, able to specify an acceptable trade-off between the control performance and the resources availability. Various control approaches have been already investigated to solve this control problem. For instance, it is relevant of Model Predictive Control, whose application for embedded cyber-physical systems has been already actively explored, as reviewed in (Lucia, Kögel, Zometa, Quevedo, and Findeisen (2016)). Another research direction considers fuzzy models of the distributed control system to further solve control problems related to packets dropping, time delays and signal quantization. See (Qiu, Gao, and Ding (2016)) for a recent survey, in which it is observed that most existing works only consider partial networked-induced phenomena to reduce the complexity of investigation.

1.1. *Varying sampling approaches*

The idea of using varying control intervals naturally arises when the available computing power devoted to feedback control is limited, e.g. in embedded systems. It is obvious that decreasing the control frequency directly decreases the amount of calculation cycles needed for control. In that case a feedback scheduler is assumed to compute on-line new control intervals according to the Central Processing Unit (CPU) load and system's state.

Different control approaches have been considered to design varying sampling controllers, e.g., see (Cloosterman et al. (2010)) for a survey. Among others, in (Hu and Michel (2000)) a stability analysis of the effect of varying sampling is provided for nonlinear systems, and in (Olm, Ramos, and Costa-Castelló (2011)) an H_∞ approach is used to conclude on the allowed sampling intervals that keep stability of a digital repetitive control scheme. In (Cuenca, Garcia Gil, Arzén, and Albertos (2009)) stability analysis of a controlled system under non uniform sampling is realized with a predictor/observer structure. A Takagi-Sugeno fuzzy modeling approach is considered in Yoneyama (2010) to design state feedback controllers with varying sampling interval. A gridding approach is used in (Sala (2005)) to design a discrete-time controller and observer with variable sampling. More recently, in (Fiter, Hetel, Perruquetti, and Richard (2015)) a robust stability framework for Linear Time Invariant (LTI) systems with varying sampling is provided, thanks to a Lyapunov-Razumikhin stability conditions approach, and can be extended to event-triggered control, self-triggered control, and state-dependent sampling. Finally Hetel et al. (2017) reviews recent results about the stability of systems under aperiodic sampling.

1.2. *Slacken scheduling approaches*

Control systems are typical real-time applications. The execution of a control task (including data communication) must be achieved under time constraints (often given as deadline constraints) to guarantee its stability and the desired control performance. Control systems are often cited in the real-time literature as cases of "hard real-time systems" where deadline violations are considered as fatal faults, e.g. (G. C. Buttazzo (2011)). To deal with, the real-time community usually considers simplified assumptions, control tasks with fixed periods, hard deadlines and known worst-case execution times. These "hard" assumptions lead to the separation of concerns between control and scheduling. However, as they do not take into account the robustness properties of feedback control systems, they also lead to over-constrained specifications and to under use the execution resources.

Robustness in control traditionally only considers the plant's parameter uncertainties. For simple SISO continuous systems it leads to design the system with phase, gain and delay margins. and the control algorithm computation time is often neglected at design time. Indeed the computation induced delay does not compromise the closed loop stability as far as it is smaller than the delay margin of the controller. Hence, the closed loop system is not only robust with respect to the plant parameters uncertainties, it is also robust with respect to unmodeled loop delays. A jitter margin also exists, as revealed in (Cervin, Lincoln, Eker, Årzén, and Buttazzo (2004)), and robustness considering timing deviations from the ideal delay-less and perfectly periodic case can be also observed for more general multi-variable systems as in, e.g., (Andrianiaina, Seuret, and Simon (2011)).

1.3. *Joint control and scheduling*

Therefore the traditional assumption considering control systems as "hard real-time" does not hold longer for *feedback* control systems. Conversely they must be considered as *weakly hard* real-time system, as defined in (Bernat, Burns, and Llamosí (2001)). It means that they are able to tolerate *specified deviations* from the ideal scheduling scheme while keeping their stability and specified performance level. Therefore, new degrees of freedom in the overall control system design can be used by jointly considering robust control and execution resources.

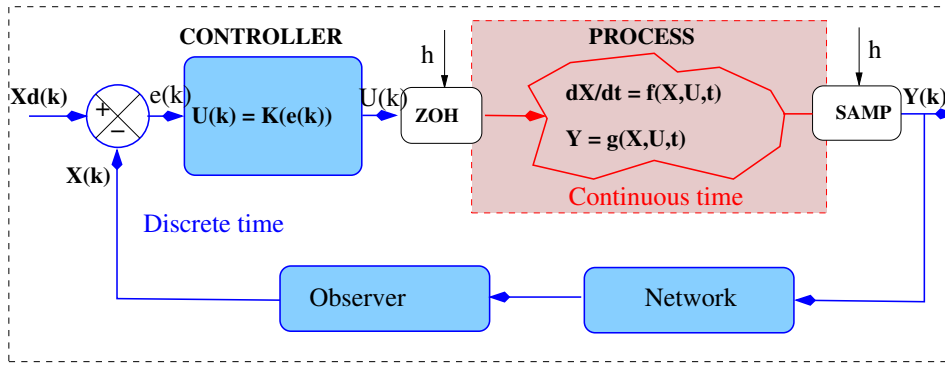


Figure 1. Control algorithms under execution resources constraints

Firstly, the execution platform (CPUs, tasks and fieldbus) can be designed using slackened scheduling parameters (i.e., with specified clock variations and sample loss rates), allowing for using off-the-shelf hardware and software components for a cheaper implementation of the system. Secondly, the robustness and adaptiveness properties of feedback control can be exploited to design new control algorithms freed from perfect equidistant sampling. Therefore, varying sampling or event-based control represent new tools to control the execution resources consumption. Finally, computing devices such as real-time schedulers are also likely to be controlled by feedback laws, as used in the founding paper (Eker, Hagander, and Arzen (2000)) where a feedback scheduler computes on-the-fly the control intervals of concurrent LQ controllers. However approaches gathering real-time scheduling and feedback control considerations are still quite rare. Let us for example cite (Blind and Allgöwer (2015)) for the analysis of closed-loop stability under weakly-hard scheduling constraints.

1.4. Problem formulation

Among other approaches, this paper aims at consistently collecting and extending some results of the authors, in particular those previously presented in (Andrianiaina, Simon, Seuret, Crayssac, and Laperche (2011); Robert, Sename, and Simon (2006, 2010); Roche, Sename, and Simon (2012)). Indeed the approach focus on the robustness of feedback loops with respect to variations of the sampling rate. It is assumed that the values of the control intervals are managed by a real-time scheduler which jointly runs with the controllers to share and manage the execution resources.

The design of effective real-time controllers for distributed and/or embedded systems can be stated as “control design under execution resources constraints”, for which co-design methodologies gathering, at design time, the constraints arising from control, computing and networking, are needed.

A controlled process is classically a physical device whose dynamics changes along a continuous time scale. Its dynamics is usually describes by some kind of differential equation, for example using a state space model :

$$\dot{x} = f(x, u, t) \quad y = g(x, u, t) \quad (1)$$

where x , y and u are vectors of continuous variables representing the state, the output and the control inputs of the plant, respectively. The functions f and g are non-linear functions, usually assumed to satisfy locally Lipschitz conditions.

The objective is to design a control law u which guarantees the performances of the closed-loop system shown in Figure 1. The control algorithms can be designed in continuous time, then the controller is discretized assuming that the sampling rate is fast enough considering the desired closed-loop dynamics. It can be also directly synthesized in discrete time, using a model of the plant sampled by a zero-order holder and a periodic clock. A control theory for linear systems sampled at fixed rates was established a long time ago, e.g. (Åström and Wittenmark (1997); Chen and Francis (1995)). However, in the context of distributed control systems, more complex assumptions are needed to consider that the control loop is implemented using a digital system where data are exchanged over networks. To ensure the performances of the control system, it is necessary to handle these phenomena from both the automatic control and the computer science viewpoints.

Outline The paper is organized as follow : the next section formalizes the problem of control under execution resources constraints, and recalls the basic timing schemes used in real-time control. Section 2.2 proposes some slackened real-time scheduling scenarii compliant with the robustness of feedback control. Then Section 3 provides a methodology to design varying sampling controllers using the sampling interval as a control parameter. A preliminary analysis of the robustness of such controllers is attempted in Section 3.4 and a study case of varying sampling control of an inverted pendulum is given in Section 4.

2. Scheduling for control

Beyond the usual separation of concern between control and computing, the design of control-aware real-time systems needs to examine scheduling schemes with both control requirements and implementation feasibility in mind, as done for some scheduling scenarii in the next sections.

2.1. Timing parameters assignment

For computer-controlled systems, the control objective is achieved by a feedback control law u which cyclically computes the control inputs $u(t_k)$ at some instants t_k where $0 = t_0 < t_1 < t_2 \dots$ and such that t_k tends to infinity as k tends to infinity. The control input depends on the tracking error signal $e(t_k) = y_d(t_k) - y(t_k)$ between the desired reference and the plant output (Figure1).

Samplings and delays introduce distortions with respect to the original continuous time framework, especially when limited computing resources induce long calculations and variations in the sampling intervals. Anyway the control objectives. i.e. stability and specified performance, must be achieved despite the imperfections of the execution resources such as limited bandwidth and varying computation times (note that quantization related problems are out of scope for the presented approach).

In large control systems, concurrent controllers are more and more executed as concurrent tasks under control of a real-time operating system (RTOS). Besides static scheduling used in highly critical system such as aircraft control, many off-the-shelf RTOS make use of fixed-priority and allow preemption between concurrent tasks. The performance of a given control law highly rely on the value of the sampling interval and of the loop latency, i.e. the delay between a measurement instant on a sensor and the instant at which the related control signal is applied to the actuator (Åström and Wittenmark (1997)). Hence, the control law behavior depends on the scheduling parameters of the real-time system, which must be designed so that the resulting actual sampling rates and latencies are compliant with the control design assumptions.

Even choosing a right value for the sampling rate of a control law is not so simple, while being decisive for the control performance. There are some rules to be applied for linear systems. Shannon's theorem gives a lower limit for the sampling rate and rules of thumb may give a first approach to select an effective control interval. For example (Åström and Wittenmark (1997)) recommends that $\omega_c h \approx 0.15 \dots 0.5$, where ω_c is the desired closed loop pulsation and h is the sampling period. This is only a preliminary setting and the sampling rate selection needs further simulations and tests, especially for non-linear systems. Anyway, for centralized systems without communication links, it is often observed that higher the sampling rate and lower the latency, better is the control performance.

However, the execution resources used to implement a control system are limited and cannot accommodate arbitrarily high sampling rates. Hence the scheduling parameters are the result of a trade-off between the control performance and the execution resource capabilities. This is especially true for control systems distributed over networks where the communication links are subject to saturation in case of fast sampling. The chart in Figure 2a –taken from (Moyné and Tilbury (2007))– shows the importance of selecting a right sampling rate to provide a satisfactory trade-off between the control performance and both the computing and networking loads.

2.2. Control aware scheduling

Implementation features such as multi-rate sampling, preemption, synchronization and various sources of delays make the run-time behavior of the controller very difficult to accurately predict. Even for the case of control tasks running in isolation inside static slots, a source of indeterminacy comes from the tasks execution time which randomly varies between a best and a worst case as in Figure 2b taken from (Hansen, Hissam, and Moreno (2009)).

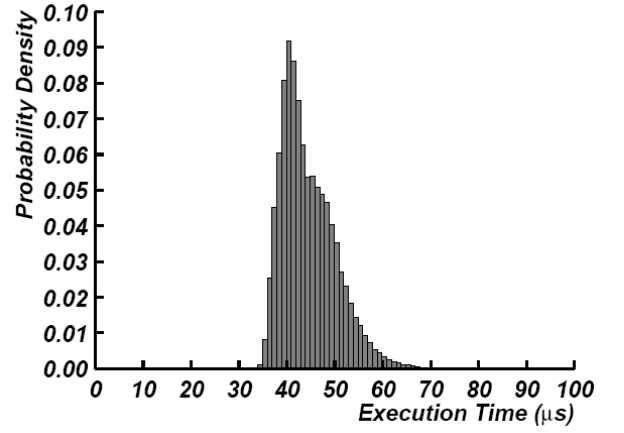
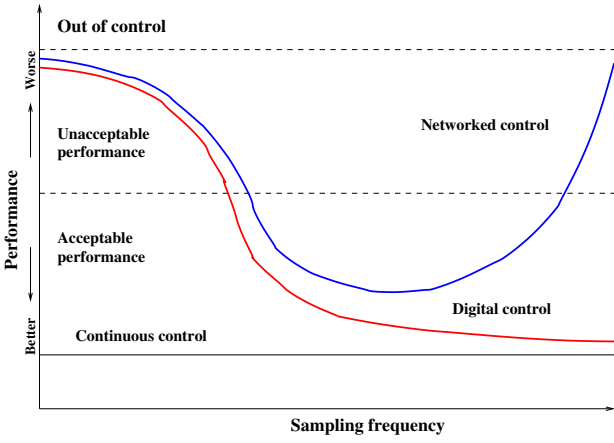


Figure 2. a) Feedback performance vs. sampling – b) Typical distribution of execution times

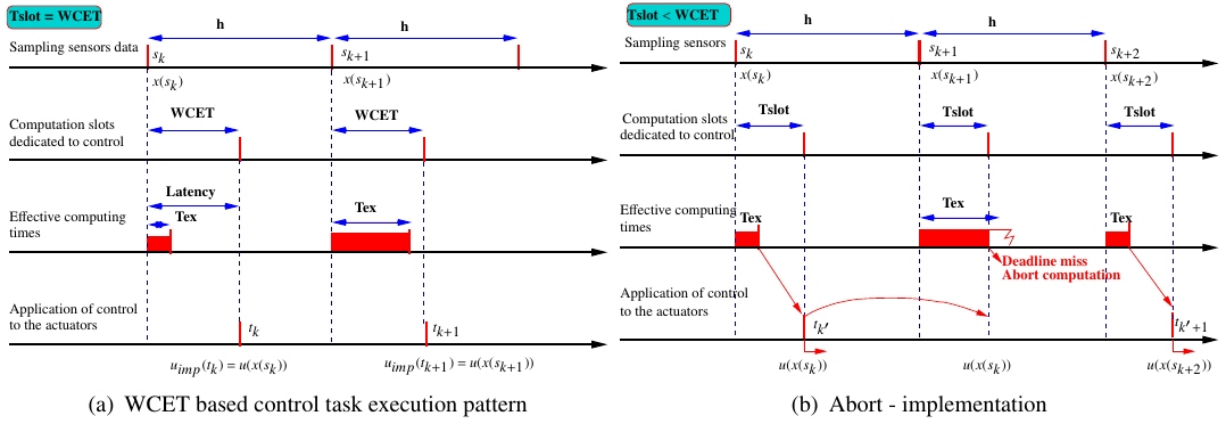


Figure 3. Different control implementation schemes

Assuming static scheduling, where fixed size slots are allocated to tasks on a shared CPU, several scheduling schemes are analyzed with control constraints in mind.

2.2.1. Clock driven periodic, hard real-time

This first scheduling scheme of a control task fits with the common assumption of perfectly periodic execution and known Worst Case Execution Time (WCET), as depicted by Figure 3(a).

A periodic time slot $T_{slot} = WCET$ is reserved for the task execution. It is triggered with a period $s_k - s_{k-1} = h$, $\forall k \in \mathbb{N}$ and $s_0 = 0$, by an equidistant clock providing measurements $x(s_k)$ at time s_k . The controller computation takes duration T_{ex} which is by definition smaller than the $WCET$. The control signal is applied and hold to the actuators at the end of the slot to avoid jitter, i.e. at time $s_k + WCET$:

$$\forall t \in [s_k + WCET, s_{k+1} + WCET[, \quad u(t) = u(x(s_k)).$$

Therefore, it is a periodic control system, with constant period h , subject to a constant delay $T_{slot} = WCET$.

This setting complies with the hard real-time assumption, and should be applied when the controller is not robust against sampling disturbances or if it is not allowed to deviate from the perfect periodic scheme by regulations. As deadline miss are considered as fatal failures, the WCET must be very carefully evaluated and, in practice, safety margins are given using over-estimates.

Anyway, as the time slots are assigned based on the WCET of the control tasks, their computation always terminate before the end of the slot and a fraction of the slot is unused. The wasted computing power is even more large when the WCET is by far larger than the average value of the observed execution time T_{ex} . Indeed it is the case for modern CPUs using hardware acceleration such as caches and pipelines, leading to very large ratio $WCET/BCET$.

	latency min max	interval min max	jitter max	interest	drawback
Hard RT	T_{slot}	h	0	periodic, easy analysis	worst-case proportioning
Abort	T_{slot}	h Nh	0	refresh inputs	needs re-initializable tasks
Skip	T_{slot} $N.T_{slot}$	h Nh	0	easy implementation	potentially unbounded
Spare	T_{slot}	h Nh	0	easy implementation	wasted spare time
Asap	BCET ... WCET	h Nh	WCET - BCET	lowered latency	jitter
Strip	T_{slot}	$h - (WCET - T_{slot})$	0	improved performance	no spare time

Table 1. Summary of scheduling patterns

2.2.2. Weakly hard real-time schemes

To improve the average utilization ratio of the execution resources in embedded controllers it is proposed to slacken the usual hard real-time constraints. The number of wasted computing cycles can be decreased by allocating computing slots smaller than the WCET at the cost of occasional data loss. It is argued that a robust feedback controller is able to keep its stability despite timing disturbances, and that the scheduling and control parameters can be jointly tuned to reach an acceptable trade-off between the control performance and the computing average efficiency. Indeed most control tasks computation are likely to finish on time, and sporadic data loss must be processed by various overrun handlers, e.g. as in (Cervin (2005)), with related consequences both for control performance and for implementation feasibility.

In the following cases a time slot $T_{slot} < WCET$ is allocated to the control task. The initial clock h of incoming sensory data keeps its original constant value, and the control signal is sent to the actuators at the end of the slots to nullify the calculation induced jitter.

In case of deadline misses referring to Figure 3(b), the execution of the running control task is **Aborted**, the current control signal is hold, and a new computation is started with the freshly available sensory data at the next allocated slot. The control signal $u(x(s_k))$ is hold for one extra period so that the deadline miss occur at time $s_k + T_{slot}$:

$$\forall t \in [s_k + T_{slot}, s_{k+2} + T_{slot}[, \quad u(t) = u(x(s_k)),$$

and for N consecutive deadline misses and data loss:

$$\forall t \in [s_k + T_{slot}, s_{k+N} + T_{slot}[, \quad u = u(x(s_k)).$$

Consequently, the control input becomes asynchronous since the difference between two control signal updates is time-varying (but bounded by h and Nh). From the implementation side, it is needed to abort the control task at arbitrary points and restart it with the right values of its internal registers : this can be difficult in particular if dynamic controllers or recursive filters are involved in the computation.

Besides this basic resource allocation scheme, others scheduling strategies can be associated with various overrun handlers (Andrianiaina, Simon, et al. (2011)). For example, it is possible to **Skip** the next clock tick and allow the task to terminate during the next slot. With **Asap**, the control signal is sent to the actuators as soon as possible , i.e. when the computation ends, to minimize the average input/output latency at the cost of output jitter. In this case it is expected –e.g. (G. Buttazzo and Cervin (2007))– that the disturbance due to the output jitter is compensated by the average lowered value of the loop latency to finally increase the control performance. Some **Spare** slots can be reserved for handling overruns. Also, rather than keeping the original value of the input interval, the triggering interval can be **Stripped**, so that the control performance is expected to improve from both the latency and sampling interval diminution.

The main features of these scheduling schemes and overrun handlers are summarized in Table 1 down below. They can be combined according to the features of various case studies, coming from both the plant dynamics, parameters uncertainties and implementation feasibility considerations.

Obviously the stability of a feedback controller subject to weakly-hard real-time constraints must be proven. For example, based on recent advances in control –using Lyapunov-Krasovskii functionals and LMIs systems solvers– it has been proven that even an aircraft flight control system can keep its stability despite a quite large amount of samples loss in the feedback control loops, allowing for an substantial lightening of the computing equipment (Andrianiaina, Seuret, and Simon (2011)).

3. LPV robust control and varying sampling

Conversely with the previous section, where fixed scheduling parameters were set to execute a given controller with a computing resource, it is now examined how these timing parameters can be adapted on-the-fly to cope with varying conditions.

Linear Parameter Varying (LPV) representations are increasingly used to deal with robust control of non-linear systems. In particular this allows to represent some non-linearities as varying parameters, and also to schedule a controller according to some performance criteria, e.g. (Shirazi, Mohammadpour Velni, and Grigoriadis (2011)). The authors have proposed (among others, e.g. (Hetel, Daafouz, and Jung (2006))) Linear Parameter Varying (LPV) models of digital control systems with varying sampling, under the assumption that the system representation is affine with respect to the sampling interval (Robert et al. (2006)).

3.1. Discrete-time model with varying sampling interval

This subsection briefly describes the considered model, following the methodology in (Robert et al. (2010)). A state space representation of a continuous time plant is considered as :

$$G : \begin{cases} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{cases} \quad (2)$$

The exact discretization of this system with a zero order hold at the sampling period h can be computed (Åström and Wittenmark (1997)), leading to the discrete-time LPV system (3)

$$G_d : \begin{cases} x_{k+1} &= A_d(h) x_k + B_d(h) u_k \\ y_k &= C_d(h) x_k + D_d(h) u_k \end{cases} \quad (3)$$

with h ranging in $[h_{min}; h_{max}]$. The corresponding sampling and hold scheme is as follow : the control signal $u[t_k]$ computed at the k^{th} instant from measure $x[t_k]$ is hold until instant t_{k+1} , which is known and given by a controlled scheduler, e.g. by a feedback scheduler or a (m, k) -firm policy. $u[t_k]$ (resp. $x[t_k]$) are then denoted u_k (resp. x_k), and h is referred to as the sampling interval, i.e:

$$h \triangleq h(k) = t_{k+1} - t_k$$

Computing A_d and B_d involve matrix exponentiation of the original A and B matrices. Since h is assumed to be in the interval $[h_{min}, h_{max}]$ with $h_{min} > 0$, the sampling interval is defined around the nominal value h_0 of the sampling range, as :

$$h = h_0 + \delta \quad \text{with} \quad h_{min} - h_0 \leq \delta \leq h_{max} - h_0 \quad (4)$$

Then we write :

$$\begin{pmatrix} A_d(h) & B_d(h) \\ 0 & I \end{pmatrix} = \begin{pmatrix} A_{h_0} & B_{h_0} \\ 0 & I \end{pmatrix} \begin{pmatrix} A_\delta & B_\delta \\ 0 & I \end{pmatrix} \quad (5)$$

$$= \exp\left(\begin{pmatrix} A & B \\ 0 & 0 \end{pmatrix} h_0\right) \cdot \exp\left(\begin{pmatrix} A & B \\ 0 & 0 \end{pmatrix} \delta\right) \quad (6)$$

The previous equation gives the exact sampling dependent model of a continuous-time state space representation. In the framework of the H_∞ approach, a discrete-time, sampling interval dependent, augmented plant is obtained from the continuous time plant and from weighting functions.

In previous works of the authors, two different methodologies have been provided for sampling interval dependent control design, the first one based on Polytopic models (Robert et al. (2010)) and the second one using the Linear Fractional Representation (LFR) approach (Roche et al. (2012)). Only the first approach is briefly described below.

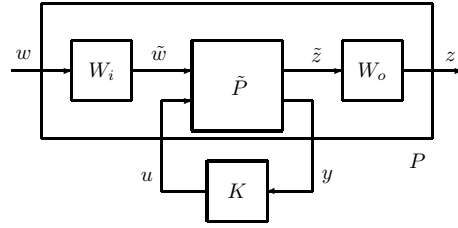


Figure 4. Weighting functions interconnection

3.2. The Polytopic approach

As seen before, A_δ and B_δ involve exponential matrices of the original A and B matrices that are not affine on δ . To get a polytopic model, a Taylor series of order \mathcal{T} is used to approximate the matrix exponential in (5), and allows to get :

$$A_\delta \approx I + \sum_{i=1}^{\mathcal{T}} \frac{A^i}{i!} \delta^i \quad \text{and} \quad B_\delta \approx \sum_{i=1}^{\mathcal{T}} \frac{A^{i-1} B}{i!} \delta^i \quad (7)$$

This leads to

$$A_d(h) = A_{h_0} A_\delta \quad \text{and} \quad B_d(h) = B_{h_0} + A_{h_0} B_\delta \quad (8)$$

Let us define $H = [\delta, \delta^2, \dots, \delta^{\mathcal{T}}]$ the vector of parameters, that belongs to a polytope \mathcal{H} with $2^{\mathcal{T}}$ vertices.

$$\mathcal{H} = \left\{ \sum_{i=1}^{2^{\mathcal{T}}} \alpha_i(\delta) \omega_i : \alpha_i(\delta) \geq 0, \sum_{i=1}^{2^{\mathcal{T}}} \alpha_i(\delta) = 1 \right\} \quad (9)$$

$$\{\delta, \delta^2, \dots, \delta^{\mathcal{T}}\}, \delta^i \in \{\delta_{min}^i, \delta_{max}^i\} \quad (10)$$

Combined with weighting functions, the polytopic model is recast into a polytopic generalized plant, further used to design self-scheduled controllers. This self-scheduled controller is given by the convex combination of $2^{\mathcal{T}}$ "vertex" controllers, where \mathcal{T} (i.e. the order of the Taylor expansion) tunes a trade-off between the approximation precision and the controller complexity.

Another way to reduce the controller complexity considers the case where the sampling interval is the only varying parameter in the model. Therefore the dependency between the variable parameters –i.e. are the successive powers of the sampling period $h, h^2, \dots, h^{\mathcal{T}}$ – is used to reduce the number of elementary vertex controllers to be combined from $2^{\mathcal{T}}$ down to $\mathcal{T} + 1$. Moreover solving the control synthesis with a smaller set of polytopic parameters drastically decreases the conservatism of the method and makes the solution easier to implement (see (Robert et al. (2010)) for details).

3.3. LPV sampling dependent control design

3.3.1. Performance specification

Using the H_∞ framework, the general control scheme of Figure 4 is considered, where W_i and W_o are weighting functions specifying the closed-loop performance (Skogestad and Postlethwaite (2005)). The objective is to synthesize a controller K that ensures internal stability and $\|\tilde{z}\|_2 < \gamma \|\tilde{w}\|_2$, where γ is the H_∞ level performance (actually the L_2 induced gain).

It is well known that the achievable performance of a feedback control systems depends on the sampling frequency. Conversely with classical robust controllers where the closed-loop objectives are statically defined by LTI weighting functions, the closed-loop performance objectives are here adapted to account for the sampling interval variations. This is done using 'sampling-dependent' weighting functions. It is enough to use only a single weighting function, W_e , acting on the tracking error, to allow for the performance adaptation with respect to the current sampling interval, as explained in (Robert et al. (2010)).

3.3.2. LPV/ H_∞ polytopic control design

When the reduced polytope is considered, $(\mathcal{T} + 1)$ controllers are synthesized at the vertex of the parameter polytope (corresponding with the extreme values of the parameters). The gain-scheduled controller $K(H)$ is then the convex combination of these controllers

$$K(H) : \begin{pmatrix} A_K(H) & B_K(H) \\ C_K(H) & D_K(H) \end{pmatrix} = \sum_{i=1}^r \alpha_i(h) \begin{pmatrix} A_{K_i} & B_{K_i} \\ C_{K_i} & D_{K_i} \end{pmatrix} \quad (11)$$

$$\text{with } \alpha_i(h) \text{ such that } H = \sum_{i=1}^r \alpha_i(h) \omega_i \quad (12)$$

where $A_{K_i}, B_{K_i}, C_{K_i}$ and D_{K_i} have the dimensions of the the original system (2)). It can be seen as a dynamic state feedback controller :

$$\begin{cases} x_{K_{k+1}} &= A_K(H)x_{K_k} + B_K(H)y_k \\ u_k &= C_K(H)x_{K_k} + D_K(H)y_k \end{cases} \quad (13)$$

for which the on-line scheduling needs to compute $\alpha_i(h)$ knowing h . Considering a Taylor's expansion at order \mathcal{T} around h_0 with

$$\delta_{min} = h_{min} - h_0 \text{ and } \delta_{max} = h_{max} - h_0$$

and the case of the reduced polytope, explicit solutions are easily recursively computed as :

$$\begin{cases} \alpha_1 = \frac{\delta_{max} - \delta}{\delta_{max}^n - \delta_{min}^n} \\ \alpha_n = \frac{\delta_{max} - \delta}{\delta_{max}^n - \delta_{min}^n} - \sum_{i=1}^{n-1} \alpha_i, \quad n = [2, \dots, \mathcal{T}] \\ \alpha_{\mathcal{T}+1} = 1 - \sum_{i=1}^{\mathcal{T}} \alpha_i \end{cases} \quad (14)$$

3.4. Robustness of LPV based varying sampling

In this section, the aim is to check the robustness of the previous control law with respect to modeling and sampling uncertainties. A preliminary analysis is given here for controller (13) to take into account the computation durations. Recall that the controlled plant is governed by the following differential equation :

$$G : \begin{cases} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{cases} \quad (15)$$

and that the dynamic output feedback controller is given by (13). Note that, conversely with the original controller (13) where the production of the control signal u_k was considered as instantaneous and synchronous with the inputs y_k , the new version of the controller accounts for a variable computation duration and assumes that the control signal is sent to the actuators as soon as it is computed (as in the **Asap** scheduling scenario of Section 2.2.2).

It is worth noting that this system evolves in continuous-time while the dynamic output feedback controller is computed in discrete-time. This particular interaction between the continuous and discrete-time systems needs a particular attention when analyzing stability and more particularly robustness. In particular, it is well-known that the stability analysis of this class of systems is difficult to extend in the case of model uncertainties. The difficulty lies in finding a model of the discretized systems (3) when the continuous-time model is not perfectly known.

However, it is possible to analyze this class of system using an alternative approach which consists in firstly rewriting the closed-loop systems as an impulsive systems, e.g. (Bainov and Simeonov (2001)) and (Briat and Seuret (2012)). To guarantee stability, the theorem taken from (Briat and Seuret (2012)) has been tested. The main motivation for choosing this stability conditions are expressed in terms of Sum of Squares which allows to take into account the particular structure of the scheduled control law.

However preliminary tests of the method show that the computation time is prohibitive, even with a very small size toy example. As the number of LMIs to be solve increase very fast with the size of the system, it is not usable even for

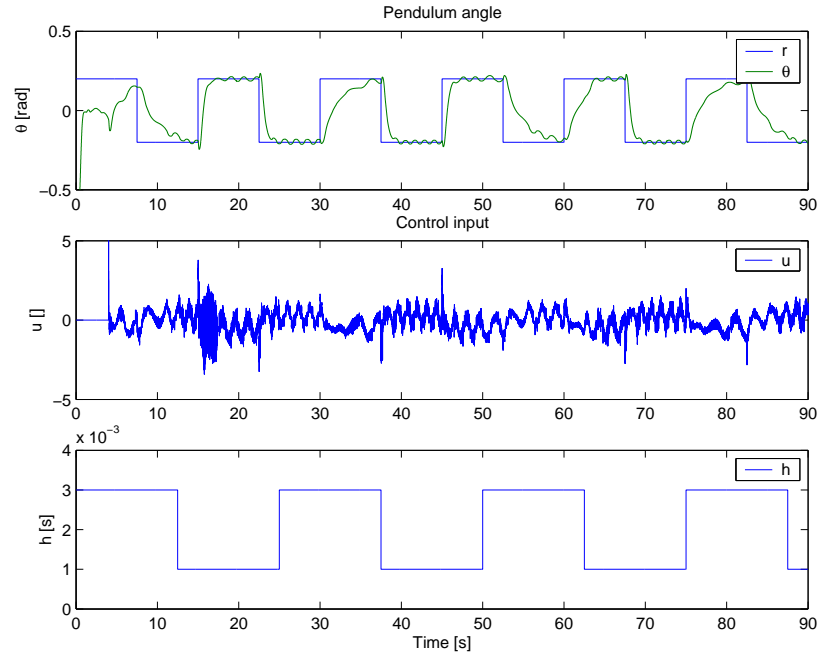


Figure 5. Position of the T pendulum under square sampling variations

the moderate size system studied in the next section. It is expected that effective solutions can be provided by other approaches, e.g. based on weakly-hard real-time considerations as suggested by (Blind and Allgöwer (2015)) for the case of switched systems. Others modeling paradigms may be also investigated, e.g., as in (Qiu, Wei, and Karimi (2015)) where time delays are handled inside hybrid systems modeled by Markov jump systems.

4. Test case

The polytopic approach of Section 3.3.2 was implemented and tested using an under-actuated "T" inverted pendulum (Educational Control Products model 505), as detailed in (Robert et al. (2010)). The objective here is to show the stability of the closed-loop system subject to specified sampling variations, and to show the feasibility of the approach using off-the-shelf real-time systems and tools.

4.1. Real-time set-up and experiments

The controller is synthesized for sampling rates varying in the interval $[10^{-3}, 3 \cdot 10^{-3}]$ s. The 1 to 3 ms sampling rate range is chosen as it fits with the frequency scaling capabilities of the chips used in many current laptops, thus enabling the controller to fully use the hardware capabilities. Note that, even if it seems to be very fast with respect to the desired bandwidth, previous experiments show that this particular open-loop unstable and under-actuated system needs fast sampling to be stabilized, whatever the control algorithm.

The controller is synthesized using a reduced polytope model and a Taylor's expansion truncated at the order 2. Hence three vertex controllers only are combined for every new value of the control interval. The controller is implemented using the Real-time Workshop and xPC Target, using the output of the robust synthesis toolbox provided by Matlab/Simulink.

In Figure 5 the pendulum angle tracks a step reference while the sampling rate is subject to step changes, as seen at the bottom of Figure 5.

As requested by the sampling-dependent performance objectives defined in Section 3.3.1, the settling time is small when the control interval is large, and conversely. Note that, even if the sampling interval is subject to abrupt changes, the control signal keeps values smaller than the actuator saturation (Figure 5 middle, where the spikes in the control signal are due to friction and elasticity in the actuator's drive).

According to (14), the computation of the polytopic coordinates reduces here to the simple explicit solutions (with

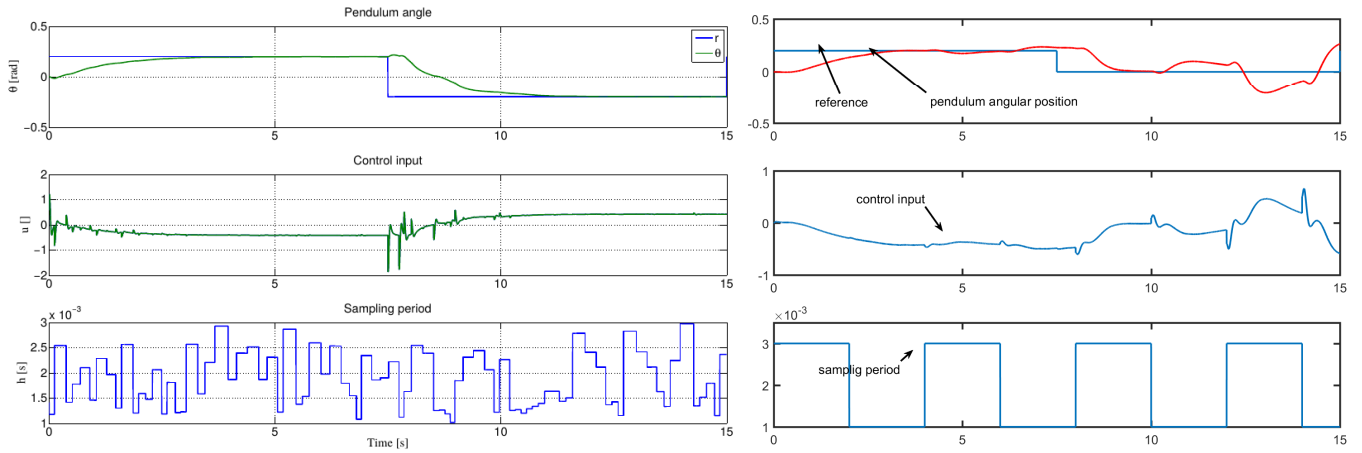


Figure 6. i) LPV control with random variations of h ii) switching between two LTI controllers

$\delta_{min} = 0$) :

$$\alpha_1 = \frac{\delta_{max} - \delta}{\delta_{max}}, \quad \alpha_2 = \frac{\delta_{max}^2 - \delta^2}{\delta_{max}^2} - \alpha_1, \quad \alpha_3 = 1 - (\alpha_1 + \alpha_2)$$

The controller reconstruction given by (11) involves 3 vertex controllers whose size is the one of the state vector of the plant, i.e. it involves a linear combination of 4×4 matrices.

Moreover the controller updates only occur when the value of the control interval changes. Hence in the case of sporadic step changes, the on-line overhead of the method is negligible compare with the cost of a classical H_∞ controller. Even in the case of continuously varying sampling, the computation of the new gains only involve the simple algebraic computations given by (14) at each sample.

Finally, note that this kind of varying sampling controller can be easily implemented using the basic features shared by almost all RTOS, i.e. programmable clocks and real-time tasks.

4.2. Further testings in simulation

The only assumption made about the sampling rate during control synthesis is that the control interval lies inside predefined bounds $[h_{min}; h_{max}]$. The interval can be changed at any speed, between any admissible values at any instant, ensuring both the system stability and the specified performance (Figure 6-i). Conversely, a naive solution uses a bank of LTI controllers, each of them synthesized for a given sampling interval. In that case it is necessary to store the internal state and matrices of all the controllers of the bank and to carefully initialize them to avoid stability problems at switching time. However, as seen in (Figure 6-ii) the performances are in that case not satisfactory compared with the LPV controller. Even if some comparison with other control methods could also make sense, Figure 6 emphasizes that the LPV approach is an interesting tool for dealing with real-time control using varying sampling rates, while being simple enough to be implemented using off-the-shelf real-time tools.

The capabilities of the polytopic controller, in particular its robustness against unmodeled delays, are now explored using simulations.

4.2.1. Control under a data dropping (m,k) -firm policy

Varying sampling may origin from several sources, it can be either a control variable used to manage the execution resources, or it can be considered as a disturbance imposed by the execution environment.

- In the first case, the control interval is a control variable used to jointly manage both the control performance and the execution resources (CPU and/or fieldbus) utilization. Besides off-line management techniques, resource sharing can be done dynamically by a feedback scheduler whose outputs are the scheduling parameters of the real-time system, in particular the control intervals of the real-time tasks and the flow of messages. The case when the control interval can take any value in the predefined interval is well suited to finely manage the CPU load in a preemptive real-time operating system;

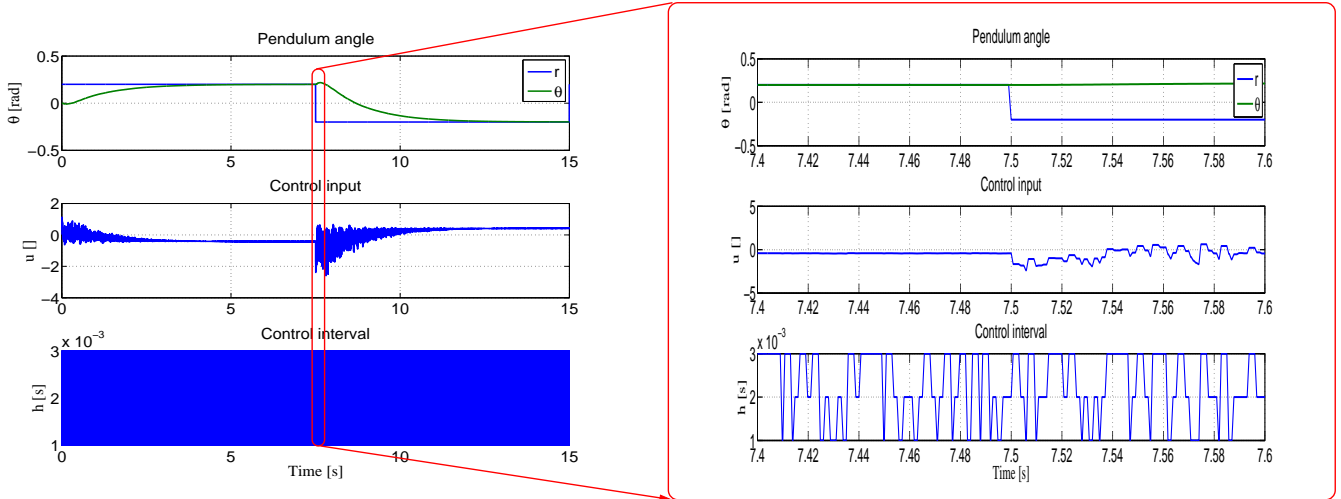


Figure 7. LPV with (1-3)-firm data dropping scheduling

- In the second case the control interval is an uncontrolled but measured parameter. However, it is assumed to lie inside predefined bounds, and can be estimated in real-time. It can be also predefined, as in the case of statically scheduled systems. For example, predefined Time Division Multiple Access (TDMA) scheduling of ultrasonic sensors is used in underwater applications (Roche, Sename, Simon, and Varrier (2011)). Selective data dropping under control of a (m,k)-firm scheduler allows for network overload management as in (Felicioni, Jia, Simonot-Lion, and Song (2010)). Note that such (m,k)-firm data dropping policy, leading to control intervals which are multiple of a base clock, is well suited for the management of the control data flow over networks because preemption cannot be used over communication links.

In the simulation plotted by Figure 7 the control interval is randomly selected with values in the set $\{1, 2, 3\}$ ms. It is for example the case of sensors and control task sampled at 1KHz, followed by a (m,k)-firm scheduler using a (1,3) dropping policy to cancel some messages sent towards the actuator. The dropping selection can be made according to some trade-off between the control quality and the network load. Here the control quality is managed through the sampling-dependent weighting functions used for the robust synthesis.

Note that it is a particular switched system (or gain scheduled system) for which the design process, using a robust \mathcal{H}_∞ synthesis and a Lyapunov function approach, ensures the system stability for all values of the sampling interval inside the predefined bounds (and not only for a subset of discrete values between the bounds).

4.2.2. Robustness with respect to unmodeled delays

Until now it is assumed that the control signal sent by the LPV controller is not subject to delays or jitter. Indeed delays and jitter are induced by the control tasks computation itself –due to varying calculation times– and to the preemption between the control tasks competing to gain access to the CPU. Also, in case of a distributed execution, delays are added by the fieldbus used to connect the controller and the actuator. Obviously these added delays are uncertain and cannot be accurately known when computing the polytopic gains and vertex controllers combination. Assuming that the output delay is bounded, a possible solution, given in (Sala, Cuenca, and Salt (2009)), consists in computing, at each control execution, a set of control signals corresponding to the expected delay range with some gridding step, sending all the set to the actuator and apply the particular control signal which matches the measured latency. However this approach is paid by overloading both the CPU and fieldbus with useless computations and communications, which may themselves increase the execution induced delays.

Indeed, as any feedback control loop, the polytopic controller is robust, to some extent, against modeling errors and unmeasured delays. Note that nothing was done in the experiments previously depicted in Figure 5 to account for the execution latencies induced by the execution and scheduling of the control law.

Additional simulations were performed to explore the robustness of the polytopic controller against unmodeled latencies. In the simulation depicted in Figure 8, the control interval was set to a constant value (2 ms) and a delay with increasing values (upto 12 ms) is inserted in the control path between the control task and the actuator. It can be observed that the control system is still stable for large values of the unmeasured delay, as instability only appears for

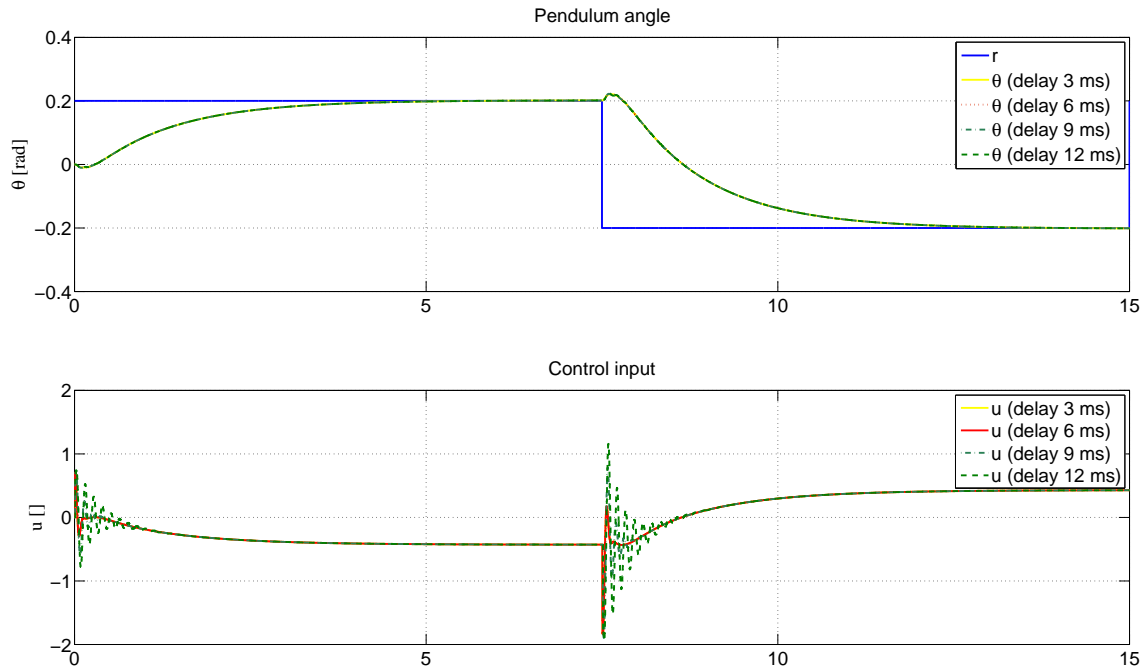


Figure 8. LPV control with unmodeled output delay

delays larger than 15 ms (a value surprisingly larger than the control interval). The response of the system degrades with an oscillatory behavior while the latency increases, and the degradation seems to be monotonically increasing with the added delay in this particular case study.

Hence, the variable sampling control synthesis using the polytopic method appears to be effective to robustly stabilize the study case plant, with respect to predefined performance objective over the range of desired control intervals. Therefore this process controller can be easily used as a building block in a Quality of Service control loop, where the higher level feedback scheduler can be designed only with the execution resources management in mind, without worrying about the stability of the low-level control loops subject to varying sampling rates.

5. Summary and perspectives

The efficient design of controllers for distributed systems needs to carefully examine the constraints arising from the domain of control, real-time computing and networking, and to early integrate these constraints in the design process. Usual control systems using a hard real-time assumption lead to over-constrain the system specification and to under-use the execution resources. Understanding how feedback controllers are robust w.r.t sampling deviations and data loss allows for the design of control systems where the real-time assumptions are slackened, and where the execution resources can be efficiently used while preserving the system's stability and specified performance.

The association of robust control and slackened real-time provides new degrees of freedom in the design of real-time control systems. Many industrial applications are concerned with similar requirements of real-time control such as robotics, automotive and aerospace industries. Several such applications for aircraft control (Andrianiaina, Seuret, and Simon (2011)), mechanical devices (Robert et al. (2010)) and underwater robotics (Roche et al. (2011)) are referenced in the paper.

Anyway research efforts are still needed to efficiently integrate the constraints arising from the control and computing domains in a system wide Quality of Service viewpoint. In particular, beyond the simulation case study, the robustness of control loops using the variable sampling rate as a control parameter of the system deserves to be further investigated. It has been already noticed, in Section 2.2.2 and citations herein, that the robustness of a feedback loop is concurrently shared between robustness with respect to uncertainties of the plant parameters and robustness considering unmodeled loop delays. Handling both the plant parameters and the sampling interval as uncertain parameters of an LPV model has been already done for specific case studies, e.g. in (Roche et al. (2012)) using Linear Fractional

Transforms, the approach deserves to be generalized.

Adding unknown bounded time delays in such models, as sketched in Section 3.4, remains to be done. Possible research directions include extended models of the hybrid control system as, e.g., Markov jump systems (Qiu et al. (2015)). Another promising way is the translation of the weakly-hard scheduling constraints (gathering aperiodic sampling and time delays) into stability conditions as done in (Blind and Allgöwer (2015)) to model the data loss process in a control loop.

Finally the real-time scheduler itself may induce other sporadic events that could disturb the control application. Such problems may also lead to research studies integrating stochastic models of the real-time environment.

References

- Andrianiaina, P.-J., Seuret, A., & Simon, D. (2011, December). Robust control under weakened real-time constraints. In *50th IEEE CDC/ECC*. Orlando, USA.
- Andrianiaina, P. J., Simon, D., Seuret, A., Crayssac, J.-M., & Laperche, J.-C. (2011, December). *Weakening Real-time Constraints for Embedded Control Systems* (Research report No. RR-7831). INRIA. Retrieved from <http://hal.inria.fr/hal-00650627>
- Åström, K. J., & Wittenmark, B. (1997). *Computer-controlled systems* (third ed.). Upper Saddle River: Prentice Hall.
- Baillieul, J., & Antsaklis, P. (2007). Control and communication challenges in networked real-time systems. *Proceedings of the IEEE*, 95(1), 9–28.
- Bainov, D., & Simeonov, P. (2001). *Systems with impulse effects: Stability, theory and applications*. New York: Springer-Verlag.
- Bernat, G., Burns, A., & Llamosí, A. (2001). Weakly hard real-time systems. *IEEE Trans. on Computers*, 50(4), 308–321.
- Blind, R., & Allgöwer, F. (2015). Towards networked control systems with guaranteed stability : Using weakly hard real-time constraints to model the loss process. In *Conference on decision and control (cdc)* (pp. pp 7510–7515). Osaka, Japan.
- Briat, C., & Seuret, A. (2012). Convex dwell-time characterizations for uncertain linear impulsive systems. *IEEE Transactions on Automatic Control*, 57(12), 3241–3246.
- Buttazzo, G., & Cervin, A. (2007, March). Comparative assessment and evaluation of jitter control methods. In *Proc. 15th international conference on real-time and network systems*. Nancy, France.
- Buttazzo, G. C. (2011). *Hard real-time computing systems: Predictable scheduling algorithms and applications*. Springer.
- Cao, X., Cheng, P., Chen, J., & Sun, Y. (2013, Feb). An online optimization approach for control and communication codesign in networked cyber-physical systems. *IEEE Transactions on Industrial Informatics*, 9(1), 439-450.
- Cervin, A. (2005, July). Analysis of overrun strategies in periodic control tasks. In *Proc. 16th ifac world congress*. Prague, Czech Republic.
- Cervin, A., Lincoln, B., Eker, J., Årzén, K.-E., & Buttazzo, G. (2004, August). The jitter margin and its application in the design of real-time control systems. In *10th int. conf. on real-time and embedded computing systems and applications*. Göteborg, Sweden.
- Chen, T., & Francis, B. (1995). *Optimal sampled-data control systems*. Springer.
- Cloosterman, M., Hetel, L., Van De Wouw, N., Heemels, W., Daafouz, J., & Nijmeijer, H. (2010). Controller synthesis for networked control systems. *Automatica*, 46(10), 1584–1594.
- Cuenca, A., Garcia Gil, P. J., Arzén, K.-E., & Albertos, P. (2009, August). A predictor-observer for a networked control system with time-varying delays and non-uniform sampling. In *Proceedings of the european control conference, budapest, aug 2009*.
- Eker, J., Hagander, P., & Arzen, K.-E. (2000). A feedback scheduler for real-time controller tasks. *Control Engineering Practice*, 8(12), pp 1369–1378.
- Felicioni, F., Jia, N., Simonot-Lion, F., & Song, Y.-Q. (2010). Overload management through selective data dropping. In *Co-design approaches for dependable networked control systems*. ISTE-Wiley.
- Fiter, C., Hetel, L., Perruquetti, W., & Richard, J.-P. (2015). A robust stability framework for LTI systems with time-varying sampling. *Automatica*, 54, 56–64.
- Hansen, J., Hissam, S., & Moreno, G. A. (2009). Statistical-based WCET estimation and validation. In *9th international workshop on worst-case execution time (WCET) analysis ecrts'09*. Dagstuhl, Germany.

- Hespanha, J., Naghshtabrizi, P., & Xu, Y. (2007). A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1), 138–162.
- Hetel, L., Daafouz, J., & Lung, C. (2006, Oct). Stabilization of arbitrary switched linear systems with unknown time-varying delays. *IEEE Transactions on Automatic Control*, 51(10), 1668 - 1674.
- Hetel, L., Fiter, C., Omran, H., Seuret, A., Fridman, E., Richard, J.-P., & Niculescu, S.-I. (2017). Recent developments on the stability of systems with aperiodic sampling: An overview. *Automatica*, 76, 309 - 335.
- Hu, B., & Michel, A. N. (2000). Stability analysis of digital feedback control systems with time-varying sampling periods. *Automatica*, 36(6), 897 - 905.
- Khaitan, S. K., & McCalley, J. D. (2015). Design techniques and applications of Cyberphysical Systems: A survey. *IEEE Systems Journal*, 9(2), 350–365.
- Lee, E. A. (2008). Cyber Physical Systems: Design challenges. In *11th IEEE Int. Symposium on Object and Component-oriented Real-time Distributed Computing (ISORC)*.
- Lucia, S., Kögel, M., Zometa, P., Quevedo, D. E., & Findeisen, R. (2016). Predictive control, embedded cyberphysical systems and systems of systems – a perspective. *Annual Reviews in Control*, 41, 193–207.
- Moyne, J., & Tilbury, D. (2007). The emergence of industrial control networks for manufacturing control, diagnostics, and safety data. *Proceedings of the IEEE*, 95(1), 29–47.
- Murray, R. M., Åström, K. J., Boyd, S. P., Brockett, R. W., & Stein, G. (2003, Apr). Future directions in control in an information-rich world. *IEEE Control Systems Magazine*, 23(2).
- Olm, J., Ramos, G., & Costa-Castelló, R. (2011, 6). Stability analysis of digital repetitive control systems under time-varying sampling period. *Control Theory Applications, IET*, 5(1), 29 -37.
- Qiu, J., Gao, H., & Ding, S. X. (2016). Recent advances on fuzzy-model-based nonlinear networked control systems: A survey. *IEEE Transactions on Industrial Electronics*, 63(2), 1207–1217.
- Qiu, J., Wei, Y., & Karimi, H.-R. (2015). New approach to delay-dependent h_1 control for continuous-time Markovian jump systems with time-varying delay and deficient transition descriptions. *Journal of the Franklin Institute*, 352(1), 189–215.
- Rajkumar, R., Lee, I., Sha, L., & Stankovic, J. (2010). Cyber-physical systems: The next computing revolution. In *Acm design automation conference*. Anaheim, California, USA.
- Robert, D., Sename, O., & Simon, D. (2006). Synthesis of a sampling period dependent controller using LPV approach. In *5th IFAC Symposium on Robust Control Design Rocond'06*. Toulouse, France.
- Robert, D., Sename, O., & Simon, D. (2010). An H_∞ LPV design for sampling varying controllers : experimentation with a T inverted pendulum. *IEEE Trans. on Control Systems Technology*, 18(3), 741–749. <http://hal.archives-ouvertes.fr/hal-00448496/en/>.
- Roche, E., Sename, O., & Simon, D. (2012). An LFR approach to varying sampling control of LPV systems: application to AUVs. In *1st International Conference on Systems and Computer Science*. Villeneuve d'Ascq, France.
- Roche, E., Sename, O., Simon, D., & Varrier, S. (2011). A hierarchical varying sampling h_∞ control of an AUV. In *IFAC World Congress*. Milano, Italy.
- Sala, A. (2005). Computer control under time-varying sampling period: An LMI gridding approach. *Automatica*, 41(12), 2077–2082.
- Sala, A., Cuenca, Á., & Salt, J. (2009). A retunable PID multi-rate controller for a networked control system. *Information Sciences*, 179(14), 2390–2402.
- Shirazi, F. A., Mohammadpour Velni, J., & Grigoriadis, K. M. (2011, feb.). An LPV design approach for voltage control of an electrostatic MEMS actuator. *Microelectromechanical Systems, Journal of*, 20(1), 302 -311.
- Skogestad, S., & Postlethwaite, I. (2005). *Multivariable feedback control: analysis and design*. John Wiley and Sons.
- Yoneyama, J. (2010). Robust control of uncertain fuzzy systems under time-varying sampling. *Fuzzy Sets and Systems*, 161(6), 859 - 871.