HAL
open science

On state-dependant sampling for nonlinear controlled systems sharing limited computational resources

Mazen Alamir

To cite this version:

Mazen Alamir. On state-dependant sampling for nonlinear controlled systems sharing limited computational resources. ECC 2007 - European Control Conference, Jul 2007, Kos, Greece. hal-00092340

HAL Id: hal-00092340

https://hal.science/hal-00092340v1

Submitted on 24 Jan 2007

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.

On state-dependant sampling for nonlinear controlled systems sharing limited computational resources

Mazen Alamir[*]

Abstract

In this paper, a framework for dynamic monitoring of sampling periods for nonlinear controlled systems is proposed. This framework is particularly adapted to the context of controlled systems sharing limited computational resources. The proposed scheme can be used in a cascaded structure with any feedback scheduling design. Illustrative examples are given to assess the efficiency of the proposed framework.

1 INTRODUCTION

The increasing use of Networked Control Systems (NECs) rises many challenging control-related problems. Among these, resource sharing is a crucial issue. Typically, this problem arises when several controlled systems share a computational facility in order to perform the control related tasks. These tasks can be performed with different quality levels that require different computational loads. The paradigm of *resource sharing* amounts to find an adaptation framework that enables to use the minimum computational load for the *prescribed level of task achievement*. This concern led both the control community [4, 5] and the real-time community [7] to adopt a hierarchical control structure (see Figure 1) where a feedback scheduler adapts the allowed sampling period h_s for each task s according to the CPU load U of the computing resources. Different schemes have been used to estimate the computational load, but mainly, this amounts to take the sum of the ratios C_s/h_s over all the tasks $s \in \{1, \ldots, n_{task}\}$, namely: $U := \sum_{s=1}^{n_{task}} \frac{C_s}{h_s}$ where

[*]M. Alamir is with CNRS-Laboratoire d'Automatique de Grenoble. mazen.alamir@inpg.fr

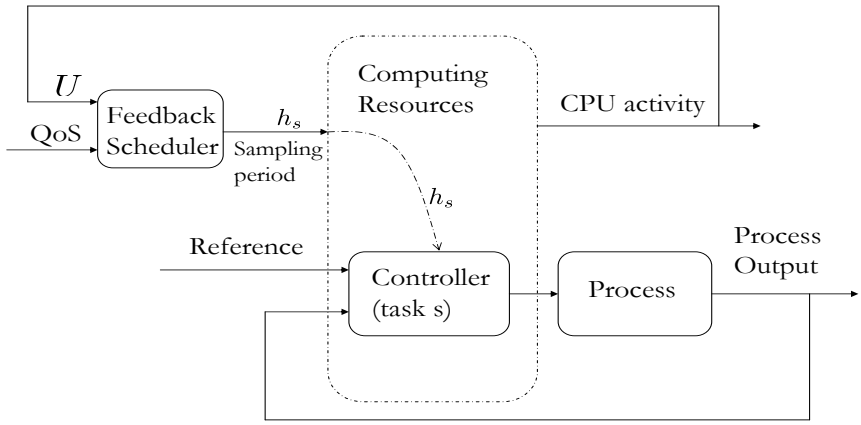


Fig. 1: Hierarchical control scheme [12, 11] in which the local state dependant sampling block proposed in this paper may be incorporated. h_s can serve as an exogenous signal giving the lower bound τ_{min} on the sampling period for the proposed state dependant sampling scheme used by the task s.

C_s is the computation time for the task s while h_s stands for the sampling period assigned to task s by the feedback scheduler. Therefore, by using the updating frequencies $f_s = 1/h_s$ as control variables the feedback scheduling amounts to control a linear system with $U = \sum_{s=1}^{n_{task}} C_s f_s$ as regulated variable and the f_i's as control inputs. This allows efficient and well established control theories handling delays [9] and/or uncertainties [11, 8] to be used in the design of the control scheduler.

Note that in the hierarchical structure of figure 1, once a sampling period h_s is assigned to the task s by the feedback scheduler based on the above over-simplified model, this task *completely* uses this updating rate even if given its state and the required performance level, this sampling period h_s is unnecessarily short.

The aim of this paper is to propose a local state dependent sampling strategy (at the task s level) that takes h_s delivered by any feedback scheduler as a minimal allowable sampling time $\tau_{min}^s = h_s$ and compute an effective *instantaneous* sampling time $\tau_s \geq h_s$ depending on the state of the system and

the required performance that is to be defined in some sense. By doing so, the effective load becomes $U^{'} = [\sum_{s=1}^{n_{task}} C_s/\tau_s] \leq U$ releasing computational capacity provided that the effective sampling periods τ_s are *returned* to the feedback scheduler in order to compute $U^{'}$. In this sense, the so obtained scheme can be viewed as a concrete implementation of the concept of *feed-forward* between the tasks and the scheduler as suggested by [1].

It is needless to note that the additional local computation load needed to perform such adaptation (at the task s level) must be lower that the gain it enables (otherwise, this would increase significantly C_s). This suggests the need for simple parametrized rules tuned through off-line dedicated optimization. Such a framework is proposed in the present paper.

It is worth emphasizing the fact that this paper does not propose a new feedback scheduling scheme but a local adaptation of the sampling period given a prescribed lower bound that is delivered by an existing feedback scheduler. That is why the paper focuses on what should be done at some task s level and therefore, the index s of the task is omitted.

The basic idea in the proposed solution is to use a state dependant sampling time scheme. This scheme is based on a well known idea according to which [2, 10]:

> An event based sampling can be more efficient than equidistant sampling. For instance, an integrator system driven by a white noise must be sampled 3-5 times faster using equidistant sampling than using event-based sampling to achieve the same output variance.

Surprisingly, despite such an impressive statement, too few works have been dedicated to state dependant sampling. Among them, let us mention [6, 3] where the cases of linear and nonlinear globally Lypschitz systems have been respectively considered form the only classical stability point of view.

The paper is organized as follows: First definitions and notation are given in section 2. The state dependant sampling scheme is derived in section 3. Section 4 explains how this scheme can be concretely used on-line. Finally, section 5 shows an illustrative example of the efficiency of the proposed scheme in handling the (stability/performance)-(computation load) trade-off.

2 DEFINITIONS & NOTATION

Consider general nonlinear systems given by

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

where x and u stand for the state and the control input respectively. It is assumed that (1) admits a state feedback control $K^* : \mathbb{R}^n \to \mathbb{R}^m$ such that the *ideal* resulting closed-loop dynamic

$$\dot{x} = f_{cl}(x) := f(x, K^*(x)), \tag{2}$$

is asymptotically stable in the Lyapunov sense with some invariant region of attraction $\mathcal{X} \subset \mathbb{R}^n$. Namely, there exists a continuous radially unbounded positive definite function $V : \mathbb{R}^n \times \mathbb{R}_+$ such that:

$$\forall x \in \mathcal{X} \quad L_{f_{cl}} V(x) < 0 \qquad (\mathcal{X} \text{ invariant under (2)}), \tag{3}$$

where $L_{f_{cl}} V$ is the Lie-derivative of V along the closed-loop vector field f_{cl}. Generally, the invariant region of attraction $\mathcal{X}$ is given by :

$$\mathcal{X} \;:=\; \mathbf{B}_V(v^c_{max}) := \Big\{ x \in \mathbb{R}^n \quad | \quad V(x) \leq v^c_{max} \Big\}, \tag{4}$$

since in this case, the invariance of $\mathcal{X}$ is a direct consequence of the inequality in (3).
In this paper, a sampled controller is derived from K^* by using a state dependent sampling period, namely:

$$u(t) \;=\; K^*(x(t_k)) \quad \forall t \in [t_k, t_{k+1}[\tag{5}$$

$$t_{k+1} \;=\; t_k + \hat{\tau}(p(t_k), x(t_k)) \quad , \quad t_0 = 0, \tag{6}$$

where the varying sampling period $\hat{\tau}(x, p)$ depends on the values at the decision instants t_k of both the state x and some exogenous input $p = (\tau_{min}, d)$ that expresses resource sharing-related concerns, namely, a minimum allowable sampling period τ_{min} and a minimum prescribed level of performance. The vector p is referred to as the *context parameter*. The concrete definition of the map $\hat{\tau}(\cdot, \cdot)$ as well as the definition of the so called *context parameter* $p = (\tau_{min}, d)$ is given in the following section.

In what follows, $X^*(\cdot, x_0)$ denotes the trajectory of the ideal continuous closed-loop dynamics (2) starting from initial state x_0. Given any function $g(\cdot)$ of the state, $g^*(t, x_0)$ is used to shortly refer to $g(X^*(t, x_0))$.

3 DEFINITION OF THE STATE DEPENDANT SAMPLING MAP $\hat{\tau}(\cdot,\cdot)$

3.1 Sketch of the key idea

For each state x and each real $\delta \in [0, \delta_{max}]$, let us consider the following sampling period:

$$\bar{\tau}(x,\delta) := Sat_{\tau_{min}}^{\tau_{max}}\left(\frac{\delta}{\|K^*(x)\|}\right) \quad ; \quad \delta \in [0, \delta_{max}]. \tag{7}$$

where K^* is the state feedback invoked in section 2 while $Sat_{\tau_{min}}^{\tau_{max}}(\cdot)$ is the saturation map with lower bound τ_{min} and upper bound τ_{max}. Recall that τ_{min} is delivered by the feedback scheduler while $\tau_{max} \geq \tau_{min}$ is the maximum allowable sampling period given the the bandwidth of exogenous disturbances. τ_{max} is assumed to be given once for all throughout the paper. The rationale behind the definition (7) is that when the control input takes high values, small sampling periods have to be used. Similarly, small control inputs denote almost *non critical* and near stability configurations allowing high sampling periods to be applied.

In this section, a somehow optimal update of the parameter δ invoked in (7) is derived depending on the state x and the context parameter p, namely:

$$\delta = \delta^{opt}(p,x) \tag{8}$$

Injecting this updating rule in (7) yields the state dependent sampling map invoked in section 2, namely:

$$\hat{\tau}(p,x) := \bar{\tau}(x, \delta^{opt}(p,x)) \tag{9}$$

The remainder of this section explains how the map $\delta^{opt}(p,x)$ is derived in order to handle issues such as stability, performance and computational load trade-off.

3.2 Derivation of the map $\delta^{opt}(p,x)$

With the definition (7), for any given value of the pair $(x_0, \delta) \in \mathcal{X} \times [\tau_{min}, \tau_{max}]$, a closed-loop trajectory is uniquely defined as follows: At initial instant $t_0 = 0$, the control $K^*(x(t_0)) = K^*(x_0)$ is computed and applied to the system until the next decision instant $t_1 := t_0 + \bar{\tau}(x(t_0), \delta)$. The new control $K^*(x(t_1))$ is computed and applied during the next sampling period

$[t_1, t_1+\bar{\tau}(x(t_1), \delta)]$ and so on. The resulting closed-loop trajectory so obtained is denoted hereafter by $X(\cdot, x_0, \delta, \tau_{min})$ since it depends on both the initial state x_0 and the parameters δ and τ_{min} used in (7). Again, given any function g of the state, the notation $g(t, x_0, \delta, \tau_{min})$ is used hereafter to shortly refer to $g(X(t, x_0, \delta, \tau_{min}))$. Moreover, the corresponding set of decision instants $\{t_k\}_{k\geq 0}$ is denoted by $\mathcal{T}(x_0, \delta, \tau_{min})$.

Depending on the current state, the value of δ must be such that the computational load is minimized while preserving the stability and while achieving the prescribed level of performance. Rigorously speaking, this makes δ state dependent. This would lead to a huge amount of off-line computation and on-line memory. That is the reason why the state space is partitioned using the level sets $\mathbf{B}_V(v)$ for $v \geq 0$ [see (4)]. More precisely, the requirements that determine the choice of δ when the current state is inside $\mathbf{B}_V(v)$ are described as follows:

Stability-related requirement. When the state is inside $\mathbf{B}_V(v)$ for some $v \leq v_{max}$, this requirement is addressed using the following condition:

$$S_T(\delta, \tau_{min}, v) := \max_{V(x_0)\leq v} \Big[V(T, x_0, \delta, \tau_{min}) \Big] \leq v_{max} \tag{10}$$

where V is the Lyapunov function invoked in section 2 while $T > 0$ is some prediction horizon. Indeed, this condition ensures that any trajectory starting inside $\mathbf{B}_V(v)$ returns into $\mathcal{X} := \mathbf{B}_V(v_{max})$ after T time units.

Performance-related requirement. In order to address this feature, a minimal performance level $d \in [0, 1]$ has to be defined. In this paper, the admissible performance index is computed by comparing the trajectory under sampling to the *ideal* sampling-free trajectory X^*, namely:

$$P_T(\delta, \tau_{min}, v) := \min_{V(x_0)\leq v} \Big[\frac{1}{1 + \|X(\cdot, x_0, \delta, \tau_{min}) - X^*(\cdot, x_0)\|_{L_2}} \Big] \geq d \tag{11}$$

where the L_2 norm is considered on the prediction horizon $[0, T]$.

Computation load related requirement. This requirement is handled assuming that the CPU time necessary to compute the feedback $K^*(x(t_k))$ at some decision instant t_k is independent of $x(t_k)$. This is typically the case when the control law $K^*(\cdot)$ is available analytically[1]. Let us denote by τ_c

[1] This is not the case for instance if the control law K^* is computed by a predictive

this computation time. This suggests that the maximal computation load during the prediction horizon $[0, T]$ only depends on the number of decision instants (t_k) that *occur* during $[0, T]$, namely:

$$C_T(\delta, \tau_{min}, v) := \max_{V(x_0) \leq v} \left[card \Big\{ \tau \in \mathcal{T}(x_0, \delta, \tau_{min}) \quad | \quad \tau \leq T \Big\} \right], \tag{12}$$

Consequently, given the context parameter $p := (\tau_{min}, d) \in [0, \tau_{max}] \times [0, 1]$ corresponding to some minimal allowable sampling period τ_{min} and a prescribed level of performance $d \in [0, 1]$, the following constrained optimization problem is defined for all sets $\mathbf{B}_V(v)$ of current states:

$$P(p, v) \ : \ \min_{\delta \in [0, \delta_{max}]} \Big[C_T(\delta, p_1, v) \Big] \text{under} \quad \begin{pmatrix} P_T(\delta, p_1, v) \geq p_2 \\ S_T(\delta, p_1, v) \leq v_{max} \end{pmatrix} \tag{13}$$

Let us denote by $\delta^{opt}(p, v)$ a solution (if any) to the optimization problem $P(p, v)$. This means that for any initial state that lies in $\mathbf{B}_V(v)$, the use of $\delta = \delta^{opt}(p, v)$ in (7) in which $\tau_{min} = p_1$ and $d = p_2$ leads to a stable behavior that meets the prescribed level of performance while minimizing the **worst case computation load**. Note finally that replacing v in $\delta^{opt}(p, v)$ by $V(x)$ yields the definition of $\delta^{opt}(p, x)$ invoked in section 3.1 with a straightforward abuse of notation.

4 OFF-LINE COMPUTATION FOR ON-LINE USE

In order to derive a finite dimensional information to be computed off-line, the set of possible values for (p, v) is given by :

$$\mathcal{A} := \underbrace{\{\tau_{min}^{(1)}, \ldots, \tau_{min}^{(n_1)}\} \times \{d^{(1)}, \ldots, d^{(n_2)}\}}_{\mathcal{A}_p} \times \underbrace{\{v^{(1)}, \ldots, v^{(n_v)}\}}_{\mathcal{A}_v},$$

where

- $\tau_{min}^{(1)}, \ldots, \tau_{min}^{(n_1)} \in [0, \tau_{max}]$ are the minimal sampling periods the feedback scheduler can assign to the system.
- $d^{(1)}, \ldots, d^{(n_2)} \in [0, 1]$ are the admissible performance levels that may be assigned.
- $v^{(1)} < \cdots < v^{(n_v)}$ are level set values such that $v^{(n_v)} = v_{max}$.

control approach since in this case, the complexity of the underlying optimization may heavily depend on the state

Once the $(n_1 \cdot n_2 \cdot n_v)$ static games $P(p, v)$ are solved off-line for all $(p, v) \in \mathcal{A}$, a three dimensional table of values can be obtained, namely

$$\mathcal{T}_\Delta := \left\{ \hat{\delta}_{ijq} \right\}_{(i,j,q) \in \{1,\ldots,n_1\} \times \{1,\ldots,n_2\} \times \{1,\ldots,n_v\}} \tag{14}$$

where $\hat{\delta}_{ijq}$ is the solution of $P((\tau_{min}^{(i)}, d^{(j)}), v^{(q)})$.
Finally, the map $\hat{\tau}(\cdot, \cdot)$ invoked in section 2 [see equations (5)-(6)] is given by :

$$\begin{aligned} \hat{\tau}(p, x) &:= \bar{\tau}(x, \hat{\delta}_{ijq}) \quad \text{where } (i, j, q) \text{ are such that} \\ p = (\tau_{min}^{(i)}, d^{(j)}) &\quad \text{and} \quad q := \min\{s \mid V(x) \leq v^{(s)}\}. \end{aligned} \tag{15}$$

Namely, the optimal parameter δ for the sampling map is obtained based on the current value of the context parameter p and the smallest region $\mathbf{B}_V(v^{(s)})$ that contains the current state. Therefore, if only one value for v is used ($n_v = 1$), the parameter δ would be only dependent on p whatever is the current state (note however that even in this case, the sampling period remains state dependant through (7) even though δ is state independent).

Note that the computation of $\hat{\tau}(p, x)$ is quasi instantaneous since it amounts to perform a look-up table operation. The resulting state dependent sampling scheme is schematically depicted on figure 2. This is detailed in the following section.

4.1 Concrete use of the table $\mathcal{T}_\Delta$

To summarize the above presentation of the state dependent sampling principle, assume that the table $\mathcal{T}_\Delta$ is available [see (14)]. Its on-line use is explained as follows:

- At each decision instant t_k, the feedback scheduler delivers the context parameter vector $p(t_k) = (\tau_{min}^{(i)}, d^{(j)})$ containing the minimal allowable sampling time as well as the required minimal performance level. This determines the indices $i(t_k)$ and $j(t_k)$.

- The remaining index q of $\hat{\delta}_{ijq}$ is related to the minimal value of $s \in \{1, \ldots, n_v\}$ such that $x(t_k) \in \mathbf{B}_V(v^{(s)})$, namely :

$$q(t_k) := \min\{s \mid V(x(t_k)) \leq v^{(s)}\} \tag{16}$$

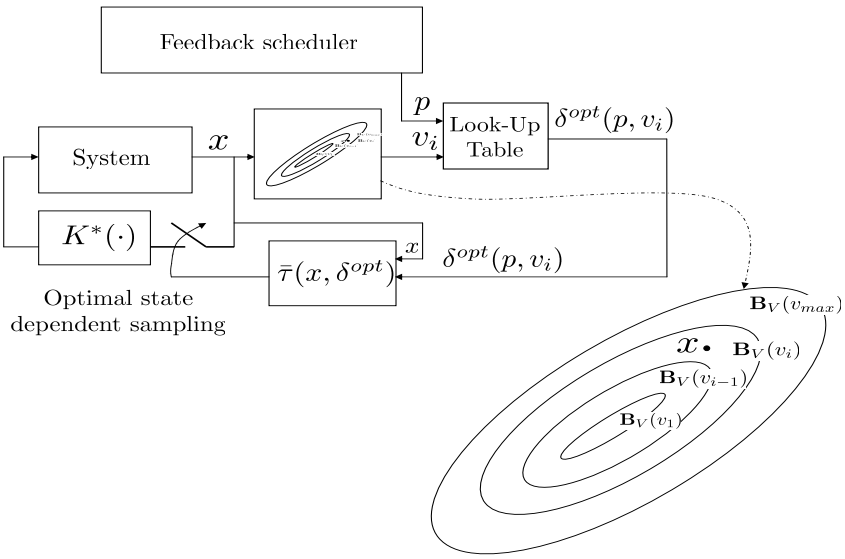


Fig. 2: Schematic view of the overall state dependent sampling scheme. The maximal $\mathbf{B}_V(v_i)$ that contains x is first computed to yield v_i, this together with the context parameter p delivered by the feedback scheduler are used to compute the optimal $\delta^{opt}(p, v_i)$. The latter is used in (7) in order to monitor the state dependent sampling rule.

- Having at hand the indices i, j, and q, the next sampling period is given by :

$$\hat{\tau}(p(t_k), x(t_k)) = Sat_{\tau_{min}}^{\tau_{max}} \left(\frac{\hat{\delta}_{ijq}(t_k)}{\|K^*(x(t_k))\|} \right)$$

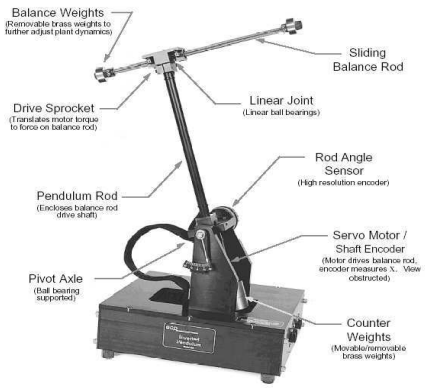
  during which the constant control $K^*(x(t_k))$ is applied to the system.

- At the next decision instant t_{k+1} given by

$$t_{k+1} = t_k + \hat{\tau}(p(t_k), x(t_k))$$

  the computation is repeated considering the new context vector $p(t_{k+1})$ (that gives $i(t_{k+1})$ and $j(t_{k+1})$) and the new location of the state (that gives $q(t_{k+1})$) and the procedure is repeated indefinitely.

(a)



(b)

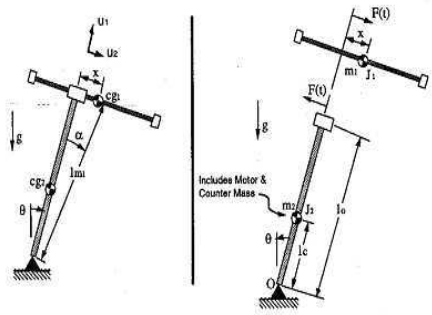


Fig. 3: The ECP-Pendulum (Model 505) description: (a) View of the experimental realization. (b) Definition of the parameters.

5 An illustrative example

5.1 The system model

Let us consider the system depicted on Figure 3. It consists of a pendulum rod that supports a sliding balance rod. The latter is driven via a belt and pulley which is controlled by a shaft connected to a DC-servo motor below the pendulum rod. Using $q = (r, \theta)^T$ to denote the vector of generalized coordinates, the Lagrange equations for the system can be given in the following standard form:

$$M(q)\ddot{q} + N(q, \dot{q})\dot{q} + G(q) = Su \tag{17}$$

where the inertia matrix $M(q)$, the Coriolis Matrix $N(q, \dot{q})$ and the gravitational term $G(q)$ are given by:

$$\begin{aligned}
M(q) \quad &: \quad = \begin{pmatrix} m_1 & m_1 l_0 \\ m_1 l_0 & J_1 + J_2 + m_1(l_0^2 + r^2) + m_2 l_c^2 \end{pmatrix} \\
N(q, \dot{q}) \quad &:= \quad \begin{pmatrix} 0 & -m_1 r\dot{\theta} \\ m_1 r\dot{\theta} & m_1 r\dot{r} \end{pmatrix} \quad ; \quad S := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\
G(q) \quad &:= \quad \begin{pmatrix} -m_1 g \sin\theta \\ -(m_1 l_0 + m_2 l_c) g \sin\theta - m_1 g r \cos\theta \end{pmatrix},
\end{aligned}$$

where all the parameters are depicted on Figure 3-(b). The numerical values of these parameters are given by $(m_1, m_2, J_1, J_2) = (0.213\ kg, 1.785\ kg, 0.01\ kg \cdot m^2, 0.0246\ kg \cdot m^2)$ and $(l_0, l_c, g) = (0.33\ m, -0.0463\ m, 9.81\ ms^{-2})$ the state

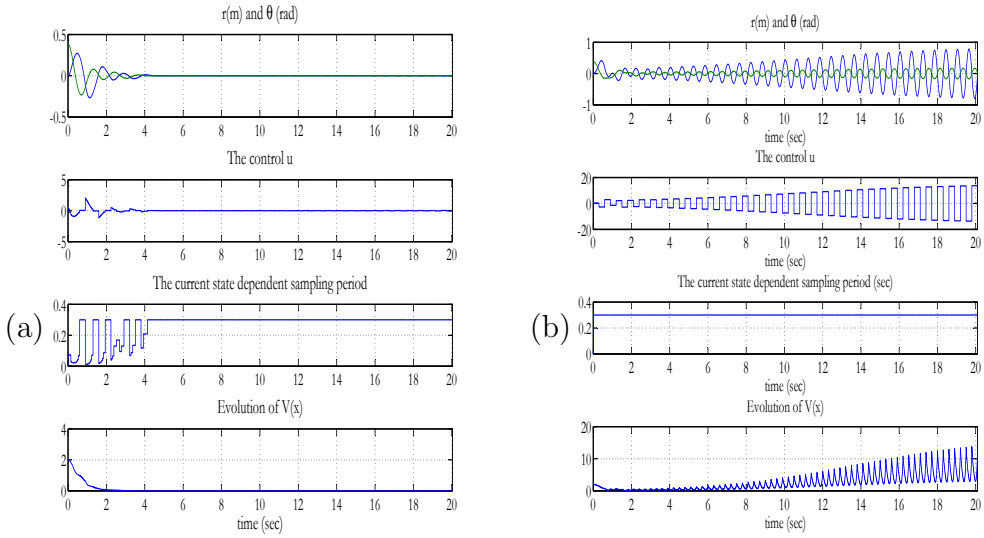


Fig. 4: State dependent sampling vs constant sampling schemes: (a) Behavior of the closed loop under state dependent sampling scheme with the parameters $(\tau_{min}, \tau_{max}, \delta) = (0.01\ s, 0.3\ s, 0.02)$. (b) Closed loop system under constant sampling period period $\tau = \tau_{max} = 0.3\ s$. Note that the state dependent sampling enables the stability to be recovered with an asymptotic computation load that would be incompatible with stability requirement under constant sampling scheme.

of the system is given by $x := (r, \theta, \dot{r}, \dot{\theta})^T$.

5.2 The control law K^* and the Lyapunov function V

The control law $K^*(\cdot)$ is designed based on the linearized system around the vertical position. This suggests the use of a quadratic Lyapunov function $V(x) = x^T P x$ where P is solution of the Lyapunov equation

$$\bar{A}^T P + P\bar{A} = -I \quad ; \quad \bar{A} := A - BK_0,$$

in which (A, B) are the linearized system's matrices while $K_0 \in \mathbb{R}^{1\times 4}$ is the state feedback that is computed using the LQR procedure with the following weighting matrices: $Q_1 = diag(10, 10, 0.1, 0.1)$ and $R_1 = 1$.

A typical scenario showing the relevance of state dependant sampling is shown on Figure 4.

5.3 Definition of the discrete set $\mathcal{A}$ and the remaining parameters

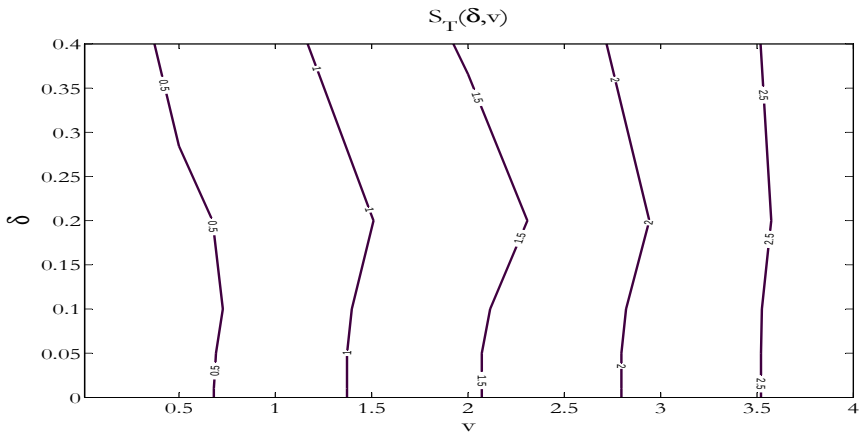


Fig. 5: Contours of the map $S_T(\delta, v)$ (for $\tau_{min} = 0.1\ s$). The value of $S_T(\delta, v)$ is lower than $v_{max} = 4$ assessing the stability of the state dependent closed loop system under this value of τ_{min} for all initial conditions in $\mathbf{B}_V(v_{max} = 4)$ and all values of $\delta \in [0, 0.4]$. Similar conclusions on the stability region can be obtained for $\tau_{min} = 0.05\ s$ and $\tau_{min} = 0.01\ s$.

The values used to built the set $\mathcal{A}$ are the following :

$$\tau_{min}^{(i)} = \{0.01\ s, 0.05\ s, 0.1\ s\} \quad i \in \{1, 2, 3\} \tag{18}$$

$$d^{(j)} = \{0.7, 0.8, 0.95\} \quad j \in \{1, 2, 3\} \tag{19}$$

$$v^{(q)} = \{0.01, 0.1, 0.25, 1, 2, 4\}\ ;\ q \in \{1, \ldots, 6\} \tag{20}$$

The possible values for the sampling parameter δ are given by :

$$\delta^{(s)} \in \Big\{0, 0.01, 0.1, 0.2, 0.4\Big\} \quad ; \quad s \in \{1, \ldots, 5\} \tag{21}$$

In order to compute the stability region $\mathbf{B}_V(v_{max})$, a parameter v_{max} is computed that makes the stability condition (10) satisfied for all possible values of τ_{min} and δ given by (18) and (21). The result is shown on Figure 5 from which it can be inferred that $v_{max} = 4$ is an admissible value leading to the region of attraction $\mathbf{B}_V(4)$.

In order to compute the table $\mathcal{T}_\Delta$ defined by (14), off-line optimizations are

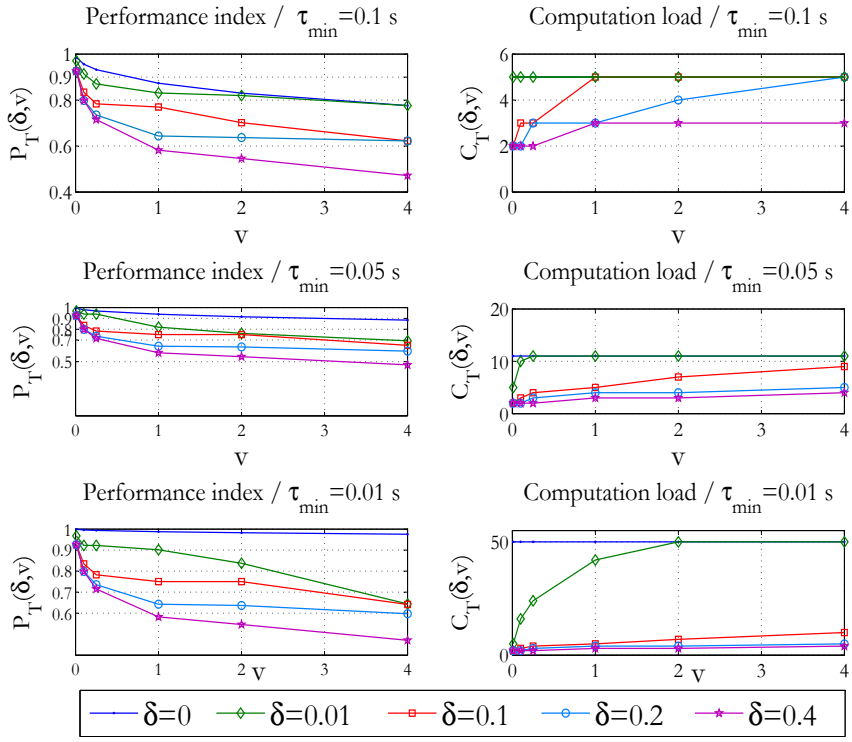


Fig. 6: The results of off-line computations of the maximal computation load $C_T(\delta^{(s)}, v^{(q)})$ given by (12) and the performance ratio $P_T(\delta^{(s)}, v^{(q)})$ given by (11) for the three possible values $\tau_{min}^{(i)}$ given by (18). The information contained in these curves enables to solve the static game (13) for all admissible value of $(\tau_{min}, d, v) \in \mathcal{A}$ in order to construct the table $\mathcal{T}_\Delta$

needed. This amounts to compute $P_T(\delta^{(s)}, \tau_{min}^{(i)}, v^{(q)})$ and $C_T(\delta^{(s)}, \tau_{min}^{(i)}, v^{(q)})$ for all possible values of the triplet $(\delta^{(s)}, \tau_{min}^{(i)}, v^{(q)})$ in the set of possible values given above. The corresponding computation results are shown on figure 6. These results suggests the following remarks:

✓ Note first that when $\delta = 0$, according to the state dependent sampling law (7), the sampling period become state independent and the minimum allowed sampling period $\tau_{min}^{(i)}$ is constantly used in a fixed sampling period framework. The three corresponding curves (one on each of the sub-figures on the left of Figure 6) gives the somehow maximum achievable performance ratio under the corresponding assigned value of τ_{min}. For instance, when $\tau_{min} = 0.1\ s$ is assigned to the system

by the feedback scheduler, if at some instant t_k, $V(x(t_k)) = 4$, then the maximal achievable performance ratio is slightly lower than 0.8 (see sub-figure (1,1) for Figure 6).

✓ Figure 6 clearly shows a quite expected fact according to which, even under state dependent sampling, the computation load decreases when the minimal allowable sampling period τ_{min} increases (See the sub-figures on the right of Figure 6). In particular, the computation load never exceeds 5 under $\tau_{min}^{(1)} = 0.1\ s$ while it may exceed 40 for states x such that $V(x) = 1$ and $\delta = \delta^{(2)} = 0.01$ (See sub-figure (2,3) of Figure 6).
The results of the off-line optimization depicted in Figure 6 are then used to construct the table $\mathcal{T}_\Delta = \{\hat{\delta}_{ijq}\}_{ijq}$ defined in (14). The result is presented on Figure 7.

5.4 Simulations

In order to assess the efficiency of the proposed state dependent sampling scheme, the system is simulated in closed-loop under the the proposed control scheme and in the presence of unmeasured periodic excitation input $w(t)$ that simulates sudden changes in the angular velocity (chocs), namely:

$$\dot{x} = \frac{d}{dt}\begin{pmatrix} r \\ \theta \\ \dot{r} \\ \dot{\theta} \end{pmatrix} = f(x,u) + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} w(t) \tag{22}$$

with the following definition of $w(t)$:

$$w(t) = \begin{cases} a & \text{if } E(t/T_w) \leq \tau_w \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

where T_w is the period of w, τ_w is the impulse duration while a is the impulse amplitude.

Two scenarios are proposed in order to illustrate the two following features:

1) Response of the closed-loop systems to changes in the performance index under constant allowable sampling period $\tau_{\min}$ (Figure 8).

$(\hat{\delta}_{1jq}, \hat{C}_T)$

$d^{(j)}$ \ $v^{(q)}$	0.01	0.1	0.25	1	2	4
0.95	(0.01,5)	(0,5)	X	X	X	X
0.8	(0.4,2)	(0.4,2)	(0.01,5)	(0.01,5)	(0.01,5)	X
0.7	(0.4,2)	(0.4,2)	(0.4,2)	(0.1,5)	(0.01,5)	(0.01,5)

Optimal δ for $\tau_{min} = 0.1\ s$ and different values of $d^{(j)}$ and $v^{(q)}$

$(\hat{\delta}_{2jq}, \hat{C}_T)$

$d^{(j)}$ \ $v^{(q)}$	0.01	0.1	0.25	1	2	4
0.95	(0.01,5)	(0,11)	(0,11)	X	X	X
0.8	(0.4,2)	(0.1,3)	(0.01,11)	(0.01,11)	X	X
0.7	(0.4,2)	(0.4,2)	(0.4,2)	(0.1,5)	(0.1,7)	(0,11)

Optimal δ for $\tau_{min} = 0.05\ s$ and different values of $d^{(j)}$ and $v^{(q)}$

$(\hat{\delta}_{3jq}, \hat{C}_T)$

$d^{(j)}$ \ $v^{(q)}$	0.01	0.1	0.25	1	2	4
0.95	(0.01,5)	(0,50)	(0,50)	(0,50)	(0,50)	(0,50)
0.8	(0.4,2)	(0.1,3)	(0.01,25)	(0.01,42)	(0.01,50)	(0,50)
0.7	(0.4,2)	(0.4,2)	(0.4,2)	(0.1,5)	(0.1,7)	(0,50)

Optimal δ for $\tau_{min} = 0.01\ s$ and different values of $d^{(j)}$ and $v^{(q)}$

Fig. 7: Computation of the table $\mathcal{T}_\Delta$ of optimal sampling parameter $\hat{\delta}_{ijq}$ defined in (14). The notation X is used to denote that the prescribed level of performance is impossible to achieve under the corresponding $\tau_{min}^{(i)}$ and the state subset $\mathbf{B}_V(v^{(j)})$. For each pair $(d^{(j)}, v^{(q)})$, the optimal parameter $\hat{\delta}_{ijq}$ as well as the corresponding optimal cost are given.

In this scenario, the following parameters are used in the definition of the disturbance signal w:

$$T_w = 1\ s \quad ; \quad \tau_w = 0.02 \quad ; \quad a = 20$$

Since the minimal allowable sampling period $\tau_{min} = 0.05\ s$ delivered by the feedback scheduler is assumed to be fixed, the computational load can be monitored using the admissible performance index d. The latter is defined as follows

$$d(t) = \begin{cases} 0.95 & \text{if } t \in [0, 10] \\ 0.7 & \text{if } t \in]10, 20] \\ 0.8 & \text{otherwise} \end{cases}$$

At the beginning, over the time interval $[0, 10]$, the desired performance level being fixed to $d = 0.95$, the controller uses entirely the minimum allowable sampling period. As soon as the performance index is decreased to $d = 0.7$ over the time interval $[20, 30]$, the controller degrades the quality of the control in order to decrease the computational load. This can be viewed on the sub-figure (3,1) of figure 8 where the evolution of the number of decision instants (t_k) is shown. Indeed, the slope of this curve (representing the mean number of decision instant per unit time) is significantly reduced over $[10, 20]$ when compared to its value on the first interval $[10, 20]$. Again, when the desired performance is slightly increased from 0.7 to 0.8, the inverse result is obtained (the achieved performance index increases together with the computation load).

2) Response of the closed-loop system to changes in the minimum allowable sampling period τ_{min} under constant admissible performance index $\mathbf{d = 0.95}$. (Figure 9-10).

In this scenario, the following parameters are used in the definition of the disturbance signal:

$$T_w = 4\ s \quad ; \quad \tau_w = 0.02 \quad ; \quad a = 5$$

Figures 9 and 10 show two scenarios for different values of the performance index $d = 0.95$ and $d = 0.7$ respectively. In both scenarios, an allowable minimum sampling time $\tau_{min} = 0.01\ s$ is used over the first 20 seconds while $\tau_{min} = 0.1\ s$ is used over the next 20 seconds.

Figure 9 shows the response of the closed-loop systems when the performance index $d = 0.95$ is assigned by the feedback scheduler. Note how increasing the allowable minimum time from $\tau_{min} = 0.01$ to $\tau_{min} = 0.1$ enables a tremendous reduction of the computation load (see the rate of variation of the number of decision instants on the sub-figure (3,1) of figure 9) without a significant effects on the achieved performance index.
Figure 10 shows the same results under the performance index $d = 0.7$. Contrary to the result of Figure 9, here increasing the minimum allowable sampling period has no significant effect on the computational load since the state dependent sampling scheme avoids using the allowable minimum sampling period by taking $\delta = 0.4$ leading to high effective on-line sampling period even when $\tau_{min} = 0.01\ s$ is used. However, the use of low performance index $d = 0.7$ clearly reduces the computational load when compared to the results of figure 9. Note that the results of the scenario depicted on

figure 9 are recalled on the sub-figures (3,1) and (3,2) in order to make the comparison easier.

6 Conclusion

In this paper, a state dependent sampling scheme has been proposed that enables to reduce the computational load while preserving stability and achieving (when possible) the admissible performance level. The proposed framework can be inserted in a hierarchical structure where a feedback scheduler feeds the scheme with a maximum allowable sampling time. Simulations on a realistic system suggests that this may significantly reduce the effective computational load releasing so computational resources for additional tasks.

References

[1] K. E. Arzén, A. cervin, and D. Henriksson. Resource-constrained embedded control systems: Possibilities and research issues. In *Co-Design for Real Time Embedded Systems (CERTES'03)*, 2003.

[2] K. J. Astrm and B. Bernhardsson. Comparison of periodic and event based sampling for first-order stochastic systems. In *Proceedings of the 14th IFAC World Congress. Beijing, P. R. China*, 1999.

[3] P. Astuti and D. Williamsons M.Corless. Digital control of nonlinear systems with state dependent sampling times. In *Proceedings of the third IFAC Symposium on Nonlinear Control Systems Design*, pages pp. 859–864, California, 1995.

[4] A. Cervin, J. Eker, B. Bernhardson, and K.E. Arzen. Feedback-feedforward scheduling of control tasks. *Real-time Systems*, 23(1-2):25–53, 2002.

[5] J. Eker, P. Hagander, and K.E. Arzen. A feedback scheduler for real-time controller tasks. *Control Engineering Practice*, 8(12):1369–1378, 2000.

[6] L. Guzzella and F. Kraus. Stability of discrete-time systems with state-dependent sampling time. In *Proceedings of the European Control Conference*, volume Vol. 4, pages pp. 3642–3648, 1995.

[7] C. Lu, J.A. Stankovic, S.H. Son, and G. Tao. Feedback control real-time scheduling: Framework, modeling and algorithms. *Real-time Systems Journal*, 23(1-2):85–126, 2002.

[8] D. Robert, O. Sename, and D. Simon. Sampling period dependent rst controller used in control/scheduling co-design. In *Proceeding of the IFAC World Congress, Praha, Czech Republic*, 2005.

[9] O. Sename, D. Simon, and D. Robert. Feedback scheduling for real-time control of systems with communication delays. In *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, Lisbon, Portugal*, 2003.

[10] D. Simon, E. Castillo, and P. freedman. Design and analysis of synchronisation for real-time closed-loop control in robotics. *IEEE Transactions on Control Systems Technology*, 6(4):445–461, 1998.

[11] D. Simon, D. Robert, and O. Sename. Robust control/scheduling co-design: application to robot control. In *Proceedings of the RTAS'05 11th IEEE Real-Time and Embedded Technology and Applications Symposium, San Francisco*, 2005.

[12] D. Simon, D. Robert, O. Sename, and O. Testa. Real-time and delay-dependent control co-design through feedback scheduling. In *Proceedings of the CERTS'03 Co-design in Embedded Real-time Systems, Porto, july*, 2003.

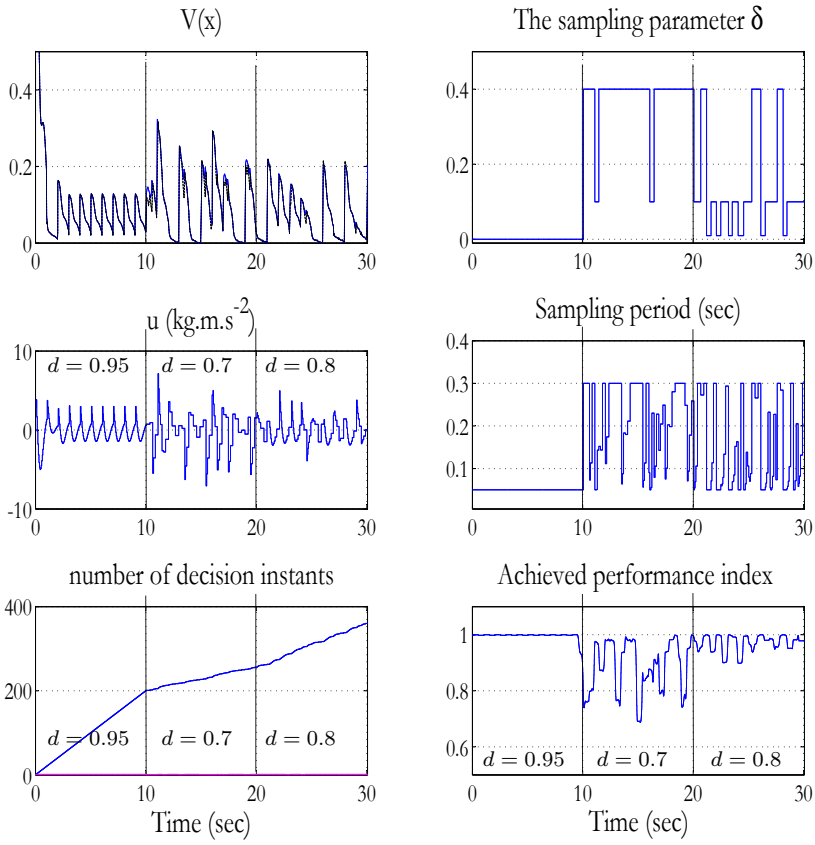


Fig. 8: **Changing the admissible performance level under constant $\tau_{\min} = 0.05$ s.** The disturbance signal w is given by (23) with $(T_w, \tau_w, a) = (1\ s, 0.02, 20)$. Three phases can be observed each of 10 seconds with different values of the performance index $d \in \{0.95, 0.7, 0.8\}$. Note how the computation load (the rate of the variation of the number of decision instants) as well as the truly achieved performance vary correspondingly.

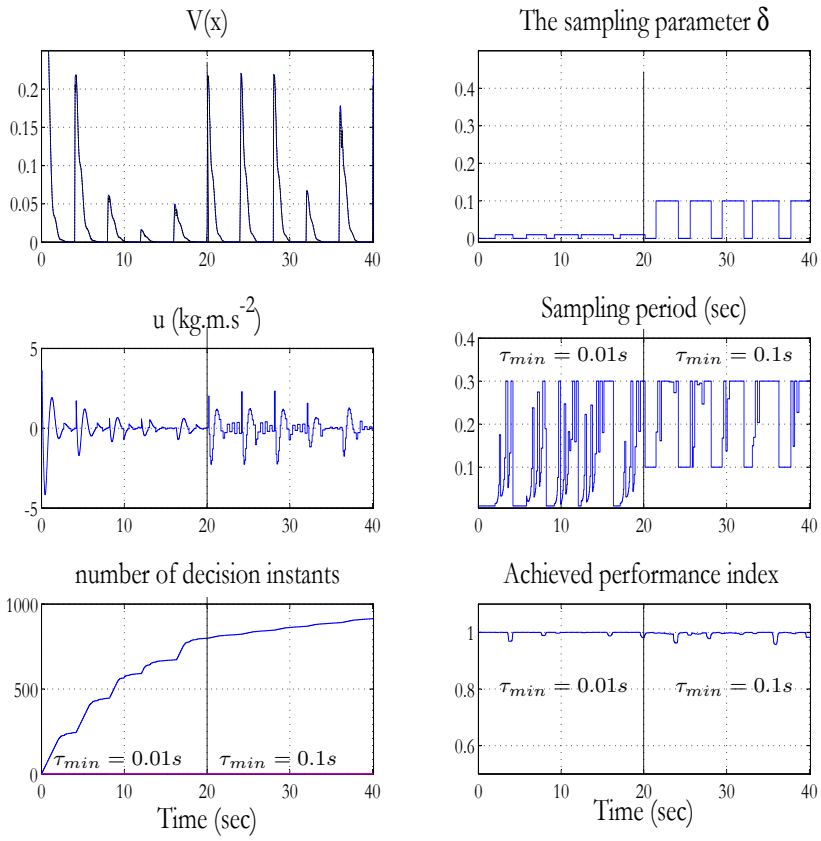


Fig. 9: **Changing the minimum allowable $\tau_{\min}$ for a given admissible performance level $\mathbf{d} = 0.95$.** The disturbance signal w is given by (23) with $(T_w, \tau_w, a) = (4\ s, 0.02, 5)$. Note how increasing the minimum allowable sampling period enables a tremendous reduction in the computational load (the rate of variation of the number of decision instants) without a significant reduction in the achieved performance index.

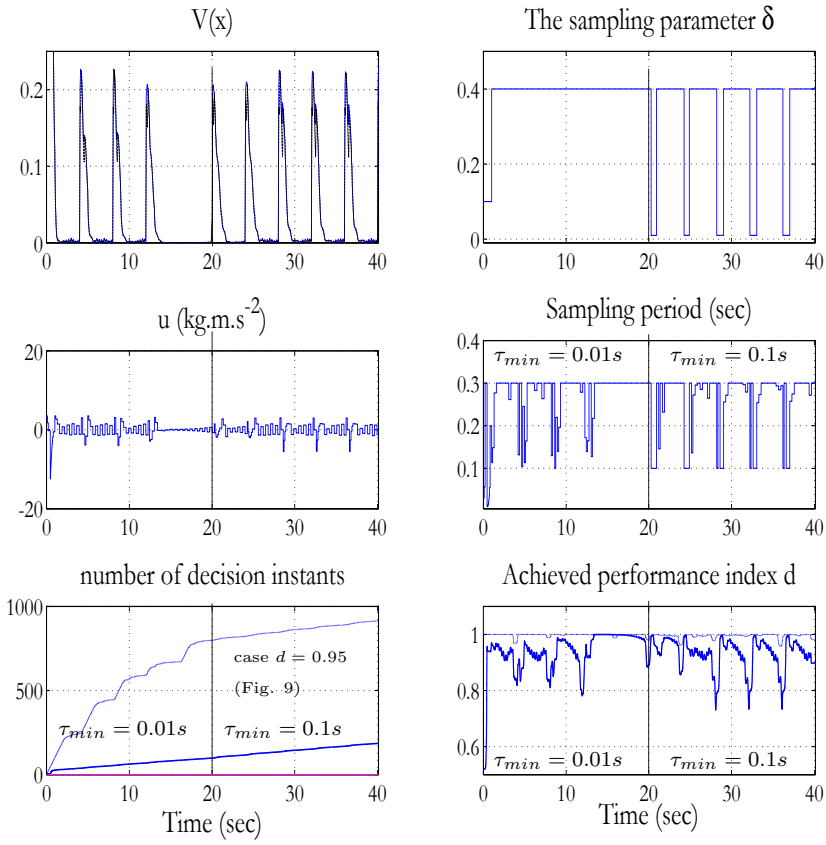


Fig. 10: **Changing the minimum allowable $\tau_{\min}$ for a given performance level $\mathbf{d = 0.7}$.** The disturbance signal w is given by (23) with $(T_w, \tau_w, a) = (4\ s, 0.02, 5)$. Contrary to the result of Figure 9, here increasing the minimum allowable sampling period has no significant effect on the computational load since the state dependent sampling scheme already avoids using the allowable minimum sampling period by taking $\delta = 0.4$ leading to high effective on-line sampling period even when $\tau_{min} = 0.01\ s$ is used. However, the use of low performance index $d = 0.7$ clearly reduces the computational load when compared to the results of figure 9. Note that the results of the scenario depicted on figure 9 are recalled on the sub-figures (3,1) and (3,2) in order to make the comparison easier.