Scheduling of controllers' update-rates for residual bandwidth utilization

Majid Zamani[1], Soumyajit Dey[2], Sajid Mohamed[2], Pallab Dasgupta[2], and Manuel Mazo Jr[3]

[1] Technische Universität München
Munich, Germany
zamani@tum.de
[2] Indian Institute of Technology
Kharagpur, India
{soumya,sajidm,pallab}@cse.iitkgp.ernet.in
[3] Delft University of Technology
Delft, The Netherlands
m.mazo@tudelft.nl

Abstract. We consider the problem of incorporating control tasks on top of a partially loaded shared computing resource for which the current task execution pattern is characterizable using a window based pattern. More specifically, we consider that the control task to be scheduled is allowed to switch between multiple controllers, each with different associated sampling rate, in order to adjust its requirement of computational bandwidth as per availability.
For the control task, we provide a novel control theoretic analysis that derives a Timed Automata (TA) based specification of allowable switchings among the different controller options while retaining the asymptotic stability of the closed loop. Our scheduling scheme computes a platform level residual bandwidth pattern from individual task level execution patterns. We leverage the TA based controller specification and the residual bandwidth pattern in order to synthesize a Linearly Priced Timed Automata for which the minimum cost reachability solution provides realizable multi-rate control schedules. The provided scheduler not only guarantees the asymptotic stability of the control loop but also increases the robustness and control performance of the implementation by maximizing the bandwidth utilization.

1 Introduction

Traditionally, digital implementations of controllers employ constant periodic sampling and control update mechanisms. Additionally, the engineers designing such implementations tend to over-provision (communication and/or computing) bandwidth to the implemented controllers. Two main reasons that justify sampling as fast as possible in a controller implementation are as follows, (i) the

available we provide a construction to obtain the sequence of update intervals consuming the maximum amount of available bandwidth.

control engineer is allowed to design a controller with traditional continuous time tools without worrying about the specific selection of sampling times; **SD:** when we talk of control implementation on loaded ECU, the RTOS already restricts the allowable scheduling periods and, (ii) the faster the sampling, the quicker a controller can react to external disturbances to the system under control. However, current trends for the implementation of complex cyber-physical systems are shifting from traditional federated architectures, where each feature runs on a dedicated Electronic Control Unit (ECU), to integrated architectures, where multiple features execute on a shared ECU. Thus, instead of over-provisioning resources, these new architectures demand flexibility and efficiency in the use of resources. In a modern automobile, for instance, features may be engaged and disengaged dynamically depending on the state of the system. Take as an example the processing of data from the rear parking sensors, which is not necessary when the gear position is in the forward direction. When a feature is not engaged, the residual bandwidth made available by the tasks omitted for that feature can be potentially harnessed by the other features running on the same processor. Such plug-and-play nature of control features is recommended by modern automotive standards like AUTOSAR [24], and is also being adopted in cyber-physical system architectures beyond the automotive domain.

In the current paper we consider architectures in which a simple ECU executes a set of tasks. We assume that tasks are described by arrival patterns expressed in *Real Time Calculus* (RTC), and the scheduling scheme is known for the system. This allows us to find periodic upper bounds on the ECU utilization and compute a recurring pattern of bandwidth availability. We address the problem of scheduling a control task in such a shared ECU under the described assumptions. The goal of the controller scheduler that we design is to maximize the use of the available bandwidth for the newly added control task. The reason to seek maximizing the use of the available bandwidth is to achieve the highest possible performance in terms of disturbance rejection, as argued earlier. For a small set of tasks with relatively simple periodic specifications, the methodology is lightweight enough to be considered as an online scheduler which can deal with task characterization changing dynamically.

In our solution, we consider control tasks with the ability to select their update (periodic) frequency and we name them "variable-rate" control tasks. This can be achieved by associating to each control task a set of controllers each requiring a different update frequency. These differ from other aperiodic control alternatives, such as event-triggered [21] (ETC) or self-triggered [3] (STC) control implementations, in that the execution times **SD:** do you mean pattern of our controllers is a controllable parameter, as opposed to being dictated by the plant in ETC or STC. Thus in our proposal, once the pattern of ECU availability is known, the scheduler can select a sequence of controllers, with their associated update frequencies, so that the stability of the control task is guaranteed and simultaneously the available resource utilization is maximized. As we explain later, this choice, unless exercised judiciously, can result in system instability and suboptimal utilization of ECU bandwidth.

Technically, the variable-rate control systems we consider are switched sampled-data systems. The switching signal determines the closed-loop dynamics through the selection of a controller and an associated sampling time. The stability analysis of switched systems has been studied in depth [6,11] and, as pointed out before, it is well known that not every possible switching sequence results in stable closed-loops, even when switching between stable systems (as is our case). Much work has also been devoted in recent years to the computation of adequate sampling intervals to retain the stability of closed-loops under sample-and-hold controller implementations [14,21]. Here we leverage ideas from both the literature on switched systems and sampled-data systems, to construct a timed automaton dictating when switching to a different controller (also termed mode) is allowed in order to retain stability of the closed-loop. In turn, this automaton implicitly defines a switched sequence of sampling rates that results in stable operation of the system. In this sense, the type of abstraction provided in the current paper resembles the one proposed in [18] for scheduling event-triggered systems. This automaton can be referenced by the scheduler to affect the switching between the sampling modes without having to compute dwell time constraints at runtime. To this end we leverage tools from linearly priced timed automata [17,4] to synthesize schedulers that maximize the resource utilization.

The idea of using multiple sampling rates to schedule control loops is not new and has been investigated previously. In [9] sampling schedules are synthesized first, and an iterative procedure is proposed to synthesize a unique controller that would result in stable operation. However, the algorithm proposed is highly heuristic and no guarantees are provided that such a controller will be found. A different approach is taken in [16] by constructing automata that provide state-based conditions forcing a change of the controller update frequency. The main shortcoming of their approach is that no proof is provided for the stability of the system across transitions. Closer to our proposal are the works [23,8] in which automata are constructed representing mode switches retaining stability of the system. These automata are then employed to perform compositional analysis to synthesize schedulers for specifications given in the form of ω-regular languages. Their proposed methodology can be applied to a variant of our problem if each of the modes in that work is considered as the discrete model associated to a controller with a specific sampling time. The main differences of the present work with those stem from the abstractions employed for the mode switches. While in [23,8] it is implicitly assumed that switches between modes occur exactly at the sampling instants dictated by the current mode, our model allows for switches not synchronized with the sampling period of the current mode. More importantly, our approach is directly applicable to non-linear systems and we extend our objective to not only guaranteeing closed-loop stability, but also to additionally maximize the utilization of the available resources. Also very closely related to our work and [23,8], is the work from [**?**,**?**] developing anytime control algorithms. In [**?**,**?**], and subsequent publications, schedulers are designed that are capable of resolving a trade-off between quality of control (measured as a stochastic notion of stability) and bandwidth utilization, all under a stochastic

scheduling framework in which the availability of the channel is modeled in a probabilistic fashion. The main limitation of this line of work is again its restricted applicability to linear time-invariant systems. In summary, the key contributions of this work can be listed as follows.

1. We provide a methodology for deriving timing specifications for variable-rate control tasks. **SD:** check
2. We provide a methodology for scheduling such tasks on partially loaded ECUs while maximizing the control robustness **SD:** check.
3. We create a tool-flow that can take as input the control theoretic model of a variable-rate controller, the existing task loading pattern of an ECU and compute a schedule for the incoming control task.

2 Notation and Preliminaries

2.1 Notation

The symbols $\mathbb{N}$, $\mathbb{N}_0$, $\mathbb{Q}_0^+$, $\mathbb{R}$, $\mathbb{R}^+$, and $\mathbb{R}_0^+$ denote the set of natural, nonnegative integer, nonnegative rational, real, positive, and nonnegative real numbers, respectively. The symbols I_n, 0_n, and $0_{n\times m}$ denote the identity matrix, zero vector, and zero matrix in $\mathbb{R}^{n\times n}$, $\mathbb{R}^n$, and $\mathbb{R}^{n\times m}$, respectively. Given a vector $x \in \mathbb{R}^n$, we denote by $\|x\|$ the Euclidean norm of x, namely, $\|x\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$. Given a matrix $M = \{m_{ij}\} \in \mathbb{R}^{n\times m}$, we denote by $\|M\|$ the induced two norm of M, namely, $\|M\| = \sqrt{\lambda_{\max}(M^T M)}$, where $\lambda_{\max}(A)$ denotes the maximum eigenvalue of a symmetric matrix A. A continuous function $\gamma : \mathbb{R}_0^+ \to \mathbb{R}_0^+$, is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\gamma(0) = 0$; γ is said to belong to class $\mathcal{K}_\infty$ if $\gamma \in \mathcal{K}$ and $\gamma(r) \to \infty$ as $r \to \infty$. A continuous function $\beta : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$ is said to belong to class $\mathcal{KL}$ if, for each fixed s, the map $\beta(r, s)$ belongs to class $\mathcal{K}$ with respect to r and, for each fixed nonzero r, the map $\beta(r, s)$ is decreasing with respect to s and $\beta(r, s) \to 0$ as $s \to \infty$. Given a measurable function $f : \mathbb{R}_0^+ \to \mathbb{R}^n$, the (essential) supremum of f is denoted by $\|f\|_\infty$. Given a tuple S, we denote by $\sigma := (S)^\omega$ the infinite sequence generated by repeating S infinitely, i.e. $\sigma := SSSS\ldots$.

2.2 Control systems

The class of control systems considered in this paper is defined as:

Definition 1. *A* control system Σ *is a tuple* $\Sigma = (\mathbb{R}^n, \mathsf{U}, \mathcal{U}, f)$*, where* $\mathbb{R}^n$ *is the state space,* $\mathsf{U} \subseteq \mathbb{R}^m$ *is the input set, and*

- $\mathcal{U}$ *is a subset of the set of all measurable functions of time, from intervals of the form* $]a, b[\subseteq \mathbb{R}$ *to* U*, with* $a < 0$ *and* $b > 0$*;*
- $f : \mathbb{R}^n \times \mathsf{U} \to \mathbb{R}^n$ *is a continuous map satisfying the following Lipschitz assumption: for every compact set* $Q \subset \mathbb{R}^n$*, there exists a constant* $Z \in \mathbb{R}^+$ *such that* $\|f(x, u) - f(y, u)\| \leq Z\|x - y\|$ *for all* $x, y \in Q$ *and all* $u \in \mathsf{U}$*.*

A locally absolutely continuous curve $\xi :]a, b[\to \mathbb{R}^n$ is said to be a *trajectory* of Σ if there exists $\upsilon \in \mathcal{U}$ satisfying $\dot{\xi}(t) = f(\xi(t), \upsilon(t))$ for almost all $t \in]a, b[$. We shall as well refer to trajectories $\xi : [0, t] \to \mathbb{R}^n$ defined on closed intervals $[0, t]$, $t \in \mathbb{R}^+$, with the understanding of the existence of a trajectory $\xi' :]a, b[\to \mathbb{R}^n$ such that $\xi = \xi'|_{[0,t]}$ with $a < 0$ and $b > t$. We also write $\xi_{x\upsilon}(t)$ to denote the point reached at time t under the input υ from the initial condition $x = \xi_{x\upsilon}(0)$; the point $\xi_{x\upsilon}(t)$ is uniquely determined due to the assumptions on f [19].

A control system Σ is said to be forward complete if every trajectory is defined on an interval of the form $]a, \infty[$. Standard sufficient and necessary conditions for a control system to be forward complete can be found in [2].

In the remainder of this paper we assume $f(0_n, 0_m) = 0_n$ which implies that 0_n is an equilibrium point for a control system $\Sigma = (\mathbb{R}^n, \{0_m\}, \mathcal{U}, f)$. Here, we recall two stability notions, introduced in [12], as defined next.

Definition 2. *A control system $\Sigma = (\mathbb{R}^n, \{0_m\}, \mathcal{U}, f)$ is globally asymptotically stable (GAS) if it is forward complete and there exist a $\mathcal{KL}$ function β such that for any $t \in \mathbb{R}_0^+$ and any $x \in \mathbb{R}^n$, the following condition is satisfied:*

$$\|\xi_{x\upsilon}(t)\| \leq \beta(\|x\|, t), \tag{2.1}$$

where $\upsilon(t) = 0_m$ for any $t \in \mathbb{R}_0^+$.

Definition 3. *A control system $\Sigma = (\mathbb{R}^n, \mathsf{U}, \mathcal{U}, f)$ is input-to-state stable (ISS) with respect to inputs $\upsilon \in \mathcal{U}$ if it is forward complete and there exist a $\mathcal{KL}$ function β and a $\mathcal{K}_\infty$ function γ such that for any $t \in \mathbb{R}_0^+$, any $x \in \mathbb{R}^n$, and any $\upsilon \in \mathcal{U}$, the following condition is satisfied:*

$$\|\xi_{x\upsilon}(t)\| \leq \beta(\|x\|, t) + \gamma(\|\upsilon\|_\infty). \tag{2.2}$$

It can be readily seen, by observing (2.1) and (2.2), that ISS implies GAS by restricting the set of inputs to $\{0_m\}$. Note that a linear control system $\dot{\xi} = A\xi + B\upsilon$ is GAS or ISS iff A is Hurwitz[4] and the functions β and γ in (2.1) and (2.2) can be computed as:

$$\beta(r, s) = \|\mathsf{e}^{As}\| r, \quad \gamma(r) = \|B\| \left(\int_0^\infty \|\mathsf{e}^{As}\| \mathsf{d}s \right) r.$$

One can characterize the aforementioned ISS property with respect to some Lyapunov function, as defined next.

Definition 4. *A function $V : \mathbb{R}^n \to \mathbb{R}_0^+$ which is continuous on $\mathbb{R}^n$ and smooth on $\mathbb{R}^n \backslash \{0_n\}$ is said to be an ISS Lyapunov function for the closed-loop system*

$$\dot{\xi} = f(\xi, K(\xi + \varepsilon)), \tag{2.3}$$

where $K : \mathbb{R}^n \to \mathbb{R}^m$, if there exist $\mathcal{K}_\infty$ functions $\underline{\alpha}, \overline{\alpha}, \gamma$, and some constant $\kappa \in \mathbb{R}^+$ such that for all $x, e \in \mathbb{R}^n$ the following inequalities are satisfied:

$$\begin{aligned} \underline{\alpha}(\|x\|) \leq V(x) \leq \overline{\alpha}(\|x\|), \\ \frac{\partial V(x)}{\partial x} f(x, K(x + e)) \leq -\kappa V(x) + \gamma(\|e\|). \end{aligned} \tag{2.4}$$

[4] A square matrix A is called Hurwitz if the real parts of its eigenvalues are negative.

The following theorem, borrowed from [20], characterizes the ISS property for the closed-loop system (2.3) in terms of the existence of an ISS Lyapunov function.

Theorem 1. *The closed-loop system* (2.3) *is ISS with respect to measurement errors* ε *if and only if there exists an ISS Lyapunov function for* (2.3).

The specific ISS characterization for linear case is reported in the Appendix.

3 Problem Formulation

Consider a control system Σ and assume that there exist p different controllers $K_i : \mathbb{R}^n \to \mathbb{R}^m$, $i \in \mathsf{S} := \{1, \ldots, p\}$, rendering the closed-loop system

$$\dot{\xi} = f(\xi, K_i(\xi + \varepsilon)) \tag{3.1}$$

ISS with respect to measurement errors $\varepsilon : \mathbb{R}_0^+ \to \mathbb{R}^n$, with associated ISS Lyapunov functions V_i and corresponding $\mathcal{K}_\infty$ functions $\underline{\alpha}_i, \overline{\alpha}_i, \gamma_i$ and positive constants $\kappa_i \in \mathbb{R}^+$ as in Definition 2.4. Now consider a variable-rate control system $\hat{\Sigma} = (\Sigma, P, \mathsf{S}, \mathcal{S})$ representing a sample-and-hold implementation of the closed loop of Σ with different controllers K_i with the associated sampling times h_i, as depicted schematically in Figure 1 in the Appendix, where $P = \{K_1, \ldots, K_p\}$, $\mathsf{S} = \{1, \ldots, p\}$, and

- $\mathcal{S}$ denotes a subset of the set of all piecewise constant cádlág (i.e. right-continuous and with left limits) functions from $\mathbb{R}_0^+$ to S with a finite number of discontinuities on every bounded interval in $\mathbb{R}_0^+$ (no Zeno behaviour). Each $\pi \in \mathcal{S}$ represents a schedule dictating which controller is active at any time $t \in \mathbb{R}_0^+$. Given any $\pi \in \mathcal{S}$, denoting the switching times as $t_0, t_1, t_2, \ldots$ (occurring at the discontinuity points of π), we denote by $p_i \in \mathsf{S}$ the value of the switching signal on the interval $[t_i, t_{i+1}[$. We assume also that the set $\mathcal{S}$ contains only elements for which there exists constants $\tau_{p_i p_{i+1}} \in \mathbb{Q}_0^+$ such that $\tau_{p_i p_{i+1}} \leq t_{i+1} - t_i$, for any $i \in \mathbb{N}_0$, and $\tau_{p_i p_{i+1}} \geq h_{p_i}$ for any $p_i, p_{i+1} \in \mathsf{S}$.

A continuous-time curve $\xi :]a, b[\to \mathbb{R}^n$ is said to be a *trajectory* of $\hat{\Sigma}$ if there exists a switching signal $\pi \in \mathcal{S}$ satisfying:

$$\begin{aligned} &\dot{\xi}(t) = f(\xi(t), \upsilon(t)) \\ &\upsilon(t) = K_{\pi(t)}(\xi(\ell h_{\pi(t)})),\ t \in [\ell h_{\pi(t)}, (\ell+1) h_{\pi(t)}[,\ \forall \ell \in \mathbb{N}_0. \end{aligned} \tag{3.2}$$

We now introduce the main problem which we plan to solve in this paper.

Problem 1. Consider a set T of real-time tasks characterized by arrival curves (α_j^l, α_j^u), $j = 1, \ldots, T$, and a control task defined as in (3.2) . Determine the schedule $\pi \in \mathcal{S}$ of controllers and associated sampling times (K_i, h_i) for the control task to maximize the utilization of the residue bandwidth left by the real-time tasks while simultaneously guaranteeing the stability of the control task.

We want to maximally utilize the available bandwidth on the average thus increasing the robustness of the controller implementation to external disturbances.

4 Adaptive Scheduling of Variable-rate Control Tasks

In order to compute a stability aware schedule for the incoming control task on an existing platform, we first need: (i) to construct an abstraction of the scheduling constraints that need to be respected to guarantee stability of the control task; (ii) to estimate the available processing bandwidth left by the real-time tasks already present in the platform. We address these challenges in the following.

4.1 Control Task Scheduling Constraints

Consider a control task defined as in (3.2) satisfying the following assumption.

Assumption 2 *Each of the pairs (K_i, h_i), $i = 1, \ldots, p$, of controller and associated sampling times are such that each sampled-data control system* (3.2) *satisfies*

$$\begin{aligned} \frac{\partial V_i(\xi(t))}{\partial x} f(\xi(t), \upsilon(t)) &\leq -\hat{\kappa}_i V_i(\xi(t)), \\ \upsilon(t) &= K_i(\xi(\ell h_i)),\ t \in [\ell h_i, (\ell+1)h_i[,\ \forall \ell \in \mathbb{N}_0, \end{aligned} \tag{4.1}$$

for some $\hat{\kappa}_i \in \mathbb{R}^+$, guaranteeing that each mode of the closed-loop system is GAS.

Remark 1. There is an abundant literature allowing to design together such pairs of controllers and sampling times satisfying Assumption 4.1, see e.g. [15] and references therein. Alternatively, one can consider that a continuous time controller is available and compute an adequate sampling time under very mild assumptions on the plant and controller (namely, continuity of the dynamics of plant and controller) such that Assumption 4.1 holds, see e.g. [14]. Additionally, the large bulk of literature on event-triggered control provides an alternative way of computing sampling times h_i satisfying our requirements through the closely related methods to compute minimum inter-sample times, see e.g. [21]. Note also that the results hereby presented can be directly extended to the case of locally asymptotically stable systems, if one instead assumes (4.1) only holds on a compact set.

We define a timed automaton (cf. Appendix for a definition) $\mathcal{T}_{ad} = (L, L_0, C, E, \mathsf{Inv})$, where $L = \mathsf{S}$, $L_0 = L$, $C = \{\zeta\}$, and

- the set E of edges is given by the collection of all tuples (i, g, r, j) such that $i, j \in \mathsf{S}$, $g = \{\zeta \mid \zeta \geq \tau_{ij}\}$ is the transition guard, and the clock reset set r is given by $\{\zeta\}$;

– the (location) invariant for mode i is given by $\mathsf{Inv}(i) := \{\zeta \mid \zeta \geq 0\}$, $\forall i \in \mathsf{S}$,

describing the set of admissible switching policies between different controllers K_i, i.e. $\mathcal{S}_{ad} \subseteq \mathcal{S}$, guaranteeing GAS of the variable-rate closed-loop system, schematically illustrated in Figure 1 in the Appendix. An example of such automaton for the case of $\mathsf{S} = \{1, 2, 3\}$ is depicted in Figure 2.

In the following, we establish which properties τ_{ij}, $i \in \{1, \ldots, p\}$ and $j \in \{1, \ldots, p\} \backslash \{i\}$, need to satisfy so that indeed $\mathcal{T}_{ad}$ characterizes a set $\mathcal{S}_{ad} \subseteq \mathcal{S}$ of stabilizing switching sequences.

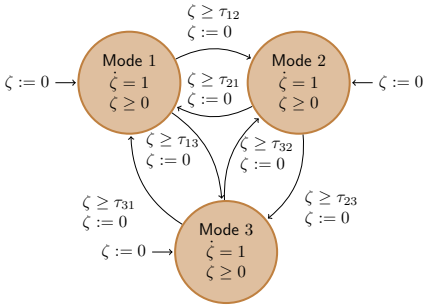

Fig. 1. Timed automaton $\mathcal{T}_{ad}$ for 3 controllers.

Assumption 3 *For any pair of $i, j \in \{1, \ldots, p\}$, there exits a constant $\mu_{ij} \geq 1$ such that*

$$\forall x \in \mathbb{R}^n, \quad V_i(x) \leq \mu_{ij} V_j(x). \quad (4.2)$$

Note that Assumption 4.3 is always satisfied for the type of ISS Lyapunov functions introduced in Theorem 2.6 whose existences are guaranteed for stabilized linear control systems.

Theorem 4. *Consider the variable-rate control system $\hat{\Sigma}$ in (3.2) and let Assumptions 4.1 and 4.3 hold. If $\log \frac{\mu_{ij}}{\rho} < \hat{\kappa}_j \tau_{ji}$, for any $i, j \in \{1, \ldots, p\}$, $i \neq j$, and some $\rho \in]0, 1[$, then Σ is GAS and $\mathcal{T}_{ad}$ using these τ_{ij} characterizes a set $\mathcal{S}_{ad}$ of stabilizing switching sequences.*

The proof of Theorem 4.4 is provided in the Appendix.

4.2 Task Set Characterization

We consider an ECU platform for which the arrival pattern of each task in an existing task set T is known. For a task $\theta_i \in T$, the RTC arrival curves (α_i^l, α_i^u) are functions from $\mathbb{R}^+$ to $\mathbb{N}$ [7,22] such that the number of instances of θ_i arriving in any time window of size t is lower bounded by $\alpha_i^l(n)$ and upper bounded by $\alpha_i^u(n)$. Let us consider the deadline of each instance of θ_i to be e_i time units. The maximum number of task instances of type θ_i that may arrive in a period of size e_i is given by a_i where $a_i := \alpha_i^u(e_i)$. A computational bandwidth which allows sufficient scheduling slots for executing a_i instances of θ_i in every interval of length e_i is sufficient for scheduling θ_i since e_i represents the deadline of each instance of θ_i.

Consider now the worst case execution time (WCET) of θ_i to be w_i CPU cycles. Thus, $a_i w_i$ number of CPU cycles need to be made available for task θ_i in every consecutive real-time interval of size e_i. Let the total available bandwidth offered by the CPU be given as H computation cycles per time unit. Assume that the existing scheduling policy for the platform allocates a $y_i \in [0, 1]$ fraction of

the total bandwidth to θ_i. We consider an *as late as possible* (ALAP) scheduling of the tasks in the period e_i, meaning that the fraction of bandwidth y allocated to task θ_i is provided in intervals of time of length $\delta_i := \frac{a_i w_i}{H y_i}$ whose end coincides with multiples of e_i. We provide a formal definition of these ALAP utilization patterns in the following.

Definition 5. *The* ALAP bandwidth utilization pattern *of a task θ_i, with period e_i, bandwidth utilization fraction y_i, and utilization time δ_i, is a function $\sigma_{\theta_i}^{(e_i,y_i,\delta_i)} : \mathbb{R}_0^+ \to [0,1]$ satisfying the following properties:*

- $\sigma_{\theta_i}^{(e_i,y_i,\delta_i)}(t+e_i) = \sigma_{\theta_i}^{(e_i,y_i,\delta_i)}(t)$
- $\sigma_{\theta_i}^{(e_i,y_i,\delta_i)}(t) = 0,\ t \in [0, e_i - \delta_i]$
- $\sigma_{\theta_i}^{(e_i,y_i,\delta_i)}(t) = y_i,\ t \in [e_i - \delta_i, \epsilon$

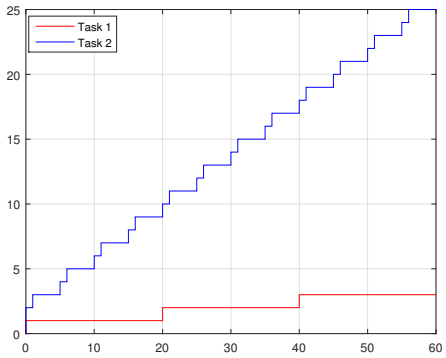

Fig. 2. Arrival curves α_1^u, α_2^u.

Given the task set $T = \{\theta_1, \ldots, \theta_n\}$ with corresponding deadlines $\{e_1, \ldots, e_n\}$ and fraction of bandwidth allocated $\{y_1, \ldots, y_n\}$, one can compute their respective δ_i's as indicated earlier. Adding up the bandwidth utilization scenarios of each of those tasks a total bandwidth utilization pattern can be obtained $\overline{\sigma}_T = \sum_{i=1}^{n} \sigma_{\theta_i}^{(e_i,y_i,\delta_i)}$. Note that this bandwidth utilization pattern is also a periodic function $\overline{\sigma}_T : \mathbb{R}_0^+ \to [0,1]$ with period $\bar{e} = l.c.m.(e_1, \ldots, e_n)$, where $l.c.m.$ stands for least common multiple, but this is not necessarily an ALAP bandwidth utilization pattern (the last two conditions of the definition may not hold for $\overline{\sigma}_T$). It may be noted that budgeting processor bandwidths in the way as discussed above is useful for satisfying the bandwidth requirement for any scheduling policy which do not allow deadline violation.

Example 1. Let us consider two tasks θ_1, θ_2 with upper arrival curves as shown in Figure 3. We have $e_1 = 30$ and $e_2 = 20$ time units. From the arrival curves we get $a_1 = \alpha_1^u(30) = 2$ and $a_2 = \alpha_1^u(20) = 10$. Thus, we have a maximum of 2 instances of θ_1 and 10 instances of θ_2 arriving inside a period of 30 and 20 time units, respectively. We reserve enough computational slots inside such periods for executing a_i instances of θ_i inside an interval of size e_i, $i = 1, 2$.

Let the WCET of θ_1, θ_2 be 30000 and 2000 CPU cycles, respectively. The CPU offers 10000 computation cycles (H) per time unit and the scheduler offers 40% and 20% of overall CPU bandwidth to θ_1 and θ_2, respectively, during execution (i.e. $y_1 = 0.4$, $y_2 = 0.2$). This means θ_1 and θ_2 gets 4000 and 2000 computation cycles per time unit, respectively, while executing. This results in the execution of every instance of θ_1 consuming $30000/4000 = 7.5$ time units. Thus, to satisfy the total worst case demand of θ_1, i.e. 2 instances in periodic intervals of size 30 time units, we require 15 time units of CPU given a bandwidth

of 40%. Similarly, we require 10 time units in periodic intervals of size 20 given a bandwidth of 20% for θ_2. Adding up these requirements point-wise, a worst case bandwidth requirement pattern, which recurs every $l.c.m.(30, 20) = 60$ time units, is computed. The resulting bandwidth pattern is illustrated in Figure 4.

4.3 Scheduler Design

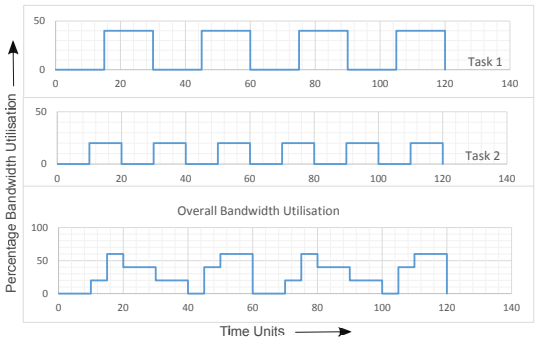

Fig. 3. ALAP Bandwidth budgets

As discussed earlier, given a control system with p different controllers (from a set $P = \{K_1, \ldots, K_p\}$) having sampling rates $\{f_1 < f_2 < \cdots < f_p\}$, $(f_i = h_i^{-1})$, we can construct a TA $\mathcal{T}$ having p number of modes, where each mode $1 \leq i \leq p$ signifies the use of controller K_i. For every possible mode switch from some mode i to some mode j, the automaton provides a timing constraint τ_{ij} signifying what is the minimum duration of using mode i (using K_i) so that a switching can be performed to mode j (start using K_j) guaranteeing that the overall closed loop system is GAS. Let the WCET of controller K_i be ω_i^c (in CPU cycles), and thus its computational requirement is $\omega_i^c f_i$ cycles per time unit. Given H available computing cycles per time unit, the fraction of bandwidth required by a controller is $r_i = \frac{\omega_i^c f_i}{H}$, and so one can compute the bandwidth requirements $\{r_1, \ldots, r_p\}$ of all controllers in P.

Given the total bandwidth utilization pattern $\overline{\sigma}_T$ for the task set T, with period $\bar{e}$, denote by $\underline{\sigma}_T := 1 - \overline{\sigma}_T$ the residual bandwidth pattern. Let us consider such a pattern $\underline{\sigma}_T$ and describe it by a string $\underline{s}_T := ((l_1, v_1), \ldots, (l_\nu, v_\nu))^\omega$, with $\sum_{i=1}^{\nu} v_i = \bar{e}$, denoting the infinitely repeating concatenation of time intervals of length v_i and associated fraction of bandwidth available l_i, with $i = 1, \ldots, \nu$. We refer to the i-th tuple in the sequence as the *i-th stage* of the pattern and to the period $\bar{e}$ as the *recurrence length* of the pattern. We also denote by $\underline{s}_T[\bar{e}] := ((l_1, v_1), \ldots, (l_\nu, v_\nu))$. We use the sequence description $\underline{s}_T$ and the availability pattern $\underline{\sigma}_T$ interchangeably in what follows.

Consider now the available bandwidth string $\underline{s}_T$, and define $S := \langle S_1, \ldots, S_\nu \rangle$ with $S_i \subseteq 2^P$, $\forall i \in \{1, \ldots, \nu\}$, such that $S_i = \{K_j \,|\, r_j \leq l_i\}$. The list S_i contains the controllers which are schedulable at each of the i-th stages of $\underline{s}_T$ in terms of the available bandwidth (but possibly leading to an unstable closed-loop operation).

Example 2. Given the instance of $\underline{\sigma}_T$ as shown in Figure 5, a possible list S for the case of 4 controllers, could take the form $S_1 = \{K_1, K_2\}, S_2 = \{K_1, K_2, K_3\}, S_3 = \{K_1\}, S_4 = \{K_1, K_2\}, S_5 = \{K_1, K_2, K_3, K_4\}$, where we have assumed that $r_i < r_j$ if $i < j$.

We specify a switching sequence ς as a string $\varsigma = ((K_{i_1}, t_{i_1}), \ldots, (K_{i_k}, t_{i_k}), \ldots, (K_{i_n}, t_{i_n}))$ indicating that controller K_{i_k} is used inside the k-th time interval $\left[\sum_{j=1}^{k-1} t_{i_j}, \sum_{j=1}^{k} t_{i_j}\right]$, of duration t_{i_k}. We consider only nontrivial sequences in the sense that $K_{i_j} \neq K_{i_{j+1}}$ for all j in a sequence. The *length* of such a sequence is given by $|\varsigma| = \sum_{j=1}^{n} t_{i_j}$. One can therefore construct from such a sequence ς a switching signal $\pi \in \mathcal{S}$ (c.f. Section 3) by letting $\pi(t) = i_k$ if $t \in \left[\sum_{j=1}^{k-1} t_{i_j} + \bar{e}s, \sum_{j=1}^{k} t_{i_j} + \bar{e}s\right]$ for some $s \in \mathbb{N}$.

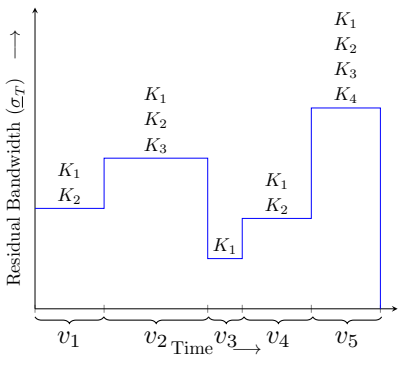

Fig. 4. Admissible Controllers

Note that the timed automaton $\mathcal{T}_{ad}$ derived in section 4.1, provides us with a set of timing constraints $\{\tau_{ij} \mid 1 \leq i \leq p \wedge 1 \leq j \leq p,\ i \neq j\}$ where a constraint τ_{ij} signifies the minimum amount of time controller K_i should execute (i.e. stay in mode i of the automaton) before a switch to controller K_j is allowed. Given $\underline{\sigma}_T$, the list S, and $\mathcal{T}_{ad}$, an admissible switching sequence can be defined as follows.

Definition 6. *Given a bandwidth pattern $\underline{\sigma}_T$ described by the string $\underline{s}_T := ((l_1, v_1), \ldots, (l_\nu, v_\nu))^\omega$ with recurrence length $\bar{e}$, the list S of admissible controllers, the timed automaton $\mathcal{T}_{ad}$, a switching sequence $\varsigma = ((K_{i_1}, t_{i_1}), \ldots, (K_{i_k}, t_{i_k}), \ldots, (K_{i_n}, t_{i_n}))$ is considered admissible if:*

- $|\varsigma| = l \times \bar{e}$, *for some* $l \in \mathbb{N}$;
- *For any* $1 \leq k \leq n$, *if* $\exists m,\ 0 \leq m < l$, *and* $\exists p,\ 1 \leq p \leq \nu$, *such that the intervals* $\left[\sum_{j=1}^{k-1} t_{i_j}, \sum_{j=1}^{k} t_{i_j}\right]$ *and* $\left[m \times \bar{e} + \sum_{j=1}^{p-1} v_j, m \times \bar{e} + \sum_{j=1}^{p} v_j\right]$ *intersect, i.e., for every k-th time interval in ς if there is a non-null intersection with an interval v_p of $\underline{\sigma}_T$, then $K_{i_k} \in S_p$ and $t_{i_k} \geq \tau_{i_k i_{k+1}}$.*

We define the notion of *bandwidth rejection* by an admissible switching sequence as follows.

Definition 7. *Given a bandwidth pattern $\underline{\sigma}_T$ with recurrence length $\bar{e}$ and the list S of admissible controllers, the* bandwidth rejection *by an admissible switching sequence $\varsigma = ((K_{i_1}, t_{i_1}), \ldots, (K_{i_k}, t_{i_k}), \ldots, (K_{i_n}, t_{i_n}))$ is given by*

$$rej(\varsigma) = \int_0^{l \times \bar{e}} (\underline{\sigma}_T(t) - r(t))\mathsf{d}t,$$

where $|\varsigma| = l \times \bar{e}$, for some $l \in \mathbb{N}$, and $r : \mathbb{R}_0^+ \to [0, 1]$ is defined as

$$r(t) = r_{i_k} \quad \textit{for} \quad t \in \left[\sum_{j=1}^{k-1} t_{i_j}, \sum_{j=1}^{k} t_{i_j}\right[.$$

In order to formally capture the set of admissible switching sequences, we construct a Linearly Priced Timed Automaton (LPTA) (cf. Appendix for a definition). Given the list S, $\underline{\sigma}_T$, and the timing constraints for stable switching

as captured by the timed automaton $\mathcal{T}_{ad}$ derived in section 4.1, the LPTA $\mathcal{T} = (L, L_0, C, E, \mathsf{Inv}, \mathcal{P})$ is defined as follows.

- $L = \{m_{i,j} \,|\, \exists (i,j) \in \{1,\ldots,\nu\} \times \{1,\ldots,p\}$ s.t. $K_j \in S_i\}$. A location $m_{i,j}$ denotes a possible choice of controller K_j inside the time interval v_i.
- $L_0 = \{m_{1,k} \in L \,|\, K_k \in S_1\}$.
- $C = \{c, x, c_g\}$. Clock c is used to keep track of the total time elapsed using the same controller mode across a sequence of intervals. Clock x tracks the time spent on all locations inside the same stage while c_g serves as a global clock.
- E contains three types of edges:
  1. *inter-stage* edges: for $m_{i,j}, m_{i+1,k} \in L$, $(m_{i,j}, \phi, C', m_{i+1,k}) \in E$ if $(K_j \in S_i) \wedge (K_k \in S_{i+1})$.
  2. *intra-stage* edges: for $m_{i,j}, m_{i,k} \in L$, $(m_{i,j}, \phi, C', m_{i,k}) \in E$ if $(K_j \in S_i) \wedge (K_k \in S_i)$.
  3. *final stage* edges: for $m_{\nu,j}, m_{1,j} \in L$, $(m_{\nu,i}, \phi, C', m_{1,j}) \in E$ if $K_i \in S_\nu$, where v_ν is the last interval defining $\underline{\sigma}_T$.
- An inter-stage edge $(m_{i,j}, \phi, C', m_{i+1,k}) \in E$ has a clock reset set $C' = \{c, x\}$ if $j \neq k$ and $C' = \{x\}$ otherwise.
- An inter-stage edge $(m_{i,j}, \phi, C', m_{i+1,k}) \in E$ has a guard $\phi = (c \geq \tau_{jk}) \wedge (x \geq v_i)$ if $j \neq k$ and $\phi = (x \geq v_i)$ otherwise.
- An intra-stage edge $(m_{i,j}, \phi, C', m_{i,k}) \in E$ shall always have $j \neq k$ by construction. For such a transition, $\phi = (c \geq \tau_{jk})$ and $C' = \{c\}$.
- A final-stage edge $(m_{\nu,i}, \phi, C', m_{1,j}) \in E$ has a clock reset set $C' = \{c, x\}$ if $i \neq j$ and $C' = \{x\}$ otherwise.
- A final-stage edge $(m_{\nu,i}, \phi, C', m_{1,j}) \in E$ has a guard $\phi = (c \geq \tau_{ij}) \wedge (x \geq v_i)$ if $i \neq j$ and $\phi = (x \geq v_i)$ otherwise.
- $\mathrm{Inv}(m_{i,j}) = \{x \leq v_i\}$, $\forall m_{i,j} \in L$. These invariants force the automaton to leave $m_{i,j}$ after spending v_i time in the mode. This takes care of the bandwidth availability requirement.
- For a location (mode) $m_{i,j}$, the cost rate function $\mathcal{P}$ is defined as, $\mathcal{P}(m_{i,j}) = (l_i - r_j)$, $\forall m_{i,j} \in L \setminus \{f\}$, and $\mathcal{P}(e) = 0$ $\forall e \in E$. The cost rate at $m_{i,j}$ is the difference between the bandwidth offered inside the interval v_i, and the bandwidth required by controller K_j, i.e. r_j. We do not assign any costs to the edge transitions.

Considering the number of stages in the residual bandwidth pattern as ν and the maximum of controller options in each stage as p (= total number of controllers available for the control loop), in the worst case the number of states in the LPTA is $O(\nu p)$, the number of inter-stage and intra-stage of edges are both $O(p^2)$. Hence, the time complexity for LPTA synthesis is $O(\nu p + p^2)$.

Remark 2. By construction, any run of $\mathcal{T}$ with length being an integer multiple of $\bar{e}$ is an admissible switching sequence.

Example 3. We exemplify the construction of $\mathcal{T}$ in Figure 6 in the Appenix for the available recurrent bandwidth pattern $\underline{\sigma}_T$ along with admissible controllers

as shown in Figure 5. The LPTA shown in Figure 6 is labeled with all the inter-stage and final-stage edges. For visual clarity we show the intra-stage edges for the interval v_1 only. We have also left out the cost rate labels of the modes.

Based on the notion of on-the-average non-utilized bandwidth, we define a switching sequence as optimal as follows.

Definition 8. *An admissible switching sequence ς^* is considered optimal if for every other admissible switching sequence ς, we have $\frac{rej(\varsigma^*)}{|\varsigma^*|} \leq \frac{rej(\varsigma)}{|\varsigma|}$.*

It may be noted that in general an admissible switching sequence can be of length which is any integer multiple of $\bar{e}$. Hence, it makes sense to consider optimality among admissible switching sequences upto a maximum length.

Definition 9. *An admissible switching sequence $\varsigma^* = (K^*_{i_1}, t^*_1)(K^*_{i_2}, t^*_2) \cdots (K^*_{i_k}, t^*_k)$ with $i_1, \ldots, i_k \in \{1, \ldots, p\}$ is considered optimal in N-unfolding of $\underline{\sigma}_T$ if $|\varsigma^*| \leq N \times \bar{e}$ and among all admissible switching sequences with length $\leq N \times \bar{e}$, ς^* incurs the least (average) bandwidth rejection, i.e. for any other admissible switching sequence ς with $|\varsigma| < N \times \bar{e}$, we have $\frac{rej(\varsigma^*)}{|\varsigma^*|} \leq \frac{rej(\varsigma)}{|\varsigma|}$.*

Computing Recurrent Schedules As a scheduling solution, we are interested in switching sequences which follow a recurring pattern just like the bandwidth pattern that recurs every $\bar{e}$ time units.

Definition 10. *Given a bandwidth pattern $\underline{\sigma}_T$ and the TA $\mathcal{T}_{ad}$, an admissible switching sequence $\varsigma^* = (K^*_{i_1}, t^*_1)(K^*_{i_2}, t^*_2) \cdots (K^*_{i_k}, t^*_k)$ with $i_1, \ldots, i_k \in \{1, \ldots, p\}$ is considered recurrent and optimal in N unfolding of $\underline{\sigma}_T$ if ς^* is* optimal in N unfolding *of $\underline{\sigma}_T$ and $t^*_k \geq \tau_{i_k i_1}$.*

The second condition captures the requirement to be satisfied by a finite length pattern to be able to recur, according to the constraints imposed by $\mathcal{T}_{ad}$. We denote such a sequence by ς^*_N.

Observe that the period $\bar{e}$ of the repeating bandwidth pattern can be very large by itself for a potentially large task set (it is the $l.c.m$ of all task deadlines). In this work, we restrict our search for recurrent optimal switching sequences to some preset N levels of unfolding of the bandwidth pattern. Let us denote the switching sequence corresponding to the run of $\mathcal{T}$ which leads to minimum cost reachability of a state (location/vertex $= l$, clock valuation $= \mathbf{v}$) by $\varsigma^*(\mathcal{T}, (l, \mathbf{v}))$. Similarly, for some initial vertex $l \in L_0$, let $\varsigma^*_r(\mathcal{T}, (l, \mathbf{V}))$ denote the recurring switching sequence corresponding to the run which starts at l (with all clock valuations being '0') and reaches l with some valuation $\mathbf{v} \in \mathbf{V}$ at minimal cost. In our case, minimal cost implies minimal rejection of available bandwidth. Such minimal cost runs, if exists, can be found by restricting the set of initial vertices of $\mathcal{T}$ to $\{l\}$ and applying minimum cost reachability analysis [17,4] for the vertex l with the valuation set as per the specification of $\mathbf{V}$.

Remark 3. For the LPTA $\mathcal{T}$ constructed from a bandwidth pattern $\underline{\sigma}_T$ of length $\bar{e}$ and TA $\mathcal{T}_{ad}$ using methods outlined earlier, if ς_N^* is the admissible switching sequence recurrent and optimal in N unfolding of $\underline{\sigma}_T$, then

$$\frac{rej(\varsigma_N^*)}{|\varsigma_N^*|} = \min\left\{ \frac{rej(\varsigma)}{|\varsigma|} \,|\, \varsigma = \varsigma_r^*(\mathcal{T}, (l, \{c_g = i \times \bar{e}\})), l \in L_0, 1 \leq i \leq N \right\}.$$

The quantity is non-trivial if there exists at least one switching sequence which recurs with period $\in \{i \times \bar{e} \,|\, 1 \leq i \leq N\}$. For computing ς_N^*, the set $\{\varsigma = \varsigma_r^*(\mathcal{T}, (l, \{c_g = i \times \bar{e})) \,|\, l \in L_0,\, 1 \leq i \leq N\}$ is enumerated. This essentially means running $N \times |L_0|$ number of minimum cost reachability analysis over the LPTA where the time taken for each analysis is not uniform and increases with c_g.

Remark 4 (State space reduction).
We have already noted that the period $\bar{e}$ of $\underline{\sigma}_T$ can be very large for a large task set. In that case it makes sense to merge consecutive stages in $\underline{\sigma}_T$ based on some tolerance value. Thus, for a given $\underline{\sigma}_T = ((l_1, v_1), \ldots, (l_\nu, v_\nu))^\omega$, and a bandwidth tolerance ϵ, if we have successive stages $(l_i, v_i), (l_{i+1}, v_{i+1})$ in $\underline{\sigma}_T$ such that if $|l_i - l_{i+1}| \leq \epsilon$ then we simply merge them to create a single stage (l, v) with $l = \min(l_i, l_{i+1})$ and $v = v_i + v_{i+1}$ along with the corresponding set of admissible controllers given by $S' = S_i \cap S_{i+1}$ for the list S. This optimization helps in significant reduction in the size of the LPTA.

Remark 5 (Multiple control loops).
The method discussed for scheduling a single control loop can be extended to admit a set of control loops on the existing ECU platform. In that case we derive the TA $\mathcal{T}_{ad}$ for each control loop (having multiple controller options) using the theory as discussed in section 4. We compute the product of such a set of TAs and subsequently apply our method for computing switching sequences.

For a large-scale control system with r control loops, each having p controller options, we may land up with p^r controller options for each stage of the LPTA in the worst-case. In general, the bandwidth restriction of each stage eliminates the inadmissible options. The complexity of minimum cost reachability analysis of LPTA is exponential in the number of clocks [4] (which is $3r$ for modeling r control loops). For an LPTA with $|L_0|$ initial states being checked with N unfolding, the analysis is performed $|L_0| \times N$ times.

5 Simulation Results

We revisit the task set $T = \{\theta_1, \theta_2\}$ of our running example. The different parameters for the example are listed in Table **??**. Considering the overall available bandwidth (H) to be 10000 cycles per time unit, the residual bandwidth pattern is computed as $\underline{\sigma}_{T1} = ((l_1, v_1), \cdots)^\omega = ((1, 10), (0.8, 5), (0.4, 5), (0.6, 10), (0.8, 10), (1, 5), (0.6, 5), (0.4, 10))^\omega$ with one time unit being 0.1 seconds. Similarly, we can compute for

another arbitrary task set, a residual bandwidth pattern, $\underline{\sigma}_{T2} = ((1,5),(0.8,5),(0.5,10),(0.4,5),(0.6,5),(0.8,10),(1,5),(0.6,5),(0.4,10),(0.45,5),(0.6,25),(0.4,5),(1,5))^{\omega}$.

We are interested in using the existing ECU to schedule a control loop for which multiple controller options with different possible sampling rates are available. We have applied the proposed approach to a number of linear control systems and the results are reported in this section as well as in the Appendix. The dynamic of each system Σ is given by $\dot{\xi} = A\xi + Bv$, for some matrices A and B of appropriate dimensions.

Task(θ_i)	θ_1	θ_2
Deadline(e_i)	$e_1 = 30$	$e_2 = 20$
$a_i = \alpha^u(e_i)$	2	10
WCET (CPU cycles)	30000	2000
% Bandwidth Allocated	40%	20%

Table 1. A sample set of tasks

For each system, we construct p stabilizing controllers $\{K_1, \ldots, K_p\}$ using classical results in linear control theory. For any $i \in \{1, \ldots, p\}$, we find Lyapunov functions $V_i(x) := x^T M_i x$, for any $x \in \mathbb{R}^n$ and some positive definite matrix $M_i \in \mathbb{R}^{n \times n}$, for the closed loop of Σ equipped with the controller K_i satisfying the LMI in (**??**) with $\kappa = i$. Furthermore, the computed Lyapunov functions V_i satisfy the inequality (4.1) with $\hat{\kappa}_i = i/2$ and the associated sampling periods $\{h_1, \ldots, h_p\}$. The controller update rates are then obtained as $f_i = (1/h_i)$. Using Theorem 4.4, we can compute τ_{ij} by choosing $\rho = 0.5$.

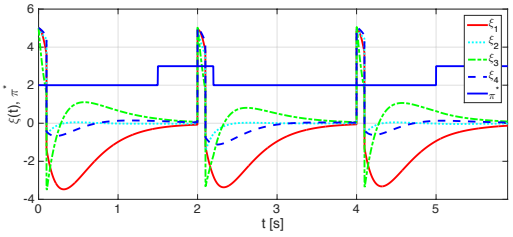

Fig. 5. Closed-loop system trajectory for Batch Reactor Process.

We consider the WCET of controller K_i to be ω_i^c (in CPU cycles) and so one can compute the bandwidth requirements for the controllers, i.e. $\{r_1, \ldots, r_p\}$, as $r_i = \frac{\omega_i^c f_i}{H}$. Given the $\underline{\sigma}_T$ for the current platform load, we can then compute the list S of admissible controllers. Using our modeling method from section 4.3, we create the LPTA specification for this scenario in UPPAAL CORA [13] to obtain the minimum cost schedule. For each example we have run the simulations for 5 unfolding to find an optimal admissible switching sequence.

For example, Figure **??** shows a simulation of a closed loop batch reactor process with $p = 3$ and step disturbances (of 0.1s width) at seconds 0, 2, and 4. It illustrates how the schedule obtained for $\underline{\sigma}_{T1}$ from Table **??** indeed retains the stability of the closed-loop under the considered disturbances.

5.1 Batch Reactor Process

The matrices A and B are given in [10] as,

$$A = \begin{bmatrix} 1.50 & 0 & 70 & -5 \\ -0.50 & -4 & 0 & 0.50 \\ 1 & 4 & -6 & 6 \\ 0 & 4 & 1 & -2 \end{bmatrix}, \; B = \begin{bmatrix} 0 & 0 \\ 5 & 0 \\ 1 & -3 \\ 1 & 0 \end{bmatrix}.$$

We consider a maximum of $p = 14$ controllers. Due to lack of space, we do not provide matrices K_i and M_i. For $i \in \{1, \ldots, p\}$, the sampling periods are $h_i = 10^{-3} \times \{2.09, 2.33, 2.60, 2.71, 2.91, 3.25, 3.61, 4.17, 4.56, 4.95, 5.38, 5.49, 5.80, 6.04\}$. The values of τ_{ij} are not mentioned in order to save space. Consider the controllers WCETs $\omega_i^c = \{23\}$. The bandwidth requirements for the controllers are computed as $r_i = \{0.96, 0.86, 0.77, 0.74, 0.69, 0.62, 0.59, 0.51, 0.49, 0.45, 0.41, 0.41, 0.40, 0.39\}$. The simulation results are summarized in Table **??**.

p	Controllers	Switching Sequence	Cost
For $\underline{\sigma}_{T1}$			
3	K_4, K_{12}, K_{14}	$[(K_{12}, 15), (K_{14}, 7), (K_{12}, 28), (K_{14}, 10)]^\omega$	14.74
6	K_4, $K_{10}, \ldots, K_{14}$	$[(K_{12}, 15), (K_{11}, 5), (K_{12}, 30), (K_{11}, 10)]^\omega$	12.75
9	K_4, $K_7, \ldots, K_{14}$	$[(K_{12}, 15), (K_8, 7), (K_{12}, 28), (K_8, 10)]^\omega$	6.83
12	$K_3, \ldots, K_{14}$	$[(K_4, 15), (K_5, 7), (K_{11}, 8), (K_4, 15), (K_{11}, 5), (K_5, 10)]^\omega$	1.43
14	$K_1, \ldots, K_{14}$	$[(K_4, 10), (K_{14}, 5), (K_2, 7), (K_9, 8), (K_{14}, 10), (K_4, 5), (K_9, 5), (K_2, 10)]^\omega$	1.58
For $\underline{\sigma}_{T2}$			
3	K_4, K_{12}, K_{14}	$[(K_{12}, 20), (K_{14}, 7), (K_{12}, 23), (K_{14}, 10), (K_{12}, 30), (K_{14}, 7), (K_{12}, 3)]^\omega$	16.98
9	K_4, $K_7, \ldots, K_{14}$	$[(K_{12}, 10), (K_{13}, 8), (K_8, 7), (K_{12}, 25), (K_8, 10), (K_{11}, 5), (K_{12}, 25), (K_8, 7), (K_{12}, 3)]^\omega$	8.31
12	$K_3, \ldots, K_{14}$	$[(K_4, 10), (K_9, 8), (K_5, 7), (K_{11}, 5), (K_4, 15), (K_{11}, 5), (K_5, 10), (K_8, 5), (K_{11}, 22), (K_5, 8), (K_{14}, 2), (K_4, 3)]^\omega$	3.26
13	$K_2, \ldots, K_{14}$	$[(K_4, 5), (K_{12}, 5), (K_8, 8), (K_3, 7), (K_{10}, 5), (K_{12}, 10), (K_4, 5), (K_{10}, 5), (K_3, 10), (K_7, 5), (K_{10}, 22), (K_3, 8), (K_{13}, 2), (K_4, 3)]^\omega$	4.09
14	$K_1, \ldots, K_{14}$	$[(K_4, 5), (K_{14}, 5), (K_7, 8), (K_2, 7), (K_9, 5), (K_{14}, 10), (K_4, 5), (K_9, 5), (K_2, 10), (K_6, 5), (K_9, 20), (K_2, 10), (K_4, 5)]^\omega$	2.63

Table 2. Results for batch reactor process with $N = 1$.

References

1. R. Alur and D. L. Dill. *Automata, Languages and Programming*, volume 443 of *LNCS*, chapter Automata for modeling real-time systems, pages 322–335. Springer, Berlin, April 1990.

2. D. Angeli and E. D. Sontag. Forward completeness, unboundedness observability, and their lyapunov characterizations. *Systems and Control Letters*, 38:209–217, 1999.
3. A. Anta and P. Tabuada. To sample or not to sample: self-triggered control for nonlinear systems. *IEEE Transaction on Automatic Control*, 55(9):2030–2042, 2010.
4. G. Behrmann, A. Fehnker, T. Hune, K. G. Larsen, P. Pettersson, J. Romijn, and F. W. Vaandrager. Minimum-cost reachability for priced timed automata. In M. Di Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, volume 2034 of *Lecture Notes in Computer Science*, pages 147–161. Springer Berlin Heidelberg, April 2001.
5. J. L. Boudec and P. Thiran. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, volume 2050 of *Lecture Notes in Computer Science*. Springer, 2001.
6. M. S. Branicky. Multiple lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):475–482, 1998.
7. S. Chakraborty, S. Künzli, and L. Thiele. A general framework for analysing system properties in platform-based embedded system designs. In *DATE*, volume 3, page 10190, 2003.
8. A. D'Innocenzo, G. Weiss, R. Alur, A. J. Isaksson, K. H. Johansson, and G. J. Pappas. Scalable scheduling algorithms for wireless networked control systems. In *IEEE International Conference on Automation Science and Engineering, CASE*, pages 409–414. IEEE, 2009.
9. D. Goswami, A. Masrur, R. Schneider, C. J. Xue, and S. Chakraborty. Multi-rate controller design for resource-and schedule-constrained automotive ecus. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 1123–1126. EDA Consortium, 2013.
10. M. Green and D. J. N. Limebeer. *Linear robust control*. Prentice Hall, August 1994.
11. J. P. Hespanha et al. Stability of switched systems with average dwell-time. In *Proceedings of the 38th IEEE Conference on Decision and Control*, volume 3, pages 2655–2660. IEEE, 1999.
12. H. K. Khalil. *Nonlinear systems*. Prentice-Hall, Inc., New Jersey, 2nd edition, 1996.
13. K. G. Larsen. Priced timed automata: Theory and tools. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS*, pages 417–425, 2009.
14. D. Nešic, A. Teel, and D. Carnevale. Explicit computation of the sampling period in emulation of controllers for nonlinear sampled-data systems. *IEEE Transactions on Automatic Control*, 54(3):619–624, 2009.
15. D. Nešić, A. R. Teel, and P. Kokotović. Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations. *Systems & Control Letters*, 38(4):259–270, 1999.
16. R. Raha, A. Hazra, A. Mondal, S. Dey, P. P. Chakrabarti, and P. Dasgupta. Synthesis of sampling modes for adaptive control. In *IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pages 294–299. IEEE, 2014.
17. J. I. Rasmussen, K. G. Larsen, and K. Subramani. Resource-optimal scheduling using priced timed automata. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 220–235. Springer, 2004.

18. A. Sharifi-Kolarijani, D. Adzkiya, and M. Mazo Jr. Symbolic abstractions for the scheduling of event-triggered control systems. *(to Appear) In Proceedings of 54st IEEE Conference on Decision and Control, Osaka, Japan*, December 2015.
19. E. D. Sontag. *Mathematical control theory*, volume 6. Springer-Verlag, New York, 2nd edition, 1998.
20. E. D. Sontag. Input to state stability: Basic concepts and results. In P. Nistri and G. Stefani, editors, *Nonlinear and Optimal Control Theory*, volume 1932 of *Lecture Notes in Mathematics*, pages 163–220. Springer Berlin Heidelberg, 2008.
21. P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, September 2007.
22. L. Thiele, S. Chakraborty, and M. Naedele. Real-time calculus for scheduling hard real-time systems. *IEEE International Symposium on Circuits and Systems. Emerging Technologies for the 21st Century.*, 4:101–104, 2000.
23. G. Weiss and R. Alur. Automata based interfaces for control and scheduling. In *Hybrid Systems: Computation and Control*, pages 601–613. Springer, 2007.
24. S. A. M. Wiesbaden. Autosar — the worldwide automotive standard for E/E systems. *ATZextra worldwide*, 18(9):5–12, 2013.

6 Appendix

6.1 ISS stability for linear control systems

Note that in the linear case any stabilizing gain K renders the closed-loop system (2.3) ISS with respect to measurement errors ε and one obtains the following specific ISS Lyapunov function characterization.

Theorem 5. *The closed-loop system $\dot{\xi} = A\xi + BK(\xi + \varepsilon)$ is ISS with respect to measurement errors ε if and only if $V(x) = x^T Px$ is an ISS Lyapunov function with $P \in \mathbb{R}^{n\times n}$ being a positive definite matrix satisfying the following linear matrix inequality (LMI)*

$$(A + BK)^T P + P(A + BK) \preceq -\kappa P, \tag{6.1}$$

for some constant $\kappa \in \mathbb{R}^+$.

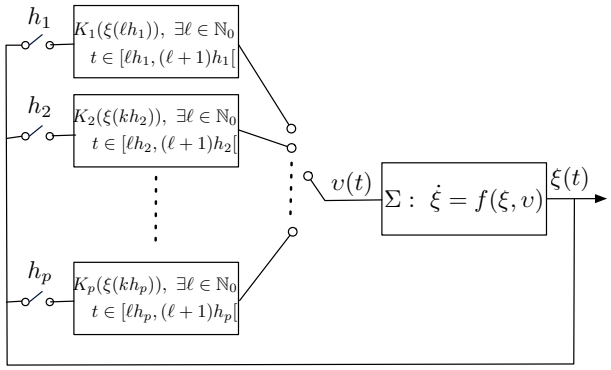

Fig. 6. Variable-rate control system $\hat{\Sigma}$.

6.2 (Priced) timed automata

We use the notion of *Timed Automaton*, introduced in [1], to describe real-time systems. A timed automaton (TA) is a directed graph equipped with real-valued variables (called clocks) modeling the logical clocks. We use C to denote a set of finitely many clocks. A clock constraint is defined by finitely many conjunctions of conditions of the form $x \sim a$ or $x - y \sim a$ with $x, y \in C$, $\sim \in \{\leq, <, =, >, \geq\}$, and $a \in \mathbb{Q}_0^+$. We use $\mathcal{B}(C)$ to denote the set of all clock constraints.

Definition 11. *A* timed automaton $\mathcal{T}$ *is a tuple $\mathcal{T} = (L, L_0, C, E, \mathsf{Inv})$ where*

- *L is a finite set of modes (or vertices);*

– $L_0 \subseteq L$ *is a set of initial modes;*
– C *is a set of finitely many real-valued clocks;*
– $E \subseteq L \times \mathcal{B}(C) \times 2^C \times L$ *is the set of edges;*
– $\mathsf{Inv} : L \to \mathcal{B}(C)$.

The semantics of a TA are defined as a transition system [**?**] where a state consists of the current location and the current value of the clocks. Two types of transitions between states are possible: the automaton may delay for some time (a delayed transition), or take an enabled edge (a discrete transition). Edges are labeled with a guard ($\in \mathcal{B}(C)$), described as a clock constraint, and a reset ($\in 2^C$). An edge transition can be taken when the value of the clocks satisfies the guard associated with the edge. Once an edge transition is taken a subset of all the clocks may be reset to zero. The model of timed automaton can be extended with a notion of price assigned to both locations and edges. Such extended automata are known as *Linearly Priced Timed Automaton* (LPTA) [4] as defined below.

Definition 12. *A* linearly priced timed automaton $\mathcal{T}$ *is a tuple* $\mathcal{T} = (L, L_0, C, E, \mathsf{Inv}, \mathcal{P})$ *where all the symbols carry their usual meanings as in TA and* $\mathcal{P} : (L \cup E) \to \mathbb{R}_0^+$ *is a function which assigns price to modes and edges.*

The semantics of LPTA is the same as a normal TA apart from the calculation of cost associated with each run possible in the underlying transition system. The price of a mode gives the cost rate of staying in that mode and the price of an edge transition gives the cost of executing that edge transition. Let us consider a run α in an LPTA with i) the sequence of nodes traversed being $l_0, \ldots, l_n$ each having prices $p_0, \ldots, p_n$, ii) the sequence of edge transitions executed being $e_1, \ldots, e_n$ with prices $q_1, \ldots, q_n$ and iii) the amount of time spent in each node being $t_0, \ldots, t_n$. For such a run, we have $\mathsf{cost}(\alpha) = \Sigma_{i=0}^{n} p_i \times t_i + \Sigma_{i=1}^{n} q_i$.

6.3 Real time calculus

Real Time Calculus (RTC), while rooted in Network Calculus [5], introduces several significant differences like modeling of remaining service. The RTC formalism was developed to capture the upper and lower limits of events and resource usage in order to model and analyze the performance of real-time embedded systems [7,22]. In the RTC formalism, the arrivals of input events of a system are specified using a (min,max) formalism. For example, the arrival pattern of a real-time task can be characterized as (t, a, b) implying that inside every time window of length t, the number of arrivals for such task instances is a minimum of a and a maximum of b.

Arrival patterns for real-time tasks are represented by a set of *arrival curves*, α^l and α^u, defined formally in Definition 2.9. The lower arrival curve $\alpha^l(t)$ and upper arrival curve $\alpha^u(t)$ represents the minimum and maximum number of events arriving within any time interval of length t. Similar to the arrival curves, we define *service curves*, $\beta^l(t)$ and $\beta^u(t)$, defined formally in Definition 2.10, to denote the resource capability of the processing unit over any time interval t.

Definition 13. *For a task θ_i, let $R[s,t[$ denote the total number of task instances arriving inside the time interval $[s,t[$. The* arrival curve *pair (α^l, α^u) for the task are functions from $\mathbb{R}_0^+$ to $\mathbb{N}$ such that $\alpha^l(t-s) \leq R[s,t[\leq \alpha^u(t-s)$ for any $s < t$, where $\alpha^u(0) = \alpha^l(0) = 0$.*

Definition 14. *For a resource p_i, let $C[s,t[$ denote the total number of free computation slots inside the time interval $[s,t[$. The* service curve *pair (β^l, β^u) for the resource are functions from $\mathbb{R}_0^+$ to $\mathbb{N}$ such that $\beta^l(t-s) \leq C[s,t[\leq \beta^u(t-s)$ for any $s < t$, where $\beta^u(0) = \beta^l(0) = 0$.*

6.4 Proof of Theorem 4.4

Proof. We show the result for the case of infinite number of switches. A proof for the case of finite switches can be written in a similar way. Let $a \in \mathbb{R}^n$ be any initial condition, $t_0 = 0$, and let $p_i \in \{1, \ldots, p\}$ denote the value of the switching signal on the interval $[t_i, t_{i+1}[$, for $i \in \mathbb{N}_0$. For all $i \in \mathbb{N}_0$ and $t \in [t_i, t_{i+1}[$ and using inequality (4.1), one gets

$$\dot{V}_{p_i}(\xi_{av}(t)) \leq -\hat{\kappa}_{p_i} V_{p_i}(\xi_{av}(t)).$$

For all $i \in \mathbb{N}_0$, $t \in [t_i, t_{i+1}]$, and by continuity of V_{p_i}, we have

$$V_{p_i}(\xi_{av}(t)) \leq V_{p_i}(\xi_{av}(t_i)) \mathsf{e}^{-\hat{\kappa}_{p_i}(t-t_i)}. \tag{6.2}$$

Particularly, for $t = t_{i+1}$ $\forall\, i \in \mathbb{N}_0$ and using Assumption 4.3, one gets

$$V_{p_{i+1}}(\xi_{av}(t_{i+1})) \leq \mu_{p_{i+1}p_i} \mathsf{e}^{-\hat{\kappa}_{p_i}(t_{i+1}-t_i)} V_{p_i}(\xi_{av}(t_i)).$$

Then, by induction, we have that for all $i \in \mathbb{N}_0$

$$\begin{aligned} V_{p_i}(\xi_{av}(t_i)) \leq{}& \mu_{p_i p_{i-1}} \mathsf{e}^{-\hat{\kappa}_{p_{i-1}}(t_i - t_{i-1})} \times \\ & \mu_{p_{i-1}p_{i-2}} \mathsf{e}^{-\hat{\kappa}_{p_{i-2}}(t_{i-1}-t_{i-2})} \ldots \mu_{p_1 p_0} \mathsf{e}^{-\hat{\kappa}_{p_0}(t_1 - t_0)} V_{p_0}(a). \end{aligned} \tag{6.3}$$

Combining (4.3) and (4.4), for all $i \in \mathbb{N}_0$ and $t \in [t_i, t_{i+1}]$, one obtains

$$\begin{aligned} V_{p_i}(\xi_{av}(t)) \leq{}& \mathsf{e}^{-\hat{\kappa}_{p_i}(t-t_i)} \mu_{p_i p_{i-1}} \mathsf{e}^{-\hat{\kappa}_{p_{i-1}}(t_i - t_{i-1})} \times \\ & \mu_{p_{i-1}p_{i-2}} \mathsf{e}^{-\hat{\kappa}_{p_{i-2}}(t_{i-1}-t_{i-2})} \ldots \mu_{p_1 p_0} \mathsf{e}^{-\hat{\kappa}_{p_0}(t_1 - t_0)} V_{p_0}(a). \end{aligned}$$

Since we consider only switching signals in $\mathcal{S}$, i.e. such that $\exists\, \tau_{p_i p_{i+1}} \in \mathbb{Q}_0^+$: $\tau_{p_i p_{i+1}} \leq t_{i+1} - t_i$ for any $i \in \mathbb{N}_0$, one can further bound as:

$$\begin{aligned} V_{p_i}(\xi_{av}(t)) &\leq \mathsf{e}^{-\hat{\kappa}_{p_i}(t-t_i)} \mu_{p_i p_{i-1}} \mathsf{e}^{-\hat{\kappa}_{p_{i-1}} \tau_{p_{i-1}p_i}} \times \\ &\quad \mu_{p_{i-1}p_{i-2}} \mathsf{e}^{-\hat{\kappa}_{p_{i-2}} \tau_{p_{i-2}p_{i-1}}} \ldots \mu_{p_1 p_0} \mathsf{e}^{-\hat{\kappa}_{p_0} \tau_{p_0 p_1}} V_{p_0}(a) \\ &\leq \mu_{p_i p_{i-1}} \mathsf{e}^{-\hat{\kappa}_{p_{i-1}} \tau_{p_{i-1}p_i}} \times \\ &\quad \mu_{p_{i-1}p_{i-2}} \mathsf{e}^{-\hat{\kappa}_{p_{i-2}} \tau_{p_{i-2}p_{i-1}}} \ldots \mu_{p_1 p_0} \mathsf{e}^{-\hat{\kappa}_{p_0} \tau_{p_0 p_1}} V_{p_0}(a). \end{aligned}$$

Finally, since $\log \frac{\mu_{ij}}{\rho} < \hat{\kappa}_j \tau_{ji}$, for any $i, j \in \{1, \ldots, p\}$, $i \neq j$, and some $\rho \in]0, 1[$, we obtain:

$$V_{p_i}(\xi_{av}(t)) \leq \rho^i V_{p_0}(a). \tag{6.4}$$

Using the first set of inequalities in Definition 2.4 and (4.5), one gets

$$\underline{\alpha}_{p_i}(\|\xi_{av}(t)\|) \leq V_{p_i}(\xi_{av}(t)) \leq \rho^i V_{p_0}(a) \leq \rho^i \overline{\alpha}_{p_0}(\|a\|),$$

which reduces to

$$\|\xi_{av}(t)\| \leq \underline{\alpha}_{p_i}^{-1}(\rho^i \overline{\alpha}_{p_0}(\|a\|)), \tag{6.5}$$

due to $\underline{\alpha} \in \mathcal{K}_\infty$. As t goes to infinity and since the number of switches are infinite (i.e. $i \to \infty$), from (**??**) we conclude that $\|\xi_{av}(t)\|$ converges to zero which completes the proof.

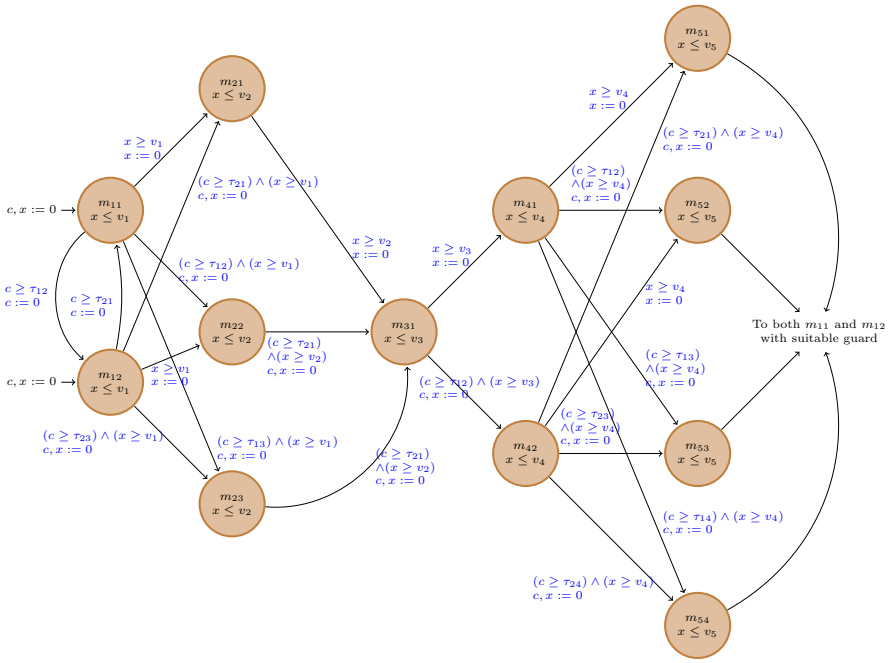

Fig. 7. Linearly Priced Timed Automaton for Stable and Schedulable Switching Sequences

6.5 Aircraft pitch control

The matrices A and B are given in [**?**] as

$$A = \begin{bmatrix} -0.313 & 56.7 & 0 \\ -0.014 & -0.43 & 0 \\ 0 & 56.7 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0232 \\ 0.0203 \\ 0 \end{bmatrix}.$$

We consider a maximum of $p = 6$ controllers. Due to lack of space, we do not provide matrices K_i and M_i. For $i \in \{1, \ldots, p\}$, the sampling periods are $h_i = 10^{-7} \times \{1.39, 3.13, 8.90, 43.0, 48.7, 109\}$. The values of τ_{ij} are:

$$[\tau_{ij}] = \begin{bmatrix} 0 & 0.31 & 0.37 & 0.45 & 1.15 & 0.80 \\ 0.48 & 0 & 0.35 & 0.46 & 1.30 & 0.86 \\ 0.90 & 0.65 & 0 & 0.49 & 1.52 & 0.98 \\ 1.70 & 1.39 & 0.99 & 0 & 1.89 & 1.15 \\ 9.62 & 8.83 & 7.82 & 6.45 & 0 & 4.48 \\ 3.51 & 3.07 & 2.52 & 1.76 & 1.83 & 0 \end{bmatrix}.$$

Consider the controllers WCETs, $\omega_i^c = 10^{-3} \times \{1, 2, 5, 23, 25, 40\}$. The bandwidth requirements for the controllers are computed as r_i=$\{0.73, 0.64, 0.57, 0.54, 0.52, 0.37\}$. Given the $\underline{\sigma}_T$ for the current platform load, the list S of admissible controllers can be found. We then use our modeling method from section 4.3 and create the LPTA specification for this scenario in UPPAAL CORA [13]. The resulting minimum cost schedule is given by the switching sequence in[5] Table **??** for different values of unfolding N. The corresponding cost value obtained from UPPAAL CORA is also included.

p	Controllers	N	Switching Sequence	Cost
For $\underline{\sigma}_{T1}$				
2	K_5, K_6	1	$[(K_6, 60)]^\omega$	13.80
3	K_4, K_5, K_6	1	$[(K_4, 20), (K_5, 30), (K_4, 10)]^\omega$	8.70
4	K_3, K_4, K_5, K_6	1	$[(K_3, 20), (K_6, 10), (K_5, 20), (K_3, 10)]^\omega$	8.10
		2	$[(K_3, 20), (K_6, 10), (K_5, 20), [(K_3, 30), (K_5, 30)]^*, (K_3, 10)]^\omega$	15.90
5	$K_3, \ldots, K_6$	2	$[(K_2, 20), (K_4, 14), (K_5, 6), (K_6, 10), [(K_2, 30), (K_6, 10), (K_5, 10), (K_6, 10)]^*, (K_2, 10)]^\omega$	20.90
For $\underline{\sigma}_{T2}$				
2	K_5, K_6	1	$[(K_6, 100)]^\omega$	20.00
3	K_4, K_5, K_6	1	$[(K_4, 25), (K_5, 25), (K_4, 18), (K_5, 22), (K_4, 10)]^\omega$	12.01
4	$K_3, \ldots, K_6$	2	$[(K_3, 26), (K_5, 24), (K_3, 18), (K_6, 10), (K_5, 12), [(K_3, 35), (K_5, 25), (K_3, 18), (K_6, 10), (K_5, 12)]^*, (K_3, 10)]^\omega$	22.00
5	$K_2, \ldots, K_6$	2	$[(K_3, 26), (K_5, 24), (K_3, 18), (K_6, 10), (K_5, 12), [(K_3, 35), (K_5, 25), (K_3, 18), (K_6, 10), (K_5, 12)]^*, (K_3, 10)]^\omega$	26.95
6	$K_1, \ldots, K_6$	2	$[(K_1, 26), (K_4, 9), (K_5, 10), (K_1, 20), (K_3, 10), (K_4, 15), [(K_1, 35), (K_4, 9), (K_5, 11), (K_1, 20), (K_3, 10), (K_4, 15)]^*, (K_1, 10)]^\omega$	25.44

Table 3. Aircraft pitch control simulation results

6.6 Ball and Beam

The matrices A and B are given in [**?**] as

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

[5] The operator $[\cdots]^*$ in the table is defined as follows: $\forall$ N, the switching sequence for any further unfolding $N + a$, is obtained by repeating the sequence under $[\cdots]^*$, a times.

We consider a maximum of $p = 8$ controllers. For $i \in \{1, \ldots, p\}$, the sampling periods are $h_i = 10^{-5} \times \{3.23, 5.18, 8.29, 13.1, 18.4, 19.7, 26.6, 28.2\}$. The values of τ_{ij} are:

$$[\tau_{ij}] = \begin{bmatrix} 0 & 0.4 & 0.4 & 0.5 & 0.5 & 0.5 & 0.4 & 0.4 \\ 1 & 0 & 0.4 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 1.9 & 1.1 & 0 & 0.5 & 0.7 & 0.6 & 0.6 & 0.6 \\ 3 & 2.3 & 1.4 & 0 & 0.8 & 0.6 & 0.7 & 0.7 \\ 26 & 23 & 21 & 18 & 0 & 14 & 11 & 7 \\ 4.5 & 3.7 & 2.9 & 1.8 & 1.1 & 0 & 0.8 & 0.9 \\ 7 & 6 & 5 & 4 & 1.3 & 2.4 & 0 & 1 \\ 12 & 11 & 9 & 8 & 2 & 6 & 4 & 0 \end{bmatrix}.$$

Consider the controllers WCETs $\omega_i^c = \{0.3, 0.4, 0.5, 0.8, 0.8, 0.8, 0.9, 0.9\}$. The bandwidth requirements for the controllers are computed as $r_i = \{0.93, 0.78, 0.62, 0.61, 0.44, 0.41, 0.34, 0.32\}$. The simulation results are summarized in Table **??**.

p	Controllers	N	Switching Sequence	Cost
For $\underline{\sigma}_{T1}$				
2	K_5, K_8	1	$[(K_8, 60)]^\omega$	16.80
3	K_5, K_7, K_8	1	$[(K_8, 60)]^\omega$	15.60
4	$K_5, \ldots, K_8$	2	$[(K_7, 24), (K_8, 26), [(K_7, 30), (K_8, 30)]^*, (K_7, 10)]^\omega$	13.78
5	$K_4, \ldots, K_8$	1	$[(K_5, 15), (K_6, 45)]^\omega$	14.40
For $\underline{\sigma}_{T2}$				
2	K_5, K_8	1	$[(K_8, 100)]^\omega$	19.50
3	K_5, K_7, K_8	1	$[(K_8, 100)]^\omega$	17.50
4	$K_5, \ldots, K_8$	2	$[(K_7, 25), (K_8, 25), (K_7, 24), (K_8, 16), [(K_7, 35), (K_8, 25), (K_7, 24), (K_8, 16)]^*, (K_7, 10)]^\omega$	29.26
5	$K_4, \ldots, K_8$	1	$[(K_5, 10), (K_7, 8), (K_6, 24), (K_7, 8), (K_6, 45), (K_5, 5)]^\omega$	18.18
		2	$[(K_5, 10), (K_7, 10), (K_6, 40), (K_7, 30), (K_6, 40), (K_5, 15), (K_6, 50), (K_5, 5)]^\omega$	35.80
		3	$[(K_5, 10), (K_7, 10), (K_6, 40), (K_7, 30), (K_6, 40), (K_5, 15), [(K_6, 24), (K_7, 21), (K_6, 40), (K_5, 15)]^*, (K_6, 50), (K_5, 5)]^\omega$	-

Table 4. Simulation results for ball and beam

6.7 Cruise Control

The matrices A and B are given in [**?**] as

$$A = [-0.05]\,,\ B = [0.001]\,.$$

We consider a maximum of $p = 9$ controllers. For $i \in \{1, \ldots, p\}$, the sampling periods are $h_i = 10^{-2} \times \{7.48, 8.42, 9.64, 11.3, 13.6, 17, 22.9, 34.8, 72.7\}$. The values

of τ_{ij} are:

$$[\tau_{ij}] = \begin{bmatrix} 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.3 & 0.3 & 0.3 & 0.3 & 0 & 0.3 & 0.3 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0 & 0.3 & 0.3 & 0.3 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0 & 0.5 & 0.5 \\ 0.7 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 & 0.7 & 0 & 0.7 \\ 1.4 & 1.4 & 1.4 & 1.4 & 1.4 & 1.4 & 1.4 & 1.4 & 0 \end{bmatrix}.$$

Consider the controllers WCETs $\omega_i^c = \{600, 600, 600, 600, 600, 600, 600, 1400, 1400\}$. The bandwidth requirements for the controllers are computed as $r_i = \{0.84, 0.72, 0.66, 0.54, 0.48, 0.42, 0.36, 0.30, 0.28\}$. The simulation results are summarised in Table **??**.

p	Controllers	N	Switching Sequence	Cost
For $\underline{\sigma}_{T1}$				
2	K_8, K_9	2	$[(K_9, 15), (K_8, 14), [(K_9, 21), (K_8, 14), (K_9, 11), (K_8, 14)]^*$, $(K_9, 17), (K_8, 14)]^\omega$	29.44
4	$K_6, \ldots, K_9$	1	$[(K_9, 15), (K_8, 5), (K_9, 30), (K_8, 10)]^\omega$	11.70
8	$K_2, \ldots, K_9$	1	$[(K_9, 15), (K_4, 5), (K_7, 10), (K_9, 15), (K_7, 5), (K_4, 10)]^\omega$	3.90
9	$K_1, \ldots, K_9$	1	$[(K_9, 10), (K_8, 5), (K_3, 5), (K_6, 10), (K_8, 10), (K_9, 5),$ $(K_6, 5), (K_3, 10)]^\omega$	5.10
For $\underline{\sigma}_{T2}$				
2	K_8, K_9	1	$[(K_9, 20), (K_8, 14), (K_9, 16), (K_8, 14), (K_9, 22), (K_8, 14)]^\omega$	15.38
4	$K_6, \ldots, K_9$	1	$[(K_9, 20), (K_8, 5), (K_9, 25), (K_8, 10), (K_9, 30), (K_8, 5),$ $(K_9, 5)]^\omega$	10.70
8	$K_2, \ldots, K_9$	1	$[(K_9, 10), (K_6, 10), (K_4, 5), (K_7, 5), (K_9, 15), (K_7, 5),$ $(K_4, 10), (K_5, 5), (K_7, 25), (K_4, 5), (K_9, 5)]^\omega$	6.65
9	$K_1, \ldots, K_9$	1	$[(K_9, 5), (K_8, 5), (K_5, 10), (K_3, 5), (K_6, 5), (K_8, 10), (K_9, 5)$, $(K_6, 5), (K_3, 10), (K_4, 5), (K_6, 25), (K_3, 5), (K_9, 5)]^\omega$	6.85

Table 5. Simulation results for cruise control

6.8 Inverted Pendulum

The matrices A and B are given in [**?**] as

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.18 & 2.67 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.45 & 31.18 & 0 \end{bmatrix} \text{ and } B = \begin{bmatrix} 0 \\ 1.82 \\ 0 \\ 4.55 \end{bmatrix}.$$

We consider a maximum of $p = 7$ controllers. For $i \in \{1, \ldots, p\}$, the sampling periods are $h_i = 10^{-5} \times \{2.44, 2.98, 3.48, 4.78, 5.91, 6.11, 6.83\}$. The values of τ_{ij}

are:

$$[\tau_{ij}] = \begin{bmatrix} 0 & 0.83 & 0.41 & 0.50 & 0.74 & 0.51 & 0.61 \\ 21.03 & 0 & 18.70 & 16.11 & 6.54 & 13.44 & 10.38 \\ 1.08 & 0.98 & 0 & 0.50 & 0.87 & 0.60 & 0.72 \\ 2.14 & 1.15 & 1.34 & 0 & 1.01 & 0.63 & 0.83 \\ 9.40 & 1.79 & 8.14 & 6.70 & 0 & 5.15 & 3.24 \\ 3.50 & 1.39 & 2.70 & 1.68 & 1.20 & 0 & 0.88 \\ 5.50 & 1.65 & 4.57 & 3.46 & 1.26 & 2.16 & 0 \end{bmatrix}.$$

Consider the controllers WCETs $\omega_i^c = \{0.23\}$. The bandwidth requirements for the controllers are computed as $r_i = \{0.95, 0.78, 0.67, 0.49, 0.39, 0.38, 0.34\}$. The simulation results are summarized in Table **??**.

p	Controllers	Switching Sequence	Cost
For $\underline{\sigma}_{T1}$			
2	K_2, K_5	$[(K_5, 60)]^\omega$	12.60
3	K_2, K_5, K_7		
4	$K_2, K_5, \ldots, K_7$		
5	$K_2, K_4, \ldots, K_7$	$[(K_6, 20), (K_5, 30), (K_6, 10)]^\omega$	12.90
6	$K_2, \ldots, K_7$	$[(K_4, 20), (K_7, 14), (K_5, 9), (K_7, 7), (K_4, 10)]^\omega$	11.28
7	$K_1, \ldots, K_7$	$[(K_2, 8), (K_7, 7), (K_3, 45)]^\omega$	12.61
For $\underline{\sigma}_{T2}$			
2	K_2, K_5	$[(K_5, 100)]^\omega$	14.00
3	K_2, K_5, K_7		
4	$K_2, K_5, \ldots, K_7$		
5	$K_2, K_4, \ldots, K_7$	$[(K_5, 10), (K_6, 17), (K_5, 23), (K_6, 17), (K_5, 16), (K_6, 17)]^\omega$	17.61
6	$K_2, \ldots, K_7$	$[(K_4, 30), (K_5, 15), (K_4, 20), (K_7, 25), (K_4, 10)]^\omega$	15.90
7	$K_1, \ldots, K_7$	$[(K_2, 5), (K_7, 5), (K_6, 10), (K_3, 17), (K_6, 13), (K_3, 45), (K_2, 5)]^\omega$	15.32

Table 6. Results for inverted pendulum with $N = 1$.

6.9 Magnetically Suspended Ball

The matrices A and B are given in [**?**] as

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 980 & 0 & -2.8 \\ 0 & 0 & -100 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 0 \\ 100 \end{bmatrix}.$$

We consider a maximum of $p = 10$ controllers. For $i \in \{1, \ldots, p\}$, the sampling periods are $h_i = 10^{-12} \times \{8.61, 8.94, 9.23, 9.31, 9.62, 9.66, 9.86, 9.93, 10, 10\}$. The

values of τ_{ij} are:

$$[\tau_{ij}] = \begin{bmatrix} 0 & 2.7 & 1.5 & 2.6 & 2.4 & 1.7 & 2.3 & 1.8 & 2.1 & 2 \\ 0.3 & 0 & 0.3 & 0.2 & 0.2 & 0.3 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.8 & 1.3 & 0 & 1.2 & 1.1 & 0.8 & 1.1 & 0.8 & 1.0 & 0.9 \\ 0.3 & 0.2 & 0.3 & 0 & 0.2 & 0.3 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.3 & 0.2 & 0.3 & 0.2 & 0 & 0.3 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.6 & 0.8 & 0.6 & 0.8 & 0.7 & 0 & 0.7 & 0.5 & 0.6 & 0.6 \\ 0.4 & 0.3 & 0.3 & 0.2 & 0.2 & 0.3 & 0 & 0.3 & 0.2 & 0.3 \\ 0.5 & 0.6 & 0.5 & 0.5 & 0.5 & 0.4 & 0.5 & 0 & 0.4 & 0.4 \\ 0.5 & 0.3 & 0.4 & 0.3 & 0.3 & 0.4 & 0.3 & 0.3 & 0 & 0.3 \\ 0.5 & 0.4 & 0.4 & 0.4 & 0.4 & 0.3 & 0.3 & 0.3 & 0.3 & 0 \end{bmatrix}.$$

Consider the controllers WCETs $\omega_i^c = 10^{-8} \times \{3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 3.5, 7, 7\}$. The bandwidth requirements for the controllers are computed as $r_i = \{0.82, 0.79, 0.38, 0.38, 0.37, 0.37, 0.36, 0.36, 0.35, 0.35\}$. The simulation results are summarized in Tables **??** and **??**.

p	Controllers	N	Switching Sequence	Cost
2	K_1, K_3	1	$[(K_3, 60)]^\omega$	10.20
4	K_1, K_3, K_6, K_8			
6	K_1, K_3, K_6, $K_8, \ldots, K_{10}$			
9	K_1, $K_3, \ldots, K_{10}$	1	$[(K_3, 26), (K_6, 3), (K_3, 13), (K_6, 3), (K_3, 15)]^\omega$	10.20
		3, 4	$[(K_6, 10), (K_3, 15), (K_6, 3), (K_3, 13), (K_6, 3), (K_3, 13), (K_6, 3), [(K_3, 25), (K_6, 3), (K_3, 13), (K_6, 3), (K_3, 13), (K_6, 3)]^*, (K_3, 26), (K_6, 3), (K_3, 13), (K_6, 3), (K_3, 15)]^\omega$	30.60, 40.80
		5	$[(K_3, 25), (K_6, 3), (K_3, 13), (K_6, 3), (K_3, 13), (K_6, 3), (K_3, 25), (K_6, 3), (K_3, 13), (K_6, 3), (K_3, 13), (K_6, 3), (K_3, 25), (K_6, 3), (K_3, 13), (K_6, 3), (K_3, 13), (K_6, 3), (K_3, 25), (K_6, 3), (K_3, 13), (K_6, 3), (K_3, 13), (K_6, 3), (K_3, 26), (K_6, 3), (K_3, 13), (K_6, 3), (K_3, 15)]^\omega$	51.00
10	$K_1, \ldots, K_{10}$	1	$[(K_3, 15), (K_8, 15), (K_3, 15), (K_6, 15)]^\omega$	3.90
		5	$[(K_3, 15), (K_6, 13), (K_8, 2), (K_3, 15), (K_6, 13), (K_8, 2), (K_3, 15), (K_8, 15), (K_3, 15), (K_8, 15), (K_3, 15), (K_6, 13), (K_8, 2), (K_3, 15), (K_6, 13), (K_8, 2), (K_3, 15), (K_6, 13), (K_8, 2), (K_3, 15), (K_8, 15), (K_3, 15), (K_6, 13), (K_8, 2), (K_3, 15), (K_6, 15)]^\omega$	19.50

Table 7. Results for levitating ball with $\underline{\sigma}_{T1}$

p	Controllers	N	Switching Sequence	Cost
2	K_1, K_3	1	$[(K_3, 100)]^\omega$	11.00
6	$K_1, K_3, K_6, K_8, \ldots, K_{10}$			
8	$K_1, K_3, K_5, \ldots, K_{10}$			
9	$K_1, K_3, \ldots, K_{10}$	1	$[(K_6, 10), (K_3, 15), (K_6, 3), (K_3, 13), (K_6, 3), (K_3, 13), (K_6, 3), (K_3, 26), (K_6, 3), (K_3, 13), (K_6, 3), (K_3, 15)]^\omega$	11.00
		3	$[(K_6, 5), (K_3, 21), (K_6, 3), (K_3, 13), (K_6, 3), [(K_3, 36), (K_6, 3), (K_3, 13), (K_6, 3), (K_3, 26), (K_6, 3), (K_3, 13), (K_6, 3)]^*, (K_3, 39), (K_6, 3), (K_3, 13)]^\omega$	33.00
10	$K_1, \ldots, K_{10}$	1	$[(K_3, 10), (K_6, 18), (K_8, 2), (K_3, 15), (K_8, 35), (K_6, 13), (K_8, 2), (K_3, 5)]^\omega$	4.70

Table 8. Results for levitating ball with $\underline{\sigma}_{T2}$

6.10 Suspension System

The matrices A and B are given in [**?**] as

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -6.57 & 0 & -25.26 & -0.14 \\ 46.94 & 0 & -48.17 & 1 \\ 1562.5 & 0 & -1844.5 & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0 & 0 \\ 0 & 6.57 \\ 0 & -46.94 \\ 0 & -1562.5 \end{bmatrix}.$$

We consider a maximum of $p = 9$ controllers. For $i \in \{1, \ldots, p\}$, the sampling periods are $h_i = 10^{-5} \times \{4.96, 5.79, 5.95, 6.18, 6.80, 6.85, 8.39, 9.42, 10.7\}$. The values of τ_{ij} are:

$$[\tau_{ij}] = \begin{bmatrix} 0 & 0.3 & 0.3 & 0.4 & 0.5 & 0.5 & 0.5 & 0.4 & 0.5 \\ 0.4 & 0 & 0.3 & 0.4 & 0.6 & 0.5 & 0.5 & 0.5 & 0.6 \\ 0.7 & 0.6 & 0 & 0.4 & 0.7 & 0.5 & 0.6 & 0.5 & 0.7 \\ 1.2 & 1 & 0.7 & 0 & 0.7 & 0.4 & 0.6 & 0.6 & 0.8 \\ 7.0 & 6.6 & 6 & 5.3 & 0 & 4.5 & 3.9 & 1.6 & 3.9 \\ 1.8 & 1.6 & 1.2 & 0.8 & 0.8 & 0 & 0.5 & 0.7 & 1 \\ 2.5 & 2.3 & 1.9 & 1.4 & 1 & 0.9 & 0 & 0.8 & 1.2 \\ 13.8 & 13.1 & 11.9 & 10.6 & 3.3 & 9.2 & 7.9 & 0 & 7.6 \\ 3.4 & 3.1 & 2.8 & 2.4 & 1.2 & 2 & 1.7 & 1 & 0 \end{bmatrix}.$$

Consider the controllers WCETs $\omega_i^c = \{0.35\}$. The bandwidth requirements for the controllers are computed as $r_i = \{0.71, 0.61, 0.59, 0.57, 0.52, 0.52, 0.42, 0.38, 0.33\}$. The simulation results are summarised in Table **??**.

p	Controllers	N	Switching Sequence	Cost
For $\underline{\sigma}_{T1}$				
2	K_5, K_8	1	$[(K_5, 34), (K_8, 16), (K_5, 10)]^\omega$	10.96
		2	$(K_5, 34), (K_8, 16), [(K_5, 34), (K_8, 26)]^\omega, (K_5, 10)$	20.52
6	$K_4, \ldots, K_9$	4	$[(K_4, 21), (K_9, 8), (K_8, 21), [(K_4, 30), (K_8, 30)]^*, (K_4, 10)]^\omega$	36.64
8	$K_2, \ldots, K_9$	1	$[(K_2, 21), (K_6, 8), (K_9, 7), (K_5, 14), (K_2, 10)]^\omega$	12.36
		2	$[(K_2, 21), (K_6, 8), (K_9, 7), (K_5, 14), (K_2, 30), (K_5, 10), (K_8, 13), (K_5, 7), (K_2, 10)]^\omega$	23.50
For $\underline{\sigma}_{T2}$				
2	K_5, K_8	1	$[(K_5, 65), (K_8, 25), (K_5, 10)]^\omega$	15.50
		2	$[\pi_1, [(K_5, 25), (K_8, 25), (K_5, 50)]^*]^\omega$	31.00
4	$K_5, K_7, \ldots, K_9$	3	$[(K_9, 65), (K_8, 25), (K_9, 35), (K_8, 25), (K_9, 80), (K_5, 20), (K_7, 15), (K_8, 25), (K_9, 10)]^\omega$	50.20
		4	$[(K_9, 65), (K_8, 25), (K_9, 35), (K_8, 25), (K_9, 75), (K_8, 25), (K_9, 80), (K_5, 20), (K_7, 15), (K_8, 25), (K_9, 10)]^\omega$	66.95
		5	$[(K_9, 65), [(K_8, 25), (K_9, 35), (K_8, 25), (K_9, 80), (K_5, 20), (K_7, 15)]^*, (K_8, 25), (K_9, 10)]^\omega$	83.65

Table 9. Results for suspension system