Analysis and Design of Real-Time Servers for Control Applications

Amir Aminifar, Enrico Bini, Petru lon Eles and Zebo Peng

Amir Aminifar, Enrico Bini, Petru lon Eles and Zebo Peng, Analysis and Design of Real-Time Servers for Control Applications, 2016, I.E.E.E. transactions on computers (Print), (65), 3, 834-846.

http://dx.doi.org/10.1109/TC.2015.2435789

Analysis and Design of Real-Time Servers for Control Applications

Amir Aminifar, Enrico Bini, Petru Eles, Zebo Peng

Abstract—Today, a considerable portion of embedded systems, e.g., automotive and avionic, comprise several control applications. Guaranteeing the stability of these control applications in embedded systems, or cyber-physical systems, is perhaps the most fundamental requirement while implementing such applications. This is different from the classical hard real-time systems where often the acceptance criterion is meeting the deadline. In other words, in the case of control applications, guaranteeing stability is considered to be a main design goal, which is linked to the amount of delay and jitter a control application can tolerate before instability. This advocates the need for new design and analysis techniques for embedded real-time systems running control applications.

In this paper, the analysis and design of such systems considering a server-based resource reservation mechanism are addressed. The benefits of employing servers are manifold: providing a compositional and scalable framework, protection against other tasks' misbehaviors, and systematic bandwidth assignment and co-design. We propose a methodology for designing bandwidth-optimal servers to stabilize control tasks. The pessimism involved in the proposed methodology is both discussed theoretically and evaluated experimentally.

Index Terms—Embedded Systems, Real-Time Systems, Real-Time Control Co-Design, Control Server, Stability, Bandwidth Minimization

1 Introduction

In embedded systems, controllers are usually implemented by software tasks, which read some input data, perform some computation, and then apply the computed signal to the plant to be controlled. When other tasks execute on the same computing unit, then the schedule of the control task is also affected by the other tasks sharing the same processing unit. As a result, the control algorithm may experience considerable amount of delay and jitter, which affect the control performance and stability of the plant.

Today, the literature does provide some results that account for the effect of the controller schedule on the system dynamics. For example, the effect on the control performance of the delay from the sensing to the actuation [1] or the effect of the jitter in the task completion are well understood [2].

Once the effect of the scheduling on the control performance is established, it is possible to perform the, so called, *real-time control co-design*: designing a controller so that the required control performance is guaranteed (stability, LQR cost minimization, etc.) and the control tasks are schedulable on the available processing unit.

In typical approaches [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], the control tasks are all designed together in a way that some global cost (function of the control cost of the individual tasks) is minimized.

The research leading to these results was supported by the ELLIIT Excellence Center, the Linneaus Center LCCC, the Marie Curie Intra European Fellowship within the 7th European Community Framework Programme, and the Swedish Research Council.

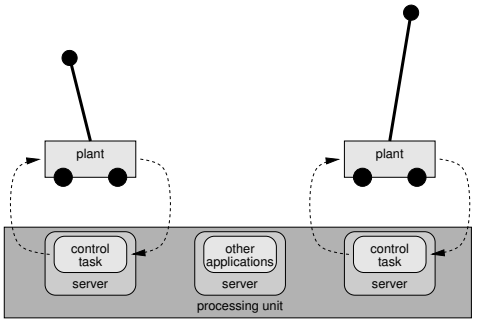


Fig. 1. Overview of the proposed approach.

By following this approach, however, the design of each control task is affected by the other control tasks, hence breaking the key engineering design principle of separation of concerns. In this paper, we propose instead to run each controller within its own server, which then isolates each control task in the execution environment (see Figure 1).

The usage of servers for control tasks presents the following advantages:

- it provides compositionality that is essential for systematic system design methodologies;
- the complexity of the design scales linearly with the number of applications;
- it protects each controller from possible misbehaviors, which may occur within other tasks and then possibly jeopardize the entire system;
- the bandwidth assignment, rather than the priority assignment, may constitute a more accurate instrument to allocate the available computing

resources;
- the simple interface provided by the resource reservation mechanism facilitates the controller–server co-design process [13], [14];
- running the controller over a dedicated server, may reduce significantly the jitter of the controller, especially if the server period is smaller than the period of the controller. In short, this is due to the fact that the server guarantees the control task a certain resource bandwidth. This is important since it is often possible to compensate for the constant part of delay, while the process of coping with the jitter is more involved.

1.1 Related work

Over the past decade, the analysis and design of real-time servers have widely developed. Feng and Mok [15] introduced the *bounded delay resource* model to facilitate hierarchical resource sharing. The schedulability analysis and server design problems for real-time applications under the *periodic resource* model have been addressed by [16], [17], [18], [19]. Easwaran et. al. [20] extended the periodic resource model to the *explicit deadline periodic* model (EDP) and developed an algorithm to compute a bandwidth optimal EDP model based abstraction. Similar to what we do in this paper, Fisher and Dewan [21] described a method to minimize the bandwidth of a server. They developed a fully-polynomial-time approximation scheme (FPTAS) to solve the problem. However, as the majority of the work in this area, they consider the task deadlines as constraints rather than the stability of the controllers.

More relevant to this work, Cervin and Eker [13] proposed the control server approach which provides a simple interface used for control-scheduling co-design of real-time systems. Fontanelli et. al. [22] addressed the problem of optimal bandwidth allocation for a set of control tasks under the time-triggered model. While exploiting this model can simplify the analysis and design problems to a great extent, by removing the element of jitter, such methods are restricted solely to the very particular time-triggered design and implementation approach that can potentially lead to under-utilization of resources or poor control performance [23]. Recently, Fontanelli et. al. propose a new model for real-time control applications [24] to investigate stochastic stability, but ignoring the dependencies among stochastic variables. In our previous work [25], we have considered the analysis and design of bandwidth-efficient control servers while guaranteeing stability. In this work, we extend our previous work [25], provide theoretical foundation and broaden the experimental evaluation. The theoretical results quantify the amount of pessimism, in the worst-case, in our proposed approach, while the experimental results quantify this pessimism in practice. In [14], we extend [25] towards a different direction presenting a controller–server co-design approach where the controller is determined in a unified process along with the server parameters.

1.2 Contributions of the paper

While the analysis and design problems of real-time servers have been discussed to a considerable degree, the server-based approach has gained less attention in the case of control applications which are fundamentally different from real-time applications with hard deadlines. In particular, as opposed to hard real-time applications, the notion of deadline is considered to be artificial for control applications. In contrast to hard real-time systems, control stability is the main property to be guaranteed for control applications. Therefore, in the case of control applications, worst-case control performance and stability should be considered instead of worst-case response time and deadline.

To approach the problem of designing stabilizing servers, the first step is to capture the stability of the controllers in terms of real-time parameters, which is facilitated by the Jitter Margin toolbox [26], [5], [2]. The stability of control applications, hence, depends not only on the amount of delay, but also on the amount of jitter the application experiences [27]. The second step is to derive analysis methods for the servers to compute the discussed real-time metrics, i.e., delay and jitter. To this end, we consider the explicit deadline periodic model and develop the worst-case and best-case response times for tasks with arbitrary deadlines within explicit deadline periodic servers with arbitrary deadlines. Having the worst-case and best-case response times, it is, then, possible to compute the delay and jitter and investigate if a control application within a given server is guaranteed to be stable.

In addition to the analysis, we also provide analytic results that can drive the design of a server towards solutions which can guarantee the stability of the controller. The aim of such a design procedure is bandwidth minimization. Since such a solution is derived using a linear upper and lower bound of the server supply function, we also evaluate the amount of pessimism introduced by our technique, both theoretically and experimentally.

2 System model

The system is composed of n plants. Each plant is controlled by a control task which is executing within a server. Below we describe the model of the plant, the control task, and the server.

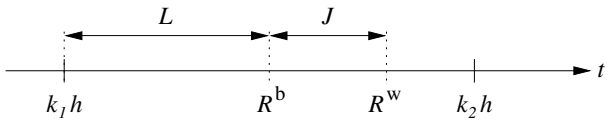


Fig. 2. Graphical interpretation of the nominal delay and worst-case response-time jitter.

2.1 Plant model

A plant is modeled by a continuous-time system of differential equations [1].

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x}, \end{aligned} \tag{1}$$

where $\boldsymbol{x}$, $\boldsymbol{u}$, and $\boldsymbol{y}$ are the plant state, the control signal, and the plant output, respectively. Since each plant is considered in isolation, we do not report the index i of the plant among all the controlled plants.

2.2 Control task model

The plant output $\boldsymbol{y}$ is sampled in a strictly periodic manner with period h.[1] The control signal $\boldsymbol{u}$ is computed by a control task τ. Such a control signal is updated any time the control task completes and is held constant between two consecutive updates.

The instants when the input $\boldsymbol{u}$ is applied to the plant do then depend on the way the task τ is scheduled. The task parameters, which describe the timing behavior of the task are:

- the *best-case execution time*, denoted by c^b;
- the *worst-case execution time*, denoted by c^w; and
- the *sampling period*, denoted by h.

In addition, the way the task is scheduled determines also the following task characteristics, which depend in turn on the above mentioned parameters:

- the *best-case response time* R^b,
- the *worst-case response time* R^w,
- the *nominal delay* (or latency), denoted by $L = R^b$, and
- the *worst-case response-time jitter* (jitter), denoted by $J = R^w - R^b$.

The terminology and the notation are illustrated in Figure 2. The nominal delay, or the latency, captures the constant part of the delay, while the jitter corresponds to the variation in the delay experienced by all instances (jobs) of a task. Note that we do not consider any deadlines for control tasks.

2.3 Server model

As introduced above, to isolate controllers from one another, each control task is bound to execute over a dedicated server. The periodic server S is described by:

- the *server budget* Q;
- the *server period* P, and
- the *server deadline* D.

This model was also called EDP (Explicit Deadline Periodic) model [20]. Every period P the server is activated. Then, it allocates Q amount of time to the task, before the server deadline expires.

The delay and jitter experienced by a task are tightly connected to the best-case and worst-case response times. To compute these two quantities, it is then necessary to determine the worst and best case scenarios with regard to the computational resource supplied by the server.

To perform worst-case analysis for the tasks running within a server, a classic approach [15], [17], [18], [19], [20] is to define the *supply lower bound function* $\mathsf{slbf}(t)$, which is formally defined as follows.

Definition 1: The supply lower bound function $\mathsf{slbf}(t)$ of a server S is the *minimum* amount of resource provided in any interval of length t.

The exact expression of $\mathsf{slbf}(t)$ of a periodic server, is

$$\mathsf{slbf}(t) = \max\{0, kQ, t - P - D + 2Q - k(P - Q)\} \tag{2}$$

with $k = \left\lfloor \frac{t-(D-Q)}{P} \right\rfloor$, and it is depicted in Figure 3(a) by a solid line (please refer to the related literature [15], [17], [18], [19], [20] for details on its computation). As the expression of (2) may be difficult to be managed, especially when the server parameters are the variables subject to optimization (as we do in this paper), it is often convenient to lower bound the $\mathsf{slbf}(t)$ by the *linear supply lower bound function* $\mathsf{lslbf}(t)$, defined as

$$\mathsf{lslbf}(t) = \max\left\{0, \alpha(t - \Delta)\right\}, \tag{3}$$

with, using Feng-Mok's notation [15], the server bandwidth α and delay Δ, defined as

$$\alpha = \frac{Q}{P}, \tag{4}$$

$$\Delta = P + D - 2Q. \tag{5}$$

The lslbf is depicted in Figure 3(a) by a dashed line.

Analogously, for the best-case analysis it is possible to compute the *supply upper bound function* $\mathsf{subf}(t)$, defined as follows.

Definition 2: The supply upper bound function $\mathsf{subf}(t)$ of a server S is the *maximum* amount of resource provided in any interval of length t.

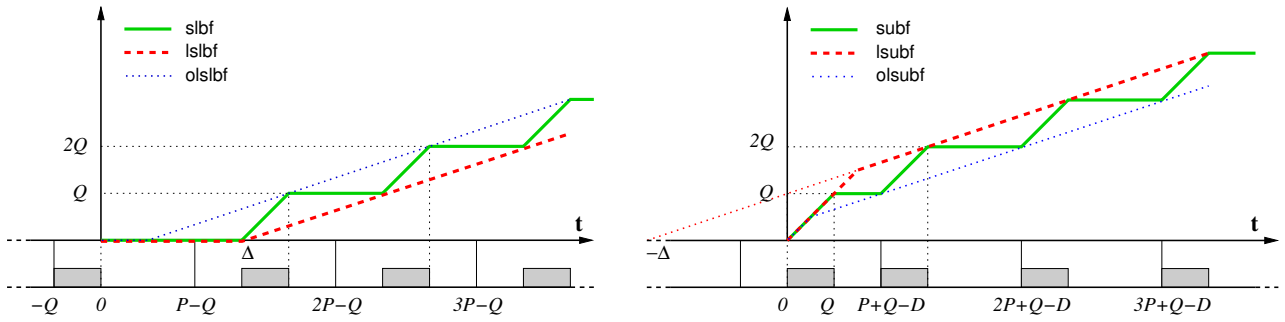
In strict analogy to the worst case examined earlier, the expression of the subf of a periodic server is

$$\mathsf{subf}(t) = \min\{t, kQ, t + P + D - 2Q - k(P - Q)\} \tag{6}$$

with $k = \left\lceil \frac{t+D-Q}{P} \right\rceil$, while the *linear supply upper bound function* is

$$\mathsf{lsubf}(t) = \min\left\{t, \alpha(t + \Delta)\right\} \tag{7}$$

1. This essentially means that there exists a dedicated hardware to sample the output of the plant strictly periodically. The output then is stored in a buffer and the controller reads the output from this buffer upon execution.



(a) Worst-case resource allocation scenario. (b) Best-case resource allocation scenario.

Fig. 3. Worst-case and best-case resource allocation scenarios.

with α and Δ as in (4) and (5), respectively. Figure 3(b) shows the subf (by a solid line) as well as the lsubf (by a dashed line).

3 Server-based analysis of control tasks

In this section, we determine the best-case and worst-case response times of the control task running within a server, as functions of the server parameters P, Q, and D. The analysis is performed with the exact slbf/subf functions of (2) and (6) (in Section 3.1) as well as with the linear bounds lslbf/lsubf of (3) and (7) (Section 3.2).

3.1 Exact characterization

In this section, the exact real-time analysis for a control task is derived. To derive the worst-case response time of a task τ, we must consider the minimum amount of resource time available to the task, which is described by $\mathsf{slbf}(t)$.

The worst-case response time R^{w} of the first job of the control task (released at time 0) is equal to the first instant when the server has necessarily provided at least c^{w} amount of time, that is

$$R^{\mathrm{w}} = \min\{t : \mathsf{slbf}(t) \geq c^{\mathrm{w}}\}. \tag{8}$$

By computing the pseudo-inverse of $\mathsf{slbf}(t)$, such a value can be computed explicitly and it is equal to

$$R^{\mathrm{w}} = D - Q + \left\lceil \frac{c^{\mathrm{w}}}{Q} \right\rceil (P - Q) + c^{\mathrm{w}}. \tag{9}$$

The proof is similar to [28].

Unfortunately, the longest response time may occur even at the later jobs, and not necessarily at the first job. This is the case since, as mentioned before, we do not enforce any task deadline, thus, response times are allowed to be longer than the sampling periods h. Therefore, we must evaluate the response times of all jobs within the busy period, as indicated by Lehoczky [29] for the arbitrary deadline case.

The worst-case response time of the control task within a server $S = (Q, P, D)$ is obtained as follows,

$$R^{\mathrm{w}} = \sup_{q \in \mathbb{N}} \left\{ D - Q + \left\lceil \frac{qc^{\mathrm{w}}}{Q} \right\rceil (P - Q) + qc^{\mathrm{w}} - (q-1)h \right\}. \tag{10}$$

We remind that (for example, see the proof of Lemma 1 in [30]) the supremum of (10) has a finite solution only when

$$\alpha = \frac{Q}{P} \geq \frac{c^{\mathrm{w}}}{h}. \tag{11}$$

In analogy with (8), the best-case response time R^{b} is defined through the subf function as follow

$$R^{\mathrm{b}} = \min\{t : \mathsf{subf}(t) \geq c^{\mathrm{b}}\}, \tag{12}$$

which can also be computed explicitly, and it is equal to

$$R^{\mathrm{b}} = \max\left\{0, 2Q - D - P + \left\lceil \frac{c^{\mathrm{b}}}{Q} \right\rceil (P - Q)\right\} + c^{\mathrm{b}}. \tag{13}$$

The proof is similar to the proof of Theorem 1 in [17].

3.2 Characterization with linear bounds

The main obstacle in using the exact response time for finding the optimal server parameters (see Section 5) is that Equations (10) and (13) involve ceiling functions. Hence, we propose to compute an upper bound $\overline{R}^{\mathrm{w}}$ to the R^{w} and a lower bound $\underline{R}^{\mathrm{b}}$ of R^{b} using, respectively, the lslbf and lsubf functions, rather than the exact ones, i.e., slbf and subf.

Observe that while this approximation involves pessimism, it is safe from the stability point of view.

By replacing the slbf in (8) with the lslbf of (3), we can readily compute the response time upper bound, which is

$$\overline{R}^{\mathrm{w}} = \frac{c^{\mathrm{w}}}{\alpha} + \Delta. \tag{14}$$

As shown in [30], such an upper bound to the response time is finite only if the server bandwidth is not smaller than the worst-case utilization of the control task, that is

$$\alpha = \frac{Q}{P} \geq \frac{c^{\mathrm{w}}}{h}. \tag{15}$$

Similarly, by replacing subf in (12) by lsubf of (7), the lower bound to the best-case response time is given by,

$$\underline{R}^{\mathrm{b}} = \max\left\{c^{\mathrm{b}}, \frac{c^{\mathrm{b}}}{\alpha} - \Delta\right\}. \tag{16}$$

4 Stability Constraint

It is well known that the delay and jitter in the execution of the control applications are decisive factors in the performance and stability of the plants associated with them. This is opposed to hard real-time systems where the systems are design based on the notion of deadline and worst-case response time. For control applications, hence, the worst-case control performance and stability are considered instead of the worst-case response time and deadline.

To quantify the tolerable amount of delay and jitter by a control application before the instability of the plant, or to guarantee a certain degree of performance, we use the Jitter Margin toolbox [26], [5], [2]. It provides sufficient stability conditions for a closed-loop system with a linear continuous-time plant and a linear discrete-time controller.

For a given nominal delay, the Jitter Margin toolbox computes the *jitter margin* (similar to the *phase margin* and *gain margin* concepts) to guarantee the required degree of performance or stability. The Jitter Margin toolbox provides the stability curve that determines the maximum tolerable response-time jitter J based on the nominal delay L. While the curve can instead be generated for a certain control performance, rather than stability, we use the phrase *stability curve* in this paper to refer to the output of the Jitter Margin toolbox. The solid curves in Figure 4 are examples of the stability curves generated by the Jitter Margin toolbox. Observe that the area below the solid curve is the stable area. The graph is generated for the plant with transfer function $\frac{1000}{s^2+s}$ and sampling period of 6 ms. The green curve is generated when a discrete-time Linear-Quadratic-Gaussian (LQG) controller is considered, whereas the blue curve is generated considering a Proportional-Integral-Derivative (PID) controller.

For a given sampling period, the stability curve can safely be approximated by a linear function of the nominal delay and worst-case response-time jitter. The linear approximation is generated by a constrained least-squared optimization on the original curve generated by the Jitter margin toolbox, which is computationally efficient. The linear *stability condition* for a control application is of the form

$$L + aJ \leq b, \tag{17}$$

where $a \geq 1, b \geq 0$. The nominal delay L identifies the constant part of the delay that the control application experiences, whereas the worst-case response-time jitter J captures the varying part of the delay

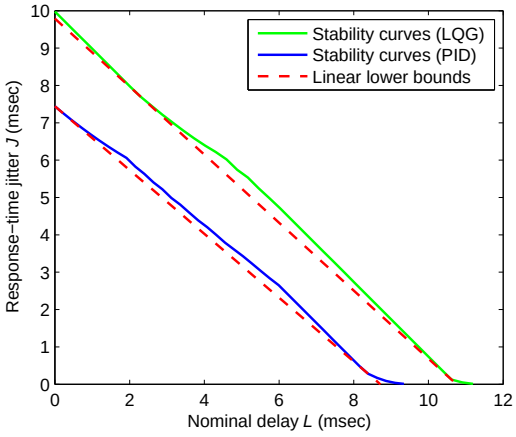


Fig. 4. The stability curves generated by the Jitter Margin toolbox and their linear lower bounds (the area below the curves is the stable area).

(see Figure 2, where R^{b} and R^{w} represent the best-case and worst-case response times, respectively). The linear lower bounds, depicted by the dashed lines, on the original curves generated by the Jitter Margin toolbox are also shown in Figure 4. In [5], Cervin et. al. discussed the fact that $L + J(L)$ is an increasing function of L, where $J(L)$ is the jitter margin for the nominal delay L. We shall show that the coefficient a is indeed always greater than 1. Let us consider two nominal delays L and L', where $L < L'$. Based on this property, we can write the following inequality,

$$L + J(L) < L' + J'(L'), \tag{18}$$

which can be simplified to,

$$a = -\frac{L' - L}{J'(L') - J(L)} > 1. \tag{19}$$

This indicates that control applications are more sensitive to the varying part of the delay than the constant part.

Often, the linear lower bound efficiently captures the stable area identified by Jitter Margin. Although we consider only a single linear function to lower bound the curve generated by the Jitter Margin toolbox in this manuscript, it is also possible to consider a piecewise linear lower bound and perform all optimizations throughout this paper for each linear section and then choose the best solution among all. However, for the sake of presentation, here, we consider that the stability curve can be efficiently bounded from below by a single linear function.

In order to apply the stability analysis discussed, the values of the nominal delay (L) and worst-case response-time jitter (J) of the control task should be computed. The two metrics are defined based on the worst-case and best-case response times as follows,

$$\begin{aligned} L &= R^{\mathrm{b}}, \\ J &= R^{\mathrm{w}} - R^{\mathrm{b}}, \end{aligned} \tag{20}$$

where R^{w} and R^{b} denote the worst-case and best-case response times, respectively. The stability constraint, hence, can be formulated as,

$$\begin{aligned} L + aJ &\leq b, \\ R^{\mathrm{b}} + a(R^{\mathrm{w}} - R^{\mathrm{b}}) &\leq b. \end{aligned} \tag{21}$$

For a given server, the stability condition (21), which is based on the exact best-case and worst-case response times, determines if the server, in the worst-case, can guarantee the stability of the control task associated with it (analysis problem).

In the context of the optimization problem, as will be discussed in Section 5, however, the presence of discontinuous operators (ceiling) in the exact expressions (10) and (13) of the worst-case and best-case response times makes them unsuitable. Hence, we use the upper/lower bound of the worst/best-case response times and redefine the nominal delay and the worst-case response-time jitter as follows,

$$\begin{aligned} \underline{L} &= \underline{R}^{\mathrm{b}}, \\ \overline{J} &= \overline{R}^{\mathrm{w}} - \underline{R}^{\mathrm{b}}. \end{aligned} \tag{22}$$

While using the linear supply bounds involves some pessimism compared to the original supply bounds, it is safe from the stability point of view [5]. Nonetheless, the amount of introduced pessimism is discussed in Section 6.

The stability constraint based on the linear bounds is given in the following,

$$\begin{aligned} b &\geq \underline{L} + a\overline{J}, \\ b &\geq \underline{R}^{\mathrm{b}} + a(\overline{R}^{\mathrm{w}} - \underline{R}^{\mathrm{b}}), \\ b &\geq a\left(\frac{c^{\mathrm{w}}}{\alpha} + \Delta\right) - (a-1)\max\left\{c^{\mathrm{b}}, \frac{c^{\mathrm{b}}}{\alpha} - \Delta\right\}, \\ &= a\left(\frac{c^{\mathrm{w}}}{\alpha} + \Delta\right) + (a-1)\min\left\{-c^{\mathrm{b}}, -\left(\frac{c^{\mathrm{b}}}{\alpha} - \Delta\right)\right\}, \end{aligned}$$

which we rewrite as

$$\min\left\{\frac{a(c^{\mathrm{w}} - c^{\mathrm{b}}) + c^{\mathrm{b}}}{\alpha} + (2a-1)\Delta - b, \; \frac{ac^{\mathrm{w}}}{\alpha} + a\Delta - (a-1)c^{\mathrm{b}} - b\right\} \leq 0. \tag{23}$$

Hence, Equation (23) describes the constraint on the server parameters (the bandwidth α and the delay Δ, see Section 2.3), which guarantees the stability of the controller running within such a server.

5 Optimal Design of Stabilizing Servers

In this section, we describe the procedure to design optimal stabilizing servers. The objective of the optimization is to minimize the utilization required in order to guarantee the stability of all control applications, that is

$$U = \sum_{i=1}^{n}\left(\alpha_i + \frac{\epsilon}{P_i}\right), \tag{24}$$

where ϵ denotes the switching overhead for the server and is considered to be strictly positive. If no overhead is considered, then the solution would be with $P \to 0$, making this an impractical server period.

We propose the implicit deadline server design, in which all server deadlines are set equal to the periods, $D_i = P_i$.

Thanks to the isolation provided by the resource allocation mechanism, the stability of each control task is guaranteed through the parameters (α and Δ) of the server running the task only (Equation (23)). Hence, the minimization of the total server utilization of (24) can be broken down into one bandwidth minimization problem for each server, rather than a more complex minimization which involves all task parameters together.

If we assume $D = P$ for all servers, we can perform the following optimization for each control application and conclude based on the obtained results,

$$\begin{aligned} \min_{\alpha,\Delta} \quad & \alpha + \frac{2\epsilon(1-\alpha)}{\Delta} \\ \text{s.t.} \quad & \min\left\{\frac{a(c^{\mathrm{w}} - c^{\mathrm{b}}) + c^{\mathrm{b}}}{\alpha} + (2a-1)\Delta - b, \; \frac{ac^{\mathrm{w}}}{\alpha} + a\Delta - (a-1)c^{\mathrm{b}} - b\right\} \leq 0. \end{aligned} \tag{25}$$

Notice that in the above cost the period P is replaced by $\frac{\Delta}{2(1-\alpha)}$, as it follows from (4)–(5) for $D = P$.

The solution to the above problem is the minimum bandwidth (including the overhead) required to guarantee stability of control task τ.

Let us proceed with finding the global optimum of the problem (25), which is concerned with a single control task in isolation. The stability constraint in (25) can be written as

$$\min\{g_{\mathrm{I}}(\alpha,\Delta), g_{\mathrm{II}}(\alpha,\Delta)\} \leq 0,$$

which is equivalent to

$$(g_{\mathrm{I}}(\alpha,\Delta) \leq 0) \quad \vee \quad (g_{\mathrm{II}}(\alpha,\Delta) \leq 0),$$

with $\vee$ denoting the *logical or* between two propositions. Thus, the problem (25) can be solved by solving individually the following two problems

$$\begin{aligned} \min_{\alpha,\Delta} \quad & \alpha + \frac{2\epsilon(1-\alpha)}{\Delta} \\ \text{s.t.} \quad & \frac{a(c^{\mathrm{w}} - c^{\mathrm{b}}) + c^{\mathrm{b}}}{\alpha} + (2a-1)\Delta - b \leq 0, \end{aligned} \tag{26}$$

and,

$$\begin{aligned} \min_{\alpha,\Delta} \quad & \alpha + \frac{2\epsilon(1-\alpha)}{\Delta} \\ \text{s.t.} \quad & \frac{ac^{\mathrm{w}}}{\alpha} + a\Delta - (a-1)c^{\mathrm{b}} - b \leq 0. \end{aligned} \tag{27}$$

and then select the best solution between the two produced by (26) and (27). Moreover, in order for the response time to be finite, the server bandwidth α has to satisfy $\alpha \geq \frac{c^{\mathrm{w}}}{h}$, which leads to an additional constraint in each of the problems (26) and (27).

To solve problems (26) and (27), we use the KKT (Karush-Kuhn-Tucker) necessary conditions for optimality [31]. According to the KKT condition, the optimum $\boldsymbol{x}^*$ of the problem

$$\begin{aligned} &\min_{\boldsymbol{x}} \ f(\boldsymbol{x}) \\ &\text{s.t.} \quad g_i(\boldsymbol{x}) \leq 0, \qquad i = 1 \dots m, \end{aligned} \tag{28}$$

must necessarily satisfy the following conditions

$$\begin{aligned} \nabla f(\boldsymbol{x}^*) + \sum_{i=1}^{m} \mu_i^* \nabla g_i(\boldsymbol{x}^*) &= \boldsymbol{0}, \\ \mu_i^* g_i(\boldsymbol{x}^*) &= 0, \qquad i = 1 \dots m, \\ \mu_i^* &\geq 0, \qquad i = 1 \dots m. \end{aligned} \tag{29}$$

For the case of our problem, it is assumed that $g_1(\boldsymbol{x})$ is associated with the stability constraints shown in problems (26) and (27), whereas $g_2(\boldsymbol{x})$ is associated with inequality (11).

Let us proceed with solving problem (26). From the KKT condition of the gradient, if we differentiate w.r.t. α and then Δ, we find

$$1 - \frac{2\epsilon}{\Delta} - \mu_1 \frac{a(c^{\mathrm{w}} - c^{\mathrm{b}}) + c^{\mathrm{b}}}{\alpha^2} - \mu_2 = 0 \tag{30}$$

$$-\frac{2\epsilon(1-\alpha)}{\Delta^2} + \mu_1(2a-1) = 0 \tag{31}$$

We consider two cases: $\mu_2 = 0$ and $\mu_2 > 0$.

$\boldsymbol{\mu_2 = 0}$: Let us first assume there is no constraint on the server bandwidth α, i.e., $\mu_2 = 0$. Since $a \geq 1$ and $\alpha < 1$, from (31), we immediately find the multiplier μ_1, that is:

$$\mu_1 = \frac{2\epsilon(1-\alpha)}{\Delta^2(2a-1)} > 0,$$

hence the constraint of (26) is active and must hold with the equal sign.

If we set, to have a more compact notation,

$$x_{\mathrm{I}} = a(c^{\mathrm{w}} - c^{\mathrm{b}}) + c^{\mathrm{b}}, \quad y_{\mathrm{I}} = \epsilon(2a-1), \quad z_{\mathrm{I}} = b, \tag{32}$$

then the equality constraint of (26) can be rewritten as

$$\frac{x_{\mathrm{I}}}{\alpha} + y_{\mathrm{I}} \frac{\Delta}{\epsilon} = z_{\mathrm{I}}, \tag{33}$$

from which we find

$$\frac{\Delta}{\epsilon} = \frac{\alpha z_{\mathrm{I}} - x_{\mathrm{I}}}{\alpha y_{\mathrm{I}}}, \tag{34}$$

and then the multiplier μ_1 is

$$\mu_1 = \frac{2(1-\alpha)}{y_{\mathrm{I}}} \left(\frac{\alpha y_{\mathrm{I}}}{\alpha z_{\mathrm{I}} - x_{\mathrm{I}}} \right)^2. \tag{35}$$

By replacing (34) and (35) in the condition (30), we find:

$$\begin{aligned} &1 - 2\frac{\alpha y_{\mathrm{I}}}{\alpha z_{\mathrm{I}} - x_{\mathrm{I}}} - \frac{2(1-\alpha)}{y_{\mathrm{I}}} \frac{\alpha^2 y_{\mathrm{I}}^2}{(\alpha z_{\mathrm{I}} - x_{\mathrm{I}})^2} \frac{x_{\mathrm{I}}}{\alpha^2} = 0 \\ &z_{\mathrm{I}}(z_{\mathrm{I}} - 2y_{\mathrm{I}})\alpha^2 - 2x_{\mathrm{I}}(z_{\mathrm{I}} - 2y_{\mathrm{I}})\alpha + x_{\mathrm{I}}(x_{\mathrm{I}} - 2y_{\mathrm{I}}) = 0 \\ &\alpha^2 - 2\frac{x_{\mathrm{I}}}{z_{\mathrm{I}}}\alpha + \frac{x_{\mathrm{I}}(x_{\mathrm{I}} - 2y_{\mathrm{I}})}{z_{\mathrm{I}}(z_{\mathrm{I}} - 2y_{\mathrm{I}})} = 0 \\ &\alpha = \alpha_{\mathrm{I}}\,(1 \pm \delta_{\mathrm{I}}) \end{aligned}$$

where we set

$$\alpha_\ell = \frac{x_\ell}{z_\ell}, \quad \delta_\ell = \sqrt{1 - \frac{z_\ell(x_\ell - 2y_\ell)}{x_\ell(z_\ell - 2y_\ell)}} \tag{36}$$

with $\ell = \mathrm{I}$. The values α_{I} and δ_{I} represent, respectively, the consumed bandwidth in absence of overhead and the increase of bandwidth needed due to overhead. Among the two solutions, the smaller one makes the corresponding value of Δ negative. Hence, the only acceptable solution for the server bandwidth is given by $\alpha_{\mathrm{I}}(1+\delta_{\mathrm{I}})$ and there is no need to check the second-order sufficient conditions.

The solution identified here corresponds to the case where there is no constraint on server bandwidth α (i.e., $\mu_2 = 0$). Therefore, if this solution satisfies constraint (11), i.e., $\alpha \geq \frac{c^{\mathrm{w}}}{h}$, then there is no need to consider the case where $\mu_2 > 0$ (because the solution in the larger search space is valid even considering this constraint); otherwise, this case has to be taken into account.

$\boldsymbol{\mu_2 > 0}$: Now let us consider the case where the solution found does not satisfy constraint (11), i.e., $\alpha < \frac{c^{\mathrm{w}}}{h}$. From Equation (30), we have,

$$\mu_2 = 1 - \frac{2\epsilon}{\Delta} - \mu_1 \frac{a(c^{\mathrm{w}} - c^{\mathrm{b}}) + c^{\mathrm{b}}}{\alpha^2}. \tag{37}$$

Substituting μ_2 in the second equality of (29), i.e., $\mu_2 g_2(\boldsymbol{x}) = \mu_2 \left(-\alpha + \frac{c^{\mathrm{w}}}{h}\right) = 0$, we obtain,

$$\underbrace{\left(1 - \frac{2\epsilon}{\Delta} - \mu_1 \frac{a(c^{\mathrm{w}} - c^{\mathrm{b}}) + c^{\mathrm{b}}}{\alpha^2}\right)}_{\text{first term}} \underbrace{\left(-\alpha + \frac{c^{\mathrm{w}}}{h}\right)}_{\text{second term}} = 0. \tag{38}$$

The above equation has at most three solutions. The solution found so far is equivalent to considering the first term to be equal to zero, i.e., $\mu_2 = 0$ (see Equation (37)). As discussed before, the solutions obtained considering the first term are invalid since they do not satisfy constraint (11). Therefore, the only valid solution is $\frac{c^{\mathrm{w}}}{h}$. In other words, the optimal solution α_{I}^* is equal to $\alpha_{\mathrm{I}}(1+\delta_{\mathrm{I}})$ except when this solution does not satisfy constraint (11). The final solution is then given by,

$$\alpha_{\mathrm{I}}^* = \max\left\{ \alpha_{\mathrm{I}}(1+\delta_{\mathrm{I}}), \frac{c^{\mathrm{w}}}{h} \right\}, \tag{39}$$

in which we also account for the constraint (11). The corresponding optimal value of the server delay Δ_{I}^* can be computed from (34).

To solve the second problem (27), we simply observe that by setting

$$x_{\mathrm{II}} = ac^{\mathrm{w}}, \quad y_{\mathrm{II}} = a\epsilon, \quad z_{\mathrm{II}} = b + (a-1)c^{\mathrm{b}}. \tag{40}$$

the constraint can be rewritten as in (33) by replacing x_{I}, y_{I}, and z_{I}, with x_{II}, y_{II}, and z_{II} of (40). Since the cost functions and the constraints of the two problems are the same, it follows that the solution is exactly the same as (39), with the corresponding replacements.

Since the two problems have to be considered in logical or, the minimal bandwidth α^* and delay Δ^* which can guarantee the stability of the control task (within the assumption of server deadline D equal to the server period P) is given by the better solution of the two problems, i.e.,

$$\min\left\{\alpha_{\text{I}}^* + \frac{2\epsilon(1-\alpha_{\text{I}}^*)}{\Delta_{\text{I}}^*}, \alpha_{\text{II}}^* + \frac{2\epsilon(1-\alpha_{\text{II}}^*)}{\Delta_{\text{II}}^*}\right\}. \quad (41)$$

After performing the above procedure for all servers and having found the minimum resource utilization required for stability of all control applications, we should now check if the resource demand is less than or equal to the resource supply. In the case of the implicit deadline servers running on a uniprocessor, the solution found is valid if and only if the utilization is less than or equal to one, i.e.,

$$\sum_{i=1}^{n}\left(\alpha_i^* + \frac{2\epsilon(1-\alpha_i^*)}{\Delta_i^*}\right) \leq 1. \quad (42)$$

6 Theoretical Guarantees

We shall now discuss the degree of pessimism introduced in the proposed approach by using the linear bounds instead of the exact response times. Towards this, we need to define the notion of optimistic supply function. Note that the optimistic supply functions are *unsafe* and are only used to quantify the amount of pessimism introduced in our approach.

The *optimistic supply lower bound function* $\mathsf{olslbf}(t)$ of a server is a linear upper bound on the supply lower bound function (as shown by the dotted line in Figure 3(a)). To obtain this, we notice that the optimistic supply functions are only $\frac{Q}{P}(P-Q)$ different from the linear supply bound functions, as it is shown in Figure 5. Since we have the linear supply lower bound function $\mathsf{lslbf}(t)$, the optimistic supply lower bound function is obtained as follows,

$$\begin{aligned}\mathsf{olslbf}(t) &= \max\left\{0, \frac{Q}{P}(t-(P+D-2Q)) + \frac{Q}{P}(P-Q)\right\}\\ &= \max\left\{0, \frac{Q}{P}(t-(D-Q))\right\}.\end{aligned}$$

Similarly, we define the *optimistic supply upper bound function* $\mathsf{olsubf}(t)$ as follows (shown by the dotted line in Figure 3(b)),

$$\begin{aligned}\mathsf{olsubf}(t) &= \min\left\{t, \frac{Q}{P}(t+(P+D-2Q)) - \frac{Q}{P}(P-Q)\right\}\\ &= \min\left\{t, \frac{Q}{P}(t+(D-Q))\right\}.\end{aligned}$$

Computing the pseudo-inverse of these optimistic supply functions, we obtain optimistic bounds for the best-case and worst-case response times,

$$\begin{aligned}\underline{R}^{\text{w}} &= \frac{c^{\text{w}}}{\alpha} + \underline{\Delta},\\ \overline{R}^{\text{b}} &= \max\left\{c^{\text{b}}, \frac{c^{\text{b}}}{\alpha} - \underline{\Delta}\right\},\end{aligned} \quad (43)$$

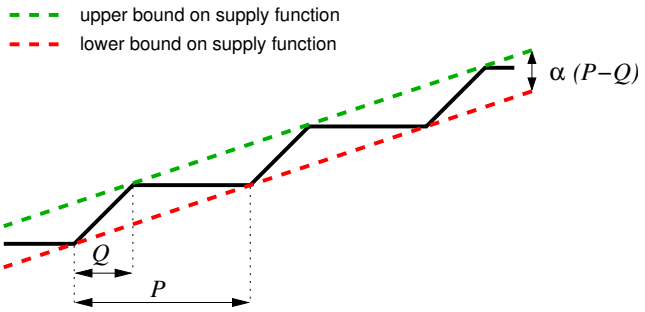


Fig. 5. The relation between linear, optimistic, and exact supply functions.

where $\underline{\Delta} = D - Q$.

The next two subsections discuss the theoretical results on the amount of pessimism involved in our design method. We shall first restrict out attention to the stability of one single controller. Then, we focus on both stability and schedulability of the set of all servers.

6.1 Stability of controllers

In this subsection, we shall focus on the stability of a single controller. The following lemma clarifies the relation between using the optimistic and exact supply functions for stability guarantees.

Lemma 1: If the stability constraint (23) of a control task is satisfied within a server $S = (Q, P, D)$ with the exact supply functions, it is also satisfied within the same server but considering the optimistic supply functions.

Proof: From the definition of the optimistic supply functions,

$$\begin{aligned}\forall t, \quad &\mathsf{subf}(t) \geq \mathsf{olsubf}(t),\\ \forall t, \quad &\mathsf{slbf}(t) \leq \mathsf{olslbf}(t).\end{aligned}$$

This, in turn, leads to the following inequalities among the exact and optimistic response times,

$$\begin{aligned}R^{\text{b}} &= \min\{t : \mathsf{subf}(t) \geq c^{\text{b}}\} \leq \min\{t : \mathsf{olsubf}(t) \geq c^{\text{b}}\} = \overline{R}^{\text{b}},\\ R^{\text{w}} &= \min\{t : \mathsf{slbf}(t) \geq c^{\text{w}}\} \geq \min\{t : \mathsf{olslbf}(t) \geq c^{\text{w}}\} = \underline{R}^{\text{w}}.\end{aligned}$$

This indicates that considering the optimistic supply functions results in a lower bound for the worst-case response time $\underline{R}^{\text{w}}$ and an upper bound for the best-case response time $\overline{R}^{\text{b}}$. Note that since $a \geq 1$, $\underline{R}^{\text{w}} \leq R^{\text{w}}$, $\overline{R}^{\text{b}} \geq R^{\text{b}}$, we have,

$$L + aJ = aR^{\text{w}} + (1-a)R^{\text{b}} \geq a\underline{R}^{\text{w}} + (1-a)\overline{R}^{\text{b}} = \overline{L} + a\underline{J},$$

where $\overline{L} = \overline{R}^{\text{b}}$ and $\underline{J} = \underline{R}^{\text{w}} - \overline{R}^{\text{b}}$. Hence, if there exists a server with the exact supply functions that can satisfy inequality (21), then the stability constraint (21) is also satisfied considering the optimistic linear supply functions, i.e.,

$$L + aJ \leq b \quad \stackrel{L+aJ \geq \overline{L}+a\underline{J}}{\Longrightarrow} \quad \overline{L} + a\underline{J} \leq b.$$

□

Lemma 2: If the stability constraint (23) of a control task is satisfied within a server $S = (Q, P, D)$ with the linear supply functions, it is also satisfied within the same server but considering the exact supply functions.

Proof: The following relations hold for the response times,

$$R^{\mathrm{b}} = \min\{t : \mathsf{subf}(t) \geq c^{\mathrm{b}}\} \geq \min\{t : \mathsf{lsubf}(t) \geq c^{\mathrm{b}}\} = \underline{R}^{\mathrm{b}},$$
$$R^{\mathrm{w}} = \min\{t : \mathsf{slbf}(t) \geq c^{\mathrm{w}}\} \leq \min\{t : \mathsf{lslbf}(t) \geq c^{\mathrm{w}}\} = \overline{R}^{\mathrm{w}}.$$

Since $a \geq 1$, we have the following inequalities,

$$aR^{\mathrm{w}} + (1-a)R^{\mathrm{b}} \leq a\overline{R}^{\mathrm{w}} + (1-a)\underline{R}^{\mathrm{b}},$$
$$L + aJ \leq \underline{L} + a\overline{J},$$

from which the theorem follows,

$$\underline{L} + a\overline{J} \leq b \quad \stackrel{L+aJ \leq \underline{L}+a\overline{J}}{\Longrightarrow} \quad L + aJ \leq b.$$

□

The next theorem establishes one important relation between stabilizing controllers based on the linear supply functions and the exact supply functions.

Theorem 1: If the stability constraint (23) of a control task is satisfied within a server $S = (Q, P, D)$ with the exact supply functions, it is also satisfied within a server $S' = (\frac{Q}{k}, \frac{P}{k}, \frac{D}{k})$ with the linear supply functions and $k \geq 1 + \frac{P-Q}{D-Q}$.

Proof: Let us first prove the following inequalities,

$$\begin{aligned} &\forall t, \quad \mathsf{subf}(t) \geq \mathsf{lsubf}'(t), \\ &\forall t, \quad \mathsf{slbf}(t) \leq \mathsf{lslbf}'(t). \end{aligned} \tag{44}$$

To prove $\mathsf{subf}(t) \geq \mathsf{lsubf}'(t)$, we derive the optimistic linear lower bound $\mathsf{olsubf}(t)$ on the exact supply upper bound function $\mathsf{subf}(t)$. If we can prove that this optimistic linear lower bound $\mathsf{olsubf}(t)$ is always greater than or equal to $\mathsf{lsubf}'(t)$, considering that it is a lower bound of $\mathsf{subf}(t)$, we have $\mathsf{subf}(t) \geq \mathsf{lsubf}'(t)$. The linear lower bound on $\mathsf{subf}(t)$ is given by (according to the definition of the optimistic supply upper bound functions),

$$\mathsf{olsubf}(t) = \min\left\{t, \frac{Q}{P}(t + (D-Q))\right\}.$$

Let us also derive the $\mathsf{lsubf}'(t)$ for the implicit deadline server $S' = (\frac{Q}{k}, \frac{P}{k}, \frac{D}{k})$,

$$\mathsf{lsubf}'(t) = \min\left\{t, \frac{\frac{Q}{k}}{\frac{P}{k}}(t + \frac{1}{k}(P + D - 2Q))\right\}.$$

Our goal is to find k such that the optimistic linear lower bound on the exact supply upper bound function $\mathsf{olsubf}(t)$ is always above the linear upper bound $\mathsf{lsubf}'(t)$. Note that, if $x \geq y$ and $x' \geq y'$, then $\min\{x, x'\} \geq \min\{y, y'\}$. Since $t \geq t$, we only need to focus on the second terms inside the $\min$-functions in $\mathsf{olsubf}(t)$ and $\mathsf{lsubf}'(t)$, i.e.,

$$\frac{Q}{P}(t + (D-Q)) \geq \frac{\frac{Q}{k}}{\frac{P}{k}}(t + \frac{1}{k}(P + D - 2Q)).$$

Re-arranging the terms, we obtain,

$$k \geq 1 + \frac{P-Q}{D-Q}.$$

Analogously, to prove $\mathsf{slbf}(t) \leq \mathsf{lslbf}'(t)$, we start by showing that the optimistic linear upper bound on $\mathsf{slbf}(t)$ is given by (according to the definition of the optimistic supply lower bound functions),

$$\mathsf{olslbf}(t) = \max\left\{0, \frac{Q}{P}(t - (D-Q))\right\}.$$

The linear lower bound on server S' is given by,

$$\mathsf{lslbf}'(t) = \max\left\{0, \frac{\frac{Q}{k}}{\frac{P}{k}}(t - \frac{1}{k}(P + D - 2Q))\right\}.$$

Observe that, if $x \leq y$ and $x' \leq y'$, then $\max\{x, x'\} \leq \max\{y, y'\}$. Since $0 \leq 0$, it is enough to focus on the second terms inside the $\max$-functions. We would like to find k such that the optimistic linear upper bound on $\mathsf{slbf}(t)$, denoted by $\mathsf{olslbf}(t)$, is always below $\mathsf{lslbf}'(t)$, i.e.,

$$\frac{Q}{P}(t - (D-Q)) \leq \frac{\frac{Q}{k}}{\frac{P}{k}}(t - \frac{1}{k}(P + D - 2Q)),$$

which simplifies to,

$$k \geq 1 + \frac{P-Q}{D-Q},$$

which is the same as the previous condition on k.

As a result of the inequalities in (44), the following relations hold for the response times,

$$R^{\mathrm{b}} = \min\{t : \mathsf{subf}(t) \geq c^{\mathrm{b}}\} \leq \min\{t : \mathsf{lsubf}'(t) \geq c^{\mathrm{b}}\} = \underline{R}^{\mathrm{b}'},$$
$$R^{\mathrm{w}} = \min\{t : \mathsf{slbf}(t) \geq c^{\mathrm{w}}\} \geq \min\{t : \mathsf{lslbf}'(t) \geq c^{\mathrm{w}}\} = \overline{R}^{\mathrm{w}'}.$$

Since $a \geq 1$, we have the following inequalities,

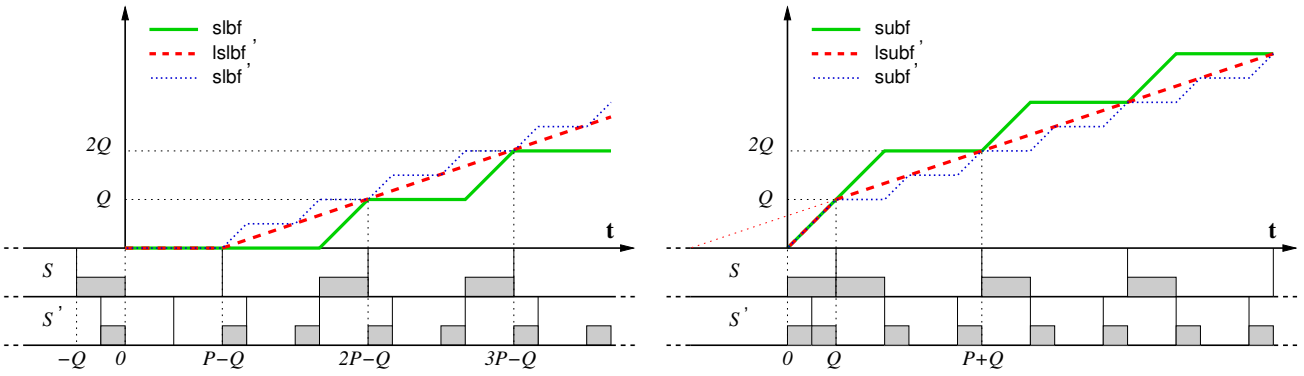
$$aR^{\mathrm{w}} + (1-a)R^{\mathrm{b}} \geq a\overline{R}^{\mathrm{w}'} + (1-a)\underline{R}^{\mathrm{b}'},$$
$$L + aJ \geq \underline{L}' + a\overline{J}',$$

from which the theorem follows,

$$L + aJ \leq b \quad \stackrel{L+aJ \geq \underline{L}'+a\overline{J}'}{\Longrightarrow} \quad \underline{L}' + a\overline{J}' \leq b.$$

Note that the bound is tight when $k = 1 + \frac{P-Q}{D-Q}$ since the linear lower bound on $\mathsf{subf}(t)$ is the same as $\mathsf{lsubf}'(t)$ and the linear upper bound on $\mathsf{slbf}(t)$ is the same as $\mathsf{lslbf}'(t)$. The tightness is in the sense that, for server $S' = (\frac{Q}{k}, \frac{P}{k}, \frac{D}{k})$, decreasing k by a small positive value, violates the inequalities in (44). □

The important message of Theorem 1 is that, if a server $S = (Q, P, D)$ (with the exact supply functions) with bandwidth $\alpha = \frac{Q}{P}$ is identified that satisfies the



(a) Worst-case resource allocation scenario (implicit deadline). (b) Best-case resource allocation scenario (implicit deadline).

Fig. 6. Worst-case and best-case resource allocation scenarios for implicit deadline server.

stability constraint (21) for the control task associated with it, then there exists a server $S' = (\frac{Q}{k}, \frac{P}{k}, \frac{D}{k})$ (with the linear supply functions) that can also satisfy the stability constraint (21) for the control task *and the required bandwidth is the same*, i.e., $\alpha' = \frac{\frac{Q}{k}}{\frac{P}{k}} = \frac{Q}{P}$.

The theorem also states that, *in the worst-case*, the server S' has to be run k times more frequently compared to S. In practice, of course, this might be a disadvantage, if the context-switch overhead is significant.

The following corollary discusses the particular case of implicit deadline servers (i.e., $D = P$) and it will be used in the next section.

Corollary 1: If the stability constraint (23) of a control task is satisfied within an implicit deadline server $S = (Q, P)$ with the exact supply functions, it is also satisfied within an implicit deadline server $S' = (\frac{Q}{2}, \frac{P}{2})$ with the linear supply function.

Proof: The proof follows by substituting $D = P$ in $k \geq 1 + \frac{P-Q}{D-Q}$ in Theorem 1, i.e., $k \geq 2$. □

For clarification see Figure 6. In Figure 6(a), the linear supply lower bound function lslbf' is both a lower bound for the exact supply lower bound function slbf' and the optimistic upper bound on slbf, i.e., olslbf. This implies that the amount of resource provided by slbf' in the worst case is more than or equal to the amount of resource provided by slbf in the worst case. Similarly, Figure 6(b) shows that the linear supply upper bound function lsubf' is both an upper bound for the exact supply upper bound function subf' and the optimistic lower bound of subf, olsubf.

The bound is tight in the sense that the linear supply lower bound functions lslbf' is not only the tightest linear lower bound on the exact supply lower bound function slbf', but also the tightest linear upper bound on the exact supply lower bound function slbf. Similar results can be derived for the linear supply upper bound function lsubf'.

6.2 Schedulability of servers

Thus far in this section, we have limited our attention only to the stability of a single server. However, for a system to be *implementable*, not only should the stability constraint be satisfied, but also the system should be schedulable. The next theorem provides an important analytical result to bound the pessimism involved in using the linear supply functions.

Note that, in many situations, the monotonicity property does not hold [32]. The following Lemma discusses the monotonicity property with respect to processor speed.

Lemma 3: If the stability constraint (23) is guaranteed for a task running within a server $S = (Q, P, D)$ on a processor and considering the linear supply functions, it is also guaranteed within the same server and on a higher speed processor.

Proof: It can be shown that if stability constraint (23) is satisfied, then, on a processor with speed augmented by a factor λ ($\lambda \geq 1$), the following stability constraint is also satisfied,

$$\min \left\{ \frac{a(c^{\mathrm{w}} - c^{\mathrm{b}}) + c^{\mathrm{b}}}{\lambda\alpha} + (2a-1)\Delta - b, \right. \\ \left. \frac{1}{\lambda}\left(\frac{ac^{\mathrm{w}}}{\alpha} - (a-1)c^{\mathrm{b}}\right) + a\Delta - b \right\} \leq 0. \quad (45)$$

The above inequality holds since the terms multiplied by the factor $\frac{1}{\lambda}$ are non-negative considering $\alpha \leq 1$ and $a \geq 1$.

Since the linear stability constraint is satisfied on a processor which is λ times faster, so is the exact stability condition (according to Lemma 2). □

Theorem 2: If a set of controllers is guaranteed to be implemented (i.e., the system is schedulable and all the plants are guaranteed to be stable) using implicit deadline servers $S_i = (Q_i, P_i)$ and considering the exact supply functions over a unit-speed processor, then the same set is guaranteed to be implemented using implicit deadline servers $S'_i = (\frac{Q_i}{2}, \frac{P_i}{2})$ considering the linear supply functions over a λ-speed processor with $\lambda = \sum_{i=1}^{n} \frac{Q_i + 2\epsilon}{P_i}$, ϵ being the switching overhead.

Proof: According to Corollary 1, if the stability constraint of a control application can be satisfied within an implicit deadline server $S_i = (Q_i, P_i)$ with the exact supply functions, then it is also satisfied within an implicit deadline server $S'_i = (\frac{Q_i}{2}, \frac{P_i}{2})$ with the linear supply functions. Therefore, server S'_i with the linear supply functions provides guarantees from the stability point of view. However, in addition to stability, the schedulability of the controllers should also be investigated.

Note that, considering the exact supply functions, the system is schedulable if and only if,

$$U = \sum_{i=1}^{n} \left(\alpha_i + \frac{\epsilon}{P_i} \right) = \sum_{i=1}^{n} \left(\frac{Q_i}{P_i} + \frac{\epsilon}{P_i} \right) \leq 1. \quad (46)$$

Now, let us consider the case where the linear supply functions are used,

$$U' = \sum_{i=1}^{n} \left(\alpha_i + \frac{\epsilon}{\frac{P_i}{2}} \right) = U + \sum_{i=1}^{n} \frac{\epsilon}{P_i}. \quad (47)$$

The system based on the linear supply functions is schedulable if,

$$\lambda \geq U + \sum_{i=1}^{n} \frac{\epsilon}{P_i}, \quad (48)$$

where λ is the relative speed of the processor. Having considered a processor which is λ times faster, it is now required to discuss the impact of this choice on stability. This is addressed by Lemma 3, which states that the stability guarantees (with regard to stability constraint (23)) are preserved on faster processors. It can also be shown that if constraint (11) is satisfied on a processor (i.e., $\frac{c^{\mathrm{w}}}{h} \leq \alpha$), then it is also satisfied on a processor which is faster (i.e., $\frac{1}{\lambda}\frac{c^{\mathrm{w}}}{h} \leq \frac{c^{\mathrm{w}}}{h} \leq \alpha$).

This result indicates that: if the stability of the controllers cannot be guaranteed on a processor with speed $\lambda = U + \sum_{i=1}^{n} \frac{\epsilon}{P_i} = \sum_{i=1}^{n} \frac{Q_i + 2\epsilon}{P_i}$ considering the linear supply functions, then it for sure cannot be guaranteed considering the exact supply functions, since it implies $U > 1$. □

Corollary 2: The factor λ in Theorem 2 is bounded from above by 2.

Proof: Let us assume the system is implementable considering the exact supply functions. This, in turn, implies that the utilization U is

$$U = \sum_{i=1}^{n} \left(\alpha_i + \frac{\epsilon}{P_i} \right) \leq 1.$$

Since $\alpha_i \geq 0$, we obtain

$$\sum_{i=1}^{n} \frac{\epsilon}{P_i} = U - \sum_{i=1}^{n} \alpha_i \leq U.$$

Observe that λ is given by,

$$\lambda = U + \sum_{i=1}^{n} \frac{\epsilon}{P_i} \leq 2 \cdot U.$$

Since $U \leq 1$ in an implementable system, the bound on λ follows: $\lambda \leq 2$. □

As discussed before, the optimal values of server parameters cannot be obtained efficiently, when the exact supply functions are used. Therefore, we consider the notion of optimistic supply functions, for which the server parameters may be computed efficiently (see Section 7).

7 Asymptotic Analysis

In this section, we shall identify a lower bound on the minimum achievable utilization by the approach discussed in the previous sections, i.e., the implicit deadline servers. We will use this bound in our experiments (Section 8.2) in order to evaluate the efficiency of our optimization technique discussed in Section 5. To obtain a tight lower bound on the minimum utilization required for guaranteeing stability of a plant associated with an implicit deadline server, optimistic linear supply functions are considered in this section.

Let us consider the optimistic upper bound on the $\mathsf{slbf}(t)$, denoted by $\mathsf{olslbf}(t)$, and the optimistic lower bound on $\mathsf{subf}(t)$, denoted by $\mathsf{olsubf}(t)$ (see Section 6). Lemma 1 states that if there exists a server with the exact supply functions that can satisfy inequality (21), then it is also possible to find a solution using the optimistic linear supply functions that satisfies inequality (21).

Forming the stability constraint based on the optimistic response times, defined in Equation (43), we realize that the stability constraint (23) remains exactly the same, but assuming $\underline{\Delta}$ instead of Δ. Now let us focus on the special case of implicit deadline server ($D = P$) that leads to $\frac{\epsilon}{P} = \frac{\epsilon(1-\alpha)}{\underline{\Delta}}$. The optimization problem then will be as follows,

$$\begin{aligned} \min_{\alpha, \underline{\Delta}} \quad & \alpha + \frac{\epsilon(1-\alpha)}{\underline{\Delta}} \\ \text{s.t.} \quad & \min \left\{ \frac{a(c^{\mathrm{w}} - c^{\mathrm{b}}) + c^{\mathrm{b}}}{\alpha} + (2a-1)\underline{\Delta} - b, \right. \\ & \left. \frac{ac^{\mathrm{w}}}{\alpha} + a\underline{\Delta} - (a-1)c^{\mathrm{b}} - b \right\} \leq 0. \end{aligned} \quad (49)$$

Notice that this optimization problem is the same as problem (25), except for the factor 2 in the term that captures switching overhead ($\frac{\epsilon(1-\alpha)}{\underline{\Delta}}$ versus $\frac{2\epsilon(1-\alpha)}{\underline{\Delta}}$). Therefore, the solution to this problem can be obtained using the approach in Section 5, simply by substituting the overhead $\frac{\epsilon}{2}$ instead of ϵ.

While the objective function of the above optimization problem is the same for both the exact and optimistic supply functions (i.e., $\alpha + \frac{\epsilon}{P}$), the stability constraint uses optimistic response times instead of the exact results. However, according to Lemma 1, if there exists a solution that guarantees the stability constraint when the exact supply functions are used, then there is also a solution considering the optimistic

TABLE 1
Example: task set data

i	c_i^{b}	c_i^{w}	h_i	a_i	b_i	$F(s)$
1	30	60	600	1.18	831	$\frac{1000}{s^2+s}$
2	92	184	920	1.16	826	$\frac{98.1}{s^2-98.1}$
3	427	854	2847	1.14	2697	$\frac{9.81}{s^2-9.81}$

TABLE 2
Example: Solution to the server design problem.

	Implicit Deadline					Asymptotic Analysis				
i	Q_i^*	P_i^*	α_i^*	Δ_i^*	O_i^*	Q_i^*	P_i^*	α_i^*	Δ_i^*	O_i^*
1	7.25	72.5	0.100	130	0.004	14.5	145	0.100	130	0.002
2	5.56	22.0	0.253	32.8	0.010	7.82	31.4	0.249	23.6	0.010
3	12.8	37.0	0.347	48.3	0.008	18.1	52.5	0.345	34.4	0.006
$\sum$			0.700		0.022			0.694		0.018

supply functions (i.e., the search space of this problem contains the search space of the exact problem). Therefore, the total utilization found in this section is lower than or equal to the one that could possibly be found for the implicit deadline server considering the exact supply functions.

8 EVALUATION

We will first illustrate and evaluate our proposed approach in Section 8.1 by a small example and later in Section 8.2 by a large set of experiments.

8.1 Illustrative example

In this section, the server design approach discussed in Section 5 will be illustrated using a small example. Further, we also compare the results to the asymptotic bound developed for the case of implicit deadline servers in Section 7.

Let us consider a set of three controllers whose parameters are reported in Table 1. In the table we report best-case and worst-case execution times (c_i^{b} and c_i^{w}), the period (h_i), the coefficients of the linear constraint between delay and jitter (a_i and b_i of the constraint in (21)), and the transfer function of the plant to be controlled. We assume a server switching overhead of $\epsilon = 0.3$. All time quantities are given in units of $0.01\ ms$ throughout this section.

The server parameters obtained after optimization are reported in Table 2. In the first column group of the table (labelled by "Implicit Deadline") we report server budgets Q_i, periods P_i, bandwidth α_i, delay Δ_i, and overhead due to switching $O_i = \frac{\epsilon}{P_i}$ for the implicit deadline design strategy (ID) proposed in Section 5. In the second column group of the table (labelled by "Asymptotic Analysis"), the corresponding results for the asymptotic analyses of the implicit deadline server (AA) in Section 7 are reported.

The total utilization obtained by the asymptotic analysis for the implicit deadline approach in Section 7 is $U_{\mathrm{AA}} = 0.71$, while the total utilization in the case of the implicit deadline servers obtained by our design approach (Section 5) is slightly higher, i.e., $U_{\mathrm{ID}} = 0.72$. The detailed calculation is given in the follow,

$$U_{\mathrm{AA}} = \sum_{i=1}^{3}\left(\alpha_i^* + \frac{\epsilon(1-\alpha_i^*)}{\Delta_i^*}\right) = \left(0.100 + \frac{0.3(1-0.100)}{130}\right) + \left(0.249 + \frac{0.3(1-0.249)}{23.6}\right) + \left(0.345 + \frac{0.3(1-0.345)}{34.4}\right) = 0.71,$$

$$U_{\mathrm{ID}} = \sum_{i=1}^{3}\left(\alpha_i^* + 2\frac{\epsilon(1-\alpha_i^*)}{\Delta_i^*}\right) = \left(0.100 + 2\frac{0.3(1-0.100)}{130}\right) + \left(0.253 + 2\frac{0.3(1-0.253)}{32.8}\right) + \left(0.347 + 2\frac{0.3(1-0.347)}{48.3}\right) = 0.72.$$

Observe that the solution found by the asymptotic analysis for the implicit deadline servers is not stable (it does not guarantee stability considering the exact stability condition). However, the solution obtained by the implicit deadline approach is guaranteed to be stable (valid considering the exact stability condition), while only 1% away from the asymptotic analysis, in terms of resource utilization. This indicates that, for the discussed example, the solution obtained by our approach is less than 1% away from the actual optimum.

8.2 Experimental results

To further evaluate our proposed server design approach we compare four different methods:

- **Implicit deadline servers:** the implicit deadline server design (ID) is proposed in Section 5.
- **Implicit deadline asymptotic analysis:** the asymptotic analysis of implicit deadline servers (AA) is discussed in Section 7 and produces solutions that are not guaranteed to be stable, but their resource utilization is less than or equal to the actual optimum.
- **Harmonic servers:** if we design the servers following the rules of Section 5, the periods of the servers will be unrelated to each other. For harmonic servers (HA), instead, we investigate the case in which we explicitly set all the server periods equal to the same value P [25].
- **General asymptotic analysis:** the general asymptotic analysis (GA) is an asymptotic analysis for both implicit deadline and harmonic servers and, therefore, it could be used as a common baseline for both servers. The idea is to consider the switching overhead negligible, in addition to considering the optimistic supply functions.

Note that the general asymptotic analysis (GA) is an approach that outperforms the optimal, in terms of total bandwidth usage, both for the implicit deadline and harmonic servers. In other words, the general asymptotic analysis (GA) produces a lower bound on the total bandwidth usage, but does not guarantee stability. Therefore, this general asymptotic analysis (GA) is considered as the baseline for the comparison. However, it is important to observe that the asymptotic analysis (AA), which also does not guarantee stability, is an approach that performs at least as well as the optimal, in terms of the required bandwidth,

if implicit deadline servers are considered. Therefore, asymptotic analysis (AA) provides a tighter lower bound for the optimal solution of implicit deadline servers. Hence, the gap between the implicit deadline (ID) and asymptotic analysis for implicit deadline server (AA) is the metric that is important for us (and not the absolute percentage reported).

We have generated 1000 benchmarks with a number of control applications from 2 to 10. The plants considered are chosen from a database consisting of inverted pendulums, ball and beam processes, DC servos, and harmonic oscillators [1], [5]. Such plants are considered to be representative of realistic control problems and are extensively used for experimental evaluation. To generate a set of random control tasks for a given utilization, the UUniFast algorithm is used [33]. The periods are chosen based on common rules of thumb [1]. Having the period and task utilization, the worst-case execution time can be computed. The switching overhead is given by $\epsilon = r \cdot \min_{i=1\ldots n}\{c_i^{\mathrm{b}}\}$, where r is randomly chosen with a uniform distribution in the interval of $[0.01, 0.10]$.

The experiments are repeated for several values of total task utilization ($\sum_{i=1}^{n} \frac{c_i^{\mathrm{w}}}{h_i}$) and the results are shown in Figure 7. The metric used for this comparison is the relative quality, defined as $\left(\frac{N_{\mathrm{GA}} - N_{\mathrm{X}}}{N_{\mathrm{GA}}} \times 100\right)$, where N_{X} and N_{GA} are the number of benchmarks for which the approach X and general asymptotic analysis, respectively, could find a valid solution. Therefore, the metric states the quality of the approach X (X could be HA, ID, or AA) compared to the general asymptotic analysis (GA). For each value of utilization, we evaluate the percentage of benchmarks for which the stability could not be guaranteed, and we call it "invalid solutions".[2]

The number of invalid solutions found for both implicit deadline and harmonic servers increases with utilization. Nevertheless, the harmonic servers perform slightly better for low utilization (50% utilization), while for high utilization (more than 55% utilization), the implicit deadline servers performs slightly better than the harmonic servers. It is also noteworthy that the gap between the implicit deadline approach (ID) and the asymptotic analysis (AA) is always less than 5%. In other words, the implicit deadline approach (ID) is less than 5% away from the theoretical optimum, for the benchmarks considered here. Interestingly, for low utilization, the harmonic (HA) performs slightly better than the asymptotic analysis for the implicit deadline (AA). The results illustrate that for high loads the possibility to assign individual server periods with the implicit deadline servers approach outweighs the advantage of potentially reduced jitters with the harmonic servers.

2. Note that, as mentioned, "valid" solutions with AA and GA are not guaranteed to be stable, as opposed to those produced with ID and HA.

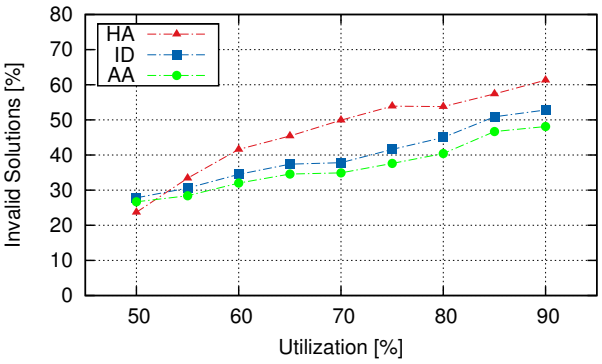


Fig. 7. The percentage of the benchmarks for which stability of the control task could not be guaranteed compared to the general asymptotic analysis (GA).

9 Conclusions

Providing guarantees for stability of control applications is perhaps the most important requirement while implementing embedded control systems. The fundamental difference between the control systems and what we classically understand by hard real-time systems advocates the need for new analysis and design techniques. In this paper, we have proposed the use of resource reservation mechanisms for designing embedded control systems. Exploiting the server mechanism provides not only compositionality, scalability, and isolation, but also a simple interface between the control stability and real-time scheduling aspects which facilitates the design process. Finally, we have addressed the analysis and design of stabilizing servers and demonstrated the efficiency of our proposed approaches both theoretically and experimentally.

Acknowledgements

The authors would like to acknowledge Prof. Anton Cervin and Dr. Bo Lincoln for helpful discussions and providing the Jitter Margin toolbox and the anonymous reviewers and editors of IEEE Transactions on Computers.

References

[1] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems*, 3rd ed. Prentice Hall, 1997.

[2] A. Cervin, "Stability and worst-case performance analysis of sampled-data control systems with input and output jitter," in *Proceedings of the 2012 American Control Conference (ACC)*, 2012.

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

[4] H. Rehbinder and M. Sanfridson, "Integration of off-line scheduling and optimal control," in *Proceedings of the 12th Euromicro Conference on Real-Time Systems*, 2000, pp. 137–143.

[5] A. Cervin, B. Lincoln, J. Eker, K. E. Årzén, and G. Buttazzo, "The jitter margin and its application in the design of real-time control systems," in *Proceedings of the 10th International Conference on Real-Time and Embedded Computing Systems and Applications*, 2004.

[6] T. Nghiem, G. J. Pappas, R. Alur, and A. Girard, "Time-triggered implementations of dynamic controllers," in *Proceedings of the 6th ACM & IEEE International conference on Embedded software*, 2006, pp. 2–11.

[7] E. Bini and A. Cervin, "Delay-aware period assignment in control systems," in *Proceedings of the 29th IEEE Real-Time Systems Symposium*, 2008, pp. 291–300.

[8] F. Zhang, K. Szwaykowska, W. Wolf, and V. Mooney, "Task scheduling for control oriented requirements for cyber-physical systems," in *Proceedings of the 29th IEEE Real-Time Systems Symposium*, 2008, pp. 47–56.

[9] P. Naghshtabrizi and J. P. Hespanha, "Analysis of distributed control systems with shared communication and computation resources," in *Proceedings of the 2009 American Control Conferance (ACC)*, 2009.

[10] R. Majumdar, I. Saha, and M. Zamani, "Performance-aware scheduler synthesis for control systems," in *Proceedings of the 9th ACM international conference on Embedded software*, 2011, pp. 299–308.

[11] P. Kumar, D. Goswami, S. Chakraborty, A. Annaswamy, K. Lampka, and L. Thiele, "A hybrid approach to cyber-physical systems verification," in *Proceedings of the 49th Design Automation Conference*, 2012.

[12] A. Aminifar, S. Samii, P. Eles, Z. Peng, and A. Cervin, "Designing high-quality embedded control systems with guaranteed stability," in *Proceedings of the 33th IEEE Real-Time Systems Symposium*, 2012, pp. 283–292.

[13] A. Cervin and J. Eker, "Control-scheduling codesign of real-time systems: The control server approach," *Journal of Embedded Computing*, vol. 1, no. 2, pp. 209–224, 2005.

[14] A. Aminifar, E. Bini, P. Eles, and Z. Peng, "Bandwidth-efficient controller-server co-design with stability guarantees," in *Proceedings of the 17th Conference for Design, Automation and Test in Europe (DATE)*, 2014.

[15] X. Feng and A. Mok, "A model of hierarchical real-time virtual resources," in *Proceedings of the 23th IEEE Real-Time Systems Symposium*, 2002, pp. 26–35.

[16] S. Saewong, R. Rajkumar, J. Lehoczky, and M. Klein, "Analysis of hierarhical fixed-priority scheduling," in *Proceedings of the 14th Euromicro Conference on Real-Time Systems*, 2002, pp. 152–160.

[17] G. Lipari and E. Bini, "Resource partitioning among real-time applications," in *Proceedings of the 15th Euromicro Conference on Real-Time Systems*, 2003, pp. 151–158.

[18] I. Shin and I. Lee, "Periodic resource model for compositional real-time guarantees," in *Proceedings of the 24th IEEE Real-Time Systems Symposium*, 2003, pp. 2–13.

[19] L. Almeida and P. Pedreiras, "Scheduling within temporal partitions: response-time analysis and server design," in *Proceedings of the 4th ACM international conference on Embedded software*, 2004, pp. 95–103.

[20] A. Easwaran, M. Anand, and I. Lee, "Compositional analysis framework using edp resource models," in *Proceedings of the 28th IEEE Real-Time Systems Symposium*, 2007, pp. 129–138.

[21] N. Fisher and F. Dewan, "A bandwidth allocation scheme for compositional real-time systems with periodic resources," *Real-Time Systems*, vol. 48, no. 3, pp. 223–263, 2012.

[22] D. Fontantelli, L. Palopoli, and L. Greco, "Optimal cpu allocation to a set of control tasks with soft real–time execution constraints," in *Proceedings of the 16th international conference on Hybrid systems: computation and control*, 2013, pp. 233–242.

[23] K. E. Årzén and A. Cervin, "Control and embedded computing: Survey of research directions," in *Proceedings of the 16th IFAC World Congress*, 2005.

[24] D. Fontanelli, L. Palopoli, and L. Abeni, "The continuous stream model of computation for realtime control," in *Proceedings of the 34th IEEE Real-Time Systems Symposium, Vancouver, Canada*, December 2013.

[25] A. Aminifar, E. Bini, P. Eles, and Z. Peng, "Designing bandwidth-efficient stabilizing control servers," in *Proceedings of the 34th IEEE Real-Time Systems Symposium*, 2013.

[26] C.-Y. Kao and B. Lincoln, "Simple stability criteria for systems with time-varying delays," *Automatica*, vol. 40, pp. 1429–1434, 2004.

[27] B. Wittenmark, J. Nilsson, and M. Törngren, "Timing problems in real-time control systems," in *Proceedings of the American Control Conference*, 1995, pp. 2000–2004.

[28] G. Buttazzo and E. Bini, "Optimal dimensioning of a constant bandwidth server," in *Proceedings of the 27th IEEE Real-Time Systems Symposium*, 2006, pp. 169–177.

[29] J. Lehoczky, "Fixed priority scheduling of periodic task sets with arbitrary deadlines," in *Proceedings of the 11th IEEE Real-Time Systems Symposium*, 1990, pp. 201–209.

[30] E. Bini, T. Huyen Châu Nguyen, P. Richard, and S. K. Baruah, "A response-time bound in fixed-priority scheduling with arbitrary deadlines," *IEEE Transactions on Computer*, vol. 58, no. 2, pp. 279–286, 2009.

[31] M. Bazaraa, H. Sherali, and C. Shetty, *Nonlinear Programming: Theory and Algorithms*. Wiley, 2006.

[32] A. Aminifar, P. Eles, Z. Peng, and A. Cervin, "Stability-aware analysis and design of embedded control systems," in *Proceedings of the International Conference on Embedded Software (EMSOFT)*, 2013, pp. 1–10.

[33] E. Bini and G. C. Buttazzo, "Measuring the performance of schedulability tests," *Real-Time Systems*, vol. 30, no. 1-2, pp. 129–154, 2005.

Amir Aminifar is a Ph.D. student in the Computer Science Department of Linköping University, Sweden. He received his B.Sc. from Sharif University of Technology in 2010. He has been a visiting graduate researcher in the Cyber-Physical group of the University of California, Los Angeles, November 2014 to January 2015.

Enrico Bini is Assistant professor at Scuola Superiore Sant'Anna, Pisa, Italy. In 2012-13, he was Marie-Curie fellow at Lund University (Sweden) investigating virtualization in Cyber-Physical Systems. In 2000 he received the Laurea degree in Computer Engineering from University of Pisa. In 2004, he completed the doctoral studies on Real-Time Systems at Scuola Superiore Sant'Anna (recipient of the "Spitali Award" for best Ph.D. thesis of the Scuola Superiore Sant'Anna). In January 2010 he also completed a Master degree in Mathematics with a thesis on optimal sampling for linear control systems.

He is an IEEE Senior member. He has published more than 80 papers (two best-paper awards) on real-time scheduling, and design and optimization methods for real-time and control systems. His more recent research interests are on optimal management of distributed and parallel resources.

Petru Eles is Professor of Embedded Computer Systems with the Department of Computer and Information Science (IDA), Linköping University. Petru Eles' current research interests include embedded systems, real-time systems, electronic design automation, cyber-physical systems, hardware/software codesign, low power system design, fault-tolerant systems, design for test. He has published a large number of technical papers in these areas and co-authored several books. Petru Eles received several best paper awards at major conferences.

Zebo Peng is Professor of Computer Systems, Director of the Embedded Systems Laboratory, and Vice-Chairman of the Department of Computer Science at Linköping University. He has published more than 300 technical papers and five books in various topics related to embedded systems, and has received four best paper awards and one best presentation award in major international conferences.