Designing Bandwidth-Efficient Stabilizing Control Servers

Amir Aminifar*, Enrico Bini†, Petru Eles*, Zebo Peng*
*Linköping University, Sweden
†Lund University, Sweden

***Abstract*—Guaranteeing stability of control applications in embedded systems, or cyber-physical systems, is perhaps the alpha and omega of implementing such applications. However, as opposed to the classical real-time systems where often the acceptance criterion is meeting the deadline, control applications do not primarily enforce hard deadlines. In the case of control applications, stability is considered to be the main design criterion and can be expressed in terms of the amount of delay and jitter a control application can tolerate before instability. Therefore, new design and analysis techniques are required for embedded control systems.**

In this paper, the analysis and design of such systems considering server-based resource reservation mechanism are addressed. The benefits of employing servers are manifold: (1) providing a compositional framework, (2) protection against other tasks misbehaviors, and (3) systematic bandwidth assignment. We propose a methodology for designing bandwidth-efficient servers to stabilize control tasks.

I. 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.

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 then possible to perform the, so called, *real-time control co-design*: designing a controller so that the control performance is guaranteed (stability, LQR cost minimization, etc.) and the control tasks are schedulable over the available processing unit.

In typical approaches [3], [4], [5], 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. 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. As illustrated in Figure 1, in this paper we propose instead to run each controller within its own server, which then isolates each control task in the execution environment. The usage of servers to protect control tasks presents the following advantages:

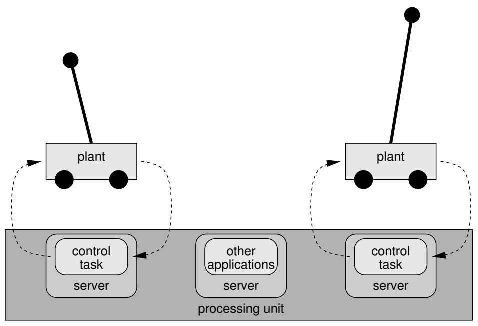


Figure 1. Overview of the proposed approach.

- it provides compositionality that is important for systematic design methodologies;
- it protects each controller from possible misbehaviors, which may occur within other tasks and then possibly jeopardize the entire system;
- the bandwidth assignment, rather that the priority assignment, may constitute a more accurate instrument to allocate the available computing resources;
- 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.

A. Related work

Over the past decade, the analysis and design of real-time servers have widely developed. Feng and Mok [6] 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 [7], [8], [9], [10]. Easwaran et. al. [11] 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. Similarly as we do in this paper, Fisher and Dewan [12] described a method to minimize the bandwidth of a server. They developed a FPTAS to solve the problem. However, as the majority of the works in this area, they considered the task deadline as constraint 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. More recently, Fontanelli et. al. [14] addressed the problem of optimal bandwidth allocation for a set of

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.

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 approaches are restricted solely to the very particular time-triggered design and implementation approach which can potentially lead to under-utilization or poor control performance [15].

B. 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 criterion 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 [16], [17], [2]. The stability of control applications depends on not only the amount of delay, but also on the amount of jitter a control application experiences [18]. 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 exact 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 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 bound the amount of pessimism introduced by our technique.

The main contribution of this paper is in providing a methodology for designing servers to stabilize control tasks, which consume the minimal bandwidth.

II. Systems model

The system is composed of n plants. Each plant is controlled by a control task which is executing within a server, as shown in Figure 1. Below we describe the model of the plant, the control task, and the server.

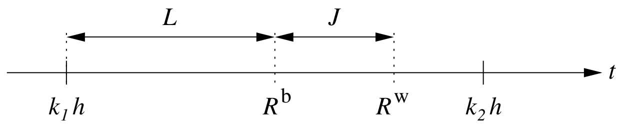


Figure 2. Graphical interpretation of the nominal delay and worst-case response-time jitter.

A. 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.

B. Control task model

The control signal $\boldsymbol{u}$ is computed by a control task τ, which samples the output $\boldsymbol{y}$ every period h. 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, and
- the *worst-case response-time jitter* (jitter), denoted by $J = R^{\mathrm{w}} - R^{\mathrm{b}}$.

The terminology and the notation are illustrated in Figure 2. Note that we do not consider any deadline for control tasks.

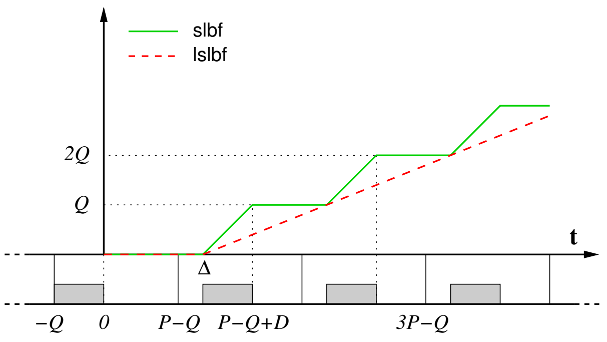
C. Server model

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:

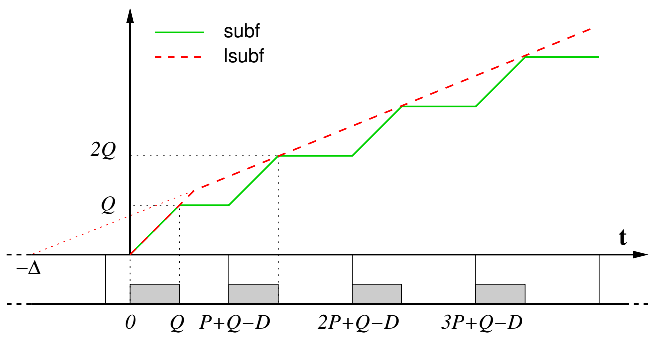
- the *server budget* Q;
- the *server period* P, and
- the *server deadline* D.

This model was also called EDP (Explicit Deadline Periodic) model by some authors [11]. 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.



(a) Worst-case resource allocation scenario.



(b) Best-case resource allocation scenario.

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

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 [6], [8], [9], [10], [11] 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

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

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 [11] 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 linearly lower bound the $\mathsf{slbf}(t)$ by the *linear supply lower bound function* $\mathsf{lslbf}(t)$, defined as

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

with, using Feng–Mok's notation [6], the server bandwidth α and delay Δ, defined as

$$\alpha = \frac{Q}{P} \quad (4)$$
$$\Delta = P + D - 2Q. \quad (5)$$

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

Analogously, the best-case analysis of the control task within the server can be performed by computing 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

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

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

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

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).

III. 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) (Section III-A) as well as with the linear bounds lslbf/lsubf of (3) and (7) (Section III-B).

A. 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 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 0) is equal to the first instant when the server has necessarily provided at least c^{w} amount of time, that is

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

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

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

The proof is similar to [19].

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 [20] for the arbitrary deadline case. An example is provided to illustrate that the worst-case response time does not necessarily occur for the instance of a task which starts the busy period. Let us consider a control task with $c^{\mathrm{w}} = c^{\mathrm{b}} = 62$ and period $h = 100$ running within a server with $Q = 44$ and $P = D = 70$. In this case, the longest response time occurs at the fifth job in the busy period. In fact, the response time of the jobs in the busy period are: 140, 128, 142, 130, **144**, 132, 120, 134, 122, 136, 124, 112, 126, 114, 128, 116, 104, 118, 106, 120, 108, and finally 96 which is smaller than the controller period.

Similar to Lehoczky's analysis for tasks with arbitrary deadlines [20], the worst-case response time of the control task within a server $S = (Q, P, D)$ is obtained as follows,

$$R^{\mathrm{w}} = \max_{q=1\dots q^{\max}} \left\{ D - Q + \left\lceil \frac{qc^{\mathrm{w}}}{Q} \right\rceil (P - Q) + qc^{\mathrm{w}} - (q-1)h \right\}, \quad (10)$$

where $q^{\max}$ is the smallest natural number q for which we have,

$$D - Q + \left\lceil \frac{qc^{\mathrm{w}}}{Q} \right\rceil (P - Q) + qc^{\mathrm{w}} \leq qh. \quad (11)$$

The inequality above identifies the first instance (smallest q) of the task under analysis in the busy period where its execution finishes before the next instance is released. We remind that inequality (11) has solution for a finite q only when

$$\alpha = \frac{Q}{P} > \frac{c^{\mathrm{w}}}{h}.$$

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

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

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

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

The proof is similar to the proof of Theorem 1 in [8].

B. Characterization with linear bounds

The main obstacle in using the exact response time for finding the optimal server parameters (see Section V) 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

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

As shown in [21], such an upper bound to the response time is valid only if the server bandwidth is not smaller than the worst-case utilization of the control task, that is

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

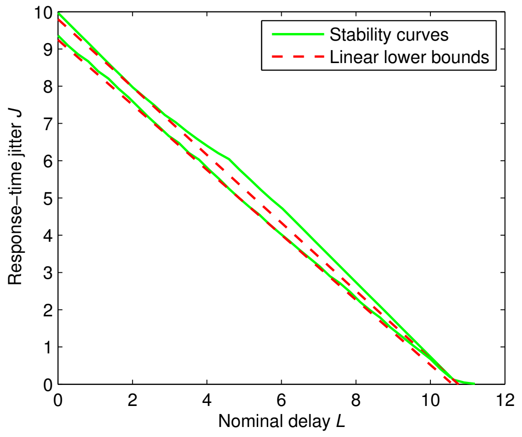


Figure 4. The stability curves generated by the Jitter Margin toolbox and their linear lower bounds (the area below the curves is the stable area).

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

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

IV. Stability constraint

To quantify the amount of delay and jitter tolerable by a control application before the instability of the plant, we use the Jitter Margin toolbox [16], [17], [2]. It provides sufficient stability conditions for a closed-loop system with a linear continuous-time plant and a linear discrete-time controller.

The Jitter Margin toolbox provides the stability curve that determines the maximum tolerable response-time jitter J based on the nominal delay L. 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 a discrete-time Linear-Quadratic-Gaussian (LQG) controller. The upper and lower solid curves correspond to sampling periods 6 ms and 12 ms, respectively.

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 *stability condition* for a control application is of the form $L + aJ \leq b$, 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 (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. Observe that the linear lower bounds can efficiently capture the stable area identified by Jitter Margin.

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,

$$\begin{aligned} L &= R^{\mathrm{b}}, \\ J &= R^{\mathrm{w}} - R^{\mathrm{b}}, \end{aligned} \tag{17}$$

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

$$\begin{aligned} L + aJ &\leq b, \\ R^{\mathrm{b}} + a(R^{\mathrm{w}} - R^{\mathrm{b}}) &\leq b. \end{aligned} \tag{18}$$

For a given server, the stability condition (18), 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 V, 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 unsuited. 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,

$$\begin{aligned} \underline{L} &= \underline{R}^{\mathrm{b}}, \\ \overline{J} &= \overline{R}^{\mathrm{w}} - \underline{R}^{\mathrm{b}}. \end{aligned} \tag{19}$$

While using the linear supply bounds involves some pessimism compared to the original supply bounds, it is safe from the stability point of view [17]. Nonetheless, the amount of introduced pessimism is discussed at the end of this section in Theorem 1.

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

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

which we rewrite as

$$\min\left\{\frac{a(c^{\mathrm{w}} - c^{\mathrm{b}}) + c^{\mathrm{b}}}{\alpha} + (2a-1)\Delta - b, \right. \\ \left. \frac{ac^{\mathrm{w}}}{\alpha} + a\Delta - (a-1)c^{\mathrm{b}} - b\right\} \leq 0. \tag{20}$$

Hence equation (20) describes the constraint on the server parameters (the bandwidth α and the delay Δ, see Section II-C), which guarantees the stability of the controller running within such a server.

We shall now discuss the degree of pessimism introduced in the analysis by using the linear bounds instead of the exact response times. The next theorem establishes one interesting consequence of the violation of (20), which is derived with the linear bounds.

Theorem 1: If the stability constraint (20) of a control task is satisfied within an implicit deadline server $S_1 = (Q, P)$ with the exact supply functions, it is also satisfied within an implicit deadline server $S_2 = (\frac{Q}{2}, \frac{P}{2})$ with the linear supply functions.

Proof: Let us first prove the following inequalities,

$$\begin{aligned} \forall t, \quad \mathsf{subf}_1(t) &\geq \mathsf{lsubf}_2(t), \\ \forall t, \quad \mathsf{slbf}_1(t) &\leq \mathsf{lslbf}_2(t), \end{aligned} \tag{21}$$

where the indices 1 and 2 correspond to servers S_1 and S_2, respectively. To prove $\mathsf{subf}_1(t) \geq \mathsf{lsubf}_2(t)$, we derive the linear lower bound on the exact supply upper bound function $\mathsf{subf}_1(t)$. If we can prove that this linear lower bound is always greater than or equal to $\mathsf{lsubf}_2(t)$, considering that it is a lower bound of $\mathsf{subf}_1(t)$, we have $\mathsf{subf}_1(t) \geq \mathsf{lsubf}_2(t)$. The linear lower bound on $\mathsf{subf}_1(t)$ is given by,

$$\min\left\{t, \frac{Q}{P}(t + (P - Q))\right\}. \tag{22}$$

Let us also derive the $\mathsf{lsubf}_2(t)$ for the implicit deadline server $S_2 = (\frac{Q}{2}, \frac{P}{2})$,

$$\min\left\{t, \frac{\frac{Q}{2}}{\frac{P}{2}}(t + 2(\frac{P}{2} - \frac{Q}{2}))\right\},$$

which is exactly the same as the linear lower bound on the exact supply upper bound function $\mathsf{subf}_1(t)$ in equation (22) (see Figure 5(b)).

Analogously, to prove $\mathsf{slbf}_1(t) \leq \mathsf{lslbf}_2(t)$, we show that the linear upper bound on $\mathsf{slbf}_1(t)$ is the same as $\mathsf{lslbf}_2(t)$ and is given by (see Figure 5(a)),

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

As a result of the inequalities in (21), the following relations hold for the response times,

$$\begin{aligned} R_1^{\mathrm{b}} &= \min\{t : \mathsf{subf}_1(t) \geq c^{\mathrm{b}}\} \leq \min\{t : \mathsf{lsubf}_2(t) \geq c^{\mathrm{b}}\} = \underline{R}_2^{\mathrm{b}}, \\ R_1^{\mathrm{w}} &= \min\{t : \mathsf{slbf}_1(t) \geq c^{\mathrm{w}}\} \geq \min\{t : \mathsf{lslbf}_2(t) \geq c^{\mathrm{w}}\} = \overline{R}_2^{\mathrm{w}}. \end{aligned}$$

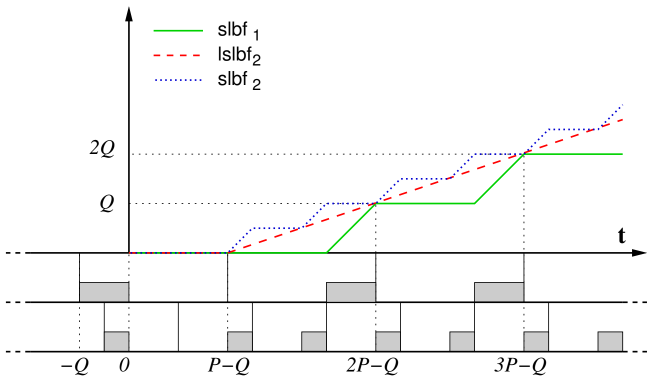
Since $a \geq 1$, we have the following inequalities,

$$\begin{aligned} aR_1^{\mathrm{w}} + (1-a)R_1^{\mathrm{b}} &\geq a\overline{R}_2^{\mathrm{w}} + (1-a)\underline{R}_2^{\mathrm{b}}, \\ L_1 + aJ_1 &\geq \underline{L}_2 + a\overline{J}_2, \end{aligned}$$

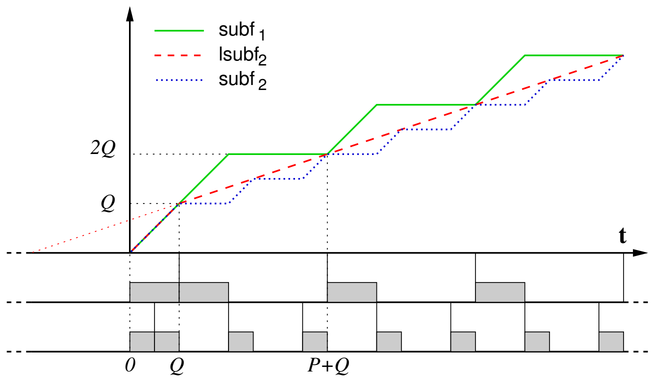
from which the theorem follows,

$$L_1 + aJ_1 \leq b \quad \overset{L_1 + aJ_1 \geq \underline{L}_2 + a\overline{J}_2}{\Longrightarrow} \quad \underline{L}_2 + a\overline{J}_2 \leq b. \tag{23}$$

Note that the bound is tight since the linear lower bound on $\mathsf{subf}_1(t)$ is the same as $\mathsf{lsubf}_2(t)$ and the linear upper bound on $\mathsf{slbf}_1(t)$ is the same as $\mathsf{lslbf}_2(t)$. The tightness is in the sense that, for server $S_2 = (\frac{Q}{2}, \frac{P}{2})$, increasing the server period $\frac{P}{2}$ or decreasing the server budget $\frac{Q}{2}$ by any small positive value, violates the inequalities in (21). ∎



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



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

Figure 5. Worst-case and best-case resource allocation scenarios for implicit deadline server.

The important message of Theorem 1 is that, if a server $S_1 = (Q, P)$ (with the exact supply functions) with bandwidth $\alpha_1 = \frac{Q}{P}$ is identified that guarantees the stability of the control task associated with it, then there exists a server $S_2 = (\frac{Q}{2}, \frac{P}{2})$ (with the linear supply functions) that can guarantee the stability of the control task and the required bandwidth is the same, i.e., $\alpha_2 = \frac{\frac{Q}{2}}{\frac{P}{2}} = \frac{Q}{P}$.

The theorem also states that, *in the worst-case*, the server S_2 has to be run at double frequency compared to S_1. In practice, of course, this might be a disadvantage, if the context-switch overhead is significant.

V. 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 two methods for server design:

- the implicit deadline servers, in which all server deadlines are set equal to the periods (Section V-A), and
- the harmonic servers, in which all server periods are equal to each other (Section V-B).

A. Design of implicit deadline servers

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 (20)). 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 all 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,

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

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

The solution to the above problem is the minimum bandwidth (included 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. Since the stability constraint in (25) can be written as

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

which is equivalent to

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

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

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

and,

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

and then select the best solution between the two solutions produced by (26) and (27).

To solve problems (26) and (27), we use the KKT (Karush-Kuhn-Tucker) necessary conditions for optimality [22]. According to the KKT conditions, the optimum $\boldsymbol{x}^*$ of the problem

$$\begin{aligned}\min_{\boldsymbol{x}} \ & f(\boldsymbol{x}) \\ \text{s.t.} \ & g(\boldsymbol{x}) \leq 0,\end{aligned} \tag{28}$$

must necessarily satisfy the following condition

$$\begin{aligned}\nabla f(\boldsymbol{x}^*) + \mu^* \nabla g(\boldsymbol{x}^*) =& \boldsymbol{0}, \\ \mu^* g(\boldsymbol{x}^*) =& 0, \\ \mu^* \geq& 0.\end{aligned} \tag{29}$$

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

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

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

Since $a \geq 1$ and $\alpha < 1$, from (31), we immediately find the multiplier μ, that is:

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

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 is

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

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

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

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:

$$\alpha_{\mathrm{I}}^* = \max\left\{\alpha_{\mathrm{I}}(1+\delta_{\mathrm{I}}), \frac{c^{\mathrm{w}}}{h}\right\}, \tag{37}$$

in which we also account for constraint (15) on the minimal server bandwidth that guarantees the validity of the response time upper bound. 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

$$x_{\mathrm{II}} = ac^{\mathrm{w}}, \quad y_{\mathrm{II}} = a\epsilon, \quad z_{\mathrm{II}} = b + (a-1)c^{\mathrm{b}}. \tag{38}$$

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 (38). Since the cost functions of the two problems are the same, it follows that the solution is exactly the same as (37), with the opportune 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.,

$$\min\left\{\alpha_{\mathrm{I}}^* + \frac{2\epsilon(1-\alpha_{\mathrm{I}}^*)}{\Delta_{\mathrm{I}}^*}, \alpha_{\mathrm{II}}^* + \frac{2\epsilon(1-\alpha_{\mathrm{II}}^*)}{\Delta_{\mathrm{II}}^*}\right\}. \tag{39}$$

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, the solution found is valid if and only if the utilization is less than or equal to one, i.e.,

$$\sum_{i=1}^{n} \left(\alpha_i^* + \frac{2\epsilon(1-\alpha_i^*)}{\Delta_i^*}\right) \leq 1. \tag{40}$$

B. Design of harmonic servers

If we design the servers following the rules of Section V-A, the periods of the servers will certainly be unrelated to one another. In this section, instead, we investigate the case in which we explicitly set all the server periods equal to the same value P. This choice has the following advantages:

- setting all periods equal to each other is certainly simpler to be implemented;
- as shown in Figure 6, it is possible to ameliorate the worst-case and the best-case scenarios for the resource supply. Such a scenario is equivalent to assuming $D =$

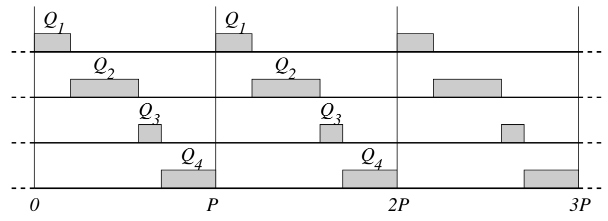


Figure 6. Server supply in the harmonic case.

Q for all servers and then allows setting (see equation (5)):

$$\Delta = P - Q = P(1-\alpha) \tag{41}$$

rather than $2(P-Q)$ as in the implicit deadline case.

However, there is also a drawback that is:

- binding all periods to be equal, it is not anymore possible to find the server parameters independently of each other as in Section V-A.

In the case of all server periods equal to P the total utilization of the control servers is given by,

$$U = \sum_{i=1}^{n}\left(\alpha_i + \frac{\epsilon}{P}\right) = \sum_{i=1}^{n}\alpha_i + \frac{n\epsilon}{P}, \tag{42}$$

where all servers have the same period, denoted by P.

For each server with bandwidth α, we consider then the following problem

$$\begin{aligned} \min_{\alpha} \quad & \alpha \\ \text{s.t.} \quad & \min\left\{\frac{a(c^{\rm w}-c^{\rm b})+c^{\rm b}}{\alpha} + (2a-1)P(1-\alpha) - b, \right. \\ & \left. \frac{ac^{\rm w}}{\alpha} + aP(1-\alpha) - (a-1)c^{\rm b} - b\right\} \le 0, \end{aligned} \tag{43}$$

which follows from the stability constraint (20) after replacing the delay Δ with the less pessimistic expression of (41), possible thanks to the assumption of all periods equal to P.

As discussed in the previous section, solving the above problem is equivalent to solving the following two problems,

$$\begin{aligned} \min_{\alpha} \quad & \alpha \\ \text{s.t.} \quad & \frac{a(c^{\rm w}-c^{\rm b})+c^{\rm b}}{\alpha} + (2a-1)P(1-\alpha) - b \le 0, \end{aligned} \tag{44}$$

and,

$$\begin{aligned} \min_{\alpha} \quad & \alpha \\ \text{s.t.} \quad & \frac{ac^{\rm w}}{\alpha} + aP(1-\alpha) - (a-1)c^{\rm b} - b \le 0, \end{aligned} \tag{45}$$

and choosing then the minimum among the two solutions.

As minimizing the server utilization α leads to an increase in the left-hand side of the stability constraints, the minimum server utilization α^* is obtained when the constraint is active, in both cases.

The stability constraints of problems (44) and (45) are given by a quadratic equation each of which with only one positive solution. The unique solution of the two problems, can be written as

$$\alpha_i^* = \alpha_i \frac{2}{\sqrt{(1-\delta_i)^2 + 4\delta_i\alpha_i} + (1-\delta_i)}, \tag{46}$$

with

$$\alpha_{\rm I} = \frac{a(c^{\rm w}-c^{\rm b})+c^{\rm b}}{b}, \quad \delta_{\rm I} = \gamma_{\rm I}P, \quad \gamma_{\rm I} = \frac{(2a-1)}{b} \tag{47}$$

determining the solution of problem (44), and

$$\alpha_{\rm II} = \frac{ac^{\rm w}}{b+(a-1)c^{\rm b}}, \quad \delta_{\rm II} = \gamma_{\rm II}P, \quad \gamma_{\rm II} = \frac{a}{b+(a-1)c^{\rm b}} \tag{48}$$

defining the solution of problem (45). Finally, since the minimal bandwidth α^* is the minimum among the two problems, we have:

$$\alpha^* = \min\{\alpha_{\rm I}^*, \alpha_{\rm II}^*\}.$$

We now observe that both $\alpha_{\rm I}^*$ and $\alpha_{\rm II}^*$ are increasing functions of P. This can be argued from the problem formulations (44)–(45). In fact, the constraint must be active, regardless of the value of P. If P increases and the left-hand side of the constraint must remain equal to 0 (since the constraint is active), then α must necessarily increase as well. Therefore, the optimal server utilization $\alpha_{\rm I}^*$ and $\alpha_{\rm II}^*$ of both cases are increasing functions of the server period P.

To study the minimal bandwidth α^*, it is necessary to determine the value of the period which separates the case when $\alpha_{\rm I}^* \le \alpha_{\rm II}^*$ and vice versa. After some algebraic manipulation we find that $\alpha_{\rm II}^* \le \alpha_{\rm I}^*$ when

$$P \le \hat{P} = \frac{(\gamma_{\rm I}-\gamma_{\rm II})(\alpha_{\rm I}-\alpha_{\rm II})}{(\gamma_{\rm II}(1-\alpha_{\rm I}) - \gamma_{\rm I}(1-\alpha_{\rm II}))(\gamma_{\rm I}\alpha_{\rm II} - \gamma_{\rm II}\alpha_{\rm I})} \tag{49}$$

with $\alpha_{\rm I}$, $\alpha_{\rm II}$, $\gamma_{\rm I}$, and $\gamma_{\rm II}$ properly defined in (47)–(48).

Therefore, for the optimal server utilization α^* we have,

$$\alpha^* = \begin{cases} \alpha_{\rm II}^* & P \in [0, \hat{P}) \\ \alpha_{\rm I}^* & P \in [\hat{P}, +\infty) \end{cases} \tag{50}$$

The solution (50) provides the minimum bandwidth, as a function of P, which guarantees the stability of one controller only, when it is running within a server with period P that can guarantee to supply the resource according to the schedule of Figure 6.

Let us now address the minimization of the total utilization (42). For this purpose we introduce again the indices of the tasks, so that α_i^* denotes the solution of (50) for the i-th controller and $\alpha_{i,\rm I}^*$ and $\alpha_{i,\rm II}^*$ denote the two solutions of (46) for the i-th task. Without loss of generality, suppose for any two control tasks τ_i and τ_j, where $i < j$, we have $\hat{P}_i \le \hat{P}_j$.

Table I
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	$\frac{9.81}{s^2-9.81}$

With this notation, the cost of (42), can be written as

$$U(P) = \frac{n\epsilon}{P} + \begin{cases} \sum_{i=1}^{n} \alpha_{i,\mathrm{II}}^{*} & P \in [0, \hat{P}_1) \\ \alpha_{1,\mathrm{I}}^{*} + \sum_{i=2}^{n} \alpha_{i,\mathrm{II}}^{*} & P \in [\hat{P}_1, \hat{P}_2) \\ \dots \\ \sum_{i=1}^{n-1} \alpha_{i,\mathrm{I}}^{*} + \alpha_{n,\mathrm{II}}^{*} & P \in [\hat{P}_{n-1}, \hat{P}_n) \\ \sum_{i=1}^{n} \alpha_{i,\mathrm{I}}^{*} & P \in [\hat{P}_n, +\infty). \end{cases} \tag{51}$$

The minimization of U over P, can then be made by minimizing U in each of the intervals of the definition (51), taking into consideration that the servers should not be overloaded. Let then U_j^* be the minimum consumed bandwidth over the interval $[\hat{P}_j, \hat{P}_{j+1})$, with the proper extension to $\hat{P}_0 = 0$ and $\hat{P}_{n+1} = +\infty$. The minimum consumed bandwidth using the scheme of harmonic periods is

$$U^* = \min_{j=0\dots n} \{U_j^*\}, \tag{52}$$

and if we set j^* the index such that $U_{j^*}^* = U^*$, then the optimal period P^* is the one such that $U(P) = U_{j^*}^*$.

If the minimum utilization found U^* is less than or equal to one, then the system is guaranteed to be stable and schedulable, since this is the necessary and sufficient condition,

$$\sum_{i=1}^{n} (Q_i + \epsilon) \le P \quad \Leftrightarrow \quad U = \sum_{i=1}^{n} \left(\alpha_i + \frac{\epsilon}{P}\right) \le 1. \tag{53}$$

VI. ILLUSTRATIVE EXAMPLE

In this section, the server design approaches discussed in the previous section will be illustrated using a small example. Let us consider a set of three controllers whose data is reported in Table I. 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 of (18)), and the transfer function of the plant to be controlled. All time quantities are given in 0.01 ms throughout this section.

If we assume a server switching overhead $\epsilon = 0.3$, then the obtained server parameters are the ones reported in Table II.

In the table we report server budgets Q_i and periods P_i for both design strategies: the implicit deadline (ID) and the harmonic periods (H) cases. The total utilization of the servers designed with equal periods, $U_{\mathrm{H}} = 0.74$, is slightly higher than the total utilization in the case of the implicit deadline servers, i.e., $U_{\mathrm{ID}} = 0.72$. The detailed calculation is given in the follow,

Table II
EXAMPLE: SOLUTION TO THE SERVER DESIGN PROBLEM.

	Implicit Deadline				Harmonic			
i	Q_i^*	P_i^*	α_i^*	Δ_i^*	Q_i^*	P_i^*	α_i^*	Δ_i^*
1	7.25	72.5	0.100	130	4.90	49.0	0.100	44.1
2	5.56	22.0	0.253	32.8	13.0	49.0	0.266	36.0
3	12.8	37.0	0.347	48.3	17.5	49.0	0.358	31.4

$$\begin{aligned} U_{\mathrm{H}} &= \textstyle\sum_{i=1}^{3} \left(\alpha_i^* + \frac{\epsilon(1-\alpha_i^*)}{\Delta_i^*}\right) = \left(0.100 + \frac{0.3(1-0.100)}{44.1}\right) \\ &\quad + \left(0.266 + \frac{0.3(1-0.266)}{36.0}\right) + \left(0.358 + \frac{0.3(1-0.358)}{31.4}\right) = 0.74, \\ U_{\mathrm{ID}} &= \textstyle\sum_{i=1}^{3} \left(\alpha_i^* + 2\frac{\epsilon(1-\alpha_i^*)}{\Delta_i^*}\right) = \left(0.100 + 2\frac{0.3(1-0.100)}{130}\right) \\ &\quad + \left(0.253 + 2\frac{0.3(1-0.253)}{32.8}\right) + \left(0.347 + 2\frac{0.3(1-0.347)}{48.3}\right) = 0.72. \end{aligned}$$

Notice that the server delays Δ_1 and Δ_3 are smaller in the case of the harmonic server than the implicit deadline server.

VII. EXPERIMENTAL RESULTS

To further compare the implicit deadline server and the harmonic server designed in the previous section, 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], [17]. Such plants are considered to be representatives of realistic control problems and are extensively used for experimental evaluation. To generate a set of random tasks for a given utilization, the UScaling algorithm is used [23]. The switching overhead is given by $\epsilon = r \times \min_{i=1\dots n}\{c_i^{\mathrm{b}}\}$, where $r \in [0.01, 0.05]$.

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 improvement, defined as $\left(\frac{N_{\mathrm{ID}}-N_{\mathrm{H}}}{N_{\mathrm{ID}}} \times 100\right)$, where N_{ID} and N_{H} are the number of benchmarks for which the implicit deadline servers and harmonic servers, respectively, could find a valid solution. Therefore, the metric states the efficiency of the implicit deadline servers compared to the harmonic servers. 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". The number of invalid solutions found for harmonic servers increases with utilization compared to the implicit deadline servers. Nevertheless, the harmonic servers perform 3.6% better for low utilization (50% utilization), while for high utilization (95% utilization) the implicit deadline servers perform 27.6% better than the harmonic servers.

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.

VIII. CONCLUSIONS

Providing guarantees for stability of control applications is perhaps the most important requirement while implementing embedded control systems. The fundamental difference

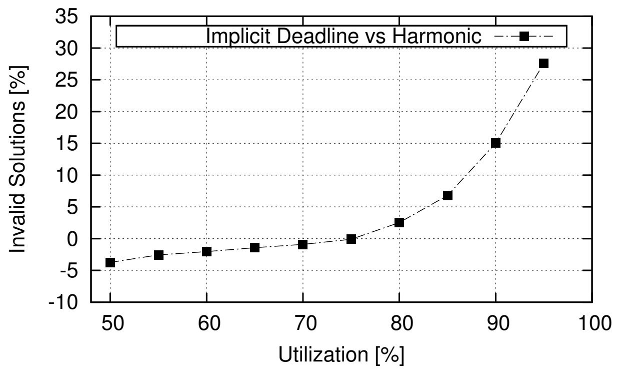


Figure 7. The percentage of the benchmarks for which stability of the control task associated with the harmonic server could not be guaranteed compared to the implicit deadline server.

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 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.

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] 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.

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

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[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] 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.

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

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

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

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

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