Harmonic Scheduling and Control Co-Design

Xu, Yang; Cervin, Anton; Årzén, Karl-Erik

Published in:
2016 IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)

DOI:
10.1109/RTCSA.2016.44

2016

Document Version:
Peer reviewed version (aka post-print)

Harmonic Scheduling and Control Co-Design

Yang Xu, Anton Cervin, Karl-Erik Årzén
Department of Automatic Control
Lund University, Sweden
Email: yang, anton, karlerik@control.lth.se

***Abstract*—Harmonic task scheduling has many attractive properties, including a utilization bound of 100% under rate-monotonic scheduling and reduced jitter. At the same time, it places a severe constraint on the task period assignment for any application. In this paper, we explore the use of harmonic task scheduling for applications with multiple feedback control tasks. We present algorithm for finding harmonic task periods: to minimizes the distance from an initial set of non-harmonic periods. We apply the algorithm in a scheduling and control co-design procedure, where the goal is to optimize the total performance of a number of control tasks that share a common computing platform. The procedure is evaluated in simulated randomized examples, where it is shown that, in general, harmonic scheduling combined with release offsets gives better control performance than standard, non-harmonic scheduling.**

I. Introduction

Industrial controllers are often implemented as periodic tasks executing under the control of a real-time operating system using fixed-priority scheduling. Furthermore, for cost-saving reasons it is not uncommon having multiple controllers sharing the same execution platform. This leads to the problem of scheduling and control co-design. Most controllers are designed assuming a constant sampling period and a negligible or constant input-output latency, also referred to as control delay, or simply delay. Due to the interference that high-priority tasks impose on lower-priority tasks these assumptions do not always hold. The key problem in scheduling and control co-design is to optimize the combined performance of all the control tasks in the systems, subject to a schedulability constraint [1]. This is done by assigning suitable task parameters and designing feedback controllers that take the resulting task schedules into account.

In our previous work [2] the optimal task period assignment problem was solved assuming controller cost functions that depended linearly on the period and the delay, and where the delay was assumed to be constant and estimated using approximative response-time analysis. More recently, in [3] we instead modeled the delay by the statistical distribution of the task response time, and an optimization-based approach was used to find the task periods for a set of controllers designed using stochastic linear-quadratic-Gaussian (LQG) control design techniques. The response-time distributions were found by simulating the task schedule. In the follow-up paper [4] our approach was instead to perturb the task periods slightly in order to obtain a finite hyperperiod, and, hence, a periodic delay pattern in the control loops. This periodicity was then explicitly accounted for in the control design, by applying periodic stochastic LQG control design.

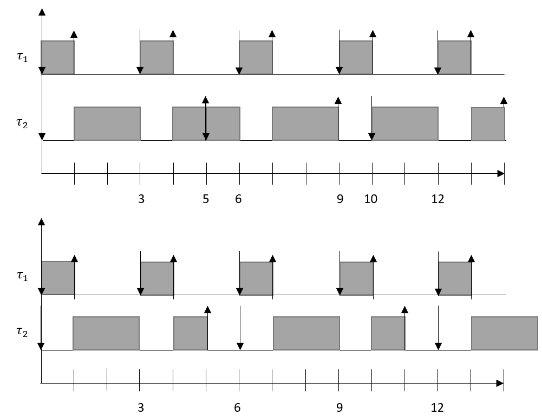

Fig. 1. Two tasks with non-harmonic periods (upper plot) and harmonic periods (lower plot). The job arrivals are shown with down-arrows and the job finishing times with up-arrows.

In the current paper the focus is again on perturbing the task periods, but in this case to make the task periods harmonic. Harmonic task sets have many attractive properties. As long as the total utilization is less than or equal to 1, the task set is schedulable under both rate-monotonic (RM) and earliest-deadline-first (EDF) scheduling. Also, assuming constant execution times, the response times and start latencies are constant—something that leads to a particularly simple LQG control design. Consider the simple example shown in Fig. 1. In both cases we have two control tasks running under RM priority assignment, where the sampling is performed at the job arrival times and the actuation is performed at the job finishing times, i.e., the control delay equals the response times. In the upper plot the tasks have periods $\{3, 5\}$ and constant execution times $\{1, 3\}$. We assume that these periods have been chosen to give good total control performance. From the figure one can see that the response time of task τ_2 varies according to the pattern $5, 4, 4, 5, 4, 4 \ldots$. If one changes the period of task τ_2 to 6, i.e., harmonize the periods, then, as shown in the lower plot, the response time will be always equal to 5, i.e., more deterministic than in the first case. This will most likely lead to worse control performance since both the period and the average delay are larger than before. However, if one also introduces an offset of 1 for τ_2 then the response time will always be equal to 4. The question is then whether the decrease in performance of control task τ_2 caused by the longer sampling period is compensated for by the increased performance caused by the shorter and constant delay obtained through the period harmonization. This is the essence of the problem that we are investigating in this paper, and the evaluations performed show that this is generally the case.

In the paper we present some new results on task period harmonization and scheduling analysis for harmonic tasks, with the twin goals of simplifying the control design and improving the performance of multi-loop control systems. The specific contributions of the paper are as follows:

- We give an algorithm for calculating the closest harmonic task periods (in the Euclidean sense) to a set of initial periods, under full utilization.
- We present a new scheduling and control co-design method based on the results above. The method strives to optimize the overall LQG control performance of a set of control tasks that share the same CPU. The new method is compared to previous work in simulated randomized examples.

Outline of the Paper

In Section II, related work is presented. In Section III, we give the basic scheduling and control system models, including the metric we use to evaluate control performance. The algorithm to harmonize a non-harmonic task set is presented in Section IV. In Section V, the harmonic task period assignment is evaluated with and without task offsets and is compared to non-harmonic period assignment. Finally, in Section VI, some concluding remarks are given.

II. Related Work

The scheduling and control co-design problem was first formulated in the seminal paper [1]. A cost function, which was a function of the task period, was introduced for each controller, and the design goal was to select periods such that the total control cost was minimized while keeping the task set schedulable. The paper [5] analyzed how the cost function of an LQG controller depends on the sampling period and proposed an on-line period adjustment algorithm.

Delay and jitter due to task scheduling can have a great impact on control performance [6], [7]. In [2] an integrated design method was proposed, where both periods and delays were taken into account when assigning control task periods. The delay was—at design time—assumed to be constant and was found through an approximate response-time analysis. In [3], a stochastic LQG scheduling and control co-design method was proposed, in which controller response times were treated as independent random variables. However, the response-time of a periodic task actually appears in a periodic pattern. Hence, in [4], a periodic stochastic LQG control design method was proposed that takes the response-time distribution of each job in a hyperperiod into account. The resulting controller has time-varying parameters and its design is quite involved.

Harmonic task periods have some potential benefits, both for scheduling analysis and for control system design [8], [9]. The hyperperiod of a harmonic task set is finite and small [10], [11]. RM scheduling of harmonic tasks is feasible if the utilization is less than or equal to 1, e.g., [12]. [13] gives an exact schedulability test for harmonic real-time tasks with integer-valued parameters, which can be checked in polynomial time. The papers [14], [15] proposed two efficient methods to assign harmonic periods to real-time tasks with period ranges: the forward approach in [14] and the backward approach in [15].

III. Real-Time and Control System Models

A. Real Time System Model

We consider a real-time system, which is implemented by n tasks, running on a single CPU under preemptive RM or EDF scheduling. The ith task, denoted by τ_i, is defined by the 4-tuple (C_i, O_i, T_i^l, T_i^u), which is defined as follows:

- The *execution time* C_i is the length of time the task τ_i takes to execute. In this paper we will assume that this is constant. For simple control algorithms, including LQG controllers, executing on single core with simplistic memory hierarchies, this assumption is reasonably realistic.
- The *offset* O_i is the instant at which the first job of task τ_i is released. If no offset is specified, then $O_i = 0$ is assumed.
- The *period* T_i is the constant time between the release of two consecutive jobs of task τ_i. It can be chosen from the allowed time interval $[T_i^l, T_i^u]$.

In the RM scheduling case, the task *priority* is assumed to be implicitly assigned by the task ordering. A smaller task index i value means higher priority. The task ordering under both RM and EDF scheduling is decided by the periods, so that a task with a shorter period has a smaller task index i. If two or more tasks have the same period, then the order among them can be chosen arbitrarily.

For the given parameters mentioned above, the following characteristics can be calculated:

- The *response time* R_{ij} is the time that elapses from the release time to the finish time of the jth job of task τ_i.
- The *start latency* S_{ij} is the time that elapses from the release time to the start time of the jth job of task τ_i.
- The task *utilization* $U_i = C_i/T_i$ measures the amount of computational resources required by the controller. We denote the *total utilization* of all control tasks by $U = \sum_i^n U_i$, which should be less than or equal to 1.

B. Control Problem Formulation

For each task τ_i, a linear time-invariant continuous-time plant is defined in state-space form as

$$\begin{aligned} \dot{x}_i(t) &= A_i x_i(t) + B_i u_i(t) + v_i(t) \\ y_i(t_k) &= C_i x_i(t_k) + e_i(t_k) \end{aligned} \tag{1}$$

where $x_i(t)$ is the state vector of the plant, $u_i(t)$ is the control input, $y_i(t)$ is the system output, and A_i, B_i, C_i are constant matrices. The disturbance $v_i(t)$ is continuous-time white noise, while the measurement noise $e_i(t)$ is discrete-time white noise.

For each plant, an LQG controller should be designed to minimize a quadratic cost function

$$J_i = \lim_{t\to\infty} \frac{1}{t} \mathrm{E} \int_0^t \left(x_i^T Q_{1ci} x_i + u_i^T Q_{2ci} u_i \right) \, \mathrm{d}\tau \qquad (2)$$

where Q_{1ci} and Q_{2ci} are symmetric positive definite weighting matrices that penalizes state deviations and the control signal effort. The LQG controller is designed off-line, taking information about the expected delay (whether constant or random) into account. The control design gives rise to a linear controller with constant parameters.

IV. Finding Harmonic Control Task Periods

In control–scheduling co-design, it is typically assumed that the period of each control task can be chosen as a real value within a (possibly infinite) period range. In a prototypical problem formulation, the performance of each controller is described by a cost function $J(T)$, which is assumed to be an increasing function of the task period T, i.e., the lower the cost, the better the performance will be. The goal is to optimize the combined performance of all control tasks subject to a utilization constraint U_b, e.g.,

$$\underset{T_1,\ldots,T_n}{\text{minimize}} \; J = \sum_{i=1}^{n} J(T_i), \quad \text{subject to } \sum \frac{C_i}{T_i} \le U_b$$

If the function $J(T^{-1})$ is convex, efficient numerical methods are available to find the global optimum [5]. We showed in [16] that the harmonic period set

$$T = MC \qquad (3)$$

with $T = \begin{bmatrix} T_1 & T_2 & \cdots & T_n \end{bmatrix}^T$ and with M being the reciprocal matrix given by

$$M = \begin{bmatrix} 1 \\ m_1 \\ \vdots \\ m_1 m_2 \cdots m_{n-1} \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{m_1} & \cdots & \frac{1}{m_1 m_2 \cdots m_{n-1}} \end{bmatrix}$$

Here, we are interested in solving a similar co-design problem, but we want to restrict the possible task periods to be harmonic. However, optimization problems involving integers (in our case the harmonic factors $m_1, \ldots, m_{n-1}$) are in general NP-hard [17], meaning that the optimal harmonic control task period assignment problem cannot be solved efficiently. Hence, we propose the following heuristic approach to harmonic control task period assignment: finding the closest harmonic period assignment to a set of initial periods. All feasible candidate solutions are then evaluated with regards to the total control performance and the best solution is chosen.

We assume that a set of full-utilization initial non-harmonic task periods are given as $T^0 = \begin{bmatrix} T_1^0 & T_2^0 & \cdots & T_n^0 \end{bmatrix}^T$. The problem is then to find a set of harmonic periods that minimizes the Euclidean distance between this set and the initial periods:

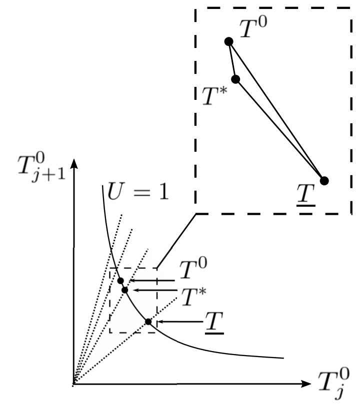

Fig. 2. Finding the closest periods set to the initial periods set.

$$\underset{T_1,\ldots,T_n}{\text{minimize}} \; \left\| T - T^0 \right\|, \quad \text{subject to } \sum_{i=1}^{n} \frac{C_i}{T_i} = 1,$$

$$\frac{T_{k+1}}{T_k} \in \mathbb{N}^+, \; k \in \{1, 2, \cdots, n-1\}$$

Theorem 1. *Let*

$$T^* = \begin{bmatrix} T_1^* & T_2^* & \cdots & T_n^* \end{bmatrix}^T$$

be the solution of the above optimization problem. Then $T^* \in \{MC\}$, where M is defined in Eq. (3), $m_i \in \left\{ \left\lfloor \frac{T_{i+1}^0}{T_i^0} \right\rfloor, \left\lceil \frac{T_{i+1}^0}{T_i^0} \right\rceil \right\}$.

Proof. Let $f : \mathbb{R}^{n-1} \to \mathbb{R}^n$ be defined as

$$T = MC = f([m_1 \; m_2 \; \ldots \; m_{n-1}])$$

where the matrix M is given in Eq. (3), and let the initial harmonic periods be $T^* = f(m^*_{\text{vector}})$, in which $m^*_{\text{vector}} = \begin{bmatrix} m_1^* \; m_2^* \; \ldots \; m_{n-1}^* \end{bmatrix}$. Further define

$$\overline{m}_{\text{vector},j} = \begin{bmatrix} m_1^* \; m_2^* \; \ldots \; \left\lceil \frac{T_{j+1}^0}{T_j^0} \right\rceil + 1 \; \ldots \; m_{n-1}^* \end{bmatrix}$$

$$\underline{m}_{\text{vector},j} = \begin{bmatrix} m_1^* \; m_2^* \; \ldots \; \left\lfloor \frac{T_{j+1}^0}{T_j^0} \right\rfloor - 1 \; \ldots \; m_{n-1}^* \end{bmatrix}$$

Let $\underline{T} = f(\underline{m}_{\text{vector},j})$. Now consider Fig. 2 where the curve represents the utilization bound. Since this curve is convex, in the triangle $T^* T^0 \underline{T}$, the angle between $T^* T^0$ and $T^* \underline{T}$ is greater than $90°$. Then

$$\left\| T - T^0 \right\| < \left\| f(\underline{m}_{\text{vector},j}) - T^0 \right\|$$

and, similarly,

$$\left\| T - T^0 \right\| < \left\| f(\overline{m}_{\text{vector},j}) - T^0 \right\|$$

The two inequalities above also apply for the high-dimensional case. By choosing different values of m_i, one can show that for each $i \leq n-1$, the two inequalities are valid. In the $n-$dimensional case, we need to check 2^{n-1} inequalities. $\square$

V. Co-Design and Evaluation

In this section we apply harmonic period assignment in control–scheduling co-design and compare the resulting performance to state-of-the-art non-harmonic co-design.

A. Co-Design Procedure

As a starting point for the co-design, we find a set of non-harmonic, real-valued task periods and corresponding controllers using the sequential search optimization method in [3], which in turn is initialized using the method in [2]. We then harmonize these periods using Theorem 2 or Theorem 3 and enumerate all possible combinations of the harmonic factors. For each harmonic period assignment, we redesign each controller based on the new period and the new (now constant) control delay. Finally, we evaluate the combined control performance of all cases and pick the best result.

All controllers are designed using the `lqgdesign` command in Jitterbug [7]. To make a fair comparison between harmonic and non-harmonic designs, we make the following assumptions for the LQG control design:

- The harmonic control design takes the constant delay into account.
- The non-harmonic control design takes the delay distribution due to task scheduling into account, resulting in a jitter-robust controller with fixed parameters [3]. The delay distribution is found through a schedule simulation in TrueTime [7].
- The target CPU utilization is 1 for both harmonic scheduling and non-harmonic scheduling. Under non-harmonic scheduling, if the response time variability is greater than the period, then the response time distribution used in the control design is truncated to the period length.

The cost function (2) for each controller under each scheduling scenario is evaluated using the TrueTime toolbox [7]. Using TrueTime it is possible to simulate real-time kernels with tasks executing, e.g., controller, code under the control of an arbitrary scheduling policy and interacting with continuous-time dynamic models representing the physical process under control. It is also possible to numerically evaluate the same quadratic cost functions that are evaluated analytically using Jitterbug. However, using TrueTime one is not, as in Jitterbug, restricted to scenarios in which the delays are independent, e.g., in the non-harmonic controller evaluations.

The design and evaluation procedure is summarized in the flow diagram in Fig. 3.

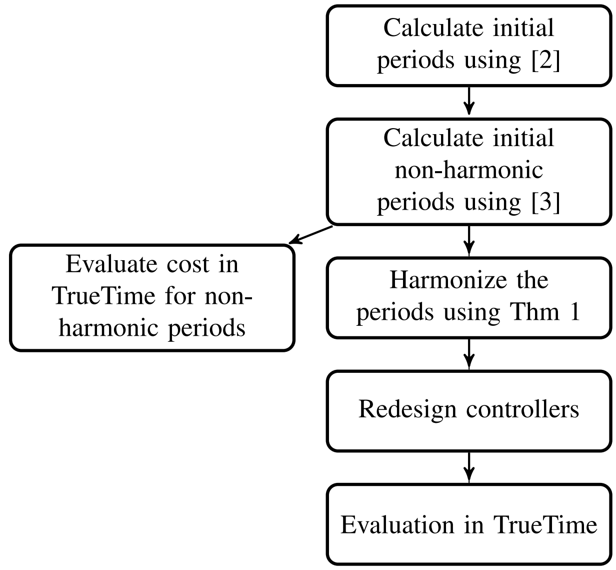

Fig. 3. Co-design and evaluation procedure for non-harmonic and harmonic designs.

B. A Simple Co-Design Example

For a simple co-design example, we choose three plants

$$P_1 = \frac{2}{s^2}, \quad P_2 = \frac{1}{s^2-3}, \quad P_3 = \frac{1}{s(s+1)}$$

to be controlled by three tasks τ_1, τ_2, τ_3. τ_1 has the highest priority and τ_3 has the lowest priority. The execution times are given as $C_1 = 0.1$, $C_2 = 0.12$, $C_3 = 0.14$. The LQG cost function parameters are given as $Q_{1c} = C^T C$, and $Q_{2c} = 0.01\mathrm{tr}\,(Q_{1c})$.

Initial non-harmonic periods are calculated by the following initialization procedure (cf. [3]):

1) For task i, assume that the LQG cost can be approximated by a linear function of the period T_i and of the delay L_i,
$$J_i = \alpha_i T_i + \beta_i L_i$$
Evaluate the sensitivity coefficients α_i and β_i at the point $T_i = C_i$, $L_i = C_i$ using numerical linearization and Jitterbug. Then use the period assignment method in [2] to minimize the LQG cost under the simplifying assumption that these are the true cost functions.
2) Use the Sequential Search method and stochastic LQG design method in [3] to find the non-harmonic periods.

The initial non-harmonic periods are $T_1^* = 0.3017$, $T_2^* = 0.4089$, $T_3^* = 0.4478$ with initial cost $J^* = 2.01$. The LQG cost is then evaluated in TrueTime for the following cases:

- **No offset.** Assume a constant delay equal to the job response time R_i to design and evaluate the LQG controllers.
- **Offset.** Add the start latency S_i as a release offset to each task; then design and evaluate the LQG controllers for the constant delay $R_i - S_i$.

The resulting periods and LQG costs are shown in the table below. No offset means sampling happens at job release time, while offset means that sampling happens at the job start.

$\{m_1, m_2\}$	T_1	T_2	T_3	$J_{\text{no offset}}$	J_{offset}
$\{1,1\}$	0.36	0.36	0.36	1.99	1.57
$\{1,2\}$	0.29	0.29	0.58	2.11	1.57
$\{2,1\}$	0.23	0.46	0.46	1.90	1.33
$\{2,2\}$	0.20	0.39	0.78	2.56	1.75

The result shows that when we approximate the initial non-harmonic task periods with harmonic ones, the costs can be better or worse than the initial cost. However, when we utilize release offsets, the cost is clearly better than the initial cost in all the cases, with the best case obtained for $\{m_1, m_2\} = \{2, 1\}$.

Continuing the same example, we also show how to find harmonic periods when there are constraints on the allowable period ranges, and then evaluate the LQG control performance, using the same plants with the same execution times as before. Assuming that the period T_i can only be chosen from $[0.6T_i^*, 1.7T_i^*]$, it follows from Theorem 3 that the possible periods are when $\{m_1, m_2\}$ is $\{1,1\}$, $\{1,2\}$, $\{2,1\}$ (as shown above), or $\{3,1\}$ (as shown below).

$\{m_1, m_2\}$	T_1	T_2	T_3	$J_{\text{no offset}}$	J_{offset}
$\{3,1\}$	0.19	0.56	0.56	3.36	2.25

It should be noted that the global optimal harmonic period assignment with the lowest total control cost is not necessarily restricted to the above cases. The control cost function could have a form so that the harmonic period assignment with the lowest cost is not among those assignments that are close to the initial non-harmonic periods in the Euclidean sense or within the ranges given by the sampling period rule of thumb. However, as shown in the general evaluation in the next section the proposed approach obtains gives considerably better control performance than the non-harmonic case.

C. Randomly Generated Example

To see if the good results shown in the simple example above holds in more general cases, we randomly generate sets of three plants from the following three plant families:

- Family I: All plants have two stable poles and are drawn from $P_1(s)$ and $P_2(s)$ with equal probability where
$$P_1(s) = \frac{1}{(s+a_1)(s+a_2)}, \quad P_2(s) = \frac{1}{s^2 + 2\zeta\omega s + \omega^2}$$
with $a_1, a_2, \zeta \in \text{unif}(0,1)$, $\omega \in \text{unif}(0,1)$.
- Family II: All plants have two stable or unstable poles, with each plant drawn with equal probability from
$$P_3(s) = \frac{1}{(s+a_1)(s+a_2)}, \quad P_4(s) = \frac{1}{s^2 + 2\zeta\omega s + \omega^2}$$
with $a_1, a_2, \zeta \in \text{unif}(-1,1)$, $\omega \in \text{unif}(0,1)$.
- Family III: All plants have three stable or unstable poles, with each plant drawn with equal probability from
$$P_5(s) = \frac{1}{(s+a_1)(s+a_2)(s+a_3)}$$
$$P_6(s) = \frac{1}{(s^2 + 2\zeta\omega s + \omega^2)(s+a_3)}$$
with $a_1, a_2, a_3, \zeta \in \text{unif}(-1,1)$, $\omega \in \text{unif}(0,1)$.

We randomly generated 20 sets of plants for each family. For the LQG controllers, we used the design parameters $Q_{1c} = C^T C$, and $Q_{2c} = 0.01\text{tr}(Q_{1c})$. The task execution times were randomly generated from $C_1 \in \text{unif}(0.09, 0.11)$, $C_2 \in \text{unif}(0.11, 0.13)$, $C_3 \in \text{unif}(0.13, 0.15)$. Task 1 has the highest priority, while task 3 has the lowest priority.

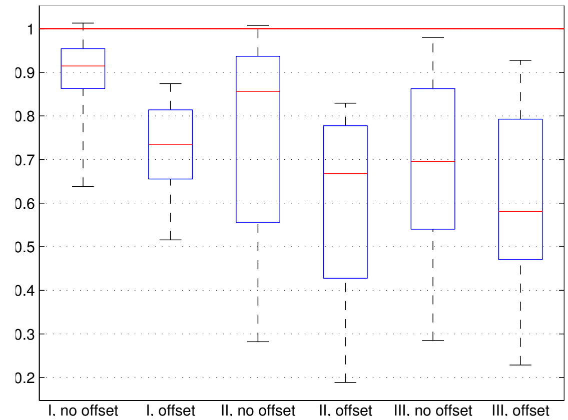

Fig. 4. Normalized costs for harmonic tasks without and with offsets

The optimization procedure to assign initial non-harmonic periods is the same as in the previous section. We find the four closest harmonic period sets to the initial periods using Theorem 1. For the harmonic periods case, the response time of each task is constant. Using this constant response time as delay, the LQG controllers are designed and the corresponding costs are evaluated. In the table below, the minimum cost, out of the four cases with harmonic periods, is given. We then add an offset to each task in order to obtain a shorter delay for task 2 and 3. The length of the offset is the start latency of each task. The constant delay is $R_i - S_i$. The we design controller and evaluate the LQG costs for this constant delay. The overall results, averaged over 20 generated plant sets for each family, are summarized below.

Family	I	II	III
non-harmonic tasks	2.92	8.61	29.46
harmonic tasks	2.53	5.17	17.76
harmonic tasks with offsets	2.03	3.91	15.43

In the above table, we design the LQG controllers and evaluate the LQG costs as follows:

- **Non-harmonic tasks.** The delay distribution is truncated to the interval $\left[R_i^{\text{best}}, R_i^{\text{best}} + T_i\right]$. The probability of a response time greater than $R_i^{\text{best}} + T_i$ is added to the probability mass function at $R_i^{\text{best}} + T_i$. We then use the truncated delay distribution to design stochastic LQG controllers. The LQG cost is evaluated in TrueTime.
- **Harmonic tasks.** We calculate 2^{n-1} sets of harmonic periods. For each set, LQG controllers with constant delays equal to R_i are designed and evaluated in TrueTime. The period set giving the smallest cost is selected.
- **Harmonic tasks with offsets.** For each set of harmonic periods, LQG controllers with constant delays equal to $R_i - S_i$ are designed and evaluated in TrueTime. The period set giving the smallest cost is selected.

We normalize the costs for non-harmonic tasks to 1 for each plant set, then normalize each cost for harmonic tasks without

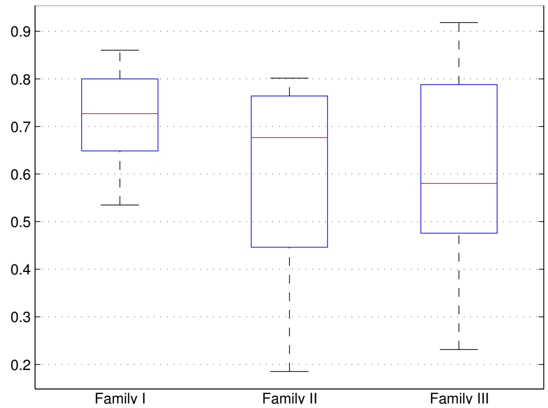

Fig. 5. Normalized costs for non-constant execution times (note different scale than Fig. 4)

or with offsets, compared with corresponding non-harmonic tasks cost. The box plot is shown in Fig. 4.

In Family III, the likelihood that the plants are unstable, and, hence, more sensitive to delays and delay jitter, is larger, and therefore the control cost is considerably higher than for Family I and II. The best results are obtained for the harmonic tasks with offsets. In this case the increase in cost caused by the period perturbation is small compared to the decrease in cost caused by the smaller and jitter-free delays.

The evaluation above is based on the assumption that the execution time are constant. However, in reality this is seldom the case. To investigate the effect of varying execution times we design the controllers using the harmonic task with offset method assuming that the execution times are constant. When we evaluate the performance we let the execution time vary from job to job according to Unif$(0.9C_i, C_i)$. The costs now become

Family	I	II	III
Cost	2.01	3.88	15.33

I.e., the costs are even smaller than before. We also make a box plot of costs for non-constant execution times compared with non-harmonic tasks costs in Fig. 5. This is, however, not so surprising since the average delay is shorter than the constant delay in the constant execution time case.

VI. Conclusion

In this paper we have investigated the harmonic scheduling and control co-design problem. Through an extensive evaluation it was shown that co-design using harmonic control task periods gives better control performance than using non-harmonic periods, when task offsets are added. The reason for this is the fact that under harmonic scheduling the response times and start latencies for each task are constant, assuming that the task execution times are constant. This can be exploited in the LQG control design through the constant control delays that it gives rise to. However, as shown in the evaluation also in the case when the task execution times vary slightly from job to job the proposed co-design method gives good results.

In order to implement the co-design method a heuristic approach to harmonic task period assignment has been presented. The method is used to find the closest harmonic periods to a set of initial periods.

Acknowledgments

The authors would like to thank the LCCC Linnaeus Center and the ELLIIT Excellence Center at Linköping and Lund.

References

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

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

[3] Y. Xu, K.-E. Årzén, E. Bini, and A. Cervin, "Response time driven design of control systems," in *Proceedings of the* 19^{th} *World Congress of the International Federation of Automatic Control*, Cape Town, South Africa, Aug. 2014, pp. 6098–6104.

[4] Y. Xu, K.-E. Årzén, A. Cervin, E. Bini, and B. Tanasa, "Exploiting job response-time information in the co-design of real-time control systems," in *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2015 IEEE 21st International Conference on*, Aug 2015, pp. 247–256.

[5] J. Eker, P. Hagander, and K.-E. Årzén, "A feedback scheduler for real-time controller tasks," *Control Engineering Practice*, vol. 8, no. 12, 2000.

[6] B. Wittenmark, J. Nilsson, and M. Törngren, "Timing problems in real-time control systems," in *In Proceedings of the American Control Conference*. Citeseer, 1995.

[7] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Årzén, "How does control timing affect performance?" *IEEE Control Systems Magazine*, vol. 23, no. 3, pp. 16–30, Jun. 2003.

[8] J. V. Busquets-Mataix, J. J. Serrano, R. Ors, P. Gil, and A. Wellings, "Using harmonic task-sets to increase the schedulable utilization of cache-based preemptive real-time systems," in *Real-Time Computing Systems and Applications, 1996. Proceedings., Third International Workshop on*. IEEE, 1996, pp. 195–202.

[9] C.-S. Shih, S. Gopalakrishnan, P. Ganti, M. Caccamo, and L. Sha, "Scheduling real-time dwells using tasks with synthetic periods," in *Real-Time Systems Symposium, 2003. RTSS 2003. 24th IEEE*. IEEE, 2003, pp. 210–219.

[10] I. Ripoll and R. Ballester-Ripoll, "Period selection for minimal hyperperiod in periodic task systems," *Computers, IEEE Transactions on*, vol. 62, no. 9, pp. 1813–1822, 2013.

[11] V. Brocal, P. Balbastre, R. Ballester, and I. Ripoll, "Task period selection to minimize hyperperiod," in *Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference on*. IEEE, 2011, pp. 1–4.

[12] C.-C. Han and H.-y. Tyan, "A better polynomial-time schedulability test for real-time fixed-priority scheduling algorithm," in *Proceedings of the* 18^{th} *IEEE Real-Time Systems Symposium*, San Francisco, (CA), U.S.A., Dec. 1997, pp. 36–45.

[13] V. Bonifaci, A. Marchetti-Spaccamela, N. Megow, and A. Wiese, "Polynomial-time exact schedulability tests for harmonic real-time tasks," in *Real-Time Systems Symposium (RTSS), 2013 IEEE 34th*, Dec 2013, pp. 236–245.

[14] M. Nasri, G. Fohler, and M. Kargahi, "A framework to construct customized harmonic periods for real-time systems," in *Real-Time Systems (ECRTS), 2014 26th Euromicro Conference on*, July 2014, pp. 211–220.

[15] M. Nasri and G. Fohler, "An efficient method for assigning harmonic periods to hard real-time tasks with period ranges," in *Real-Time Systems (ECRTS), 2015 27th Euromicro Conference on*, July 2015, pp. 149–159.

[16] Y. Xu, K.-E. Årzén, and A. Cervin, "Lqg-based scheduling and control co-design using harmonic task periods," Department of Automatic Control, Lund University, Sweden, Technical Report TFRT-7646, Jun. 2016.

[17] C. H. Papadimitriou, "On the complexity of integer programming," *Journal of the ACM (JACM)*, vol. 28, no. 4, pp. 765–768, 1981.