Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Control Performance Evaluation of Selected Methods of Feedback Scheduling of Real-time Control Tasks [⋆]

Camilo Lozoya, Pau Martí, Manel Velasco, and
Josep M. Fuertes

Automatic Control Department, Technical University of Catalonia, Pau Gargallo 5, 08028 Barcelona, Spain (Tel: +34 93 8967798; e-mails: {camilo.lozoya,pau.marti,manel.velasco,josep.m.fuertes}@upc.edu).

Abstract: Feedback scheduling (FS) often refers to the problem of sampling period selection for real-time control tasks that compete for limited computing resources such as processor, network, or battery power. Its goal is to optimize the aggregated control performance achieved by all tasks by efficiently using the scarce resources. In this paper representative existing FS methods are selected, their main features are identified, and a simple control performance evaluation is performed. The latter shows that a) jitters in job executions hide the true performance that can be achieved by the analyzed FS methods, b) after completely removing the degrading effects of jitters, the performance of each FS method dramatically changes, and c) the relative benefit provided by each method depends on the type of perturbations affecting the plants.

Keywords: Computer controlled systems; Real-time computer systems; Sampling periods; Resource allocation; Performance evaluation.

1. INTRODUCTION

Traditionally, real-time computer controlled systems are implemented using hard real-time periodic tasks [Årzén et al., 2000]. Each periodic task is statically assigned a constant execution rate. However, the embedded systems market requires systems with more and better functionalities at lower prices. A consequence is that control applications must be implemented in platforms where resources are scarce and/or where increasing performance is a must [Buttazzo, 2006]. And the traditional static periodic approach to control systems implementation fails at minimizing resource utilization and maximizing control performance.

To provide solutions fulfilling the tight demands posed by modern embedded systems, approaches to control performance and resource optimization for adaptive real-time embedded control systems have been receiving increased attention. Such approaches focus on different methods for selecting sampling periods for real-time control tasks, which determine resource utilization (or alternatively, task set schedulability) as well as overall control performance.

Recently, a taxonomy on sampling period selection for resource-constrained real-time control tasks was presented by Lozoya et al. [2007]. Two main tendencies are identified: (1) Feedback Scheduling (FS) and (2) Event-based Scheduling (ES). The main difference between them is that FS primarily looks at the problem of optimizing control performance by fully exploiting the available resources. On the contrary, ES looks at the problem of minimizing resource utilization while ensuring system stability or bounding the inter-sampling dynamics. This paper presents a control performance evaluation of representative FS approaches whose solutions are based on different optimization criteria.

The evaluation reveals a key aspect: jitters in job executions hide the true performance that can be achieved by the analyzed FS methods. Real-time scheduling introduces timing uncertainty in each task job start and finishing time. In the standard practice of real-time implementation of control loops where sampling and actuation occurs at the beginning and end of each job execution, the timing uncertainty produces sampling jitter and latency jitter, which deteriorate control performance [Årzén et al., 2000]. With the presence of jitters, the performance analysis coincides with the main conclusions drawn in the evaluation presented by Cervin et al. [2006].

However, after removing jitters, the performance evaluation shows completely different results because the performance of each FS method dramatically changes. The evaluation then permits to identify where performance improvements come form, and it shows that the relative benefit provided by each FS method depends on the type of perturbations affecting the plants.

The rest of the paper is structured as follows. Section 2 reviews the basics on feedback scheduling. Section 3 summarizes the evaluated methods. Section 4 and 5 details the simulation set-up and the customization of the evaluated methods for fair comparison. Section 6 presents the performance evaluation and Section 7 concludes the paper.

[⋆] This work was supported in part by ARTIST2 NoE on Embedded Systems Design IST-2004-004527, and by CICYT Control, Communications and Real-time Computing in Distributed Embedded Nodes DIP2007-61527.

978-1-1234-7890-2/08/$20.00 © 2008 IFAC

10.3182/20080706-5-KR-1001.1995

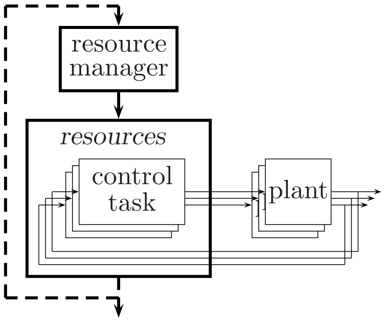

Fig. 1. Feedback scheduling scheme

2. PRELIMINARIES ON FEEDBACK SCHEDULING

Feedback scheduling refers to the problem of sampling period selection for real-time control tasks that compete for limited computing resources such as processor, network, or battery. Its goal is to optimize the aggregated control performance achieved by all tasks by efficiently using the scarce resources.

The logics of the standard feedback scheduling architecture are shown in Fig. 1. Several control tasks in charge of controlling plants compete for limited *resources*. The resource manager implements the solution provided by each FS method, that is, it dictates how resources are assigned to each control task in order to optimize overall control performance.

The allocation of resources is commonly formulated as a constrained optimization problem, as follows:

minimize (maximize):	penalty (benefit) on control performance
with respect to:	sampling periods or job sequence execution
subject to:	closed loop stability and task set schedulability

The objective function of the optimization problem relates control performance and resource utilization, the later usually in terms of the sampling periods (task periods) or frequencies. The optimization variables usually are the set of sampling periods to be assigned to all control tasks.

The optimization problem is constrained by two key aspects. The set of optimal sampling periods must guarantee closed loop stability and task set schedulability. Stability is either guaranteed by the formulation of the optimization problem, or it is not explicitly imposed in the formulation but analyzed after solving the optimization problem. Task set schedulability is often imposed by resource utilization tests. Additional constraints can be added.

A few methods, instead of providing optimal sampling periods, provide job sequences. That is, the outcome of the optimization problem is an optimal sequence of jobs for each control task to be executed periodically.

3. EVALUATED FS METHODS

The main differences between the existing methods depend on the following aspects:

- When to solve the optimization problem: off-line or on-line.
- Which kind of dynamics, if any, is accounted for in the optimization problem: resource and plant.
- What type of solution provides the optimization problem: sampling periods or job sequences.

The following subsections present the evaluated methods, and characterizes them according to the above aspects. Although not exhaustive, they are representative of the existing research (see [Lozoya et al., 2007] for more details).

3.1 Static Approach

This is the only approach that does not belong to the class of feedback scheduling methods but it is here included for comparative purposes. It implements the traditional approach to real-time implementation of computer controlled systems. That is, each control task is assigned offline an *arbitrary* sampling period selected according to well established procedures [Åström et al., 1997], taking also into account task set utilization.

After selecting the periods, control tasks are scheduled under Earliest Deadline First (EDF) scheduling [Liu et al., 1973]. This algorithm admits the highest processor utilization (100%) while guaranteeing task set schedulability. Therefore, no off-line or on-line optimization is performed.

3.2 Off-line FS

The off-line FS is represented by the work by Seto et al. [1996] which can be considered one of the seminal papers on sampling period selection subject to control performance optimization for real-time control systems.

The objective function describes the *a priori* relation between a control performance index expressed in terms of cost and a range of sampling frequencies. This relation is approximated by a decreasing exponential function. After guaranteeing a maximum feasible period to each control task, an off-line optimization procedure re-scales periods until the task set is feasible under EDF while minimizing the cost. Therefore, off-line optimization is performed, and once periods are set, control tasks are schedule under EDF.

3.3 On-line Resource Aware FS

The method presented by Eker et al. [2000] is the first one that uses the term *feedback scheduler*. The key aspect is to on-line adjust sampling periods considering the dynamics of the processor load. Looking at the outer loop of Fig. 1, the resource manager is the feedback scheduler that, having available the system workload from the real-time kernel, i.e. *resource aware* (RA) and given a utilization setpoint, keeps the desired utilization by modifying workload via on-line sampling period selection, while optimizing the total control performance.

After deriving how control performance depends on sampling periods in terms of standard optimal control cost functions, the method presents a recursive resource allocation optimization routine. Due to computational overhead, for the case of linear plants and assuming no actuators'

Approach	Opt.	When	Dynamics	PI
Static approach	No			
Seto et al. [1996]	Yes	Off		
Eker et al. [2000]	Yes	On	kernel	
Martí et al. [2004]	Yes	On	kernel/plant	Inst.
Castañé et al. [2006]	Yes	On	kernel/plant	FH
Ben Gaid et al. [2006]	Yes	On	kernel/plant	FH

Table 1. Key features of evaluated FS methods.

saturation cost functions are approximated by quadratic functions of the sampling period, which permits to derive explicit solutions to the optimization problem. Therefore, using these solutions, sampling periods can be derived on-line, and tasks scheduled for example by EDF.

3.4 On-line Resource and Control Aware FS

A step further is to optimize total control performance by on-line adjusting sampling periods according to both kernel workload and plants' dynamics, i.e., *resource and control aware* (RCA). The intuitive idea behind this kind of approaches is to provide more processing capacity to control tasks whose plants are experiencing severe transients due to e.g. perturbations or noise. They can be schematically understood looking at the outer loop of Fig. 1 and considering that the resource manager is fed back with information concerning both resource utilization and plant dynamics.

Representative methods are given for example by Martí et al. [2004] and Castañé et al. [2006]. Both approaches assume that the feedback scheduler has available the state of all controlled plants when solving the optimization procedure. Depending on how states are treated in the optimization, two flavors can be distinguished:

- *Instantaneous* (Inst.): the current state is the only information of the plants that is considered in the optimization procedure. This is the approach adopted by Martí et al. [2004] where the on-line optimization procedure mandates that after assigning a maximum sampling period to each control task, the leftover processor time should be used to dynamically shorten the period of the control tasks whose plant is experiencing the highest error. The later, included in the objective function, is defined as a function of the plant current state.
- *Finite horizon* (FH): the current state is the initial condition for predicting over a finite horizon the future plants' dynamics. This is the approach adopted by Castañé et al. [2006], where the feedback scheduler executes at a predefined rate to optimally on-line adjust sampling periods according to current and predicted plants' transient dynamics.

3.5 On-line Cyclic Executive FS

The last evaluated method presented by Ben Gaid et al. [2006] represents a class of methods that differs from the on-line RCA FS one in the sense that the outcome of the optimization routine is not a set of sampling periods but sequences of jobs for the set of control tasks. These sequences of jobs are known as *cyclic executives* (CE) [Locke, 1992].

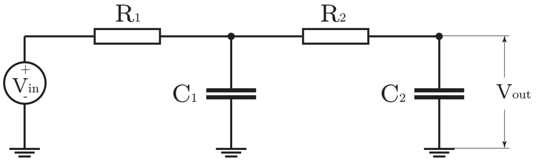

Fig. 2. Electronic circuit scheme

In particular, in [Ben Gaid et al., 2006] an off-line optimal scheduler with an on-line heuristic suboptimal routine that produces optimal sequences of jobs in terms of minimizing control cost is presented. The off-line scheduler computes off-line sequences using optimal control theory based on a prediction of the plant dynamics over a finite time horizon, and the on-line routine switches sequences at run-time according to the current dynamics.

3.6 Summary of key features

Table 1 presents a summary of the key features of each method under evaluation. Columns refer to a) whether optimization is carried out, b) when it is carried out (off-line or on-line), c) what dynamics (kernel workload and/or plant dynamics) are included in the optimization when it is performed on-line, and d) which type of performance index (instantaneous or finite horizon) is included in the objective function of the optimization procedure when plant dynamics are accounted for.

Although the last two rows are characterized by the same features, it is important to notice that the last approach does not provides periods for control tasks but optimal sequences of job executions.

4. SIMULATION SET-UP

The evaluation presented in this paper focuses on control performance and aims at identifying key features for each of the evaluated methods. Therefore, rather than being exhaustive, a simple but complete simulation set-up has been designed. Simulations have been carried out with the *Truetime* simulator [Henriksson et al., 2002] to implement the multitasking processor together with each FS strategy. The performance analysis targets to measure the impact of each strategy on the controlled plant dynamics.

4.1 Controlled plants

Three identical voltage stabilizers in the form of an RCRC circuit were used as controlled plants (Fig. 2). Each voltage stabilizer is controlled by a single control task. The three control tasks concurrently execute on a real-time kernel.

Taking into account that the electronic components values are $R_1 = R_2 = 330\text{k}\Omega$ and $C_1 = C_2 = 100\text{nF}$, a continuous state space description of the voltage stabilizer analyzed in terms of currents is given by

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -918.27 & -90.90 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 918.27 \end{bmatrix} u(t), \qquad (1)$$

where x_1 corresponds to the output voltage V_{out}, and x_2 is $\dot{q}_2/C_2$, being $\dot{q}_2$ the current at R_2.

For each voltage stabilizer, the control objective is to maintain stable the output voltage V_{out} according to a reference value of 2.4V, regardless of the load perturbations simulated as pulses affecting the output voltage.

4.2 Perturbations

For the simulation purposes, it is assumed that the three plants start in steady state at the reference value. Each plant is perturbed up to 30 times by randomly generated pulses. Each plant is affected by a single perturbation during each simulation period of $3s$. This is repeated 30 times, and the average value is reported. This permits to *control* the number of perturbations, and the time of their occurrence with respect to the simulation period.

Several evaluation scenarios, described below, have been designed. The first one, *random*, permits to obtain a general evaluation of all the methods. The following three, *independent*, *overlapped*, and *simultaneous*, permit to assess the FS methods according to the patterns of perturbations affecting the plants. The scenarios are:

- Random: one perturbation for each of the three plants can randomly occur during each simulation period.
- Independent: one perturbation for each of the the three plants occurs during each simulation period in such a way that only one plant is perturbed at a given time. Moreover, no perturbation will affect any of the two other plants until the perturbed plant reaches the steady state, which takes no longer than 0.5s. This is achieved by forcing one perturbation in a first plant between 0s and 0.5s, another perturbation in the second plant between 1s and 1.5s, and the last perturbation in the third plant between 2s and 2.5s. But the exact time instant that perturbations occur within these time intervals is randomly generated.
- Overlapped: one perturbation for each of the the three plants occurs during each simulation period in such a way that only one plant is perturbed at a given time, but overlapping between transients are forced. Therefore, after one perturbation affects the first plant, another perturbation affects the second plant before the first settles, and the last perturbation affects the third plant before the second plant settles. However, the exact time instant that perturbations occur following this pattern is randomly generated.
- Simultaneous: one perturbation for each of the three plants occurs during each simulation period in such a way that the three plant are perturbed at the same time. However, the exact time instant that perturbations occur is randomly generated.

Finally, the pulse magnitude is also randomly generated within a specified range of 0.4V to 0.8V.

4.3 Evaluation metric

The evaluation during each simulation period (t_{sim}) is performed using a continuous standard quadratic cost function

$$J = \int_{0}^{t_{sim}} \left[x^T(t)Qx(t) + u^T(t)Ru(t)\right] dt. \qquad (2)$$

Since the focus of the controller is to ensure fast tracking of the output variable, the weighting matrices are specified as

$$Q = \begin{bmatrix} 400 & 0 \\ 0 & 1 \end{bmatrix} \quad and \quad R = 1.$$

For the simulation purposes, it is assumed that plants' states are available and therefore there is no need for observers.

4.4 Controller design

The tracking is achieved by controller designed to place the continuous closed loop poles at $s_{1,2} = -103.93 \pm 87.1i$. Therefore, depending on the sampling period that applies, denoted by h, the continuous closed loop poles are mapped into the discrete-time domain, $z_{1,2} = e^{s_{1,1}h}$, and then the discrete-time controller is obtained by standard pole placement. This is either performed off-line or on-line, depending on the evaluated method.

5. TASKS AND PERIODS FOR EACH FS METHOD

The periods of the three control tasks are allowed to take values within 20ms and 40ms. For the simulation purposes, the execution time of each task is specified to be 10ms. Since the focus of this paper is evaluation of control performance, no changes in the kernel workload and utilization set-point have been injected. Therefore, for the on-line methods, task periods' changes are only based on the dynamics of the controlled plants, not on the kernel workload.

The specific tasks periods for each of the evaluated methods are described next:

- Static approach: periods for each task are heuristically selected as follows

task1	20ms
task2	40ms
task3	40ms

  and the corresponding controllers designed before run-time.
- Off-line FS [Seto et al., 1996]: since the three plants are equal, the optimization procedure mandates to execute each task with a period of 30ms. Therefore, the three tasks execute the same controller designed with this period.
- On-line RA FS [Eker et al., 2000]: for the sake of the comparative analysis, the initial periods for the three tasks are the same as in the static approach, and the on-line recursive optimization routing stabilizes the three periods. From the periods initial values to the final values, the controller that is used is redesigned at run-time according to the current period that applies.
- On-line RCA FS - Inst. [Martí et al., 2004]: the final outcome of the method mandates to consider at run-time only two periods, 20ms or 40ms. Tasks (and controller' gains) switch between these two periods whenever the plant with highest error changes.

- On-line RCA FS - FH [Castañé et al., 2006]: this method mandates to switch periods at run-time continuously within the specified range according to the optimization procedure. Switches of tasks periods (and controllers' gains) occur every 100ms, which is the period of the so-called feedback scheduler.
- On-line CE FS [Ben Gaid et al., 2006]: this method mandates to switch between sequences of jobs executions according the on-line heuristic routine. For the sake of the performance analysis, since the type of perturbations and the corresponding plants' deviations are known before running the simulations, specific heuristic job sequences have been designed off-line, and switched at run-time in such a way that control performance is maximized.

5.1 Control task model

All the evaluated approaches except the last one execute tasks scheduled by EDF. Therefore, each control task is implemented as a hard real-time periodic task, where sampling (input) and actuation (output) occurs at the beginning and at the end of each job execution . This model is identified by Årzén et al. [2000] as a common practice implementation of control loops using real time technology.

The last evaluated method uses a sort of cyclic executives rather than periodic tasks.

6. CONTROL PERFORMANCE EVALUATION

This section summarizes the performance evaluation. First, the evaluation for the *random* scenario is presented, and then the three specific scenarios *independent*, *overlapped*, and *simultaneous* are analyzed.

6.1 Random Perturbations

Table 2 summarizes the performance numbers of the evaluated methods when perturbations occur randomly. The lower the numbers, which were obtained by the cost function given in (2), the better the approach. The table shows two columns, named *single processor* and *independent processors*.

The single processor column refers to the simulation numbers obtained when the three control tasks execute under EDF sharing a single processor. Looking at the numbers, it can be observed that

- the performance achieved by the static approach is the worst. This is an expected result, here corroborated, because all the other approaches perform some sort of optimization. Therefore, their performance was expected to be better.
- the performance achieved by the last evaluated method is, by long, the best. This is a bit surprising because the job sequences were heuristically designed.

The later observation reveals a key feature of the simulations. All the methods except the last one are implemented using the control task model explained in section 5.1. The last one uses cyclic executives. A well-known property of the implementation of control loops using cyclic executives is that jitters in job executions can be easily accounted for in the controller design [Locke, 1992]. In other words, the degrading effects that jitters have on control performance disappear. Therefore, the simulation numbers of all the methods except the last are affected by the degrading effects of jitters, hiding the true performance that could be achieved, which is unfair.

Approach	Single proc.	Indep. proc.
Static approach	109.05	105.82
Off-line FS Seto et al. [1996]	100.58	96.59
On-line RA FS Eker et al. [2000]	99.92	96.74
On-line RCA FS - Inst. Martí et al. [2004]	90.63	64.41
On-line RCA FS - FH Castañé et al. [2006]	100.61	86.99
On-line CE FS Ben Gaid et al. [2006]	82.46	82.46

Table 2. Evaluation in terms of jitters.

The evaluation presented by Cervin et al. [2006] overcomes the jitter problem by imposing a one-sample delay in the methods affected by jitters. Although in the paper it is outlined that imposing a one-sample delay has also devastating effects in terms of degrading control performance, [Cervin et al., 2006] conclude that FS methods based on cyclic executives provide the best performance, as it also could be concluded looking at the first column of Table 2.

To overcome the limitations of imposing a one-sample delay, in this paper a different approach is taken. The FS methods are evaluated as if their tasks were executing in isolation, evaluation that is reported in the second column of Table 2, named independent processors.

That is, the numbers reported in the second column correspond to the same simulations but each task is considered to be executing on a dedicated processor. Therefore, all tasks are assigned the same periods or sequences given by each FS method, as in case reported in the first column, but their execution is not affected by jitters. As it can be seen (second column of Table 2):

- All evaluated FS methods provide better control performance than the static approach.
- There is a noticeable difference in performance between the last three FS methods with respect to the first two. The last three provide optimal sampling periods taking into account kernel workload but also plant's dynamics. It can be deduced that considering the state of the plants is the key for providing better performance.
- The two first FS methods perform with no significant difference. On-line RA FS changes periods at run-time according to the kernel workload. Since no changes in the workload have been injected, On-line RA FS can not take advantage of its adaptability and once the optimization routine stabilizes the periods (which finally take the same values than [Seto et al., 1996], that is 30ms), its performance is very similar to the off-line FS.
- The difference in performance between the last three FS methods is analyzed in more detail in next Section.

Approach	Indep.	Over.	Sim.
Static approach	104.2	104.2	103.45
Off-line FS Seto et al. [1996]	98.55	94.42	94.69
On-line RA FS Eker et al. [2000]	95.55	95.39	97.93
On-line RCA FS - Inst. Martí et al. [2004]	63.62	64.24	88.97
On-line RCA FS - FH Castañé et al. [2006]	81.95	86.98	100.14
On-line CE FS Ben Gaid et al. [2006]	83.55	88.17	94.54

Table 3. Evaluation in terms of perturbations.

6.2 Detailed Evaluation

Table 3 shows the performance evaluation of the methods as if tasks were executing in isolation (free of jitters) and focusing the analysis on the pattern of perturbations: independent, overlapped or simultaneous.

The first observation is that the conclusions drawn in the previous section still hold but with some variation. Still all FS methods provide better performance than the static approach. However, the clear difference on performance between the first two and last three FS methods does not exist in all the scenarios. It still hold in the independent and overlapped. But in the simultaneous scenario, the performance achieved by all five methods is clearly worst than in the two other scenarios. Moreover, all the methods perform approximately in the same range. One reason is that when perturbations occur simultaneously, there is no satisfactory sampling period readjustment because all control tasks are facing an unacceptable situation and all would like to be assigned more resources, i.e. shorter sampling periods, at the same time, which is not possible due to resource limitations.

6.3 Discussion

In the presented performance evaluation it is important to discuss two aspects that have a strong influence on the results or analysis: a) cost function, and b) resource utilization and computational overhead.

All the FS methods are optimal with respect to a specific cost function in terms of control performance. In addition, some methods consider noise in the optimization procedure. Therefore, to perform a fair comparison is not straightforward. The simplicity of the evaluation setup presented in this paper aims at overcoming this difficulty and providing a fair comparison, in order to extract key conclusions for future research and practice.

The evaluation has only considered control performance. But it is also of prime importance to evaluate resource utilization to assess which methods have the ability to save more resources. And last, all the on-line methods add computational overhead at the real-time kernel level, which also deserves a deeper analysis.

7. CONCLUSIONS

Feedback scheduling of real-time control tasks has received increased attention in the real-time and control research communities in the last years. This paper has presented a control performance analysis that reveals three key aspects. First, it shows that FS has the ability to improve control performance with respect to the standard approach to real-time implementation of control loops. Second, it has been shown that jitters in job executions deteriorate control performance and hide the true performance that can be achieved by these methods. Therefore, they have to be somehow considered in the evaluation. And third, the analysis indicates that those methods that readjust periods (or job sequences) considering plants' dynamics are able to provide the best control performance.

Future work will focus extending the presented analysis. First, the presented performance evaluation will be completed by looking at event-based scheduling approaches, as well as by analyzing resource utilization and computational overhead. Second, experimental evaluations will be also performed.

REFERENCES

K.-E. Årzén, A. Cervin, J. Eker, and L. Sha. An Introduction to Control and Scheduling Co-design. In *39th IEEE Conference on Decision and Control*, 2000.

K.J. Åström, and B. Wittenmark. *Computer-Controlled Systems*. Third Edition, Prentice–Hall, 1997.

M-M. Ben Gaid, A. Çela, Y. Hamam, and C. Ionete. Optimal Scheduling of Control Tasks with State Feedback Resource Allocation. In *2006 American Control Conference*, 2006.

G. Buttazzo. Research Trends in Real-Time Computing for Embedded Systems. *ACM SIGBED Review*, volume 3, number 3, 2006.

R. Castañé, P. Martí, M. Velasco, A. Cervin, and D. Henriksson. Resource Management for Control Tasks Based on the Transient Dynamics of Closed-Loop Systems. In *18th Euromicro Conf. on Real-Time Systems*, 2006.

A. Cervin,and P. Alriksson. Optimal On-Line Scheduling of Multiple Control Tasks: A Case Study. In *18th Euromicro Conference on Real-Time Systems*, 2006.

J. Eker, P. Hagander, and K.-E. Årzén. A Feedback Scheduler for Real-time Control Tasks. *Control Engineering Practice*, volume 8, number 12, 2000.

D. Henriksson, A. Cervin, and K.-E. Årzén. TrueTime: Simulation of Control Loops Under Shared Computer Resources. In *15th IFAC World Congress*, 2002.

C.L. Liu, and J.W. Layland. Scheduling Algorithms for Multiprogramming in a Hard real-Time Environment. *Journal of the ACM*, vol. 20, n. 1, pages 40–61, 1973.

C.D. Locke. Software Architecture for Hard Real-time Applications: Cyclic Executives vs. Fixed Priority Executives. *Real-Time Systems*, vol. 4 n. 1, pp. 37–53, 1992.

C. Lozoya, M. Velasco, and P. Martí. A 10-Year Taxonomy on Prior Work on Sampling Period Selection for Resource-Constrained Real-Time Control Systems. In Work-in-Progress session of the *19th Euromicro Conference on Real-Time Systems*, 2007.

P. Martí, C. Lin, S. Brandt, M. Velasco and J.M. Fuertes. Optimal State Feedback Based Resource Allocation for Resource-Constrained Control Tasks. In *25th IEEE Real-Time Systems Symposium*, 2004.

D. Seto, J.P. Lehoczky, L. Sha, and K. Shin. On Task Schedulability in Real-time Control Systems. In *17th IEEE Real-Time Systems Symposium*, 1996.