Minimizing Control Cost in Resource-Constrained Control Systems: from Feedback Scheduling to Event-driven Control

Camilo Lozoya, Pau Martí, and Manel Velasco

Abstract— This paper evaluates approaches aimed at minimizing aggregated control cost of a set of controllers that concurrently execute sharing limited computing resources. The evaluation focuses on feedback scheduling and event-driven control methods. The performance results drive the analysis to explore self-triggered controllers in the context of minimizing control cost when given a fixed amount of computing resources. This leads to the formulation of an optimization problem, that for given example, is numerically solved. The solution helps understanding the behavior of self-triggered controllers.

I. INTRODUCTION

Applications in networked and embedded control systems are severely constrained in terms of both available resource and performance requirements [1]. Lately, the real-time and control communities have provided diverse theoretical results on both control and resource optimization for resource-constrained computing systems concurrently executing several controllers. Loosely speaking, most of these results suggest to efficiently select the controllers' sampling periods. Hence, controllers' execution rates are different from those provided by the standard periodic sampling approach [2].

Recently, a taxonomy on methods for sampling period selection in resource-constrained control systems was presented [3]. Two main tendencies were identified: feedback scheduling (FS) and event-driven control systems (ED). The main difference between them is that the former primarily looks at the problem of optimizing control performance by fully exploiting the available computing resources. This is achieved by dynamically varying at run-time each controller rate of progress within feasible periodic sampling frequencies, e.g. [4], [5], [6], [7], [8], [9], [10].

On the contrary, the execution of event-driven controllers aims at minimizing resource utilization while ensuring stability or bounding the inter-sampling dynamics. This is achieved by executing controllers without periodic requirements: controllers' jobs are only executed "when needed", e.g. [11], [12], [13], [14], [15], [16], [17], [18], [19], [20].

In [21], the control performance of a set of concurrent controllers was analyzed under the application of various feedback scheduling approaches. The first contribution of this paper is to extend the performance analysis considering different event-based control schemes. In particular, the focus is on the so called "self-triggered" controllers, e.g. [17], [20],

which are controllers characterized, in general, by a non-periodic sequence of job activations, and where at each job execution, apart from performing sampling, control algorithm computation and actuation, the next job activation time is calculated as a function of the plant state.

The analysis reveals that self-triggered controllers provide the best control performance while using the same or similar amount of computing resources than those consumed by feedback scheduling approaches. According to this result, the paper focuses on optimal self-triggered controllers. The second contribution of the paper is a numerical analysis on the performance achieved by these optimal controllers, which helps understanding their potential benefits.

The rest of this paper is structured as follows. Section II presents the extended performance analysis. Section III formulates the self-triggered optimal controller problem. Section IV provides a numerical solution and discusses relevant performance results. Section V concludes the paper.

II. PERFORMANCE ANALYSIS

Control performance optimization and efficient use of the available computing resources are two key elements in the design of resource-constrained control applications. This section extends the performance evaluation presented by [21] in two directions. First, it incorporates as a new evaluation metric the computational demand. Second, it also evaluates selected self-triggered control schemes. A full report on this evaluation can be found in [22].

A. Background

We consider an scenario with n control loops competing for a shared computing resource such as processor capacity or network bandwidth. In particular, processor capacity is considered in this paper. However, the results can be easily extended considering other type of resources. Each control loop contains the controller characterized by an state feedback gain L^i and the controlled plant, which can be modeled by the linear continuous time state-space representation

$$\begin{array}{rcl} \dot{x}^i(t) & = & A^i\,x^i(t) + B^i\,u^i(t) \\ y^i(t) & = & C^i\,x^i(t) \end{array} \qquad (1)$$

with $x^i \in \mathbb{R}^{n \times 1}$, $A^i \in \mathbb{R}^{n \times n}$, $B^i \in \mathbb{R}^{n \times m}$, $u^i \in \mathbb{R}^{m \times 1}$, and $C^i \in \mathbb{R}^{1 \times n}$. Let

$$u^i(t) = u^i_k = L^i\,x^i(a^i_k) = L^i\,x^i_k \qquad \forall t \in [a^i_k, a^i_{k+1}[\quad (2)$$

be the control updates given by each linear feedback controller L^i using only samples of the state at discrete instants $a^i_0, a^i_1, \ldots, a^i_k, \ldots$ Between two consecutive control updates,

$u^i(t)$ is held constant. In periodic sampling we have $a^i_{k+1} = a^i_k + h^i$, where h^i is the period of the controller. The controller execution time is given by c^i.

For the evaluated feedback scheduling approaches [4], [6], [7], the feedback gain L^i is designed as mandated by each method in the discrete time domain considering

$$\begin{aligned} x^i_{k+1} &= \Phi^i(h^i)\,x^i_k + \Gamma^i(h^i)\,u^i_k \\ y^i_k &= C^i\,x^i_k \end{aligned} \tag{3}$$

with $\Phi^i(t) = e^{A^i t}$ and $\Gamma^i(t) = \int_0^t e^{A^i s}\mathrm{d}s B^i$, and where h^i may vary following different patterns. On the contrary, for the evaluated self-triggered control approaches [17], [20], the feedback gain is designed in the continuous time domain.

Regardless of the design procedure for the feedback gain, all the controllers are characterized by the sampling interval h^i that will vary according to the particular approach under evaluation. For the feedback scheduling approaches, in general, each task is associated a cost function $J^i(h^i)$, which gives the control cost as a function of the sampling interval. Then, h^i is the solution of solving

$$\text{minimize} \qquad \sum_{i=1}^{n} J^i(h^i) \qquad \text{w.r.t. } h^i \tag{4}$$

$$\text{subject to} \qquad \sum_{i=1}^{n} \frac{c^i}{h^i} \le U_{ref} \tag{5}$$

where U_{ref} is the desired resource utilization level for the set of control loops.

For the self-triggered control approaches, the variation on the sampling interval is given by $h^i = \Lambda^i(x^i_k, \Upsilon^i, \eta^i)$, where $\Lambda(\cdot)$ is the time spent by each closed loop trajectory from the sampled state $x^i_k = x(a^i_k)$ to reach the given boundary. Boundaries can be described by

$$f^i(e^i_k(t), x^i_k, \Upsilon^i) \le \eta^i \tag{6}$$

where Υ^i is a set of free parameters of f^i, η^i is the error tolerance, and $e^i_k(t) = x^i(t) - x^i_k$ is the error evolution between consecutive samples with $t \in [a^i_k, a^i_{k+1}[$. Therefore, the complete dynamics of the each event-driven system is given by

$$\begin{aligned} a^i_{k+1} &= a^i_k + \Lambda^i(x^i_k, \Upsilon^i, \eta^i) \\ x^i_{k+1} &= (\Phi^i(\Lambda^i(x^i_k, \Upsilon^i, \eta^i)) + \\ & \quad \Gamma^i(\Lambda^i(x^i_k, \Upsilon^i, \eta^i))L^i)x^i_k. \end{aligned} \tag{7}$$

In [23] it is highlighted that to find an expression for $\Lambda^i(x^i_k, \Upsilon^i, \eta^i)$ is sometimes feasible by approximating Φ and Γ by Taylor expansion. An alternative technique for finding Λ^i under several assumptions is given in [20]. Otherwise, Λ^i can only be computed numerically, according to the particular formulation given in each approach.

B. Plants and Simulation Setup

Each controller is in charge of controlling a double integrator electronic circuit, whose state space model is

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & -23.809524 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ -23.809524 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} x \end{aligned} \tag{8}$$

where x_1 is the output voltage.

Simulations have been carried out with the *Truetime* simulator [24] to implement the multitasking processor together with each FS and ED strategy. Three identical double integrators were used as controlled plants. Each one is controlled by a single control task. For the simulation purposes, it is assumed that the initial condition of the three plants has a random non-zero value. When a a task is activated the control actions move the plant state to the equilibrium. The activation of the control task for each plant is produced after a random time-delay. In this way the values of the initial condition and the starting time of each control loop are randomly generated. However it is important to mention that the same random values are used for every evaluated method, in order to have a fair comparison. The duration of each simulation period is 5 seconds, and a total of 30 different simulation runs were conducted for each evaluated method. The reported results corresponds to the average values of all the simulation runs.

C. Evaluated Methods and Main Parameters

The following list summarizes the evaluated methods (see each reference for further information or [22]). Control tasks are scheduled under the Earliest Deadline First (EDF) scheduling [25]. Table I shows key parameters for each method. Each task executes for 10ms, and tasks periods have been selected to have a processor utilization of 50%.

- *Static Approach*: this approach does not belong to feedback scheduling methods nor to self-triggered control methods but it is here included for comparative purposes. Each control task is assigned off-line a sampling period (see Table I for the details). Each task period and controller gain remains constant at run-time.
- *Off-line FS*: this approach is represented by the work by [4]. The objective function (4) 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. Table I shows the stabilized periods.
- *On-line instantaneous (Inst) FS:* a step further is to optimize control performance by on-line adjusting sampling periods according to the plants' dynamics. In [6] the current state is the only information of the plants that is considered in (4). The final outcome of the method mandates to consider at run-time only two periods. Tasks (and controller' gains) switch between these two periods whenever the plant with highest error changes. Table I shows the sampling period values considering that task 1 has the largest instantaneous error.
- *On-line finite horizon (FH) FS:* as before, periods are adjusted online according to the plant dynamics. However, in this case, the current state is the initial condition for predicting the future plants' dynamics over a finite horizon. This prediction is then placed in (4) and solved at run time, as detailed in [7]. Switches of

Approach	Type	Task 1	Task 2	Task 3	η
Static		0.0300	0.0600	0.0600	
Off-line FS	FS	0.0400	0.0400	0.0400	
On-line FS-Inst.	FS	0.0300	0.0600	0.0600	
On-line FS-FH	FS	0.0300	0.0600	0.0600	
Self-triggered ED	ED	0.0417	0.0417	0.0417	0.69
Robust Self-trig. ED	ED	0.0453	0.0453	0.0453	0.36

TABLE I

TASKS SAMPLING PERIODS (SECONDS) FOR THE EVALUATED METHODS.

Approach	Type	Control cost	Resource utilization
Static		1.1742	0.51
Off-line FS [4]	FS	1.1013	0.51
On-line FS-Inst.[6]	FS	1.0624	0.50
On-line FS-FH [7]	FS	1.0663	0.54
Self-triggered ED [17]	ED	1.0456	0.49
Robust Self-trig. ED [20]	ED	1.0438	0.50

TABLE II

CONTROL PERFORMANCE AND RESOURCE UTILIZATION RESULTS

tasks periods (and controllers' gains) occur every 15ms, which is the period of the so-called feedback scheduler. Table I shows the sampling period values considering that task 1 has the largest finite-horizon error.

- *Self-triggered ED*: This method is based on event-driven control and is represented by [17]. In this case the event condition is a particular function of the system state with a tolerated error η. This parameter can be used to adjust the processor load to a 50% of processor utilization approximately. Table I shows the expected average sampling periods. Switches of sampling intervals at run-time are performed according to online computations while the feedback gains remain constant.

- *Robust Self-triggered ED*: This method is also based on event-driven control and is represented by [20]. The key difference with respect to the previous one is that this method provides design guidelines based on robust control techniques for the controller gain while in the previous one, the controller can be freely chosen. In addition, the event condition has different parameters than in the previous case. As before, η was selected to provide a 50% processor load approximately, and switches of sampling intervals occur at run-time while keeping constant the feedback gain.

For the FS methods, control gains are designed in the discrete-time domain to be optimal with respect to the control evaluation metric (9) using standard procedures. Depending on the sampling intervals that may apply, different discrete-time controllers gains have to be designed. However, the self-triggered control methods cannot use the same optimal techniques because no periodic sampling occurs. To obtain an appropriate (in terms of fair performance evaluation) L^i for the event-driven methods, an iterative optimization algorithm was implemented that, given a specific η, and according to the plant dynamics and the cost function (9), searches for the L^i value that provides the minimum control cost. For [20], the optimization search also includes the defining parameters of the boundary.

D. Evaluation Metrics

The two main criteria in order to evaluate the selected methods are: control performance and resource optimization. Therefore, two evaluation metrics have been defined for this purpose. Control performance is measured during each simulation period (t_{sim}) using a continuous standard quadratic

cost function

$$J_{control} = \int_0^{t_{sim}} x^T(t)Qx(t) + u^T(t)Ru(t)\mathrm{d}t \qquad (9)$$

where the weighting matrices Q and R are the identity.

Resource utilization is measured as a percentage of use of the processor during each simulation period (t_{sim}) as

$$J_{resource} = \frac{1}{t_{sim}} \sum_{i=1}^{n} E^i \qquad (10)$$

where n is the number of tasks sharing the same processor, and E corresponds to the total processor time assigned to each specific control task during the simulation period. This explains why resource utilization is not exactly 50%.

E. Performance Results

Table II summarizes the simulation results of the FS and ED methods. Analyzing the control performance, we observe that the static method provides the worst performance, as expected. Then, considering only the FS methods, the on-line algorithms provide the best performance. Comparing FS versus ED methods, it can be noticed that performance and resource utilization is similar, although the last method produces the best results. Note also that the evaluated FS approaches in this paper where the best among other approaches evaluated in [21].

In summary, the analysis shows that self-triggered controllers can provide the best control performance while using the same or similar amount of computing resources than those consumed by feedback scheduling approaches enabled with periodic optimal controllers.

III. OPTIMAL SELF-TRIGGERED CONTROLLER FORMULATION

The preceding analysis points out that event-driven control methods are good candidates in approaches to control performance optimization when computing resources are limited. Focusing on a single controller, and similar to the formulation of an optimal periodic controller, in the following we formulate a generic optimization problem for a self-triggered controller. The optimization goal is to minimize the cost while using the same amount of resources than the periodic optimal controller. See [26] for the dual problem.

Let be a standard quadratic cost function in continuous time defined as

$$J(L, \Upsilon, \eta) = \int_0^{a_\ell} x(t)^T Q_c x(t) + u(t)^T R_c u(t) \mathrm{d}t + \\ x(a_\ell)^T N_c x(a_\ell) \tag{11}$$

The optimal boundary and regulator design problem *to minimize cost* can be formulated as

$$\text{minimize} \quad J(L, \Upsilon, \eta) \quad \text{w.r.t. } L, \Upsilon, \eta \tag{12}$$
$$\text{subject to} \quad x_{k+1} = (\Phi(\Lambda(x_k, \Upsilon, \eta)) + \\ \Gamma(\Lambda(x_k, \Upsilon, \eta))L)x_k \tag{13}$$
$$a_{k+1} = a_k + \Lambda(x_k, \Upsilon, \eta) \tag{14}$$
$$\frac{\sum_{\ell=0}^{k-1} \Lambda(x_\ell, \Upsilon, \eta)}{k} \geq h \tag{15}$$

where (15) enforces an average sampling period larger or equal than some given lower bound h, and where the system dynamics are given by (13)-(14). Enforcing (15) allows an easy comparison with the cost achieved by an optimal h-periodic controller. In this study, Λ is restricted to provide sampling intervals obeying the quadratic event condition

$$[x_{k+1} - x_k]^T M_1 [x_{k+1} - x_k] = \eta x_k^T M_2 x_k \tag{16}$$

where matrices M_1 and M_2, and η are optimization variables, as well as the controller gain L. Quadratic functions for event conditions of the form (16) are a typical choice, like the ones of the evaluated event-driven methods [17], [20]. In addition, in [23] it was shown that for these type of conditions, an approximated explicit solution to the problem of calculating the next activation time exists. In particular, from the event condition (16), the next activation time can be deduced to be the positive value of

$$t = \frac{\sqrt{-4[A_{cl}x_k]^T M_1 [A_{cl}x_k](-\eta) x_k^T M_2 x_k}}{2[A_{cl}x_k]^T M_1 [A_{cl}x_k]} \tag{17}$$

where $A_{cl} = (A - BL)$ [23]. This facilitates the numerical simulation analysis.

IV. NUMERICAL ANALYSIS

A simulation process is conducted to evaluate the control performance of an optimal self-triggered controller that numerically solves problem (12)-(15) using Montecarlo methods. This controller is compared with the performance provided by an optimal periodic controller, both controlling a double integrator circuit (8). Since in an event-driven approach the sampling is non-periodic, an average sampling period value is considered in order to have a fare comparison with the periodic controller. As outlined before, for the self-triggered controller, η defines the average sampling period, so different values of η were selected during this study.

A. Influence of the Initial Condition

For periodic optimal controllers, it is well known that the optimal cost depends on the initial condition [2]. The first analysis aims at studying if this is also true for the case of an optimal self-triggered controller.
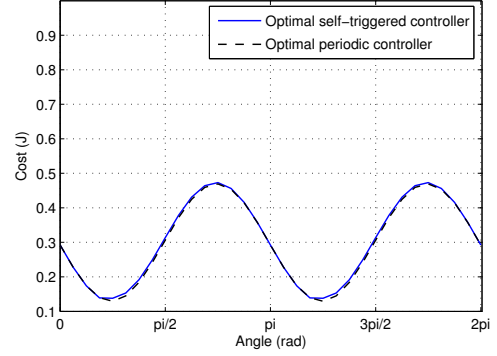

Fig. 1. Control cost variations as a function of the initial state, for a small sampling period.
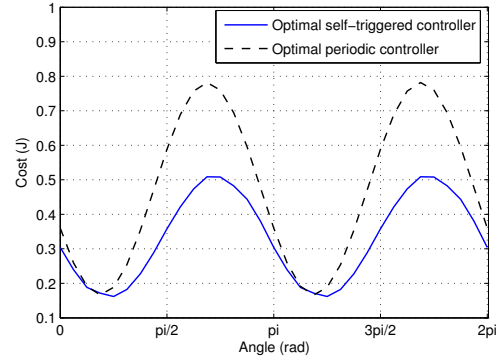

Fig. 2. Control cost variations as a function of the initial, for a large sampling period.

To do so, different initial conditions are considered in order to evaluate the control cost. Figure 1 plots the control cost achieved by both controllers for "any" initial state vector with $|x| = 2$. That is, the angle of the state vector varies from 0 to 2π, covering all possible state vector directions, while keeping its norm equal to 2. As expected, the cost varies depending on the initial condition, and it is symmetric. It is important to note that for both controllers, the sampling period is small. Specifically, for the periodic controller $h = 0.0062s$, and for the self-triggered controller, $\eta = 0.03$ in order to obtain in average the same sampling interval. Looking at performance, the cost is practically the same in both cases.

B. Influence of the Sampling Period

Usually, for many linear systems, the longer the sampling period, the worst the control cost. The analysis in this section studies whether this property also holds for the optimal self-triggered controller.

Figure 2 shows the same information than Figure 1 but when the sampling period for the periodic controller is $h = 0.0453s$, and when $\eta = 0.30$ for the self-triggered controller, which gives the same sampling interval in average. As it can be seen in the figure, the optimal self-triggered controller
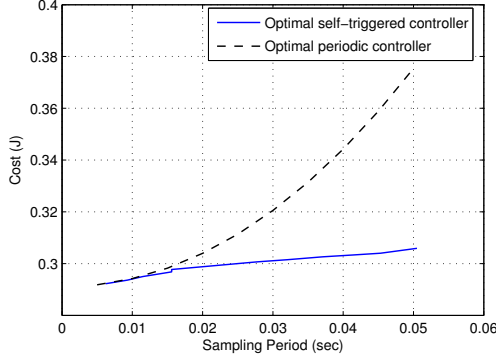
Fig. 3. Control cost for different sampling periods (h) for a given initial condition.
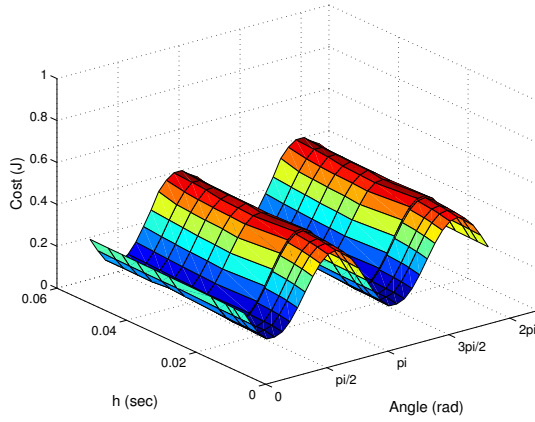

Fig. 4. Self-triggered controller cost for different sampling periods and initial conditions.

Fig. 5. Periodic controller cost for different sampling periods and initial conditions.

Approach	Type	Control cost	Resource utilization
Optimal Self-Trig. ED	ED	1.0363	0.52

TABLE III

CONTROL PERFORMANCE AND RESOURCE UTILIZATION FOR THE OPTIMAL SELF-TRIGGERED CONTROLLER STRATEGY

provides in general better performance (lower cost) except for specific directions where there is no cost benefit when compared to the optimal periodic controller.

To further analyze the influence of the sampling period, Figure 3 plots the cost of both controllers as a function of the sampling period for a given initial condition (with angle equal to zero). Again, for the case of the self-triggered controller, η has been tuned to provide in average the same sampling period with the periodic optimal controller. The figure shows an interesting property: the longer the sampling period, the better is the optimal self-triggered controller compared to the optimal periodic controller.

C. Cost, Sampling Period, and Initial Condition

This last evaluation focuses on the relation of control cost, sampling period, and initial condition, as shown in Figures 4 and 5 for the optimal self-triggered controller and for the optimal periodic controller, respectively. Based on these figures, it can be stated that for small sampling periods, the control cost is equal for both controllers regardless of the initial condition. However, for the case of the periodic controller (Figure 5), when the sampling period increases, there is an exponential growth in control cost in those
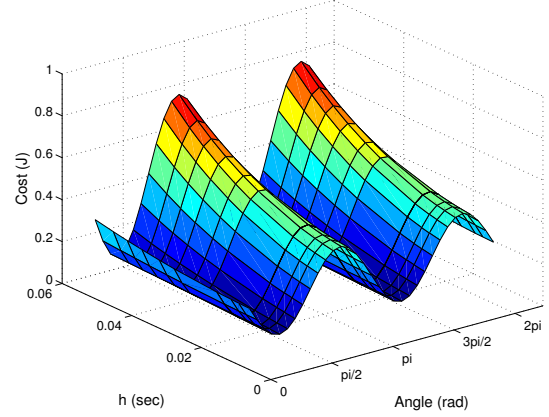
regions of the state space where the cost is already high due to the initial condition. On the contrary, the self-triggered controller, for long sampling periods, only experiences just small linear cost increments. This indicates that it is robust to increments on the average sampling interval.

D. Evaluation in the Multitasking System

To complete the performance evaluation of the optimal self-triggered controller solving the formulated problem (12)-(15), the evaluation presented in section II is recovered and extended with this new controller. That is, three optimal self-triggered controllers each one in charge of a double integrator circuit are simulated under EDF, providing a processor load of 52% with an expected average sampling period $h = 0.0478$s for each of the three tasks (achieved with $\eta = 0.37$).

Table III completes the evaluation shown in table II with the performance numbers of the three optimal self-triggered controllers. Comparing both tables, it can be observed that this new strategy is the best in terms of control performance, i.e., lowest cost. Note also that the improvement could have been bigger if longer sampling periods were applied.

E. Discussion

At this point, some preliminary conclusions can be extracted for considering when the application of optimal self-triggered controllers can be beneficial. The first one is that for severe limited computing systems, the use of longer sampling periods is a key point for saving resources. Therefore, optimal self-triggered controllers are the best choice for providing the best control performance compared to optimal periodic controllers.
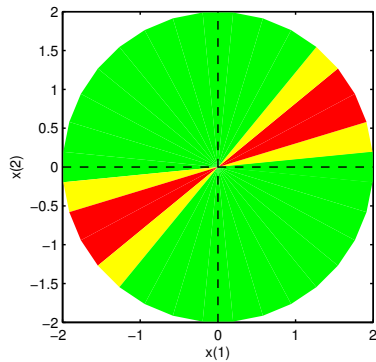
Fig. 6. Self-triggered controller impact over the state space plane.

Secondly, it is also of interest to observe which partitions of the state space favor the performance of optimal self-triggered controllers. To this extend, Figure 6 identifies in which areas the system trajectory should move to provide better control performance, in the state space plane formed by the two state variables of (8) . The three colors indicate how large is the impact/benefit of the optimal self-triggered controller compared with the optimal periodic one, for a specific sampling period ($h = 0.0321s$). Red color indicates that the self-triggered controller provides no-benefit compared with the periodic one. Yellow color indicates a small benefit, up to 5%. And green color indicates a bigger benefit. According to the previous results, for larger sampling periods the green areas increases while for smaller sampling periods the red areas increase.

In any case, optimal self-triggered controllers moving in the green area will provide benefit with respect to the optimal periodic controller in terms of control cost. To guarantee this benefit, a simple approach would be to enforce a non-oscillating dynamic once the closed-loop system trajectory enters into a green area. Or to enforce an oscillating dynamic jumping between green areas. Future work will focus on designing optimal self triggered controllers by constraining their state vector inside these green areas.

V. CONCLUSIONS

This paper has evaluated existing techniques for control performance and resource optimization in resource constrained control systems when several control loops share limited computing resources. The analysis has shown that event-driven control methods, and in particular, self-triggered controllers, are good candidates for these scenarios.

In addition, a detailed analysis has been performed for the case of an optimal self-triggered controller. Interesting properties have been observed, providing preliminary insight for understanding self-triggered controllers behavior.

REFERENCES

[1] G. Buttazzo, "Research Trends in Real-Time Computing for Embedded Systems," *ACM SIGBED Review*, v. 3, n. 3, 2006.
[2] K.J. Åström and B. Wittenmark, *Computer controlled systems*, Prentice Hall, 1997.
[3] 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 *in WiP of the 9th Euromicro Conference on Real-Time Systems*, 2007.
[4] D. Seto, J.P. Lehoczky, L. Sha, and K.G. Shin, K.G., "On task schedulability in real-time control systems," *in 17th IEEE Real-Time Systems Symposium*, Washington, DC, USA, 1996.
[5] J. Eker, P. Hagander, and K.-E. Årzén, "A Feedback Scheduler for Real-time Control Tasks," *Control Engineering Practice*, volume 8, number 12, 2000.
[6] 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.
[7] 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 Conference on Real-Time Systems*, 2006.
[8] 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.
[9] P. Martí, C. Lin, S. Brandt, M. Velasco, and J.M. Fuertes, "Draco: Efficient Resource Management for Resource-Constrained Control Tasks," *IEEE Transactions on Computers*, Jan. 2009.
[10] M-M. Ben Gaid, A. Çela, and Y. Hamam, "Optimal Real-Time Scheduling of Control Tasks with State Feedback Resource Allocation," *IEEE Trans. on Control Systems Technology*, unpublished, 2009.
[11] K.-E. Årzén, "A Simple Event-Based PID Controller", *in 14th World Congress of IFAC*, January, 1999.
[12] W.P.M.H. Heemels, R.J.A. Gorter, A. van Zijl, P.P.J. van den Bosch, S. Weiland, W.H.A. Hendrix and M.R. Vonder, "Asynchronous measurement and control: a case study on motor synchronization," *Control Engineering Practice*, Vol. 7, pp. 1467–1482, 1999.
[13] K. J. Åström and B. Bernhardsson, "Comparison of Riemann and Lebesgue sampling for first order stochastic systems," *in 41st IEEE Conference on Decision and Control*, December 2002.
[14] M. Velasco, P. Martí and J.M. Fuertes, "The Self Triggered Task Model for Real-Time Control Systems," *in WiP of the 24th IEEE Real-Time Systems Symposium*, Decembre 2003.
[15] P. Tabuada, Event-triggered real-time scheduling of stabilizing control tasks, *IEEE Trans. on Automatic Control*, 52(9), 2007, pp. 1680-1685.
[16] M. Lemmon, T. Chantem, X. S. Hu, and M. Zyskowski, "On self-triggered full information h-infinity controllers," *in Proceedings of the 10th International Conference on Hybrib Systems: Computation and Control*, Pisa, Italy, Apr. 2007.
[17] A. Anta and P. Tabuada, "Self-triggered stabilization of homogeneous control systems", *in 2008 American Control Conference*, 2008.
[18] W.P.M.H. Heemels, J.H. Sandee, and P.P.J. van den Bosch, Analysis of event-driven controllers for linear systems, *International Journal of Control*, 81(4), 2008, pp. 571-590.
[19] T. Henningsson, E. Johannesson, A. Cervin, "Sporadic Event-Based Control of First-Order Linear Stochastic Systems," *Automatica*, vol. 44, n. 11, pp. 2890–2895, Nov. 2008
[20] X. Wang, and M. Lemmon, "Self-triggered Feedback Control Systems with Finite-Gain L2 Stability" *IEEE Transactions on Automatic Control*, v. 45, n. 3, pp. 452–467, March 2009.
[21] C. Lozoya, P. Martí, M. Velasco and J.M. Fuertes, "Control Performance Evaluation of Selected Methods of Feedback Scheduling of Real-time Control Tasks," *in 17th IFAC World Congress*, July, 2008.
[22] C. Lozoya, P. Martí, M. Velasco and J.M. Fuertes, "Simulation Study on Control Performance and Resource Utilization for Resource-Constrained Control Systems" *ESAII-RR-09-01 Technical Report*, Barcelona, Spain, Marchc, 2009 (http://www.upcnet.es/ pmc16/09RREvaluation.pdf).
[23] M. Velasco, P. Martí and E. Bini, "Control-driven Tasks: Modeling and Analysis", *in 29th IEEE Real-Time Systems Symposium*, 2008.
[24] D. Henriksson, A. Cervin, and K.-E. Årzén, "TrueTime: Simulation of control loops under shared computer resources," *in 15th IFAC World Congress*, 2002.
[25] 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.
[26] P. Martí, M. Velasco, and E. Bini, "The Optimal Boundary and Regulator Design Problem for Event-Driven Controllers," in *12th International Conference on Hybrid Systems: Computation and Control*, San Francisco, CA, USA, April 2009.