Sampling Jitter Compensation in Real-time Control Systems

Pau Marti and Josep Mª Fuertes
Dept. of Automatic Control
Technical Univ. of Catalonia, Spain
{pmarti,villa,pepf}@esaii.upc.es

Gerhard Fohler
Dept. of Computer Engineering
Mälardalen University, Sweden
gfr@mdh.se

1. Introduction

In control theory, sampling and actuation are generally assumed synchronous and periodic, and a highly deterministic timing of an implementation is assumed [4]. When a control algorithm is executed by a task (or by a set of subtasks) in a multitasking real-time system, those assumptions are not met. This causes control performance degradation and even instability [7]. In addition, the scheduling algorithm may over-constrain the system when trying to fulfil the stringent timing constraints that control theory mandates, resulting in a poor schedulability

In this paper, we propose an integrated approach to the design of real-time computer-controlled systems in order to obtain both the best control performance as well as the best system's schedulability.

Real-time scheduling algorithms introduce various forms of jitter to each task instance execution. As a consequence, if those tasks are executing traditional control designs (control tasks), the system performance is not as good as expected. On the other hand, by using an adequate control design strategy that takes into account those jitters, the system performance can improve dramatically. In order to fully use these more suitable control design strategies, the scheduling algorithm may have to provide, at some point, valuable information to each control task instance for compensating the current introduced jitters. At the same time these control design strategies will provide new flexible constraints to each control task instance, allowing a better schedulability of the whole set of tasks (control and non control tasks).

In this paper we focus on the sampling jitter problem and show that by using this compensation approach, the control performance degradation due to the sampling jitter introduced by real-time scheduling algorithms is eliminated while guaranteeing stability [8].

2. Related work

Recently, several works have been presented addressing important issues in the field of real-time control systems. For a state-of-the-art, see [2], where the compensation approach to help to improve control performance is also suggested. The integration of the compensation approach in a joint approach to control and scheduling, opens the possibility of better schedulability. In particular, we will discuss how offline schedulability analysis along with online scheduling and the use of online control compensation can be used to achieve these goals. With our approach, we can address the problems due to sampling jitters – not addressable using traditional EDF and RM based scheduling and by previous real-time and control integration approaches - by making use of control properties and through online compensation.

For space limitations, we refer to [9] for an extended revision of relevant work related to our approach.

3. Problem definition

The main parts of a control loop are sampling, control algorithm computation and actuation. Sampling should be performed at the same instant every period (h), which means a constant sampling period. The control algorithm computation should start and finish as soon as possible after the sample is available. The actuation should be performed immediately after the algorithm computation, or at a fixed instant after the sampling, which means a constant actuation interval (depending on how the controller was designed). In control theory, the three main parts of a control loop are assumed instantaneous (Fig.1).

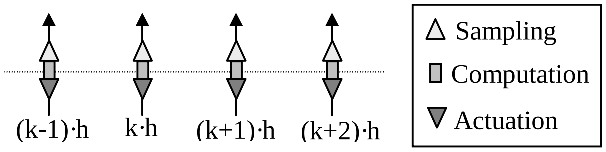

Figure 1. Ideal sequence of control task instances executing a control loop

There are two ways of implementing control loops in RT systems. The single task approach performing the three actions sequentially and the multitask approach, each task performing one or more parts of a control loop (this approach can include sampling and actuation via interrupts –strictly periodic- or the actuation via strictly periodic dummy task).

In addition, taking into account the different ways of implementing a control loop, different control system models (descriptions) will apply depending on the control loop implementation strategy (discrete time system [4], discrete time system with time delays [10] and discrete time system with actuation in the next sampling [4]).

Both task approaches (single task or multitask) implementing a control loop may violate one or more control loop assumptions when scheduled in a multitasking real-time system due to the introduced jitters. Sampling may be not constant. Control algorithm computation may start later than the instant in which the sample is available and the control algorithm computation may not be instantaneous and even have varying computation time. Actuation then will be performed at varying time instants (a_k). For an overview of which violations can appear for each task approach, see [9].

In this paper we focus on sampling jitter (h_k). Whether the control loop has been implemented using the single task or multitask approach, sampling jitter implies that the separation between consecutive samples is not constant.

To avoid the degradation that sampling jitter introduces, we can compensate for it using a more suitable control design strategy, the compensation approach. Afterwards, each task instance (implementing the new control design) will work jointly with the scheduler, compensating for the degradation and allowing a better schedulability of the whole system.

4. Compensation approach: Discussion

4.1. A suitable control design strategy

The main idea behind the compensation approach, that was suggested in [1] and [3] in order to compensate for variations from sample to sample, and further developed in [7], is to adjust at runtime the controller parameters at each control task instance execution according to the actual jitters. A stability analysis of this method is developed in [8].

The compensation approach is used for compensating the degradation that irregular sampling causes into the control system response. An example, obtained using the simulator presented in [5], is given in Figure 2 to show the effectiveness of this approach. Figure 2 (top) shows the ideal inverted pendulum angle closed loop system response. Figure 2 (center), shows the effects (instability) of sampling jitter on the system response. Figure 4 (bottom) shows the effectiveness of the compensation approach on the closed loop system response. For further details, see [7]

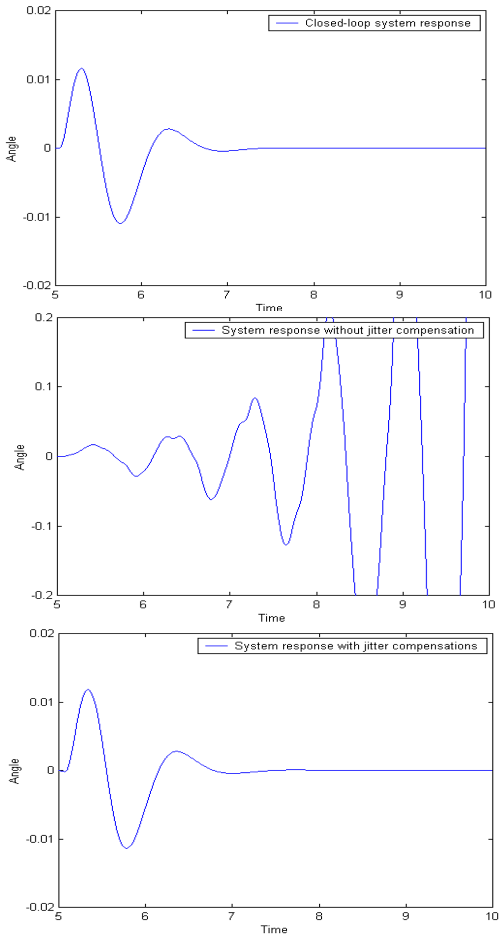

Figure 2. Compensation approach effectiveness

4.2. Implementation cost

An important factor in the implementation of the compensation approach is its computational cost. In Figure 3 it can be clearly seen that the increase of a control task execution time degrades the control system performance.

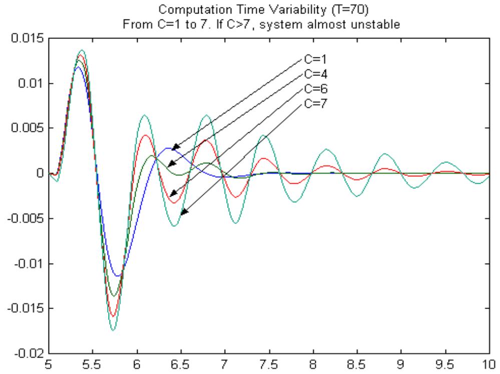

Figure 3. System degradation due to execution time increase

What we know is that at each control task instance execution, the controller parameters must be updated according to the actual sampling jitter (h_k). If this actualization is performed by online extra calculations, the introduced overhead that will depend on the control design strategy that is being used must be assessed. For example, for a discretization of a continuous time designed PID controller, the overhead is insignificant, but for a discrete time design and pole placement controller, the introduced overhead is important ($O(n^4)$, where n is the closed loop system matrix dimension.) [9].

Therefore, two implementation possibilities must be considered:

Runtime recalculations: for insignificant overhead, all parameters recalculations can be performed at runtime.

Offline recalculations: for significant overhead, the parameter updates can be pre-calculated offline and stored in a look-up table. In this later case, at runtime, the controller parameters will be updated by accessing the look-up table. The look-up tables will have the sampling jitter h_k as a input parameter, an for each h_k, the recalculated controller parameters must be stored. To size the tables, we know that the sampling jitter will have a bounded variability ($h_{min} < h_k < h_{max}$, where h_{min} shall include the minimum task instance execution time -if instantaneous, its size should be the system clock tick size- and h_{max} shall be less than four times the systems characteristic's frequency). Therefore, the worst assumption is to give full range to h_k, knowing that the sampling interval is a multiple of the clock tick size. Therefore, the size of these tables are estimated to be of a few Kb [9] is:

$$\text{Size(table)}=(h_{max}-h_{min})*\text{ ticksize}*\text{size(controlparameters)}$$

4.3. Information availability

Another important factor in the implementation of the compensation approach is whether all the necessary information to recalculate the controller parameters is available when it's needed.

To compensate sampling jitter, the control design mandates that each sampling interval h_k must be known at the beginning of the each control task instance execution. Runtime recalculations are possible because h_k can be known online by time measurements. Offline recalculations are possible because h_k or its bounded variability can be known offline. It must be pointed out that before all the parameters recalculations are done, the stability of the system depending on h_k has been tested. For further details, see [8]

5. A new schedulability problem

In the previous section we have discussed the suitability of the compensation approach for the implementation of real-time computer-controlled systems. It has been shown that this approach can compensate itself sampling jitters that current scheduling algorithms such as RM or EDF [6] based algorithms may introduce.

In addition, if the compensation approach is utilized, the traditional timing constraints (period, WCET, deadline) that are currently used in the schedulability problem will have a new dimension.

Instead of having a period (h), a bounded interval of possible periods will apply ($h_{min} < h_k < h_{max}$). Deadlines, from a control viewpoint, will not be a (relative) fixed time but another interval of possible deadlines that will probably have associate a quatily of service parameter. WCET will remain for the worst case assumption.

6. Summary and Ongoing Activities

In this paper, we proposed a new approach for real-time scheduling on control systems by compensating for sampling jitter with adjustment of controller parameters. We calculate adjusted parameters for a set of sampling jitter values offline, which are used online to compensate for the introduce control degradation while guaranteeing stability.

We have focussed on sampling jitter in this paper. Currently, we are investigating and extending the proposed compensation approach to offline calculate adjustment parameters to all types of jitter, ie, computation and actuation.

Here we have calculated parameters to be used on a set of jitter values. We are envisioning the basic methodology to be applied in an "inverse" way as well: instead of calculating controller parameters to compensate for variations in scheduling intervals, we can use the parameters to define scheduling intervals. That way more flexible schedules can be used while maintain stability. We are currently looking into the novel scheduling schemes required.

7. References

[1] P. Albertos and J. Salt. "Digital Regulators Redesign with Irregular Sampling", *11th IFAC World Congress* (Preprints), vol 8, pp 157-161, 1990

[2] K-E. Årzen, B. Bernhardsson, J. Eker, A. Cervin, K. Nilsson, P. Persson and L. Sha "Integrated Control and Scheduling". Research report ISSN 0820-5316. Dept. Automatic Control, Lund Institute of Technology, 1999

[3] K.-E. Årzen, A. Cervin, J. Eker and L. Sha. "An Introduction to Control and Scheduling Co-Design", *39th IEEE Conference on Decision and Control, Sydney*, Australia, December 12-15, 2000

[4] K.J. Åström and B. Wittenmark. *Computer-Controlled Systems. Third edition*. Prentice Hall. 1997

[5] J.Eker and A. Cervin "A Matlab Toolbox for Real-Time and Control Systems Co-Design", *in Proc. of the 6th Int. Conference on Real-Time Computing Systems and Applications*, Hong Kong, China, December 1999

[6] C. Liu and J. Layland "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment". *J.ACM*, 20, 46-61. 1973

[7] P. Marti, R. Villa, J.M. Fuertes and G. Fohler, "On Real-Time Control Tasks Schedulability". *Accepted for publication to European Control Conference,* September, 2001

[8] P. Marti, R. Villa, J.M. Fuertes and G. Fohler "Stability of On-line Compensated Real-Time Scheduled Control Tasks" Research report ESAII-RT-01-01. Automatic Control Department, Technical University of Catalonia. March 2001

[9] P. Marti, R. Villa, J.M. Fuertes and G. Fohler "An approach to joint control and scheduling" Research report ESAII-RT-01-03. Automatic Control Department, Technical University of Catalonia. April 2001

[10] Wittenmark B., J. Nilsson and M. Törngren. "Timing Problems in Real-Time Control Systems". *Proc. Of the American Control Conference*. US. 1995