Demo Abstract: Implementation of Self-triggered Controllers

Antonio Camacho, Pau Martí, and Manel Velasco
Automatic Control Dept., Technical University of Catalonia
Barcelona, Spain
{*antonio.camacho.santiago,pau.marti,manel.velasco*}*@upc.edu*

Enrico Bini
Scuola Superiore Sant'Anna
Pisa, Italy
e.bini@sssup.it

Abstract

A self-triggered controller is characterized, in general, by a non-periodic sequence of job activations. Each job execution, apart from performing sampling, control algorithm computation and actuation, calculates the next job activation time as a function of the plant state.

This demo paper presents an implementation of self-triggered controllers. Specifically, an unstable plant implemented by a double integrator electronic circuit is controlled by a self-triggered controller running on top of a real-time kernel on a low-cost micro-controller.

1. Introduction

Control systems can be described by the evolution of the plant dynamics over time. The trajectory is sometimes represented in some space domain such as the state-space. Hence, control systems can be entirely described using two domains, the time and space domains.

Periodic control bases its operation by activating controller jobs according to a discretization in the time domain. Commonly, this discretization is regular, providing equidistant activation instants, i.e. constant sampling period.

For various type of event-driven control approaches the activation of controller jobs occurs when the system trajectory crosses *boundaries* that define a discretization in the space domain. For "self-triggered" controllers, e.g. [1], [4], the activation times are computed at each job execution, as illustrated in Figure 1 that shows the pseudo-codes of a periodic controller and a self-triggered controller.

```
periodic_controller{
   wait(h);
   x_i=read_input();
   u_i=Lx_i;
   write_output(u_i);
}
self_triggered_controller{
   wait(next_act);
   x_i=read_input();
   u_i=Lx_i;
   write_output(u_i);
   next_act=compute_next_activation_time(x_i);
}
```

Figure 1: Periodic vs. self-triggered controller.

2. Plant, Controller and Simulation Results

The state-space model for the electronic circuit is

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & -23.809524 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ -23.809524 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} x. \end{aligned} \quad (1)$$

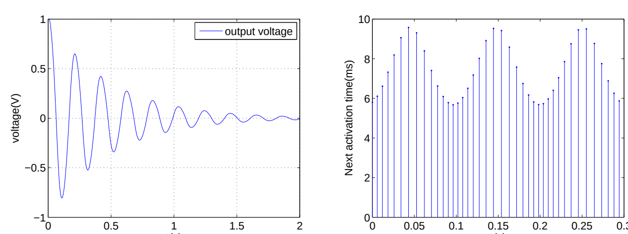
where the output voltage is the circuit output.

A self-triggered controller as been simulated giving a strong oscillatory response. It has been designed in continuous time by standard pole placement, placing the continuous closed loop poles at $p_{1,2} = -4 \pm 30i$. Its gain is $L = \begin{bmatrix} 1.6158 & -0.3360 \end{bmatrix}$. The controller is executed obeying the quadratic event condition

$$[x_{k+1} - x_k]^T M_1 [x_{k+1} - x_k] = \eta x_k^T M_2 x_k \quad (2)$$

with matrices M_1 and M_2 being the identity, and the tolerated error $\eta = 0.05$, and where x_k denotes the plant state at time t_k. In [3] it was shown that for these types of condition, an approximated solution to the problem of calculating the next activation time exists.

Figure 2 (a) shows the simulated closed loop response achieved by the self-triggered controller. As it can be seen, from the initial state $x = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$, the output voltage strongly oscillates tending to zero. Figure 2 (b) shows the pattern of the jobs' activations for the self-triggered controller. The x-axis is simulation time (note that only 0.3s are displayed), and the y-axis is the sampling interval in



(a) Plant response (b) Zoomed activation times

Figure 2: Self-triggered controller simulation

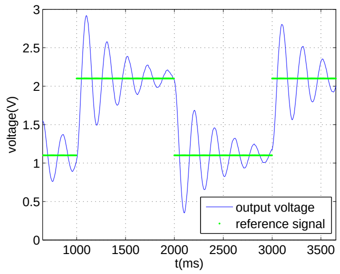


Figure 3: Experimental system response

milliseconds. Each job activation time is represented by a vertical line, whose height indicates the next job activation time. Hence it can be noticed that denser activations occur when the heights are shorter. The sequence of sampling intervals show an oscillatory pattern, that varies between 0.005 s and $0.010s$ and repeats approximately every 0.1 s.

3. Implementation Strategy and Results

The aperiodic task model seems to be the natural abstraction for implementing event-driven controllers. In particular, their execution could be enforced by the EDF scheduling algorithm for aperiodic tasks with dynamic arrivals. However, since arrival times are not known a priori, the guarantee test has to be done dynamically. Hence there is no a priori guarantee that all jobs will be executed.

To overcome this problem, [3] investigated whether event-driven controllers could be implemented using the periodic task model and scheduled by standard EDF or FP scheduling algorithms for periodic workloads. Their results show how both standard response time analysis and time demand analysis for FP, and demand bound function for EDF can be tailored for scheduling event-driven controllers. The main steps toward these results lie on the ability of (i) estimating the maximum amount of computational resource required by each controller, and (ii) computing the average period that permits to define the controller utilization.

The implementation of self-triggered controllers using periodic tasks requires reseting their associated application timers or alarms at each job execution with the calculated next activation time. Taking this into account, the event-driven controller is implemented by a periodic task executing on the Erika 1.4.3 real-time kernel from Evidence Srl[1] and scheduled under EDF scheduling algorithm. The executing platform is the full Flex board, also from Evidence, equipped with a dsPIC.

Figure 3 shows the response of the experimental feedback control loop. Although in the experiment the controller goal is to track a reference signal while in the simulation is

1. http://www.evidence.eu.com

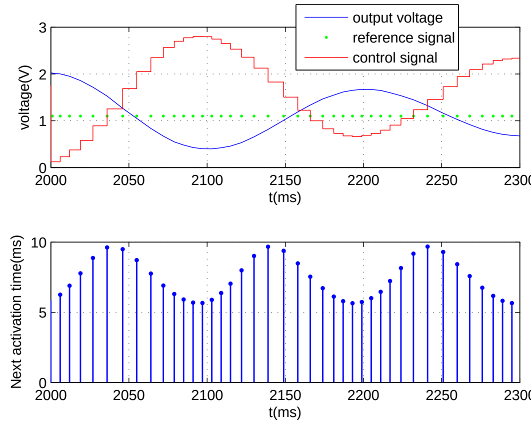


Figure 4: Zoomed response and activation times

to settle from an initial condition different from zero, the achieved dynamics are the same. Figure 4 shows a zoomed view of the experimental plant response and activation times. In the plant response sub-figure, the control signal is also displayed in order to permit observing that control actions are aperiodic. The bottom sub-figure shows the activation times in the same layout as in Figure 2(a). The same type of activation times is achieved.

4. Conclusion

This paper has shown that implementation of advanced control results, i.e. self-triggered controllers, is feasible with the appropriated real-time analysis and kernel support. See [2] for a full paper on this topic.

Acknowledgment

This work was partially supported by NoE ARTISTDESIGN IST-FP7-2008-214373, and by Spanish Ministerio de Educacin y Ciencia Project CICYT DPI2007-61527.

References

[1] A. Anta and P. Tabuada, “Self-triggered Stabilization of Homogeneous Control Systems,” in *2008 American Control Conference*, 2008.

[2] A. Camacho, P. Martí, Manel Velasco, and Enrico Bini, “Implementation of Self-triggered Controllers,” submitted to *IEEE Transactions on Industrial Informatics*, 2009. (available at http://paginespersonals.upcnet.es/~pmc16/09ieeetiievents.pdf)

[3] M. Velasco, P. Martí, and E. Bini, “Control-driven Tasks: Modeling and Analysis,” in *29th IEEE Real-Time Systems Symposium*, Barcelona, Spain 2008.

[4] X. Wang, and M. Lemmon, “Self-triggered Feedback Control Systems with Finite-Gain L2 Stability” *IEEE Transactions on Automatic Control*, accepted July, 2008.