Resource-Aware Control and Dynamic Scheduling in CPS

W.P.M.H. Heemels(✉)

Control System Technology Group,
Department of Mechanical Engineering,
Eindhoven University of Technology, Eindhoven, The Netherlands
m.heemels@tue.nl
http://www.heemels.tue.nl

1 Introduction

Recent developments in computer and communication technologies are leading to an increasingly networked and wireless world. In the context of control for cyber-physical systems (CPS) this raises new challenging questions, especially when the computation, communication, energy and actuation resources (for control) of the system are limited and/or shared by multiple control tasks. Examples include limitations in the battery power for wireless sensors, restrictions on actuator moves to avoid strain, multiple actuators sharing the same hardware resource (e.g., several motors sharing one amplifier), many control tasks sharing the same processor and/or communication medium, and so on. Such limitations obstruct the use of classical techniques for the design of feedback control algorithms for CPS and call for new *resource-aware* controller synthesis paradigms.

These new resource-aware control systems typically have to take both discrete and continuous decisions. For instance, in the control of a robot arm in which the motors driving the joints share the same amplifier (and consequently only one joint can be powered at a time), the control system would have to determine based on, e.g., position and velocity information of the robot, which joint (discrete decision) to power and which value of the torque (continuous value) to apply in order that the robot carries out its overall motion task in a desirable manner. Clearly, this a hybrid co-design problem in which both discrete and continuous decisions have to be taken by the resource-aware control algorithm preferably exploiting real-time measurement information available on the physical system.

In this talk two perspectives on this general hybrid co-design problem are discussed.

The work of Maurice Heemels was partially supported by the Dutch Science Foundation (STW) and the Dutch Organization for Scientific Research (NWO) under the VICI grant "Wireless controls systems: A new frontier in automation" (Project number 11382).

C. Berger and M.R. Mousavi (Eds.): CyPhy 2015, LNCS 9361, pp. 1–6, 2015.
DOI: 10.1007/978-3-319-25141-7_1

2 Dynamic Scheduling and Control

The first approach is based on control and scheduling co-design and has similarities to well-known time-sharing solutions. Essentially the time line is divided into specific slots and in each slot one of the (feedback) control tasks is allowed to access the shared resource being, for instance, a computation, communication or actuation device. As an example consider the networked control system (NCS) in Fig. 1 in which we have a physical plant controlled by a feedback controller over a shared communication network. The physical plant is equipped with n_y sensor nodes measuring $y^1, y^2, \ldots, y^{n_y}$, respectively, and there are n_u actuator nodes for which the controller produces the control values $u^1, u^2, \ldots, u^{n_u}$, respectively. As the network is shared among these nodes and communication constraints prohibit that multiple nodes transmit at the same time, at each transmission instant only one of these nodes can transmit its corresponding values (e.g., if sensor node 2 is allowed to communicate at time t then $y^2(t)$ is communicated and $\hat{y}^2(t)$ is updated to this value). Clearly, this calls for a network protocol deciding in which order nodes can communicate (discrete decisions) *and* a feedback controller that based on the received measurement information $\hat{y}^1, \hat{y}^2, \ldots, \hat{y}^{n_y}$ determines the control values $u^1, u^2, \ldots, u^{n_u}$. This is essentially a co-design problem as the choice of network protocol will influence which controller yields optimal performance and behaviour of the overall CPS.

Compared to common scheduling approaches of control loops, which typically use fixed periodic (round robin) schedules, in our solution we strive for *dynamic scheduling* of control tasks based on measured information obtained from the physical plant to be controlled. By exploiting this information in the decision process improved overall performance of the CPS can be achieved. We discuss a modelling framework and solution strategies for this hybrid co-design problem in which the control and scheduling algorithm has to take both discrete and continuous decisions. In fact, we are able to guarantee that the proposed dynamic scheduling and control method will outperform any given periodic scheduling and control solution in terms of improved overall performance. This part is mainly based on our work in [2–4].

3 Event-Triggered and Self-triggered Control

The first approach takes a rather 'global' view aiming at scheduling *all* tasks such that the resource constraints are adhered to. The second approach, described next, adopts a different point of view as each individual control task aims at only requesting access to a resource when it really has to, i.e., all tasks try to operate under a "minimal attention policy" [7], while still guaranteeing desired overall stability and performance specifications. Consequently, this setup is more self-organising as each task determines locally and independently when to execute. It does not require a global view of the CPS in the design phase and its implementation, although still one has to verify that the overall resource constraints are met.

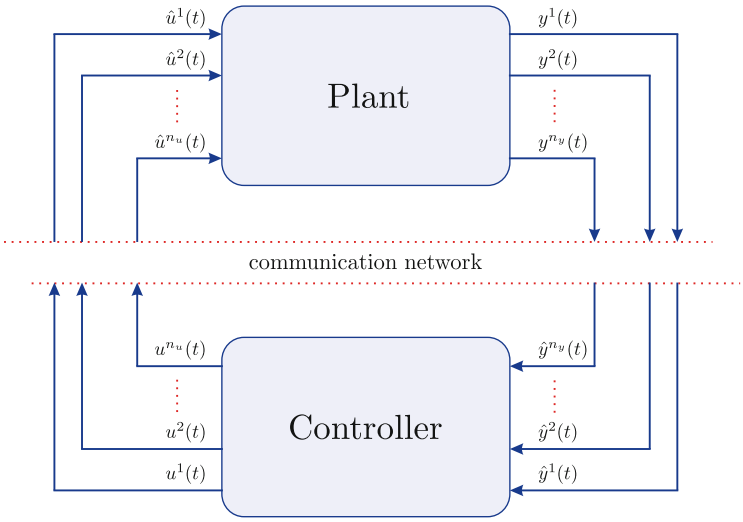

Fig. 1. NCS with a shared communication network.

Before explaining this more self-organising resource-aware control strategy, let us shortly review the conventional method for implementing feedback control tasks. In the majority of digital control applications the execution of control tasks is performed in a periodic time-triggered fashion (connecting conveniently to the periodic scheduling of multiple tasks). A drawback of the time-triggered paradigm is that control tasks are executed independently of the state of the plant and the actual need to execute these tasks. In fact, the decision of executing a control task is (almost) always taken in an "open-loop" fashion; there is no feedback-based decision mechanism active that determines whether or not it is actually necessary to carry out specific sensing, communication, computation, or actuation (update) tasks in order to realise the desired stability or performance properties. It is only the elapse of a certain time-period (the sampling period) that determines the triggering of the next control task. For instance, in Fig. 2 the communication between the sensor and controller (which is assumed to be a costly and scarce resource in this particular setup) is triggered by a clock resulting in equidistant transmissions along the time axis irrespective of the actual need to communicate certain sensor information. Clearly, periodic execution of control tasks can result in a significant waste of valuable system resources, as tasks are executed even if it is not needed to do so in order to guarantee the stability and performance specifications. As a consequence, one may want to reconsider the classical time-triggered periodic control paradigm in the resources for executing the control tasks are limited. In such cases, aperi control strategies that allow the inter-execution times of control tasks to vary in time are potentially better equipped to handle these constraints compared to

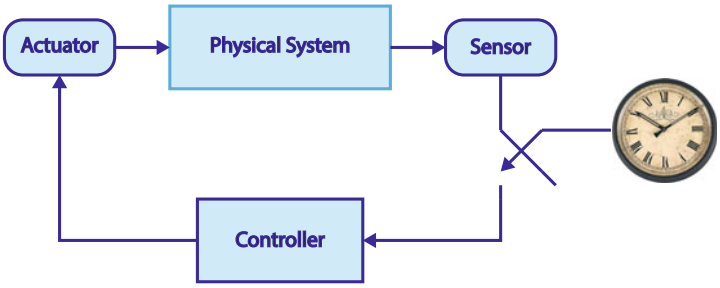

Fig. 2. Periodic time-triggered control.

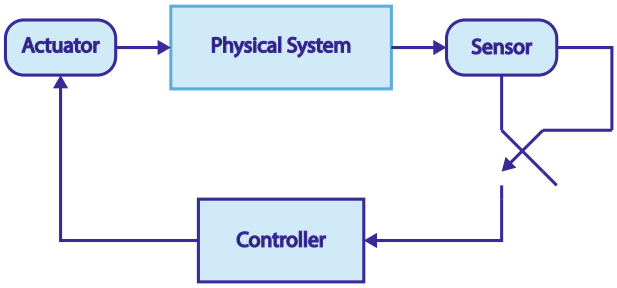

Fig. 3. Event-triggered control.

time-triggered control. In this talk we discuss two aperiodic control strategies being event-triggered and self-triggered control, see [10] for a recent overview.

In *event-triggered control*, see, e.g., [5,6,9,11–13] and the references therein, executions of control tasks are triggered by well-designed events involving the system's state, output, or other available information in an attempt to bring feedback in the sensing, communication, and actuation processes. As an example, in the setup of Fig. 3 the sensor could decide to transmit a measured output to the controller only when the current measurement deviates significantly from the previously transmitted value.

In *self-triggered control*, see, e.g., [1,8,14] and the references therein, the next execution time is precomputed at the current execution time based on predictions using previously received data and knowledge of the system's behaviour. This is illustrated in Fig. 4. The controller determines the next transmission/execution time for the sensor. Interestingly, in between execution times the sensor and the controller can go to 'sleep'. They only have to wake-up again at the next execution time. Clearly, this is beneficial for saving valuable system's resources, certainly when battery-powered devices such as wireless sensors are used. Note that in both event-triggered and self-triggered control the discrete decision is related to determining whether or not to transmit (at a certain time) and the continuous decisions are related to the selection of the control inputs.

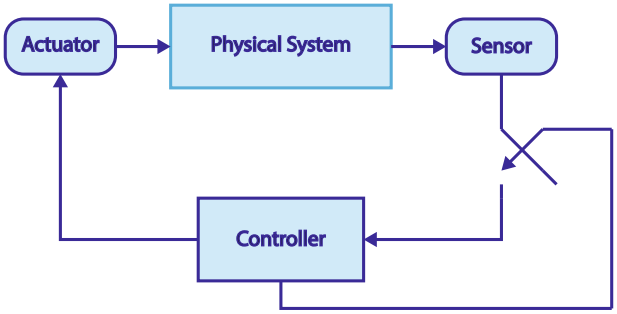

Fig. 4. Self-triggered control.

4 Overview

An interesting observation is that both approaches discussed in Sects. 2 and 3 lead to the situation that individual control tasks are no longer executed in a periodic time-triggered fashion, but in an aperiodic execution pattern with varying inter-execution times, see Fig. 5. This feature forms an important distinction

Fig. 5. Paradigm shift: From periodic execution to aperiodic execution of control tasks.

Fig. 6. A platoon of vehicles that communicate wirelessly (photograph courtesy of TNO).

with respect to the conventional periodic time-triggered scheduling of control tasks. By abandoning the periodic scheduling of control tasks, the aim is to realise better trade-offs between the overall performance of the CPS and the required resource utilisation.

For both approaches we discuss some of the main results and illustrate them by various applications in cooperative driving (Fig. 6), robotics, control of inverted pendulums, and fast mixing of polymers. We will also discuss open questions and interesting challenges for the future.

References

1. Anta, A., Tabuada, P.: To sample or not to sample: self-triggered control for nonlinear systems. IEEE Trans. Autom. Control **55**(9), 2030–2042 (2010)
2. Antunes, D., Heemels, W.P.M.H.: Performance analysis of a class of linear quadratic regulators for switched linear systems. In: IEEE Conference on Decision and Control (CDC), pp. 5475–5480, December 2014
3. Antunes, D., Heemels, W.P.M.H.: Rollout event-triggered control: beyond periodic control performance. IEEE Trans. Autom. Control **59**, 3296–3311 (2014)
4. Antunes, D., Heemels, W.P.M.H., Hespanha, J.P., Silvestre, C.: Scheduling measurements and controls over networks - Part II: Rollout strategies for simultaneous protocol and controller design. In: American Control Conference (ACC) 2012, pp. 2042-2047 June 2012
5. Arzén, K.-E.: A simple event-based PID controller. Prepr. IFAC World Conf. **18**, 423–428 (1999)
6. Aström, K.J., Bernhardsson, B.M.: Comparison of periodic and event based sampling for first order stochastic systems. In: Proceedings of IFAC World Conference, pp. 301–306 (1999)
7. Brockett, R.W.: Minimum attention control. In: IEEE Conference on Decision and Control (CDC), pp. 2628–2632 (1997)
8. Gommans, T.M.P., Antunes, D., Donkers, M.C.F., Tabuada, P., Heemels, W.P.M.H.: Self-triggered linear quadratic control. Automatica **50**, 1279–1287 (2014)
9. Heemels, W.P.M.H., Gorter, R.J.A., van Zijl, A., van den Bosch, P.P.J., Weiland, S., Hendrix, W.H.A., Vonder, M.R.: Asynchronous measurement and control: a case study on motor synchronisation. Control Eng. Pract. **7**(12), 1467–1482 (1999)
10. Heemels, W.P.M.H., Johansson, K.H., Tabuada, P.: An introduction to event-triggered and self-triggered control. In: IEEE Conference on Decision and Control (CDC), pp. 3270–3285, December 2012
11. Heemels, W.P.M.H., Sandee, J.H., van den Bosch, P.P.J.: Analysis of event-driven controllers for linear systems. Int. J. Control **81**(4), 571–590 (2008)
12. Lunze, J., Lehmann, D.: A state-feedback approach to event-based control. Automatica **46**, 211–215 (2010)
13. Tabuada, P.: Event-triggered real-time scheduling of stabilizing control tasks. IEEE Trans. Autom. Control **52**, 1680–1685 (2007)
14. Velasco, M., Fuertes, J.M., Marti, P.: The self triggered task model for real-time control systems. In: Proceeding of IEEE Real-Time Systems Symposium, pp. 67–70 (2003)