Case studies in event-driven control*

J.H. Sandee[1], W.P.M.H. Heemels[1,2] and P.P.J. v.d. Bosch[1]

[1] Eindhoven Univ. of Techn., P.O. Box 513, 5600 MB Eindhoven, The Netherlands.
[2] Embedded Systems Institute, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
E-mail: {j.h.sandee,m.heemels,p.p.j.v.d.bosch}@tue.nl

Abstract. The majority of research in control engineering considers periodic or time-triggered control systems with equidistant sample intervals. However, practical cases abound in which it is of interest to consider event-driven control systems, where the sampling is event-triggered. Although there are various benefits of using event-driven control like reducing resource usage (e.g. processor and communication load), their application in practice is hampered by the lack of a system theory for event-driven control systems. In this paper we present two types of event-driven controllers and show their potential via industrially relevant case studies and indicate initial theoretical results.

1 Introduction

The majority of research in digital control theory and engineering considers periodic or time-driven control systems in which continuous-time signals are represented by their sampled values at a fixed sample frequency. This leads to equidistant sampling intervals for which the analysis and synthesis problems can be coped with by the vast literature on sampled-data systems.

In most applications, these digital control algorithms are implemented in a real-time embedded software environment. As a consequence of the time-driven nature of controllers, control engineers pose strong, non-negotiable requirements on the real-time implementations of their algorithms as the required control performance can only be guaranteed in this manner. In the end, this leads to non-optimal solutions if the design problem is considered from a multi-disciplinary system perspective. As an example, time-driven controllers perform control calculations all the time at a fixed high rate, so also when nothing significant has happened in the process. This is clearly an unnecessary waste of resources like processor usage and communication bus load. As a consequence, a time-driven controller might not be optimal, when considered in a broader sense.

To reduce the severe real-time constraints imposed by the control engineer and the accompanying disadvantages, this paper proposes to drop the strict

* This work has been carried out as part of the Boderc project under the responsibility of the Embedded Systems Institute. This project is partially supported by the Dutch Ministry of Economic Affairs under the Senter TS program. This research was sponsored by the European 6th Framework Network of Excellence HYCON (contract number FP6-IST-511368).

requirement of equidistant sampling. The claim is that this enables better multidisciplinary trade-off making to achieve a better overall system performance. This type of feedback controllers are called *event-driven controllers* as it is an event (e.g. the arrival of a new measurement), rather than the elapse of time, that triggers the controller to perform an update. As event-driven control loops typically deal with discrete events with strong interaction with the continuous-time dynamics of the plant, they can be considered as a specific class of hybrid systems.

In literature, only few examples of event-driven control have been presented and hardly any theory on control performance analysis can be found. Two good overviews can be found in [1] and [2]. To stimulate research in this direction, we consider in this paper two types of event-driven controllers and show their potential via case studies and indicate initial ideas for analyzing the resulting loops. Further details and related research can be found in [3].

2 Event-driven control for reducing resource usage

We consider the system described by

$$\dot{x}(t) = A_c x(t) + B_c u(t) + E_c w(t) \tag{1a}$$

$$u(t) = F x(\tau_k), \qquad \text{for } t \in [\tau_k, \tau_{k+1}) \tag{1b}$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ the control input and $w(t)$ the unknown disturbance, respectively, at time $t \in \mathbb{R}_+$. As a controller a discrete-time state-feedback controller with gain $F \in \mathbb{R}^{m \times n}$ is considered, i.e. $u_k = F x_k$, where $x_k = x(\tau_k)$, $u_k = u(\tau_k)$ using the zero-order hold $u(t) = u_k$ for all $t \in [\tau_k, \tau_{k+1})$.

The *control update times* τ_k are in conventional time-driven control related through $\tau_{k+1} = \tau_k + T_s$, where T_s is a fixed sample time, meaning that the control value is updated every T_s time units. To reduce the number of control calculations, we propose not to update the control value if the state $x(\tau_k)$ is contained in a set $\mathcal{B}$ around the origin. The control update times are now

$$\tau_{k+1} = \inf\{jT_s > \tau_k \mid j \in \mathbb{N},\ x(jT_s) \notin \mathcal{B}\}. \tag{2}$$

The control objective is a “practical stabilization problem” in the sense of controlling the state towards a region Ω close to the origin and keeping it there, as asymptotic stability cannot be obtained because the plant is operated in open-loop inside the set $\mathcal{B}$.

The performance of this novel control strategy is addressed in terms of ultimate boundedness and guaranteed speed of convergence. Depending on the particular event-triggering mechanism used for the control updates, properties like ultimate boundedness for the perturbed event-driven linear system can be derived either from a perturbed discrete-time linear system or from a perturbed discrete-time piecewise linear (PWL) system. Since results for ultimate boundedness are known for discrete-time linear systems and piecewise linear systems, these results can be carried over to event-driven controlled systems. In this way we can tune the parameters of the controller to obtain satisfactory control performance on one hand and low processor/communication load on the other. An initial experimental case study in paper flow control in a printer investigates the achievable reduction in the processor load by the particular type of event-driven controllers proposed here [3, Ch. 6]. In the typical case study, the processor load can be reduced with 50% without sacrificing the control performance significantly.

3 Sensor-based event-driven control

A second line of event-driven control is sensor-based control, which is related to the situation in which the measurement method is intrinsically event-based in nature. Examples are e.g. internal combustion engines that are sampled against engine speed; level sensors for measuring the height of a fluid in a tank; and transportation systems where the longitudinal position of a vehicle is only known when certain markers are passed. We will introduce sensor-based event-driven control via a typical and industrially relevant example of motor control, although the lines of reasoning in this section are more general.

In the case study, we use an (extremely) low resolution encoder to measure the angular position of a motor. The event-driven controller is designed such that actuation is performed at the detection of an encoder pulse. In this way, the controller can use the *exact* position measurement, and is not affected by the quantization errors of the encoder. Moreover, the controller can respond fast to measurement data. When the motor is not running at constant velocity, the updates are not equidistant in time. It is therefore not possible to use classical design methods which assume that updates are equally spaced in time. We can however apply variations on classical design methods if we define our models of the plant and the controller in the (angular) position domain instead of the time domain. This idea is based on the observation that the encoder pulses arrive equally spaced in the position domain. It is shown that, by applying this event-driven controller, we not only decrease the encoder resolution - and therefore the system cost price - but also the average processor load, compared to the originally applied controller in industry. This was accomplished without degrading the control performance. In the typical example of a motor controller applied to transport images through a printer we could accurately control the motor by means of an encoder with a resolution of only 1 pulse per revolution, with the controller running at an average sample frequency of 62 Hz. Compared with the originally applied controller, running at a constant sample frequency of 250 Hz in combination with an encoder resolution of 12 pulses per revolution, the processor load was reduced by a factor 5.

4 Conclusions

Although in many practical control problems it is natural and logical to use event-driven controllers, their application is scarce in both industry and academia. A major reason why time-driven control still dominates is the absence of a system theory for event-driven control loops. To stimulate research in this direction, this paper presented two types of event-driven controllers with a clear industrial relevance. Given the potential benefits of such controllers as shown in the case studies, we believe it is worthwhile to invest research effort in this line of work and to develop a mature theory for event-driven control systems.

References

1. Årzén, K.E.: A simple event-based PID controller. In: 14th World Congress of IFAC. Volume 18., Beijing, P.R. China (1999) 423–428
2. Åström, K.J., Bernhardsson, B.M.: Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In: 41st IEEE Conference on Decision and Control. (2002) 2011–2016
3. Sandee, J.H.: Event-driven control in theory and practice - trade-offs in software and control performance. Phd thesis, Eindhoven Univ. of Techn., Netherlands (2006)