Event-Based Control Enters the Real-Time World: Perspectives and Pitfalls

Silvano Seva
DEIB, Polytechnic University of Milan, Italy
silvano.seva@polimi.it

William Fornaciari
DEIB, Polytechnic University of Milan, Italy
william.fornaciari@polimi.it

Alberto Leva
DEIB, Polytechnic University of Milan, Italy
alberto.leva@polimi.it

Abstract

In the last years, event-based control techniques have been gaining a steadily increasing importance owing to the advantages they bring, such as reduced network traffic, low actuator wear, reduced energy consumption of the involved devices. Applying the event-based paradigm in the context of real-time control opens up new opportunities, but introduces new challenges as well. In this paper we provide an overview of both opportunities and challenges, outlining the major problems to be tackled and as a consequence future research directions.

2012 ACM Subject Classification Computer systems organization → Real-time system architecture; Computer systems organization → Embedded systems

Keywords and phrases Real-time control, Wireless control, Event-based control, Cyber-physical systems, Industrial control networks, Industry 4.0

Digital Object Identifier 10.4230/OASIcs.NG-RES.2021.4

1 Introduction and motivation

In the last years, the industrial environment has been characterised by the emergence of the "wireless factory" concept, fostered by paradigms like the Industry 4.0 (I40) and the Industrial Internet of Things (IIoT) ones. A prominent argument to promote the wireless factory idea is a strong reduction of cables, and therefore the mitigation of the related problems concerning for example – and most notably – installation and maintenance.

A key feature of the so emerging *scenario* is the use of wireless communication techniques also for control applications [4, 8, 12], where real-time requirements inevitably come into play. As such, the advantages of wireless communications come at a cost in terms of

1. tighter energy efficiency requirements, as in many cases cabling reduction and system layout reconfigurability call for battery-operated devices,
2. and increased criticality of band occupation, as one transmission medium can host a large number of applications, some of which requiring real-time guarantees in terms of latency, data rate, and so on.

Event-Based Control (hereinafter EBC for short) helps mitigating these issues by transmitting measurement and control data "only when needed". The consequent energy saving is quite evident, as a notoriously battery-killing action for wireless devices is the exchange of data, due to the high power demand of the radio transceiver. Not equally obvious are the advantages as for band occupation, as these are in fact relevant only when slack reclamation techniques are employed, allowing other applications to reuse time slots temporarily made empty as some transmission "was not needed".

Second Workshop on Next Generation Real-Time Embedded Systems (NG-RES 2021).
Editors: Marko Bertogna and Federico Terraneo; Article No. 4; pp. 4:1–4:11
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

It is quite intuitive to notice, as our main point, that EBC has a potentially strong impact on real-time properties, no matter how formulated. As we shall discuss later on, giving up the periodic transmissions of fixed-rate digital control makes end-to-end latency (from the occurrence of a physical event in the controlled object till the physical reaction on that object) involve new phenomena, deeply intertwined with the synthesis of a control law.

Said otherwise, from a real-time perspective, EBC couples control algorithm, processor and network scheduling in a much tigher and more complicated manner than fixed-rate control does. As will be shown, for example, the idea itself of latency (a very typical subject of real-time requirements) needs extending to distinguish a "cyber" latency – the one addressed in the mainstream real-time literature – and a "cyber-physical" latency. The latter heavily depends not only on the workload required by the control law, but also on the way that law is conceived and tuned – an important subject in industrial control, see e.g. [5, 22] – and even on how the corresponding algorithms invoked by the event-generation mechanism. Needless to say, therefore, "real-time EBC" poses more than one challenging problem.

In this paper, continuing the research presented in [23], we analyse the real-time EBC *scenario* from the control theory and engineering viewpoint, but with an eye on the underlying architecture and technology as this is made necessary in the light of the considerations just reported, evidencing some of the new challenges arising when EBC systems coexist with other real-time applications and proposing possible solutions.

2 Event-based control in a nutshell

The core idea of EBC is to act on the controlled system not periodically, as in standard digital control, but "only when necessary". Many meanings can be attributed to this idea of "necessity", and since we are not providing here a complete treatise but just the bare necessary for this paper, we only describe the so-called "Send on Delta" one (SoD for short) as it is the most widely used in the applications. The very intuitive operation of a SoD sampler sensor, that triggers a control action by transmitting a new sample of the controlled variable when this "has varied enough", is illustrated in Figure 1 and its caption.

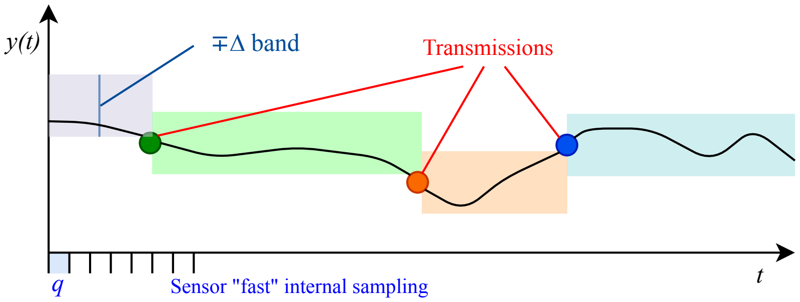

Figure 1 Send on Delta (SoD) sampler operation in the periodic case – the sensor transmits a sample of the controlled variable $y(t)$ at the first integer multiple of the time quantum q where it differs in magnitude more than Δ from the last transmitted one.

In practice, one typically completes the mechanism in Figure 1 with a timeout, i.e., makes the sensor transmit a new sample unconditionally after a number N_{to} of *quanta* since the last transmission. Besides possible influences on stability that we are not discussing herein, this is an intuitively necessary means to watch over the sensor and check it stays alive.

Rigorously speaking, we are here limiting the scope to *periodic* EBC, as in the most general theory events are not constrained to occur at multiples of any time quantum (contrary to what we are assuming here right from the explanatory Figure 1). However, given the clocked nature of any digital computing system, the periodic EBC context is general enough for us. The reader willing to deepen his/her knowledge can refer e.g. to the recent survey [2] and its huge bibliography. To give here just a rapid idea about how beneficial a properly designed EBC can be in terms of saved transmissions, Figure 2 reports a snapshot of the operation of a properly tuned EBC loop in the presence of typical measurement noise; some brief explanatory comments are provided in the caption.

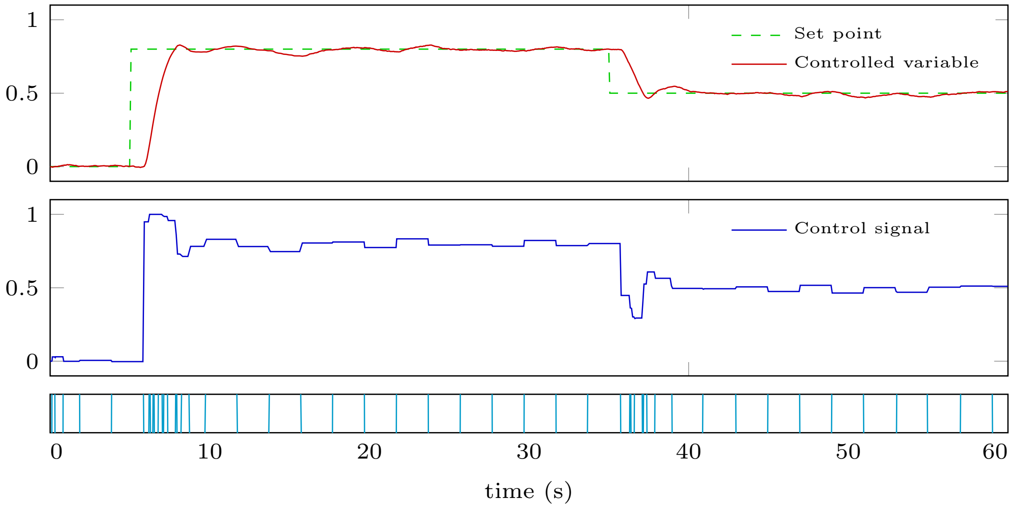

Figure 2 EBC in action: set point and controlled variable (top), control signal (centre) and events (bottom); the transmission saving with respect to fixed-rate control – where these would have to always occur at the fastest pace observed *and needed* during rapid signal variations – is apparent; periodic events when the system is (almost) at rest are due to SoD timeout, and their slow pace would not suffice to keep the loop under proper control in the face of significant *stimuli*.

As just said, however, EBC needs to be designed properly, or disasters can occur owing to the controller not acting timely. From the methodological standpoint, the main issue with EBC (also in the periodic case) is that the classical theory of fixed-rate sampled-data digital control ceases to apply, as the time span in between two subsequent control computations is not constant. In fixed-rate control, ensuring stable and correct operation of the closed-loop system ultimately calls for a proper choice of the sampling period, and there are established techniques for this purpose. In our EBC context, the role of the sampling period is played cooperatively by two actors, namely the time quantum q and the parameters pertaining to event generation (in SoD, the threshold Δ and the timeout N_{to}).

In extreme synthesis, assuming that the control design process follows the very common *modus operandi* to first determine a continuous-time controller and then its digital realisation, obtaining the latter in the (periodic) EBC context means

1. choosing the event generator parameters in such a way that events are generated frequently enough to ensure closed-loop stability and to not excessively deteriorate performance with respect to that ideally provided by the continuous-time controller, while at the same time avoiding too frequent spurious events (owing typically to measurement noise) to not excessively stress communciation channels, controller and actuator;
2. converting the continuous-time controller to a discrete-time one suitable for updating its output and state in steps that are not uniformly spaced in time (although distances are quantised) to get the required control algorithm.

An important research domain concerns extending tuning techniques conceived for fixed-rate control realisations, which for industry standard controllers form a large *corpus* as shown e.g. in [22], to serve for EBC. But in addition to this research, that concentrates on syntesising the control algorithm rather than on the underlying architecture, for our purposes it is worth here noticing that EBC quite apparently revolutionises the usage of computing and network resources. An immediately noticeable fact is that the said usage is intrinsically non uniform and can exhibit hard-to-predict bursts, but if one focuses on the real-time context, there is more. Analysing the so emerging *scenario* is the subject of the following sections.

3 New challenges

As mentioned in the introduction, abandoning the “classical” fixed-rate digital control techniques in favour of EBC, alongside the previously outlined advantages, poses new challenges from the technological point of view. Some are in fact variations or enhancements of already known ones, for example in the domain of mixed criticality, while others are specific to the EBC context. Among the relevant ones, we evidence here a wider variability of the control latency, tight requirements in terms of network synchronisation and significant impacts on the schedulability of control tasks. In this section we analyse in detail these issues, compatibly with space limitations.

Control latency

Event-based control affects latency “as seen by the plant”, that is, the amount of time since some fact occurs till the controller reacts. Consider, for example, the case of a disturbance applied to the process input: if a fixed-rate control scheme is used, the controller surely reacts to the disturbance starting at the first control step immediately following the time instant when the disturbance effect becomes visible on the controlled variable. From this we have that the control latency, as seen by the process, is bounded and at most equal to the width of one additional time step with respect to the process natural response delay. In the case of EBC, on the contrary, the controller reaction time is affected by various factors, like the mechanism used to generate events, the presence of a timeout, and others.

Following this brief analysis, we can introduce the concept of “cyber-physical control latency”, explicitly denoting the influence on the controller response time of two different components, one attributable to both the network and computing infrastructures - the cyber part - and the other one - the physical one - to the process structure, through the event-triggering mechanism.

In the case of a fixed-rate control system, the contribution to the overall control latency of the physical component is negligible with respect to the cyber one, since the controller will react to any change in the process output within, at most, one time step, irrespectively of the process internal state, the extent of output and measurement noises and so forth. From the point of view of control system’s stability and performance, this represents an good situation: once the controller step rate has been properly computed and estimates of the amount of network-induced delay are available, the impact of the control latency on phase margin (one of the key parameters which allow to describe both control stability and performance) can be straightforwardly computed and, eventually, compensated with *ad-hoc* techniques.

On the other hand, with event-based control systems the situation is more difficult: here, the physical component has a significant contribution on the overall amount of the control latency, also increasing its temporal variability. This physics-induced latency is ascribable

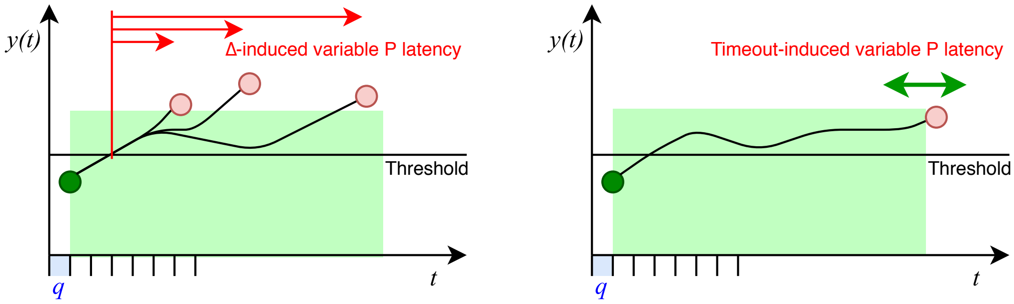

Figure 3 Example of variable physical latency in the case one has to detect when some variable $y(t)$ exceeds a threshold: P latency variability can come from both the SoD threshold Δ (left) and the timeout N_{to} (right); in either case, moreover, the process state has influences the P latency variability by dictating – together with the inputs – the variation rate of $y(t)$.

to different phenomena, two of which are now described: one of them, depicted in the left part of Figure 3, is the rate of variation of the process output variable, determining the time span between the instant when a perturbed movement of the process output arises due to an external disturbance and the one when a corrective control action is applied, this one triggered by the process output variable crossing the threshold inside the event generator.

The other one, shown in the right part of Figure 3, is constituted by process perturbations causing its output variable to have an erratic behaviour, but inside the event-triggering dead band: given that the event-triggering threshold has to be chosen appropriately also considering the maximum tolerable deviation from the reference signal, such phenomena could anyway be detrimental in terms of control performance. In this situation, the amount of time before a corrective control action is applied depends on the characteristics of the timeout mechanism - which is always a good idea to have, to avoid running the closed-loop system in open loop for excessive amounts of time - periodically and unconditionally triggering a new control action: in the worst case, the one when the perturbation begins immediately after the last timeout-triggered control action has been applied, the control latency amounts to one full timeout period. This problem is apparently an EBC peculiarity, relevant for real time control as it pertains to latency.

Synchronisation

When an EBC scheme is employed, communications between sensors and controller and/or controller and actuators can be infrequent, however their timing must be always precise enough to allow them to happen properly. Although in a fixed-rate control application there are plenty of signal fronts to keep all the network nodes synchronised, in an event-based one this may well not be the case: consider, in this respect, that the time between two subsequent control computations, also in the fixed-rate case, is in general far larger than the time scale of network protocol transactions.

Another fact worth noticing, as testified by the major industry standards [17, 14, 28][1], is that Time Division Multiple Access (TDMA) or even polling-based access schemes are widely adopted when dealing with communication networks for control applications. Since in periodic EBC adopting TDMA (polling would make no sense) implies that a slot must be

---

[1] There are exceptions like e.g. CAN, but mostly limited to vehicular applications and wired settings, where the problem of minimising the radio listening consumption does not exist.

reserved for each *possible* transmission, bandwidth saving can be jeopardised unless some slack reclamation technique is in place for non time-critical traffic to opportunistically occupy unused slots.

The consequent need for the originators of that traffic to carry out a reliable clear channel assessment apparently tightens the synchronisation needs. Also, in fixed-rate control missing one sample of the controlled variable is an information loss immediately cured by the next one. In EBC there can be no "next one" for a long time, causing highly undesired behaviours. As such, this is an example of pre-existing problem exacerbated by EBC.

Schedulability of control tasks

When dealing with task schedulability in a real-time system, the presence of event-based control tasks in a task set can have non negligible effects in terms of schedule feasibility and system overload. Before better analysing these effects, let us consider, for a comparison, the case of a task set containing only fixed-rate control tasks: here, to each controller (or group of controllers) corresponds a periodic task with a fixed and know execution period and, from the schedulability point of view, the problem is the "classic" one of finding a feasibile schedule for a given set of tasks. On the other hand, event-based techniques, due to their underlying principle of acting on the plant "only when needed", pose some concerns for what regards the scheduling of CPU resources: each control task maintains its requirements in terms of guaranteed periodic execution, but its contribution to the overall CPU load is not constant, since it will be executed only sporadically and for a limited amount of time.

When EBC tasks are present, then, the feasibility of a task schedule enters the "cyber-physical" domain: the same physical phenomena that affect the control latency, shown in the previous point, can strongly determine the time distribution of an EBC task. In this regard, we also introduce the concept of "event storm": it may happen, during the normal operation of the system, that a large number control tasks, if not *all* of them, are simultaneously woken up due as a consequence of some external physical phenomena causing the generation of controller events. When an event storm occurs, CPU utilisation factor suddenly spikes up to a value which can be greater than 100%, causing some tasks to miss their deadline.

The key point of such a phenomenon lays in the fact that an event storm is caused by some event happening in the physical world, and these events are substantially not predictable: this means that new techniques to ensure proper task schedulability in presence of EBC applications have to be devised. Similarly to what can be done with TDMA slots, however, the aperiodic nature of the EBC tasks brings into play also some advantages, allowing to achieve better CPU and network utilisation by re-assigning the otherwise unused CPU time.

Summing up, the EBC context complicates *a priori* architecture sizing, because utilisation bursts can be much higher then in fixed-rate control, and the inherently sporadic (but possibly transiently concentrated) events, make it hardly possible to figure out hyer-periods to ground task allocation upon. Said otherwise, EBC strongly affects – to not say just breaks – the customary connection between real-time and periodic tasks, turning the exception of a latency-constrained non-periodic task – at least as long as control is the purpose – into the normal case.

4 Proposals

The sporadic behaviour of EBC tasks, occupying network and computing resources only when some corrective control action has to be applied, can be favourably exploited to improve the performances of the computing infrastructure they are based on, for example exploiting the approach proposed in [20]. In this section we develop our treatise by analysing the possible earnings that can be gathered when dealing with schedulability of either CPU or network resources.

Concerning CPU utilisation, the underlying principle of EBC stating that the controller is run “only when needed” results in the fact that is no more necessary to have an always running periodic task for each controller: although the constraint of ensuring enough CPU time to each (event-based) control task is still present - to not affect the stability and performance of the closed-loop system - there is now the possibility to re-allocate the otherwise unused CPU time to other tasks whenever the corresponding controller is in idle state. This, evidently, is not the case with fixed-rate controllers: since, in this case, the closed-loop system to which they belong is not designed to be run in open loop - or, at least, the safety of such a behaviour is not guaranteed -, a value for the control action has to be computed at each time step, even when the control error is zero. However, as described in Section 3, event-based control tasks can also have detrimental effects on the feasibility of a CPU schedule in occurrence of event storms. In this respect, two different approaches are possible: the first one, conservative, proceeds by considering all the event-based controllers as fixed-rate ones and then requiring to the schedulability analysis to guarantee that the overall task set never reaches a CPU utilisation factor greater than 100%. With this approach, an event storm simultaneously activating all the control tasks does not have a destructive impact on the overall system performance, while keeping the possibility of reusing empty CPU slots for other non-critical tasks.

The second approach, applicable when some degradation in the control performance is tolerable, calls for a subdivision of the event-based control tasks in two sets, whether a degradation of control performance is acceptable or not. The outcomes of this subdivision, then, provide some room to safely undersize the required computing resources by making the scheduling system structured such that, in case of CPU overload - i.e. due to an event storm - the tasks associated to the control loops accepting a performance degradation are not always assigned their CPU time, leaving computing resources to the other, more critical, tasks.

The same principle of re-allocating the otherwise unused time slots can be applied to network resources too. Data transmission on control networks is often managed through TDMA schemes so as to have an almost constant and known in advance (cyber) control latency: since each control task is uniquely assigned a transmission slot, all the variable delays introduced by collisions and access contention are automatically removed. Especially with battery-operated wireless devices, using event-based control techniques coupled with a well-synchronised TDMA scheme allows to reach remarkable energy savings, enhancing the devices’ operating time and reducing the maintenance costs. These advantages, however, are counterbalanced by a poor utilisation of the wireless transmission medium: the fact that each transmission slot is uniquely assigned to a control task means that it cannot be reused whenever the event-based control system is in the idle state, a situation which happens quite often and for significant periods of time. To avoid wasting this precious bandwidth, various techniques can be used. One possibility is implementing a slack reclamation technique, making each network node capable of detecting, for each time slot, if the slot assignee is effectively transmitting data: if not, that otherwise empty slot can be reused for other transmissions. For such a mechanism to be feasible, however, a very precise synchronisation among all the nodes is required, such that the residual synchronisation error is well below the width of a time slot. Another observation has to be made about which kind of data can be effectively exchanged through the re-used slots: given how this mechanism works, data exchange through these slots is affected by a wide temporal variability in terms of available bandwidth and latency, both depending on how many slots are effectively available in a transmission round. Given these characteristics, then, data exchanged through otherwise empty slots cannot have strict requirements in terms of real-time performance: this, however, leaves room for all the data transmissions serving ancillary functionalities always present in an industrial plant, such as non-critical monitoring of process parameters, and signalling and so on.

Another way for a more efficient utilisation of the available bandwidth from the event-based control tasks is abandoning the TDMA scheme in favour of a CSMA (Carrier Sense Multiple Access) one: instead of having uniquely-assigned time slots also for the event-based control tasks, resulting in the aforesaid bad utilisation of the available bandwidth, the communication mechanism can be made such that all the packets containing data for event-based control are exchanged through opportunistic time slots, where more than one network node attempts to transmit its payload. It is not a mystery that this scheme has effects on the control latency: lost the guarantees given by a TDMA scheme, the value of the network-induced latency has to be determined on a probabilistic basis in terms of "worst case cyber-physical latency". This expression poses the accent on the fact that the overall transmission latency is both due to network characteristics and physical phenomena affecting the access contention to the transmission medium and the generation of events starting a data transmission. If we go deeply into the problem, however, we have to observe that control applications are more tolerant to latency issues with respect to others like, for example, signal processing ones: while in the second case a non-tight latency bound can disrupt the final results (think to the case of an FFT task: a change in the sample arrival rate shifts the resulting spectrum), in a control loop the value of latency bounds appears more indirectly, in terms of stability degree, absence of oscillations, small response time deterioration, and so forth. A hybrid approach is also possible, by subdividing a transmission round in two parts: the first one managed through a TDMA scheme for all the event-based control tasks whose execution is somehow critical and requires for strict bounds on the variability of transmission latency and the remaining one accessed with a CSMA technique, for all the tasks able to tolerate a wider variability of the transmission latency.

5 Related work

On the systems and control front, EBC dates back to pioneering works such as [3], where the idea of lightening the control network load was proposed and developed on a significantly heuristic basis. Methodological studies on the properties of such newly introduced loops came in the following decades and yielded neat results, see e.g. [19, 27], while the influences of the EBC framework on the synthesis of controllers came into the research *arena* [15, 16]. The presence of EBC also required new models for the network as seen "externally" by controllers in terms of dynamic systems [30]. At the same time, pilot and research-targeted realisations started appearing – see for example [10] and several analogous works – paving the way to addressing real industrial cases [9, 8].

As already said, a recent and complete survey on the overall subject is [2], while another one more geared to industrial applications can be found in [11]. Considering this huge research *corpus*, the main conclusions for the purpose of our research is that powerful analysis methods are nowadays available for EBC, but the intertwined effects of event triggering mechanism and control algorithm are still being explored, so that in fact a systematic approach to tuning event-related parameters together with those of a control law is still in its infancy – especially if industry-standard solutions are sought, specifications are tight, the cost of resources makes rules out over-provisioning *a priori*, or any combination of the above.

On the technological side, the problem of resource scheduling in presence of both period and aperiodic tasks has been already analysed in the past - see, for example, works like [18] and [13]. On this basis, a good starting point for future works aiming to both improvements to EBC task schedulability and a more efficient utilisation of CPU time left free by idle EBC tasks is constituted by the current state of the art on the schedulability of sporadic tasks, with works like [7, 21, 6].

From the network scheduling point of view, instead, EBC represents a completely new use case for the current state of the art, for a variety of reasons. The first one is the management of latency: while in fixed rate control the addition of one time step to the estimated - or measured - cyber latency is a correct overbound for the total cyber-physical latency, with EBC this assumption cannot be held true anymore due to the presence of a strong physical component influencing the total control latency. To this extent, Figure 3 shows two notable cases. Coming to the transmission protocols for control networks, EBC ideally requires for schemes allowing for non periodic data transmission but without queues: an unusual requirement for a transmission protocol from both the cyber and cyber-physical points of view. The *rationale* for such a requirement resides in the fact that, for a control system, a measurement sample arriving “late” to the controller becomes useless, since it conveys information about an old state of the process, which in the meantime has surely changed. From this point of view, a network protocol without queues dropping the old packets of favour of newer ones represents a more effective situation. Given such a *scenario*, the presence of a valid scheme for the synchronisation among the network nodes allows for the implementation of suitable protocols. To this aim, works like [26] and its successive extensions [24, 25] provide a valuable ground for future developments.

On a wider perspective, the current research activity concentrates on the IIoT paradigm and on event-based wireless communication [1, 29] but, to the best of the authors’ knowledge, the problem of cyber-physical latency is hardly mentioned, let alone of a formalisation of the connected problems.

6 Conclusions

Using event-based techniques for process control brings in numerous advantages, especially when battery-operated wireless sensors and actuators are involved. However, from the technological point of view, applying such techniques in a real-time context poses new and important challenges: in this paper we have briefly analysed these issues with a focus on both computational and network resources, showing the impact of EBC tasks on the feasibility of a CPU schedule and the existing trade-offs between energy and bandwidth saving. Another relevant point is constituted by control latency, which becomes more dependent on the physical phenomena inherent with the process being controlled. To this aim, we have introduced the concept of “cyber-physical control latency” and detailed the nature of its cyber and physical components.

Following the analysis of these new challenges, we have outlined some solutions allowing for a safe implementation of EBC techniques in a real-time context, also pointing towards a better utilisation of both CPU and network resources through slack-reclamation techniques. Future work in this direction points towards a deeper analysis of the issues here presented followed by the devise of adequate methods for reclaiming the otherwise unused resources.

References

1 G. Aceto, V. Persico, and A. Pescape. A survey on information and communication technologies for Industry 4.0: state-of-the-art, taxonomies, perspectives, and challenges. *IEEE Communications Surveys & Tutorials*, 21(4):3467–3501, 2019.

2 E. Aranda Escolástico, M. Guinaldo, R. Heradio, J. Chacon, H. Vargas, J. Sánchez, and S. Dormido Bencomo. Event-based control: A bibliometric analysis of twenty years of research. *IEEE Access*, 8:47188–47208, 2020.

3 K.E. Årzén. A simple event-based PID controller. In *Proc. 14th IFAC World Congress*, volume 18, pages 423–428, Beijing, China, 1999.

4 S.A. Ashraf, I. Aktas, E. Eriksson, K.W. Helmersson, and J. Ansari. Ultra-reliable and low-latency communication for wireless factory automation: from LTE to 5G. In *Proc. 21st IEEE International Conference on Emerging Technologies and Factory Automation*, Berlin, Germany, 2016.

5 K.J. Åström. Event based control. In A. Astolfi and L. Marconi, editors, *Analysis and Design of Nonlinear Control Systems*, pages 127–147. Springer, Berlin, 2008.

6 S. Baruah, V. Bonifaci, G. D'Angelo, H. Li, A. Marchetti Spaccamela, S. Van Der Ster, and L. Stougie. Preemptive uniprocessor scheduling of mixed-criticality sporadic task systems. *Journal of the ACM*, 62(2):14:1–14:33, 2015.

7 S. Baruah and N. Fisher. The partitioned scheduling of sporadic real-time tasks on multiprocessor platforms. In *Proc. 2005 International Conference on Parallel Processing Workshops*, pages 346–353, Oslo, Norway, 2005.

8 T. Blevins, D. Chen, S. Han, M. Nixon, and W. Wojsznis. Process control over real-time wireless sensor and actuator networks. In *Proc. 17th IEEE International Conference on High Performance Computing and Communication*, 2015.

9 T. Blevins, M. Nixon, and W. Wojsznis. Event based control applied to wireless throttling valves. In *Proc. 1st International Conference on Event-based Control, Communication, and Signal Processing*, Kraków, Poland, 2015.

10 J. Chacón, J. Sánchez, L. Yebra, A. Visioli, and S. Dormido. Experimental study of two event-based PI controllers in a solar distributed collector field. In *Proc. 12th European Control Conference*, pages 626–631, Zürich, Switzerland, 2013.

11 M. Dotoli, A. Fay, M. Miśkowicz, and C. Seatzu. Advanced control in factory automation: a survey. *International Journal of Production Research*, 55(5):1243–1259, 2017.

12 Y. Wei et Al. RT-wifi: Real-time high-speed communication protocol for wireless cyber-physical control applications. In *2013 IEEE 34th Real-Time Systems Symposium*, pages 140–149, Nashville, Tennessee, 2013.

13 G.Lipari and G. Buttazzo. Schedulability analysis of periodic and aperiodic tasks with resource constraints. *Journal of Systems Architecture*, 46(4):327–338, 2000.

14 S.M. Hassan, R. Ibrahim, K. Bingi, T.D. Chung, and N. Saad. Application of wireless technology for control: A WirelessHART perspective. *Procedia Computer Science*, 105(supplement C):240–247, 2017.

15 D. Henriksson and A. Cervin. Optimal on-line sampling period assignment for real-time control tasks based on plant state information. In *Proc. 44th IEEE Conference on Decision and Control*, pages 4469–4474, Seville, Spain, 2005.

16 B. Hensel, J. Ploennigs, V. Vasyutynskyy, and K. Kabitzsch. A simple PI controller tuning rule for sensor energy efficiency with level-crossing sampling. In *Proc. 9th IEEE International Multi-Conference on Systems, Signals and Devices*, pages 1–6, Chemnitz, Germany, 2012.

17 D. Jansen and H. Buttner. Real-time ethernet: the EtherCAT solution. *Computing and Control Engineering*, 15(1):16–21, 2004.

18 K. Jeffay, D.F. Stanat, and C.U. Martel. On non-preemptive scheduling of period and sporadic tasks. In *Proc. 12th IEEE Real-Time Systems Symposium*, pages 129–139, San Antonio, TX, USA, 1991.

19 J. Lunze and D. Lehmann. A state-feedback approach to event-based control. *Automatica*, 46(1):211–215, 2010.

20 M. Maggio, F. Terraneo, and A. Leva. Task scheduling: A control-theoretical viewpoint for a general and flexible solution. *ACM Transactions on Embedded Computing Systems*, 13(4):1–22, 2014.

21 M. Marouf, L. George, and Y. Sorel. Schedulability analysis for a combination of non-preemptive strict periodic tasks and preemptive sporadic tasks. In *Proc. 17th IEEE International Conference on Emerging Technologies and Factory Automation*, pages 1–8, Kraków, Poland, 2012.

22 A. O'Dwyer. *Handbook of PI and PID controller tuning rules.* Imperial college press, London, United Kingdom, 2009.

23 S. Seva, C.E. Lukaschewsky Mauriziano, W. Fornaciari, and A. Leva. A low energy FPGA platform for real-time event-based control. In *Proc. 1st Workshop on Next Generation Real-Time Embedded Systems*, Bologna, Italy, 2020.

24 F. Terraneo, A. Leva, S. Seva, M. Maggio, and A. V. Papadopoulos. Reverse flooding: Exploiting radio interference for efficient propagation delay compensation in WSN clock synchronization. In *2015 IEEE Real-Time Systems Symposium*, pages 175–184, Rome, Italy, 2015.

25 F. Terraneo, P. Polidori, A. Leva, and W. Fornaciari. TDMH-MAC: Real-time and multi-hop in the same wireless MAC. In *Proc. 39th IEEE Real-Time Systems Symposium*, pages 277–287, Nashville, TN, USA, 2018.

26 F. Terraneo, L. Rinaldi, M. Maggio, A. V. Papadopoulos, and A. Leva. FLOPSYNC-2: Efficient monotonic clock synchronisation. In *Proc. 35th IEEE Real-Time Systems Symposium*, pages 11–20, Rome, Italy, 2014.

27 Y. Wang, W.X. Zheng, and H. Zhang. Dynamic event-based control of nonlinear stochastic systems. *IEEE Transactions on Automatic Control*, 62(12):6544–6551, 2017.

28 Xuepei Wu and Lihua Xie. On the wireless extension of PROFINET networks. In *Proc. 2019 IEEE VTS Asia Pacific Wireless Communications Symposium*, pages 1–5, Singapore, 2019.

29 H. Yang, K. Zhang, K. Zheng, and Y. Qian. Leveraging linear quadratic regulator cost and energy consumption for ultra-reliable and low-latency IoT control systems. *IEEE Internet of Things Journal*, 7(9):8356–8371, 2020.

30 X.M. Zhang, Q.L. Han, and Bao-Lin B.L. Zhang. An overview and deep investigation on sampled-data-based event-triggered control and filtering for networked systems. *IEEE Transactions on Industrial Informatics*, 13(1):4–16, 2016.