

# A Review of Co-Design Approaches and Tools for Control of Cyber-Physical Systems

Noraide Md Yusop<sup>1,2\*</sup>, Rosbi Mamat<sup>1</sup>, Sundet Gamzat<sup>3</sup>, Yugai Vyacheslav<sup>3</sup>, Tokhisheva Gulden<sup>4</sup> and Zaharuddin Mohamed<sup>1</sup>

<sup>1</sup>Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

<sup>2</sup>Advanced Technology Training Centre (ADTEC), Batu Pahat, Johor, Malaysia

<sup>3</sup>Department of Automation of Manufacturing Processes, Abylkas Saginov Karaganda Technical University, Karaganda, Kazakhstan

<sup>4</sup>Faculty of Energy, Saken Seifullin Kazakh Agrotechnical Research University, Astana, Kazakhstan

\*Corresponding author: [noraide@jtm.gov.my](mailto:noraide@jtm.gov.my)

Submitted 20 January 2024, Revised 30 March 2024, Accepted 15 April 2024, Available online 03 May 2024.

Copyright © 2024 The Authors.

**Abstract:** Systems that integrate between computers, communication and control systems are known as Cyber-Physical Systems (CPSs), and effective control of such systems is challenging. With advancement of the related areas and many systems are inter-connected, CPSs are becoming important elements especially in Industry 4.0 and 5.0. This paper presents a review of co-design approaches for control of CPS, in which desired performance in both layers (cyber and physical) are considered during the controller design. The approach is found to be suited well for CPS. Moreover, in this paper, a special focus is given to event-based control approaches as promising results are reported. In this case, event-based state-space and PID controllers are reviewed. In addition, tools used for the development and performance analyses of co-design approaches are discussed. Several control challenges related to CPS control have also been identified. It is envisaged that this paper can provide valuable knowledge for researchers to embark into the research of control of CPS.

**Keywords:** Co-design; Cyber-physical system; Event-based controller; PID controller; Review.

## 1. INTRODUCTION

In the first two decades of the 21st century, the technologies of computer, communication, and control systems have grown rapidly. This technological development creates sophisticated systems that involves the integration of these fields. These integration systems are called cyber-physical systems (CPSs). The term CPS was coined in 2006 by Helen Gill, the Director of Computer Information and Science Engineering (CISE) at National Science Foundation (NSF), USA [1]. Later, several definitions of CPS emerged in the literature. These include:

- CPS is a system that integrates the cyber world with the physical world, where computational and physical components of such systems are tightly inter-connected and coordinated to work effectively together, sometimes with humans in the loop [2].
- CPS is an orchestration of computers and physical systems. Embedded computers monitor and control physical processes usually with feedback loops, where physical processes affect computations and vice versa [3].
- CPSs as engineered systems that are built from, and depend upon, the seamless integration of computation and physical components to enable the capability, adaptability, scalability, resiliency, safety, security, and usability that will expand the horizons of the system [4].
- CPS is the next generation of engineered systems that require tight integration of computing, communication, and control techniques [5 - 7].

From these definitions, common characteristics for CPS are:

- Integration of cyber elements (computation, software, and networking) and physical elements (engineered systems and human factor).
- A feedback loop system which involves the physical processes, computations (simulation and decision making), sensing and actuation elements, and monitoring and control elements.
- Networked, tightly coupled, inter-connected processes and mediating between computing and physical entities.

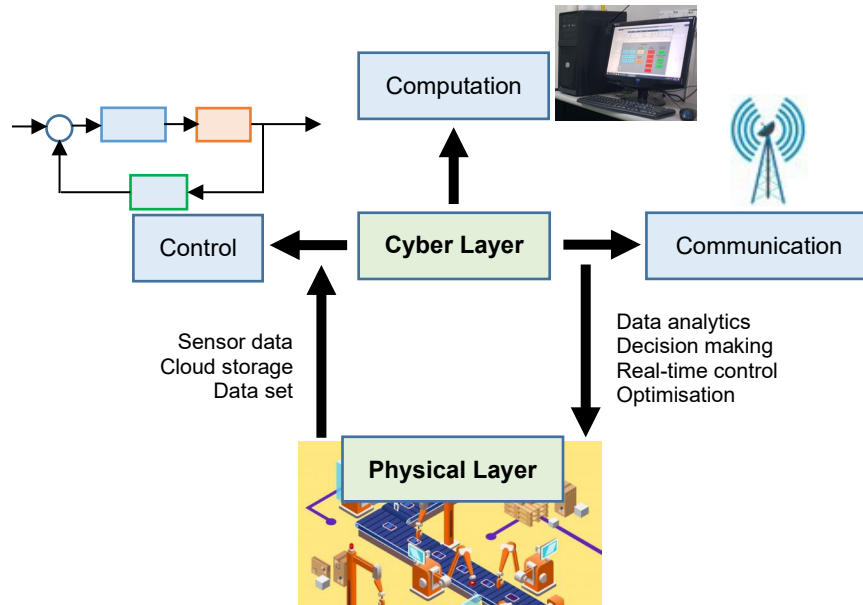


Figure 1. Relationships of CPS elements.

Generally, the fields of control, communication and computing are separate areas and have their distinct theories and bases. Analytical methods and tools for solving their problems are designed for their respective fields. However, in these days many complex systems have been developed with tight integration within each field as a result of the advanced technologies. CPS is introduced due to the increasing of systems that have integration between computation, communication and control elements (the cyber part) and physical processes such as heating, cooling, motion, vibrations, etc. (the physical part). Engineering and computing communities begin to realise the challenge in designing such systems is not due to the limited resources either from cyber or physical, but it comes from their interaction [1]. Kim and Kumar [6] have given a review on how CPS becomes an important topic of research.

Figure 1 shows the relationships between cyber and physical components for CPS which makes CPS a heterogeneous system that combines hardware, software, sensors, actuators, and other components [8]. Basically, CPS is a system that integrate several systems, such as embedded systems, networked control systems (NCS) and Internet of Things (IoT). Therefore, the analysis and design of the CPS are based on unified dynamics, which emerge from interactions between physical, computer, software, and networks [9].

Although CPS was introduced in 2006, the research on CPS become popular among researchers after it was featured in the IEEE Proceeding in 2012. Articles in this publication have sparked many ideas in the CPS research. However, most of the articles have not made a clear distinction between the CPS and NCS. In fact, the main difference between these two systems is the integration between the cyber and physical processes, where CPS requires the enhancement of the integration to provide a tight and seamless integration to be a more efficient system [3, 6]. In CPS, the enhancement of the integration is possible to implement before the control system is established on the network. A CPS work as in [10] can be an example of the integration's enhancement, where the cyber model is designed by abstracting the execution rate of a real-time feedback control and combining it into a physical continuous state space model.

As CPSs are gaining popularity among researchers and the systems are important components in Industry 4.0 and 5.0, several control approaches have been proposed and implemented. However, the CPSs are also becoming more complex especially for a large-scale system, and thus, control challenges increase. Furthermore, due to the tight integration requirements of cyber and physical spaces, an effective control technique is required to fulfill both performance goals. This paper presents a review of co-design approaches and its tools for control of CPSs. In particular, the review gives a special focus on event-based controllers, which are based on state-space and Proportional-Integral-Derivative (PID) controllers. Several control challenges are also given at the end of this paper. It is envisaged that this paper can provide valuable knowledge for research in control of CPS.

## 2. CO-DESIGN CONTROL APPROACHES

The main objective in control of CPS is to bring the system to the desired physical performance such as stability, timing performance and system optimisation, while optimising computational design based on RTS issues such as resource utilisation (CPU and bandwidth), concurrency and scheduling problems, and quality of service (QoS). A conventional method which uses individual design and analysis tools with respect to their own domains has limited interaction. In contrast, a co-design approach involves sharing, comparing, and gathering the knowledge and perspectives brought by the stakeholders involved in the design process [11]. As CPSs are designed from the association of control systems, communication (networking) and computing, a co-design approach of CPS requires more integrated approaches with a feature to include more compartments of interaction between physical and cyber systems. Figure 2 shows the concept of co-design in designing and analysing the CPS.

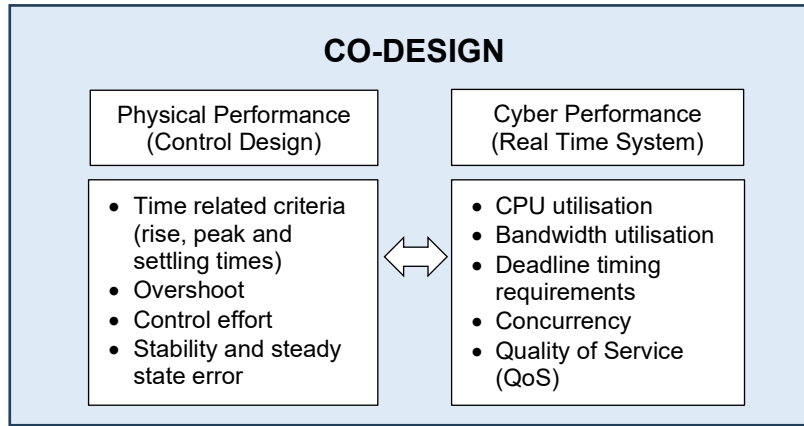


Figure 2. Co-design components in control of CPS.

In co-design, both physical and cyber issues will be considered fairly, where the controller for physical systems will consider cyber performances and vice versa. This method can give a mutual benefit to one another in order to optimise their limited resources. Co-design technique in control system design can be categorised into co-design for multitask control and single task control. Another co-design control approach which is gaining popularity is an event-based controller. These controllers will be reviewed in the following sections.

## 2.1 Multi-Task Control

In embedded systems, control and scheduling co-design becomes an issue when several controller tasks share the same embedded systems (controller) and disrupt the ideal sampling and actuation patterns. To allow this single controller multitasking, the controller task must be scheduled accordingly. One of the early works on control and scheduling co-design was published in [12], where they proposed an offline optimisation sampling frequency using a controller performance index to formulate a minimum cost function according to schedulable constraint. Then, Eker *et al.* [13] proposed a feedback scheduler which was utilised from a linear quadratic (LQ) state feedback controller subject to the sampling interval for online sampling rate adjustment.

This work was extended in [14] by using a feedforward general linear dynamic controllers based on CPU utilisation. This early work which namely feedback and feedforward scheduling apparently managed to assign a new sampling frequency by solving cyber utilisation problem which significantly affect the physical performance. However, an extra task is required in order to regulate the sampling frequency. The feedback and feedforward scheduling are able to reduce the uncertainties in control task period, jitter, and execution time, thus make things easier for the control system design, since it relieves the requirement for robust control regarding the delay and jitter.

In the advanced work, Xu *et al.* [15] have leveraged the optimal control theory of linear quadratic Gaussian (LQG) as a co-design technique by measuring the cost function of the total control loop as a control performance metric to LQG design to achieve an optimal task scheduling. The review and evaluation for four proposed LQG-based co-design methods namely delay-aware [16], stochastic [17], periodic [18], and harmonic can be found in [19]. The LQG co-designs have shown that by distributing an optimal scheduling task in real time scheduling, better control performance can be achieved but with the cost of highly intensive computation.

## 2.2 Single-Task Control

Anytime control is one of the co-designs that fall under this category [20]. It is a type of control that can modify their structure based on the performance of the RTS. There are two types of modification that can be applied: model reduction [21] and performance reduction [22]. Model-predictive control (MPC) is an example for anytime control application, where if the RTS is overloaded, the MPC's model can be reduced by skipping any optional control algorithm or by decreasing the performance optimisation with less objective function. In the recent work, Pant *et al.* [23] proposed a co-design of the anytime estimation and MPC algorithms with the manner of delays and inaccuracies issues. Their anytime estimation has the flexibility to reduce the estimation time when the computational resources become overloaded, hence achieved the control objective at a reduced energy cost. The anytime control has shown its capability to reduce the computational cost for the MPC. However, the MPC itself is computationally expensive as the MPC algorithms repetitively calculate the optimisation problem for every step using a prediction over specific time horizon to compute a control input.

In the diverse co-design work, Bradley and Atkins [24] proposed co-design for CPS concept in mathematical model using ordinary differential equations (ODEs). The idea was to co-regulate cyber and physical systems through LQR. The concept is similar to feedback scheduling and time-varying sampling and control to provide a sampling time according to the plant state condition. Full state feedback gain was calculated using an LQR, hence the controller managed to optimise input for both cyber and physical states. In order to create a simulation for time-varying sampling rate, a new simulation technique has been designed by cascading the solver for ODE. Their approach would be more convincing if the model has appropriate coupling term on the interaction between cyber and physical errors rather than just co-regulate in the LQR gain.

### 2.3 Control Strategies

In the literature, most of the co-design techniques were implemented in context of “cyber”, where the designed is considering the task scheduling in RTS particularly for the multitask control. The scheduling is adjusted to keep the physical performance. On the other hand, the co-design in a single control task gives more engagements for the physical part in the co-design where the controller is adjusted to relax the computation.

A brief summary of the main advantages and disadvantages of the existing co-designs which related to CPS is given in Table 1. It can be shown that the co-design technique is able to give an enhancement of the integration between cyber and physical elements and there is an opportunity to explore the event-based controller to be a CPS feedback controller. To achieve this, a framework of the co-design using the event-based controller as a CPS feedback controller is required, where this framework can distinguish the CPS design over the other systems such as digital control system, network control system, etc.

Table 1. Comparisons of existing co-design strategies.

Approaches	Method	Advantage	Disadvantage
Cyber approach - Real time system (Static Scheduling)	Optimal sampling and scheduling	The controller is able to control multi-task/systems by optimising the task frequency and scheduling for limited computational resources.	The control input is updated by following the scheduling task without taking into account the dynamic of the physical performance.
Cyber approach - Real time system (Dynamic Scheduling)	Feedback-feedforward scheduling	Dynamic sampling and scheduling are given based on computational utilisation and controller performance.	The controller needs to perform and additional task that results in a higher computational requirement.
Physical approach - Control system design	Anytime control	Give the flexibility to the receding horizon control to reduce their model or optimization constraint when the CPU utilisation becomes overload.	The controller still requires high computational load.
	Coupled cyber-physical co-regulation	A sampling frequency is introduced as cyber state to be used in co-regulating with the physical state.	The controller only reduces the sampling frequency but still updating the control input when the system response has reach steady-state condition.
	Event-based controller	There is a flexibility in the design process to make a trade-off between the cyber and physical performances. The control updates are reduced when the physical state has reached the steady-state condition.	Requires an extra algorithm to modify the time-triggered controller and the output response suffers from sticking and limit cycles issues.

### 3. EVENT-BASED CONTROLLERS

In order to allow flexibility to the system to make trade-off between the computation and control performance, an event-based controller has been proposed by [25]. The main objective of the event-based controller was to provide a control input based on the event, instead of depending on a specific sampling time. The event was determined by an event-detector by applying a logic rule namely event-triggering mechanism/condition. Event-based controllers immediately change the real-time system's task period in reaction to the performance of the application.

It is worth mentioning that as an attempt for an event-based PID controller to reduce the control computation, it exploits the timing of control sequence, and the sampling has become asynchronous. This form of sampling has resulted in two significant deteriorations to the output signal: (1) sticking, and (2) limit cycles [26, 27]. Sticking is the phenomenon where the event stop triggering even the output response still not reach the steady-state. This problem mostly happens due to the absolute relative error and the triggering mechanism only measure the error. On the other hand, limit cycle is the case with an oscillatory response near the steady-state due to the output response unable to stay inside the steady-state boundary after the control signal stop updating [26, 27].

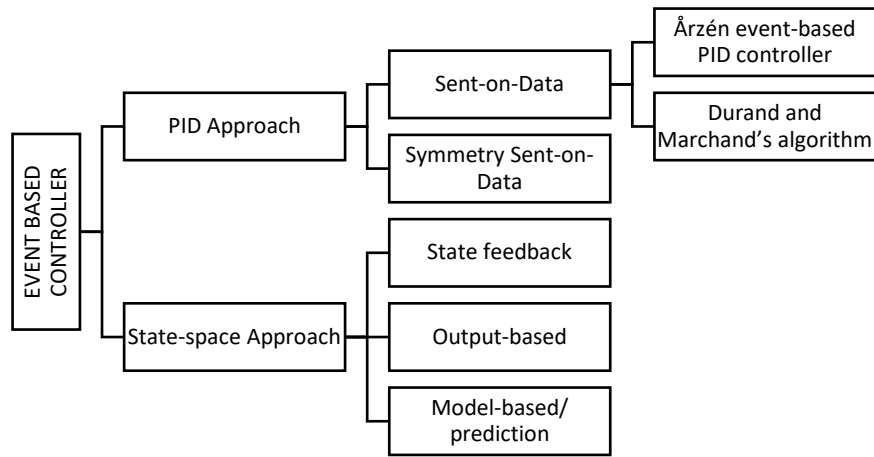


Figure 3. Approaches in event-based controllers.

Recent demand on the co-design for CPS applications has triggered the attention of many researchers and is becoming an urgent demand to develop, improve and test new control strategies for CPS. Among the approaches, event-based feedback control appears to be more inclined to the physical approach, more specifically, control system approach. The event-based feedback control is not a new idea, and it has been popularised by [28] that analysed and compared between the time-triggered sampling and event-triggered sampling, and [25] that proposed a simple event-based PID controller.

Several control strategies have also been proposed to meet the event-based controller requirements. They can be categorised into two main configurations: (1) event-based state-space control, and (2) event-based PID control, as shown in Figure 3. Control algorithms that fall under these two approaches are also listed in Figure 3. The following section provides a review of the existing event-based state-space and PID approaches.

### 3.1 Event-based State-Space Controller

A state-space model is a mathematical representation of a physical system as a set of input, output, and state variables related by differential equations. The state variables define the values of the output variables. The event-detector for event-based state-space can be in a continuous or periodic form. For a continuous event-detector, a lower bound execution time is applied to avoid Zeno behaviour, where the event happens excessively fast when the system state approaches a steady state [29, 30]. The continuously checking of event-triggered condition in continuous state-feedback event-based controller can cause the event-triggering mechanism to create some overhead. As for that reason, it is suggested to use a discrete or periodic sample to verify the event-triggering condition [31, 32, 33].

It is important to note that in the state-space approach, the event-based control is mostly called by event-triggered control (ETC). An early continuous event-triggered state-feedback was used as a co-scheduling in RTS as reported in [34]. Their ETC was not for reducing the control computation, rather to co-scheduling the control task for multitask control with the objective to ensure the control tasks meet their deadline while maintaining plant stabilisation. The first attempt to develop a theory of state-feedback ETC to reduce the computation was made by [35]. They proposed to invoke the control updates whenever the state exceed a certain threshold that able to destabilise the system. In [36], the error between measured continuous state-feedback and estimated state-feedback was used to determine the control updates. This was only generated if the estimation error exceed a given sensitivity bound, in order to reduce the control updates due to the effect of disturbance. The state-feedback ETC has also been developed under the effect of delay as reported in [37].

In a state-feedback controller, all the states are accessible to be used in control calculation. In contrast, an output-based controller only uses the output to estimate. The first paper that formally analyse the stability of output-based controller in ETC was published by [38], where the controller was used for the decentralised event-detector. In another work, Zhang and Feng [39] proposed exponentially decreasing event threshold condition for both continuous and discrete event-detectors for the observer-based ETC. Dolk *et al.* [40] have also developed the output-based controller with dynamic ETC, which leads to trade-off between robustness, performance and network utilisation.

In another event-based state space approach, Heemels and Donkers [33] put an effort to implement periodic event-triggered control (PETC) for a model-based controller. As the model-based controller has three functions which are estimate/predict, correct, and control, they proposed to use an observer-based controller and an event-triggering mechanism to reduce the communication between the sensor-to-controller and controller-to-actuator channels. The states estimated by the observer were used by the event-triggering mechanism to decide either the estimated states can be delivered or not to the controller for control computation. By doing so, the control computation implemented only when the error between estimated states and current states exceeds a predefined threshold. In another work, Garcia and Antsaklis [41] proposed a model-based ETC with the effect of common issues of NCS subject to quantisation and time-varying delays. The plant model was used to generalise the zero-order-hold (ZOH) and provides a stability threshold for ETC to reduce the feedback rate from the sensor to controller. In a recent work, the model-based ETC has been extended for a nonlinear system [42] and a hybrid system [43].

In a state-feedback event-based controller, the triggering mechanism must calculate all states conditions, which imply that all states must be measured by the sensor. Therefore, the cost for sensor installation increases. Meanwhile, output-based and

model-based controllers require an estimator to estimate the states, hence extra computation is required by the system which subsequently increased the computational burden.

As a lot of computation is required in the state-space controller, which might contradict with the objective of event-based control to reduce the computation time. Even though the event-based strategy in state-space controller is able to reduce the communication and control updates, it is more practical for CPS to use a controller with less computation. Moreover, most of the event-based state-space approaches focus on relaxing the control updates subject to system stability. This method guarantees to maintain the stability of the system. However, the control performance may generate a sticking and limit cycles output response.

### 3.2 Event-based PID Controller

Despite the existence of more advanced control strategies, it is becoming extremely difficult to ignore the feasibility of PID controller in solving the control problems. In general, a PID controller is a single-input single-output (SISO) controller with various available analyses and design techniques. The PID controller is a well-known feedback control strategy among control engineers due to its favourable features, such as a simple control algorithm and easy implementation in industrial applications. An event-based PID controller introduced by [25] was the first event-based controller debut in control system. This article has shown the benefits of event-based controller in reducing the number of control-computation updates as the control updates were based on the event instead of predetermined timing. In the work, the event was determined by the event detector that uses a logical condition as an event triggering mechanism. The controller only computes and update the control signal when the relative error between the measured signal and reference cross a certain level *elim*, or if the control updates interval reaches a maximal sampling period. The relative error level crossing is also known as send-on-delta (SOD) signal sampling [44] or deadband sampling [45].

Durand and Marchand [46] revised the work of [25] by improving the PI discretisation algorithm from forward difference to backward difference approximations. The triggering mechanism has also been simplified by removing the safety limit condition. However, this raised the overshoot problem when the new set point was given after a long steady-state. As the maximal sampling period was removed, the interval time without control updates can grow without limit which in turn will result the integral part to calculate a large value of control signal and contributing to overshoot response. They suggested three improved algorithms to overcome the overshoot drawback namely, saturation, forgetting factor and hybrid algorithms. As a result, more control computation was able to be reduced with the cost of extra computation due to the exponential function. They have also extended their proposed event-based algorithms on a real time application of a mini quadrotor helicopter [47]. The experimental results showed the effectiveness of the proposed event-based PID algorithms with a reduction of the computing and communication resources utilisation. However, their work would have been more relevant to CPS co-design if the algorithms complexity and evaluation over network environment is included.

The logic rule for event condition/triggering mechanism is an important criterion in event-based PID controller. The research that focused on this rule was reported in [48], where five error-based triggering mechanisms as listed in Table 2 have been compared. They have suggested that the simplest triggering mechanism (absolute relative) was the best option in balancing the trade-off between the performance of the control and the reduction of the control computation.

Table 2. Different event triggering mechanisms [48].

Triggering mechanisms	Logic condition	The control is updated when:
Relative error	$ e(t_k) - e(t_j)  \geq elim$	The different between current error and last time error is greater than <i>elim</i> .
Integral relative error	$\int_k^{k+1}  e(t_k) - e(t_j)  dt \geq elim$	The integral absolute error (IAE) value for relative error is greater than <i>elim</i> .
Predictive error	$ e_p(t) - e(t)  \geq elim$	The different between current error and predictive error is greater than <i>elim</i> .
Integral predictive error	$\int_k^{k+1}  e_p(t) - e(t)  dt \geq elim$	The IAE value for predictive different is greater than <i>elim</i> .
Integral square relative error	$\int_k^{k+1} [e(t_k) - e(t_j)]^2 dt \geq elim$	The square IAE value for relative error is greater than <i>elim</i> .

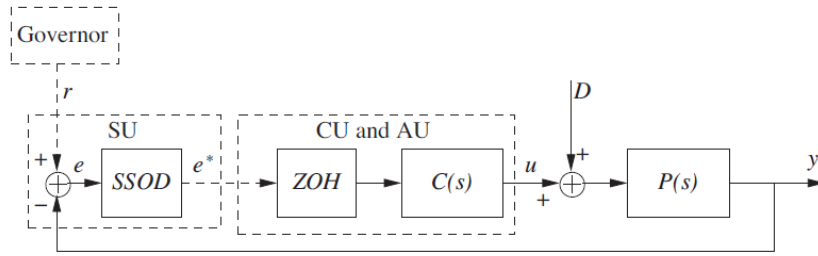


Figure 4. Block diagram for SSOD-PI control scheme. SU, CU and AU are sensor, controller and actuator units respectively.

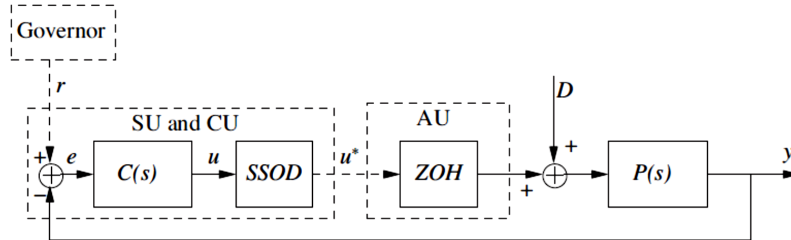


Figure 5. Block diagram for PI-SSOD control scheme.

Typically, the earlier research on event-based PID controller based on the level-crossing or SOD sampling was unsuccessful in avoiding the sticking and limit cycles issues. In parallel line of research, an effort to remedy the sticking and limit cycles issues has been conducted. In another work, the SOD sampling has been improvised with the symmetry send-on-delta (SSOD) as found in [49]. They proposed to generalise the measurement error,  $e(t)$  to become symmetry error,  $e^*(t)$  with the scale of  $elim$ , then the control signal was calculated using the  $e^*(t)$ . Their work was based on the PI controller and two types of SSOD were introduced namely PI-SSOD and SSOD-PI. SSOD-PI generalized the error before control computation, while PI-SSOD calculated the control signal first before generalized the control signal  $u(t)$ . Figures 4 and 5 show the block diagram for SSOD-PI and PI-SSOD respectively.  $r$  and  $y$  are the system's input and output respectively, and  $P(s)$  is the plant. They also suggested two sufficient conditions for the proposed event-based SSOD controller in order to avoid the limit cycles occurrences. Simulation and experimental results have shown that the response manages to avoid the limit cycles occurrences if the values of PI gains are inside the control parameters determined by the suggested sufficient condition.

As the PI gain parameters for event-based SSOD should follow a certain range described by the sufficient condition for limit cycles avoidance, the available time-triggered continuous tuning rules for PI gains are no longer be explicitly adopted on event-based SSOD. Therefore, the direction of event-based SSOD has been focused on formulating a new tuning rule. Initiate by the event-based SSOD inventor itself the ST (settling time) tuning rule for SSOD-PI has been developed [50, 51]. Based on the step response for various normalised first order system, the PI gain parameters were determined based on the objective to search for a minimum settling time. The ST tuning rule managed to produce a response without limit cycles, but the response was considered slow as compared to the established tuning rules such as AMIGO (Approximate M-constrained Integral Gain Optimization) [52] and SIMC (Simple Internal Model Controller) [53].

In another study, Romero and Sanchís [54] have developed a tuning rule for SSOD-PI based on the phase margin to oscillations constraint, which was determined in Nyquist plot. The SSOD response was plotted in Nyquist plot using a describing function (DF) [55], while the phase different between the SSOD and open-loop process plots was determined as a new phase margin in order to maximising the PI gain parameters. The expanded version on SSOD-PID was presented in [56].

Subsequently, the interest on event-based SSOD tuning has been increased covering various tuning techniques. For the robust tuning, Ruiz *et al.* [57] have used a characteristic stability map, which was a combination of SSOD and robust stability Nyquist plots. Miguel *et al.* [58] included the optimisation in tuning the SSOD-PID by utilising the Tsytkin margin ( $M_T$ ) to minimising the IAE. Sánchez *et al.* [48] used an identification method to tune the PI gain parameters by introducing an oscillation margin. The same authors have also improved the SSOD Nyquist plot to validate the event-based SSOD tuning rule [59].

Even though, many tuning rules have been proposed for event-based SSOD to avoid a limit cycles, the study to directly utilise an available tuning rule from time-triggered in event-based PID controller has not been published. It can be a useful if the event-based controller can use the large number of well-established tuning rules applied in industry in order to achieve difference performances requirement or robustness aims [25, 49, 60, 61]. In general, the event-based controllers used in SOD and SSOD in the literature were described as a PID controller. However, in their design and analysis, only PI controller has been applied. This is because in the process industry, a first order plus dead times (FOPDT) is sufficient to represent the process and the PI controller has a good performance to control the process [62]. The PI controller also has an advantage in CPS design because the algorithm is simple and the gain parameters is easy to tune, resulting in low computational requirement as compared to the PID controller. Moreover, most of the published articles on event-based PID do not evaluate their proposed strategies over the network environment. The analysis of the even-based controller on the network is an important criterion to evaluate its viability as a CPS co-design as the CPSs are mostly run in the network environment.

Table 3. Comparisons of existing event-based controller strategies.

Approaches	Advantages	Disadvantages
Event-based state-space approach	Preserves commonly used stability concepts and analysis.	State-dependent threshold requires more computation for event detector, and the response most probably will generate a limit cycles condition.
Event-based PID controller (SOD)	The algorithm is simple which allows to be implemented in real application.	A sticking problem and limit cycles response.
Event-based PID controller (SSOD)	Able to generate response without a limit cycle.	Extra computation is compulsory to generalise the error. The strategies restrict the PI gain value, thus a new tuning rule for PI gain is required.

One of the event-based SSOD implementations in network environment was reported in [63]. The technique was designed together with a wireless sensor network for a greenhouse. The AMIGO tuning rules were used as a tuning rule for their PI-SSOD and SSOD-PI techniques. As expected, the event-based SSOD strategies were able to reduce the number of communications while maintaining control performance. However, the result showed a significant limit cycles occurrence due to the tuning values for PI gains which were out of the range as suggested by [50]. Pawlowski *et al.* [63] also showed that the event-based PID controller was able to be implemented in network environment and as it can reduce computation iteration. Therefore, the potential of the event-based PID controller as a feedback controller for CPS needs to be explored. Moreover, a proper design framework of event-based PID controller consists of controller design, trade-off and evaluation must be determined. A summary of the main advantages and disadvantages of the existing event-based controller is listed in Table 3.

#### 4. TOOLS FOR EVENT-BASED CONTROLLERS

For an effective control, a co-design CPS framework should consist of a specific tool and method for controller design and evaluation. For simulations of control system, the Matlab/Simulink software is widely used for controller design and performance analyses. However, as the CPS consists of computation, communication and control, the requirement for network control system and RTS features must be included. In an important work, Cervin *et al.* [64] has developed a MATLAB/Simulink-based simulation toolbox for embedded and networked control systems called as TrueTime. The toolbox is very useful, and it has been increasingly used by researchers as a CPS evaluation tool [15, 65, 66, 67, 68, 69].

As the CPS is a heterogeneous design, it is difficult to obtain an explicit reliability function of the CPS that contain a mapping of the performance's trade-off between computation, communication and control [67]. In this case, the Monte-Carlo simulation (MCS) for the CPS evaluation can be considered [67, 70, 71, 72]. Moreover, several researchers have combined the TrueTime and MCS simulations as reported in [15, 69, 73]. Similarly, it will be interesting to explore the combination of the TrueTime and MCS simulations for evaluating of the event-based controller.

An event-based strategy requires extra algorithm to modify the time-triggered approach in order to reduce the control updates. For the event-based PID approach, Durand *et al.* [47] used an expensive algorithm which was exponential function in their event-based PI algorithm, while SSOD event-based PID methods requires additional computation to generalise the measurement error [49, 50, 51]. On the other hand, a higher computation algorithm is required for the event-based state-space as compared to the event-based PID as the control computation needs to calculate the control signal for all states [35]. Moreover, an extra algorithm is required for output-based and model-based controllers to estimate the states [33]. Since the goal of an event-based strategy is to reduce the computation, control algorithms with less computational time should be considered. The algorithm complexity prevalently calculated using the big-O notation [74]. However, the big-O notification is in term of the order of the algorithm complexity, not the actual execution time.

The complexity of the algorithm has also been determined using a running time for algorithm execution [75]. The number of arithmetic operations performed by the algorithm is commonly used to estimate time complexity, assuming that each arithmetic operation takes a fixed amount of time to perform [76]. According to [77], the energy cost of basic arithmetic in a 45 nm CMOS process for 16-bit floating addition and multiplication is equivalent to 0.4 pJ and 1.1 pJ, respectively. Therefore, as an alternative, computational energy consumption and algorithm complexity can be calculated based on the number of arithmetic operations in the algorithm, which can be used to evaluate the performance of the event-based strategy.

#### 5. CONTROL CHALLENGES

Based on the review and the previous discussions, the following challenges in control of CPS can be considered:

- (a) The cyber part plays a role in computing a control signal for the actuator using an information from the sensor in feedback closed-loop control. The feedback loops that constantly transfer and compute the control signal at every predefined sampling time will keep utilising the cyber energy. Therefore, to enhance the integration of cyber and physical parts, this computation cost should be included as an objective in controller design by allowing the trade-off between cyber and physical performances.

- (b) A co-design technique such as event-based controller can play an important role in control of CPS [78]. However, a framework or guideline procedures to utilise the event-based controller as a CPS feedback controller is not completely available.
- (c) A review has shown that the event-based PID can be a good candidate for an effective co-design technique in order to incorporate CPS in controlling a plant. However, due to the sticking and limit cycles issues, a well-established tuning rule for PID controller is not suitable for the event-based PID. Moreover, most of the efforts in improving the event-based PID will introduce new computation which increase the computational cost. Also, there is a need to improve the event-based PID with a simple algorithm which is able to address the sticking and limit cycles problems under the presence of network issue.
- (d) CPS mostly runs on the network environment where the timing imperfections in computing and communication components affect the system performance and reliability. In RTS, control computation is one of the scheduling tasks, and thus it will inherit several issues such as delay jitter, task execution time scheduling and task preemption. The variation of input-output delay over the period can deteriorate the control performance and possibly de-stabilise the system. Hence, to enable event-based PID as a CPS feedback controller, it should be evaluated with the presence of these timing constraints.
- (e) A new design methodology to co-design the controller in order to fulfil both cyber and physical objectives is required. This co-design framework should have a structure in designing a controller that is able to reduce the computation usage while maintaining physical performance, and the evaluation technique that can facilitate the method to consider timing constraints in CPS.

## 6. CONCLUSION

This paper has provided a review of co-design approaches for control of CPS. The approach is well suited for the CPS design to enhance the integration by enabling the trade-off between control and computational performances. Several controllers have been discussed, and it seems that the event-based controller has a direct link with a feedback control design. A review of event-based controller strategies has considered state-space and PID approaches. It was found that the event-based PID controller is more practical to be used as a CPS co-design with less computational time. However, such a strategy needs to be improved to cater the issues of sticking and limit cycles response. Besides, this paper has also described tools used in developing and assessing the co-design approaches for CPS. Several control challenges for CPS control have also been identified.

## ACKNOWLEDGEMENT AND FUNDING

The authors gratefully acknowledged the Universiti Teknologi Malaysia, Malaysia for the financial support through UTM Encouragement Grant (UTMER) Vote No. 20J46.

## DECLARATION OF CONFLICTING INTERESTS

The authors declare no potential conflicts of interest with respect to the research and publication of this article.

## REFERENCES

- [1] E. A. Lee and S. A. Seisha, *Introduction to Embedded Systems - A Cyber-Physical Systems Approach*, MIT Press, Cambridge, USA, 2017.
- [2] NIST, *National Institute of Standards and Technology*, USA, 2013.
- [3] E. A. Lee, The past, present and future of cyber-physical systems, *Sensors*, 15(3), 2015, 4837-4869.
- [4] NSF, *Cyber Physical System*, 2021, <https://www.nsf.gov/pubs/2021/nsf21551/nsf21551.htm>.
- [5] R. Raj, I. Lee, L. R. Sha and J. A. Stankovic, Cyber-physical systems: The next computing revolution, *Design Automation Conference*, Anaheim, USA 2010, 791-736.
- [6] K. -D. Kim and P. R. Kumar, Cyber-physical systems: A perspective at the centennial, *Proceedings of the IEEE*, 100(1), 2012, 1287-1308.
- [7] J. Sztipanovits et al., Toward a science of cyber-physical system integration, *Proceedings of the IEEE*, 100(1), 2012, 29-44.
- [8] A. Parnianifard et al., Robust optimal design of FOPID controller for five bar linkage robot in a cyber-physical system: A new simulation-optimization approach, *PLoS ONE*, 15(11), 2020, e0242613.
- [9] S. A. Seshia, S. Hu, W. Li and Q. Zhu, Design automation of cyber-physical systems: challenges, advances, and opportunities. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(9), 2017, 1421-1434.
- [10] J. Bradley and E. Atkins, Optimization and control of cyber-physical vehicle systems, *Sensors*, 15(9), 2015, 23020-23049.
- [11] C. Aubrun, D. Simon and Y. Q. Song, *Co-Design Approaches for Dependable Networked Control Systems*, John Wiley and Sons, Hoboken, USA, 2013.
- [12] D. Seto, J. Lehoczky, L. Sha and K. G. Shin, On task schedulability in real-time control systems, *IEEE Real-time Technology and Applications Symposium (RTAS)*, Washington, USA, 1996, 13-21.
- [13] J. Eker, P. Hagander and K. -E. Årzén, A feedback scheduler for real-time control tasks, *Control Engineering Practice*, 8(12), 2000, 1369-1378.
- [14] A. Cervin, J. Eker, B. Bernhardsson and K. Erik, Feedback-feedforward scheduling of control tasks, *Real-Time Systems*, 23(1), 2002, 25-53.

- [15] Y. Xu, K. -E Årzén, E. Bini and A. Cervin, LQG-based control and scheduling co-design, *IFAC PapersOnLine*, 50(1), 2017, 5895-5900.
- [16] E. Bini and A. Cervin, Delay-aware period assignment in control systems, *Proceedings of the 29th IEEE Real-Time Systems Symposium*, Barcelona, Spain, 2008.
- [17] Y. Xu, K. -E. Årzén, E. Bini and A. Cervin, Response time driven design of control systems, *IFAC Proceedings*, 47(3), 2014, 6098-6104.
- [18] Y. Xu et al., Exploiting job response-time information in the co-design of real-time control systems, *IEEE 21st International Conference on Embedded and Real-Time Computing Systems and Applications*, Hong Kong, China, 2015, 247-256.
- [19] Y. Xu, A. Cervin and K. E. Arzen, Harmonic scheduling and control co-design, *IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, Daegu, South Korea, 2016, 182-187.
- [20] K. -E. Årzén, A. Cervin, J. Eker and L. Sha, An introduction to control and scheduling codesign, *IEEE Conference on Decision and Control*, Sydney, Australia, 2000, 4865-4870.
- [21] R. Bhattacharya and G. Balas, Anytime control algorithm: Model reduction approach, *Journal of Guidance, Control, and Dynamics*, 27(5), 2004, 767-776.
- [22] D. Fontanelli, L. Greco and A. Bicchi, Anytime control algorithms for embedded real-time systems, *Lecture Notes in Computer Science*, 2008, 158-171.
- [23] Y. V. Pant, et al., Anytime computation and control for autonomous systems, *IEEE Transactions on Control Systems Technology*, 29(2), 2020, 768-779.
- [24] J. M. Bradley and E. M. Atkins, Coupled cyber-physical system modeling and coregulation of a cubesat, *IEEE Transaction on Robotics*, 31(2), 2015, 443-456.
- [25] K. -E. Årzén, A simple event-based PID controller, *IFAC Proceedings*, 32(2), 1999, 8867-8692.
- [26] A. Cervin and K. J. Åström, On limit cycles in event-based control systems, *46th IEEE Conference on Decision and Control*, New Orleans, USA, 2007.
- [27] Á. Ruiz, J. E. Jiménez, J. Sánchez and S. Dormido, A practical tuning methodology for event-based PI control, *Journal of Process Control*, 24(1), 2014, 278-295.
- [28] K. J. Åström and B. Bernhardsson, Comparison of Riemann and Lebesgue sampling for first order stochastic systems, *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, USA, 2002, 2011-2016.
- [29] Y. Or and A. D. Ames, Stability and completion of Zeno equilibria in Lagrangian hybrid systems, *IEEE Transactions on Automatic Control*, 56(6), 2011, 1322-1336.
- [30] M. Lemmon, Event-triggered feedback in control, estimation, and optimization. *Lecture Notes in Control and Information Sciences: Networked Control Systems*, 2010, 293-358.
- [31] A. Eqtami, D. V. Dimarogonas and K. J. Kyriakopoulos, Event-triggered control for discrete-time systems, *Proceedings of the American Control Conference*, Baltimore, USA, 2010, 4719-4724.
- [32] W. P. M. H. Heemels, M. C. F. Donkers and A. R. Teel, Periodic event-triggered control for linear systems, *IEEE Transactions on Automatic Control*, 58, 2013, 847-861.
- [33] W. Heemels and M. Donkers, Model-based periodic event-triggered control for linear systems, *Automatica*, 49, 2013, 698-711.
- [34] P. Tabuada, Event-triggered real-time scheduling of stabilizing control tasks, *IEEE Transactions on Automatic Control*, 52(9), 2007, 1680-1685.
- [35] W. Heemels, J. H. Sandee, and P. P. J. V. D. Bosch, Analysis of event-driven controllers for linear system, *International Journal of Control*, 84(4), 2008, 571-590.
- [36] J. Lunze and D. Lehmann, A state-feedback approach to event-based control, *Automatica*, 46(1), 2010, 211-215.
- [37] D. Yue, E. Tian and Q. L. Han, A delay system method for designing event-triggered controllers of networked control systems, *IEEE Transactions on Automatic Control*, 58(2), 2013, 475-481.
- [38] M. C. F. Donkers and W. P. M. H. Heemels, Output-based event-triggered control with guaranteed gain and improved and decentralized event-triggering, *IEEE Transactions on Automatic Control*, 57(6), 2012, 1362-1376.
- [39] J. Zhang and G. Feng, Event-driven observer-based output feedback control for linear system, *Automatica*, 50(7), 2014, 1852-1859.
- [40] V. S. Dolk, D. P. Borger and W. P. M. H. Heemels, Output-based and decentralized dynamic event-triggered control with guaranteed LP-gain performance and zeno-freeness, *IEEE Transaction on Automatic Control*, 62(1), 2017, 34-39.
- [41] E. Garcia and P. J. Antsaklis, Model-based event-triggered control for systems with quantization and time-varying network delays, *IEEE Transactions on Automatic Control*, 58(2), 2013, 422-434.
- [42] R. Dou and Q. Ling, Model-based periodic event-triggered control strategy to stabilize a scalar nonlinear system, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(9), 2021, 5322-5335.
- [43] K. Liu, A. Teel, X. Sun and X. Wang, Model-based dynamic event-triggered control for systems with uncertainty: a hybrid system approach, *IEEE Transactions on Automatic Control*, 66(1), 2021, 444-451.
- [44] M. Miskowicz, Send-on-delta concept: An event-based data reporting strategy, *Sensors*, 6, 2016, 49-63.
- [45] V. Vasyutynskyy and K. Kabitzsch, Towards comparison of deadband sampling, *IEEE International Symposium on Industrial Electronics*, Vigo, Spain, 2007, 2899-2904.
- [46] S. Durand and N. Marchand, Further results on event-based PID controller, *European Control Conference (ECC)*, Budapest, Hungary, 2009, 1979-1984.
- [47] S. Durand, B. Boisseau, N. Marchand and J. F. Guerrero-Castellanos, Event-based PID control: Application to a mini quadrotor helicopter, *Control Engineering and Applied Informatics*, 20(1), 2018, 36-47.

- [48] J. Sánchez, M. Á. Guarnes and S. Dormido, On the application of different event-based sampling strategies to the control of a simple industrial process, *Sensors*, 9, 2009, 6795-6818.
- [49] M. Beschi, S. Dormido, J. Sanchez and A. Visioli, Characterization of symmetric send-on-delta PI controllers, *Journal of Process Control*, 22, 2012, 1930-1945.
- [50] M. Beschi, S. Dormido, J. Sanchez and A. Visioli, Tuning rules for event-based SSOD-PI controllers, *Proceedings of the 20th Mediterranean Conference on Control and Automation*, Barcelona, Spain, 2012.
- [51] M. Beschi, S. Dormido, J. Sanchez and A. Visioli, Tuning of symmetric send-on-delta proportional-integral controllers, *IET Control Theory & Applications*, 8, 2014, 248-259.
- [52] K. J. Åström and T. Hägglund, Revisiting the Ziegler–Nichols step response method for PID control, *Journal of Process Control*, 14(6), 2004, 635-650.
- [53] S. Skogestad, Simple analytic rules for model reduction and PID controller tuning, *Journal of Process Control*, 13(4), 2003, 291-303.
- [54] J. Romero and R. Sanchís, A new method for tuning PI controllers with symmetric send-on-delta sampling strategy, *ISA Transactions*, 64, 2016, 161-173.
- [55] A. Gelb and W. Vander Velde, *Multiple-Input Describing Functions and Nonlinear System Design*, McGraw-Hill, New York, USA, 1968.
- [56] J. Romero and R. Sanchís, Tuning and robustness analysis of event-based PID controllers under different event generation strategies, *International Journal of Control*, 91, 2018, 1567-1587.
- [57] A. Ruiz, et al., A unified event-based control approach for FOPTD and IPTD processes based on the filtered smith predictor, *Journal of the Franklin Institute*, 354(2), 2017, 1239-1264.
- [58] O. Miguel-Escrig, J. -A. Romero-Perez, and R. Sanchis-Llopis, Tuning PID controllers with symmetric send-on-delta sampling strategy, *Journal of the Franklin Institute*, 357(2), 2019, 832-862.
- [59] J. Sánchez, M. Guinaldo, S. Dormido and A. Visioli, Validity of continuous tuning rules in event-based PI controllers using symmetric send-on-delta sampling: An experimental approach, *Computers and Chemical Engineering*, 139, 2020, 106878.
- [60] A. O'Dwyer, *Handbook of PI and PID Controller Tuning Rules*, 3rd Edition, Imperial College Press, London, UK, 2009.
- [61] T. Samad, A survey on industry impact and challenges thereof, *IEEE Control Systems Magazine*, 37(1), 2017, 17-18.
- [62] K. J. Åström and T. Hägglund, *Advanced PID Control*, The Instrumentation, Systems, and Automation Society, North Carolina, USA, 2006.
- [63] A. Pawlowski, et al., Application of SSOD-PI and PI-SSOD event-based controllers to greenhouse climatic control, *ISA Transactions*, 65, 2016, 525-536.
- [64] A. Cervin, J. Eker, B. Bernhardsson and K. Erik, Feedback-feedforward scheduling of control tasks, *Real-Time Systems*, 23(1), 2002, 25-53.
- [65] H. Gawand, A. K. Bhattacharjee and K. Roy, Real time jitters and cyber physical system, *Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics*, Delhi, India, 2014, 2004-2008.
- [66] S. Balasubramanian, B. Subathra, R. C. Hemesh, S. Gurusamy and S. Srinivasan, On simulating processor schedules and network protocols within CPS using TrueTime, *IEEE International Conference on Computational Intelligence and Computing Research (ICIC)*, Madurai, India, 2015, 1-6.
- [67] Z. Fang, H. Mo, Y. Wang and M. Xie, Performance and reliability improvement of cyber-physical systems subject to degraded communication networks through robust optimization, *Computers and Industrial Engineering*, 114, 2017, 166-174.
- [68] X. Chen, W. Zhang, T. Qian and W. Tang, Co-simulation of cyber physical microgrid considering economic constraints, *5th Asia Conference on Power and Electrical Engineering*, Chengdu, China, 2020.
- [69] A. Cervin and M. T. Andren, LQG-optimal versus simple event-based PID controllers, *Proceedings of the American Control Conference*, Denver, USA, 2020, 3678-3684.
- [70] W. Liu, et al., Reliability modeling and evaluation of active cyber physical distribution system, *IEEE Transactions on Power Systems*, 33(6), 2018, 7096-7108.
- [71] A. Y. Mussard, D. B. Rawat and M. Garuba, Data validation and correction for resiliency in mobile cyber-physical systems, *16th IEEE Annual Consumer Communications and Networking Conference*, Las Vegas, USA, 2019, 1-4.
- [72] K. Cao, et al., Edge intelligent joint optimization for lifetime and latency in large-scale cyber-physical systems, *IEEE Internet of Things Journal*, 9(22), 2022, 22267-22279.
- [73] A. Cervin, LQG-optimal PI and PID control as benchmarks for event-based control, *Second International Conference on Event-based Control, Communication, and Signal Processing*, Krakow, Poland, 2016, 1-8.
- [74] X. Chen, W. Zhang, T. Qian and W. Tang, Co-simulation of cyber physical microgrid considering economic constraints, *5th Asia Conference on Power and Electrical Engineering*, Chengdu, China, 2020.
- [75] N. M. H. Norsahperi and K. A. Danapalasingam, An improved optimal integral sliding mode control for uncertain robotic manipulators with reduced tracking error, chattering, and energy consumption, *Mechanical Systems and Signal Processing*, 142, 2020, 106747.
- [76] M. Sipser, *Introduction to the Theory of Computation*, Third ed., Course Technology, Boston, USA, 2013.
- [77] M. Horowitz, Computing's energy problem (and what we can do about it), *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, San Francisco, USA, 2014.
- [78] K. Liu, A. Teel, X. Sun and X. Wang, Model-based dynamic event-triggered control for systems with uncertainty: a hybrid system approach, *IEEE Transactions on Automatic Control*, 66(1), 2021, 444-451.