Dynamic Decentralized Periodic Event-Triggered Control for Wireless Cyber-Physical Systems

Anqi Fu, *Member, IEEE,* and Julie A. McCann *Member, IEEE*

***Abstract*—This paper adds dynamic variables to decentralized periodic event-triggered control, denoting dynamic decentralized periodic event-triggered control for wireless cyber-physical systems whose sensors are distributed and bandwidths are limited, and studies the stability and $\mathcal{L}_2$-gain. In dynamic decentralized periodic event-triggered control, dynamic variables that require only local information are designed and added to the event-triggered mechanisms. This dynamic variable can guarantee that the event-triggered mechanisms still be localized which does not require information from other nodes. For implementations with dynamic decentralized periodic event-triggered control, at each periodic sampling time the event-triggered mechanisms decide events based on new measurements from local sensors. Those measurements that trigger the events are transmitted to controllers, and after network transmission delays, they are received and the controller computes the control actions. The transmissions of these control actions from controllers to actuators are also decided by event-triggered mechanisms. The feasibility of the approach is shown via two examples. Numerical results show that, with the same stability and performance, dynamic decentralized periodic event-triggered control can save more than 40% events, making it more suitable for wireless cyber-physical systems.**

***Index Terms*—dynamic event-triggered mechanism; decentralized periodic event-triggered control; hybrid system; wireless cyber-physical system; smart water distribution systems.**

I. Introduction

Cyber-physical systems (CPSs) are an integration of physical processes and sensing, communication, control, and computation. In CPSs, the physical components and software components are deeply coupled, presenting more possibilities in accomplishing complex tasks, improving efficiency, and enlarging the physical scales of the systems. Applications of CPSs include smart water distribution systems, smart grids, precision farming etc. More often today, wireless networks are used to relay data in CPSs. As a trade-off, these wireless CPSs have both decreased communication bandwidth compared with wired systems and where they are battery powered, they require frugal use of communications for energy savings. Therefore, it is important to reduce the transmissions from each of the CPS tasks to better share the limited bandwidth and minimize hardware maintenance. In this paper we focus on how to reduce the transmissions from control tasks in CPSs.

This work was supported in part by Alan Turing Institute Data Centric Engineering (DCE) Programme sponsored by Lloyds Registry and in part by EPSRC Science for Sensor Systems Software (S4) Programme Grant number EP/N007565/1.

A. Fu is with the Faculty of Information Technology, Beijing University of Technology, Beijing 100124, P.R.China, he was with the Department of Computing, Imperial College London, London SW7 2AZ, UK. J. McCann is with the Department of Computing, Imperial College London, London SW7 2AZ, UK. Email: a.fu@bjut.edu.cn, j.mccann@imperial.ac.uk.

In CPSs, the control task aims to guarantee that the physical systems is stable while satisfying predesigned performance guarantees; and robustness and resilience of the systems. Typically feedback control loops are introduced for these control tasks. A feedback loop requires sensors to measure the system output, a controller to compute the control actions for the actuators based on the measurements, and communication channels to connect the sensors controller and actuators to close the loop. Normally, the feedback control loops are closed in a periodic manner, named periodic time-triggered control (TTC). However, TTC may result in bandwidth waste due to unnecessary data communications where nothing interesting is happening to the physical system under focus. Event-triggered control (ETC), on the other hand, triggers control actions only when necessary and has been proposed in e.g. [1]–[11].

Periodic event-triggered control (PETC, [12]–[14]) is an aperiodic triggering mechanism that introduces an event-triggered mechanism (ETM) to the TTC. At each periodic sampling time, the ETM determines whether the current measurement will be used to compute the control actions based on an event-triggered condition. However, this centralized periodic event-triggered control (CPETC) requires the whole vector of the current measurements for decision making. Therefore, unless the sensor nodes are co-located, CPETC cannot save any bandwidth compared with TTC. For large physical systems, whose sensors are usually not co-located, decentralized periodic event-triggered control (DPETC) is thus proposed in [12], [15]–[18]. In DPETC, the ETM is distributed to the sensor nodes who trigger events based on local information only. Previous work [19] has shown the efficiency of the DPETC in a smart water distribution system.

In this paper, we extend DPETC to further reduce the transmissions. Inspired by [20]–[23], we introduce dynamic variables to DPETC, and propose a dynamic decentralized periodic event-triggered control (DDPETC). In DDPETC, a dynamic variable is designed and added to each of the local ETM. This dynamic variable can recharge and discharge based on local system information according to the system's dynamics. Therefore, with this new dynamic variable, the ETM remains localized. In DDPETC, at each sampling time an event will be triggered locally when two conditions are satisfied: the local sampling error (the error between the current system output and sampled output) is large enough; and the local dynamic variable is non-positive. Compared with the static decentralized periodic event-triggered control (SDPETC) which only employs the first condition, DDPETC has a higher potential to reduce the number of events communicated.

Compared with [20] and [22], our proposed DDPETC considers periodic sampling instead of continuous sampling, and thus the sensors are not required to work continuously, which

further saves energy consumption; compared with [21] and [23], our work considers decentralized ETC instead, making our approach suitable for systems whose sensor nodes are not co-located. Further, compared with [12], in addition to adding the dynamic variables, our work also considers transmission delays, making our approach more practical as we demonstrate in the latter sections of this paper.

We apply our proposed DDPETC to two numerical examples: a batch reactor and a smart water distribution system. The water system is built on two physical system simulation toolboxes in Matlab: Simscape and Simscape fluids, which make the model highly representative to a real physical system. Compared with our prior work using the Waterbox in [19] which applies a switched controller, the current water system example in this paper removes the switches and applies a simple control gain as the controller, and the physical scale of the water system is also much larger.

This paper is organized as follows. Section II presents the necessary notations and introduces the system architecture. The main result is presented in Section III. We demonstrate the feasibility of the approach through two numerical examples in Section IV and conclude the work in Section V.

II. Notation and Problem definition

We denote the n-dimensional Euclidean space by $\mathbb{R}^n$. We use $\mathbb{R}^+$ and $\mathbb{R}_0^+ = \mathbb{R}^+ \cup \{0\}$ to denote the positive real numbers. We use $\mathbb{N}$ to denote the natural numbers including zero, while we use $\mathbb{N}^+$ to denote the set of natural numbers without including zero. The usual Euclidean (l_2) vector norm is denoted by $|\cdot|$. A symmetric matrix $M \in \mathbb{R}^{n\times n}$ is said to be positive definite, which is denoted by $M \succ 0$, if $x^{\mathrm{T}}Mx > 0$ for all $x \neq 0$, $x \in \mathbb{R}^n$. For a symmetric matrix $M \in \mathbb{R}^{n\times n}$, $\lambda_{\min}(M)$ and $\lambda_{\max}(M)$ denote the minimum and maximum eigenvalue of M, respectively. For a locally integrable signal w: $\mathbb{R}_0^+ \to \mathbb{R}^n$, we denote by $\|w\|_{\mathcal{L}_2} = \sqrt{\int_0^\infty |w(t)|^2 \mathrm{d}t}$ its $\mathcal{L}_2$-norm, provided the integral is finite. Furthermore, we define the set of all locally integrable signals with a finite $\mathcal{L}_2$-norm as $\mathcal{L}_2$. For the sake of brevity, $\begin{bmatrix} A & B \\ B^{\mathrm{T}} & C \end{bmatrix}$ sometimes is written as $\begin{bmatrix} A & B \\ * & C \end{bmatrix}$.

The notions of exponential stability and $\mathcal{L}_2$-gain will be central to our later discussion. These are defined next.

Definition II.1. *(Exponential Stability) [24] A system $\dot{\xi}(t) = f(\xi(t))$, $t \in \mathbb{R}_0^+$, $\xi(t) \in \mathbb{R}^n$ is said to be globally exponentially stable (GES), if there exist $a, c \in \mathbb{R}^+$ that for any $t_0 \geq 0$ satisfy $|\xi(t, \xi(0))| \leq c|\xi(0)|e^{-at}, \forall t \geq t_0$.*

Definition II.2. *($\mathcal{L}_2$-gain) [12] The system $\dot{\xi}(t) = f(\xi(t), w(t))$, $z(t) = g(\xi(t), w(t))$ is said to have an $\mathcal{L}_2$-gain from w to z smaller than or equal to γ, if there exists a $\mathcal{K}_\infty$ function $\delta : \mathbb{R}^{n_\xi} \to \mathbb{R}^+$ such that for any $w \in \mathcal{L}_2$, any initial state $\xi(0) = \xi_0 \in \mathbb{R}^{n_\xi}$, the corresponding solution to the system satisfies $\|z\|_{\mathcal{L}_2} \leq \delta(\xi_0) + \gamma\|w\|_{\mathcal{L}_2}$.*

Consider a linear time-invariant (LTI) system as

$$\begin{cases} \dot{\xi}_p(t) = A_p\xi_p(t) + B_p\hat{v}(t) + Ew(t) \\ y(t) = C_p\xi_p(t), \end{cases} \tag{1}$$

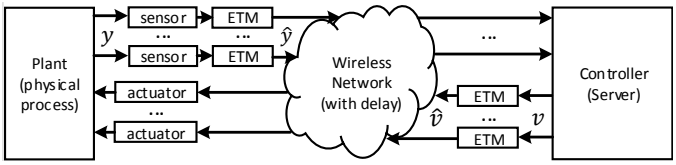


Figure 1. System architecture.

where $\xi_p(t) \in \mathbb{R}^{n_p}$, $\hat{v}(t) \in \mathbb{R}^{n_v}$, $y(t) \in \mathbb{R}^{n_y}$, and $w(t) \in \mathbb{R}^{n_w}$ denote the state vector, input vector, output vector, and disturbance of the plant at time t respectively.

The plant is controlled by a continuous time linear controller, given as

$$\begin{cases} \dot{\xi}_c(t) = A_c\xi_c(t) + B_c\hat{y}(t) \\ v(t) = C_c\xi_c(t) + D_c\hat{y}(t) \end{cases} \tag{2}$$

where $\xi_c(t_k) \in \mathbb{R}^{n_c}$, $\hat{y}(t_k) \in \mathbb{R}^{n_y}$, and $v(t_k) \in \mathbb{R}^{n_v}$ are the state vector, input vector, and output vector of the controller at time t respectively.

For the implementation (1) and (2), we denote two vectors:

$$\hat{u}(t) := \begin{bmatrix} \hat{y}(t) \\ \hat{v}(t) \end{bmatrix} \in \mathbb{R}^{n_u}, u(t) := \begin{bmatrix} y(t) \\ v(t) \end{bmatrix} \in \mathbb{R}^{n_u}, \tag{3}$$

where $n_u := n_y + n_v$. From definition, it is easy to see that $\hat{u}(t_k)$ and $u(t)$ are input and output of the implementation respectively.

We assume that, when $\hat{u}(t) = u(t)$, $\forall t \in \mathbb{R}_0^+$, the implementation (1) and (2) is stable. In this work, we consider the scenario that, the input $\hat{u}(t)$ is sampled and updated from the output $u(t)$ by a dynamic decentralized periodic ETM. The system structure is shown in Figure 1. The design of this ETM will be the center of the work.

Consider the following periodic time sequence:

$$\mathcal{T}_k := \{t_k | t_k := kh,\, k \in \mathbb{N}\}. \tag{4}$$

The sensors are required to measure the local system output in either continuous-time or discrete-time at each $t \in \mathcal{T}_k$. The continuous-time measurement is not mandatory. Our simulation result in Section IV will show that, for some systems this continuous-time measurement can further reduce the number of events.

The local ETMs, on the other hand, only execute at each $t \in \mathcal{T}_k$, after receiving the new measurements. At each sampling time t_k, define $o_i(t_k)$, $i \in \{1, \cdots, n_u\}$ the local information available to the ETM, $\Psi_i(o_i) \geq 0$ the local event-triggered condition, which will be designed later. Whenever this event-triggered condition is satisfied, the corresponding output elements will be transmitted. Define a set $\mathcal{J}$ indicates if there is an event at each t_k:

$$\mathcal{J} := \{i | i \in \{1, \ldots, n_u\}, \Psi_i(o_i) \geq 0\}. \tag{5}$$

Define $\mathcal{J}_c := \{1, \cdots, n_u\} \setminus \mathcal{J}$.

In this work, we also consider transmission delays introduced by the networks. We assume that, the transmission delay is a fixed value d satisfies $0 \leq d < h$. Fixed d is applicable, since all the transmissions that have delays between $[0, d]$ can be updated at the same time d, regardless of the various

receiving times. Thus, the measurement $u_i(t_k)$ that triggers an event at t_k will be received and used to update the system input $\hat{u}_i(t)$ at $t_k + d$.

In between two update time $t_k + d$ and $t_{k+1} + d$, the input $\hat{u}(t)$ will hold by a zero-order hold (ZOH) mechanism. To analyze the system dynamics with this transmission delay, a memory vector $s(t) := \begin{bmatrix} s_y^{\mathrm{T}}(t) & s_v^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}}$ is introduced, where $s(t) \in \mathbb{R}^{n_u}$, $s_y(t) \in \mathbb{R}^{n_y}$, and $s_v(t) \in \mathbb{R}^{n_v}$. $s(t)$ is updated as $\forall t =]t_k, t_{k+1}]$:

$$s_i(t) := \begin{cases} u_i(t_k),\ i \in \mathcal{J} \\ s_i(t_k),\ i \in \mathcal{J}_c, \end{cases} \tag{6}$$

and $\hat{u}(t)$ is updated by $\forall t =]t_k + d, t_{k+1} + d]$:

$$\hat{u}_i(t) = \begin{cases} s_i(t_k + d),\ i \in \mathcal{J} \\ \hat{u}_i(t_k + d),\ i \in \mathcal{J}_c. \end{cases} \tag{7}$$

At each sampling time t_k, all those sensors that have events, i.e. $\forall i \in \mathcal{J}$, their new measurements $u_i(t_k)$ are updated to $s_i(t)$ first; after delay d, these new measurements $u_i(t_k)$ will then be updated to the system input $\hat{u}_i(t)$ via $s_i(t)$. This procedure represents the event-triggered update mechanism with respect to transmission delay. And Equation (6) and (7) describe this procedure in mathematics.

For the hybrid system (1) (2) (4) (5) (6) and (7), define its state as $(\xi(t), \tau(t))$, where

$$\begin{cases} \xi(t) := \begin{bmatrix} \xi_p^{\mathrm{T}}(t) & \xi_c^{\mathrm{T}}(t) & \hat{u}^{\mathrm{T}}(t) & s^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{n_\xi} \\ \tau(t) := t - t_k,\ t \in [t_k, t_{k+1}], \end{cases} \tag{8}$$

in which $n_\xi := n_p + n_c + 2n_u$, $\tau : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ is the elapsed time since the last sampling time and it satisfies

$$\dot{\tau}(t) = 1,\ t \in [t_k, t_{k+1}].$$

Denote a set $\mathcal{X}$ as $(x, r) \in \mathcal{X} \subseteq \mathbb{R}^{n_\xi} \times [0, h]$, such that $x = \xi(t)$ and $r = \tau(t)$ for some $t \in \mathbb{R}_0^+$.

Following [25], now we conclude the dynamics of the hybrid system as:

1) Flow set: The flow set is given by:

$$\mathcal{C} := \{(x, r) | (x, r) \in \mathcal{X}, r \neq d, r \neq h\}. \tag{9}$$

2) Flow dynamic: The flow dynamic is given by:

$$(\dot{x}, \dot{\tau}) \in \mathcal{F}(x, r) := (\bar{A}x + \bar{B}\omega, 1), \tag{10}$$

where $\mathcal{F}$ is the flow map, and

$$\bar{A} := \begin{bmatrix} A_p & \mathbf{0} & \mathbf{0} & B_p & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_c & B_c & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},\ \bar{B} := \begin{bmatrix} E \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix},$$

$\mathbf{0}$ is a zero matrix of proper dimension.

3) Jump set: The jump set is given as:

$$\mathcal{D} := \mathcal{D}_d \cup \mathcal{D}_h, \tag{11}$$

where

$$\mathcal{D}_d := \{(x, r) | (x, r) \in \mathcal{X}, r = d)\}$$
$$\mathcal{D}_h := \{(x, r) | (x, r) \in \mathcal{X}, r = h)\}.$$

4) Jump dynamic: The jump dynamic is given as:

$$(x^+, r^+) \in \mathcal{G}(x, r) := \begin{cases} \mathcal{G}_{\tilde{J}_\mathcal{J}}(x, r),\ \forall (x, r) \in \mathcal{D}_d \\ \mathcal{G}_{\acute{J}_\mathcal{J}}(x, r),\ \forall (x, r) \in \mathcal{D}_h, \end{cases} \tag{12}$$

where $\mathcal{G}_{\tilde{J}_\mathcal{J}}$ and $\mathcal{G}_{\acute{J}_\mathcal{J}}$ are jump maps:

$$\mathcal{G}_{\tilde{J}_\mathcal{J}}(x, r) := (\tilde{J}_\mathcal{J} x, r)$$
$$\mathcal{G}_{\acute{J}_\mathcal{J}}(x, r) := (\acute{J}_\mathcal{J} x, 0),$$

with

$$\tilde{J}_\mathcal{J} := \begin{bmatrix} I & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I - \Gamma_\mathcal{J}^y & \mathbf{0} & \Gamma_\mathcal{J}^y & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I - \Gamma_\mathcal{J}^v & \mathbf{0} & \Gamma_\mathcal{J}^v \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & I \end{bmatrix}$$

$$\acute{J}_\mathcal{J} := \begin{bmatrix} I & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I & \mathbf{0} & \mathbf{0} \\ \Gamma_\mathcal{J}^y C_p & \mathbf{0} & \mathbf{0} & \mathbf{0} & I - \Gamma_\mathcal{J}^y & \mathbf{0} \\ \mathbf{0} & \Gamma_\mathcal{J}^v C_c & \Gamma_\mathcal{J}^v D_c & \mathbf{0} & \mathbf{0} & I - \Gamma_\mathcal{J}^v \end{bmatrix},$$

in which I is an identity matrix of proper dimension. $\Gamma_\mathcal{J}^y$ and $\Gamma_\mathcal{J}^v$ are diagonal matrices such that:

$$\Gamma_\mathcal{J}^y(i, i) = \begin{cases} 1,\ i \in \mathcal{J} \\ 0,\ i \in \mathcal{J}_c \end{cases},\ \Gamma_\mathcal{J}^v(i, i) = \begin{cases} 1,\ i \in \mathcal{J} \\ 0,\ i \in \mathcal{J}_c. \end{cases}$$

Finally, we give a performance function as:

$$z(t) := \bar{C}\xi(t) + \bar{D}w(t), \tag{13}$$

where $\bar{C}$ and $\bar{D}$ are pre-designed matrices.

III. Main result

Inspired by [21], consider some dynamic variables $\eta_i(t)$, $i \in \{1, \cdots, n_u\}$, whose dynamics are given as:

$$\begin{aligned} \dot{\eta}_i(t) &= -2\rho\eta_i(t) + k_i^f u_i^2(t),\ (x, r) \in \mathcal{C} \\ \eta_i(t^+) &= \begin{cases} \eta_i(t) + k_i^g u_i^2(t),\ i \in \mathcal{J}, (x, r) \in \mathcal{D}_h \\ \eta_i(t) + k_i^{g_c} u_i^2(t) - \beta_i \xi^{\mathrm{T}}(t) Q_i \xi(t) \\ i \in \mathcal{J}_c, (x, r) \in \mathcal{D}_h \\ \eta_i(t),\ (x, r) \in \mathcal{D}_d, \end{cases} \end{aligned} \tag{14}$$

where ρ is a pre-defined positive scalar, k_i^f, k_i^g, β_i, and $k_i^{g_c}$, are pre-defined non-negative scalars. $Q_i \in \mathbb{R}^{n_\xi} \times \mathbb{R}^{n_\xi}$ is a matrix defined as:

$$Q_i := \begin{bmatrix} (1 - \sigma_i) C^{\mathrm{T}} \Gamma_i C & (1 - \sigma_i) C^{\mathrm{T}} \Gamma_i D - C^{\mathrm{T}} \Gamma_i & \mathbf{0} \\ * & (D - I)^{\mathrm{T}} \Gamma_i (D - I) - \sigma_i D^{\mathrm{T}} \Gamma_i D & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

where σ_i is a positive scalar, $\Gamma_i \in \mathbb{R}^{n_u} \times \mathbb{R}^{n_u}$ is a matrix with $\Gamma_i(i, i) = 1$ and all the other elements equal to zero. C and D are matrices defined as:

$$C := \begin{bmatrix} C_p & \mathbf{0} \\ \mathbf{0} & C_c \end{bmatrix},\ D := \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ D_c & \mathbf{0} \end{bmatrix}.$$

From (14), one can see that, $\eta_i(t)$ is updated base on $\eta_i(0)$, ρ, k_i^f, k_i^g, $k_i^{g_c}$, β_i, $\hat{u}_i(t)$, and $u_i(t)$, all of which can be pre-stored or obtained locally.

By now we can give the decentralized event-triggered condition as:

$$\begin{aligned}\mathcal{J} &= \{i|\Phi(o_i(t_k)) \geq 0\} \\ &= \{i|\eta_i(t_k) \leq 0 \wedge \xi^{\mathrm{T}}(t_k)Q_i\xi(t_k) \geq 0\}.\end{aligned} \tag{15}$$

(15) shows that at each time t_k, $k \in \mathbb{N}$, after receiving the new measurements, the mechanism will check if the quadratic condition $\xi^{\mathrm{T}}(t_k)Q_i\xi(t_k) \geq 0$ (representing the local sampling error is big enough) and the condition $\eta_i(t_k) \leq 0$ (representing the local dynamic variable is non-positive) are satisfied. If both conditions are satisfied, the mechanism will determine that there is an event locally.

According to (6) and (7), we can deduce that at time t_k, if $\xi^{\mathrm{T}}(t_k)Q_i\xi(t_k) \geq 0$, the following inequalities hold:

$$\begin{cases}\xi^{\mathrm{T}}(t_k)\acute{Q}_i\xi(t_k) \geq 0 \\ \xi^{\mathrm{T}}(t_k+d)\tilde{Q}_i\xi(t_k+d) \geq 0,\end{cases} \tag{16}$$

where

$$\acute{Q}_i := \begin{bmatrix}(1-\sigma_i)C^{\mathrm{T}}\Gamma_i C & \mathbf{0} & (1-\sigma_i)C^{\mathrm{T}}\Gamma_i D - C^{\mathrm{T}}\Gamma_i \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ * & * & (D-I)^{\mathrm{T}}\Gamma_i(D-I) - \sigma_i D^{\mathrm{T}}\Gamma_i D\end{bmatrix}$$

$$\tilde{Q}_i := \begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0} \\ * & \Gamma_i & -\Gamma_i \\ * & * & (1-\sigma_i)\Gamma_i\end{bmatrix}.$$

Consider the following Riccati differential equation:

$$\begin{aligned}\frac{d}{dt}P_j =& -\bar{A}^{\mathrm{T}}P_j - P_j\bar{A} - Y^{\mathrm{T}}K^f Y - 2\rho P_j - \theta^{-2}\bar{C}^{\mathrm{T}}\bar{C} \\ &- (P_j\bar{B} + \theta^{-2}\bar{C}^{\mathrm{T}}\bar{D})M(\bar{B}^{\mathrm{T}}P_j + \theta^{-2}\bar{D}^{\mathrm{T}}\bar{C}),\end{aligned} \tag{17}$$

where $M = (I - \theta^{-2}\bar{D}^T\bar{D})^{-1}$, $j \in \{0, 1\}$, and

$$\begin{aligned}Y &:= \begin{bmatrix}C_p & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & C_c & D_c & \mathbf{0} & \mathbf{0} & \mathbf{0}\end{bmatrix} \\ K^f &:= \mathrm{diag}(k_1^f, \cdots, k_{n_u}^f) \\ K_{\mathcal{J}}^g &:= \mathrm{diag}(k_1^k, \cdots, k_{n_u}^k),\end{aligned}$$

with

$$k_i^k = \begin{cases}k_i^g, & \text{if } i \in \mathcal{J} \\ k_i^{g_c}, & \text{if } i \in \mathcal{J}_c.\end{cases}$$

By the definitions, one can see that $u(t) = Y\xi(t)$.

From (17), the Hamiltonian matrix can be constructed as:

$$H := \begin{bmatrix}H_{11} & H_{12} \\ H_{21} & H_{22}\end{bmatrix}, \tag{18}$$

where

$$\begin{cases}H_{11} := \bar{A} + \rho I + \gamma^{-2}\bar{B}M\bar{D}^{\mathrm{T}}\bar{C} \\ H_{12} := \bar{B}M\bar{B}^{\mathrm{T}} \\ H_{21} := -\bar{C}^{\mathrm{T}}(\gamma^2 I - \bar{D}\bar{D}^{\mathrm{T}})^{-1}\bar{C} - Y^{\mathrm{T}}K^f Y \\ H_{22} := -(\bar{A} + \rho I + \gamma^{-2}\bar{B}M\bar{D}^{\mathrm{T}}\bar{C})^{\mathrm{T}}.\end{cases}$$

From (18), the matrix exponential can be obtained as:

$$F(r) := e^{-Hr} = \begin{bmatrix}F_{11}(r) & F_{12}(r) \\ F_{21}(r) & F_{22}(r)\end{bmatrix}. \tag{19}$$

Assumption III.1. *$F_{11}(r)$ is invertible $\forall r \in [0, h]$.*

Since $F_{11}(0) = I$ and $F_{11}(r)$ is continuous, the Assumption III.1 will always hold for sufficiently small h. If Assumption III.1 holds, $-F_{11}^{-1}(r)F_{12}(r)$ is a positive semi-definite matrix. Then, we can define the matrix $S(r)$ that satisfies $S(r)S^{\mathrm{T}}(r) := -F_{11}^{-1}(r)F_{12}(r)$.

Next, we present the main result of this paper.

Theorem III.2. *Consider the hybrid system (9) (10) (11) (12), event-triggered condition (15), update mechanism (6) (7), assume Assumption III.1 holds, and there exist matrices P_{0h}, $P_{1d} \succ 0$, and scalars k_i^f, k_i^g, β_i, $k_i^{g_c}$, $\acute{\mu}_{\mathcal{J}}^i$, $\acute{\mu}_{\mathcal{J}_c}^i$, $\tilde{\mu}_{\mathcal{J}}^i$, $\tilde{\mu}_{\mathcal{J}_c}^i \geq 0$, $\forall i \in \{1, \cdots, n_u\}$, such that the following LMIs hold:*

$$\begin{aligned}&\begin{bmatrix}G_1 & \acute{J}_{\mathcal{J}}^{\mathrm{T}}F_{11}^{-\mathrm{T}}(d)P_{1d}S(d) & \acute{J}_{\mathcal{J}}^{\mathrm{T}}G_2 \\ * & I - S^{\mathrm{T}}(d)P_{1d}S(d) & \mathbf{0} \\ * & * & G_2\end{bmatrix} \succ 0 \\ &\begin{bmatrix}G_3 & \tilde{J}_{\mathcal{J}}^{\mathrm{T}}F_{11}^{-\mathrm{T}}(h-d)P_{0h}S(h-d) & \tilde{J}_{\mathcal{J}}^{\mathrm{T}}G_4 \\ * & I - S^{\mathrm{T}}(h-d)P_{0h}S(h-d) & \mathbf{0} \\ * & * & G_4\end{bmatrix} \succ 0,\end{aligned} \tag{20}$$

where

$$\begin{aligned}G_1 &:= P_{0h} - \sum_{i\in\mathcal{J}}(\acute{\mu}_{\mathcal{J}}^i\acute{Q}_i) + \sum_{i\in\mathcal{J}_c}(\acute{\mu}_{\mathcal{J}_c}^i\acute{Q}_i) - Y^{\mathcal{T}}K_{\mathcal{J}}^g Y + \Xi \\ G_2 &:= F_{11}^{-\mathrm{T}}(d)P_{1d}F_{11}^{-1}(d) + F_{21}(d)F_{11}^{-1}(d) \\ G_3 &:= P_{1d} - \sum_{i\in\mathcal{J}}(\tilde{\mu}_{\mathcal{J}}^i\tilde{Q}_i) + \sum_{i\in\mathcal{J}_c}(\tilde{\mu}_{\mathcal{J}_c}^i\tilde{Q}_i) \\ G_4 &:= F_{11}^{-\mathrm{T}}(h-d)P_{0h}F_{11}^{-1}(h-d) + F_{21}(h-d)F_{11}^{-1}(h-d),\end{aligned}$$

and

$$\begin{aligned}\Xi &:= \Xi_I^{\mathrm{T}}\Xi_Q\Xi_I \\ \Xi_I &:= \begin{bmatrix}I & \cdots & I\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{(n_\xi n_u)\times n_\xi} \\ \Xi_Q &:= diag\ (\gamma_1\beta_1 Q_1, \cdots \gamma_{n_u}\beta_{n_u}Q_{n_u}) \\ \gamma_i &:= \begin{cases}1,\ i \in \mathcal{J}_c \\ 0,\ i \in \mathcal{J},\end{cases}\end{aligned}$$

then the system is GES with the convergence rate ρ when $w = 0$; and the $\mathcal{L}_2$-gain from w to z is smaller than or equal to γ.

Proof. In this proof, we replace $\sum_{i\in\{1,\cdots,n_u\}}$ by $\sum$ for sake of brevity. Consider the following Lyapunov function:

$$U(x, r) = V(x, r) + \sum \eta_i, \tag{21}$$

where

$$V(x, r) = \begin{cases}x^{\mathrm{T}}P_1(r)x,\ r = [0, d] \\ x^{\mathrm{T}}P_0(r)x,\ r = [d, h].\end{cases}$$

The basic idea of the proof is inspired by the proofs of Theorem III.4 in [26] and Theorem III.2 in [12]. The proof consists of four steps presented as follows.

The first step is to show that η_i is non-negative $\forall t \in \mathbb{R}_0^+$. This step is detailed into 4 cases:

- When $(x,r) \in \mathcal{C}$, by applying comparison lemma (see e.g. Lemma 2.5 in [27]), it is easy to see that $\eta_i(t) > 0$.
- When $(x,r) \in \mathcal{D}_d$, by the result from the case when $(x,r) \in \mathcal{C}$, we have $\eta_i(t) > 0$.
- When $(x,r) \in \mathcal{D}_h$ and $\xi^{\mathrm{T}}(t)Q_i\xi(t) < 0$, $\eta_i(t)$ increases during the jumps. Together with the result when $(x,r) \in \mathcal{C}$, thus $\eta_i(t) > 0$.
- When $(x,r) \in \mathcal{D}_h$ and $\xi^{\mathrm{T}}(t)Q_i\xi(t) \geq 0$, if $\eta_i(t^+) \leq 0$, according to event-triggered condition (15), an event is triggered, i.e. $(x,r) \in \mathcal{J}$. According to (14), $\eta_i(t^+)$ will increase. As a result, $\eta_i(t^+) > 0$.

Then, we need to show that U satisfies $\forall t \in \mathbb{R}_0^+$:

$$c_1|\xi|^2 + \sum|\eta_i| \leq U(\xi,\tau) \leq c_2|\xi|^2 + \sum|\eta_i|.$$

Here, c_1 and c_2 are scalars that satisfy $0 < c_1 \leq c_2$. We finish this step by finding such c_1 and c_2.

According to [26] and [12], $\forall \tau \in [0,h]$, it holds that $P_i(\tau) \succ 0$, $i \in \{0,1\}$, and

$$c_1 = \min\{\min_{\tau\in[0,h]} \lambda_{\min}(P_0(\tau)), \min_{\tau\in[0,d]} \lambda_{\min}(P_1(\tau))\}$$
$$c_2 = \max\{\max_{\tau\in[0,h]} \lambda_{\max}(P_0(\tau)), \max_{\tau\in[0,d]} \lambda_{\max}(P_1(\tau))\}.$$

The third step is to show that U has a supply rate $\theta^{-2}z^{\mathrm{T}}z - w^{\mathrm{T}}w$ and decay rate 2ρ during flow. From (14) (17) and (21), it follows that:

$$\begin{aligned}\frac{d}{dt}U \leq& -2\rho V - \theta^{-2}z^{\mathrm{T}}z + w^{\mathrm{T}}w - u^{\mathrm{T}}K^f u\\ &- \sum(2\rho\eta_i - k_i^f u_i^2)\\ =& -2\rho U - \theta^{-2}z^{\mathrm{T}}z + w^{\mathrm{T}}w.\end{aligned}$$

The last step is to show that U does not increase during jumps. According to the proof of Theorem III.2 of [12], it holds that:

$$\begin{aligned}P_0(d) =& F_{21}(h-d)F_{11}^{-1}(h-d) + F_{11}^{-\mathrm{T}}(h-d)(P_0(h)+\\ & P_0(h)S(h-d)(I - S^{\mathrm{T}}(h-d)P_0(h)S(h-d))^{-1}\\ & S^{\mathrm{T}}(h-d)P_0(h)F_{11}^{-1}(h-d)\\ P_1(0) =& F_{21}(d)F_{11}^{-1}(d) + F_{11}^{-\mathrm{T}}(d)(P_1(d) + P_1(d)S(d)\\ & (I - S^{\mathrm{T}}(d)P_1(d)S(d))^{-1}S^{\mathrm{T}}(d)P_1(d)F_{11}^{-1}(d).\end{aligned}$$

By applying the Schur complement and S-procedure (see e.g. [28]) to (20), it follows that: when $(x,r) \in \mathcal{D}_h$:

$$\begin{aligned}U(t) =& V(t) + \sum\eta_i(t)\\ =& \xi^{\mathrm{T}}(t)P_{0h}\xi(t) + \sum\eta_i(t)\\ \geq& \xi^{\mathrm{T}}(t)(\acute{J}_{\mathcal{J}}^{\mathrm{T}}P_{10}\acute{J}_{\mathcal{J}} + Y^{\mathrm{T}}K_{\mathcal{J}}^g Y - \Xi)\xi(t) + \sum\eta_i(t)\\ =& \xi^{\mathrm{T}}(t)\acute{J}_{\mathcal{J}}^{\mathrm{T}}P_{10}\acute{J}_{\mathcal{J}}\xi(t)\\ & + \sum(\eta_i(t) + k_i^k u_i^2(t) - \gamma_i\beta_i\xi^{\mathrm{T}}(t)Q_i\xi(t))\\ =& \xi^{\mathrm{T}}(t^+)P_{10}\xi(t^+) + \sum\eta_i(t^+) = U(t^+).\end{aligned}$$

When $(x,r) \in \mathcal{D}_d$:

$$\begin{aligned}U(t) =& V(t) + \sum\eta_i(t) = \xi^{\mathrm{T}}(t)P_{1d}\xi(t) + \sum\eta_i(t)\\ \geq& \xi^{\mathrm{T}}(t)\tilde{J}_{\mathcal{J}}^{\mathrm{T}}P_{0d}\tilde{J}_{\mathcal{J}}\xi(t) + \sum\eta_i(t)\\ =& \xi^{\mathrm{T}}(t^+)P_{0d}\xi(t^+) + \sum\eta_i(t^+) = U(t^+).\end{aligned}$$

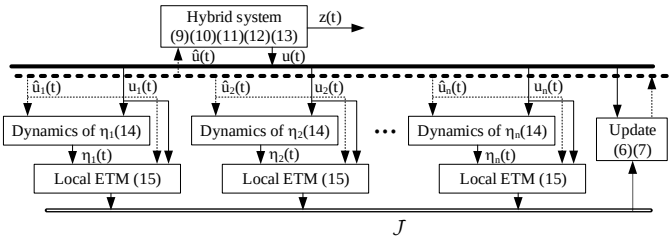


Figure 2. Schematic diagram of the approach, showing the presented method and algorithm.

These 4 steps together guarantee that U is a proper Lyapunov function and the system is GES with a convergence rate ρ when $w = 0$ and has an $\mathcal{L}_2$-gain from w to z smaller than or equal to γ. This concludes our proof. □

When we look at event-triggered condition (15), it is easy to see that the ETM requires execution periodically at each sampling time t_k only. Meanwhile, for the dynamic variable η_i, if $k_i^f = 0$, $\forall i \in \{1,\cdots,n_u\}$, the evolution of η_i does not require continuous system output during flows according to (14). As a result, the system output is only required periodically at each t_k; however, if $\exists i \in \{1,\cdots,n_u\}$ such that $k_i^f \neq 0$, the corresponding system output elements are require continuously.

Normally, one can set $k_i^f = 0$, $\forall i \in \{1,\cdots,n_u\}$ to save sensor node energy consumption, that is sensors typically sleep with discrete time duty-cycles. However, in some cases, $k_i^f \neq 0$ can lead to fewer number of events requiring being triggered, while guaranteeing the same stability and $\mathcal{L}_2$-gain performance. We will show this in Section IV. Besides, if $k_i^f = 0$, $\forall i \in \{1,\cdots,n_u\}$, the first order differential equation in (14) can be turned into a multiplication, since $e^{-2\rho h}$, $e^{-2\rho(h-\tau_d)}$, and $e^{-2\rho\tau_d}$ can be pre-computed and stored in the sensor nodes. This can dramatically reduce the computation load when the system is running. Solving the LMIs (20) should be finished before the execution of the system, and thus it is an off-line computation. We use SeDuMi as the solver for YALMIP. According to the user's guide for SeDuMi interface, the computational complexity of SeDuMi is in $\mathcal{O}(n^2m^{2.5} + m^{3.5})$, in which n is the number of decision variables and m is the number of rows of an LMI.

Remark III.3. *For Theorem III.2, when k_i^f, k_i^g, β_i, $k_i^{g_c} = 0$, $\forall i \in \{1,\cdots,n_u\}$, then $\eta_i(t) = 0$, $\forall i \in \{1,\cdots,n_u\}$, $t \in \mathrm{R}_0^+$. Thus, the dynamic event-triggered condition (15) turns into static event-triggered condition.*

Figure 2 shows a schematic diagram to explain the presented approach. Figure 3 shows a flow chart to show how the presented method works.

IV. Numerical example

We illustrate the presented theory using two numerical examples in this section.

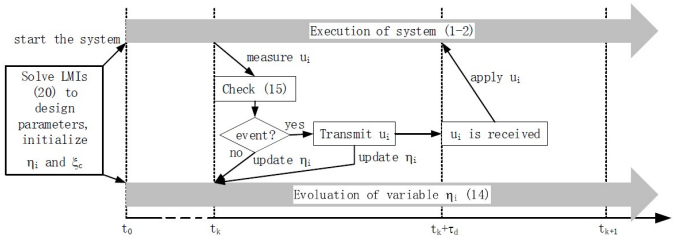


Figure 3. Flow chart of the approach, showing how the presented method works.

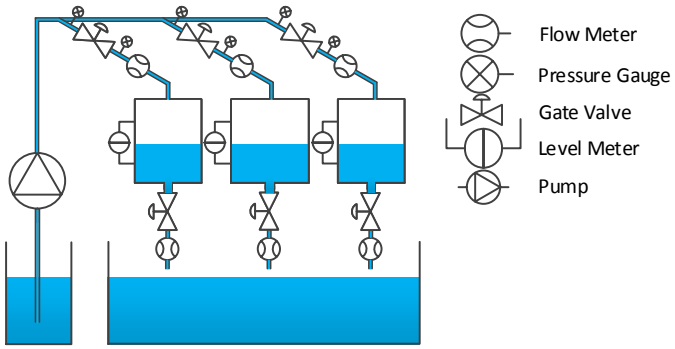


Figure 5. Smart water distribution system structure.

A. Batch reactor

The first example is a batch reactor system from [29]. The controller is given as:

$$A_c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} B_c = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} C_c = \begin{bmatrix} -2 & 0 \\ 0 & 8 \end{bmatrix} D_c = \begin{bmatrix} 0 & -2 \\ 5 & 0 \end{bmatrix}.$$

Other parameters are given as $h = 0.01s$, $\tau_d = 0.001s$, $\rho = 0.01$. For the performance z, we choose $z(t) = [1\,0\,0\,0]\xi_p(t)$. The initial state is given as $\xi_p(0) = [10\ -10\ -10\,10]^{\mathrm{T}}$, $\xi_c(0) = \mathbf{0}$, $\hat{y}(0) = C_p\xi_p(0)$, and $\hat{v}(0) = D_cC_p\xi_p(0)$.

For SDPETC, a feasible solution to the LMI (20) can be found when $\sigma_i = 0.01$, $i \in \{1, \cdots, 4\}$, $\gamma = 0.9$. After executing for $10s$, a total of 896 events are triggered. For DDPETC, let $\eta_i(0) = 0$, $k_i^f = 0$, $\sigma_i = 0.01$, $\forall i \in \{1, \cdots, 4\}$, with $\gamma = 0.9$, a feasible solution can be found. A total of 540 events are triggered after $10s$. One can see that, with the presented DDPETC, the number of events are reduced dramatically. Figure 4 shows when $w = 0$, system state evolution and event instances of SDPETC and DDPETC.

Further simulation shows that, during [0,10] the total event saving is 39.7%, during [7,10] the total saving is 40.2%. Therefore, the savings are almost equal during transient period and after that. Let k_i^f, $i \in \{1, \cdots, 4\}$ equal to 0.01, 0.05, 0.1, a total of 565, 527, 572 events are generated respectively. One can see that, the savings are not obvious for different k_i^f in this system. Therefore, for this system, $k_i^f = 0$ is suggested.

B. Smart water distribution system

The second example is a smart water distribution system based on [19]. The model is implemented in Matlab/Simulink using Simscape toolbox and Simscape Fluids toolbox, both of which are developed for physical systems simulation. The basic structure of the WaterBox testbed used in [19] remains the same, however some of the parameters are modified to be more realistic. Firstly, the scale is enlarged by extending the pipe length. Secondly, the tanks are enlarged to supply enough water to the consumers. Finally, the two pumps are reduced to one to simplify the control strategy and the pump is more powerful to give enough water head to the pipe network. The scale of the current model considers a water distribution system that is used at a wine estate in the south of England, which is about 4 hectares. Figure 5 shows the structure of the water distribution system. The system identification and controller design follow the method given in [19].

The goal of the water distribution system can be simplified as to stabilize the water levels of the three District Meter Area (DMA) tanks at predesigned constant reference levels. The system state is chosen as the difference between the reference levels and the current water levels; the control actions are the open levels of the valves. The pump is executed with a constant velocity as electronic machinery input. Since the system states can be measured directly, the dynamic controller (2) is replaced by a control gain, i.e. $v(t_k) = K\xi_p(t_k)$. And the transmissions from the controller to the actuators are not included in the ETM anymore.

The plant dynamic is identified and linearized at reference level of 3 meters. Its dynamic is shown as:

$$A_p = \begin{bmatrix} -8.367 & 0 & 0 \\ 0 & -6.276 & 0 \\ 0 & 0 & -5.020 \end{bmatrix} \times 10^{-4},$$

$$B_p = \begin{bmatrix} 0.1068 & -0.0371 & -0.0371 \\ -0.0279 & 0.0801 & -0.0279 \\ -0.0223 & -0.0223 & 0.0641 \end{bmatrix}, E = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

The control gain matrix is designed as:

$$K = \begin{bmatrix} -0.3024 & -0.0089 & -0.0238 \\ 0.0228 & -0.3034 & -0.0073 \\ 0.0357 & 0.0215 & -0.3024 \end{bmatrix}.$$

Other parameters are given as $h = 2s$, $\tau_d = 0.5s$, $\rho = 0.001$, $z(t) = [1\,1\,1]\xi_p(t)$. The initial state is $\xi_p(0) = [-3\ -3\ -3]^{\mathrm{T}}$, i.e. the water levels start from 0.

For SDPETC, given $\sigma_1 = \sigma_2 = \sigma_3 = 0.1$, $\gamma = 0.9$, the LMI of (20) is feasible. After executing for $20000s$, a total of 19784 events are triggered. For DDPETC, let $k_1^f = k_2^f = k_3^f = 0$, a feasible solution can be found when $\sigma_1 = \sigma_2 = \sigma_3 = 0.1$, $\gamma = 0.9$. With the same execution period, a total of 9362 events are triggered. This means, a total of 52.68% events are saved, which in turn means the inter-battery change lifetime is more than doubled, for example. Figure 6 shows the state evolution and event instances of SDPETC, DDPETC with $k_1^f = k_2^f = f_3^f = 0$, and DDPETC with $k_1^f = k_2^f = f_3^f = 0.05$. One can see that, the event intervals in DDPETC are much larger than those in SDPETC.

Further simulation shows that, during [6000,20000], DDPETC can save 52.75% events, which is almost equal to the saving of 52.68% during [0,20000]. Therefore, the savings are almost equal during transient period and after that. Let k_1^f, k_2^f, and k_3^f equal to 0.01, 0.02, 0.05, 0.1, a total of

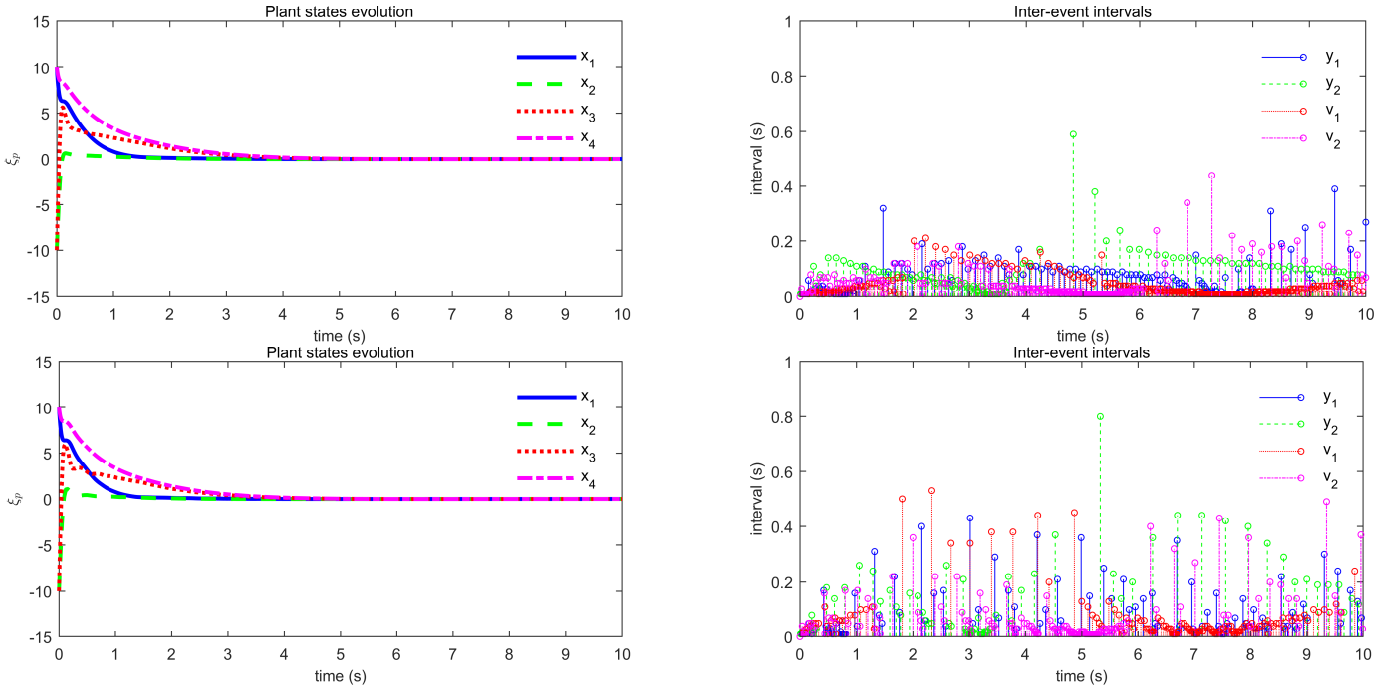


Figure 4. Simulation results of batch reactor with SDPETC (the upper figures) and DDPETC (the lower figures) when $w = 0$: system state evolution and event instances, indicating the event intervals are larger in DDPETC.

8584, 7956, 6716, 4894 events are generated respectively, e.g. 8.31%, 15.2%, 28.26%, 47.72% savings each compared with the case when $k_1^f = k_2^f = k_3^f = 0$. Define the state error as $e(t) = x^o - \xi_p(t)$, where x^o is the equilibrium point, for the smart water distribution system, $x_i^o = 0$, $\forall i \in \{1, 2, 3\}$. After the state is steady, $e_i(t)$ satisfies $e_i(t) \leq \bar{e}$. The $\bar{e}$ are increased by -0.72%, 7.9%, 20.14%, 98.56% respectively, showing a trade-off between the total number of events and the steady state error.

V. Conclusion

In this work, we add dynamic variables to decentralized periodic event-triggered control (DPETC) and thus present a dynamic decentralized periodic event-triggered control (DDPETC). The evolution of the dynamic variables based only on local available information, guaranteeing the event-triggered mechanisms (ETMs) still be decentralized that do not require information from other nodes in the system. With the help of these additional dynamic variables, the DDPETC can trigger fewer events compared with the static decentralized periodic event-triggered control (SDPETC), i.e. the DPETC without the dynamic variables, while guaranteeing the same stability and $\mathcal{L}_2$-gain performance. We illustrate the DDPETC in two examples: an ideal batch reactor and a smart water distribution system built on Simscape toolbox and Simscape fluids toolbox, both of which are for physical systems simulation in Matlab. Both examples have shown the feasibility of the proposed DDPETC. Further results show that, compared with the SDPETC, DDPETC can save more than 40% transmissions. For the smart water distribution system, if the dynamic variables evolute considering the system output continuously, further transmissions can be saved compared with the discrete time system output approach. However, the steady state error will increase as a trade-off. DDPETC employs decentralized event-triggering mechanisms and can largely reduce the transmissions with the help of additional dynamic variables. Therefore, DDPETC is more suitable for wireless cyber-physical systems, whose sensors are not co-located and communication bandwidth is reduced compared with the wired ones. Our further work includes applying this DDPETC to real physical systems.

References

[1] Mahmoud Abdelrahim, Romain Postoyan, Jamal Daafouz, and Dragan Nešić. Stabilization of nonlinear systems using event-triggered output feedback controllers. *IEEE Transactions on Automatic Control*, 61(9):2682–2687, 2016.

[2] Dimos V Dimarogonas, Emilio Frazzoli, and Karl H Johansson. Distributed event-triggered control for multi-agent systems. *IEEE Transactions on Automatic Control*, 57(5):1291–1297, 2012.

[3] MCF Donkers and WPMH Heemels. Output-based event-triggered control with guaranteed $\mathcal{L}_\infty$-gain and improved and decentralized event-triggering. *IEEE Transactions on Automatic Control*, 57(6):1362–1376, 2012.

[4] Fulvio Forni, Sergio Galeani, Dragan Nešić, and Luca Zaccarian. Event-triggered transmission for linear control over communication channels. *Automatica*, 50(2):490–498, 2014.

[5] Toivo Henningsson, Erik Johannesson, and Anton Cervin. Sporadic event-based control of first-order linear stochastic systems. *Automatica*, 44(11):2890–2895, 2008.

[6] Manuel Mazo Jr. and Ming Cao. Asynchronous decentralized event-triggered control. *Automatica*, 50(12):3197–3203, 2014.

[7] Manuel Mazo Jr. and Paulo Tabuada. Decentralized event-triggered control over wireless sensor/actuator networks. *IEEE Transactions on Automatic Control*, 56(10):2456–2461, 2011.

[8] Anton Selivanov and Emilia Fridman. Event-triggered h_infinity control: A switching approach. *IEEE Transactions on Automatic Control*, 61(10):3221–3226, 2016.

[9] Paulo Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, 2007.

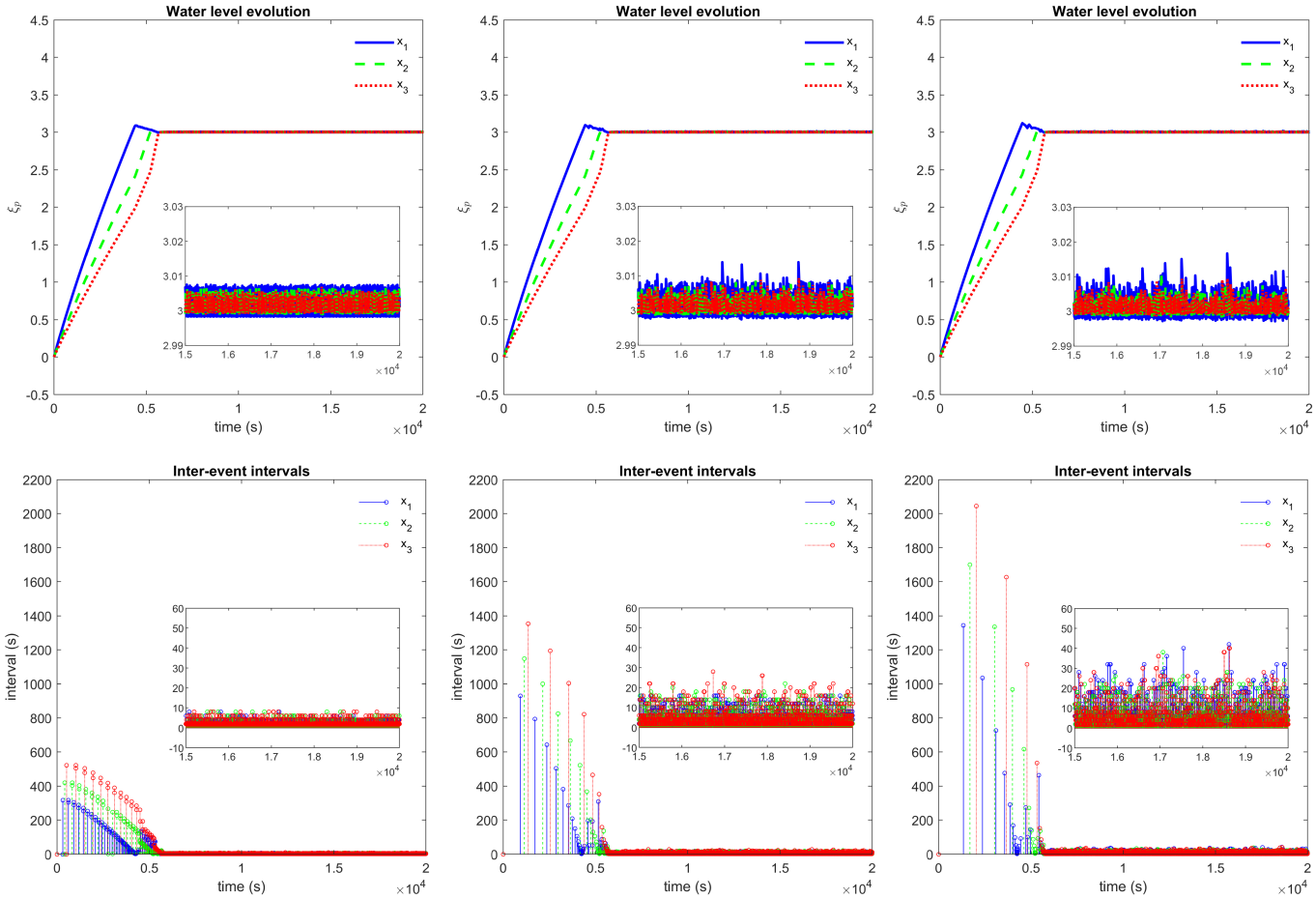


Figure 6. Siumulation results of smart water distribution system with SDPETC (left), DDPETC with $k_1^f = k_2^f = k_3^f = 0$ (middle), and DDPETC with $k_1^f = k_2^f = f_3^f = 0.05$ (right): state evolution and event instances. The floating figures show the zoomed details during the period [15000,20000]. Showing the event intervals are much larger in DDPETC; and the steady state error and the event intervals increase when $k_1^f = k_2^f = f_3^f = 0.05$.

[10] Pavankumar Tallapragada and Nikhil Chopra. Decentralized event-triggering for control of nonlinear systems. *IEEE Transactions on Automatic Control*, 59(12):3312–3324, 2014.

[11] Xiaofeng Wang and Michael D Lemmon. On event design in event-triggered feedback systems. *Automatica*, 47(10):2319–2322, 2011.

[12] WPMH Heemels, MCF Donkers, and Andrew R Teel. Periodic event-triggered control for linear systems. *IEEE Transactions on Automatic Control*, 58(4):847–861, 2013.

[13] Dong Yue, Engang Tian, and Qing-Long Han. A delay system method for designing event-triggered controllers of networked control systems. *IEEE Transactions on Automatic Control*, 58(2):475–481, 2013.

[14] P. Yu and D. V. Dimarogonas. Explicit computation of sampling period in periodic event-triggered multiagent control under limited data rate. *IEEE Transactions on Control of Network Systems*, 6(4):1366–1378, 2019.

[15] Anqi Fu and Manuel Mazo, Jr. Decentralized periodic event-triggered control with quantization and asynchronous communication. *Automatica*, 94:294–299, 2018.

[16] Xiaohua Ge and Qing-Long Han. Distributed event-triggered h filtering over sensor networks with communication delays. *Information Sciences*, 291:128–142, 2015.

[17] Yanpeng Guan, Qing-Long Han, and Xiaohua Ge. On asynchronous event-triggered control of decentralized networked systems. *Information Sciences*, 425:127–139, 2018.

[18] WPMH Heemels and MCF Donkers. Model-based periodic event-triggered control for linear systems. *Automatica*, 49(3):698–711, 2013.

[19] Sokratis Kartakis, Anqi Fu, Manuel Mazo Jr., and Julie A McCann. Communication schemes for centralized and decentralized event-triggered control systems. *IEEE Transactions on Control Systems Technology*, 2017.

[20] Mahmoud Abdelrahim, Victor Sebastiaan Dolk, and WPMH Heemels. Event-triggered quantized control for input-to-state stabilization of linear systems with distributed output sensors. *IEEE Transactions on Automatic Control*, 2019.

[21] DP Borgers, VS Dolk, and WPMH Heemels. Dynamic periodic event-triggered control for linear systems. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, pages 179–186. ACM, 2017.

[22] VS Dolk, DP Borgers, and WPMH Heemels. Output-based and decentralized dynamic event-triggered control with guaranteed l_ p-gain performance and zeno-freeness. *IEEE Transactions on Automatic Control*, 62(1):34–49, 2017.

[23] Antoine Girard. Dynamic triggering mechanisms for event-triggered control. *IEEE Transactions on Automatic Control*, 60(7):1992–1997, 2015.

[24] Eduardo D Sontag. Input to state stability: Basic concepts and results. In *Nonlinear and optimal control theory*, pages 163–220. Springer, 2008.

[25] Rafal Goebel, Ricardo G Sanfelice, and Andrew R Teel. *Hybrid Dynamical Systems: modeling, stability, and robustness*. Princeton University Press, 2012.

[26] DP Borgers, VS Dolk, and WPMH Heemels. Riccati-based design of event-triggered controllers for linear systems with delays. *IEEE Transactions on Automatic Control*, 63(1):174–188, 2018.

[27] Hassan K Khalil. *Noninear Systems*. Prentice-Hall, New Jersey, 1996.

[28] Stephen P Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear matrix inequalities in system and control theory*, volume 15. SIAM, 1994.

[29] Gregory C Walsh and Hong Ye. Scheduling of networked control systems. *Control Systems, IEEE*, 21(1):57–65, 2001.