IFAC Journal of Systems and Control 29 (2024) 100268

Contents lists available at ScienceDirect

IFAC Journal of Systems and Control

journal homepage: www.elsevier.com/locate/ifacsc

Full length article

# Periodic event-triggered controller design with Bayesian optimization: An emulation-based approach

Kazumune Hashimoto

*Graduate School of Engineering, Osaka University, Yamadaoka 2-1, Suita, Japan*

ARTICLE INFO

*Article history:*
Received 11 February 2024
Received in revised form 30 May 2024
Accepted 17 June 2024
Available online 24 June 2024

*Keywords:*
Event-triggered control
Self-triggered control
Bayesian optimization

ABSTRACT

In recent years, event-triggered control has emerged as a promising strategy for addressing resource constraints in networked control systems (NCSs), such as limited life-time of battery capacity. This paper explores the development of a periodic event-triggered controller through a model-free design, assuming unknown plant dynamics. The design of the event-triggered controller employs an emulation-based approach, which is divided into solving two sub-problems: the problem for designing parameters for the time-triggered (periodic) control law, and the problem for designing parameters for the event-triggered condition. In particular, both problems will be solved through the usage of two distinct Bayesian optimization algorithms. The first problem is addressed by adapting basic Bayesian optimization to include network utilization by considering the number of communication time steps during optimization. The second problem employs constrained Bayesian optimization to incorporate explicit performance constraints within the optimization process. A numerical example is provided to demonstrate the effectiveness of the proposed method.



## 1. Introduction

Event-triggered control is one of the promising control schemes that deal with resource constraints, such as limited life-time of battery capacity for networked control systems (see, e.g., Heemels, Johansson, and Tabuada (2012)). The basic idea of the event-triggered control is to trigger a control execution or a networked communication only after a certain occurrence of events, which are generated by a well-designed event-triggered condition. Mainly, the design procedure of the event-triggered control has two approaches. The first one is the so-called *emulation-based* approach (see, e.g., Heemels et al. (2012)). In this approach, we first design parameters for the control law with an assumption that the time-triggered (periodic) controller is implemented. These parameters are designed by evaluating a certain performance index, such as a Lyapunov function candidate, an infinite-horizon cost function, and so on. Once the parameters for the control law are designed, we then design parameters for an event-triggered condition. Indeed, the majority of the existing design methods of the event-triggered control follows the emulation-based approach, such as the ones based on Lyapunov stability (Tabuada, 2007), $\mathcal{L}_2/\mathcal{L}_\infty$-gain stability (Donkers & Heemels, 2011; Wang & Lemmon, 2009), quadratic cost performance (Antunes & Heemels, 2014; Gommans, Antunes, Donkers, Tabuada, & Heemels, 2014; Seuret, Prieur, Tarbouriech, & Zaccarian, 2013), cyber security (Qi, Tang, Zhao, Xing, & Qiu, 2023), for multi-agent systems (Anand, Kumar, Kumar, & Arkdev, 2023; Berkel & Liu, 2018; Boardman, Harden, & Martínez, 2021; Dimagoronas, Frazzoli, & Johansson, 2012; Hashimoto, Onoue, Ogura, & Ushio, 2021; Li, Wang, Xu, & Fang, 2023) to name a few. In addition, some sophisticated approaches of the event-triggered control schemes have been developed, such as a *periodic* event-triggered control, in which the event-triggered condition is evaluated only periodically (not continuously) (Hashimoto, Adachi, & Dimarogonas, 2017; Heemels & Donkers, 2013; Postoyan, Anta, Heemels, Tabuada, & Nesic, 2013). Moreover, a dynamic event-triggered control has been proposed, in which the threshold of the event-triggered condition is dynamic (i.e., time-varying) (Girard, 2014). The second one is the so-called *co-design* approach, in which both the controller and the event-triggered condition are simultaneously designed, see, e.g., Seuret, Prieur, Tarbouriech, and Zaccarian (2016) and Tarbouriech, Seuret, da Silva, and Sbarbaro (2016).

In this paper, we are interested in designing an event-triggered controller with an assumption that the dynamics of the plant is *unknown* apriori. Such a scenario holds practical significance, particularly when deriving an accurate model of the plant proves challenging, given the complexities and nonlinearity inherent in its dynamics. This challenge commonly arises in various applications, such as mechanical systems, autonomous vehicles, power consumption management in multi-story buildings, and addressing periodic errors in astrophotography systems, among others

*E-mail address:* hashimoto@eei.eng.osaka-u.ac.jp.

https://doi.org/10.1016/j.ifacsc.2024.100268

(Beckers, Kulic, & Hirche, 2019; Hashimoto, Yoshimura, & Ushio, 2020; Hewing, Liniger, & Zeilinger, 2018; Jain, Nghiem, Morari, & Mangharam, 2018; Klenske, Zeilinger, Scholkopf, & Hennig, 2019). Specifically, we advocate for an emulation-based, model-free approach to design a periodic event-triggered controller. The choice of this approach stems from its capacity to replicate control performance across both time-triggered and event-triggered conditions. By strategically designing parameters for the event-triggered condition, we aim to minimize communication time steps while maintaining control performance comparable to the time-triggered counterpart (refer to Section 3.4 for detailed elaboration). Our methodology involves decomposing the problem into two sub-problems: designing parameters for the time-triggered control law and configuring parameters for the event-triggered condition. The former problem will be formulated as an optimization problem that minimizes a certain cost function involving state-and-input trajectories. The latter problem will be formulated as a *constrained* optimization problem that minimizes the number of communication time steps while satisfying a certain constraint on the control performance. Both of the two problems are indeed black-box optimization problems, due to the fact that the dynamics of the plant is unknown apriori. In this paper, we propose to employ the *Bayesian optimization* (Mockus, 1989) in order to solve the two problems. The Bayesian optimization is known to be a useful method to solve black-box optimization problems (see, e.g., Mockus (1989)). The Bayesian optimization algorithm becomes particularly advantageous over swarm intelligence algorithms (e.g. particle swarm optimization) when an objective function is expensive to measure (e.g., it requires a very long time to run the experiment for measuring control performances). In other words, it is known to efficiently find the optimal parameter with a small number of data points that are observed from experiment. Therefore, application of the Bayesian optimization to the design of event-triggered strategy for networked control system is crucial, since network resources such as the life-time of the battery powered devices are often limited and the number of experimental evaluations should be kept as small as possible. While the Bayesian optimization has been originally developed in the machine learning community as a way to tune hyper-parameters (Frazier, 0000), it has been gaining an increased attention to tune controllers for dynamical systems, see, e.g., Abbracciavento, Zinnari, Formentin, Bianchessi, and Savaresi (2023), Bansal, Calandra, Xiao, Levine, and Tomlin (2017), Neumann-Brosig, Marco, Schwarzmann, and Trimpe (2020) and Marco, Hennig, Bohg, Schaal, and Trimpe (2016). In this paper, we propose two different types of the Bayesian optimization algorithms for solving the two problems. While the former problem (design of the time-triggered control law) could be solved via the basic Bayesian optimization algorithm, it will be pointed out that it could result in a significant amount of communication time steps (i.e., the network utilization) for solving the optimization problem. Therefore, we propose an alternative algorithm by estimating the performance of the time-triggered controller through the implementation of the event-triggered controller, aiming to achieve resource savings for NCSs. The illustrative example will demonstrate the applicability of our approach, showing that the proposed algorithm results in a significant reduction of the number of the communication time steps with respect to the basic Bayesian optimization algorithm (for details, see Section 6). For the latter problem (design of the event-triggered condition), it will be pointed out that the basic Bayesian optimization algorithm is not applicable, since we need to deal with a *constraint* that is imposed in the optimization problem. Hence, we propose an alternative Bayesian optimization algorithm, which is based on a *constrained* Bayesian optimization (Gardner, Kusner, Xu, Weinbrger, & Cunningham, 2014), which can deal with constraints that are imposed in the optimization problem. As a consequence, solving the above two problems allows us to obtain the event-triggered controller, such that the number of communication time steps is minimized while achieving the control performance close enough to the time-triggered case.

*(Main contribution of this paper):* As previously mentioned, various automatic controller tuning methods for dynamical systems using Bayesian optimization have been introduced in recent years, as evidenced by references such as Abbracciavento et al. (2023), Bansal et al. (2017), Neumann-Brosig et al. (2020) and Marco et al. (2016). However, we contend that existing research has not explored the design of event-triggered controllers. This paper's primary contribution is the introduction of Bayesian optimization algorithms for crafting a periodic event-triggered controller through an emulation-based strategy. As we will see in later sections, synthesizing the periodic event-triggered controller based on the Bayesian optimization is not trivial, since a direct application of the Bayesian optimization would lead to significant expense of network utilization while optimizing the control parameters (for details, see Section 4). Specifically, our contributions are twofold:

- We introduce a resource-aware algorithm to configure parameters for the control law, ensuring that the number of communication time steps (i.e., network utilization) is explicitly considered during the black-box optimization process.
- We utilize a constrained Bayesian optimization algorithm for designing parameters for the event-triggered condition, aimed at reducing communication time steps while adhering to a specified control performance constraint for the event-triggered controller.

This paper is organized as follows. In Section 2, we provide preliminaries of the Gaussian process regression and a basic algorithm of the Bayesian optimization. In Section 3, we formulate a problem to design parameters for a time-triggered control law (Problem 1), and a problem to design parameters for an event-triggered condition (Problem 2). In Section 4, we provide a solution approach to Problem 1. In Section 5, we provide a solution approach to Problem 2. In Section 6, we illustrate the effectiveness of the proposed approach through a simulation example. We finally conclude in Section 7.

*Notation.* Let $\mathbb{N}$, $\mathbb{N}_{\geq 0}$, $\mathbb{N}_{>0}$, $\mathbb{N}_{a:b}$ be the sets of integers, non-negative integers, positive integers, and the integers in the interval $[a, b]$, respectively. Let $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{a:b}$ be the sets of reals, non-negative reals, positive reals, and the reals in the interval $[a, b]$, respectively. We denote by $\|\cdot\|$ the Euclidean norm.

## 2. Preliminaries

In this section, we provide the concept of the Gaussian process regression and a basic algorithm of the Bayesian optimization.

### 2.1. Gaussian process regression

We first recall some basic concepts of the Gaussian process (GP) regression (Rasmussen & Williams, 2006). Consider a static nonlinear input–output map: $y = G(\theta)$, where $\theta \in \mathbb{R}^n$ is the input, $y \in \mathbb{R}$ is the output and $G : \mathbb{R}^n \to \mathbb{R}$ is the *unknown* function. In the general regression problem, the goal is to estimate $G(\cdot)$ based on a set of input–output training data, i.e., $\mathcal{D} = \{(\theta_j, y_j)\}_{j=1}^N$ with $y_j = G(\theta_j), j \in \mathbb{N}_{1:N}$, where $N$ denotes the number of training data. In the *GP* regression, it is particularly assumed that the function $G(\cdot)$ follows the GP. That is, for every set of training inputs $\theta_j \in \mathbb{R}^n$, $j = 1, \ldots, N$, the joint probability of the corresponding set of

outputs $Y = [y_1, y_2, \ldots, y_N]^\mathsf{T}$ follows the multivariate Gaussian distribution, i.e., $Y \sim \mathcal{N}(0, K)$, where $K \in \mathbb{R}^{N\times N}$ is the covariance matrix. Specifically, $K$ is given of the form $K_{ij} = \mathsf{k}(\theta_i, \theta_j)$, where $K_{ij}$ is the $(i, j)$-component of $K$ and $\mathsf{k} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is the positive definite kernel function. An example of the kernel function is the squared exponential covariance function:

$$\mathsf{k}(\theta_i, \theta_j) = \alpha^2 \exp\left(-\frac{1}{2}(\theta_i - \theta_j)^\mathsf{T} \Lambda^{-1}(\theta_i - \theta_j)\right), \tag{1}$$

where $\Lambda = \mathrm{diag}\left(\lambda_1^2, \ldots, \lambda_N^2\right)$, with $\{\alpha, \lambda_1, \ldots, \lambda_N\}$ being the hyper parameters. Given the training data $\mathcal{D} = \{(\theta_j, y_j)\}_{j=1}^N$, it is shown that the predictive distribution of the output of $G$ with a new test input $\theta$ follows the Gaussian distribution: $p(y|\theta, \mathcal{D}) = \mathcal{N}(\mu(\theta), \sigma^2(\theta))$, where

$$\mu(\theta) = \mathsf{k}_*^\mathsf{T}(\theta) K^{-1} Y, \tag{2}$$

$$\sigma^2(\theta) = \mathsf{k}(\theta, \theta) - \mathsf{k}_*^\mathsf{T}(\theta) K^{-1} \mathsf{k}_*(\theta), \tag{3}$$

with $\mathsf{k}_*(\theta) = [\mathsf{k}(\theta, \theta_1), \ldots, \mathsf{k}(\theta, \theta_N)]^\mathsf{T}$.

### 2.2. Bayesian optimization

Here we recall a basic algorithm of the Bayesian optimization (Frazier, 0000). The Bayesian optimization is known to be a method for finding the global optimum of the *black-box* optimization problem:

$$\theta^* = \arg\min_{\theta\in\mathcal{K}} G(\theta) \tag{4}$$

where $\theta \in \mathbb{R}^n$ is the input, $\mathcal{K} \subset \mathbb{R}^n$ is the parameter space of the input, and $G : \mathcal{K} \to \mathbb{R}$ is the unknown function to be minimized. Roughly speaking, the Bayesian optimization iteratively selects a new input $\theta \in \mathcal{K}$ and measures the corresponding output $G(\theta)$ to collect the training data, and estimate $G$ by the GP regression. The basic algorithm of the Bayesian optimization is shown in Algorithm 1. In the algorithm, we let $N_{\text{init}}$ and $N_{\max}$ be the number of iterations for the initialization and for the implementation of the Bayesian optimization, respectively. The algorithm starts with the *Initialization phase*, where the parameter $\theta$ is randomly picked up to evaluate and collect the training data to estimate $G(\cdot)$ by the GP regression (line3–line6). In (5), $\mu_0(\theta)$ and $\sigma_0(\theta)$ are the mean and the variance for the GP model of $G(\cdot)$ after the initialization phase, respectively. Then, the main procedure of the Bayesian optimization is given, where, for each $j$th round, we implement the three steps (Step 1, 2, 3 in Algorithm 1). First, an input to evaluate (denoted as $\theta_j$) is chosen by maximizing the so-called *acquisition function* $\alpha(\cdot; \mathcal{D})$ (Step 1). A typical example for this choice is the *expected improvement* (EI) (Frazier, 0000):

$$\alpha(\theta; \mathcal{D}) = \mathbb{E}\left[\max\left(0, G_{\min} - G(\theta)\right)\right], \tag{9}$$

where $G_{\min}$ is the minimum value of $G$ among the observed measurements that are stacked in $\mathcal{D}$, and $\mathbb{E}[\cdot]$ denotes the expectation over the GP posterior of the objective function based on the current training data. That is, using (9), we select $\theta$ such that the corresponding output provides the best improvement with respect to the current minimum one. The analytical computation of (9) can be indeed derived, see, e.g., Frazier (0000) for details. Then, the corresponding value $y_j = G(\theta_j)$ is measured through the experiment, and update the training data $\mathcal{D}$ (Step 2). Finally, based on the updated training data, it estimates $G(\cdot)$ by the GP regression (Step 3). In (8), $\mu_j(\theta)$ and $\sigma_j(\theta)$ are the mean and the variance for the GP model of $G(\cdot)$ obtained at the $j$th round, respectively.

**Algorithm 1:** Basic algorithm of the Bayesian optimization.

**Input :** k (kernel function for the GP model of $G$);
**Output:** $\theta^*$ (optimal parameter as the solution to (4));

1 $\mathcal{D} \leftarrow \{\}$ (initialize the training data);
2 *(Initialization phase):*
3 **for** $j = 1 : N_{\text{init}}$ **do**
4 Select $\theta \in \mathcal{K}$ randomly and provide an experiment to measure $y = G(\theta)$. Then, update the training data as $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\theta, y)\}$.
5 **end**
6 Using $\mathcal{D}$, estimate $G(\theta)$, $\theta \in \mathcal{K}$ by the GP regression:

$$G(\theta) \sim \mathcal{N}(\mu_0(\theta), \sigma_0^2(\theta)); \tag{5}$$

7 *(Implementation of the Bayesian optimization):*
8 **for** $j = 1 : N_{\max}$ **do**
9 *(Step 1):* Select $\theta_j \in \mathcal{K}$ by maximizing the acquisition function $\alpha$, i.e.,

$$\theta_j \leftarrow \arg\max_{\theta\in\mathcal{K}} \alpha(\theta; \mathcal{D}); \tag{6}$$

*(Step 2):* Provide an experiment to measure $y_j = G(\theta_j)$. Then, update the training data as

$$\mathcal{D} \leftarrow \mathcal{D} \cup \{(\theta_j, y_j)\}; \tag{7}$$

*(Step 3):* Using $\mathcal{D}$, estimate $G(\theta)$, $\theta \in \mathcal{K}$ by the GP regression:

$$G(\theta) \sim \mathcal{N}(\mu_j(\theta), \sigma_j^2(\theta)); \tag{8}$$

10 **end**
11 Return the input evaluated with the smallest value of $G$ among the training data, or return the input that minimizes the mean of the GP model for $G$.

## 3. Problem formulation

### 3.1. Dynamics

Let us consider the following nonlinear continuous-time systems:

$$\dot{x}(t) = f(x(t), u(t)), \ \ x(0) = \bar{x}, \tag{10}$$

for all $t \in \mathbb{R}_{\geq 0}$, where $x \in \mathbb{R}^{n_x}$ is the state, $u \in \mathbb{R}^{n_u}$ is the control input, $\bar{x}$ is the initial state, and $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is the map representing the transition dynamics, which is *unknown* apriori. Although the transition dynamics is unknown, it is assumed that the equilibrium point is known; without loss of generality, we assume that the equilibrium point is the origin, i.e., $0 = f(0, 0)$. Additionally, the system (10) can either be open-loop stable or unstable around this equilibrium point. The control goal of this paper is to asymptotically stabilize the system towards the origin, i.e., $x(t) \to 0$, $t \to \infty$.

### 3.2. Periodic event-triggered control

In this paper, we consider an event-triggered control system as illustrated in Fig. 1. As shown in the figure, the event-triggered mechanism (ETM) is equipped with the sensor and determines the communication time steps to transmit state measurements to the controller over the communication network. In particular, we employ a *periodic* event-triggered control (Heemels, Donkers, & Teel, 2013), in which the sensor *periodically* measures the state information and the ETM evaluates the event-triggered condition to

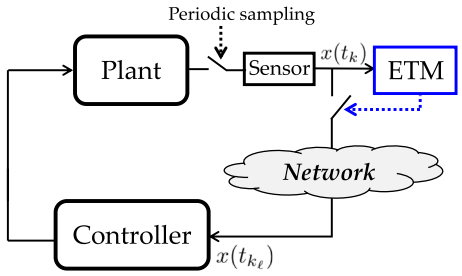



**Fig. 1.** Periodic event-triggered control system considered in this paper.

determine the communication time steps. In contrast to the conventional event-triggered control in which the event-triggered condition has to be evaluated continuously, the periodic event-triggered control requires the evaluation of the event-triggered condition only periodically. As stated in Heemels et al. (2013), this leads to several benefits, such as guaranteeing the positive minimum inter-event times, the ability to incorporate the ETM in the digital devices, and so on.

To illustrate the role of the ETM in more details, let

$$t_k = k\underline{h} \in \mathbb{R}_{\geq 0}, \ \ k \in \mathbb{N}_{\geq 0} \tag{11}$$

be the *sampling time steps* when the sensor measures the state information. Here, $\underline{h} > 0$ is a given sampling period that is chosen as the minimum time duration that the digital sensing device is able to measure the state information. Using the sampled states, the ETM determines whether they should be transmitted to the controller over the communication network. Specifically, for each $t_k$, $k \in \mathbb{N}_{\geq 0}$, the ETM evaluates the following event-triggered condition:

$$g(x(t_k), x(t_\ell); \theta_{ev}) < 0, \tag{12}$$

where $t_\ell$ represents the latest communication time before $t_k$, and $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ is a given function that characterizes the event-triggered condition. As shown in (12), $g$ is characterized by a parameter $\theta_{ev} \in \mathbb{R}^{n_{ev}}_{\geq 0}$, which will be designed later in this paper ($n_{ev}$ denotes the dimension of $\theta_{ev}$). In what follows, $\theta_{ev}$ is called an *event-triggered parameter*. A well-known example of the function $g$ is the static event-triggered condition:

$$g(x(t_k), x(t_\ell); \lambda) = \|x(t_k) - x(t_\ell)\| - \lambda \|x(t_k)\|, \tag{13}$$

where $\lambda \in \mathbb{R}_{\geq 0}$ is the (one-dimensional) event-triggered parameter to be designed, see, e.g., Heemels et al. (2012). Moreover, if the dynamic event-triggered controller is given, the corresponding event-triggered parameter will be multi-dimensional, i.e., $n_{ev} > 1$; see, e.g., Girard (2014) as well as the simulation example given later in this paper (Section 6). It is assumed that $\theta_{ev} \in \mathcal{K}_{ev} \subset \mathbb{R}^{n_{ev}}_{\geq 0}$, where $\mathcal{K}_{ev}$ is a given compact set containing the origin in the interior, i.e., $0 \in \mathcal{K}_{ev}$. Using the event-triggered condition (12), the ETM decides to transmit $x(t_k)$ *only when* (12) is violated. Let

$$t_{k_\ell} \in \mathbb{R}_{\geq 0}, \ \ \ell \in \mathbb{N}_{\geq 0}, \tag{14}$$

with $k_0 = 0$, $k_\ell < k_{\ell+1}$, $\forall \ell \in \mathbb{N}_{\geq 0}$ be the *communication time steps* when the ETM decides to transmit the state information to the controller based on the event-triggered condition (12). Then, the communication time steps can be recursively given as follows: $t_{k_0} = t_0 = 0$ ($k_0 = 0$) and

$$t_{k_{\ell+1}} = \inf\{t_{k_m} \in \mathbb{N}_{\geq 0} : \ k_m > k_\ell \wedge g(x(t_{k_m}), x(t_{k_\ell}); \theta_{ev}) \geq 0\}, \tag{15}$$

for all $\ell \in \mathbb{N}_{\geq 0}$.

Regarding the above event-triggered strategy, it is assumed that the following holds:

$$\theta_{ev} = 0 \implies t_{k_{\ell+1}} = t_{k_\ell} + \underline{h}, \tag{16}$$

for all $\ell \in \mathbb{N}_{\geq 0}$. In other words, setting $\theta_{ev} = 0$ implies that the communication is enforced at *every sampling time step*, i.e., $k_{\ell+1} = k_\ell + 1$, $\forall \ell \in \mathbb{N}_{\geq 0}$. Hence, setting $\theta_{ev} = 0$ implies that the *time-triggered* (periodic) controller is implemented with the sampling period $\underline{h}$. For example, if $g$ is given by (13), the condition (16) indeed holds since $g(x(t_{k_{\ell+1}}), x(t_{k_\ell}); \lambda) = \|x(t_{k_{\ell+1}}) - x(t_{k_\ell})\| \geq 0$ with $\lambda = 0$ for all $\ell \in \mathbb{N}_{\geq 0}$, which implies that the event-triggered condition (12) is violated at every sampling time step $t_k$, $k \in \mathbb{N}_{\geq 0}$.

The control input is updated only when it receives the state information from the ETM. To illustrate this, let $\hat{x}(t)$ be given by

$$\hat{x}(t) = x(t_{k_\ell}), \ \ \forall t \in [t_{k_\ell}, t_{k_{\ell+1}}). \tag{17}$$

That is, $\hat{x}(t)$ indicates the latest state information that is received from the ETM. Then, the control input is updated as follows:

$$u(t) = \pi(\hat{x}(t); \theta_c), \tag{18}$$

where $\pi(\cdot; \theta_c)$ is a given control law and is parameterized by $\theta_c \in \mathbb{R}^{n_c}$, which will be designed later in this paper ($n_c$ denotes the dimension of $\theta_c$). In what follows, the parameter $\theta_c$ is called the *control parameter*. For simplicity, it is assumed that $\theta_c \in \mathcal{K}_c \subset \mathbb{R}^{n_c}$, where $\mathcal{K}_c$ is a given compact set.

### 3.3. Cost function to be evaluated

Let us now define the cost function to be evaluated for designing both the event-triggered parameter $\theta_{ev}$ and the control parameter $\theta_c$. To this end, let $x(t_k)$, $u(t_k)$, $k \in \{0, \dots, H-1\}$ with $x(0) = \bar{x}$ for a given $H \in \mathbb{N}_{>0}$ be the state and the control trajectories at the sampling time steps generated according to (10), respectively. For simplicity, we let

$$\mathbf{x}(H) = \{x(t_k), \ \ k \in \{0, \dots, H-1\}\} \tag{19}$$

$$\mathbf{u}(H) = \{u(t_k), \ \ k \in \{0, \dots, H-1\}\}. \tag{20}$$

Then, we denote the cost function by

$$J(\mathbf{x}(H), \mathbf{u}(H)) \geq 0. \tag{21}$$

Here, different types of the cost functions can be considered. For example, one can consider the following quadratic type cost function:

$$J(\mathbf{x}(H), \mathbf{u}(H)) = \sum_{k=0}^{H-1} \left\{ x(t_k)^\mathsf{T} Q x(t_k) + u(t_k)^\mathsf{T} R u(t_k) \right\}, \tag{22}$$

where $Q$ and $R$ are the positive definite matrices. Moreover, one can also consider the convergence time towards the neighborhood of the origin:

$$J(\mathbf{x}(H), \mathbf{u}(H)) = \min_k \left\{ t_k : \forall k' \in [k, H-1], \|x(t_{k'})\|_\infty \leq \eta \right\},$$

where $\eta > 0$ is a given threshold to characterize the convergence, and $\|\cdot\|_\infty$ denotes the $\infty$-norm.

Note that the state and the control input trajectories are determined through the selection of both the control parameter $\theta_c$ and the event-triggered parameter $\theta_{ev}$. Hence, we can define a mapping (denoted by $E$) from each pair $(\theta_c, \theta_{ev}) \in \mathcal{K}_c \times \mathcal{K}_{ev}$ onto the corresponding state and input trajectories as follows:

$$E(\theta_c, \theta_{ev}) = \{\mathbf{x}(H; \theta_c, \theta_{ev}), \mathbf{u}(H; \theta_c, \theta_{ev})\}, \tag{23}$$

where $\mathbf{x}(H; \theta_c, \theta_{ev})$ and $\mathbf{u}(H; \theta_c, \theta_{ev})$ are the state and the input trajectories by applying the (periodic) event-triggered controller with the control and the event-triggered parameters being given by $\theta_c$ and $\theta_{ev}$, respectively. Then, we can define a mapping from each pair $(\theta_c, \theta_{ev}) \in \mathcal{K}_c \times \mathcal{K}_{ev}$ onto the corresponding cost as follows:

$$G(\theta_c, \theta_{ev}) = (J \circ E)(\theta_c, \theta_{ev}). \tag{24}$$

That is, $G(\theta_c, \theta_{ev})$ represents the cost by applying the event-triggered controller with the control and the event-triggered parameters being given by $\theta_c$ and $\theta_{ev}$, respectively. Note that, although the shape of $J$ is known, the shape of $G$ is *unknown*, since the dynamics of the plant is unknown apriori (i.e., the mapping $E$ is unknown). Note also that, $G(\theta_c, 0)$ represents a cost under the time-triggered controller with the control parameter being given by $\theta_c$, since setting $\theta_{ev} = 0$ implies that $t_{k_{\ell+1}} = t_{k_\ell} + \underline{h}$ for all $\ell \in \mathbb{N}_{\geq 0}$ (see (16)).

In addition to the above, let $\mathcal{I}(\theta_c, \theta_{ev}) \in \mathbb{N}_{>0}$ be the total number of communication time steps until $t_{H-1}(= (H-1)\underline{h})$ by applying the event-triggered controller with the control and the event-triggered parameters given by $(\theta_c, \theta_{ev}) \in \mathcal{K}_c \times \mathcal{K}_{ev}$ i.e.,

$$\mathcal{I}(\theta_c, \theta_{ev}) = \max\{\ell \in \mathbb{N}_{\geq 0} : t_{k_\ell} \leq t_{H-1}\}. \tag{25}$$

Here, we state that the number of communication time steps depends on the choice of $\theta_c$ and $\theta_{ev}$. This is due to the fact that the communication time steps are determined through the state trajectory as in (15), and, as previously mentioned, the state trajectory is determined through the choice of $(\theta_c, \theta_{ev})$. Note that, since setting $\theta_{ev} = 0$ implies the implementation of the time-triggered controller (again, see (16)), we have

$$\mathcal{I}(\theta_c, \theta_{ev}) \leq \mathcal{I}(\theta_c, 0) = H \tag{26}$$

for all $(\theta_c, \theta_{ev}) \in \mathcal{K}_c \times \mathcal{K}_{ev}$ with $\theta_{ev} \neq 0$.

### 3.4. Problem formulation

In this paper, we design the event-triggered controller by following the *emulation-based approach* (Heemels & Donkers, 2013; Heemels et al., 2012). Mainly, the emulation-based approach consists of the following two steps. First, we find the optimal control parameter $\theta_c$ under the *time-triggered* controller (i.e., $\theta_{ev} = 0$), such that the corresponding cost function defined in (21) is minimized. More formally, we aim at solving the following problem:

**Problem 1.** *Find the control parameter $\theta_c^* \in \mathcal{K}_c$, such that*

$$\theta_c^* = \arg\min_{\theta_c \in \mathcal{K}_c} G(\theta_c, 0). \quad \square \tag{27}$$

Addressing Problem 1 has practical significance, as it enables us to explore the underlying performance of time-triggered controllers, which are of considerable interest in various applications. Furthermore, by optimizing performance within a time-triggered framework, we can closely replicate this performance while devising the event-triggered control systems. Specifically, after solving Problem 1 to identify the optimal parameter $\theta_c^*$, it becomes feasible to design an event-triggered parameter $\theta_{ev}$, such that the number of communication time steps can be minimized while the corresponding control performance is close enough to the time-triggered case $G(\theta_c^*, 0)$. For an in-depth explanation of this methodology, refer to Problem 2 as detailed subsequently. Note that Problem 1 is a *black-box optimization problem*, since, as previously mentioned, the shape of the function $G$ is unknown. As will be described in the next section, we provide the Bayesian optimization algorithm to solve this optimization problem, such that the amount of the network utilization can be taken into account.

Now, let $\widehat{\theta}_c^*$ be an approximated solution to Problem 1. In the second step of the emulation-based approach, we aim at designing the event-triggered parameter $\theta_{ev}$. Specifically, the event-triggered parameter is designed to minimize the number of communication time steps while satisfying a certain constraint on the control performance with respect to the time-triggered case. More formally, we aim at solving the following problem:

**Problem 2.** *Let $\widehat{\theta}_c^*$ be an approximated solution to Problem 1. Then, find the event-triggered parameter $\theta_{ev}^* \in \mathcal{K}_{ev}$, such that:*

$$\theta_{ev}^* = \arg\min_{\theta_{ev} \in \mathcal{K}_{ev}} \mathcal{I}(\widehat{\theta}_c^*, \theta_{ev}) \tag{28}$$

$$\text{s.t. } C(G(\widehat{\theta}_c^*, \theta_{ev}), G(\widehat{\theta}_c^*, 0)) \leq 0, \tag{29}$$

*where $C : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a given function that represents a constraint on $G(\widehat{\theta}_c^*, \theta_{ev})$.* $\square$

For example, one can define the function $C$ as:

$$C(G(\widehat{\theta}_c^*, \theta_{ev}), G(\widehat{\theta}_c^*, 0)) = G(\widehat{\theta}_c^*, \theta_{ev}) - G(\widehat{\theta}_c^*, 0) - \epsilon. \tag{30}$$

with $\epsilon > 0$ being a given threshold. Then, Problem 2 implies that, we aim at finding $\theta_{ev}$ such that the number of communication time steps is minimized while the corresponding cost is close enough to $G(\widehat{\theta}_c^*, 0)$ with respect to the threshold $\epsilon$. Another example may include the relative error-based constraint function:

$$C(G(\widehat{\theta}_c^*, \theta_{ev}), G(\widehat{\theta}_c^*, 0)) = \frac{G(\widehat{\theta}_c^*, \theta_{ev}) - G(\widehat{\theta}_c^*, 0)}{G(\widehat{\theta}_c^*, 0)} - \epsilon,$$

where, as above, $\epsilon \in (0, 1]$ is a given threshold.

## 4. Solution to Problem 1 : designing the control parameter

In this section, we provide an approach to designing the control parameter $\theta_c$ as the solution to Problem 1. In particular, we provide a solution to Problem 1 by employing the Bayesian optimization, whose basic concept has been described in Section 2.2. Before stating our main approach, we first provide a *naive* solution to Problem 1, which is a straightforward application of Algorithm 1 but it exhibits a critical issue regarding the expense of the network utilization. With a slight abuse of notation, here we let $G(\theta_c) = G(\theta_c, 0)$. Since we want to optimize $G(\theta_c)$, $\theta_c \in \mathcal{K}_c$, the simplest way is to implement the time-triggered controller and estimate $G(\theta_c)$, $\theta_c \in \mathcal{K}_c$ by the GP regression. That is, following line 7–line 10 in Algorithm 1, we could execute the following three steps for each round of the Bayesian optimization:

(Step 1) Select $\theta_c \in \mathcal{K}_c$ by maximizing a certain acquisition function;
(Step 2) Implement the time-triggered controller (with the sampling period $\underline{h}$) and measure $y = G(\theta_c)$. Based on this measurement, update the training data as $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\theta_c, y)\}$;
(Step 3) Estimate $G(\theta_c)$, $\theta_c \in \mathcal{K}_c$ by the Gaussian Process regression.

In the above, Step 1, 2, 3 are corresponding to Step 1, 2, 3 in Algorithm 1, respectively. However, the main drawback of the above approach is that, in order to measure the cost value in Step 3, communication must be given at *every sampling time step*. In other words, it must require the maximal number of communication time steps in order to measure the cost value for each round of the Bayesian optimization. This fact is clearly not preferable, since it expends a significant amount of network utilization (e.g., energy consumption of the battery powered devices), aiming at finding the solution to Problem 1.

### 4.1. Solution approach

Motivated by the issue presented in the previous section, in this paper we provide an alternative approach to the naive solution, so that the network utilization can be explicitly taken into account while solving Problem 1. First, suppose that, in Step 1 for each round of the Bayesian optimization, we have the option to select not only the control parameter $\theta_c \in \mathcal{K}_c$, but also the event-triggered parameter $\theta_{ev} \in \mathcal{K}_{ev}$. In other words, we allow the implementation of the event-triggered controller with $\theta_{ev} \neq 0$ in

Step 2 and measure the corresponding cost $G(\theta_c, \theta_{ev})$ (despite the fact that we would like to optimize $G(\theta_c, 0)$). From (26), implementing the event-triggered controller with $\theta_{ev} \neq 0$ in Step 2 has the potential to reduce the number of communication time steps in contrast to the time-triggered case $\theta_{ev} = 0$. Then, in Step 3, the function $G(\theta_c, \theta_{ev})$, $(\theta_c, \theta_{ev}) \in \mathcal{K}_c \times \mathcal{K}_{ev}$ is estimated based on the GP regression, which is then utilized to estimate the value of $G(\theta_c, 0)$, which we want to optimize. Iterating the above procedure has the potential to reduce the number of communication time steps, since we implement the event-triggered controller (instead of the time-triggered controller) for each round of the Bayesian optimization.

A crucial problem of the above strategy is, however, how to select suitable parameters to evaluate in Step 1, or in other words, how to define the acquisition function, so that the minimum of $G(\theta_c, 0)$ can be efficiently found. Intuitively, as $\theta_{ev}$ is selected far from 0 in Step 1, we have the potential to achieve a smaller number of communication time steps in Step 2. For example, if the event-triggered condition is based on (13), the number of the communication time steps will be smaller as $\lambda$ (i.e., $\theta_{ev}$) is selected larger, since we increase the threshold $\lambda$ for the event-triggered condition. On the other hand, the estimation accuracy of $G(\theta_c, 0)$ that we want to optimize will be worse as $\lambda$ is selected larger, since we measure the cost value with $\lambda$ being far from 0. Hence, when selecting $\theta_{ev}$ in Step 1, we need to consider the trade-off between the communication load and the estimation accuracy for $G(\theta_c, 0)$.

To make our solution approach more specific, we proceed by defining the map

$$P : \mathcal{K}_{ev} \to \mathbb{R}_+, \tag{31}$$

with the property that $0 \in \arg\max_{\theta_{ev} \in \mathcal{K}_{ev}} P(\theta_{ev})$. In what follows, the map $P$ is called a *penalty map*. That is, we assign the largest penalty for $\theta_{ev} = 0$, and, otherwise, we assign the equal or smaller penalties. For example, if the event-triggered condition is based on (13) with $\lambda\ (= \theta_{ev}) \in \mathcal{K}_{ev} = [0, 1] \subset \mathbb{R}_{\geq 0}$, one possible choice of the penalty map is

$$P(\lambda) = -\gamma_0 \lambda + \gamma_1, \tag{32}$$

for all $\lambda \in [0, 1]$, where $\gamma_0, \gamma_1 > 0$ are the tuning parameters such that $\gamma_1 > \gamma_0$ so as to fulfill $P(1) > 0$. (32) implies that we assign a smaller penalty as $\lambda$ is selected larger, since it has the potential to reduce the number of communication time steps. As will be seen below, this penalty map is incorporated in the acquisition function as a role to regulate the trade-off between the number of communication time steps and the estimation accuracy of $G(\theta_c, 0)$.

Based on the above notations and definitions, let us now present the main algorithm to solve Problem 1. The proposed algorithm is shown in Algorithm 2. In the algorithm, we let $N_{\text{init}}$ and $N_{\max}$ be the number of iterations for the initialization and for the implementation of the Bayesian optimization, respectively. Moreover, we denote by $\mu_j(\theta_c, \theta_{ev})$, $\sigma_j(\theta_c, \theta_{ev})$, $j \in \mathbb{N}_{1:N_{\max}}$ the GP mean and the variance of $G(\theta_c, \theta_{ev})$ obtained in Step 3 after the $j$th round of the Bayesian optimization, respectively.

As previously mentioned, the proposed approach is essentially different from the naive approach given in the previous subsection, in the sense that we select not only the control parameter $\theta_c$ but also the event-triggered parameter $\theta_{ev}$ in Step 1. In other words, we implement the event-triggered controller in Step 2 and measure the corresponding cost $G(\theta_c, \theta_{ev})$. Thus, the function $G(\theta_c, \theta_{ev})$, $(\theta_c, \theta_{ev}) \in \mathcal{K}_c \times \mathcal{K}_{ev}$ is estimated by the GP regression as shown in Step 3. In order to take the trade-off between the number of communication time steps and the estimation accuracy of $G(\theta_c, 0)$, we make use of the following acquisition function

**Algorithm 2:** Solution approach to Problem 1.

**Input :** k (kernel function for the GP model of $G$);
$P$ (penalty map);
**Output:** $\theta_c^*$ (optimal control parameter as the solution to Problem 1);

$\mathcal{D} \leftarrow \{\}$ (initialize the training data);

*(Initialization phase):*

**for** $j = 1 : N_{\text{init}}$ **do**

  Select $(\theta_c, \theta_{ev}) \in \mathcal{K}_c \times \mathcal{K}_{ev}$ randomly and provide an experiment to measure $y = G(\theta_c, \theta_{ev})$. Then, update the training data as $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\theta_c, \theta_{ev}), y\}$.

**end**

Using $\mathcal{D}$, estimate $G(\theta_c, \theta_{ev})$, $(\theta_c, \theta_{ev}) \in \mathcal{K}_c \times \mathcal{K}_{ev}$ by the GP regression:

$$G(\theta_c, \theta_{ev}) \sim \mathcal{N}(\mu_0(\theta_c, \theta_{ev}), \sigma_0^2(\theta_c, \theta_{ev})) \tag{33}$$

*(Implementation of the Bayesian optimization):*

**for** $j = 1 : N_{\max}$ **do**

  *(Step 1):* Select $(\theta_c, \theta_{ev}) \in \mathcal{K}_c \times \mathcal{K}_{ev}$ by maximizing the acquisition function $\alpha$, i.e.,

$$(\theta_c, \theta_{ev}) \leftarrow \underset{(\theta_c', \theta_{ev}') \in \mathcal{K}_c \times \mathcal{K}_{ev}}{\arg\max}\ \alpha_j(\theta_c', \theta_{ev}'; \mathcal{D}),$$

  where $\alpha_j$ is defined in (37).

  *(Step 2):* Provide an experiment to measure $y = G(\theta_c, \theta_{ev})$. Then, update the training data as

$$\mathcal{D} \leftarrow \mathcal{D} \cup \{(\theta_c, \theta_{ev}), y\}; \tag{34}$$

  *(Step 3):* Using the new $\mathcal{D}$, estimate $G(\theta_c, \theta_{ev})$, $(\theta_c, \theta_{ev}) \in \mathcal{K}_c \times \mathcal{K}_{ev}$ by the GP regression:

$$G(\theta_c, \theta_{ev}) \sim \mathcal{N}(\mu_j(\theta_c, \theta_{ev}), \sigma_j^2(\theta_c, \theta_{ev})) \tag{35}$$

**end**

Return the control parameter $\theta_c$ that minimizes the mean of the objective function that we want to minimize, i.e.,

$$\widehat{\theta}_c^* \leftarrow \underset{\theta_c \in \mathcal{K}_c}{\arg\min}\ \mu_{N_{\max}}(\theta_c, 0). \tag{36}$$

Then, provide an experiment to measure $\widehat{G}^* = G(\widehat{\theta}_c^*, 0)$.

in Step 1:

$$\alpha_j(\theta_c', \theta_{ev}'; \mathcal{D}) = \frac{\mu^* - \mathbb{E}\left[\mu_+^*(\theta_c', \theta_{ev}')\right]}{P(\theta_{ev}')}, \tag{37}$$

where

$$\mu^* = \min_{\theta_c \in \mathcal{K}_c} \mu_{j-1}(\theta_c, 0) \tag{38}$$

$$\mu_+^*(\theta_c', \theta_{ev}') = \min_{\theta_c \in \mathcal{K}_c} \mu_j(\theta_c, 0;\ \theta_c', \theta_{ev}'). \tag{39}$$

In (38), $\mu^*$ represents the minimum GP mean for the objective function to be optimized (i.e., $G(\theta_c, 0)$) based on the current training data obtained after the previous round. To describe the meaning of $\mu_j(\theta_c, 0; \theta_c', \theta_{ev}')$ in (39), suppose that the control and the event-triggered parameters are selected as $\theta_c'$ and $\theta_{ev}'$ in Step 1 at the $j$th round, respectively, and the corresponding cost are measured in Step 2 to obtain the new output $y' = G(\theta_c', \theta_{ev}')$. Then, $\mu_j(\theta_c, 0; \theta_c', \theta_{ev}')$ denotes the GP mean with the input $(\theta_c, 0)$, based on the new training data $\mathcal{D}' = \mathcal{D} \cup \{(\theta_c', \theta_{ev}'), y'\}$. Hence, $\mu_+^*(\theta_c', \theta_{ev}')$ in (39) represents the minimum GP mean for the objective function to be optimized based on the new training data $\mathcal{D}'$, with the assumption that the control and the event-triggered parameters are selected as $\theta_c'$ and $\theta_{ev}'$ at Step 1, respectively. Note that, while $\mu^*$ is known, $\mu_+^*$ is *unknown*, since we have not yet measured the corresponding cost value $y' = G(\theta_c', \theta_{ev}')$. Hence, we

take the expectation as in (37) by assuming that $G$ follows the current GP posterior: $G(\theta_c, \theta_{ev}) \sim \mathcal{N}(\mu_{j-1}(\theta_c, \theta_{ev}), \sigma^2_{j-1}(\theta_c, \theta_{ev}))$. Therefore, the term $\mu^* - \mathbb{E}[\mu^*_+(\theta'_c, \theta'_{ev})]$ in (37) indicates the *expected improvement* of the minimum GP mean of the objective function to be optimized. Moreover, as shown in (37) we divide the improvement by the corresponding penalty $P(\theta'_{ev})$. This indeed has a role to avoid the trivial case when $\theta_{ev} = 0$ is always selected in Step 1; by dividing with $P(\theta'_{ev})$, we can increase the possibility to select the event-triggered parameter with $\theta_{ev} \neq 0$ so as to reduce the number of communication time steps. Once the pair $(\theta_c, \theta_{ev})$ is selected, the remaining steps are the same as Algorithm 1; provide the experiment to measure the cost value (Step 2), and update the GP model (Step 3).

Several remarks on the computation of $\mathbb{E}[\mu^*_+(\theta'_c, \theta'_{ev})]$ are given as follows. As shown in (37), for each $j$th round we need to compute the acquisition function that involves the expectation of the minimum value of $\mu_j(\theta_c, 0; \theta'_c, \theta'_{ev})$ over the GP posterior $G(\theta_c, \theta_{ev}) \sim \mathcal{N}(\mu_{j-1}(\theta_c, \theta_{ev}), \sigma^2_{j-1}(\theta_c, \theta_{ev}))$. To compute this expectation in a tractable way, we first limit the control and the event-triggered parameter spaces to those that consist of only *finite* number of points in $\mathcal{K}_c$ and $\mathcal{K}_{ev}$, respectively, which are denoted as $\widetilde{\mathcal{K}}_c \subset \mathcal{K}_c$, $\widetilde{\mathcal{K}}_{ev} \subset \mathcal{K}_{ev}$. Then, we have

$$\begin{aligned}\mu^*_+(\theta'_c, \theta'_{ev}) &= \min_{\theta_c \in \widetilde{\mathcal{K}}_c} \mu_j(\theta_c, 0;\ \theta'_c, \theta'_{ev}) \\ &= \min_{\theta_c \in \widetilde{\mathcal{K}}_c} \left\{\mu_{j-1}(\theta_c, 0) + \widetilde{\sigma}_j(\theta_c, 0; \theta'_c, \theta'_{ev})Z\right\},\end{aligned}$$

where $Z$ is the Gaussian distributed random variable with zero mean and the variance $\sigma_{j-1}(\theta'_c, \theta'_{ev})$, and

$$\begin{aligned}\widetilde{\sigma}_j(\theta_c, 0; \theta'_c, \theta'_{ev}) =& \mathsf{k}_{j-1}((\theta_c, 0), (\theta'_c, \theta'_{ev}))\{\mathsf{k}_{j-1}((\theta'_c, \theta'_{ev}), \\ & (\theta'_c, \theta'_{ev}))\}^{-1},\end{aligned}$$

where $\mathsf{k}_{j-1}$ is the kernel function corresponding to the GP model of $G$ after the $j-1$th round. Therefore, the expectation of $\mu^*_+$ can be approximated based on the collection of the sampled points: $\mu^{*(1)}_+, \mu^{*(2)}_+, \ldots, \mu^{*(S)}_+$, where $\mu^{*(s)}_+ = \min_{\theta_c \in \widetilde{\mathcal{K}}_c} \left\{\mu_{j-1}(\theta_c, 0) + \widetilde{\sigma}_j(\theta_c, 0; \theta'_c, \theta'_{ev})z_s\right\}$ with $z_s \in \mathbb{R}$, $s \in \mathbb{N}_{1:S}$ being the samples generated from the Gaussian distribution with zero mean and the variance $\sigma_{j-1}(\theta'_c, \theta'_{ev})$, and $S$ denotes the number of samples to approximate the expectation.

### 4.2. Stability analysis

Let $d \geq 0$ be a discretization level to approximate the expectation $\mu^*_+$ computed by $d = \max_{\theta_c \in \widetilde{\mathcal{K}}_c} \min_{\theta'_c \in \mathcal{K}_c} \|\theta_c - \theta'_c\|$, and let $\widehat{\theta}^*_{c,j}$ be the optimal solution suggested by Algorithm 2 at the $j$th iteration, $\widehat{\theta}^*_{c,j} = \arg\min_{\theta_c \in \mathcal{K}_c} \mu_j(\theta_c, 0)$. To analyze stability, we need the following assumption.

**Assumption 1.** The following assumptions are made: (i) the kernel function is a squared exponential (1); (ii) there exists an $\delta > 0$ such that, for all $\theta_c \in \Theta = \{\theta'_c \in \mathcal{K}_c : G(\theta'_c, 0) \leq G(\theta^*_c, 0) + \delta\}$, the origin is asymptotically stable by applying the time triggered controller with $\theta_c$. □

The assumption (ii) implies that asymptotic stability of the origin is guaranteed by applying the time-triggered controller when the control parameter is chosen within the neighborhood of the optimum $\theta^*_c$.

Then, we can establish following result that the control parameter obtained by Algorithm 2 asymptotically stabilizes the system as $N_{\max} \to \infty$.

**Proposition 1.** *Let Assumption 1 hold. For every $p \in [0, 1)$, there exists a discretization level $d$ such that, with probability at least $1-p$, $\lim_{j\to\infty} \widehat{\theta}^*_{c,j} \in \Theta$. In addition, $\lim_{j\to\infty} \widehat{\theta}^*_{c,j} = \theta^*_c$ as $d \to 0$.* □

Proposition 1 states that if $\widetilde{\mathcal{K}}_c$ is dense enough to precisely approximate $\mathcal{K}_c$, the resulting control parameter leads to asymptotic stability of the origin. In addition, optimality is guaranteed if $\widetilde{\mathcal{K}}_c = \mathcal{K}_c$ and as the iteration goes to infinity.

**Proof.** First, for every $p \in [0, 1)$ and $d \geq 0$, there exists $K_p > 0$ such that $\lim_{j\to\infty} G(\widehat{\theta}^*_{c,j}, 0) \leq G(\theta^*_c, 0) + K_p d$ with probability at least $1 - p$ (for details, see Theorem 1 of Poloczek, Wang, and Frazier (2017) or Theorem 4 of Frazier, Powell, and Dayanik (2009) for this analogous result). This implies that, if $d$ is chosen small enough to fulfill $d \leq \delta/K_p$, we have $\lim_{j\to\infty} G(\widehat{\theta}^*_{c,j}, 0) \leq G(\theta^*_c, 0) + \delta$ and thus $\lim_{j\to\infty} \widehat{\theta}^*_{c,j} \in \Theta$. Moreover, as $d \to 0$ we have $\lim_{j\to\infty} G(\widehat{\theta}^*_{c,j}, 0) \leq G(\theta^*_c, 0)$ which implies $\lim_{j\to\infty} \widehat{\theta}^*_{c,j} = \theta^*_c$ as $d \to 0$. □

## 5. Solution to Problem 2: designing the event-triggered parameter

Let $\widehat{\theta}^*_c \in \mathcal{K}_c$ be the optimal control parameter estimated by implementing AlgorithmProblem 2, and let $\widehat{G}^* = G(\widehat{\theta}^*_c, 0)$ be the corresponding (estimated) optimal cost. Based on this solution, we now proceed by designing the event-triggered parameter $\theta_{ev} \in \mathcal{K}_{ev}$ by solving Problem 2. Using $\widehat{\theta}^*_c$ and $\widehat{G}^*$ and following Problem 2, the optimization problem that we seek to solve here is given as follows:

$$\min_{\theta_{ev} \in \mathcal{K}_{ev}} \mathcal{I}(\widehat{\theta}^*_c, \theta_{ev}), \quad \text{s.t. } C(G(\widehat{\theta}^*_c, \theta_{ev}), \widehat{G}^*) \leq 0. \tag{40}$$

Since $G(\widehat{\theta}^*_c, \theta_{ev})$ and $\mathcal{I}(\widehat{\theta}^*_c, \theta_{ev})$ are both functions of only the event-triggered parameter $\theta_{ev}$, we can let, with a slight abuse of notation, $\mathcal{I}(\theta_{ev}) = \mathcal{I}(\widehat{\theta}^*_c, \theta_{ev})$ and $C(\theta_{ev}) = C(G(\widehat{\theta}^*_c, \theta_{ev}), \widehat{G}^*)$. Then, the optimization problem simply becomes

$$\min_{\theta_{ev} \in \mathcal{K}_{ev}} \mathcal{I}(\theta_{ev}), \quad \text{s.t. } C(\theta_{ev}) \leq 0, \tag{41}$$

Note that $\mathcal{I}(\cdot)$ and $C(\cdot)$ are both unknown functions. (41) differs from (4), in the sense that the constraint $C(\theta_{ev}) \leq 0$ should be explicitly taken into account when solving the optimization problem. Hence, to solve (41), we modify the standard Bayesian optimization algorithm (Algorithm 1) in the following ways. First, in the initialization phase and in Step 3 for each round of the implementation, not only the objective function $\mathcal{I}$ but also the constraint function $C$ are estimated based on the training data with the GP regression. Hence, we have

$$\mathcal{I}(\theta_{ev}) \sim \mathcal{N}(\mu_{I,j}(\theta_{ev}), \sigma^2_{I,j}(\theta_{ev})) \tag{42}$$

$$C(\theta_{ev}) \sim \mathcal{N}(\mu_{C,j}(\theta_{ev}), \sigma^2_{C,j}(\theta_{ev})), \tag{43}$$

where $\mu_{I,j}$, $\sigma_{I,j}$ (resp. $\mu_{C,j}, \sigma_{C,j}$) will denote the mean and the variance of the GP model of $\mathcal{I}$ (resp. $C$) based on the training data after the $j$th round of the Bayesian optimization, respectively. We denote by $\mathsf{k}_{\mathcal{I}}(\cdot)$ and $\mathsf{k}_C(\cdot)$ the corresponding kernels. Second, in order to incorporate the constraint when selecting the parameter to evaluate, we introduce the following acquisition function in Step 1:

$$\alpha(\theta_{ev}; \mathcal{D}_1, \mathcal{D}_2) = B(\theta_{ev})\mathbb{E}\left[\max\left(0, \mathcal{I}_{\min} - \mathcal{I}(\theta_{ev})\right)\right], \tag{44}$$

where $\mathcal{D}_1, \mathcal{D}_2$ are the training data for $\mathcal{I}$ and $C$, respectively (defined specifically in Algorithm 3), and $B(\theta_{ev}) \in \{0, 1\}$ is defined as $B(\theta_{ev}) = 1$ if $\mu_{C,j}(\theta_{ev}) + \beta_j \sigma^2_{C,j}(\theta_{ev}) \leq 0$ and $B(\theta_{ev}) = 0$ otherwise, with $\beta_j > 0$ being defined later in this section. Intuitively, we allow the expected improvement to take positive values only if the constraint $C(\theta_{ev}) \leq 0$ is satisfied. Note that the term $\mathbb{E}\left[\max\left(0, \mathcal{I}_{\min} - \mathcal{I}(\theta_{ev})\right)\right]$ follows the standard expected improvement (EI) and can be computed in an explicit form; see Section 2.2 as well as Frazier (0000). Based on the above, we summarize the approach in Algorithm 3 as the solution to (40).

**Algorithm 3:** Solution approach to Problem 2.

**Input :** $\mathsf{k}_1, \mathsf{k}_2$ (kernel functions for the GP model of $\mathcal{I}$ and $C$); $\theta_c$, $G^*$ (optimal control parameter and the corresponding optimal cost by implementing Algorithm 2)

**Output:** $\theta_{ev}^*$ (optimal event-triggered parameter as the solution to Problem 2);

1 $\mathcal{D}_1, \mathcal{D}_2 \leftarrow \{\}$ (initialize the training data);

2 *(Initialization phase):*

3 **for** $j = 1 : N_{\text{init}}$ **do**

4 Select $\theta_{ev} \in \mathcal{K}_{ev}$ randomly and provide an experiment to measure $y_1 = \mathcal{I}(\theta_{ev})$ and $y_2 = C(\theta_{ev})$. Then, update the training data as $\mathcal{D}_1 \leftarrow \mathcal{D}_1 \cup \{(\theta_{ev}, y_1)\}$, $\mathcal{D}_2 \leftarrow \mathcal{D}_2 \cup \{(\theta_{ev}, y_2)\}$.

5 **end**

6 Using $\mathcal{D}_1$, $\mathcal{D}_2$, estimate both $\mathcal{I}(\theta_{ev})$ and $C(\theta_{ev})$, $\theta_{ev} \in \mathcal{K}_{ev}$ using the GP regression:

$$\mathcal{I}(\theta_{ev}) \sim \mathcal{N}(\mu_{I,0}(\theta_{ev}), \sigma_{I,0}^2(\theta_{ev})) \tag{45}$$

$$C(\theta_{ev}) \sim \mathcal{N}(\mu_{C,0}(\theta_{ev}), \sigma_{C,0}^2(\theta_{ev})), \tag{46}$$

*(Implementation of the Bayesian optimization):*

7 **for** $j = 1 : N_{\max}$ **do**

8 *(Step 1):* Select $\theta_{ev,j} \in \mathcal{K}_{ev}$ by maximizing the acquisition function $\alpha$:

$$\theta_{ev,j} \leftarrow \arg\max_{\theta'_{ev} \in \mathcal{K}_{ev}} \alpha(\theta'_{ev}; \mathcal{D}_1, \mathcal{D}_2),$$

where $\alpha$ is defined in (44).

9 *(Step 2):* Provide an experiment to measure $y_{\mathcal{I},j} = \mathcal{I}(\theta_{ev,j})$, $y_{C,j} = C(\theta_{ev,j})$. Then, update the training data as

$$\mathcal{D}_1 \leftarrow \mathcal{D}_1 \cup \{(\theta_{ev,j}, y_{\mathcal{I},j})\}, \quad \mathcal{D}_2 \leftarrow \mathcal{D}_2 \cup \{(\theta_{ev}, y_{C,j})\};$$

*(Step 3):* Using the new $\mathcal{D}_1$, $\mathcal{D}_2$, estimate $\mathcal{I}(\theta_{ev})$ and $C(\theta_{ev})$, $\theta_{ev} \in \mathcal{K}_{ev}$ by the GP regression:

$$\mathcal{I}(\theta_{ev}) \sim \mathcal{N}(\mu_{I,j}(\theta_{ev}), \sigma_{I,j}^2(\theta_{ev})), \tag{47}$$

$$C(\theta_{ev}) \sim \mathcal{N}(\mu_{C,j}(\theta_{ev}), \sigma_{C,j}^2(\theta_{ev})). \tag{48}$$

10 **end**

11 Return the event-triggered parameter $\theta_{ev}^*$ among the training input, such that the corresponding output minimizes $\mathcal{I}$.

**Remark 1.** As shown in Algorithm 3, the implementation of the Bayesian optimization starts with the empty training data for $\mathcal{I}$, i.e., $\mathcal{D}_1 \leftarrow \{\}$ (line 1). However, since we have already implemented the event-triggered controller in Algorithm 2, the number of the communication time steps could be measured during the implementation of Algorithm 2 (in addition to $G$), and make use of them in Algorithm 3 as the initial training data to estimate $\mathcal{I}$. Similarly, the training data for $C$, i.e., $\mathcal{D}_2$, could be also obtained before the implementation of Algorithm 3, using the optimal cost $G^*$ and the training data for $G$ after the implementation of Algorithm 2. In those cases, the initialization phase in Algorithm 3 may not be necessary, as a sufficient number of training data has been already collected before implementing Algorithm 3. □

### 5.1. Stability analysis

Finally, we investigate theoretical analysis of Algorithm 3. Proving the optimality of the constrained Bayesian optimization algorithm considered in this paper is indeed challenging; nevertheless, constraint satisfaction (or, asymptotic stability of the origin) will be guaranteed by applying the event-triggered controller based on the parameter suggested by Algorithm 3. Before providing the analysis, we need to make certain *smoothness* assumptions on $C(\cdot)$.

**Assumption 2.** The following assumptions are made: (i) the function $C(\cdot)$ lies in the reproducing kernel Hilbert space (RKHS) corresponding to the kernel $\mathsf{k}_C$; (ii) $\|C\|_{\mathsf{k}_C} \leq B_C$ for some $B_C > 0$, where $\|\cdot\|_{\mathsf{k}_C}$ denotes the induced norms of the RKHS; (iii) there exists an $\bar{\epsilon} > 0$ such that, for all $\theta_{ev} \in \Theta_{ev} = \{\theta'_{ev} \in \mathcal{K}_{ev} : G(\theta_c^*, \theta'_{ev}) \leq G(\theta_c^*, 0) + \bar{\epsilon}\}$, the origin is asymptotically stable by applying the event-triggered controller with $(\theta_c^*, \theta_{ev})$.

The assumption that $C(\cdot)$ lies in the RKHS with a bounded norm implies that the function is given of the form $C(\theta_{ev}) = \sum_{n=1}^{\infty} \alpha_n \mathsf{k}_C(\theta_{ev}, \bar{\theta}_n)$, where $\bar{\theta}_n \in \mathcal{K}_{ev}$, $n \in \mathbb{N}_{>0}$ are the representer points and $\alpha_n \in \mathbb{R}$, $n \in \mathbb{N}_{>0}$ are the parameters that decay sufficiently fast as $n$ increases, so that $\sum_{n=1}^{\infty} \alpha_n < \infty$. Note that this assumption is typical for characterizing uncertainty bounds with the GP regression (see, e.g., Berkenkamp, Moriconi, Schoellig, and Krause (2016)), and several ways to compute $B_C$ have recently been proposed (see, e.g., Hashimoto, Saoud, Kishida, Ushio, and Dimarogonas (2022) and Maddalena, Scharnhorst, and Jones (2021)). Moreover, assumption (iii) implies that asymptotic stability of the origin is guaranteed by applying the event-triggered controller when the event-triggered parameter is chosen such that the corresponding cost is close enough to the time-triggered case $G(\theta_c^*, 0)$.

Now, let $\widehat{\theta}_{ev,j}^* \in \mathcal{K}_{ev}$ be the optimal solution suggested by Algorithm 3 at the $j$th iteration, $\widehat{\theta}_{ev,j}^* = \arg\min_{\theta_{ev} \in \mathcal{K}_{ev}} \mu_{I,j}(\theta_{ev})$, s.t. $\mu_{C,j}(\theta_{ev}) + \beta_j \sigma_{C,j}^2(\theta_{ev}) \leq 0$. In addition, let $Y_{C,j}$ be a vector that stacks all measurements of $C(\cdot)$ until the $j$th iteration, i.e., $Y_{C,j} = [y_{C,0}, \ldots, y_{C,j}]^\top$, and let $K_C$ be the corresponding covariance matrix. Then, we obtain the following result:

**Proposition 2.** *Let Assumptions 2 hold and suppose that $C(\cdot)$ is characterized by (30) with $\widehat{\theta}_c^* = \theta_c^*$ and $\epsilon \leq \bar{\epsilon}$. Moreover, let $\beta_j > 0$ be given by $\beta_j = \sqrt{B_C^2 - Y_{C,j}^\top K_C^{-1} Y_{C,j}}$ for all $j = 1, 2, \ldots, N_{\max}$. Then, for all $j = 1, 2, \ldots, N_{\max}$, the origin is asymptotically stable by applying the event-triggered controller with $(\widehat{\theta}_c^*, \widehat{\theta}_{ev,j}^*)$.* □

Proposition 2 implies that the algorithm picks up the event-triggered parameter that fulfills asymptotic stability of the origin for all the iterations $j = 1, 2, \ldots, N_{\max}$.

**Proof.** For every $\theta_{ev} \in \mathcal{K}_{ev}$, we have

$$\begin{aligned} |\mu_{C,j}(\theta_{ev}) - C(\theta_{ev})| &\leq \mathsf{k}_{C,j}(\theta_{ev}, \theta_{ev})^{-1/2} \|\mu_{C,j} - C\|_{\mathsf{k}_C} \\ &= \sigma_{C,j}(\theta_{ev}) \|\mu_{C,j} - C\|_{\mathsf{k}_C}, \end{aligned} \tag{49}$$

where the inequality follows from the Cauchy–Schwarz inequality, and $\mathsf{k}_{C,j} : \mathcal{K} \times \mathcal{K} \to \mathbb{R}_{\geq 0}$ is given by

$$\mathsf{k}_{C,j}(\theta_{ev}, \theta'_{ev}) = \mathsf{k}_C(\theta_{ev}, \theta'_{ev}) - \mathsf{k}_{*C}^\top(\theta_{ev}) K_C^{-1} \mathsf{k}_{*C}(\theta').$$

Then, from the proof of Lemma 7.2 in Srinivas, Krause, Kakade, and Seeger (2012), it follows that

$$\|\mu_{C,j} - C\|_{\mathsf{k}_{C,j}}^2 = B_C^2 - Y_{C,j}^\top K_C^{-1} Y_{C,j}. \tag{50}$$

Hence, letting $\beta_j = \sqrt{B_C^2 - Y_{C,j}^\top K_C^{-1} Y_{C,j}}$, we have $|\mu_{C,j}(\theta_{ev}) - C(\theta_{ev})| \leq \beta_j \sigma_{C,j}(\theta_{ev})$ for all $\theta_{ev} \in \mathcal{K}_{ev}$. Therefore, for all $\theta_{ev} \in \mathcal{K}_{ev}$,

$$\begin{aligned} \mu_{C,j}(\theta_{ev}) + \beta_j \sigma_{C,j}(\theta_{ev}) &= \mu_{C,j}(\theta_{ev}) - C(\theta_{ev}) + \beta_j \sigma_{C,j}(\theta_{ev}) + C(\theta_{ev}) \\ &\geq C(\theta_{ev}). \end{aligned}$$

Hence, $\mu_{C,j}(\theta_{ev}) + \beta_{C,j} \sigma_{C,j}(\theta_{ev}) \leq 0$ implies $C(\theta_{ev}) \leq 0$. Now, from the definition of $\widehat{\theta}_{ev,j}^*$, we have $\mu_{C,j}(\widehat{\theta}_{ev,j}^*) + \beta_{C,j} \sigma_{C,j}(\widehat{\theta}_{ev,j}^*) \leq 0$ for all $j = 1, 2, \ldots, N_{\max}$, which implies $C(\widehat{\theta}_{ev,j}^*) \leq 0$ for all

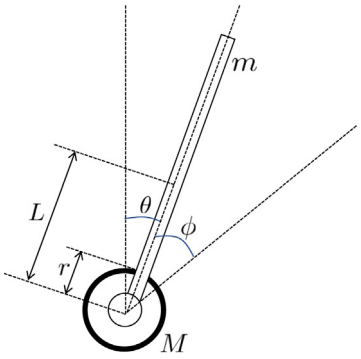
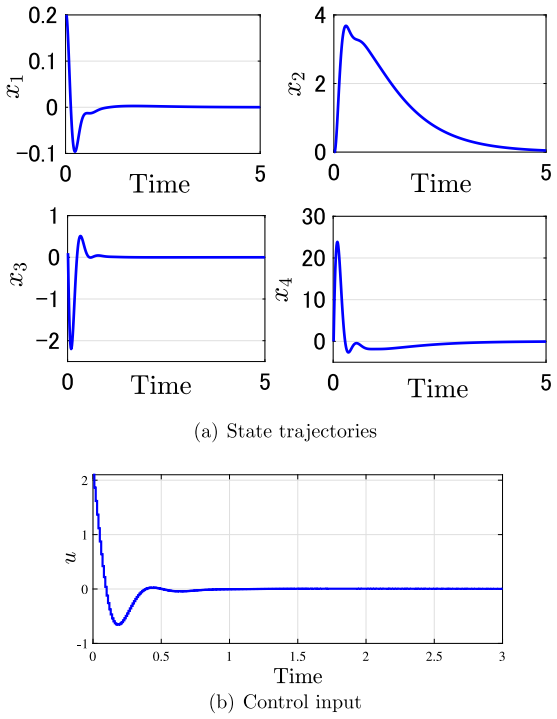



**Fig. 2.** Illustration of the two-wheeled inverted pendulum vehicle.

**Table 1**
Physical parameters used in the simulation.

| | | |
|---|---|---|
| Mass of the vehicle body | $m$ | 0.5 kg |
| Total mass of the wheels | $M$ | 0.07 kg |
| Inertia of the vehicle body | $J_V$ | $3.0 \times 10^{-3}$ kg m$^2$ |
| Inertia of the wheel | $J_W$ | $8.6 \times 10^{-6}$ kg m$^2$ |
| Inertia of the motor | $J_M$ | $1.3 \times 10^{-7}$ kg m$^2$ |
| Distance from wheel to the center | $L$ | 0.14 m |
| Radius of the wheel | $r$ | 0.025 m |
| Viscous friction coefficient | $\nu$ | $1.0 \times 10^{-4}$ kg m$^2$/s |
| Torque coefficient of the motor | $K_T$ | $2.8 \times 10^{-3}$ N m$^2$/A |
| Gear ratio | $R_G$ | 30 |
| Torque efficiency | $\eta$ | 0.75 |
| Gravity coefficient | $g$ | 9.8 m/s$^2$ |




(a) State trajectories

(b) Control input

**Fig. 3.** (a) State trajectories by applying the time-triggered controller with the control parameter found by Algorithm 2; (b) Corresponding control input.

$j = 1, 2, \ldots, N_{\max}$. From Assumption 2 and the assumption that $C(\cdot)$ is given by (30) with $\epsilon \leq \bar{\epsilon}$, it follows that the origin is asymptotically stable by applying the event-triggered controller with $(\widehat{\theta}_c^*, \widehat{\theta}_{ev,j}^*)$. □

## 6. Simulation result

In this section we illustrate the effectiveness of the proposed approach through a numerical simulation. The simulation was implemented on Matlab 2016a under Windows 10, Intel(R) Core (TM) i5-7600 CPU 3.50 GHz, 8.00 GB RAM.

As an example of the simulation, we consider a control problem of a two-wheeled inverted pendulum vehicle (see Fig. 2), whose dynamics, which is completely unknown apriori, is governed by

$$\{(M+m)r^2 + mLr\cos\theta + J_W + R_G J_M\}\ddot{\theta} - mLr\dot{\theta}^2\sin\theta + \{(m+M)r^2 + J_W + R_G^2 J_M\}\ddot{\phi} + \nu\dot{\phi} = \eta R_G K_T u \tag{51}$$

$$\{(M+m)r^2 + 2mLr\cos\theta + mL^2 + J_V + J_W + R_G J_m\}\ddot{\theta} + \{(m+M)r^2 + mLr\cos\theta + J_W + R_G^2 J_M\}\ddot{\phi} + \nu\dot{\phi} = mLr\dot{\theta}^2\sin\theta + mgL\sin\theta \tag{52}$$

where $\theta$, $\phi$, and $u$ are the angle of the vehicle body, angle of the wheel, and the force or the torque applied to the wheel, respectively. Moreover, the physical parameters used in the above are shown in Table 1. The state vector is $x = [\theta, \phi, \dot{\theta}, \dot{\phi}]^\mathsf{T} \in \mathbb{R}^4$, and the input is $u \in \mathbb{R}$. The initial state is given by $x(0) = [0.2, 0, 0, 0]^\mathsf{T}$. It is assumed that $\underline{h} = 1.0 \times 10^{-2}$, $H = 1.0 \times 10^3$, and the cost function is given by the convergence time towards the neighborhood of the origin:

$$J(\mathbf{x}(H), \mathbf{u}(H)) = \min_k \left\{ t_k : \forall k' \in [k : H-1],\ \|x(t_{k'})\|_\infty \leq \eta \right\},$$

with $\eta = 1.0 \times 10^{-4}$. Moreover, it is assumed that the control law is given by $u(t) = [\kappa_1, \kappa_2, \kappa_3, \kappa_4]\hat{x}(t)$, where $\theta_c = [\kappa_1, \kappa_2, \kappa_3, \kappa_4]^\mathsf{T} \in \mathbb{R}^4$ is the control parameter. The control parameter space is given by $\mathcal{K}_c = \{[\kappa_1, \kappa_2, \kappa_3, \kappa_4]^\mathsf{T} \in \mathbb{R}^4 : 0 \leq \kappa_1, \kappa_2, \kappa_3, \kappa_4 \leq 10\}$, from which we aim to find the optimal $\theta_c^*$ by applying Algorithm 2. For the event-triggered strategy, we consider the static case:

$$g(x(t_k), x(t_\ell); \lambda) = \|x(t_k) - x(t_\ell)\| - \lambda\|x(t_k)\|, \tag{53}$$

where $\lambda \in \mathcal{K}_{ev}$ is the event-triggered parameter with $\mathcal{K}_{ev} = \{\lambda \in \mathbb{R}_{\geq 0} : 0 \leq \lambda \leq 0.6\}$.

### 6.1. Solution to Problem 1

Now, let us design the optimal control parameter $\theta_c^*$ by applying Algorithm 2 as the solution to Problem 1. During the implementation of this algorithm, we set $N_{init} = 20$, $N_{\max} = 40$, and the kernel function for the GP model of $G(\theta_c, \theta_{ev})$ is given by the squared exponential covariance function (1), whose hyperparameters are updated for each round in Algorithm 2 using the evidence maximization (Rasmussen & Williams, 2006). Moreover, the penalty map $P : \mathcal{K}_{ev} \to \mathbb{R}_+$ is assumed to be given by (32) with $\gamma_0 = 0.1, \gamma_1 = 0.1$. Fig. 3 illustrates the resulting state trajectories by applying the control law under the time-triggered controller with the optimal control parameter obtained by Algorithm 2. The figure shows that the (time-triggered) controller designed by the proposed approach successfully stabilizes the system towards the origin. The optimal control parameter estimated by Algorithm 2 is $\widehat{\theta}_c^* = [10, 0.04, 1, 0.06]^\mathsf{T}$ and the corresponding optimal cost (convergence time) is $\widehat{G}^* = 8.6$ s.

The left figure of Fig. 4 illustrates the average values of the computed optimal cost obtained by applying Algorithm 2 under different selections of $N_{\max} \in \{40, 60, 80, 100\}$. Here, for each $N_{\max} \in \{40, 60, 80, 100\}$, the average was taken by applying Algorithm 2 for 20 times. To make comparisons, we have also plotted the result by the *naive* approach described in Section 4 (i.e., the time-triggered controller is implemented in Step 2 for every round of the implementation of the Bayesian optimization).

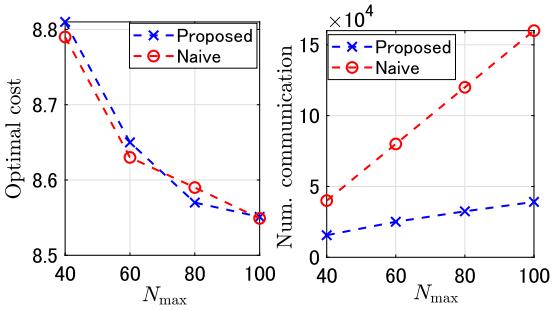



**Fig. 4.** Average values of the computed optimal cost (left) and the total number of communication time steps required to implement Algorithm 2 (right).

The figure shows that the optimal costs obtained by applying the naive approach and Algorithm 2 are almost the same. Moreover, the right figure of Fig. 4 illustrates the average of the total number of communication time steps required to implement Algorithm 2 (as well as the naive approach). The figure shows that the proposed approach requires much smaller total number of communication time steps than the naive approach. Therefore, it is shown that the proposed approach results in a reduction of the network utilization for solving the optimization problem (Problem 1), while providing almost the same optimal cost as the naive approach.

### 6.2. Solution to Problem 2

Recall in Section 6.1 that the optimal control parameter and the corresponding optimal convergence time by applying Algorithm 2 are $\widehat{\theta}_c^* = [10, 0.04, 1, 0.06]^\mathsf{T}$ and $\widehat{G}^* = 8.6$ s, respectively. The fact that the optimal convergence time under the time-triggered controller is 8.6 s is useful to design the event-triggered parameter. That is, since we know that the underlying performance of the time-triggered controller is 8.6, this information can be utilized to emulate the control performance under the event-triggered controller. Suppose that we would like to design the event-triggered parameter under the constraint that the maximum allowable convergence time is up to 5% with respect to the time-triggered case. In other words, we impose the constraint that the convergence time should be less than $8.6 \times 1.05 \approx 9.0$ s. Hence, we aim at solving the following problem:

$$\min_{\theta_{ev}\in[0,0.6]} \mathcal{I}(\widehat{\theta}_c^*, \theta_{ev}), \quad \text{s.t. } G(\widehat{\theta}_c^*, \theta_{ev}) \le 9.0. \tag{54}$$

Fig. 5 illustrates the resulting state trajectories by applying Algorithm 3. During the implementation of Algorithm 3, we set $N_{\text{init}} = 10$, $N_{\max} = 30$. It can be shown that the corresponding convergence time is 8.8 s, which satisfies the desired constraint. The corresponding inter-event times are shown in Fig. 6. The resulting optimal event-triggered parameter estimated by applying Algorithm 3 and the corresponding number of communication time steps are $\widehat{\theta}_{ev}^* = 0.1$ and $\mathcal{I}(\widehat{\theta}_c^*, \widehat{\theta}_{ev}^*) = 312$, respectively.

### 6.3. Employing a dynamic event-triggered controller to solve Problem 2

In the previous subsection, we design the static event-triggered controller to solve Problem 2. In this subsection, we investigate a *dynamic* event-triggered controller, in which we incorporate an internal variable for the threshold of the event-triggered condition. As will be seen below, employing the dynamic event-triggered controller will provide a less number of communication time steps than the static case, while preserving the same control performance as the time-triggered case. Here,

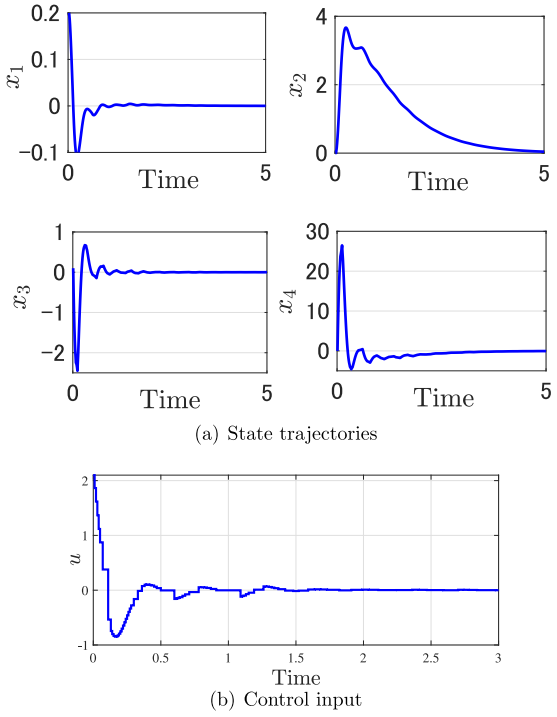



**Fig. 5.** (a) State trajectories by applying the static event-triggered controller with the event-triggered parameter found by Algorithm 3; (b) Corresponding control input.

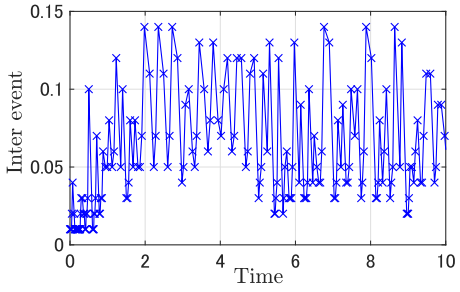



**Fig. 6.** Inter-event times corresponding to the result of Fig. 5.

we make use of the solution that has been obtained for Problem 1 $\widehat{\theta}_c^* = [10, 0.04, 1, 0.06]^\mathsf{T}$ and $\widehat{G}^* = 8.6$ s, and focus on solving Problem 2 to design the dynamic event-triggered controller. More specifically, we consider the event-triggered strategy (15) with

$$g(x(t_{k_m}), x(t_{k_\ell}); \theta_{ev}) = -\lambda_1 e^{-\lambda_2 t_{k_m}} + \lambda_3 \left( \|x(t_{k_m}) - x(t_{k_\ell})\| - \lambda_4 \|x(t_{k_m})\| \right),$$

where $\theta_{ev} = [\lambda_1, \lambda_2, \lambda_3, \lambda_4]^\mathsf{T} \in \mathbb{R}^4$ is the event-triggered parameter to be designed. The term $\lambda_1 e^{-\lambda_2 t_{k_m}}$ serves as an internal variable having the role to "push down" the value of $g$ so as to increase the possibility to satisfy the event-triggered condition. It is assumed that $\theta_{ev} \in \mathcal{K}_{ev}$ with $\mathcal{K}_{ev} = \{[\lambda_1, \lambda_2, \lambda_3, \lambda_4]^\mathsf{T} \in \mathbb{R}^4 : 0 \le \lambda_1, \lambda_2, \lambda_3, \lambda_4 \le 1.0\}$.

Figs. 7 and 8 illustrate the state trajectories and the corresponding inter event times by applying the dynamic event-triggered parameter found by applying Algorithm 3. During the implementation of Algorithm 3, we set $N_{\text{init}} = 10$, $N_{\max} = 30$. It can be shown that the convergence time is 8.8s, which satisfies the desired constraint. Moreover, as shown in Fig. 8, the inter-event times for the dynamic event-triggered case tend to be larger than the static case as the time evolves. Indeed, the total

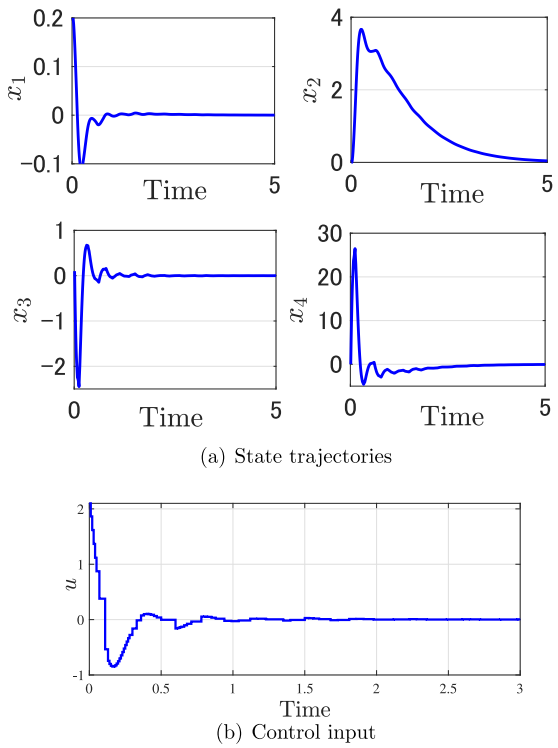



**Fig. 7.** (a) State trajectories by applying the *dynamic* event-triggered controller with the event-triggered parameter found by Algorithm 3; (b) Corresponding control input.

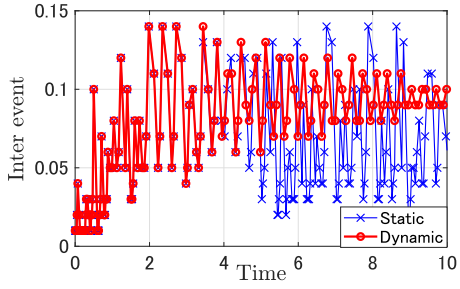



**Fig. 8.** Inter-event times corresponding to the result of Fig. 7 (red dots) and the previous static case (blue dots). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

number of communication time steps is $\mathcal{I}(\widehat{\theta}_c^*, \widehat{\theta}_{ev}^*) = 256$, which is less than the static case (i.e., 312), showing the effectiveness of employing the dynamic event-triggered controller.

## 7. Conclusion

This paper investigates a model-free, emulation-based design of a periodic event-triggered controller with the assumption that the dynamics of the plant is unknown apriori. In particular, we provide two different types of the Bayesian optimization algorithms to design parameters for the control law and for the event-triggered condition. The problem for designing the control law is solved based on a multi-information source optimization (MISO), which is a modification from the basic Bayesian optimization, such that the number of communication time steps can be taken into account while solving the optimization problem. The problem for designing parameters for the event-triggered condition is solved via constrained Bayesian optimization, which is a modification from the basic Bayesian optimization, such that a certain constraint in the optimization problem can be taken into account. Finally, a numerical example of a control problem of a two-wheeled inverted pendulum vehicle illustrates the effectiveness of the proposed approach.

## CRediT authorship contribution statement

**Kazumune Hashimoto:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Formal analysis, Conceptualization.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Kazumune Hashimoto reports financial support was provided by Osaka University. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work is partially supported by JST CREST JPMJCR201, JST ACT-X JPMJAX23CK, and JSPS KAKENHI Grant 21K14184, and 22KK0155.

## References

Abbracciavento, F., Zinnari, F., Formentin, S., Bianchessi, A. G., & Savaresi, S. M. (2023). Multi-intersection traffic signal control: A decentralized MPC-based approach. *IFAC Journal of Systems and Control*, *23*, Article 100214.

Anand, S., Kumar, M., Kumar, S., & Arkdev (2023). Discrete-time prediction based event-triggered controller design: An application to networked multi-area power system with time delays. *IFAC Journal of Systems and Control*, *25*, Article 100220.

Antunes, D., & Heemels, W. P. M. H. (2014). Rollout event-triggered control: Beyond periodic control performance. *IEEE Transactions on Automatic Control*, *59*(12), 3296–3311.

Bansal, S., Calandra, R., Xiao, T., Levine, S., & Tomlin, C. J. (2017). Goal-driven dynamics learning via Bayesian optimization. In *Proceedings of the 56th IEEE international conference on decision and control* (pp. 5168–5173).

Beckers, T., Kulic, D., & Hirche, S. (2019). Stable Gaussian process based tracking control of Euler-Lagrange systems. *Automatica*, *103*, 390–397.

Berkel, F., & Liu, S. (2018). An event-triggered cooperation approach for robust distributed model predictive control. *IFAC Journal of Systems and Control*, *6*, 16–24.

Berkenkamp, F., Moriconi, R., Schoellig, A. P., & Krause, A. (2016). Safe learning of regions of attraction for uncertain, nonlinear systems with gaussian processes. In *2016 IEEE 55th conference on decision and control* (pp. 4661–4666). IEEE.

Boardman, B., Harden, T., & Martínez, S. (2021). Multi-agent motion planning with sporadic communications for collision avoidance. *IFAC Journal of Systems and Control*, *15*, Article 100126.

Dimagoronas, D. V., Frazzoli, E., & Johansson, K. H. (2012). Distributed event-triggered control for multi-agent systems. *IEEE Transactions on Automatic Control*, *57*(5), 1291–1297.

Donkers, M. C. F., & Heemels, W. P. M. H. (2011). Output-based event-triggered control with guaranteed $\mathcal{L}_\infty$ gain and decentralized event-triggering. *IEEE Transactions on Automatic Control*, *57*(6), 1362–1376.

Frazier, P. I. A tutorial on Bayesian optimization, arxiv, available online at https://arxiv.org/pdf/1807.02811.pdf.

Frazier, P., Powell, W., & Dayanik, S. (2009). The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, *21*(4), 599–613.

Gardner, J. R., Kusner, M. J., Xu, Z. E., Weinbrger, K. Q., & Cunningham, J. P. (2014). Bayesian optimization with inequality constraints. In *Proceedings of international conference on machine learning* (pp. 937–945).

Girard, A. (2014). Dynamic triggering mechanisms for event-triggered control. *IEEE Transactions on Automatic Control*, *60*(7), 1992–1997.

Gommans, T., Antunes, D., Donkers, T., Tabuada, P., & Heemels, M. (2014). Self-triggered linear quadratic control. *Automatica*, *50*(4), 1279–1287.

Hashimoto, K., Adachi, S., & Dimarogonas, D. V. (2017). Event-triggered intermittent sampling for nonlinear model predictive control. *Automatica*, *81*, 148–155.

Hashimoto, K., Onoue, Y., Ogura, M., & Ushio, T. (2021). Event-triggered control for mitigating SIS spreading processes. *Annual Reviews in Control*, *52*, 479–494.

Hashimoto, K., Saoud, A., Kishida, M., Ushio, T., & Dimarogonas, D. (2022). Learning-based symbolic abstractions for nonlinear control systems. *Automatica*.

Hashimoto, K., Yoshimura, Y., & Ushio, T. (2020). Learning self-triggered controllers with Gaussian processes. *IEEE transactions on cybernetics*, *51*(12), 6294–6304.

Heemels, W. P. M. H., & Donkers, M. C. F. (2013). Model-based periodic event-triggered control for linear systems. *Automatica*, *49*(3), 698–711.

Heemels, W. P. M. H., Donkers, M. C. F., & Teel, A. R. (2013). Periodic event-triggered control for linear systems. *IEEE Transactions on Automatic Control*, *58*(4), 847–861.

Heemels, W. P. M. H., Johansson, K. H., & Tabuada, P. (2012). An introduction to event-triggered and self-triggered control. In *Proceedings of the 51st IEEE conference on decision and control* (pp. 3270–3285).

Hewing, L., Liniger, A., & Zeilinger, M. N. (2018). Cautious NMPC with Gaussian process dynamics for autonomous miniature race cars. In *Proceedings of 2018 European control conference*.

Jain, A., Nghiem, T. X., Morari, M., & Mangharam, R. (2018). Learning and control using gaussian processes: towards bridging machine learning and controls for physical systems. In *Proceedings of the 9th ACM/IEEE international conference on cyber-physical systems*.

Klenske, E. D., Zeilinger, M. N., Scholkopf, B., & Hennig, P. (2019). Gaussian process-based predictive control for periodic error correction. *IEEE Transactions on Control Systems Technology*, *24*(1), 390–397.

Li, J., Wang, L., Xu, M., & Fang, Y. (2023). Adaptive event-triggered robust model predictive control for multiagent systems with time-varying delay under arbitrary network topology. *IEEE Systems Journal*, *17*(4), 5381–5392. http://dx.doi.org/10.1109/JSYST.2023.3313232.

Maddalena, E. T., Scharnhorst, P., & Jones, C. N. (2021). Deterministic error bounds for kernel-based learning techniques under bounded noise. *Automatica*.

Marco, A., Hennig, P., Bohg, J., Schaal, S., & Trimpe, S. (2016). Automatic LQR tuning based on Gaussian process global optimization. In *Proceedings of 2016 IEEE international conference on robitics, automation* (pp. 270–277).

Mockus, J. (1989). *Bayesian Approach to Global Optimization: Theory and Applications*. Kluwer Academic Publishers.

Neumann-Brosig, M., Marco, A., Schwarzmann, D., & Trimpe, S. (2020). Data-efficient autotuning with Bayesian optimization: An industrial control study. *IEEE Transactions on Control Systems Technology*, *28*(3), 730–740.

Poloczek, M., Wang, J., & Frazier, P. I. (2017). Multi-information source optimization. In *Proceedings of advances in neural information processing systems* (pp. 4288–4298).

Postoyan, R., Anta, A., Heemels, W. P. M. H., Tabuada, P., & Nesic, D. (2013). Periodic event-triggered control for nonlinear systems. In *Proceedings of the 52nd IEEE conference on decision and control* (pp. 7397–7402).

Qi, Y., Tang, Y., Zhao, X., Xing, N., & Qiu, J. (2023). Dual event-triggered control for asynchronous scheduling parameter varying networked switched systems under DoS attacks. *IEEE Systems Journal*, *17*(4), 5854–5865.

Rasmussen, C. F., & Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.

Seuret, A., Prieur, C., Tarbouriech, S., & Zaccarian, L. (2013). Event-triggered control with LQ optimality guarantees for saturated linear systems. In *9th IFAC symposium on nonlinear control systems* (pp. 341–346).

Seuret, A., Prieur, C., Tarbouriech, S., & Zaccarian, L. (2016). LQ-based event-triggered controller co-design for saturated linear systems. *Automatica*, *74*, 47–54.

Srinivas, N., Krause, A., Kakade, S. M., & Seeger, M. W. (2012). Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, *58*(5), 3250–3265.

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

Tarbouriech, S., Seuret, A., da Silva, J. M. G., & Sbarbaro, D. (2016). Observer-based event-triggered control co-design for linear systems. *IET Control Theory Applications*, *10*(18), 2466–2473.

Wang, X., & Lemmon, M. D. (2009). Self-triggered feedback control systems with finite $\mathcal{L}_2$ gain stability. *IEEE Transactions on Automatic Control*, *54*(3), 452–467.