# Event-Triggered Control Design with Performance Barrier

Pio Ong and Jorge Cortés

*Abstract*— **This paper revisits the problem of designing opportunistic state-triggered conditions for stabilization, focusing on the balance between conserving resources (e.g., minimizing the number of triggers) and meeting a desired level of performance. Traditionally, event-triggered control design focuses on ensuring stabilization while conservatively enforcing that the specified performance is met. We take a different approach by considering the desired performance as part of the trigger design. Inspired by the concept of Control Barrier Function, our proposed design allows the system to deviate from Lyapunov's condition for asymptotic stability when the system is doing well in term of performance. We characterize the benefits of the proposed approach in terms of increased inter-event time, robustness to delays in the evaluation of the trigger, and flexibility for distributed implementation.**

## I. Introduction

Control systems are usually implemented on a digital platform. As a result, the control signals are rarely updated continuously. In the past decade, event-triggered control has been gaining popularity with its opportunistic approach to control implementation in discrete settings. The main idea is to conserve resources by updating the control signal only when it is *necessary* rather than periodically. We emphasize the word *necessary* here because generally conserving resources will lead to sacrificing some performance, and it is unclear when it is required to update a signal in order to meet a desired performance. Generally, when coming up with a design, there is a need to answer the question of how to strike a balance between the two factors. In this paper, we are interested in studying how to design a state-based trigger condition given a desired level of performance.

*Literature Review*. In this paper, we rely on three bodies of literature. The first body of work which we rely on is the literature on opportunistic and event-triggered control. We recommend the reader [1] and references therein for the introduction to the topic. In the literature, we have identified the main type of trigger condition designs to be derivative based like presented in [2]–[6]. This type of design focuses on ensuring the closed-loop system's stability through monitoring the time-derivative of the Lyapunov function. Although this type of design can be adapted to ensure the desired performance, we find that it is not the most efficient because performance is only considered after the proposition of the design process. Another type of design is the value-based design which uses the value of the Lyapunov function to monitor when to update the control. A simple example is Lyapunov Sampling [7], [8] which updates the control when the Lyapunov function drops below certain percentage threshold. The problem with such a method is that it may be difficult to determine what amount of threshold is allowed, and the design can be very conservative in term of achieving performance. Nevertheless, we notice in [9] that when the value-based trigger is used in combination with a predefined performance function, not only does the problem mentioned is eliminated, but the amount of inter-event time is also extended to the longest possible while satisfying the predefined level of performance. However, we see that this kind of design lacks the robustness to control implementation delay. On the contrary, the work in [10], which combines the idea of the derivative-based design with Lyapunov Sampling, retains some level of robustness, but suffers from the drawback of Lyapunov Sampling as well as reliance on time trigger due to the possibility of not triggering. To the best of our knowledge, there is no work that combines derivative-based trigger design and a value-based trigger design in combination with a predefined performance function like we shall do in this paper. In doing so, we rely on the body of work on control barrier function. The idea is that like in control barrier function, presented in [11], we have safe and unsafe states, safe being the states that satisfy performance at a given time. Recent development in control barrier function inspires the development of our paper, namely the work in [12] which uses the idea of Nagumo's Theorem presented in [13]. Lastly, because of the potential of our work, we pay attention to the numerous works of event-triggered control in the context of networked systems, e.g. [14]–[18]. One particular interesting phenomenon in extending event-triggered control to distributed settings is that Zeno behavior can be introduced. [19] shows that to avoid such problem, it is natural to violate Lyapunov's condition for stability by adding a positive term to the design. This paper does something similar except that the term we add is not a constant, but rather depends on the performance.

*Statement of Contributions:* This paper revisits the problem of designing a state-triggered conditions for stabilization. We consider the common scenario where conserving resources is preferred but the performance of the system must also satisfy a desired level. For our proposed designs, we directly incorporate the performance requirement into the trigger condition. We first consider the centralized case where all the states are updated simultaneously. Then we extend our consideration to distributed scenarios. The contributions of this paper are threefold. The first is the design of event-triggered laws that efficiently extend the minimum inter-event time between triggers while meeting a desired level of performance. Our design is based on the examining simultaneously the time-derivative and the value of the Lyapunov

This work was supported by NSF award CNS-1329619.

Pio Ong and Jorge Cortés are with Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA 92093, USA, {piong,cortes}@eng.ucsd.edu

function and does not extend the inter-event time to the extreme so that the closed-loop system remains robust to control implementation delay. Our second contribution is the establishment of a uniform lower bound in the inter-event times of our design in the centralized case, which thereby rules out the possibility of Zeno behavior. The result is based on idea that the inter-event times for our design must be longer than that of a known design that is also lower bounded. In addition, we provide an expression to find the lower bound for our design in linear system case with an exponential desired performance. Finally, our third contribution is an extension of our design into the distributed settings. Through simulations, we show that in the distributed scenarios, the design can fix the Zeno problem that traditional event-triggered design suffer, and we show that the new design opens up the possibility for trigger coordination in a networked system. All proofs are omitted for reasons of space and will appear elsewhere.

*Notation:* We denote by $\mathbb{N}$ and $\mathbb{R}$ the set of natural and real numbers, respectively. For $n \in \mathbb{N}$, we use the notation $[n]$ to denote the set $\{1, \ldots, n\}$. Given $x \in \mathbb{R}^n$, $\|x\|$ denotes its Euclidean norm. We denote the identity matrix by $I_n \in \mathbb{R}^{n\times n}$. For a square matrix $C$, $\operatorname{eig}(C)$ denotes its set of eigenvalues. The function $f$ is locally Lipschitz if, for every compact set $\mathcal{S}_0 \subset \mathbb{R}^n$, there exists a positive constant $L$, termed Lipschitz constant, such that $\|f(x) - f(y)\| \leq L\|x - y\|$, for all $x, y \in \mathcal{S}_0$. For a twice continuously differentiable, scalar-valued function $g : \mathbb{R}^n \to \mathbb{R}$, we let $\nabla g : \mathbb{R}^n \to \mathbb{R}^n$ and $\nabla^2 g : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ denote its gradient and Hessian functions. The function $g$ is convex if $\nabla^2 g \succeq 0$, and concave if $\nabla^2 g \preceq 0$. We use the notation $\mathcal{L}_f$ for the Lie derivative operator on the vector field $f : \mathbb{R}^n \to \mathbb{R}^n$. We call a continuous function $h : \mathbb{R} \to \mathbb{R}$ a class-$\mathcal{K}$ function if it is strictly increasing and $h(0) = 0$. In addition, we call the function class-$\mathcal{K}_\infty$ if the function also satisfies $\lim_{r\to\infty} h(r) = \infty$. Lastly, in a networked system, we denote a graph by $G = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V}$ as the set of vertices and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. $j$ is a neighbor of $i$ if and only if $(i, j) \in \mathcal{E}$. We write $\mathcal{N}_i$ to be a discrete set of number of nodes that include node $i$ and all its neighbors, and $\mathcal{N}_i^2$ the set of nodes including $\mathcal{N}_i$ and all neighbors of each $j \in \mathcal{N}_i$. We add the subscript $x_{\mathcal{N}_i}$ to represent the subvector of a vector $x$ formed from the entries in the set $\mathcal{N}_i$.

## II. Problem Formulation

Consider a nonlinear controlled system

$$\dot{x} = f(x, u),\ x \in \mathbb{R}^n,\ u \in \mathbb{R}^m$$

with $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. In a digital implementation of a state-feedback control signal $u = k(x)$ with the controller function $k : \mathbb{R}^n \to \mathbb{R}^m$, the sample-and-hold strategy is more practical than providing a continuous signal. The control is only updated at a specific time $t_k$ and held up until $t_{k+1}$ when the controller provides an adjustment to the control signal. As a result, the closed-loop system is given by,

$$\dot{x} = f(x, k(x + e)) \tag{1}$$

where the error $e = x(t_k) - x$ is the state deviation from last update iteration $k$. In event-triggered control, it is preferable to extend the time interval at which the signal is held. To do so, $t_{k+1}$ is determined iteratively by a prescribed criterion. In order to come up with a criterion for a general non-linear system, a common starting point is to assume that there exists a known Input-to-State Stability (ISS) Lyapunov function for the system, see e.g. [2], [3], [5]. In other words, there exists a smooth function $V : \mathbb{R}^n \to \mathbb{R}$ such that there exist class-$\mathcal{K}_\infty$ functions $\underline{\alpha}$, $\overline{\alpha}$, $\alpha$, and $\gamma$ satisfying

$$\underline{\alpha}(\|x\|) \leq V(x) \leq \overline{\alpha}(\|x\|) \tag{2a}$$

$$\mathcal{L}_f V(x) \leq -\alpha(\|x\|) + \gamma(\|e\|). \tag{2b}$$

Under such characterizations, [2], for example, provides the following trigger design

$$t_{k+1} = \big\{t > t_k \mid -\sigma\alpha(\|x\|) + \gamma(\|e\|) = 0\big\} \tag{3}$$

with the design parameter $\sigma \in (0, 1)$. As such, the Lie derivative (2b) can be guaranteed to be less than or equal to zero, and stabilization is achieved. As one can deduce, the choice of parameter $\sigma$ in the design directly impacts the convergence performance. On the other hand, the choice will also affect the occurrence frequency of the update trigger. In the scenario where the desired convergence performance is specified, it is possible to select the largest $\sigma$ that meet the specified performance in order to extending the minimum inter-event time. However, as will be shown in this paper, it is not the most efficient way. Our paper improves upon the design in order to extend the inter-event times while satisfying the specified performance. The basic premise of this paper is to incorporate the performance specification into the trigger condition by using it as a *barrier*, preventing the violation of the specification.

Before we proceed to our design propositions, we discuss our assumptions on the how the performance is specified. First, we assume that the specification prescribes an upper bound to how fast the Lyapunov function converges to zero. In other words, the specification function, $S : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}_+$, requires at all time that

$$V(x(t)) \leq S(x_0, t). \tag{4}$$

Generally, we cannot hope that a sample-and-hold control strategy will do better than a system with continuous state-feedback control signal. Therefore, we have to make assumptions on the function $S$ so that it is not too strict. In our paper, we follow the common practice and assume that the system in consideration is ISS with known functions in conditions (2a) and (2b). Then, we assume the performance specification function satisfies the following:

1) $S$ is the unique solution to the differential equation

$$\dot{S} = -h(S),\ S(x_0, 0) \geq V(x_0)$$

where $h$ is a locally Lipschitz, class $\mathcal{K}$ function (this implies $S$ is a class $\mathcal{KL}$ function, cf. [20, Lemma 4.4]);
2) Let $\bar{V}$ be a continuous function such that $\bar{V}(x) \geq V(x)$ for all $x$. We assume that $\alpha(\|x\|) > h(\bar{V}(x))$ for all $x$.

Particularly, let $c : \mathbb{R}^n \to \mathbb{R}$ be such that $\alpha(\|x\|) = c(x)h(\bar{V}(x))$. Note that $\min_x c(x) = c_{\min} > 1$.

The first condition is simply assuring that there is an explicit form to the derivative of the given performance function. Note that although this assumption does not permit all of the possible requirement functions that specifies asymptotic stability of the origin, it does allow the exponential form, $S(x_0, t) = V(x_0)\exp(-rt)$, which is one of the most natural performance requirement especially for linear system. The second condition makes sure that at least in a continuous implementation of the controller, $\frac{d}{dt}V(x) < -h(V(x))$ so that it is possible to guarantee the specified performance.

## III. Performance Based Event-Triggered Control Designs

In this section, we incorporate the performance specification into the event-triggered control design. We assume that the trigger condition can be checked continuously, i.e. the states can be accessed in real-time to check if the condition is met. We begin by adapting the trigger design (3), which monitors time derivative of the Lyapunov function, to accommodate the specification function.

*Lemma 3.1: (Derivative-Based Design):* Let $g : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be any function such that

$$\mathcal{L}_f V(x) \le g(x,e) \le -\alpha(\|x\|) + \gamma(\|e\|).$$

For any $k \in \mathbb{N} \cup 0$, if $V(x(t_k)) \le S(x_0, t_k)$ and

$$t_{k+1} = \min_t \big\{ t > t_k \mid g(x(t), e(t)) + h(\bar{V}(x(t))) = 0 \big\} \quad (5)$$

then $V(x(t)) \le S(x_0, t)$ for time $t \in [t_k, t_{k+1})$. $\square$

Lemma 3.1 states that using design (5) will lead to a trajectory that meets the performance specification for each iteration $k$. The key idea of the design is to keep the time-derivative of Lyapunov function to be below the amount that would satisfy the Comparison Lemma with respect to the desired performance. While this will lead to meeting the performance specified, it may not be the most efficient. The intuition comes from the fact that during the whole iteration $(t_k, t_{k+1})$, we can guarantee that $\frac{d}{dt}V(x(t)) < -h(V(x(t))$, which implies that the Lyapunov function decrease at a faster rate than necessary to satisfy the specified performance. Therefore, we should be able to allow a longer inter-event time even if $\frac{d}{dt}V(x(t)) > -h(V(x(t))$ for some time to compensate the prior overperformance.

With the above in mind, one natural trigger design may be the following,

$$t_{k+1} = \big\{ t > t_k \mid S(x_0, t) - V(x(t)) = 0 \big\}, \quad (6)$$

because it immediately guarantees that $S(x_0, t) \ge V(x(t))$ regardless of its evolution. However, problems with such straightforward design are:

1) either the exact value of the Lyapunov function is required, or the specification function would have be a bound on $\bar{V}$;
2) the design lacks robustness in that there is no room for neither error nor trigger time delay.

These problems motivate our following design which is a combination of designs (5) and (6).

*Proposition 3.2: (Performance Barrier Design):* For any function $\beta$ that is a $\mathcal{K}_\infty$ function on $[0, \infty)$ and 0 elsewhere, let $t_{k+1}$ be determined as follows,

$$t_{k+1} = \min_t \big\{ t > t_k \mid g(x(t), e(t)) + h(\bar{V}(x(t))) = \beta\big(S(x_0, t) - \bar{V}(x(t))\big) \big\}. \quad (7)$$

If $S(x_0, t_k) \ge V(x(t_k))$, then $S(x_0, t) \ge V(x(t))$ is satisfied for $t \in [t_k, t_{k+1}]$. $\square$

The main idea behind the design (7) is inspired by the recent development in control barrier function presented in [12], which uses Nagumo's Theorem from [13]. The statement in the theorem can be roughly interpreted that a set is forward invariant if and only if the state flow inward (or tangent to) at the boundary. In other words, it does not matter how the state evolves in the interior. Similarly, in our case, we have ensured that at any time the trajectory evolves to the *performance boundary* where $V(x(t)) = S(x_0, t)$, the Lyapunov function decreases faster than the desired performance. We next elaborate on the advantages of the proposed design.

*Remark 3.3: (Increase in Inter-event Time):* Comparing the traditional design (5) with the proposed design (7), the latter has a higher inter-event time. This is because both designs share the same lefthand side, whereas the righthand side of (7) remains greater than zero. The benefit of including the performance in consideration is that the user will no longer need to worry about the trade-off between minimum inter-event time and performance. Instead, we can guarantee the performance while extending the inter-event time. Our design does not extend the inter-event time to be as long as possible because such a design, cf. (6), loses robustness. We discuss this point in the following remark. •

*Remark 3.4: (Robustness to Trigger Delay):* One of the advantages of our design is that even though the inter-event time is extended, the design will leave some residual, $S(t_{k+1}) - V(x(t_{k+1})) > 0$. This will translate to allowing some time that the system can delay the control implementation while still satisfying the specified performance. Although characterizing an expression for the amount of time is beyond the scope of this paper, a simulation on a linear system example has shown that this time is lower bounded. •

While Proposition 3.2 shows that desired performance is fulfilled during each iteration $k$, we cannot yet conclude that it will do so at all time because we have not discarded the possibility of Zeno behavior, i.e. the sequence of trigger time, $\{t_k\}_{k=0}^{\infty}$, converges to a value. The next result will remove such possibility by providing a constant lower bound to all of the inter-event times.

*Proposition 3.5: (Minimum Inter-event Time):* If the following functions are locally Lipschitz:

1) $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ with constant $L$;
2) $\gamma : \mathbb{R} \to \mathbb{R}$ with constant $L_\gamma$;
3) $h^{-1} : \mathbb{R} \to \mathbb{R}$ with constant $L_{h^{-1}}$;
4) $\underline{\alpha}^{-1} : \mathbb{R} \to \mathbb{R}^n$ with constant $L_{\underline{\alpha}^{-1}}$,

then for the trigger design (7), the inter-event times are lower bounded as follows

$$t_{k+1} - t_k > \frac{1}{LP + L} \tag{8}$$

for all $k \in \mathbb{N} \cup 0$ where $P = \frac{L_{\alpha^{-1}} L_{h^{-1}} L_\gamma}{c_{min} - 1}$. $\square$

With Proposition 3.5, Zeno behavior is ruled out and the control system with trigger design (5) is guaranteed to meet the specified performance at all time. Note that although the minimum inter-event time for our design (7) is longer than design (5), the expression given by (8) ignores the performance residual term, $S(t) - V(x)$ that provides the extra time. Unfortunately, we do not have the expression for the longer inter-event time, and it will be the topic of our future research. However, for linear cases with exponential performance specification, we can give an expression for the minimum inter-event time for our design.

In order to give an expression for the longer minimum inter-event time, we consider the following special case,

$$f(x, k(x+e)) = (A + BK)x + BKe, \tag{9a}$$

$$V(x) = x^T P x, \tag{9b}$$

$$\mathcal{L}_f V(x) = -x^T Q x + 2x^T PBKe \tag{9c}$$

$$S(x_0, t) = V(x_0)\exp(-rt) \tag{9d}$$

with constants $r > 0$, $B$, $K$, positive definite matrix $P$, $-Q = P(A + BK) + (A + BK)^T P$, and Hurwitz $A + BK$. Also, to satisfy our assumption of the performance specification, we must have

$$c(x) = \frac{x^T Q x}{r x^T P x} \geq \frac{\min \operatorname{eig}(Q)}{r \max \operatorname{eig}(P)} > 1.$$

For the above system, we can give the following design,

*Corollary 3.6: (Linear Case with Exponential Desired Performance):* For the system given by equations (9). For any positive constant $c_\beta$, if the control is updated at the time determined iteratively with

$$\begin{aligned} t_{k+1} = \min_t \big\{ t > t_k \mid\ & x^T (rP - Q)x + 2x^T PBK(x_k - x) \\ & = c_\beta (V(x_0)\exp(-rt) - x^T P x) \big\}, \end{aligned} \tag{10}$$

then $V(x(t)) \leq V(x_0)\exp(-rt)$ is satisfied for all time. $\square$

Corollary 3.6 shows one simple application of Proposition 3.2. In its trigger design, it uses the exact terms of Lyapunov function and its time-derivative. In addition, it picks the simplest $\beta$ function which is a linear function. For this trigger design, our next result gives the minimum inter-event time.

*Proposition 3.7: (Minimum Inter-event Time for Linear Case):* Consider the linear system (9) with the control update at time given by the trigger design (10). Define the following time varying matrix,

$$\begin{aligned} M(\tau) = c_\beta P \exp(-r\tau) + G(\tau)^T ((c_\beta - r)P + Q) G(\tau) \\ + 2G(\tau)^T PBK(I_n - G(\tau)) \end{aligned} \tag{11}$$

where $G(\tau) = \exp(A\tau) + \int_0^\tau \exp(A(\tau - s)) ds BK$. Then, the inter-event time is lowered bounded as follows

$$t_{k+1} - t_k \geq \min \big\{ \tau > 0 \mid \det(M(\tau)) = 0 \big\}. \tag{12}$$

for all $k \in \mathbb{N} \cup 0$. $\square$

Proposition 3.7 provides a method to calculating the minimum inter-event time using our design for linear case with exponential performance. Note that we have an expression that only depends on time and no longer on the state, which implies that it can be calculated offline.

## IV. Flexibility for Distributed Implementation

In this section, we expand on the development above to applications in distributed settings. As will be shown, in addition to the aforementioned advantages in Remarks 3.3 and 3.4, we find that our design in distributed scenarios is less susceptible to Zeno behavior than traditional designs.

Before we dive into how our design can be implemented, we first review how event-triggered control in general might be applied in a distributed settings. Consider a network of agents whose interconnection is represented by a graph $G = ([n], \mathcal{E})$ with a closed loop networked system under the following dynamics,

$$\dot{x}_i = f_i(x_{\mathcal{N}_i}, k_i(x_{\mathcal{N}_i}, e^{(i)}_{\mathcal{N}_i})). \tag{13}$$

The above equation suggests that the dynamics of the states in each node $x_i$ depend on its neighboring nodes $j \in \mathcal{N}_i$, and the control is applied via state feedback controller. The errors $e^{(i)}_{\mathcal{N}_i} = x_{\mathcal{N}_i}(t_{k_i}) - x_{\mathcal{N}_i}(t)$ are introduced from not updating the controller continuously at node $i$. The event-triggered control problem seeks to solve how to efficiently update the controller at each node (resetting the errors to zero). To begin, we assume that the continuously controlled system is proven to be asymptotically stable by using the Lyapunov function with the following structure,

$$V(x) = \sum_i V_i(x_{\mathcal{N}_i}), \tag{14}$$

where each of the function $V_i$ are positive definite with respect to the sub-states $x_{\mathcal{N}_i}$. Here we allow the Lyapunov function to be distributed to the nodes. Then the time derivative of the Lyapunov function is given by

$$\begin{aligned} \frac{d}{dt} V(x) &= \sum_i \sum_{j \in \mathcal{N}_i} \nabla V_i(x_{\mathcal{N}_i}) f_j(x_{\mathcal{N}_j}, e^{(j)}_{\mathcal{N}_j}) \\ &= \sum_i \frac{dV_i}{dt}(x_{\mathcal{N}_i^2}, e^{(\mathcal{N}_i)}_{\mathcal{N}_i^2}). \end{aligned} \tag{15}$$

Note here that $f_j$ above can depend on the states of two-hop neighbors of node $i$, so in order to enable the ability to calculate the time derivative of $V_i$, we assume that the communication graph connects two-hop neighbors (two hops in dynamics but one hop in communication). In term of stabilization, it is sufficient to enforce

$$\frac{dV_i}{dt}(x_{\mathcal{N}_i^2}, e^{(\mathcal{N}_i)}_{\mathcal{N}_i^2}) = \sum_{j \in \mathcal{N}_i} \nabla V_i(x_{\mathcal{N}_i}) f_j(x_{\mathcal{N}_j}, e^{(j)}_{\mathcal{N}_j}) < 0$$

for each $i$ at all time. This is generally done by making sure that $\frac{dV_i}{dt}(t_{k_i}) < 0$ and tracking when the inequality above approaches equality. Note here the restrictiveness of this requirement. Whether or not this is possible depends largely on the structure of the network. Here in order to continue with our discussion, we generally state that such can be realized because each node tries to contribute in reducing the Lyapunov function. Then after proving that there is no Zeno behavior, the system is guaranteed asymptotically stability. What we have given thus far is a generic case of a networked system with assumptions we think are not far-fetched.

Next, we discuss the possibility of extending our algorithm from Proposition 3.2 for distributed implementation. To do so, we assume that

$$\frac{dV_i}{dt}(t_{k_i}) < -h(\bar{V}_i(t_{k_i})) \tag{16}$$

where $\bar{V}_i$ is an upper bound function to $V_i$. This is a stricter assumption than earlier. For this paper, we say that this is possible each node tries to contribute in reducing the Lyapunov function by a certain rate. Drawing parallel to the narrative of Section III, one can realize a specified performance requirement, in a similar manner to design (5), by determining the trigger time according to

$$t_{k_i+1} = \min_t \left\{ t > t_{k_i} \mid g_i(x_{\mathcal{N}_i^2}, e_{\mathcal{N}_i^2}^{(\mathcal{N}_i)}) + h(\bar{V}_i(x_{\mathcal{N}_i})) = 0 \right\}$$

if $h$ is a concave function in addition to being class-$\mathcal{K}$, and the function $g_i$ is an upper bound to $\frac{dV_i}{dt}$. However, such a design is too conservative for the same reason we discussed on Lemma 3.1. Instead, we can envision a less-conservative approach with the following

*Proposition 4.1: (Performance Barrier Design in Distributed Settings):* Consider the networked system (13). Assume the following:

1) the Lyapunov function and its time-derivative has the structure given in (14) and (15);
2) at the time of each update $t_{k_i}$, for each node $i$,
$$\frac{dV_i}{dt}(x_{\mathcal{N}_i}, 0) \le g_i(x_{\mathcal{N}_i}, 0) < -h(\bar{V}_i(x_{\mathcal{N}_i}));$$
3) each node $i$ can access the states of two-hops neighbors at all time;
4) the desired performance function can be distributed into the network $S(x_0, t) = \sum_i S_i(t)$ such that $S_i(t_{k_i}) \ge V_i(x_{\mathcal{N}_i}(t_{k_i}))$;
5) the inter-event time of the design presented below are lower bounded by $\tau > 0$.

Let the time at which node $i$ update its control be determined according to

$$\begin{aligned} t_{k_i+1} = \min_t \{ t > t_k \mid g_i(x_{\mathcal{N}_i^2}, e_{\mathcal{N}_i^2}^{(\mathcal{N}_i)}) + h(\bar{V}_i(x_{\mathcal{N}_i})) \\ = \beta(S_i(t) - \bar{V}_i(x_{\mathcal{N}_i})) \} \end{aligned} \tag{17}$$

with a concave, class-$\mathcal{K}$ function $h$, and a convex, class-$\mathcal{K}$ function $\beta$. Then, $V(x(t)) \le S(x_0, t)$ at all time. □

Proposition 4.1 shows that the concept of performance barrier presented in this paper can be applied to distributed scenarios under a set of assumptions. The first three assumptions are also shared with traditional event-triggered designs, as explained earlier in the section. Assumption 4 on the list can be easily satisfied by breaking down the performance function proportionally to the initial breakdown of the Lyapunov function, i.e. $S_i(t) = \frac{V_i(x_0)}{V(x_0)} S(x_0, t)$. Assumption 5 arises from the observation that a decentralized implementation of an event-triggered update can lead to Zeno behavior, cf. [16], [19]. We present next a sufficient condition guaranteeing the existence of a Zeno-free performance barrier design.

*Proposition 4.2: (Existence of Non-Zeno Design):* Consider the networked system (13) with the first four assumptions given by Proposition 4.1. Let $t_{k_i}$ be the last time at which the control is updated at node $i$. For each iteration $k_i$ at node $i$, define the latest time at which $V_i$ does not violating the performance specification

$$t_{k_i}^* = \min_t \left\{ t > t_{k_i} \mid 0 = S_i(t) - V_i(x_{\mathcal{N}_i}(t)) \right\}. \tag{18}$$

If there exists a positive constant bound $\tau_{\max} \le t_{k_i}^* - t_{k_i}$, then there exists a convex, class-$\mathcal{K}$ function $\beta$ such that the inter-event times for the design (17) with $\bar{V}(x) = V(x)$ are lower bounded by a positive constant $\eta\tau_{\max}$ for any $\eta \in (0, 1)$. □

Proposition 4.2 shows that there exists a performance barrier design that satisfies the fifth assumption in Proposition 4.1 if the inter-event time to reach the condition in (18) is lower-bounded. The lower-boundedness of such a condition is a necessary but not sufficient condition for proving minimal inter-event time for a derivative-based design. In contrast, if such a condition can be proven, we can hope to find a performance barrier type of design while there is no guarantee for a derivative-based design (this is illustrated through simulations in Section V).

## V. Simulation Example

Consider the following feedback-controlled linear system,

$$\dot{x} = \begin{bmatrix} 2 & 2 & 1 & 0 & 0 \\ -3 & -3 & 0 & 2 & 0 \\ 1 & 0 & -2 & 3 & 1 \\ 0 & 1 & 3 & -4 & 5 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix} x + u = Ax + u,$$

$$u = -\begin{bmatrix} 3 & 2 & 1 & 0 & 0 \\ -3 & 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 3 & 1 \\ 0 & 1 & 3 & 0 & 5 \\ 0 & 0 & 1 & -2 & 2 \end{bmatrix} x = Kx.$$

The closed-loop system is stable with the Lyapunov function $V(x) = \frac{1}{2}x^T x$ such that $Q = \text{diag}\begin{bmatrix} 1 & 3 & 2 & 4 & 1 \end{bmatrix}$ is positive definite. We can see that the minimum decay rate in the continuous implementation is given by minimum eigenvalue of $Q$ which is 1. Suppose that the desired performance on a discrete platform is to satisfy a decay rate of $r = 0.5$, i.e. $S(x_0, t) = V(x_0)\exp(-0.5t)$. Then we can use either the traditional trigger design (5) or the proposed performance barrier trigger design in this paper (10) with $c_\beta = 1$.

We simulate the system with the initial condition $x_0 = \begin{bmatrix} -4 & 2 & -2 & 1 & 3 \end{bmatrix}^T$. Figure 1 shows that both triggers satisfy the desired performance criterion. However, as predicted by Remark 3.3, the proposed design has longer inter-event times on average, cf. Figure 2, with an increase of minimum inter-event time from $0.2558$ to $0.4516$ for the duration of the simulation.

Next, we simulate the distributed case for the same system. We note here that first we attempted a derivative-based trigger design; however, the inter-event times get smaller

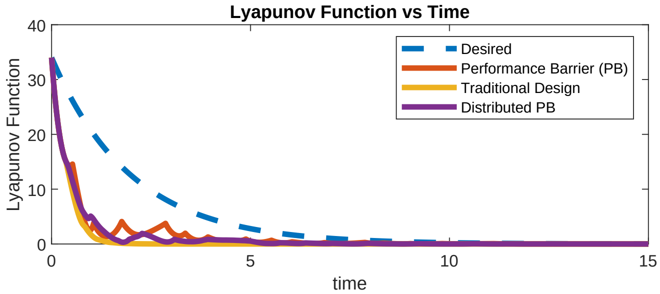


Fig. 1. Comparison of evolution of the Lyapunov function. In all cases, the desired performance is satisfied.

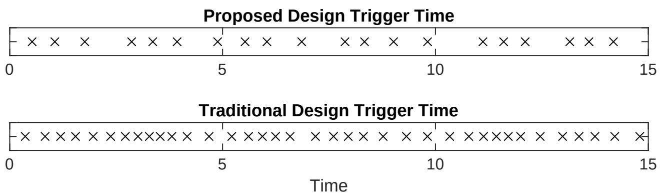


Fig. 2. Comparison of triggering times between proposed and tradditional trigger designs.

than 0.0001 in all nodes after a few iterations, suggesting the design suffers from Zeno behavior. Then we apply a performance barrier design based on (17) for this system. Figure 3 shows the time at which the trigger occurs for each node, marked by the symbol $\times$. The minimum inter-event time during the entire duration is on node $4$ with the value of $0.1055$, and we conclude from the simulation that the design is Zeno-free. This illustrates the idea presented in the discussion of Proposition 4.2.

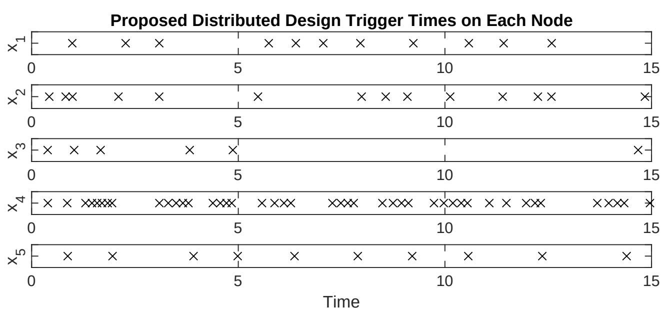


Fig. 3. Trigger times on each node using our proposed distributed design.

## VI. Conclusions

We have revisited the event-triggered control design problem, addressing the question of how to balance trigger frequency and performance by including the desired performance in the design. This has led us to propose a design that uses the desired performance as a *barrier* for the trigger. The resulting design has several advantages such as increased inter-event time, robustness to delays in the evaluation of the trigger, possibilities of avoiding Zeno-behavior in distributed scenarios, and amenability to distributed implementation. Future work will seek to characterize these advantages, seeking explicit expressions for the increase in inter-event time and the robustness margin, and exploring the design of $\beta$ functions in our trigger that ensures non-Zeno behavior. Lastly, we want to explore the idea of using the performance residual $S_i(x_0, t_k) - V_i(t_k)$ in a networked system. The idea is to allocate partially the performance residual to the neighbor nodes to reduce the triggering frequency.

## References


[1] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *IEEE Conf. on Decision and Control*, (Maui, HI), pp. 3270–3285, 2012.

[2] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.

[3] M. Abdelrahim, R. Postoyan, J. Daafouz, and D. Nešić, "Stabilization of nonlinear systems using event-triggered output feedback controllers," *IEEE Transactions on Automatic Control*, vol. 61, pp. 2682–2687, Sept 2016.

[4] W. P. M. H. Heemels, M. C. F. Donkers, and A. R. Teel, "Periodic event-triggered control based on state feedback," in *IEEE Conf. on Decision and Control*, pp. 2571–2576, Dec 2011.

[5] R. Postoyan, P. Tabuada, D. Nešić, and A. Anta, "A framework for the event-triggered stabilization of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 60, pp. 982–996, April 2015.

[6] B. A. Khashooei, D. J. Antunes, and W. P. M. H. Heemels, "Output-based event-triggered control with performance guarantees," *IEEE Transactions on Automatic Control*, vol. 62, pp. 3646–3652, July 2017.

[7] M. Velasco, P. Martí, and E. Bini, "On Lyapunov sampling for event-driven controllers," in *IEEE Conf. on Decision and Control*, pp. 6238–6243, Dec 2009.

[8] S. Durand, N. Marchand, and J. F. Guerrero-Castellanos, "Simple Lyapunov sampling for event-driven control," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 8724–8730, 2011.

[9] M. Mazo Jr., A. Anta, and P. Tabuada, "On self-triggered control for linear systems: Guarantees and complexity," in *European Control Conference*, (Budapest, Hungary), pp. 3767–3772, Aug. 2009.

[10] A. Seuret and C. Prieur, "Event-triggered sampling algorithms based on a Lyapunov function," in *IEEE Conf. on Decision and Control*, pp. 6128–6133, Dec 2011.

[11] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007.

[12] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[13] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*. Birkhäuser Boston, 2007.

[14] X. Wang and M. D. Lemmon, "Event-triggering in distributed networked control systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 586–601, 2011.

[15] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1291–1297, 2012.

[16] P. Tallapragada and N. Chopra, "Decentralized event-triggering for control of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 59, pp. 3312–3324, Dec 2014.

[17] C. Nowzari and J. Cortés, "Distributed event-triggered coordination for average consensus on weight-balanced digraphs," *Automatica*, vol. 68, pp. 237–244, 2016.

[18] C. Nowzari and J. Cortés, "Team-triggered coordination for real-time control of networked cyberphysical systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 1, pp. 34–47, 2016.

[19] M. C. F. Donkers and W. P. M. H. Heemels, "Output-based event-triggered control with guaranteed $L_\infty$-gain and improved and decentralised event-triggering," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1362–1376, 2012.

[20] H. K. Khalil, *Nonlinear Systems*. Prentice Hall, 3 ed., 2002.