Robust Dynamic Event-Triggered Coordination With a Designable Minimum Inter-Event Time

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at George Mason University

By

James Berneburg
Bachelor of Science
George Mason University, 2017

Director: Dr. Cameron Nowzari, Professor
Department of ECE Department

Fall Semester 2020
George Mason University
Fairfax, VA

# Table of Contents

# List of Tables

# List of Figures

# Abstract

ROBUST DYNAMIC EVENT-TRIGGERED COORDINATION WITH A DESIGNABLE MINIMUM INTER-EVENT TIME

James Berneburg

George Mason University, 2020

Thesis Director: Dr. Cameron Nowzari

This paper revisits the classical multi-agent average consensus problem for which many different event-triggered control strategies have been proposed over the last decade. Many of the earliest versions of these works conclude asymptotic stability without proving that Zeno behavior, or deadlocks, do not occur along the trajectories of the system. More recent works that resolve this issue either: (i) propose the use of a dwell-time that forces inter-event times to be lower-bounded away from zero but sacrifice asymptotic convergence in exchange for practical convergence (or convergence to a neighborhood); (ii) guarantee non-Zeno behaviors and asymptotic convergence but do not provide a positive minimum inter-event time guarantee; or (iii) are not fully distributed.

Additionally, the overwhelming majority of these works provide no form of robustness analysis on the event-triggering strategy. More specifically, if arbitrarily small disturbances can remove the non-Zeno property then the theoretically correct algorithm may not actually be implementable.

Instead, this work for the first time presents a fully distributed, robust, dynamic event-triggered algorithm, for general directed communication networks, for which a desired positive minimum inter-event time can be chosen by each agent in a distributed fashion. Simulations illustrate our results.

# Chapter 1: Introduction

Systems composed of individually controlled agents are increasingly common and a very active area of research. Such systems are designed for many different applications including the coordination of unmanned air vehicles, distributed reconfigurable sensor networks, and attitude alignment for satellites, etc; see [1] and [2] and their references. These are often intended to fulfill some coordinated task, but require distributed control to be scaled with large systems. In this case, communication limitations, such as wireless bandwidth, mean that agents cannot be assumed to have continuous access to others' states. Therefore, many works have recently considered communication to be a limited resource, where individual agents must autonomously schedule when to take various actions, rather than doing so periodically or continuously.

A common solution to these types of problems comes in the form of event-triggered coordination, where actions occur at specific instances of time when some event condition is satisfied, such as when an error state [3] or a clock state [4] hits some threshold. A similar strategy is self-triggered control, where the controller uses state information to schedule events ahead of time. An introduction to these ideas for single-plant systems is found in [5].

One potential problem in event-triggered coordination is the Zeno phenomenon, where the number of events triggered goes to infinity in a finite time period. This is problematic as it asks for solutions that cannot be realized by actual devices. A way to prevent this problem is to design triggering conditions that guarantee a positive minimum inter-event time (MIET) exists. Note that this is different from first designing an event-triggering condition, and then afterwards *forcing* a minimum inter-event time (dwell-time), which can negatively affect convergence guarantees. For example, the self-triggered strategy in [6] enforces a MIET and so only guarantees convergence to a set.

As noted in [5], the existence of a positive MIET is important to ensure that the event-triggering mechanism does not become unimplementable because it requires actions to be taken arbitrarily fast. This issue has been addressed recently for single-plant systems, e.g., in [7], where a general method for achieving stabilization based on Lyapunov functions is developed, and [8], where a general framework for event-triggering mechanisms is provided using the hybrid systems formalism of [9]. Hybrid systems formulations seem especially useful in networked control systems because they conveniently describe systems with continuous-time dynamics and discrete-time memory updates via communication. However, this is still a major challenge for multi-agent systems with distributed information.

To address this, we turn to a simple but widely applicable canonical problem: multi-agent average consensus. Consensus problems are when multiple agents, each with its own dynamics and limited access to the other agents' states, are intended to be stabilized such that all the agents' states are equal. Applications include distributed computing, networks of sensors, flocking, rendezvous, attitude alignment, and synchronization of coupled oscillators; see [10], [2] and [1] and their references.

Event-triggered strategies for consensus problems have been studied quite extensively over the last decade, with some of the earliest works appearing in 2009 [11–13]. We refer to [14] for a detailed survey on the history of this problem but summarize the relevant points next.

A seminal work on this topic is [15], which develops centralized event- and self-triggered strategies that lead to multi-agent consensus, and then modifies them to be distributed. Unfortunately, although the centralized event-triggered strategy is able to guarantee a positive MIET, the distributed strategies are unable to guarantee the prevention of Zeno solutions. Similarly, the results in [16] are unable to exclude Zeno behavior.

Some works have addressed this issue by considering a periodically sampled (or sampled-data) implementation to trivially address this issue, but this assumes perfect synchronization among the entire network which is neither practical or scalable [17–20]. More recent works have even considered asynchronous periodic implementations but these require some

sort of global knowledge to find periods that will work [21, 22].

More related to our work, [3, 23, 24] present distributed event-triggered strategies for the consensus problem that prevent Zeno solutions and ensure convergence to consensus. The first two include an explicit function of time in the trigger mechanism, while the third uses a dynamic triggering mechanism, by including a virtual state. While these are a good start, unfortunately none of these can guarantee a positive MIET for the agents, which is our main goal.

The distributed event-triggered strategy in [25] is able to guarantee convergence to consensus with a positive MIET enforced; however, it requires global parameters in order to design each agent's controller, so it is not fully distributed. Alternatively, and most similarly to the methods used in this work, the authors of [4] utilize a hybrid systems formulation to solve a closely related problem in which a different communication model is considered. In their work they show the existence of a positive MIET for a fully distributed event-triggered strategy that guarantees asymptotic convergence with an event trigger that employs a dynamic virtual state; however, this work still requires a type of synchronization as agents need to trigger events in pairs.

Finally, another important consideration is the robustness of a MIET. We note here that we are primarily concerned with the robustness of the event-triggering strategy rather than robustness in terms of feedback stabilization. The authors of [26] acutely point out that, even if an event-triggered controller may guarantee a positive MIET, it is possible that arbitrarily small disturbances remove this property which means it is still equally useless for implementing on physical systems. Another potential issue for event-triggered strategies is robustness to imperfect event detection. More specifically, analysis on event-based solutions often rely on the very precise timings of actions in response to events. Consequently, we are also interested in designing a solution that is robust to small timing errors in determining when event conditions have been satisfied.

*Statement of contributions:* This problem revisits a simple single-integrator multi-agent average consensus problem originally conceived in [11–13]. By re-formulating the problem

using hybrid systems, we are able to provide a novel algorithm with several key fundamental improvements over similar event-triggered strategies in the literature. The contributions of this paper are threefold. First, we provide the first known fully distributed solution that guarantees a positive MIET. Second, we develop a method to design the triggering functions such that each agent is able to independently prescribe their guaranteed MIET in a distributed way; which has important implications on implementability of the proposed algorithms in real applications. Finally, we investigate the robustness of the MIET against both imperfect event detection, for which we provide an alternative robust trigger, and additive state disturbances, discussing its effects. Simulations illustrate our results.

# Chapter 2: Mathematical Problem

## 2.1 Preliminaries

The Euclidean norm of a vector $v \in \mathbb{R}^n$ is denoted by $||v||$. An n-dimensional column vector with every entry equal to 1 is denoted by $\mathbf{1}_n$, and an n-dimensional column vector with every entry equal to 0 is denoted by $\mathbf{0}_n$. The minimum eigenvalue of a square matrix $A$ is given by eigmin($A$) and its maximum eigenvalue is given by eigmax($A$). The cardinality of a finite set $\mathcal{N}$ is denoted by $|\mathcal{N}|$. Given a vector $v \in \mathbb{R}^N$, we denote by diag($v$) the $N \times N$ diagonal matrix with the entries of $v$ along its diagonal.

Young's inequality is

$$xy \leq \frac{a}{2}x^2 + \frac{1}{2a}y^2, \tag{2.1}$$

for $a > 0$ and $x, y \in \mathbb{R}$ [27].

By $V^{-1}(C)$ where $C \subset \mathbb{R}^m$, we denote the set of points $\{s \in \mathbb{R}^n : V(s) \in C\}$, for a function $V : \mathbb{R}^n \to \mathbb{R}^m$. By $\mathbb{R}_{\geq 0}$ we denote the set of nonnegative real numbers, and by $\mathbb{Z}_{\geq 0}$ we denote the set of nonnegative integers. The closure of a set $U \in \mathbb{R}^n$ is denoted by $\overline{U}$. The domain of a mapping $f$ is denoted by dom $f$ and its range is denoted by range $f$.

**Graph Theory** An weighted graph $\mathcal{G} = (V, \mathcal{E}, A)$ has a set of vertices $V = \{1, 2, ..., N\}$, a set of edges $\mathcal{E} \subset V \times V$, and an adjacency matrix $A \in \mathbb{R}^{N \times N}$ with each entry $a_{ij} \in \mathbb{R}_{\geq 0}$, where $a_{ij} > 0$ if $(i, j) \in \mathcal{E}$, and $a_{ij} = 0$ otherwise. For a digraph (directed graph), edge $(i, j)$ is distinct from edge $(j, i)$. A path between vertex $i$ and vertex $j$ is a finite sequence of edges $(i, k)$, $(k, l)$, $(l, m)$, ..., $(n, j)$. A digraph is strongly connected if there exists a path between any two vertices. For an edge $(i, j)$, $j$ is an out neighbor of $i$ and $i$ is an in neighbor

Table 2.1: Agent $i$ model definitions.

| Definition | Domain |
|---|---|
| $q_i = [x_i, \widehat{x}_i, \chi_i]^T$ | $\in \mathbb{R}^3$ |
| $v_i = (q_i, \{\widehat{x}_j\}_{j \in \mathcal{N}_i^{\text{out}}})$ | $\in \mathbb{R}^3 \times \mathbb{R}^{\lvert\mathcal{N}_i^{\text{out}}\rvert}$ |
| $e_i = x_i - \widehat{x}_i$ | $\in \mathbb{R}$ |
| $\widehat{z}_i = (L\widehat{x})_i = \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij}(\widehat{x}_i - \widehat{x}_j)$ | $\in \mathbb{R}$ |
| $\widehat{\phi}_i = \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij}(\widehat{x}_i - \widehat{x}_j)^2$ | $\in \mathbb{R}_{\geq 0}$ |

of $j$. A weighted digraph has a weighted adjacency matrix $A$ where the $ij$th element is the weight for edge $(i, j)$. The in-degree, $d_i^{\text{in}}$, for a vertex $i$ is the sum of all the weights for the edges that correspond to its in neighbors, and the out-degree, $d_i^{\text{out}}$, is the same for its out neighbors. A weight-balanced digraph is a digraph where $d_i^{\text{in}} = d_i^{\text{out}} = d_i$ for each vertex $i$. For a weight-balanced digraph, the degree matrix $D^{\text{out}} = D^{\text{in}}$ is a diagonal matrix with $d_i$ as the $i$th diagonal element, and the Laplacian is $L = D^{\text{out}} - A$.

**Hybrid Systems** A hybrid system $\mathcal{H} = (C, f, D, G)$ is a tuple composed of a flow set $C \in \mathbb{R}^n$, where the system state $x \in \mathbb{R}^n$ continuously changes according to $\dot{x} = f(x)$, and a jump set $D \in \mathbb{R}^n$, where $x$ discretely jumps to $x^+ \in G(x)$, where $f$ maps $\mathbb{R}^n \to \mathbb{R}^n$ and $G : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is set valued [9, Definition 2.2]. While $x \in C$, the system can flow continuously and while $x \in D$, the system can jump discontinuously.

A compact hybrid time domain is a subset $E_{\text{compact}} \subset \mathbb{R} \times \mathbb{N}$ for which $E_{\text{compact}} = \cup_{j=0}^{J-1}([t_j, t_{j+1}], j)$, for a finite sequence of times $0 \leq t_0 \leq t_1 \leq ... \leq t_J$, and a hybrid time domain is a subset $E \subset \mathbb{R} \times \mathbb{N}$ such that $\forall (T, J) \in E$, $E \cap ([0, T] \times \{0, 1, ...J\})$ is a compact hybrid time domain [9, Definition 2.3]. The hybrid time domain is used to keep track of both the elapsed continuous time $t$ and the number of discontinuous jumps $j$.

## 2.2 Problem Formulation

We begin by stating a long-standing version of the event-triggered consensus problem. We then show why existing solutions to it are not pragmatic and how we reformulate the problem to obtain solutions that can be implemented on physical platforms. Please see Table 2.1 for a summary of notation used in the following.

Consider a group of $N$ agents whose communication topology is described by a directed, weight-balanced, and strongly connected graph $\mathcal{G}$ with edges $\mathcal{E}$ and Laplacian matrix $L$. Each agent is able to receive information from its out neighbors and send information to its in neighbors, and each weight of the graph is a gain applied to the information sent from one agent to another.

The state of each agent $i$ at time $t \geq 0$ is given by $x_i(t)$ with single-integrator dynamics

$$\dot{x}_i(t) = u_i(t), \tag{2.2}$$

where $u_i$ is agent $i$'s input. It is well known that the input

$$u_i(t) = -\sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij}(x_i(t) - x_j(t)) \tag{2.3}$$

drives all agent states to the average of the initial conditions [1], which is defined as

$$\bar{x} \triangleq \frac{1}{N}\sum_{i=1}^{N} x_i(0).$$

Note that under the control law (2.3), the average $\bar{x}(t)$ is an invariant quantity. Defining $x = [x_1, x_2, \ldots, x_N]^T$ and $u = [u_1, u_2, \ldots, u_N]^T$ as the vectors containing all the state and input information about the network of agents, respectively, we can describe all inputs together by

$$u(t) = -Lx(t).$$

However, in order to implement this control law, each agent must have continuous access to the state of each of its out neighbors. Instead, we assume that each agent $i$ can only measure its own state $x_i$ and must receive neighboring state information through wireless communication. We consider event-triggered communication and control where each agent only broadcasts its state to its neighbors at discrete instances of time. More formally, letting $\{t_\ell^i\}_{\ell\in\mathbb{Z}_{\geq 0}} \subset \mathbb{R}_{\geq 0}$ be the sequence of times at which agent $i$ broadcasts its state to its in neighbors $j \in \mathcal{N}_i^{\text{in}}$, the agents instead implement the control law

$$u(t) = -\widehat{z} \triangleq -L\widehat{x}(t), \tag{2.4}$$

where $\widehat{x} = [\widehat{x}_1, \ldots, \widehat{x}_N]^T$ is the vector of the last broadcast state of each agent. More specifically, given the sequence of broadcast times $\{t_\ell^i\}_{\ell\in\mathbb{Z}_{\geq 0}}$ for agent $i$, we have

$$\widehat{x}_i(t) = x_i(t_\ell^i) \quad \text{for} \quad t \in [t_\ell^i, t_{\ell+1}^i). \tag{2.5}$$

Note that the input (2.4) still ensures that the average of all agent states is an invariant quantity because $\dot{\bar{x}} = \frac{1}{N}\mathbf{1}_N^T\dot{x} = \frac{1}{N}\mathbf{1}_N^T(-L\widehat{x}) = 0$, which follows from the weight-balanced property of the graph.

At any given time $t \geq 0$, we define

$$v_i(t) \triangleq (x_i(t), \widehat{x}_i(t), \{\widehat{x}_j(t)\}_{j\in\mathcal{N}_i^{\text{out}}}) \tag{2.6}$$

as all the dynamic variables locally available to agent $i$. The problem of interest, formalized below, is then to obtain a triggering condition based on this information such that the sequence of broadcasting times $\{t_\ell^i\}_{\ell\in\mathbb{Z}_{\geq 0}}$, for each agent $i$, guarantees that the system eventually reaches the average consensus state.

**Problem 1. (Distributed Event-Triggered Consensus)** Given the directed, weight-balanced, and strongly connected graph $\mathcal{G}$ with dynamics (2.2) and input (2.4), find a

triggering condition for agent $i$, which depends only on locally available information $v_i$, such that $x_i \to \bar{x}$ for all $i \in \{1, \ldots, N\}$.

This problem was first formulated in [11,12] in 2009. Since then, there have been many works dedicated to this problem in both the undirected [3,15–17,23,28] and directed [18,19, 29] cases. Unfortunately, although the above referenced papers provide theoretical solutions to this problem, we are not aware of a single solution which can be implemented on physical systems some practical concerns, described next, are considered. For details on the history of this problem and its theoretical solutions, we refer the interested reader to [14], but we summarize the main points here.

The earliest solutions to this problem did not adequately investigate the Zeno phenomenon, which invalidates their correctness [15,16]. In particular, these solutions to Problem 1 did not rule out the possibility of Zeno behavior, meaning that it was possible for a sequence of broadcasting times $\{t_\ell\}_{\ell \in \mathbb{Z}_{\geq 0}}$ to converge to some finite time $t_\ell \to T > 0$. This is clearly troublesome since all theoretical analysis then falls apart after $t > T$, invalidating the asymptotic convergence results. More recently, the community has acknowledged the importance of ruling out Zeno behavior to guarantee that all the sequences of times $t_\ell^i \to \infty$ as $\ell \to \infty$ for all $i \in \{1, \ldots, N\}$. While enforcing this additional constraint on the sequences of broadcasting times guarantees that *theoretically* the solutions will converge to the average consensus state, we must consider other important practical issues as well.

Even guaranteeing that Zeno behavior does not occur, so that the inter-event times are strictly positive for all agents $i$,

$$t_{\ell+1}^i - t_\ell^i > 0,$$

is unfortunately still not enough to guarantee that the solution can be realized by physical devices. This is because, although the inter-event times are technically positive, they can become arbitrarily small to the point that no physical hardware exists that can keep up with the speed of actions required by the event-triggered algorithm. The solutions in [23], [24], [3],

and [29] have this problem. This is inherently different from guaranteeing a strictly positive MIET $T$, where $t^i_{\ell+1} - t^i_\ell \geq T > 0$, which is the focus of our work here.

Specifically, we consider the case where each agent $i \in \{1, \ldots, N\}$ has some maximum rate $\frac{1}{\tau_i}$ at which it can take actions (e.g., broadcasting information, computing control inputs). Therefore, each agent $i$ cannot process two events in less than $\tau_i$ seconds, which is written mathematically as

$$t^i_{\ell+1} - t^i_\ell \geq \tau_i,$$

for all $\ell \in \mathbb{Z}_{\geq 0}$. Note that there are also solutions that guarantee a MIET, but make other sacrifices to do so. For example, the solution in [23] is able to guarantee a MIET under certain conditions, but convergence is only to a neighborhood of consensus. Additionally, the algorithms in [20, 25] are able to enforce a MIET, but only by using global parameters of the system to design the algorithm, which is impractical in cases where the parameters may change or are otherwise difficult to measure. The algorithm in [4] is fully distributed and has a positive MIET, but it still requires pair of agents to trigger events at the same time, which necessitates synchronization. With all this in mind, we reformulate Problem 1 such that solutions to the problem can be implemented on physical platforms given that each agent $i$ is capable of processing actions at a frequency of up to $\frac{1}{\tau_i}$.

**Problem 2. (Distributed Event Triggered Consensus with Designable MIET)** Given the directed, weight-balanced, and strongly connected graph $\mathcal{G}$ with dynamics (2.2), input (2.4), and the minimum periods $(\tau_1, \ldots, \tau_N)$ for each agent, find a triggering condition for each agent $i$, which depends only on local information $v_i$, such that $x_i \to \bar{x}$ and

$$\min_{\ell \in \mathbb{Z}_{\geq 0}} t^i_{\ell+1} - t^i_\ell \geq \tau_i, \tag{2.7}$$

for all $i \in \{1, \ldots, N\}$.

To the best of our knowledge, Problem 2 has not yet been fully solved. Rather than being

able to guarantee a strictly positive minimum inter-event time, similar works often settle for only ruling out Zeno executions, meaning communication between agents may still need to occur arbitrarily fast in order for the convergence results to hold. Conversely, other works more simply force a MIET through the use of a dwell-time at the cost of losing the exact asymptotic convergence guarantee in exchange for practical consensus. The works [3,23,24] only preclude Zeno behavior but cannot guarantee that (2.7) holds even for arbitrarily small minimum periods $\tau_i$. The algorithm proposed in [25] comes close but requires global system information. The methodology used in [4] is similar to ours here but ultimately solves a different problem. Thus, we provide the first complete solution to Problem 2. For now, we consider no state disturbances and perfect event detection, but we will relax these assumptions in Section 3.2, where we study the robustness of the event-triggering strategy with respect to various forms of disturbances.

### 2.2.1 Hybrid Systems Formulation

In order to solve Problem 2, we first reformulate it using hybrid systems tools, similar to [4]. We refer to the original state $x$ as the 'physical' state that represents the actual state that we wish to control. Separately, we maintain a set of 'cyber' or virtual states corresponding to the internal memory of each agent. Given the communication model described by (2.5), it seems natural to keep track of the last broadcast state $\widehat{x}_i$ for each agent as one of the cyber states. Additionally, we introduce an extra virtual state $\chi_i$ for each agent $i \in \{1, \ldots, N\}$ to introduce dynamics into our triggering strategy, and we collect these components in the vector $\chi = [\chi_1, \chi_2, \ldots, \chi_N]^T$. Note that the internal variable $\chi_i$ is only available to agent $i$. In this work we consider scalar internal variables $\chi_i \in \mathbb{R}$ but note that more sophisticated controllers or even learning-based controllers could be captured by increasing the complexity of the internal variables. The hybrid systems formulation will aid us here in properly modeling both the continuous-time dynamical system with discrete-time memory and control updates.

We now define the extended state vector for agent $i$ as

$$q_i(t) = \begin{bmatrix} x_i(t) \\ \widehat{x}_i(t) \\ \chi_i(t) \end{bmatrix}, \tag{2.8}$$

and the extended state (capturing both 'physical' and 'cyber' states) of the entire system is $q = [q_1^T, q_2^T, \ldots, q_N^T]^T \in \mathbb{R}^{3N}$. With a slight abuse of notation, we now redefine the local information (2.6) available to agent $i$ at any given time as its own extended state $q_i$ and the last broadcast states of its out-neighbors $\{\widehat{x}_j\}_{j\in\mathcal{N}_i^{\text{out}}}$,

$$v_i(t) \triangleq (q_i(t), \{\widehat{x}_j(t)\}_{j\in\mathcal{N}_i^{\text{out}}}). \tag{2.9}$$

In this work, the goal is to use the internal variable $\chi_i$ to determine exactly when agent $i$ should broadcast its current state to its neighbors. To achieve this, we let this state take values $\chi_i \geq 0$, and prescribe an event trigger whenever $\chi_i = 0$. The exact dynamics of $\chi_i$ will be designed in Section 2.3 to guarantee a solution to Problem 2. More specifically, at any given time $t \geq t_\ell^i$, the next triggering time $t_{\ell+1}^i$ is given by

$$t_{\ell+1}^i = \inf\{t \geq t_\ell^i : \chi_i(t) = 0 \text{ and } \widehat{x}_i \neq x_i\}, \tag{2.10}$$

for all $\ell \in \mathbb{Z}_{\geq 0}$, for each agent $i$.

With the role of the new internal variable $\chi_i$ established (although its dynamics will be designed later), we can now formalize our hybrid system

$$\mathcal{H} = (C, f, D, G). \tag{2.11}$$

We refer to [9] for formal hybrid systems concepts and assume the reader is familiar with

this formalism.

**Flow set** (between event-triggering times): The flow set $C$ for the entire system is given by

$$C = \{q \in \mathbb{R}^{3N} : \chi_i \geq 0 \text{ for all } i \in \{1, \ldots, N\}\}. \tag{2.12}$$

While the system state $q \in C$, the system flows according to $f$

$$\dot{q} = f(q) = \begin{bmatrix} f_1(v_1) \\ \vdots \\ f_N(v_N) \end{bmatrix} \quad \text{for } q \in C, \tag{2.13}$$

with the individual extended states evolving according to

$$\dot{q}_i = f_i(v_i) \triangleq \begin{bmatrix} -\widehat{z}_i \\ 0 \\ \gamma_i(v_i) \end{bmatrix}, \tag{2.14}$$

where $\gamma_i$ is the function to be designed.

The first row is exactly $\dot{x}_i = u_i$ as defined in (2.4), the second row says the last broadcast state is not changing between event-triggers $\dot{\widehat{x}}_i = 0$, and the last row is the dynamics of the internal variable $\dot{\chi}_i = \gamma_i(v_i)$. Recall that $v_i$ defined in (2.9) only contains information available to agent $i$ and thus the dynamics of the internal variable $\chi_i = \gamma_i(v_i)$ must be a function only of $v_i$.

**Jump set** (at event-triggering times): The jump set $D$ for the entire system is then given by

$$D = \cup_{i=1}^{N} \{q \in \mathbb{R}^{3N} : \chi_i \leq 0\}. \tag{2.15}$$

Although in a general hybrid system there may be no notion of distributed information, since in this work the jumps correspond to *some* agent triggering an event, we formally define

$$D_i \triangleq \{q \in \mathbb{R}^{3N} : \chi_i \leq 0\} \tag{2.16}$$

as the subset of $D$ corresponding to when specifically agent $i$ is responsible for the jump. For $q \in D_i$, we consider the following local jump map

$$g_i(q) = \begin{bmatrix} q_1^+ \\ \vdots \\ q_i^+ \\ \vdots \\ q_N^+ \end{bmatrix} \triangleq \begin{bmatrix} q_1 \\ \vdots \\ \begin{pmatrix} x_i \\ x_i \\ \chi_i \end{pmatrix} \\ \vdots \\ q_N \end{bmatrix}. \tag{2.17}$$

More specifically, letting $t_\ell^i$ be the time at which agent $i$ triggers its $\ell$th event $q(t_\ell^i) \in D_i$, this map leaves the physical state and the dynamic variable unchanged $x_i^+ = x_i, \chi_i^+ = \chi_i$, and updates its "last broadcast state" to its current state $\widehat{x}_i^+ = x_i$. Note also that this leaves all other agents' states unchanged $q_j^+ = q_j$ for all $j \neq i$.

Now, since multiple agents may trigger events at once, the jump map must be described by a set-valued map $G : \mathbb{R}^{3N} \rightrightarrows \mathbb{R}^{3N}$ [4, 30], where

$$G(q) \in \{\dots, g_i(q), \dots\}, \tag{2.18}$$

for all $i$ such that $q \in D_i$. Note that this construction of the jump map ensures that it is outer-semicontinuous, which is a requirement for some hybrid systems results.

Now, we reformulate Problem 2 in a more structured manner by using the hybrid system (2.11). More specifically, by using this formulation we have formalized the objective of finding a local triggering strategy to designing the function $\gamma_i$ that depends only on the local information $v_i$ defined in (2.9).

**Problem 3. (Distributed Event-Triggered Consensus with Designable MIET)** Given the directed, weight-balanced, and strongly connected graph $\mathcal{G}$ with dynamics (2.2), input (2.4), and the minimum periods $(\tau_1, \ldots, \tau_N)$ for each agent, find the dynamics of the virtual state, $\gamma_i(v_i)$, such that $x_i \to \bar{x}$ and

$$\min_{\ell \in \mathbb{Z}_{\geq 0}} t^i_{\ell+1} - t^i_\ell \geq \tau_i,$$

for all $i \in \{1, \ldots, N\}$.

## 2.3 Dynamic Event-Triggered Algorithm Design

In order to solve Problem 3, we perform a Lyapunov analysis to design $\gamma \triangleq \dot{\chi}$, the dynamics of $\chi$. Inspired by [4], we use a Lyapunov function with two components: $V_P$ represents the physical aspects of the system, while $V_C$ represents the cyber aspects, related to communication and error. For convenience, let $e \triangleq x - \widehat{x}$ denote the vector containing the error for each agent's state, which is the difference between the actual state and the last broadcast state. We consider

$$V_P(q) = (x - \bar{x}\mathbf{1}_N)^T(x - \bar{x}\mathbf{1}_N) = ||x - \bar{x}\mathbf{1}_N||^2$$

$$V_C(q) = \sum_{i=1}^{N} \chi_i.$$

Note $V_C \geq 0$ because $\chi_i \geq 0$ for all $i \in \{1, \ldots, N\}$. We then consider the Lyapunov function

$$V(q) = V_P(q) + V_C(q). \tag{2.19}$$

Note that $V(q) \geq 0$ is continuously differentiable for all $q \in \mathbb{R}^{3N}$. Moreover, $V(q) = 0$ when all agents have reached their target state and each clock-like variable $\chi_i$ is equal to 0.

Now we will examine the evolution of $V$ along the trajectories of our algorithm to see under what conditions it is nonincreasing, and design $\gamma$ accordingly. In order to do so, we will have to split $\dot{V}$ into components $\dot{V}_i$ such that $\dot{V} = \sum_{i=1}^{N} \dot{V}_i$ and each $\dot{V}_i$ depends only on the local information $v_i$ available to agent $i$. Choosing $V$ properly to ensure that $\dot{V}$ can be split like this is essential to designing $\gamma_i$ and doing so is nontrivial. Recalling our system flow dynamics (2.13), we write

$$\dot{V}_P = -2(x - \bar{x}\mathbf{1}_N)^T \widehat{z}$$

$$\dot{V}_C = \sum_{i=1}^{N} \gamma_i.$$

Because the graph is weight-balanced, $\bar{x}\mathbf{1}_N^T L = \mathbf{0}_N^T$. Therefore, $\dot{V}_P = -2x^T\widehat{z} = -2\widehat{x}^T\widehat{z} - 2e^T\widehat{z}$ and

$$\dot{V} = \dot{V}_P + \dot{V}_C = -2\widehat{x}^T\widehat{z} - 2e^T\widehat{z} + \sum_{i=1}^{N} \gamma_i.$$

Expanding this out and using notation defined in Table 2.1 yields

$$\dot{V} = \sum_{i=1}^{N} \left( - \sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij} (\widehat{x}_i - \widehat{x}_j)^2 - 2 e_i \widehat{z}_i + \gamma_i \right) \tag{2.20}$$

$$\dot{V} = \sum_{i=1}^{N} \dot{V}_i = \sum_{i=1}^{N} \left( -\widehat{\phi}_i - 2 e_i \widehat{z}_i + \gamma_i \right).$$

We are now interested in designing $\gamma_i$ for each agent $i \in \{1, \ldots, N\}$ such that $\dot{V}(q) \leq 0$. Therefore, we choose

$$\gamma_i = \sigma_i \widehat{\phi}_i + 2 e_i \widehat{z}_i, \tag{2.21}$$

where $\sigma_i \in (0, 1)$ is a design parameter. Note that whenever an agent $i$ triggers an event the error $e_i$ is immediately set to 0, and since $\dot{\chi}_i = \gamma_i \geq 0$ at these times we ensure that $\chi_i \geq 0$ at all times $t \geq 0$. We can now write the derivative of the Lyapunov function as

$$\dot{V} = \sum_{i=1}^{N} -(1 - \sigma_i) \widehat{\phi}_i \leq 0. \tag{2.22}$$

This choice of the clock-like dynamics $\gamma_i$ is continuous in $q_i$ for constant $\widehat{x}$ and ensures that $\dot{V} \leq 0$.

### Algorithm Synthesis

With the dynamics $\gamma_i$ of the clock-like variable $\chi_i$ defined for each agent $i \in \{1, \ldots, N\}$, we can now summarize all the components of our synthesized distributed dynamic event-triggered coordination algorithm and formally describe it from the viewpoint of a single agent.

Table 2.2: Distributed Dynamic Event-Triggered Coordination Algorithm.

Initialization; at time $t = 0$ each agent $i \in \{1, \dots, N\}$ performs:

1: Initialize $\widehat{x}_i = x_i$
2: Initialize $\chi_i = 0$

At all times $t$ each agent $i \in \{1, \dots, N\}$ performs:

1: **if** $\chi_i = 0$ and $e_i \neq 0$ **then**
2: set $\widehat{x}_i = x_i$ (broadcast state information to neighbors)
3: set $u_i = -\sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij}(\widehat{x}_i - \widehat{x}_j)$ (update control signal)
4: **else**
5: propagate $\chi_i$ according to its dynamics $\gamma_i$ in (2.21)
6: **end if**
7: **if** new information $\widehat{x}_k$ is received from some neighbor(s) $k \in \mathcal{N}_i^{\text{out}}$ **then**
8: update control signal $u_i = -\sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij}(\widehat{x}_i - \widehat{x}_j)$
9: **end if**

The control input at any given time $t \geq 0$ is

$$u_i(t) = -\widehat{z}_i(t) = -\sum_{j \in \mathcal{N}_i^{\text{out}}} w_{ij}(\widehat{x}_i(t) - \widehat{x}_j(t)).$$

The sequence of event times $\{t_\ell^i\}_{\ell \in \mathbb{Z}_{\geq 0}}$ at which agent $i$ broadcasts its state to its neighbors is the time the clock-like variable reaches zero when that agent's error is nonzero, i.e.,

$$t_\ell^i = \inf\{t \geq t_{\ell-1}^i : \chi_i(t) = 0 \text{ and } e_i \neq 0\}.$$

The algorithm is formally presented in Table 2.2.

# Chapter 3: Results

## 3.1 Main Results

Here we present the main results of the paper by discussing the properties of our algorithm. We begin by finding the guaranteed positive minimum inter-event time (MIET) for each agent and showing how it can be tuned individually.

**Theorem 3.1.1** (Positive MIET)**.** Given the hybrid system $\mathcal{H}$, if each agent $i$ implements the distributed dynamic event-triggered coordination algorithm presented in Table 2.2 with $\sigma_i \in (0, 1)$, then the inter-event times for agent $i$ are lower-bounded by

$$T_i \triangleq \frac{\sigma_i}{d_i} > 0. \tag{3.1}$$

That is, $t^i_{\ell+1} - t^i_\ell \geq T_i$ for all $i \in \{1, \ldots, N\}$ and $\ell \in \mathbb{Z}_{\geq 0}$.

*Proof.* See the appendix. □

**Remark 3.1.2** (Design Trade off)**.** The design parameter $\sigma_i$ represents the trade off between larger inter-event times and faster convergence speeds. Larger $\sigma_i$ makes the magnitude of $\dot{V}$ smaller (2.22). However, it is a coefficient of the nonnegative term in $\chi_i$'s dynamics (2.21), so larger $\sigma_i$ means longer inter-event times and a longer MIET (3.1). Additionally, note that the MIETs can be guaranteed up to a maximum of $\frac{1}{d_i}$.

Next, we present our main convergence result. To the best of our knowledge, this is the first work to design a fully distributed event-triggered communication and control algorithm that guarantees asymptotic convergence to the average consensus state with a lower bound on the agent-specific MIET that can be chosen by the designer.

**Theorem 3.1.3** (Global Asymptotic Convergence)**.** Given the hybrid system $\mathcal{H}$, if each agent $i$ implements the distributed dynamic event-triggered coordination algorithm presented in Table 2.2 with $\sigma_i \in (0, 1)$, then all trajectories of the system are guaranteed to asymptotically converge to the set

$$\{q : \widehat{\phi}_i = 0 \; \forall \; i\}.$$

*Proof.* See the appendix. □

**Remark 3.1.4** (Convergence)**.** From Theorem 3.1.3, the algorithm presented in Table 2.2 does not entirely solve Problem 3, because agents only converge to $\{q : \widehat{\phi}_i = 0 \; \forall \; i\}$. To have $x_i = \bar{x}$, we must have both $\phi_i = 0$ and $e_i = 0$, for each agent $i$. However, under the algorithm presented in Table 2.2, each agent $i$ will not broadcast if $\phi_i = 0$, even if it has nonzero error. Therefore, if the algorithm is modified with an additional trigger which guarantees that each agent $i$ will always broadcast again, eventually, when $e_i \neq 0$, then full convergence is guaranteed. A very simple way to do so is to set a maximum time $T^i_{\max} \geq T_i$ between events for each agent $i$, so that, if it triggers an event at time $t^i_\ell$ and $\phi_i = 0$, another event is guaranteed by time $t^i_\ell + T^i_{\max}$. •

### Minimum Inter-Event Time (MIET) Design

As noted in Remark 3.1.2, the design parameter $\sigma_i$ can be used to choose the desired MIET for agent $i$, up to a maximum of $\frac{1}{d_i}$. According to Problem 2, we must be able to guarantee that the lower-bound on the MIET $T_i$ as provided in Theorem 3.1.1 is greater than or equal to the prescribed $\tau_i$.

Consequently, if $\tau_i < \frac{1}{d_i}$ for all $i \in \{1, \ldots, N\}$, then it is easy to see how (3.1) in Theorem 3.1.1 can directly be used to choose the design parameter appropriately for each agent. In the case that there exists some agent(s) $j$ such that $\tau_j \geq \frac{1}{d_j}$, then the graph must be redesigned so that $d_j$ is lower for each agent $j$. Note that there exist distributed

methods of choosing these gains for an existing strongly connected digraph so that it will be weight-balanced [31, 32].

## 3.2 Robustness

A problem with many event triggered algorithms is a lack of robustness guarantees in the triggering strategies. In particular, we consider the effects of two different types of disturbances that are often problematic for event-triggered control systems, discussing how our algorithm can be implemented to preserve the MIET in the presence of additive state disturbances and providing a modification that makes it robust against imperfect event detection.

### Robustness Against State Disturbances

We first analyze the robustness of our MIET against state disturbances. As noted in [26], simply guaranteeing a positive MIET may not be practical if the existence of arbitrarily small disturbances can remove this property, resulting again in solutions that might require the agents to take actions faster than physically possible in an attempt to still ensure convergence. Therefore, it is desirable for our algorithm to exhibit robust global event-separation as defined in [26], which means that the algorithm can still guarantee a positive MIET for all initial conditions even in the presence of state disturbances, which is referred to as a robust MIET.

More formally, we desire the MIET given in (3.1) to hold, even in the presences of arbitrary disturbances. Instead of the deterministic dynamics (2.2), consider

$$\dot{x}_i(t) = u_i(t) + w_i(t), \tag{3.2}$$

where $w_i(t)$ is an arbitrary, unknown, additive state disturbance applied to each agent's state. However, ensuring that the MIET holds in these circumstances depends on the specific implementation, as discussed in the following remark.

**Remark 3.2.1** (Trigger Robustness to Disturbances)**.** Note that, following an event at time $t^i_\ell$, each agent $i$ does not even need to check the trigger condition (2.10) until time $t^i_\ell + T_i$, because, by Theorem 3.1.1, it cannot be satisfied before that time, in the absence of any disturbance. Therefore, in implementation, each agent can wait $T_i$ seconds after triggering an event before triggering a new one. This will have no effect on the performance of the algorithm in the absence of disturbances, but it will ensure that the MIET is observed in the presence of disturbances of the form given in (3.2). Additionally, see the definition of the hybrid system $\mathcal{H}'$ in (A.4) for how this can be modeled in theory. •

Note that this does not guarantee convergence all the way to consensus in the presence of disturbances, simply that the positive MIET will be preserved. Analyzing the actual convergence properties in any formal sense is beyond the scope of this work. Instead, we consider the following simple example to show how the algorithm may handle a disturbance. If each $w_i$ is an independent and identical Gaussian process with zero mean and variance $\sigma^2$, then dynamics of the average position, $\dot{\bar{x}}$, will be a random variable with $E[\dot{\bar{x}}] = 0$ and $\text{var}(\dot{\bar{x}}) = \sigma^2/N$. This indicates that $\bar{x}$ is a Wiener process, which has a Gaussian distribution with a mean equal to the initial average and a variance of $t\frac{\sigma^2}{N}$. We demonstrate the effects of such a disturbance in Section 3.3.

### Robustness Against Imperfect Event Detection

In addition to robustness against state disturbances, another important source of uncertainty that cannot be overlooked in event-triggered control systems is imperfect event detection. Event-triggered controllers are generally designed and analyzed assuming very precise timing of different actions is possible while continuously monitoring the event conditions. This is not only impractical but problematic if not accounted for in the event-triggered control design.

The algorithm presented in Table 2.2 is no longer guaranteed to converge if there are delays in the triggering times. More specifically, let $t^{i^*}_{\ell+1}$ be the time at which the event

condition (2.10) is actually satisfied, but the condition is not actually detected until the actual triggering time $t^i_{\ell+1} = t^{i^*}_{\ell+1} + \delta t^i_\ell$, where $\delta t^i_\ell > 0$ is a random delay in the detection of the event that depends on the hardware/software actually being utilized.

Unfortunately, due to the precise definition of our local jump set (2.16), even an arbitrarily small delay may negate our main convergence result in Theorem 3.1.3. Fortunately, our analysis hinges on only requiring an event is triggered *by* $t^{i^*}_{\ell+1}$ rather than precisely *at* this time. Thinking of this $t^{i^*}_{\ell+1}$ then as a deadline for the broadcast event rather than the precise time it is required, we can simply enlarge our jump set as needed to ensure the deadlines can still be met even with the delays.

More specifically, with a known positive upper-bound on the delay $\Delta_i \geq \delta t^i_\ell$, we can enlarge the jump set to be

$$D_i \subset \widetilde{D}_i \triangleq \{q \in \mathbb{R}^{3N} : \chi_i \leq \frac{e_i^2 \Delta_i}{T_i - \Delta_i}\}. \tag{3.3}$$

This condition will guarantee that even with random bounded time-delays $\delta t^i_\ell \leq \Delta_i$, the time $t^i_{\ell+1} + \delta t^i_{\ell+1}$ at which the delayed event condition actually occurs will still respect the deadline $t^i_{\ell+1} + \delta t^i_{\ell+1} \leq t^{i^*}_{\ell+1}$. Note that our condition requires that $\Delta_i < T_i = \frac{\sigma_i}{d_i}$ and also reduces the MIET by $\Delta_i$. Since $\sigma_i$ is a design parameter, this reveals that the maximum tolerable delay is $\frac{1}{d_i}$. This result is formalized next.

**Theorem 3.2.2** (Robust Convergence with MIET)**.** Given the hybrid system $\mathcal{H}$, using instead the enlarged jump set $\widetilde{D}_i$, if each agent $i$'s delay satisfies $\delta t^i_\ell \leq \Delta_i$ for all $\ell \in \mathbb{Z}_{\geq 0}$, the distributed dynamic event-triggered coordination algorithm presented in Table 2.2, except with events triggered according to (3.3) with $e_i \neq 0$, guarantees that all trajectories of the system asymptotically converge to the set

$$\{q : \widehat{\phi}_i = 0 \ \forall \ i\}.$$

Moreover, the inter-event times for agent $i$ satisfy

$$t^i_{\ell+1} - t^i_\ell \geq \widetilde{T}_i \triangleq \frac{\sigma_i}{d_i} - \Delta_i > 0. \tag{3.4}$$

*Proof.* See the appendix. □

**Remark 3.2.3** (Trigger Robustness to Delays)**.** The intuitive implication of Theorem 3.2.2 is the following. Rather than agent $i$ waiting for the condition (2.10) to be satisfied exactly and responding immediately, it simply begins triggering an event when condition (2.10) could be satisfied soon, and as long as the event can be detected and fully responded to before then, the algorithm will work as intended. However, note that this imposes a trade off because triggering earlier will result in a shorter guaranteed MIET, as shown in the result of Theorem 3.2.2. •

## 3.3 Simulations

To demonstrate our distributed event-triggered control strategy, we perform various simulations using $N = 5$ agents and a directed graph whose Laplacian is given by

$$L = \begin{bmatrix} 2 & -1 & 0 & 0 & -1 \\ 0 & 2 & 0 & 0 & -2 \\ -2 & 0 & 2 & 0 & 0 \\ 0 & -1 & -2 & 3 & 0 \\ 0 & 0 & 0 & -3 & 3 \end{bmatrix}.$$

Additionally, we consider that agents have minimum operating periods of $[\tau_1, \tau_2, \tau_3, \tau_4, \tau_5] = [0.4, 0.25, 0.25, 0.1, 0.2]$. Because the agents have out degrees of $[d_1, d_2, d_3, d_4, d_5] = [2, 2, 2, 3, 3]$, we use (3.1) and choose $[\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5] = [0.9, 0.4, 0.4, 0.3, 0.6]$, so that the guaranteed MIETs for the agents are $[T_1, T_2, T_3, T_4, T_5] = [0.45, 0.2, 0.2, 0.1, 0.2]$.

In light of Remark 3.1.4, we also have a secondary trigger, so that each agent will

trigger its next event at most 2 seconds after its previous event. All simulations use the initial condition $\widehat{x}(0) = x(0) = [-1, 0, 2, 1, 2]^T$, to explore the effects of different design parameters.

Figure 3.1 shows the main results. Figure 3.1 (a) shows the positions of the agents over time, demonstrating that they converge to initial average position, indicated by the dashed line, and the bottom plot shows the evolution of the Lyapunov function, which can be seen to be nonincreasing, although the physical portion and the cyber portion are allowed to increase individually. The top plot in Figure 3.1 (b) shows the evolution of the clock-like state, $\chi_5$, for agent 5, and the bottom plot shows when each agent triggers an event, demonstrating the asynchronous, aperiodic nature of event triggering. Figure 3.1 (c) shows the inter-event times for all agents, with the horizontal lines indicating the theoretical lower bounds. For each agent $i$, the minimum inter-event time $T_i$ can be seen to be respected, and the bound appears to be tight, as expected from the theoretical analysis.

Figure 3.2 (a) shows the effect of applying an additive white Gaussian noise disturbance, i.e. $\dot{x} = u + w$, where each element of $w$ is an independent and identically distributed Gaussian process, with zero mean and a variance of $\sigma_w^2 = 0.1$. To ensure that the MIET is respected in the presence of noise, the algorithm is implemented in a self-triggered fashion. That is, instead of measuring $e_i$ to propagate the dynamics (2.21), $e_i$ is approximated assuming no noise. This simulation suggests that the expected value of each agent's state is the current average position, although that average can now change with time. Figure 3.2 (b) shows that the minimum inter-event times are still respected with this implementation.

Next, to show the effect of the design parameter $\sigma_i$ on the algorithm's performance, we set $\sigma_i = \sigma$ for each agent $i$ and varied it. Figure 3.2 (c) shows the results on 2 statistics: the average communication rate $r_{\text{com}}$, which is the number of events divided by the simulation length, and the cost $\mathcal{C}$, defined as follows. Considering each agent's difference from the average as an output, similar to [33], we adopt the square of the $\mathcal{H}_2$-norm of the system as

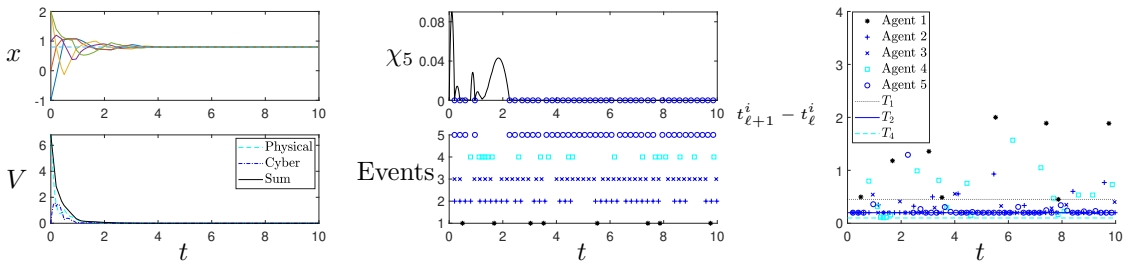


Figure 3.1: Plots of the simulation results of the dynamic event-triggered algorithm showing (a) the trajectories of the agents (top), with the dashed line representing the average, and the evolution of the whole Lyapunov function $V$ (bottom) as well as the physical component $V_P$ and the cyber component $V_C$; (b) the dynamic variable $\chi_5$ for agent 5 (top) and rows of stars indicating the event times of all agents (bottom); and (c) the inter-event times $t^i_{\ell+1} - t^i_\ell$ of each agent $i$, with the lower bounds as computed by (3.1) marked by the lines.

a cost performance metric

$$\mathcal{C} \triangleq \int_{t=0}^{t=t_{\max}} \sum_{i=1}^{N} \left(x(t) - \bar{x}\mathbf{1}_N\right)^2, \tag{3.5}$$

where $t_{\max} = \infty$. However, for simplicity in simulations, we use the rough approximation of $t_{\max} = 20$ being the simulation length. The choice of parameter $\sigma$ can be seen to be a trade off between cost (speed of convergence) and communication rate. Higher values of $\sigma$ result in a higher cost, but also requires less communication and results in a higher MIET by (3.1).

## 3.4 Conclusions

This paper has used the multi-agent average consensus problem to present a dynamic agent-focused event-triggered mechanism which ensures stabilization and prevents Zeno solutions by allowing for a chosen minimum inter-event time for each agent. The algorithm is fully distributed in that it not only requires no global parameters, but the correctness of the algorithm can also be guaranteed by each agent individually. That is, no global conditions (besides connectivity of the graph) need to even be checked to ensure the overall system

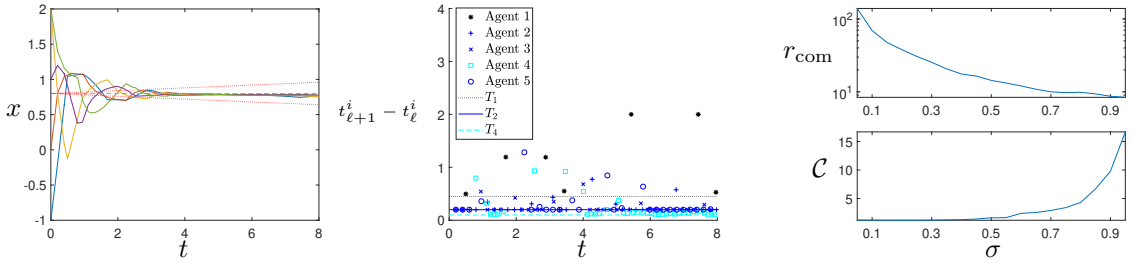


Figure 3.2: Simulation results with additive state disturbances showing (a) the trajectories of the agents subjected to zero-mean additive white Gaussian noise with a variance of 0.1 (the dashed blue line indicates the expected value of the average position ($\bar{x}(0)$), and the dotted red lines show the variance over time ($\bar{x}(0) \pm t\frac{\sigma_w^2}{N}$); (b) the inter-event times $t^i_{\ell+1} - t^i_\ell$ of each agent $i$, with the lower bounds as computed by (3.1) marked by the lines; and (c) the communication rate (top) and $\mathcal{H}_2$-norm cost (3.5) (bottom) as we vary the design parameter $\sigma$.

asymptotically converges. Additionally, it provides robustness against missed event times, guaranteeing convergence as long as events are triggered within a certain window of time.

While this work has presented an algorithm that distributed agents can implement to guarantee asymptotic convergence, further research is needed to study the transient properties of our proposed algorithm and related algorithms. We plan to examine this algorithm to see if it can guarantee exponential convergence to consensus, and, in particular, how the secondary trigger discussed in Remark 3.1.4 should be designed for good performance.

# Appendix A: Hybrid Systems Results and Proofs

**Proof of Theorem 3.1.1 [Positive MIET]**

In order to compute the MIET for agent $i$, we formulate and solve a relevant optimal control problem. Without loss of generality, let $t = 0$ be the time at which agent $i$ triggers an event. We then want to find the smallest possible time $T_i > 0$ at which it can trigger its next event.

The relevant states and initial conditions of our optimal control problem are the local auxiliary state and error, so we define $\zeta_1 \triangleq \chi_i + e_i^2$ and $\zeta_2 \triangleq e_i = x_i - \widehat{x}_i$. This definition of $\zeta_1$ will make the optimal control problem simpler, while containing the relevant information. The dynamics is then given by

$$\dot{\zeta} = \begin{bmatrix} \dot{\zeta}_1 \triangleq \dot{\chi}_i = \gamma_i + 2e_i\dot{e}_i \\ \dot{\zeta}_2 \triangleq \dot{e}_i = -\widehat{z}_i \end{bmatrix} = \begin{bmatrix} \sigma_i\widehat{\phi}_i, \\ -\widehat{z}_i \end{bmatrix}. \tag{A.1}$$

Since an event has just been triggered at $t = 0$, the event condition (2.16) specifies the initial condition $\zeta_1(0) = \zeta_2(0) = 0$. We now wish to find the smallest possible time $T_i > 0$ such that the event condition is satisfied again $\zeta_1(T_i) = \zeta_2(T_i) \neq 0$, which is equivalent to $\chi_i = 0$, $e_i \neq 0$.

In addition to the state $\zeta$ and the constant $\sigma_i$, the dynamics depends on the information provided by neighbors $\{\widehat{x}_j\}_{j\in\mathcal{N}_i^{\text{out}}}$. Since these are unbounded variables, we treat each $\widehat{x}_i - \widehat{x}_j$, for $j \in \mathcal{N}_i^{\text{out}}$, as an input and search for the 'optimal' values that minimize $T_i$. Formally, we seek to minimize

$$J = \int_0^{T_i} 1dt. \tag{A.2}$$

The Hamiltonian is

$$H = 1 + \lambda_1 \sigma_i \widehat{\phi}_i - \lambda_2 \widehat{z}_i$$

where $\lambda \in \mathbb{R}^2$ is the costate. The costate equation ([34]) is

$$-\dot{\lambda} = \frac{\partial H}{\partial \zeta}^T = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The stationarity condition, $\frac{dH}{d(\widehat{x}_i - \widehat{x}_j)} = 0$ [34], implies

$$0 = 2\sigma_i w_{ij}(\widehat{x}_i - \widehat{x}_j)\lambda_1 - w_{ij}\lambda_2, \quad \forall j \in \mathcal{N}_i^{\text{out}}.$$

This allows us to write the state as a function of time, and applying the terminal condition of $\zeta_1(T_i) = \zeta_2(T_i) \neq 0$ allows us to solve for $T_i$. We solve for the costate, using the fact that the Hamiltonian is constant at $H = 0$ [34]. The solutions for the state, costate, 'input,' and minimum time are

$$\begin{aligned}
\zeta_1(t) &= \frac{d_i}{\sigma_i}\left(\frac{\lambda_2}{2\lambda_1}\right)^2 t, \\
\zeta_2(t) = e_i(t) &= -\frac{d_i \lambda_2}{2\sigma_i \lambda_1} t, \\
\lambda_1(t) &= \frac{\sigma_i}{d_i e_i^2(T_i)}, \\
\lambda_2(t) &= -\frac{2\sigma_i}{d_i e_i(T_i)}, \\
\widehat{x}_i - \widehat{x}_j &= -\frac{e_i(T_i)}{\sigma_i}, j \in \mathcal{N}_i^{\text{out}}, \\
T_i &= \frac{\sigma_i}{d_i},
\end{aligned} \tag{A.3}$$

where $e_i(T_i) \neq 0$ is the error at the next event time. Interestingly, the minimum inter-event time $T_i$ does not depend on $e_i(T_i)$ which concludes the proof. ■

**Proof of Theorem 3.1.3 [Global Asymptotic Convergence]**

To aid in several proofs, we redefine the hybrid system (2.11) to a nearly identical one for analysis purposes only. This is a more generalized version of (2.11), with the individual extended states $q_i$ defined in (2.8) augmented by an additional 'timer' state that keeps track of how much time passes between events. This will allow us to provide a more formal analysis of the asymptotic properties of our algorithm when events are triggered within windows of time rather than at particular time instances. Let

$$q_i' \in \begin{bmatrix} q_i \\ \mathcal{T}_i \end{bmatrix} \in \mathbb{R}^4,$$

where $\mathcal{T}_i \in \mathbb{R}_{\geq 0}$ is the time elapsed since agent $i$'s last event. Letting $q' = [{q_1'}^T, {q_2'}^T, \ldots, {q_N'}^T]^T \in \mathbb{R}^{4N}$ denote the full augmented state vector, we can now redefine our system (2.11) with some minimal modifications to the flow/jump sets/maps. The hybrid system we now consider is given by

$$\mathcal{H}' = (C', f', D', G'). \tag{A.4}$$

The flow set remains essentially unchanged,

$$C' = \{q \in \mathbb{R}^{4N} : \chi_i \geq 0 \text{ for all } i \in \{1, \ldots, N\}\}. \tag{A.5}$$

The only difference to the flow map is the propagation of the timer while the state is

flowing,

$$\dot{q}_i' = f_i'(q') \triangleq \begin{bmatrix} f_i(v_i) \\ f_{\mathcal{T}}(\mathcal{T}_i) \end{bmatrix} \quad \text{for } q' \in C',$$

$$f_{\mathcal{T}}(\mathcal{T}_i) \triangleq \begin{cases} 1, & \mathcal{T}_i < T_i \\ [0,1], & \mathcal{T}_i = T_i \\ 0, & \mathcal{T}_i > T_i \end{cases}.$$

Recall that $v_i$ is simply the information locally available to agent $i$ as defined in (2.6), and note that $\dot{\mathcal{T}}_i = f_{\mathcal{T}}(\mathcal{T}_i)$ ensures that $\frac{d}{dt}\mathcal{T}_i = 1$ until $\mathcal{T}_i = T_i$, where the timer stops counting. Thus, the timer is bounded, even if inter-event times are not.

For a dwell-time $s_i \in (0, \frac{\sigma_i}{d_i}]$, the jump set is enlarged,

$$D' = \cup_{i=1}^{N} \{q' \in \mathbb{R}^{4N} : \mathcal{T}_i \geq s_i\}. \tag{A.6}$$

Note that the auxiliary variable is already restricted so that $\chi_i \geq 0$ at all times, and the original jump set (2.16), for agent $i$, was only designed as the boundary of the flow set $C$, requiring an immediate trigger response. Instead, the new jump set (A.6) allows the possibility for agents to trigger before reaching the boundary of the flow set as long as $s_i > 0$ seconds have passed. Due to the guaranteed positive MIET result of Theorem 3.1.1, having $s_i \in (0, \frac{\sigma_i}{d_i}]$ ensures that the state cannot escape the flow set before it is allowed to jump. Also, the MIET result is a prerequisite to using the modified system $\mathcal{H}'$, ensuring that the trajectories with the dwell-time $s_i$ match the trajectories of the original system $\mathcal{H}$.

The only difference to the original jump map (2.17) is the resetting of the timer $\mathcal{T}_i = 0$

for the triggering agent. For $q' \in D_i'$, we consider the following local jump map

$$g_i'(q') = \begin{bmatrix} {q_1'}^+ \\ \vdots \\ {q_i'}^+ \\ \vdots \\ {q_N'}^+ \end{bmatrix} \triangleq \begin{bmatrix} q_1' \\ \vdots \\ \begin{pmatrix} q_i^+ \\ 0 \end{pmatrix} \\ \vdots \\ q_N' \end{bmatrix}.$$

Now, since multiple agents may trigger events at once, the jump map must be described by a set-valued map $G' : \mathbb{R}^{4N} \rightrightarrows \mathbb{R}^{4N}$ [4, 30], where

$$G'(q') \in \{\dots, g_i'(q'), \dots\}, \tag{A.7}$$

for all $i$ such that $q' \in D_i'$. Note that this construction of the jump map still ensures that it is outer-semicontinuous, which is a requirement for some hybrid systems results.

With our new hybrid system (A.4) fully defined, our goal is to show all trajectories of the system with valid initial conditions asymptotically converge to the set

$$\mathcal{A} \triangleq \{q' \in \mathbb{R}^{4N} : \widehat{\phi}_i = 0 \ \forall \ i\}.$$

**Lemma 1** (General Asymptotic Convergence)**.** Given the hybrid system $\mathcal{H}'$, for any $q(0) \in \mathbb{R}^{4N}$ such that $x_i \in \mathbb{R}$, $\widehat{x}_i \in \mathbb{R}$, $\chi_i \geq 0$, and $\mathcal{T}_i \geq T_i$, for $i = 1, \dots, N$, the system state is guaranteed to asymptotically converge to the set $\mathcal{A}$.

We show this by appealing to an invariance principle for hybrid systems [9, Theorem 8.2]. Since we are not concerned with the state of the timers $\mathcal{T}_i$, with a slight abuse of notation we redefine $V(q') = V(q)$ as in (2.19). We will now show that $V$ is nonincreasing along all trajectories of $\mathcal{H}'$. We show that $q'$ converges to the largest invariant weakly

invariant set $\mathcal{A} \subset \{q' \in \mathbb{R}^{4N} : \dot{V}(q') = 0\} \cup \{q' \in D' : G'(q') \subset D'\}$.

While the state is flowing in the original system $\mathcal{H}$ ($q \in C$), we have already shown that the clock defined by (2.21) ensures that $\dot{V} < 0 \ \forall q' \notin \mathcal{A}$ (2.22). Since the modified flow set and map still don't have any direct effect on $V$, this property is preserved. When the system jumps in the original system $\mathcal{H}$ ($q \in D$), we have

$$V\left(G(q)\right) - V(q) = 0,$$

because $V$ does not depend on $\widehat{x}$. Similarly since the timer does not affect the function $V$ at all this property is also preserved in the modified system $\mathcal{H}'$.

Since $V$ is unchanged during jumps and $\dot{V}(q') < 0$ for all $q' \in C' \setminus \mathcal{A}$, we know that $V$ is nonincreasing along all trajectories of $\mathcal{H}'$. Since the modified jump set (A.6) guarantees an agent $i$ cannot jump twice in any $s_i > 0$ second period, all solutions to $\mathcal{H}'$ do not exhibit Zeno behavior. One can then show that both $f'$ and $G'$ are outer semicontinuous, ensuring that $q'$ asymptotically approaches the nonempty largest invariant set in $V^{-1}(c) \cap \mathbb{B}$ for some $c \geq 0$. Since $V$ is radially unbounded with respect to $x$ and $\chi$ this set and the solutions to $x$ and $\chi$ are bounded. Since $C' \cup D' = \mathbb{R}^{4N}$, all solutions are complete.

Therefore, every solution to $\mathcal{H}'$ such that $x_i \in \mathbb{R}$, $\widehat{x}_i \in \mathbb{R}$, $\chi_i \geq 0$, and $\mathcal{T}_i \geq T_i$, for $i = 1, \ldots, N$ is complete, bounded, and asymptotically converges to $\mathcal{A}$. ■

**Lemma 2** (Time until next event)**.** For any time $t \in [t^i_\ell, t^i_{\ell+1})$, let $t^i_{\ell+1}$ be the next time that agent $i$ would trigger an event under (2.10). That is,

$$t^i_{\ell+1} = \inf\{t' \geq t : \chi_i(t') = 0 \text{ and } e_i(t') \neq 0\}.$$

Under the hybrid system (2.11) with $\gamma_i$ defined in (2.21), the time remaining until the next

event is lower-bounded by

$$t^i_{\ell+1} - t \geq \begin{cases} T_i \left(1 - \frac{e_i^2}{\chi_i + e_i^2}\right), & \text{for } (\chi_i, e_i) \neq (0,0), \\ T_i, & \text{otherwise.} \end{cases} \tag{A.8}$$

*Proof:*

This proof relies on examining each agent as a local system, as defined in (A.1). We must find the minimum time to reach a point such that $\chi_i = 0$ and $e_i \neq 0$ from any initial point with $\chi_i(0) \geq 0$ and $e_i(0) \in \mathbb{R}$. We must consider three cases.

- **Case 1:** $\chi_i(0) = 0$ **and** $e_i(0) = 0$

  In this case, we can use Theorem 3.1.1 directly, and the minimum time is $T_i$.

- **Case 2:** $\chi_i(0) > 0$ **and** $e_i(0) = 0$

  Note that in the proof of Theorem 3.1.1, the dynamics of $\zeta$ does not depend on $\chi_i$. Therefore, we can use the same reasoning to find the minimum time for $\chi_i$ to reach a point such that $\chi_i(t) = \chi_i(0)$ and $e_i \neq 0$, which must happen before $\chi_i(t) = 0$. Therefore, the minimum time is lower bounded by $T_i$.

- **Case 3:** $\chi_i > 0$ **and** $e_i \neq 0$

  First, we show that all points in this set lie along the optimal trajectories given by (A.3). Writing the states as explicit functions of time, we have

$$\chi_i(t) = \zeta_1 - \zeta_2^2 = \frac{e_i^2(T_i)}{T_i} t - \left(\frac{e_i(T_i)}{T_i}\right)^2 t^2$$

$$e_i(t) = \zeta_2 = \frac{e_i(T_i)}{T_i} t. \tag{A.9}$$

  To show that we can reach a point $(\chi^*, e^*)$, we must find a $t^* \in (0, T_i]$, such that $\chi^* = \chi_i(t^*)$ and $e^* = e_i(t^*)$, for some value of $e_i(T_i) \neq 0$. Therefore, we use (A.9) to

solve for $e_i(T_i)$ and $t^*$, yielding

$$e_i(T_i) = \frac{T_i e^*}{t^*}$$

$$t^* = \frac{T_i e^{*2}}{x^* + e^{*2}}. \tag{A.10}$$

Note that $e_i(T_i) \neq 0$ and $t^* \in (0, T_i]$ if $e^* \neq 0$ and $\chi^* \geq 0$, so these points lie along optimal trajectories. Intuitively, then, $t^*$ is the time it would take to reach some point $(\chi^*, e^*)$ along an optimal trajectory. Therefore, we can apply the principle of optimality to find that the remaining time until the next event on this optimal trajectory is $T_i - t^*$, and there can be no shorter time before an event.

Putting this all together, the time to reach any point such that $\chi_i = 0$ and $e_i \neq 0$ from any initial point with $\chi_i(0) \geq 0$ and $e_i(0) \in \mathbb{R}$ is lower bounded by

$$\begin{cases} T_i \left(1 - \frac{e_i^2}{\chi_i + e_i^2}\right), & \text{for } (\chi_i, e_i) \neq (0,0) \\ T_i, & \text{otherwise} \end{cases}.$$

■

**Proof of Theorem 3.2.2 [Robust Convergence with MIET]**

This proof follows from Lemma 2 and the analysis in its proof. We first show the existence of a positive MIET, then show that $\chi_i$ remains nonnegative in the presence of the delays $\delta t_\ell^i$, for $i = 1, \ldots, N$. This guarantees that the algorithm in Theorem 3.2.2 is described by hybrid system $\mathcal{H}'$, and we can apply our general convergence result Lemma 1.

First, note that the trigger condition given by (3.3), with $e_i \neq 0$, corresponds to setting $t^* = T_i - \Delta_i$ from (A.10). Because this $t^*$ corresponds to the time elapsed to reach this point along a time optimal trajectory, as described in (A.3), we can conclude that $T_i - \Delta_i$ gives us the minimum time for this trigger condition to be satisfied from the previous event. Therefore, $\widetilde{T}_i = T_i - \Delta_i \leq t_{\ell+1}^i - t_\ell^i$, and the condition $T_i > \Delta_i$ ensures that $\widetilde{T}_i$ is positive.

Now that a positive minimum inter-event time has been established, we must ensure that $\chi_i$ remains nonnegative in the presence of the delays $\delta t_\ell^i$, for $i = 1, \ldots, N$.

Note that the trigger condition $e_i \neq 0$ with the system in the extended jump set (3.3) corresponds to the remaining time from (A.8) in Lemma 2 being equal to $\Delta_i$, or formally,

$$\Delta_i = T_i \left(1 - \frac{e_i^2}{\chi_i + e_i^2}\right).$$

Since $\delta t_\ell^i \leq \Delta_i$, we can conclude that the event is acted upon before $\chi_i = 0$, $e_i \neq 0$ is satisfied, so $\chi_i$ remains positive, for $i = 1, \ldots, N$. Therefore, the algorithm in Theorem 3.2.2 will be described by hybrid system $\mathcal{H}'$. Finally, we can apply our general convergence result Lemma 1 to guarantee convergence to the set $\{q \in \mathbb{R}^{3N} : \widehat{\phi}_i = 0 \ \forall \ i\}$. ■

# Bibliography

# Bibliography


[1] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[2] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71–82, 2007.

[3] X. Yi, K. Liu, D. V. Dimarogonas, and K. H. Johansson, "Distributed dynamic event-triggered control for multi-agent systems," in *IEEE 56th Annual Conference on Decision and Control*, Melbourne, VIC, Australia, 2017, pp. 6683–6688.

[4] C. De Persis and R. Postoyan, "A Lyapunov redesign of coordination algorithms for cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 808–823, 2017.

[5] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *IEEE Conference on Decision and Control*, Maui, Hawaii, USA, 2012, pp. 3270–3285.

[6] C. De Persis and P. Frasca, "Robust self-triggered coordination with ternary controllers," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 3024–3038, 2013.

[7] A. V. Proskurnikov and M. Mazo, Jr., "Lyapunov design for event-triggered exponential stabilization," in *HSCC '18: 21st International Conference on Hybrid Systems: Computation and Control*, Porto, Portugal, 2018, pp. 111–119.

[8] R. Postoyan, P. Tabuada, D. Nesic, and A. Anta, "A framework for the event-triggered stabilization of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 982–996, 2015.

[9] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems*. Princeton, NJ, USA: Princeton University Press, 2012.

[10] B. Liu, W. Lu, and T. Chen, "Consensus in networks of multiagents with switching topologies modeled as adapted stochastic processes," *SIAM Journal on Control and Optimization*, vol. 49, no. 1, pp. 227–253, 2011.

[11] D. V. Dimarogonas and E. Frazzoli, "Distributed event-triggered control strategies for multi-agent systems," in *47th Annual Allerton Conference*, Monticello, IL, USA, 2009, pp. 906–910.

[12] D. V. Dimarogonas and K. H. Johansson, "Event-triggered control for multi-agent systems," in *48th IEEE Conference on Decision and Control*, Shanghai, P.R., China, 2009, pp. 7131–7136.

[13] E. Kharisov, X. Wang, and N. Hovakimyan, "Distributed event-triggered consensus algorithm for uncertain multi-agent systems," in *AIAA Guidance, Navigation, and Control Conference*, Toronto, ON, Canada, 2010, pp. 1–15.

[14] C. Nowzari, E. Garcia, and J. Cortés, "Event-triggered communication and control of networked systems for multi-agent consensus," *Automatica*, vol. 105, pp. 1 – 27, 2019.

[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] E. Garcia, Y. Cao, H. Yu, P. Antsaklis, and D. Casbeer, "Decentralized event-triggered cooperative control with limited communication," *International Journal of Control*, vol. 86, no. 9, pp. 1479–1488, 2013.

[17] X. Meng and T. Chen, "Event based agreement protocols for multi-agent networks," *Automatica*, vol. 49, no. 7, pp. 2125 – 2132, 2013.

[18] X. Meng, L. Xie, Y. C. Soh, C. Nowzari, and G. J. Pappas, "Periodic event-triggered average consensus over directed graphs," in *54th IEEE Conference on Decision and Control*, Osaka, Japan, 2015, pp. 4151–4156.

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

[20] A. Amini, A. Asif, and A. Mohammadi, "CEASE: A collaborative event-triggered average-consensus sampled-data framework with performance guarantees for multi-agent systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 23, pp. 6096 – 6109, 2018.

[21] F. Xiao and T. Chen, "Sampled-data consensus in multi-agent systems with asynchronous hybrid event-time driven interactions," *Systems and Control Letters*, vol. 89, pp. 24 – 34, 2016.

[22] Y. Liu, C. Nowzari, Z. Tian, and Q. Ling, "Asynchronous periodic event-triggered coordination of multi-agent systems," in *IEEE 56th Annual Conference on Decision and Control*, Melbourne, VIC, Australia, December 2017, pp. 6696–6701.

[23] G. S. Seyboth, D. V. Dimarogonas, and K. H. Johansson, "Event-based broadcasting for multi-agent average consensus," *Automatica*, vol. 49, no. 1, pp. 245 – 252, 2013.

[24] B. Cheng and Z. Li, "Consensus of linear multi-agent systems via fully distributed event-triggered protocols," in *36th Chinese Control Conference*, Dalian, China, 2017, pp. 8607–8612.

[25] V. S. Dolk, M. Abdelrahim, and W. P. M. H. Heemels, "Event-triggered consensus seeking under non-uniform time-varying delays," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 096–10 101, 2017.

[26] D. P. Borgers and W. P. M. H. Heemels, "Event-separation properties of event-triggered control systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2644 – 2656, 2014.

[27] G. H. Hardy, J. E. Littlewood, and G. Polya, *Inequalities*. Cambridge, UK: Cambridge University Press, 1952.

[28] J. Berneburg and C. Nowzari, "Distributed dynamic event-triggered coordination with a designable minimum inter-event time," in *American Control Conference*, Philadelphia, PA, USA, 2019, pp. 1424 – 1429.

[29] P. Xu, C. Nowzari, and Z. Tian, "A class of event-triggered coordination algorithms for multi-agent systems on weight-balanced digraphs," in *American Control Conference*, Milwaukee, WI, USA, 2018, pp. 5988–5993.

[30] R. G. Sanfelice, J. J. B. Biemond, N. van de Wouw, and W. P. M. H. Heemels, "An embedding approach for the design of state-feedback tracking controllers for references with jumps," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 11, pp. 1585–1904, 2014.

[31] B. Gharesifard and J. Cortés, "Distributed strategies for generating weight-balanced and doubly stochastic digraphs," *European Journal of Control*, vol. 18, no. 6, pp. 539 – 557, 2012.

[32] A. I. Rikos, T. Charalambous, and C. N. Hadjicostis, "Distributed weight balancing over digraphs," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 2, pp. 190 – 201, 2014.

[33] S. Dezfulian, Y. Ghaedsharaf, and N. Motee, "On performance of time-delay linear consensus networks with directed interconnection topologies," in *American Control Conference*, Milwaukee, WI, USA, 2018, pp. 4177–4182.

[34] F. L. Lewis, D. L. Vrabie, and V. L. Syrmos, *Optimal Control*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2012.

# Curriculum Vitae

James Berneburg received the Associate of Arts and Sciences Degree in Science from Lord Fairfax Community College in May 2012, and the BS in Electrical Engineering from George Mason University in 2017. He is currently pursuing a Ph.D. in electrical engineering from George Mason University. His current research interests include multi-agent systems, event-triggered control, and nonlinear control.