LYAPUNOV-BASED INDIRECT CONTROL OF ROBOTIC SYSTEMS

By

PATRICK MCGILL AMY

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2025

To my wife Josie and to my family for their unwavering support and endless love

ACKNOWLEDGMENTS

I thank my advisor, Dr. Warren Dixon, for his mentorship and patience during my PhD career. I thank my committee members, Dr. Matthew Hale, Dr. Jane Shin, and Dr. John Shea for their invaluable guidance and thoughtful feedback. I thank my colleagues in the Nonlinear Control and Robotics Laboratory at the University of Florida and the Naval Surface Warfare Center Panama City Division for inspiring me to challenge myself and supporting me along the way.

“If perhaps I have seen a great distance, it is because I have stood on the shoulders of Giants.”

My Giants: Dr. John Amy, Dr. Warren Dixon, Dr. Matthew Hale, Dr. Patrick Walters, Dr. Zachary Bell, Dr. Federico Zegers, Dr. Patryk Deptula, Dr. Ryan Licitra

TABLE OF CONTENTS

LIST OF TABLES

Table page

LIST OF FIGURES

LIST OF ABBREVIATIONS

ETC	Event-Triggered Control
FOV	Field Of View
GES	Globally Exponentially Stable
HDI	Hybrid Differential Inclusion
IET	Inter-Event Time
MAS	Multi-Agent System
PAS	Practically Asymptotically Stable
STC	Self-Triggered Control
UUB	Uniformly Ultimately Bounded

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

LYAPUNOV-BASED INDIRECT CONTROL OF ROBOTIC SYSTEMS

By

Patrick McGill Amy

August 2025

Chair: Warren E. Dixon
Major: Mechanical Engineering

Robotic systems operating in austere environments face unique navigation and localization challenges. An important factor when developing robotic systems for austere environments is the size of the system. The solutions to navigation and localization for such environments often require cost-prohibitive or space-prohibitive sensors. If a team of robotic systems is employed, then the cost-capability gap can be mitigated. Indirect control is a field of research that involves the design of control laws to regulate a directly controllable agent's state and indirectly regulate, through an uncertain influence dynamic, an agent without an explicit control input. This dissertation investigates stability methods and control designs for a small team of robotic agents with navigation and localization capabilities to indirectly control a larger team of robotic agents with limited navigation or localization capabilities.

CHAPTER 1
INTRODUCTION

1.1 Background

Unmanned robotic systems have been used in a multitude of tasks (mine hunting, space exploration, environmental monitoring, search and rescue, etc) and have demonstrated the efficacy of employing robotic platforms for such tasks. As more robotic platforms become commercially available, more applications for robotic teams are relevant. Utilizing a robotic team can increase task completion efficiency and decrease the overall risk (schedule impacts, cost, injury to personnel, etc.) associated with the task. Operating robotic teams in a coordinated manner, including incorporating other related systems (sensors, communication relays, etc.) to boost task efficiency, forms a multi-agent system (MAS).

It is often cost prohibitive to create a MAS with robotic systems with robust navigation sensor suites (fiber optic gyro, DVL, USBL sonar, etc.) and high bandwidth communication hardware. The cost of a MAS is directly related to the cost of each agent, a MAS comprised of robots with basic navigation sensors (GPS, SLAM, etc) and communication hardware (wifi, etc.) will be more affordable, hence feasible, than a MAS comprised of robots equipped with state-of-the-art sensing and communication hardware. Sometimes, robust capabilities are necessary to accomplish basic navigation and localization; for example, navigating underwater or in a gps-denied envrionment. By making use of inter-agent interactions, an agent with a full sensor suite could aide an agent with basic capabilities in challenging environments. Utilizing a few capable robots to mitigate the technological deficiencies of many incapable robots can allow for large multi-agent operations in previously infeasible domains.

1.2 Multi-Agent Systems

A multi-agent system is comprised of a number of individual nodes, or agents, that are connected in a way that permits information exchanges [1]. A MAS can consist of

sensors, computers, manned/unmanned platforms, among many other entities. A MAS comprised of only one type of agent is referred to as a homogeneous MAS; conversely, if a MAS consists of different types of agents (sensors, robots, communication relays, etc.) the MAS is referred to as heterogeneous [1, 2]. Heterogeneous MASs are of interest when it is desired to leverage capable platforms to compensate for under-equipped platforms in-situ [3].

There are two major inter-node communication paradigms in MASs: centralized and distributed [4, 5]. In centralized communication, every node in the MAS broadcasts and receives information to and from a central node [6, 7]. In this convention, all transmitted information must pass through the central node for it to reach its destination [6, 7]. In distributed (or decentralized) communications, each node in the MAS exchanges information with the nodes that are within its communication range [1]. In this paradigm, information flows from the source node to its destination node through a path of interconnected nodes [1]. There are advantages and disadvantages to each paradigm, these advantages and disadvantages become more important as a MAS increases in size (increasingly more nodes are added) [8]. The control of a MAS fits within the same two paradigms: centralized and distributed [4]. In centralized control of a MAS, a central node computes the control input for every node in the system based on its available information [6, 7]. Conversely, in distributed control each node in the MAS computes its own control input based on the local information available [5].

Using graphs to represent a MAS allows for the inclusion of physical constraints from onboard hardware in the theoretical model of the system [9–11]. A graph-based representation of a MAS uses nodes to represent the agents in the MAS, and edges between specific agent pairs to depict the ability to exchange information [1]. For example, if there is a range-based constraint between nodes, an edge may be removed from the graph if the corresponding nodes become too far apart; similarly, an edge may form in the graph if the corresponding nodes become sufficiently close [9]. Desirable

attributes of a MAS, such as guaranteed and persistent connectivity between agents, can be attained by proving certain characteristics of the associated graph (i.e. edge preservation) [1, 9].

1.3 Event-Triggered and Self-Triggered Consensus Control for Multi-Agent Systems

Common MASs of interest are swarms of aerial drones, teams of mobile robot manipulators, and satellite constellations, all of which operate with finite power and communicate through limited bandwidth communication networks [12–14]. It is advantageous to design control algorithms that mandate inter-agent communication among MAS agents only when essential to maintain a desired performance level [15]. Doing so negates the need of continuous communication, which helps prevent bandwidth saturation. Reducing network activity is particularly desirable when the communication network (and the MAS in general) grows in size [15–17]. Reducing inter-agent communication and achieving a desired performance level necessitates intermittent and asynchronous communication between agents [5, 18]. Aperiodic communication means that agents broadcast their information at non-continuous, possibly irregular intervals. Asynchronous communication indicates that the broadcast times of different agents are not coordinated or synchronized with one another [5].

To address the physical constraints in a MAS, event-triggered control (ETC) and self-triggered control (STC) offers advantages over periodic control to address energy, computation, and communication constraints when designing feedback control laws [5, 6, 17, 19]. When feedback control laws are computed continuously or at uniform time intervals, those control laws can be categorized as periodic control [5]. Periodic control encompasses traditional continuous and discrete time control [5]. Conversely, aperiodic control involves feedback control updates that occur in non-uniform time intervals [5]. Event-triggered and self-triggered control are aperiodic control methods that are comprised of two elements: a feedback controller that computes the control

input, and a trigger generator that dictates when to update the control input [5]. In ETC, the trigger generator (also referred as a trigger mechanism or trigger function) is typically a function of the state of the system, requiring periodic sampling of the system's state [15, 17]. Self-triggered control differs from ETC: during a control update, the time of the next control update is computed by using state estimates projected forward in time, typically by using state data and the plant dynamics [5, 20]. Both ETC and STC results typically employ a zero-order hold between control updates, where the control update is held constant between updates [4, 5, 15].

There are ETC results that specifically aim to regulate the rate of inter-agent communications, the authors of which develop a trigger generator that dictates communication transmissions [9, 19, 21, 22]. Sometimes this is separately referred to as event-triggered communication, other times it is still referred to as event-triggered control, even if the control of the system is not bound to a trigger generator. Event-triggered coordination, however, refers specifically to the combination of event-triggered control and event-triggered communication [21, 23–28]. In event-triggered coordination, an agent's control updates and communication transmissions occur based on a trigger generator [21, 23–26, 29, 30]. Similar extensions of self-triggered control also appear in related literature [31].

When an event-trigger condition is met, an event occurs and the time at which this happens is called an event time [16]. The difference between consecutive event times is called the inter-event time, and the minimum inter-event time (MIET) is defined as the greatest lower bound on the inter-event times [16]. It is necessary in ETC work to show that consecutive inter-event times do not tend to zero (i.e., Zeno behavior). Zeno behavior occurs when a process requires infinitely many discrete transitions in a finite time duration, which is not implementable on physical systems [16, 21, 25]. In [16, 32, 33] the properties of inter-event times were studied; specifically, in [16] the existence of positive MIETs for common classes of event-triggers was formalized. It is frequently

shown that a novel event-trigger has Zeno-free behavior but does not guarantee that the developed solution is implementable on hardware because the MIET from the event-trigger would require an unfeasible communication rate for common hardware. The idea of designing an ETC system to achieve a desired MIET that results in an implementable ETC formulation has been studied in results such as [9, 24, 25, 34–38]. There is a distinction between a user-defined MIET and a designable MIET: a user-defined MIET requires system parameters to depend on the chosen MIET, whereas a designable MIET requires its system parameters and variables to be tuned to achieve the desired MIET [9, 24, 25, 34–38]. In [38], among others, the event-trigger function is designed to produce the desired MIET of the system. The authors of [35, 36] present a centralized dynamic event-trigger mechanism and a distributed dynamic event-trigger mechanism that can achieve consensus over strongly connected digraphs with a designed MIET. In [25], the authors presented an ETC formulation that took advantage of individual agent clock-like variables to achieve average consensus with a user-defined MIET. Apart from [24] and [25], the idea of an ETC formulation with a user-defined MIET, as opposed to a designed MIET, has not extensively been studied.

A common control objective used to demonstrate the efficiencies gained in ETC and STC is the consensus control objective [4, 17, 20, 39–43]. The consensus, or agreement, protocol involves each agent in a MAS using locally available information to reach a globally agreed upon state [1]. For a MAS with N agents, where $i \in 1, 2, ..., N$, then the classic consensus protocol for agent i is given by

$$\dot{x}_i(t) = \sum_{j \in \mathcal{N}(i)} \left(x_j(t) - x_i(t) \right),$$

where x_i and x_j are the states of agent i and agent j, respectively [1]. In this classic example, consensus is achieved when the inter-agent distances are zero, i.e., $\|x_i - x_j\| = 0$ for all $i, j \in \mathcal{V}$ [1]. The consensus problem is often considered since other MAS coordination problems (rendezvous, distributed estimation, formation control, etc.)

can be expressed as a consensus problem [8, 9, 38, 44–46]. Expanding the consensus objective to reaching a neighborhood about the globally agreed upon state, yields the approximate consensus problem. Consensus-type protocols can also be extended to the approximate form, and two common approximate consensus protocols are approximate rendezvous and approximate leader-follower consensus [2, 9, 38, 42, 47]. To execute these desired consensus protocols, the intermittent communication and control characteristics inherent to these MASs must be included in the feedback control design.

1.3.1 Event-Triggered Consensus Control

An early example of event-based sampling for single feedback control loops is proposed in [48], where the state is periodically sampled and the control update is computed and applied only when measurements satisfied a design threshold. This approach was shown to outperform periodic sampling and control for a first-order system [48]. In results such as [15], a triggering mechanism for a state feedback controller where the trigger mechanism is a function of the norm of the difference between the system's current state and the system's state at the previous sampling instant. When this trigger function met a prescribed threshold, the control input would be computed and applied [15]. When the control update is computed, the trigger function would update its internal values and reset [15]. This result was guaranteed to exhibit a certain level of performance, and it was shown that the inter-execution times did not converge to zero - specifically, that these inter-execution times are lower bounded by a positive constant [15].

Further development of event-triggered control is presented in [7] and [6], where a control scheme is designed to decentralize a centralized event-triggered condition that depends solely on locally sampled information. Additionally, the presented trigger mechanism permits increasing the inter-execution times between controller re-computations without affecting the guaranteed performance. The event-triggered broadcasting of state

information in distributed networked control systems with data dropouts and transmission delay is examined in [49] and [50]. Similar to [15], the proposed event-triggering scheme in [49] and [50] requires a subsystem to broadcast its state information to its neighbors only when the subsystem's local state error exceeds a specified threshold. This results in [49] and [50] shows a Lyapunov-based approach to stabilize a distributed control system over an ad-hoc network in the absence of synchronization within the communication network. Foundational works, such as [7, 15, 48, 50], suggest that event-triggered control reduces resource utilization while maintaining a suitable control performance in a MAS, and [18] presents a comprehensive overview and theory for event-driven systems that validates the performance of these results. In [51], an ETC scheme is presented whose closed-loop system can achieve approximate performance of the behavior of the corresponding continuous state-feedback system. Furthermore, in [51], it is shown that the sensitivity bound $\bar{e}$ of the event-trigger can be chosen such that the event-based control loop tracks the continuous state-feedback loop with a desired precision. The inter-agent communication frequency adjusts based on the disturbance, such that if the disturbance is sufficiently small, a trigger-event will not occur.

Event-triggered control has been repeatably shown to be suitable for coordinated control of MASs with network constraints. Following the methods in [15], [17] presents both a centralized and distributed ETC design, and both are shown to be effective control designs for a class of cooperative control problems. The overall systems achieve average consensus asymptotically. The class of cooperative control problems described in [17] are a set of problems that can be expressed as a first-order agreement problem, as discussed in [52]. Furthermore, [17] assumes that all agents periodically monitor their neighbors' states and that each agent updates its control law at its own event-times, and also when each of its neighbors experience an event-time. In the previous work [4], each agent required global knowledge of the initial average of the MAS to

implement the control strategy. The approach in [17] shows that this assumption is not required and global knowledge of the initial average of the MAS is not needed.

Similar to the ETC design in [17], the control design in [53] dictates that information exchanges between agents occur in an event-based scheme. Specifically, the continuous monitoring of the neighbors' states is not required to achieve consensus; however, each agent updates its control input when it transmits or receives a new information. This approach is similar to [50], where each agent has global information of the consensus goal, but in [53] the consensus point is unknown to the agents, and each agent strictly has access to local information. In [53], similar to [15], the trigger mechanism of each agent is a function of the measurement error and an event is triggered when the norm of the measurement error meets a predefined criteria. However, the proposed trigger functions in [53] exhibit exponentially decreasing trigger criteria overtime with non-negative offsets. This approach guarantees asymptotic convergence to average consensus or to approximate consensus.

The results discussed previously utilized static, undirected communication graphs to represent inter-agent communication in the considered MAS. In [54–56], ETC designs are presented where switching topologies and directed graphs are used to represent inter-agent interactions in the MAS. In particular, the authors in [56] present an event-triggered consensus algorithm for switching network topologies with distributed and sampled-data event-trigger mechanisms. The distributed event-trigger mechanism, in this case, is driven by sequential communication exchanges between an agent experiencing a control update and its neighboring agents. In [54], an ETC design is applied to a MAS comprised of agents modeled by a linear multi-agent systems, $x_i = Ax_i + Bu_i$. It was shown that the system matrix A is not required to be stable, and the Laplacian matrix of the communication topology is not required to be symmetric. In [55], a distributed ETC strategy is presented for a MAS whose agents communicate

based on a directed graph. Conditions are presented to ensure consensus is achieved for the proposed ETC design.

1.3.2 Self-Triggered Consensus Control

As discussed previously, self-triggered control requires each agent to compute its next update time during a control update [57]. Specifically, in a self-triggered formulation, the next time, $t_i + 1$, the control law is updated is predetermined at the previous event time t_i. This predetermination of control update times frees the agent from monitoring the state error measurement that triggers the control action between consecutive updates, whereas in event-triggered control the continuous monitoring of a trigger condition is required [57]. In [58], a self-triggered implementation was developed with control law execution times that are defined by a function of the dynamics of the system, the desired performance, and the current measurement of the state. The self-trigger conditions were derived for two classes of systems: state-dependent homogeneous systems and polynomial systems. The rate of consensus of the presented self-trigger conditions was shown to be related to how often the controller executed. The more often the controller executed, the better the convergence: this is a common observation in ETC and STC [5, 15, 17, 58].

In [17], ETC and STC were shown to be capable of achieving cooperative control algorithms that can be expressed as a first order agreement problem. In [20] and [17], event-triggered and self-triggered control formulations were presented in both a centralized and decentralized form, and compared in terms of control update frequency. The self-triggered formulation resulted in more controller updates than the event-triggered scheme. The larger controller updates was attributed to the self-trigger scheme over-approximating how often control updates were required based on the state of the agent, the states of its neighbors, and the network [17, 20]. For such a self-triggered scheme, the inter-execution times (the time between control updates) are strictly positive except when all the agents in the MAS reach agreement (all agents are

in consensus). This implies that Zeno-free behavior is achieved while not in consensus [17].

In [42], a hybrid self-triggered/time-triggered approach was developed that displayed better performance than the self-trigger schemes proposed in [17], in terms of reduced trigger events, control updates, and communication transmissions. This improved performance claimed in [42], was attributed to extending the available information of an agent from its surrounding neighbors to its two-hop neighbors results. Additionally, in [59], a hybrid self-triggered/time-triggered approach is presented. Specifically, for a MAS with N agents, where $i \in 1, 2, ..., N$, let the time an event occurs for agent i be denoted t_k^i, for $(k = 1, 2, ...)$. The event times, presented in [42], of agent i are computed by

$$t_{k+1}^i = t_k^i + \max\left\{\tau_k^i, b_i\right\},$$

where b_i is a strictly positive real number and τ_k^i is the self-trigger condition. This approach bounds the inter-event time by the strictly positive number b_i, which implies that all agents in the MAS will exhibit Zeno-free behavior. This, however, does not result in a free choice of the minimum inter-event times as b_i was shown to be upper bounded to ensure asymptotic stability [42, Theorem 1].

In [60], the authors present both an ETC and an STC design that do not require any global information *a priori*. This design is shown to achieve consensus exponentially fast, and is free from Zeno behavior. The STC design was formulated to avoid the continuous sensing and listening that is required of the ETC scheme. The STC does not require each agent to perpetually monitor incoming information from neighboring agents, each agent predicts its next trigger-time and shares this information with its neighboring agents at the current trigger instant. This reduces resource usage by each agent as state sensing and broadcasting occurs at one's trigger-times, and network monitoring for incoming information occurs at the neighboring agent's triggering times [60].

1.3.3 Extensions of Event-Triggered Consensus

Robotic teams are increasingly being used in automated sensing applications (forestry surveying, ocean floor mapping, search and rescue, etc.) to complete these mission objectives more efficiently than previous methods [13, 14, 61]. These automated sensing applications require multiple coordination objectives to complete the given mission [13, 14, 61]. For example, if a robotic team must move in a formation to collectively observe a large swath of its environment, an initial rendezvous step where the robots congregate to a starting area to reduce initial formation control error can increase overall performance [62]. The goal of rendezvous for a MAS is to position every agent in the MAS to common point, rendezvous refers to achieving consensus in position. Some examples of applications of MAS rendezvous are robotic logistics support, satellite rendezvous and docking, and multi-robot cooperative landing [12, 44, 63]. As discussed previously, many MAS control objectives (formation control, orientation alignment, target tracking, etc.) can be expressed as a consensus problem, and the consensus problem is well studied within network control literature [8, 46, 64]. MAS control design for physical systems must accommodate hardware constraints while guaranteeing the desired control objective is achieved; however, hardware constraints elicit non-trivial challenges to consensus results [24]. These hardware constraints could include sensor sampling rates, or communication bandwidth and range limitations [9, 16, 24, 25, 38, 65]. To demonstrate a MAS control objective result on a team of robots, for example, the hardware constraints resulting from the robot's communication medium must be considered.

The results in [9] and [38] provide a basis for achieving the MAS rendezvous control objectives for a MAS whose agents have hybrid dynamics. Specifically, those results present a MAS model with agent dynamics based on the hybrid differential inclusions (HDI) framework whose inter-agent communications are determined by an ETC mechanism. In [9], a class of distributed event-trigger functions that yield approximate rendezvous with graph preservation and positive MIETs is formalized.

Additionally, in [38] the approximate rendezvous problem is achieved for second-order MASs.

1.4 Single Agent and Multi-Agent Indirect Control

The pursuit-evasion problem involves one (or more) pursuers attempting to intercept one (or more) evaders, during which the evaders attempt to avoid interception [66–68]. The pursuit-evasion problem is studied in the field of robotics such as automated surveillance, missile interception, animal control, etc [69–71]. Research across multiple disciplines has led to rapid advances in robotic pursuit-evasion; however, despite investigating the same problem, the developed approaches differ significantly based upon contributions from each discipline [72]. Many results pertaining to the pursuit-evasion problem have come from the fields of game theory and control theory [72], the focus in the following discussions will be on the contributions from the control theory discipline. In particular, control strategies are presented in [69, 73] which can guarantee the pursuers capture the evader if a sufficient number of pursuers are present - the minimum number of pursuers needed for guaranteed capture were shown to depend on the pursuer-evader speed ratio. Alternatively, an escape strategy for out-maneuvering pursuer agents using mathematical frameworks based on Apollonius circles was presented in [71].

Herding is a type of pursuit-evasion problem, and it differs from many pursuit-evasion results because herding includes regulating the evader to a control objective (goal location, velocity alignment, etc.) after interception [74]. Herding can be considered a leader-follower problem because the pursuers control the evaders upon interception; however, herding differs from the typical leader-follower problem due to the difference in the use of indirect versus direct control [74–78]. In leader-follower problems, the typical control objective involves direct control over every agent, whereas in herding the evader agents do not have an explicit control input [74, 77]. In herding, evaders are only indirectly controlled through an influence function that includes the

states of the pursuer and evader [74–78]. For example, the influence a pursuer can exert on an evader could be a function based on the distance between the two [74, 77]. When an evader has the same goal as the pursuers, then the evader is referred to as cooperative, but when an evader does not have a goal or has a goal different from the pursuers', then the evader is deemed agnostic or antagonistic [79–81].

Herding results can be categorized as belonging to either control theory methods or rule-based methods [82–85]. There is a subclass of learning-based methods containing results that make use of off-line training to learn a heuristic or a ruled-based type of herding method [83–85]. Learning-based methods are categorized with rule-based methods as they face the same limitations [82, 86]. Classic examples of the rule-based method are presented in [82] and [87], where the pursuers act according to a designed heuristic or behavior. Learning-based methods involve training different models how to intercept evaders or drive a flock [84, 85]. A flock refers to a cohesive evader group, and the typical model of how evaders interact in a flock is presented in [88], where evaders use relative measurements to others evaders to form a flock. These methods, along with the rule-based methods have been combined with classic path-planning methods to enable pursuers to regulate evaders through cluttered environments [84, 85, 89–92]. Control theory methods involve developing feedback control strategies for the pursuer to regulate the evader(s) to a desired set-point [75, 76, 79, 80, 93, 94]. Learning-based and rule-based methods are generally limited to a relatively-small flock size, and this motivates the use of control theory methods applied in a distributed pursuer scenario to address herding larger flocks [82–86].

Each class of herding methods can be further divided by how many pursuer agents are considered. In the single pursuer agent case, one pursuer will either drive a flock or individual agent(s) to a goal [74, 77]; however, in multi-agent herding more than one pursuer agent drives a group of evaders, a single evader, or multiple independent evaders to a goal [78, 80, 93, 94]. Contributions from each of these classes will be

discussed in 1.4.1 and 1.4.2 to illustrate the differences in approach and problem formulation.

1.4.1 Single Agent Indirect Control

The single agent herding problem considers one pursuer regulating one or more evaders, or a flock of evaders to a control objective. In [86, 95, 96], the evaders are assumed to adhere to the previously discussed flock dynamics. The results in [82, 86, 87, 95–98] follow formulated heuristics or rules in order to achieve the outlined herding problem. In [96], the herding problem was formulated such that the goal of the pursuer was to push a trespassing flock out of the protected region. The pursuer would interact with the flock by sequentially positioning itself at a periodically refreshed set of points. The set of points are chosen to maximize the deflection of the flock's flight path. This approach relies on the inherent stability of the flocking dynamics to prevent fragmentation of the flock.

In [86] and [98], the pursuer is not assumed to have information on the state of the flock. The pursuer is modeled with a limited field-of-view (FOV); where, the pursuer is able to determine the state of the evader(s) within its FOV. The evaders in [98] flock together as described in [88], and three control objectives are compared: farthest-agent targeting, center-of-group targeting, and online target switching. Furthermore, in [98] the efficacy of each control objective was shown to depend on how closely evaders flocked together. In [86], the evader's flock dynamics were modeled to be attracted to the center of mass of at least half of the total number of evaders. This model allows flocks to be broken into manageable sub-groups by the pursuer. This result, [86], allowed the pursuer to take one of three actions: (1) if the pursuer is too close, it will not move; (2) if all evaders are within a radius about the global center of mass (GCM) of the flock, then the pursuer aims towards a steering point directly behind the flock relative to the goal; (3) if an evader is outside the radius about the GCM, the pursuer aims to a point directly behind the evader relative to the flock to force that evader back to the flock. The

authors of [95], build upon this approach by combining tasks (2) and (3). The pursuer controls the flock by positioning itself relative to the GCM of the flock to use its influence to push the flock in the desired direction. The pursuer can move in a v-formation about the steering point of the flock to push evaders closer to the GCM.

The result presented in [74] solves the single agent herding problem by formulating a sliding-mode controller (SMC) to compensate for uncertainties in the evader(s) motion model. In [77], an adaptive control approach was adopted in place of the high-frequency SMC, where a data-based parameter estimation method, integral concurrent learning (ICL) was used to learn the linearly parameterized (LP) evader dynamics by keeping a history of input–output data to improve parameter estimation and facilitate the switched systems analysis [99]. In both [74] and [77], the pursuer–evader interactions were modeled by a distance-based potential function. In [78], a more general, but unknown, distance-based pursuer-evader interaction is considered where the only assumption is that the evader reacts to a repulsion force from the pursuer. Furthermore, in [78], the unknown pursuer-evader interaction was approximated using a neural network (NN), where ICL was integrated within the NN update law to achieve a transient performance required to develop dwell-time conditions for switching.

In [75] and [76], the herding problem with uncertainties in both the pursuer and evader dynamics was addressed. The uncertain dynamics in both agents were approximated by using ICL, and an approximate dynamic programming (ADP) strategy, as presented in [100], was used to develop an approximate optimal solution to the optimal control problem using a function approximation method. The closed-loop herding system was shown to be uniformly ultimately bounded (UUB) by performing a Lyapunov-based stability analysis. This result was further developed in [101], where a multi-timescale Lyapunov-based deep neural network (Lb-DNN) was used approximate the uncertain dynamics.

1.4.2 Multi Agent Indirect Control

Multi agent herding involves either assigning individual pursuers from a team to herd individual evaders (one-to-one herding), controlling a team of pursuers to herd one evader (many-to-one herding), or controlling a team of pursuers to herd a flock of evaders [102–106]. Few multi agent herding results consider pursuers making coordinated actions to intercept and corral multiple evaders. An obvious approach to multi-agent herding of a flock of evaders would be to position each pursuer on the perimeter of the flock in such a way that the indirect control of the pursuers on the flock keeps them contained within the encircled pursuers [91, 105, 106]. This approach was used in [106] and [105], where the approach, referred to as "herding by caging," was influenced from robotic grasping literature. In [106], the caging verification problem for a fixed number of pursuers was addressed to demonstrate how a sufficiently large team of pursuers can surround, and "cage", a group of evaders. The authors of [105], extended the result from [106], and presented algorithms to optimize the time to compute a valid cage formation and to minimize the number of pursuer robots while prioritizing caging formations. These results focus on computing cage formations that are considered complete or safe. The dynamics and control of the formation is not considered. Additionally, and the stability of the system is not addressed [105]. Furthermore, inter-evader interactions are not considered, allowing for collisions between evaders. These caging methods assume that once the cage formation encompasses the evaders, then the cage is a static object [105, 106]. To move the cage to the goal, the authors show the cage is robust to perturbations, and by applying sequentially small perturbations the herders can move the formation slowly to follow the desired path [105, 106].

In [91], a similar containment approach to the caging method, called StringNet, was shown to be capable of encircling a flock and regulating it to a goal. StringNet is formed by line segments between pursuers, and evaders are modeled to sense the strings when within a detectable region. The result in [91] requires the pursuers to approach

the flock in an arc formation, and the pursuer at each end of the arc would move around the flock in opposite directions to form a new edge. This approach ultimately turns the pursuers graph representation from a line graph to a cycle graph. Once the flock is encircled by the StringNet, the pursuers would herd the flock to a goal region. The StringNet result was further developed in [102] where the desired pursuer formation positions and path was determined with a mixed integer quadratic program (MIQP). This MIQP minimized the distance (and travel time) to the formation for each pursuer. The results in [90, 91, 102], differ from [105, 106] by the fact that the formation of evaders isn't considered to be static configuration, but treated as a dynamic system.

In contrast to the encirclement of evaders by pursuers, [104] proposed that a flock could be controlled at numerous "steering points" along the perimeter of a flock. As long as the pursuers were able to stay at the steering points, then with coordination between pursuers, the flock could be controlled to a goal location. Several methods to assign each pursuer to a steering point were presented, in particular the *Vector Projection* method was introduced. The *Vector Projection* method requires each pursuer to count the number of pursuers on its left or right side, if there's $k-1$ pursuers on the left (or right), then the shepherd will got to the k-th steering point from the left (or right). The efficiency of this assignment technique was compared to other common methods (global distance minimization and greedy distance minimization) . Finally, [104] uniquely considered the case where separate flocks could be combined if pursuers are able to drive the two flocks sufficiently close. In [107], nuanced inter-agent behaviors were considered. The evaders were assumed to act as a flock but each evader could take a "random action" that would disrupt an individual evader's flocking behavior. Additionally, a repulsive force between herders to avoid overlapping herding areas was included. It was shown that a single herder and a pair of herders could move the flock to goal location. Unique to the presented approach, the indirect control of the flock by the herders was modeled using a heuristic that used the orientation of the pursuer(s).

A number of different strategies to assign each herder in a team to an evader were presented and compared in [108]. These strategies require communication between herders to deconflict evader assignments, incorporating coordination between pursuers. When a pursuer is assigned a target, the pursuer intercepts the evader and drives it to a goal region by indirectly controlling it. The pursuer(s) will repeat the assignment and herding phases until all evaders are inside the goal.

The approach in [93], addressed the same problem formulation as [107] and [108], but presented a closed-loop feedback control strategy instead of a heuristic or behavior result. The authors assumed the pursuer/evader assignment had been completed, and the authors considered the case where the maximum velocity of the pursuers was equal to that of the evaders. A Lyapunov-based analysis was provided to show the proposed strategy was globally asymptotically stable. This stability result led the authors to adopt an initial approach phase that the pursuers must complete to avoid the evaders dispersing and becoming unreachable. It was apparent in the provided simulations that the pursuers initial state had to be sufficiently far away from the evaders. The case where a pursuer was initially amongst the evaders was not presented. This result was further developed in [94] by including an adaptation law to address uncertainty in the evader's model.

The authors of [78] and [90] presented a control-methods approach to address the multiple pursuers intercepting and herding a single evader. In [90], a finite-time stabilizing, state-feedback controller is presented to regulate each pursuer into a desired formation about the evader while avoiding inter-agent collisions and collisions with obstacles. In [78], the authors presented a neural network-based estimation scheme to approximate the influence dynamics between the pursuer and evader. The developed controller for the pursuers makes use of these estimates to guarantee that the interception and herding objectives are achieved. The stability of the closed loop error system was shown to be UUB by using a Lyapunov-based analysis. The results

in [79, 80, 109] use multiple pursuers with a pursuer-evader interaction known *a priori* to herd multiple evaders to a desired location as a group. The general approach adopted in these results is to control the pursuers to create an arc formation that lies on a circle about the centroid of the evaders' positions and treating the resulting pursuer-evader system as a constrained unicycle-like object. The arc formation of pursuers is controlled to align the direction of the formation towards the desired goal, and the pursuers adapt the curvature of the arc and the spacing along the arc (inter-pursuer edge lengths) in order to herd the evaders to the goal.

1.5 Outline of Dissertation

Chapter 2 explores the approximate consensus problem for a multi-agent system under intermittent communication. To accommodate the limits of digital communication hardware, an event-triggered controller is presented that satisfies a minimum inter-event time constraint as specified by the user. The proposed event-triggered controller is bounded *a priori* via the exploitation of a projection operator that yields workspace invariance, which can be leveraged to promote safety. The consensus problem is posed as a set stabilization problem and, using a Lyapunov-based stability analysis for hybrid dynamical systems, a ν-expansion of the agreement subspace is shown to be asymptotically stable. Specifically, by increasing the region within which the consensus criteria is satisfied, an asymptotically stable control design is achieved.

Chapters 3-4 investigate unique indirect control problems motivated by sensor or communication limitations. Chapter 3 explores the indirect control problem for a single pursuer agent regulating a single target agent to a goal location. To accommodate the constraints of sensing hardware, an event-triggered inter-agent influence model between the pursuer agent and target agent is considered. Motivated by fielded sensing systems, an event-triggered controller and trigger mechanism are presented that satisfy a user-selected minimum inter-event time. The combined pursuer-target system is modeled as a switched system that alternates between stable and unstable modes. A

dwell-time analysis is completed to develop a closed-form solution for the maximum time the pursuer agent can allow the target agent to evolve in the unstable mode before requiring a control input update. The presented trigger function is designed and shown to produce inter-event times that are upper-bounded by the maximum dwell time.

Chapter 4 considers an indirect control problem for a single pursuer agent regulating a single target agent where inter-agent interactions can only occur within the FOV of the target agent. By formulating the system as a switched dynamical model with stable and unstable subsystems, Chapter 4 presents a comprehensive dwell-time analysis to create a switching controller that can regulate a constant speed target agent toward a desired trajectory.

1.6 Notation

Specific notation is used throughout this dissertation to represent mathematical concepts. To facilitate the subsequent development, the most common notations are described in this section. The set of real numbers and integers are represented by $\mathbb{R}$ and $\mathbb{Z}$, respectively. The set of positive and strictly positive real numbers greater than $z \in \mathbb{R}$ are represented by $\mathbb{R}_{\geq z} \in [z, \infty)$ and $\mathbb{R}_{>z} \in (z, \infty)$. Similarly, the set of positive and strictly positive integers greater than $z \in \mathbb{R}$ are denoted by $\mathbb{Z}_{\geq z} \triangleq \mathbb{R}_{\geq z} \cap \mathbb{Z}$ and $\mathbb{Z}_{>z} \triangleq \mathbb{R}_{>z} \cap \mathbb{Z}$. Consider the sets A, B, the mapping f from A to values of B is denoted, $f : A \to B$. The set-valued mapping f from A to B is denoted $f : A \rightrightarrows B$ [110]. The $c \in \mathbb{R}$ sub-level and super-level sets of a real-valued function f on a set A and is denoted by $[f \leq c]$ and $[f \geq c]$, respectively [110]. The p^{th} standard basis vector is denoted $\mathsf{e}_p \in \mathbb{R}^N$. The Euclidean norm of a vector $u \in \mathbb{R}^n$ is denoted by $\|u\| \triangleq \sqrt{u^\top u}$. The infinity norm of a vector $u \triangleq (u_1, u_2, \ldots, u_n)$, where $u_i \in \mathbb{R}^N$ and $u \in \mathbb{R}^{nN}$, is denoted $\|u\|_\infty \triangleq \max\{\|u_1\| : i \in S\}$, where $S \triangleq [1, 2, ..., N]$. The inner product of two vectors $u, v \in \mathbb{R}^n$ is denoted $\langle u, v \rangle \triangleq u^\top v$. The Kronecker product between $A \in \mathbb{R}^{n \times m}$ and $B \in \mathbb{R}^{p \times q}$, is represented by $A \otimes B \in \mathbb{R}^{np \times mq}$. The identity matrix, of size $n \times n$, is denoted I_n. Let $1_n, 0_n \in \mathbb{R}^n$ represent the column vectors with all elements being one

and zero, respectively. Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$, the i^{th} eigenvalue of A, in non-decreasing order, is denoted by $\lambda_i(A) \in \mathbb{R}$. Furthermore, let $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ denote the minimum and maximum eigenvalues of A. Given a set $\mathcal{A} \subset \mathbb{R}^n$ and a point $x \in \mathbb{R}^n$, the distance between x and $\mathcal{A}$ is denoted $|x|_{\mathcal{A}} \triangleq \inf\|x-y\| : y \in \mathcal{A} \in \mathbb{R}_{\geq 0}$ [38]. The complement and orthogonal complement of a set $\mathcal{A}$ are denoted $\mathcal{A}^c$ and $\mathcal{A}^{\perp}$, respectively [38]. The component of $x \in \mathbb{R}^{nN}$, denoted $x^{\perp}$, in $\mathcal{A}^{\perp}$ is computed using $x^{\perp} = \mathbf{S}x$, where the linear projection operator $\mathbf{S} \triangleq \left(I_N - \frac{1}{N}1_N 1_N^{\top}\right) \otimes I_n \in \mathbb{R}^{nN}$ [38]. For $N \in \mathbb{Z}_{\geq 1}$, let $[N] = \{1, 2, \ldots, N\}$ [38]. For $x_p \in \mathbb{R}^n$ and $p \in [N]$, the stacked vector $x \triangleq \left[x_1^{\top}, x_2^{\top}, \ldots, x_N^{\top}\right] \in \mathbb{R}^{nN}$ can be written as $x \triangleq (x_p)_{p\in\mathcal{V}}$ [38].

1.6.1 Graphs

This section introduces the specific notation for graph theory used throughout this dissertation, providing a consistent framework for the discussion and analysis of graph-related concepts. Let $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$ denote an undirected and connected graph. The vertex set in $\mathcal{G}$ is denoted $\mathcal{V} \triangleq [N]$, for $N \in \mathbb{Z}_{>1}$, and the vertices in $\mathcal{G}$ are called nodes [1]. The edge set in $\mathcal{G}$ is $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, where an edge exists between two nodes if information can be exchanged [1]. Recall, a path exists between nodes $p, q \in \mathcal{V}$, if there is a sequence of distinct vertices such that $(v_0 = p, \ldots, v_k = q)$, for $k \in \mathbb{Z}_{\geq 0}$, $(v_{s-1}, v_s) \in \mathcal{E}$, and $s \in [k]$ [17]. The neighbor set of node p is denoted $\mathcal{N}_p \triangleq \{q \in \mathcal{V} \setminus \{p\} : pq \in \mathcal{E}\}$, and represents the set of nodes that have an edge connected to node p [1]. Additionally, the adjacency matrix of $\mathcal{G}$ is denoted $\mathscr{A} \triangleq [a_{pq}] \in \mathbb{R}^{N\times N}$, where $a_{pq} = 1$ if and only if $(p, q) \in \mathcal{E}$; otherwise, $a_{pq} = 0$ [1]. Since edges describe information links between nodes, self loops are not considered, i.e., $a_{ii} \triangleq 0$ for all $i \in \mathcal{V}$ [9]. The degree matrix of $\mathcal{G}$ is denoted $\Delta \triangleq \text{diag}\,(\mathscr{A} \cdot \mathbf{1}_N) \in \mathbb{R}^{N\times N}$ [1]. The Laplacian matrix is denoted $L \triangleq \Delta - \mathscr{A} \in \mathbb{R}^{N\times N}$ [1]. The Laplacian of a complete graph on N nodes is denoted $L_C \in \mathbb{R}^{N\times N}$ [9].

1.6.2 Hybrid Dynamical Systems

This section outlines the notation adopted for hybrid dynamical systems, ensuring coherence and clarity in the formulation and discussion of the ensuing material. A hybrid

system $\mathcal{H}$ with data (C, f, D, G) is defined as

$$\mathcal{H}: \begin{cases} \dot{z} = f(z), & z \in C, \\ z^{+} \in G(z), & z \in D, \end{cases}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ denotes the single-valued flow map that models continuous behavior, $C \subseteq \mathbb{R}^n$ denotes the flow set over which the system continuously evolves, $G : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ denotes the set-valued jump map that models discrete behavior, and $D \subseteq \mathbb{R}^n$ denotes the jump set over which the system discretely evolves [110]. Given a set-valued mapping $H : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$, the domain of H is the set dom $H \triangleq \{x \in \mathbb{R}^m : H(x) \neq \varnothing\}$ [110, Definition 2.1]. A solution ϕ to the hybrid system $\mathcal{H}$ is parameterized with respect to hybrid-time as denoted by $(t, j) \in \mathbb{R}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ [110]. Observe that t and j represent continuous-time and discrete-time, respectively [110]. The domain dom$\phi \subset \mathbb{R}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ is called a hybrid time domain if for all $(T, J) \in$ dom ϕ, dom $\phi \cap ([0, T] \times \{0, 1, ..., J\})$ can be expressed as $\cup_{j=0}^{J}(I_j \times \{j\})$, where $I_j \triangleq [t_j, t_{j+1}]$ for a time sequence $0 = t_0 \leq t_1 \leq ... \leq t_{J+1}$ [110]. Note that t_j indicates the j^{th} instant the state z jumps [110]. A solution ϕ to $\mathcal{H}$ is called maximal if ϕ cannot be extended, that is, ϕ is not a truncated version of another solution [110]. A solution is called complete if its domain is unbounded [110]. The set $\mathcal{S}_{\mathcal{H}}$ contains all maximal solutions to $\mathcal{H}$, where the set $\mathcal{S}_{\mathcal{H}}(\xi)$ contains all maximal solutions to $\mathcal{H}$ with initial condition ξ [110]. A hybrid system $\mathcal{H}$ with data (C, f, D, G) is said to satisfy the *hybrid basic conditions* if it satisfies the conditions in [110, Assumption 6.5]. A switched system is generally described as a continuous-time system with discrete switching events [111]. A switched system can be viewed as a hybrid system where the discrete behavior is reduced to a class of switching patterns [111]. A hybrid arc ϕ is a function $\phi :$ dom $\phi \to \mathbb{R}^n$, where $x(\phi)$ is used to represent the value of the solution $\phi(t, j)$ to $\mathcal{H}$. Further discussion on relevant notation can be found in references such as [9, 38, 110, 112].

CHAPTER 2
MULTI-AGENT APPROXIMATE CONSENSUS WITH A USER-SPECIFIED MINIMUM INTER-EVENT TIME

This work explores the approximate consensus problem for a multi-agent system under intermittent communication. To accommodate the limits of digital communication hardware, an event-triggered controller is presented that satisfies a minimum inter-event time constraint as specified by the user. The developed event-triggered controller is bounded *a priori* via the exploitation of a projection operator that yields workspace invariance, which can be leveraged to promote safety. The consensus problem is posed as a set stabilization problem and, using a Lyapunov-based stability analysis for hybrid dynamical systems, a ν-expansion of the agreement subspace is shown to be asymptotically stable, i.e., by increasing the region within which consensus criteria is satisfied an asymptotically stable control design is achieved. A simulation example of a multi-agent system conducting approximate consensus in the 2-D plane with a user-defined minimum inter-event time is presented to illustrate the contribution of this work.

2.1 Problem Formulation

Consider a MAS of $N \in \mathbb{Z}_{\geq 2}$ agents. The single-integrator dynamics of agent $p \in \mathcal{V}$ is

$$\dot{x}_p = u_p, \tag{2–1}$$

where $x_p \in \mathbb{R}^n$ and $u_p \in \mathbb{R}^n$ denote the state and control input of the agent, respectively. More complex dynamics can be considered but for ease of exposition, single integrator dynamics are used. This contribution only requires the control input to be bounded since single integrator dynamics are employed. If higher order dynamics were incorporated, the agent dynamics, along with the control input, would have to be bounded. For $\dot{x}_p = f(x_p) + u_p$, an upper bound $\bar{f}$ on $f(x_p)$ and a bounded u_p must exist to bound $\dot{\mathbf{T}}_p$ as stated in Definition 2.1, if $\bar{f}$ exists, and u_p is given in (2–6), then (2–14) becomes $\mathfrak{m}_p = 4rk_1\sqrt{\mathfrak{h}_p}\left(\bar{f} + rk_1N + rk_1\left|\mathcal{N}_p\right|\right)\left|\mathcal{N}_p\right|$. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected, undirected, and

static graph modeling a communication network that links the agents of the ensemble. To reduce the amount of network resources needed to support information sharing between agents, an event-triggered control strategy is adopted.

The objective of this work is to design a distributed event-triggered controller for each agent $p \in \mathcal{V}$ that enables the MAS to achieve ν-approximate consensus, as defined is [38], while accommodating the sampling limit of the onboard hardware. In particular, given a user-defined sampling limit $\tau_{\min} \in \mathbb{R}_{>0}$, the minimum inter-event time of an admissible trigger $\mathbf{T}_p$ must satisfy $\tau_p \geq \tau_{\min}$ for each agent $p \in \mathcal{V}$.

2.2 Hybrid System Modeling

Let $k_1 \in \mathbb{R}_{>0}$ be a user-defined parameter and $\eta_p \in \mathbb{R}^n$ be the zero-order hold sampled consensus error for agent $p \in \mathcal{V}$, which evolves according to the hybrid system

$$\begin{aligned} \dot{\eta}_p &= 0_n, & \mathbf{T}_p(\xi) \geq 0 \\ \eta_p^+ &= k_1 \sum_{q \in \mathcal{N}_p} (x_q - x_p), & \mathbf{T}_p(\xi) \leq 0. \end{aligned} \tag{2–2}$$

A solution to the hybrid system in (2–2) yields a sample-and-hold approximation of the distributed consensus controller in [52, Equation A1]. To facilitate the analysis, consider the error between the zero-order hold consensus error and an estimated consensus error,

$$\tilde{\eta}_p \triangleq \eta_p - \hat{\eta}_p \in \mathbb{R}^n \tag{2–3}$$

where $\hat{\eta}_p \triangleq k_1 \sum_{q \in \mathcal{N}_p} (\hat{x}_q - \hat{x}_p)$ is a continuously computed estimate of the consensus error. Using (2–3), the trigger function of agent p is selected as

$$\mathbf{T}_p(\xi) \triangleq \mathfrak{h}_p - \|\tilde{\eta}_p\|^2. \tag{2–4}$$

Let $x \triangleq (x_p)_{p \in \mathcal{V}} \in \mathbb{R}^{nN}$ and $\eta \triangleq (\eta_p)_{p \in \mathcal{V}} \in \mathbb{R}^{nN}$ denote the ensemble form of the state and sampled consensus error of the MAS, then $\xi \triangleq [x^\top, \eta^\top]^\top \in \mathcal{X}$ and $\mathcal{X} \triangleq \mathbb{R}^{nN} \times \mathbb{R}^{nN}$ represents the state variable and state space of the hybrid system for the MAS, respectively.

Definition 2.1. A trigger function $\mathbf{T}_p(\xi)$ is said to be *admissible* if it satisfies the following properties[1] :

1. $\mathbf{T}_p$ is a differentiable function of ξ;
2. $\dot{\mathbf{T}}_p \geq -\mathfrak{m}_p$ for some constant $\mathfrak{m}_p \in \mathbb{R}_{>0}$;
3. If ξ jumps in response to $\mathbf{T}_p(\xi) \leq 0$, then $\mathbf{T}_p^+ = \mathfrak{h}_p$ for some constant $\mathfrak{h}_p \in \mathbb{R}_{>0}$.

Under (2–2)–(2–4), a jump for agent p may occur only when $\mathbf{T}_p(\xi) \leq 0$. If a jump occurs in response to $\mathbf{T}_p(\xi) \leq 0$, then (2–2)–(2–4) imply that $\tilde{\eta}_p^+ = 0_n$ and $\mathbf{T}_p^+ = \mathfrak{h}_p$. Furthermore, if a jump occurs in response to $\mathbf{T}_p(\xi) \leq 0$ only, then $\tilde{\eta}_q^+ = \tilde{\eta}_q$ and $\mathbf{T}_q^+ = \mathbf{T}_q$ for all $q \in \mathcal{V} \setminus \{p\}$. Specifically, when a jump occurs, agent p samples the relative displacements to its neighbors, updates η_p, and transmits $[x_p^\top, \eta_p^\top]^\top$ to its neighbors.

Under (2–2) and (2–3), the trigger function $\mathbf{T}_p$ in (2–4) satisfies Items 1) and 3) of Definition 2.1. The only property left to satisfy is Item 2), i.e., uniformly lower bounding the trigger's time derivative. Observe that

$$\dot{\mathbf{T}}_p = -2\tilde{\eta}_p^\top \dot{\tilde{\eta}}_p = 2k_1 \tilde{\eta}_p^\top \sum_{q \in \mathcal{N}_p} (u_q - u_p), \tag{2–5}$$

which is obtained by substituting (2–1), the flow equation in (2–2), and the time derivative of (2–3) into the time derivative of $\mathbf{T}_p$ in (2–4). As a result, $\dot{\mathbf{T}}_p$ may be bounded as in Item 2) provided $\tilde{\eta}_p$ and $\{u_q\}_{q \in \mathcal{V}}$ are bounded. One way to design controllers that are bounded *a priori* is by leveraging a projection operator (see [113]). Boundedness can be achieved via the forward invariance of compact sets.

Let $\mathrm{proj}\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ denote the Lipschitz continuous projection operator in [113, Equation 4], which has parameters $(\theta_0, \varepsilon) \in \mathbb{R}_{>0} \times \mathbb{R}_{>0}$. By [113, Property 1], the dynamical system $\frac{\mathrm{d}}{\mathrm{d}t}\hat{\theta} = \mathrm{proj}(\mu, \hat{\theta})$, with $(\hat{\theta}, \mu) \in \mathbb{R}^n \times \mathbb{R}^n$ and initial condition $\hat{\theta}(0) \in \mathbb{B}(\theta_0) \triangleq \{s \in \mathbb{R}^n \colon \|s\| \leq \theta_0\}$, renders the closed ball $\mathbb{B}(\theta_0 + \varepsilon)$ forward invariant.

[1] The symbol ξ represents the state variable of the hybrid system for the MAS which is presented in Section 2.2.

Let $\Omega \triangleq \{z \in \mathbb{R}^n \colon \|z - c\| \leq r\}$ represent the workspace of all agents in the MAS, where $c \in \mathbb{R}^n$ and $r \triangleq \theta_0 + \varepsilon \in \mathbb{R}_{>0}$ are user-defined constants. Further, consider the Lipschitz function $h \colon \mathbb{R}^n \to \mathbb{R}^n \colon s \mapsto s - c$. Using the projection operator $\text{proj}(a, b)$ and the translation h, the controller of agent p is

$$u_p \triangleq \text{proj}\left(\eta_p, h\left(x_p\right)\right). \tag{2–6}$$

The projection operator from [113] is

$$\text{proj}\left(\eta_p, h\left(x_p\right)\right) =$$

$$\begin{cases} \eta_p, & \text{if } p_l\left(h\left(x_p\right)\right) \leq 0 \\ \eta_p, & \text{if } p_l\left(h\left(x_p\right)\right) \geq 0 \\ & \text{and } \nabla p_l\left(h\left(x_p\right)\right)^\top \left(h\left(x_p\right)\right) \leq 0 \\ \left(I_n - \Phi\left(h\left(x_p\right)\right)\right)\eta_p, & \text{otherwise.} \end{cases} \tag{2–7}$$

In (2–7), $\Phi\left(h\left(x_p\right)\right) \triangleq \frac{p_l(h(x_p))\nabla p_l(h(x_p))\nabla p_l(h(x_p))^\top}{\nabla p_l(h(x_p))^\top \nabla p_l(h(x_p))}$ and $p_l\left(h\left(x_p\right)\right) \triangleq \frac{h(x_p)^\top (h(x_p)) - \|\theta_0\|^2}{\varepsilon^2 + 2\varepsilon\|\theta_0\|}$.

Substituting (2–6) into (2–1) yields $\dot{x}_p = \text{proj}(\eta_p, h(x_p))$, where η_p is piece-wise constant during flows. A continuous state must be consistently measured for the trigger mechanism to evolve in time and satisfy the event-trigger conditions. Since single-integrator dynamics[2] are considered, if agent $q \in \mathcal{N}_p$ broadcasts (x_q, η_q) whenever a jump occurs in response to $\mathbf{T}_q(\xi) \leq 0$, then agent p can use the dynamics $\dot{x}_q = \text{proj}(\eta_q, h(x_q))$ with initial condition (x_q, η_q) to compute the trajectory of agent q during a flow interval for each agent $q \in \mathcal{N}_p$. Additionally, agent p can compute x_p utilizing $\dot{x}_p = \text{proj}(\eta_p, h(x_p))$ and initial condition (x_p, η_p), such that (x_p, η_p) can be computed instead of continuously measured, which also satisfies the requirement

[2] Using higher order dynamics to model the agent states would require that either $x_q - x_p$ or x_p be continuously measured by agent p, consistent with ETC literature (e.g., [5, 7, 9, 17, 38, 53, 65, 114]).

to achieve a continuously evolving trigger mechanism. The computed trajectories $\{x_q \colon q \in \mathcal{N}_p \cup \{p\}\}$ enable the computation of $\mathbf{T}_p(\xi)$ without requiring continuous measurements of x_q or $x_q - x_p$ for all $q \in \mathcal{N}_p \cup \{p\}$. Therefore the estimate $\hat{\eta}_p$ in (2–3) closely tracks the actual consensus error since the system is disturbance free; hence $\hat{x}_p = x_p$ and $\hat{x}_q = x_q$.

Remark 2.1. Observe that $h(x_p) \in \mathbb{B}(\theta_0 + \varepsilon)$ if and only if $x_p \in \Omega$. Also, the substitution of $\dot{x}_p = \mathsf{proj}(\eta_p, h(x_p))$ into the time derivative of $h(x_p)$ yields $\frac{\mathrm{d}}{\mathrm{d}t} h(x_p) = \mathsf{proj}(\eta_p, h(x_p))$. Thus, if $x_p(0) \in \{z \in \mathbb{R}^n \colon \|z - c\| \leq \theta_0\} \subset \Omega$, then $h(x_p(0)) \in \mathbb{B}(\theta_0 + \varepsilon)$ and the dynamics of $h(x_p)$ renders $\mathbb{B}(\theta_0 + \varepsilon)$ forward invariant by [113, Property 1], implying that $x_p(t) \in \Omega$ for all $t \geq 0$. In other words, x_p is bounded provided $x_p(0) \in \{z \in \mathbb{R}^n \colon \|z - c\| \leq \theta_0\}$.

Let $\mathcal{H}$ represent the hybrid system for the MAS with state variable ξ and state space $\mathcal{X}$. Using the trigger function $\mathbf{T}_p$ in (2–4), the flow and jump sets of $\mathcal{H}$ are

$$C \triangleq \bigcap_{p \in \mathcal{V}} [\mathbf{T}_p \geq 0], \quad D \triangleq \bigcup_{p \in \mathcal{V}} [\mathbf{T}_p \leq 0], \tag{2–8}$$

respectively. To assist the development, let $D_p \triangleq [\mathbf{T}_p \leq 0]$ denote the jump set corresponding to agent p and $\mathsf{proj}(\eta, x) \triangleq (\mathsf{proj}(\eta_p, h(x_p)))_{p \in \mathcal{V}} \in \mathbb{R}^{nN}$ be a concatenation of all controllers in the MAS. Substituting (2–1), the flow equation in (2–2), and the controller in (2–6) for every $p \in \mathcal{V}$ into the time derivative of ξ yields $\dot{\xi} = f(\xi)$, where the single-valued flow map $f \colon \mathcal{X} \to \mathcal{X}$ is

$$f(\xi) \triangleq \begin{bmatrix} \mathsf{proj}(\eta, x) \\ 0_{nN} \end{bmatrix}. \tag{2–9}$$

Recall that x_p evolves according to the continuous-time system in (2–1). Therefore, x_p is mapped to itself during jumps, that is, $x_p^+ = x_p$ in response to a jump caused by $\mathbf{T}_q(\xi) = 0$ for any $q \in \mathcal{V}$. Since $x_p^+ = x_p$ for all $p \in \mathcal{V}$, then $x^+ = x$ from Definition 2.1. Next, recall that η_p evolves according to the hybrid system in (2–2), where $\eta_p^+ = k_1 \sum_{q \in \mathcal{N}_p} (x_q - x_p)$ if and only if a jump is triggered by $\mathbf{T}_p(\xi) = 0$. Consequently, the set-valued jump map of

$\mathcal{H}$ is $G\colon \mathcal{X} \rightrightarrows \mathcal{X}$ with

$$G(\xi) \triangleq \{G_p(\xi) : \xi \in D_p \text{ for some } p \in \mathcal{V}\},$$
$$G_p(\xi) \triangleq \begin{bmatrix} x \\ \left[\eta_1^\top, ..., \eta_{p-1}^\top, (\eta_p^+)^\top, \eta_{p+1}^\top, ..., \eta_N^\top\right]^\top \end{bmatrix}. \tag{2–10}$$

Upon inspection, the flow and jump sets are closed, the single-valued flow map is continuous, and the set-valued jump map is outer semi-continuous and locally bounded. For these reasons, the hybrid system $\mathcal{H}$ with data (C, f, D, G) satisfies the hybrid basic conditions in [110, Assumption 6.5], implying that $\mathcal{H}$ is nominally well-posed by [110, Theorem 6.8]. Let

$$\mathcal{A} \triangleq \left\{\xi \in \mathcal{X}\colon \|x^\perp\| = 0\right\} = \mathrm{A} \times \mathbb{R}^{nN}, \tag{2–11}$$

and observe that the distance between ξ and $\mathcal{A}$ is $|\xi|_{\mathcal{A}} = \|x^\perp\|$. With respect to the objective, if we can show that all maximal solutions of $\mathcal{H}$ converge to within a distance of ν from the set $\mathcal{A}$, then the MAS achieves ν-approximate consensus.

Definition 2.2. A closed set $\mathcal{A}$ is said to be *practically asymptotically stable* (PAS) for a hybrid system $\mathcal{H}$ if there exists a constant $\nu \in \mathbb{R}_{>0}$ and $\mathcal{KL}$ function β such that every maximal solution ϕ to $\mathcal{H}$ satisfies the following for all $(t, j) \in$ dom ϕ,

$$|\phi(t, j)|_{\mathcal{A}} \leq \beta\left(|\phi(0, 0)|_{\mathcal{A}}, t + j\right) + \nu.$$

2.3 Supporting Lemmas

Given a maximal solution ϕ of the hybrid system $\mathcal{H}$, the following result supports the analysis of $\phi(t, j)$ for arbitrarily large $t + j \in \mathbb{R}_{\geq 0}$ such that $(t, j) \in$ dom ϕ.

Lemma 2.1. Every maximal solution ϕ of the hybrid system $\mathcal{H}$ with data (C, f, D, G) is complete.

Proof. The single-valued flow map f in (2–9) is Lipschitz continuous since the proj(a, b) is a Lipschitz continuous function by [113, Property 4] and the composition of Lipschitz

continuous functions is itself Lipschitz continuous. Given $\dot{\xi} = f(\xi)$ with initial condition $\xi_0 = \phi(0,0) \in \overline{C} \setminus D$, there exists a nontrivial maximal solution ϕ for $\mathcal{H}$ that satisfies the initial condition. Since f is Lipschitz continuous, Case (b) in [110, Proposition 2.10] does not occur. Furthermore, since $G(D) \subset C \cup D$ under the construction of $\mathcal{H}$, Case (c) in [110, Proposition 2.10] does not occur. Therefore, ϕ is complete. ■

Definition 2.3. Let $\mathbf{T}_p$ be an admissible trigger function as defined in Definition 2.1, and let $\{t_k^p\}_{k=0}^{\infty}$ be an increasing sequence of event times for agent p, where t_k^p denotes the k^{th} instant ξ jumped in response to $\mathbf{T}_p(\xi) \leq 0$. The positive difference between consecutive event times, $t_{k+1}^p - t_k^p$, is uniformly bounded away from zero (see [9, Theorem 1]). Specifically, for all $k \in \mathbb{Z}_{\geq 0}$, $t_{k+1}^p - t_k^p \geq \tau_p$, where

$$\tau_p \triangleq \frac{\mathfrak{h}_p}{\mathfrak{m}_p}. \tag{2–12}$$

The parameter τ_p in (2–12) is henceforth referred to as the *minimum inter-event time of* $\mathbf{T}_p$.

The next result states that the hybrid system $\mathcal{H}$ has admissible trigger functions. Specifically, Item $2)$ in Definition 2.1 is satisfied. To aid in the presentation of the result, let

$$T_{\min} \triangleq \min\{\tau_p \colon p \in \mathcal{V}\} \in \mathbb{R}_{>0} \tag{2–13}$$

denote the smallest minimum inter-event time of the MAS.

Lemma 2.2. Let ϕ be a maximal solution of the hybrid system $\mathcal{H}$. If the initial condition of $\mathcal{H}$ is selected to satisfy $\phi(0,0) \in \Phi \triangleq \{\xi \in \mathcal{X} \colon \forall_{p \in \mathcal{V}} \, \|x_p - c\| \leq \theta_0\}$, then the parameter $\mathfrak{m}_p$ may be selected as

$$\mathfrak{m}_p = 4rk_1^2\sqrt{\mathfrak{h}_p}(N + |\mathcal{N}_p|)|\mathcal{N}_p| \tag{2–14}$$

for every $p \in \mathcal{V}$. In addition,

$$T_{\min}\left(\frac{j}{N} - 1\right) \leq t \tag{2–15}$$

for all $(t,j) \in$ *dom* ϕ.

Proof. Fix a $p \in \mathcal{V}$, and recall

$$\dot{\mathbf{T}}_p = 2k_1 \tilde{\eta}_p^\top \sum_{q \in \mathcal{N}_p} (u_q - u_p) \,.$$

When $\phi \in C$, the definition of $\mathbf{T}_p$ in (2–4) and the definition of the flow set in (2–8) imply $\|\tilde{\eta}_p(\phi)\| \leq \sqrt{\mathfrak{h}_p}$. When $\phi \in D$, $\|\tilde{\eta}_p(\phi)\| \leq \sqrt{\mathfrak{h}_p}$ since maximal solutions of $\mathcal{H}$ are complete, that is, $\mathbf{T}_p(\phi(t,j)) \geq 0$ for all $(t,j) \in \text{dom } \phi$. Otherwise, the solution ϕ terminates prior to achieving $\mathbf{T}_p(\phi) < 0$, implying that ϕ was not maximal. Hence, $\|\tilde{\eta}_p(\phi(t,j))\| \leq \sqrt{\mathfrak{h}_p}$ for all $(t,j) \in \text{dom } \phi$. Since $\mathcal{H}$ renders Ω forward invariant given $\phi(0,0) \in \Phi$ and Remark 2.1, $\|x_p(\phi(t,j)) - c\| \leq r$ for all $(t,j) \in \text{dom } \phi$. By definition, η_p is a piece-wise continuous computation of $k_1 \sum_{q \in \mathcal{N}_p}(x_q - x_p)$ along ϕ, and $\|k_1 \sum_{q \in \mathcal{N}_p}(x_q - x_p)\| \leq 2rk_1|\mathcal{N}_p|$, η_p can be bounded as $\|\eta_p\| \leq 2rk_1|\mathcal{N}_p|$. By [113, Property 3], $\|u_p\| = \|\text{proj}(\eta_p, h(x_p))\| \leq \|\eta_p\|$. Therefore,

$$\begin{aligned} |\dot{\mathbf{T}}_p| &\leq 2k_1 \|\tilde{\eta}_p\| \sum_{q \in \mathcal{N}_p} (\|\eta_q\| + \|\eta_p\|) \\ &\leq 4rk_1^2 \sqrt{\mathfrak{h}_p}(N + |\mathcal{N}_p|)|\mathcal{N}_p|, \end{aligned}$$

and the result for $\mathfrak{m}_p$ follows.

Next, suppose $(t',j'), (t,j) \in \text{dom } \phi$ such that $0 \leq t - t' \leq T_{\min}$ and $0 \leq j - j'$. With respect to $\mathcal{H}$, at most N triggers may jump twice during a flow interval of length $T_{\min}$. Hence, the number of jumps between (t',j') and (t,j) is bounded as $j - j' \leq N((t-t')/T_{\min} + 1)$, which implies that

$$T_{\min}\left(\frac{j - j'}{N} - 1\right) \leq t - t'.$$

Setting (t',j') to $(0,0)$ leads to the desired result. ∎

Remark 2.2. Leveraging the definition in (2–12) and the equality in (2–14),

$$\tau_p = \frac{\sqrt{\mathfrak{h}_p}}{4rk_1^2(N + |\mathcal{N}_p|)|\mathcal{N}_p|}. \tag{2–16}$$

Consequently, for every $p \in \mathcal{V}$, the minimum inter-event time constraint $\tau_{\min} \leq \tau_p$ can be satisfied, given (2–16), through the selection of the parameters $\mathfrak{h}_p$, k_1, r with consideration to the system configuration governed by N and $|\mathcal{N}_p|$. Solving (2–16) for $\mathfrak{h}_p$ and setting $\tau_p = \tau_{\min}$ informs how a specified minimum inter-event time can be used to solve for the system parameters. The system parameters must also be selected based on the stability criteria discussed in the next section.

2.4 Stability Analysis

The following items are provided to streamline the presentation of the main result, given in Theorem 2.1. Let $\tilde{\eta} \triangleq (\tilde{\eta}_p)_{p\in\mathcal{V}} \in \mathbb{R}^{nN}$denote the concatenation of the sample-and-hold errors of the MAS. Recall the projection $\mathbf{S}$ from Section 1.6. Let $x^{\perp} = \mathbf{S}x = (s_p)_{p\in\mathcal{V}}$ for an appropriate collection of vectors $\{s_1, s_2, ..., s_N\} \subset \mathbb{R}^n$. Moreover, for each $p \in \mathcal{V}$,

$$\sum_{q\in\mathcal{N}_p} (x_q - x_p) = -\left(\mathsf{e}_p L \otimes I_n\right) x, \tag{2–17}$$

where e_p represents the p^{th} standard basis vector of $\mathbb{R}^N$ and L is the Laplacian matrix of the communication graph $\mathcal{G}$. Given a user-defined constant $\kappa \in \mathbb{R}_{>0}$, let

$$\mathsf{c}_1 \triangleq 2k_1\lambda_2\left(L\right) - \frac{1}{\kappa} \in \mathbb{R}, \quad \mathsf{c}_2 \triangleq \frac{\kappa}{2}\sum_{p\in\mathcal{V}} \mathfrak{h}_p \in \mathbb{R}. \tag{2–18}$$

Recall $\mathcal{G}$ is connected, undirected, and static. Thus, $\lambda_2(L) > 0$ along any solution ϕ of $\mathcal{H}$.

Theorem 2.1. For every agent $p \in \mathcal{V}$, if $k_1 > 0$, $\theta_0 > 0$, $\varepsilon > 0$, $\mathfrak{h}_p > 0$, $\mathfrak{m}_p$ satisfies (2–14), and $\kappa > 1/(2k_1\lambda_2(L))$, then $c_1 > 0$, $c_2 > 0$, and the set $\mathcal{A}$ is PAS[3] with respect to $\Phi \cap C$ for the hybrid system $\mathcal{H}$ with data (C, f, D, G). In particular, for every maximal solution ϕ

[3] The qualifier in PAS is restricted to the set of initial conditions in $\Phi \cap C$, enabling the use of Lemma 2.2 and the satisfaction of $\mathbf{T}_p(\phi(0,0)) \geq 0$ for each $p \in \mathcal{V}$ given any maximal solution ϕ of $\mathcal{H}$ with $\phi(0,0) \in \Phi \cap C$.

such that $\phi(0,0) \in \Phi \cap C$,

$$|\phi(t,j)|_{\mathcal{A}} \leq |\phi(0,0)|_{\mathcal{A}} \alpha_1 \exp(-\alpha_2 (t+j)) + \nu, \tag{2–19}$$

for any $(t,j) \in \mathit{dom}\ \phi$ and for some $\epsilon \in (0,1)$, where

$$\begin{aligned} \nu = \sqrt{\frac{c_2}{2c_1}},\ \alpha_1 \triangleq \exp\left(\frac{(1-\epsilon)c_1 T_{\min}}{2}\right), \\ \alpha_2 \triangleq \min\left\{\frac{\epsilon c_1}{2}, \frac{(1-\epsilon)c_1 T_{\min}}{2N}\right\}. \end{aligned} \tag{2–20}$$

Proof. Consider the Lyapunov-like function

$$V : \mathcal{X} \to \mathbb{R}_{\geq 0} : \xi \mapsto \frac{1}{2} x^\top \mathbf{S} x, \tag{2–21}$$

which is a continuously differentiable function of ξ by construction. Since $\mathbf{S}$ is idempotent, $V(\xi)$ can be equivalently written as

$$V(\xi) = \frac{1}{2} x^\top \mathbf{S}\mathbf{S} x = \frac{1}{2}\|x^\perp\|^2 = \frac{1}{2}|x|_{\mathcal{A}}^2. \tag{2–22}$$

When $\xi \in C$, the change in $V(\xi)$ is computed using $\dot{V}(\xi) = \langle \nabla V(\xi), f(\xi) \rangle$. Recall that $f(\xi)$ is the flow map of $\mathcal{H}$ provided in (2–9). Hence, the substitution of (2–9) into the time derivative of (2–21) yields

$$\dot{V}(\xi) = x^\top \mathbf{S} \mathsf{proj}(\eta, x). \tag{2–23}$$

By using $\tilde{\eta} = (\tilde{\eta}_p)_{p \in \mathcal{V}}$, the identity $\mathsf{proj}(-a,b) = -\mathsf{proj}(a,b)$, $x^\perp = \mathbf{S}x = (s_p)_{p \in \mathcal{V}}$, the sample-and-hold error in (2–3), [113, Property 2], and the expression in (2–17), the time derivative of $V(\xi)$ in (2–23) can be bounded as

$$\dot{V}(\xi) = \sum_{p \in \mathcal{V}} s_p^\top \mathsf{proj}\left(-\left(k_1 \left(\mathsf{e}_p L \otimes I_n\right) x - \tilde{\eta}_p\right), h(x_p)\right) \leq -k_1 x^\top \mathbf{S}\left(L \otimes I_n\right) x + x^\top \mathbf{S}\tilde{\eta}. \tag{2–24}$$

The time derivative of $V(\xi)$ in (2–24) can be bounded using Young's inequality and the definition of $L \otimes I_n = (L \otimes I_n)\mathbf{S}$ and $x^{\perp} = \mathbf{S}x$, as

$$\dot{V}(\xi) \leq -k_1 \lambda_2(L) \|x^{\perp}\|^2 + \frac{1}{2\kappa}\|x^{\perp}\|^2 + \frac{\kappa}{2}\|\tilde{\eta}\|^2. \tag{2–25}$$

Summing the triggers in (2–4) over all $p \in \mathcal{V}$ yields

$$\|\tilde{\eta}\|^2 = \sum_{p \in \mathcal{V}} (\mathfrak{h}_p - \mathbf{T}_p(\xi)). \tag{2–26}$$

Using (2–26), $\kappa > 0$, and the fact that $\mathbf{T}_p(\xi) \geq 0$ for all $p \in \mathcal{V}$ when $\xi \in C$, the inequality in (2–25) enables the derivation of

$$\dot{V}(\xi) \leq -\left(k_1 \lambda_2(L) - \frac{1}{2\kappa}\right) \|x^{\perp}\|^2 + \frac{\kappa}{2}\sum_{p \in \mathcal{V}} \mathfrak{h}_p. \tag{2–27}$$

Since $2V(\xi) = \|x^{\perp}\|^2$ given (2–22), the inequality in (2–27) can be equivalently written as

$$\dot{V}(\xi) \leq -c_1 V(\xi) + c_2. \tag{2–28}$$

When $\xi \in D$ and $g \in G(\xi)$, the change in $V(\xi)$ is computed using $V(g) - V(\xi)$. Since $V(\xi)$ only depends on x, which evolves continuously under $\mathcal{H}$, $V(\xi)$ evolves continuously as well. Therefore, $V(g) = V(\xi)$ whenever there is a jump. Let ϕ be a maximal solution of $\mathcal{H}$ with initial condition $\phi(0,0) \in \Phi$, and let $0 = t_0 \leq t_1 \leq ... \leq t_{j+1} \leq t$ satisfy

$$\text{dom}\, \phi \bigcap ([0, t_{j+1}] \times \{0, 1, ..., j\}) = \bigcup_{s=0}^{j} ([t_s, t_{s+1}] \times \{s\}).$$

Observe that $\phi(r,s) \in C$ for each $s \in \{0, 1, ..., j\}$ and almost all $r \in [t_s, t_{s+1}]$. In addition, (2–28) implies

$$\frac{d}{dr} V(\phi(r,s)) \leq -c_1 V(\phi(r,s)) + c_2. \tag{2–29}$$

Integrating (2–29) over all but finitely many $r \in [t_s, t_{s+1}]$ yields

$$V(\phi(t_{s+1}, s)) \leq V(\phi(t_s, s)) \exp(-c_1(t_{s+1} - t_s))$$

$$+\frac{c_2}{c_1}\left(1-\exp\left(-c_1\left(t_{s+1}-t_s\right)\right)\right). \tag{2–30}$$

Next, we have that $\phi(t_s, s-1) \in D$ for each $s \in \{0, 1, ..., j\}$, and $V(g) = V(\xi)$ implies

$$V\left(\phi\left(t_s, s\right)\right) - V\left(\phi\left(t_s, s-1\right)\right) = 0. \tag{2–31}$$

By inductively stitching (2–30) and (2–31) along the solution ϕ, it follows that

$$V\left(\phi\left(t, j\right)\right) \leq V\left(\phi\left(0, 0\right)\right)\exp\left(-c_1 t\right) + \frac{c_2}{c_1}\left(1-\exp\left(-c_1 t\right)\right). \tag{2–32}$$

For all $(t, j) \in$ dom ϕ, the substitution of $2V(\xi) = |x|_{\mathcal{A}}^2$ from (2–22) into (2–32) yields

$$\left|\phi\left(t, j\right)\right|_{\mathcal{A}} \leq \left|\phi\left(0, 0\right)\right|_{\mathcal{A}}\exp\left(-\frac{c_1}{2}t\right) + \sqrt{\frac{c_2}{2c_1}}. \tag{2–33}$$

Let $\epsilon \in (0, 1)$, and note that

$$-\frac{c_1}{2}t \leq -\min\left\{\frac{\epsilon c_1}{2}, \frac{(1-\epsilon)c_1 T_{\min}}{2N}\right\}(t+j) + \frac{(1-\epsilon)c_1 T_{\min}}{2}, \tag{2–34}$$

which is derived using (2–15). The substitution of (2–34) into (2–33) leads to the desired result in (2–19). ■

2.5 Discussion

The main unique result of this chapter is the development of a freely selectable minimum inter-event timing condition for an event-triggered ν-approximate consensus controller. A Lyapunov-based stability analysis indicates that a user cannot select an arbitrarily large minimum inter-event time without some loss of system performance. The stability analysis shows that both convergence speed and accuracy are affected by the choice of minimum inter-event time. In the discussion below, it is shown that k_1 is computed from the choice of $\tau_{\min}$ and if an arbitrarily large minimum inter-event time is desired, this result will support it, albeit with arbitrarily low convergence accuracy and speed. Additionally, the following discussion describes how to compute system gains

and parameters to achieve ν-approximate convergence performance while adhering to a prescribed minimum inter-event time.

Ideally, a user would select control parameters that yield a small value for ν and large value for τ_p with $p \in \mathcal{V}$, where $\limsup_{t+j\to\infty} |\phi(t,j)|_{\mathcal{A}} = \nu$ and τ_p is a lower bound on the minimum inter-event time of the trigger function $\mathbf{T}_p$ for agent $p \in \mathcal{V}$. However, the control parameters that satisfy the constraints $\tau_p \geq \tau_{\min}$ for each $p \in \mathcal{V}$ are not unique and their selection affects the value of ν, i.e., how close the MAS gets to consensus in the states $\{x_p\}_{p\in\mathcal{V}}$. Hence, a systematic method for selecting the controller parameters that minimize ν while satisfying $\tau_p \geq \tau_{\min}$ for each agent $p \in \mathcal{V}$ and the sufficient conditions of Theorem 2.1 is motivated.

Suppose the number of agents in the MAS and the undirected, connected, and static communication graph are specified, that is, N and $\mathcal{G}$ with $\lambda_2(L) > 0$ are fixed. To simplify the satisfaction of the sufficient conditions of Theorem 2.1, let $\mathfrak{h} = \mathfrak{h}_p$ for all $p \in \mathcal{V}$ and $\delta > 1$ be user-defined parameters. Furthermore, let

$$\kappa = \frac{\delta}{2k_1\lambda_2(L)}. \tag{2–35}$$

Substituting $\mathfrak{h} = \mathfrak{h}_p$, (2–18), and (2–35) into the equation for ν in (2–20) yields

$$\nu = \frac{\delta}{4k_1\lambda_2(L)}\sqrt{\frac{N\mathfrak{h}}{\delta-1}}. \tag{2–36}$$

Substituting $\mathfrak{h} = \mathfrak{h}_p$ into (2–16) yields

$$\tau_p = \frac{\sqrt{\mathfrak{h}}}{4rk_1^2\left(N + |\mathcal{N}_p|\right)|\mathcal{N}_p|} \tag{2–37}$$

for each $p \in \mathcal{V}$. Let $d_{\max} \triangleq \max\{|\mathcal{N}_p| \colon p \in \mathcal{V}\}$ denote the largest degree in $\mathcal{G}$, and observe that

$$\tau \triangleq \frac{\sqrt{\mathfrak{h}}}{4rk_1^2\left(N + d_{\max}\right)d_{\max}}. \tag{2–38}$$

Here, (2–38) shows that $\tau \geq \tau_{\min}$ which implies that $\forall_{p\in\mathcal{V}}\, \tau_p \geq \tau_{\min}$. Leveraging the sufficient condition in (2–38), let

$$k_1 = \sqrt{\frac{\sqrt{\mathfrak{h}}}{4r\tau_{\min}\left(N + d_{\max}\right) d_{\max}}}. \tag{2–39}$$

Substituting (2–39) into (2–36) yields

$$g(\mathfrak{h}, \delta) \triangleq \frac{\delta}{2\lambda_2\left(L\right)} \sqrt{\frac{r\tau_{\min}\sqrt{\mathfrak{h}} N \left(N + d_{\max}\right) d_{\max}}{\delta - 1}}, \tag{2–40}$$

where $g(\mathfrak{h}, \delta) \geq \nu$. The free variables in (2–40) are $\mathfrak{h}$ and δ, which can be selected by solving the following optimization problem:

$$\begin{aligned} &\text{minimize } g(\mathfrak{h}, \delta)^2 \\ &\text{subject to } (\mathfrak{h}, \delta) \in \mathbb{R}_{>0} \times \mathbb{R}_{>1}. \end{aligned} \tag{2–41}$$

2.6 Simulation Example

Numerical experiments demonstrate the efficacy of the event-triggered consensus controller in (2–2) and (2–6). In practice, the constant $\tau_{\min}$ is determined by the hardware of the MAS, e.g., the sampling time of the slowest sensor or processing time. For this simulation, $\tau_{\min} = 0.01$ is selected. The workspace parameters (i.e., c and $r = \theta_0 + \varepsilon$) can be selected according to the initial configuration of the MAS. In practice, initial configuration $x(0, 0)$, workspace parameters c and r would have to be computed in a distributed manner. That is, given $x(0, 0)$, the parameters c, θ_0, and ε can be selected to ensure $x(0, 0) \in \Omega$ and thus satisfy the sufficient condition of Lemma 2.2. The following parameters were selected $n = 2$, $N = 10$, $T_{\min} = 0.01$, $\varepsilon = 1.35$, $\delta = 2$, $\alpha_1 = 1$, and given a randomly generated initial configuration $x(0, 0)$ and graph $\mathcal{G}$, the following simulation were computed $\theta_0 = 13.54$, $r = 14.89$, $c = [4.81, 12.97]^\top$, $d_{\max} = 7$, $\lambda_2(L) = 0.91$, $\mathfrak{h} = 1 \times 10^{-5}$, $k_1 = 6.7 \times 10^{-3}$, $\nu = 0.82$, $\epsilon = 0.98$, $\kappa = 163.75$, $c_1 = 6.1 \times 10^{-3}$, $c_2 = 8.2 \times 10^{-3}$, and $\alpha_2 = 4.58 \times 10^{-8}$. Note that $\mathfrak{h}$ and δ were determined by minimizing $g^2(\mathfrak{h}, \delta)$ over $[1 \times 10^{-5}, +\infty) \times [1 + 1 \times 10^{-5}, +\infty)$. Solving the optimization

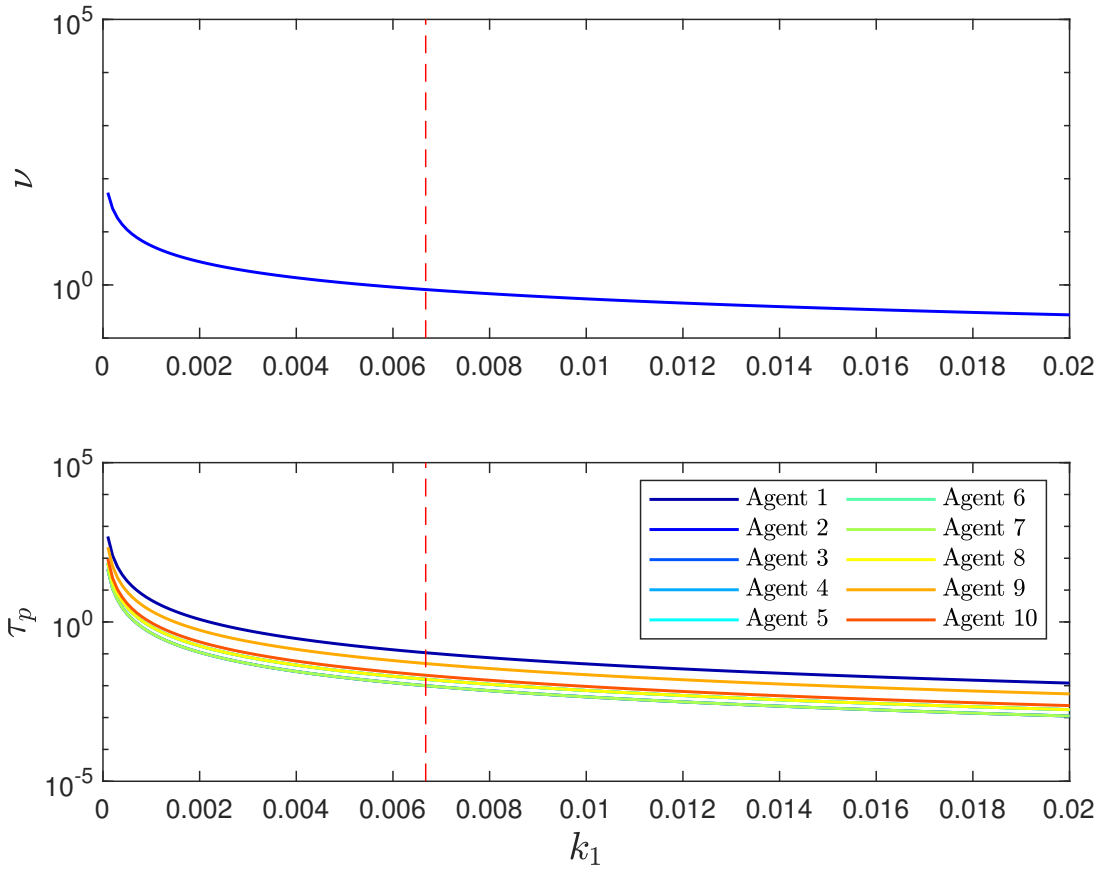

Figure 2-1. The top plot indicates larger values of k_1 lead to smaller minimum inter-event times and smaller values of ν, demonstrating the performance trade-off. The red dashed line depicts the selected value of $k_1 = 6.7 \times 10^{-3}$, which shows the compromise made in performance to satisfy a minimum inter-event time constraint.

problem in (2–41) would lead to $\nu = 0.1$ and $k_1 = 1.40 \times 10^{-4}$, which would substantially elongate the convergence time of $|\xi|_{\mathcal{A}}$ to 0. Hence, the domain of (2–41) was modified from $\mathbb{R}_{>0} \times \mathbb{R}_{>1}$ to $\mathbb{R}_{\geq 1\times 10^{-5}} \times \mathbb{R}_{\geq 1+1\times 10^{-5}}$ to shorten the simulation time. The trade-off between performance (i.e., small values of ν are better) and minimum inter-event times is summarized in Fig. 2-1. To help visualize the satisfaction of the minimum inter-event time constraint for each agent, we introduce timers, i.e., let $\beta_p \in \mathbb{R}_{\geq 0}$ be a timer variable

	$\tau_{\min}$	k_1	ν	$T_{\text{convergence}}$
Simulation 1	0.01	6.7×10^{-3}	0.82	371.71
Simulation 2	0.1	2.1×10^{-3}	2.59	740.62
Simulation 3	0.5	9.44×10^{-4}	5.79	940.24
Simulation 4	1.0	6.68×10^{-4}	8.19	1049.40
Simulation 5	1.1	6.37×10^{-4}	8.59	1409.0

Table 2-1. Convergence results from repeated simulations with an increasing $\tau_{\min}$.

for agent p that evolves according to

$$\begin{aligned} \dot{\beta}_p &= 1, \quad \mathbf{T}_p(\xi) \geq 0 \\ \beta_p^+ &= 0, \quad \mathbf{T}_p(\xi) \leq 0. \end{aligned} \tag{2–42}$$

Given a maximal solution ϕ of $\mathcal{H}$, the trajectory $\beta_p(\phi(t,j))$ can be used to determine the length of each flow interval of agent p. In particular, the height of each peak of $\beta_p(\phi(t,j))$ in Fig. 2-7 represents the length of the corresponding flow interval for agent p, which must be greater than $\tau_{\min}$ to satisfy the minimum inter-event time constraint.

From Fig. 2-1, (2–16), (2–20), and (2–33) the minimum inter-event time constraint is satisfied while creating a trade-off in performance. The speed of convergence can be sacrificed for accuracy of convergence (size of ν resulting from the system parameters) based on the chosen system parameters. This means that ν-approximate consensus can be achieved quickly for a large ν or ν-approximate consensus can be achieved slowly for a small ν, all while achieving the specified minimum inter-event time.

Multiple simulations were performed with the same system parameters, but $\tau_{\min}$ was increased from 0.01 to 1.1. The convergence results are shown in Table 2-1. The convergence time, $T_{\text{convergence}}$ was recorded to be the time when $|\phi(t,j)|_{\mathcal{A}} \leq \nu$. From Table 2-1, as $\tau_{\min}$ is increased, the convergence time increases and the convergence accuracy decreases.

2.7 Concluding Remarks

The developed solution to the ν-approximate consensus problem using a distributed event-triggered coordination scheme for a MAS with homogeneous dynamics and a

specific communication minimum inter-event time. The developed control design is PAS and the workspace is forward invariant. The forward invariance of the agents' workspace shows an aspect of safety in the presented control design. Additionally, a bounded control input and an appropriate event trigger mechanism can yield a user-specified minimum inter-event time.

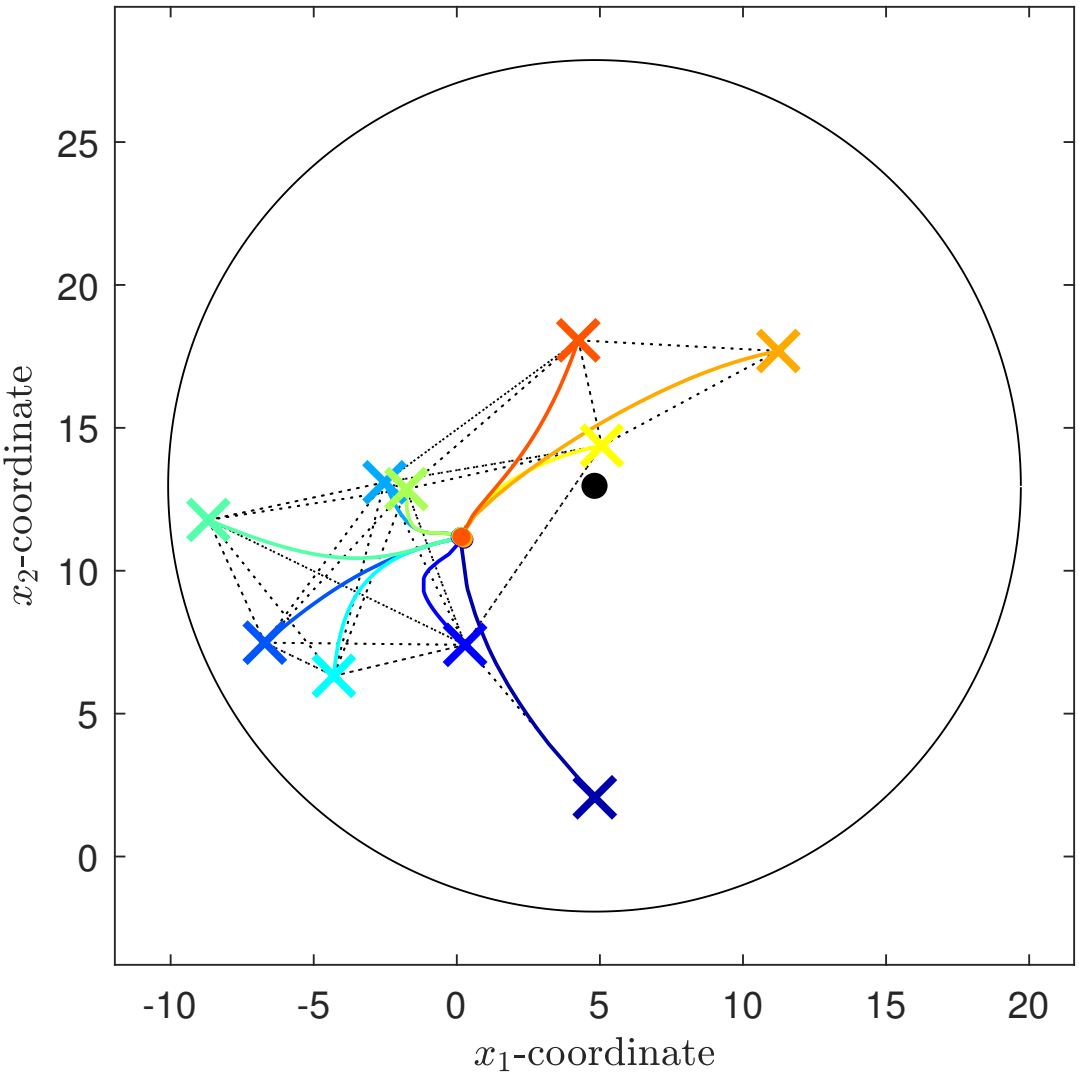

Figure 2-2. Depiction of the initial and final configuration (colored $\times$'s, $\bullet$'s), agent trajectories (colored curves), and communication graph $\mathcal{G}$ (black dashed lines). The boundary and center of the workspace Ω are represented by the black circle and disk.

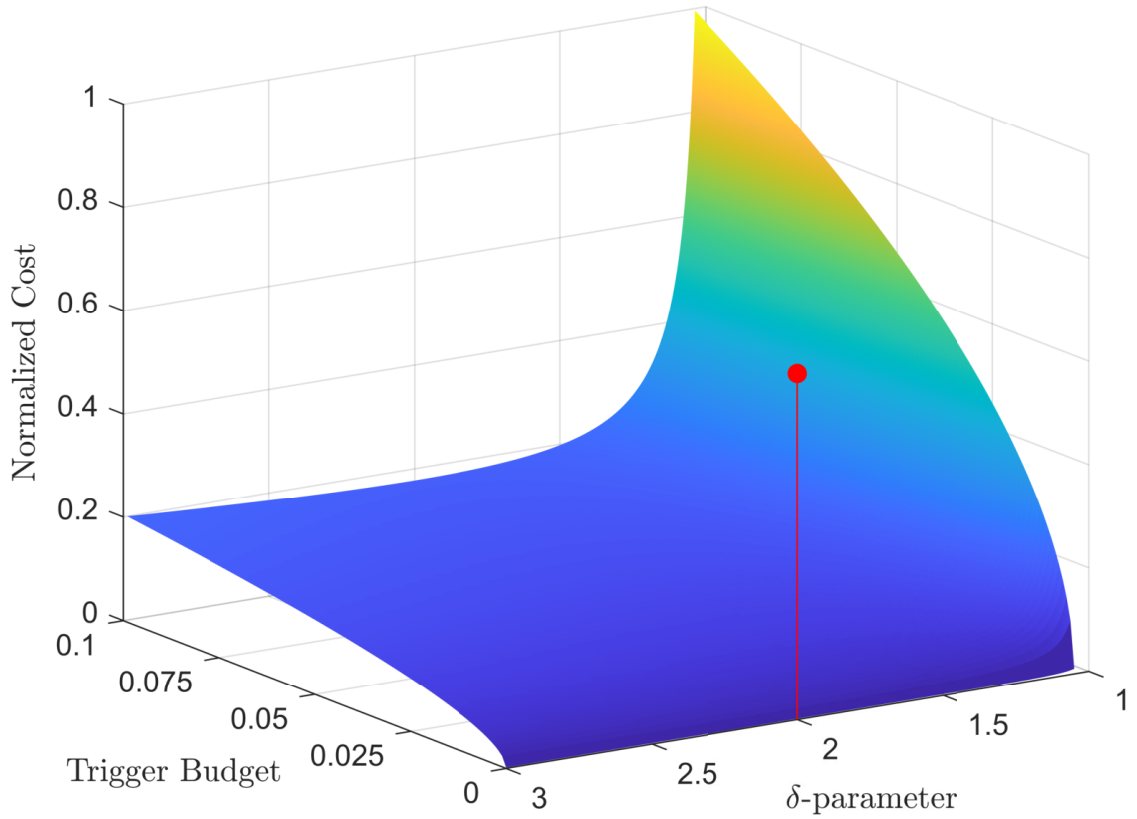

Figure 2-3. Depiction of the normalized surface $g(\mathfrak{h}, \delta)$ over $[1 \times 10^{-5}, 1 \times 10^{-1}] \times [1.05, 3]$. MATLAB's `fmincon` function was used to minimize $g(\mathfrak{h}, \delta)$ over $[1 \times 10^{-5}, +\infty) \times [1 + 1 \times 10^{-5}, +\infty)$, which successfully found the minimizer $(1 \times 10^{-5}, 2)$ as represented by the red stem. Note that "Trigger Budget" refers to the variable $\mathfrak{h}$.

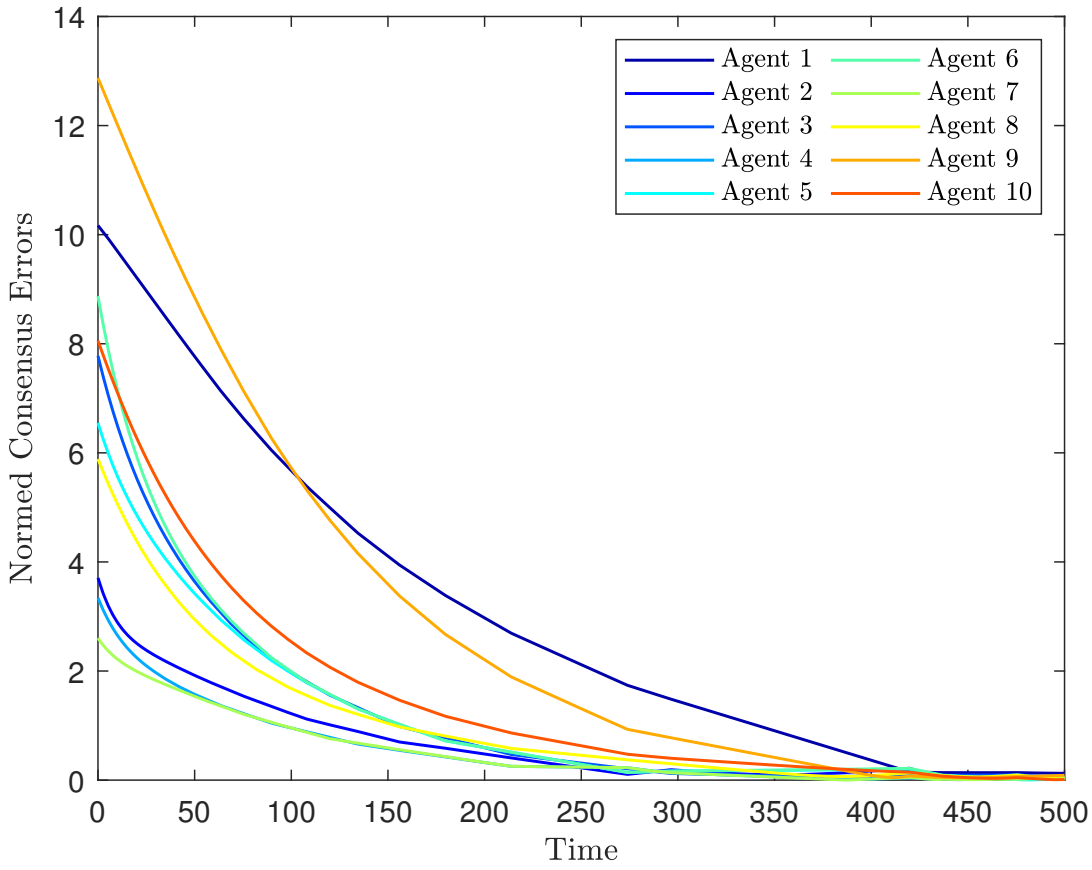

Figure 2-4. Depiction of the MAS normed consensus errors. The relatively small value of k_1 led to a relatively large simulation time.

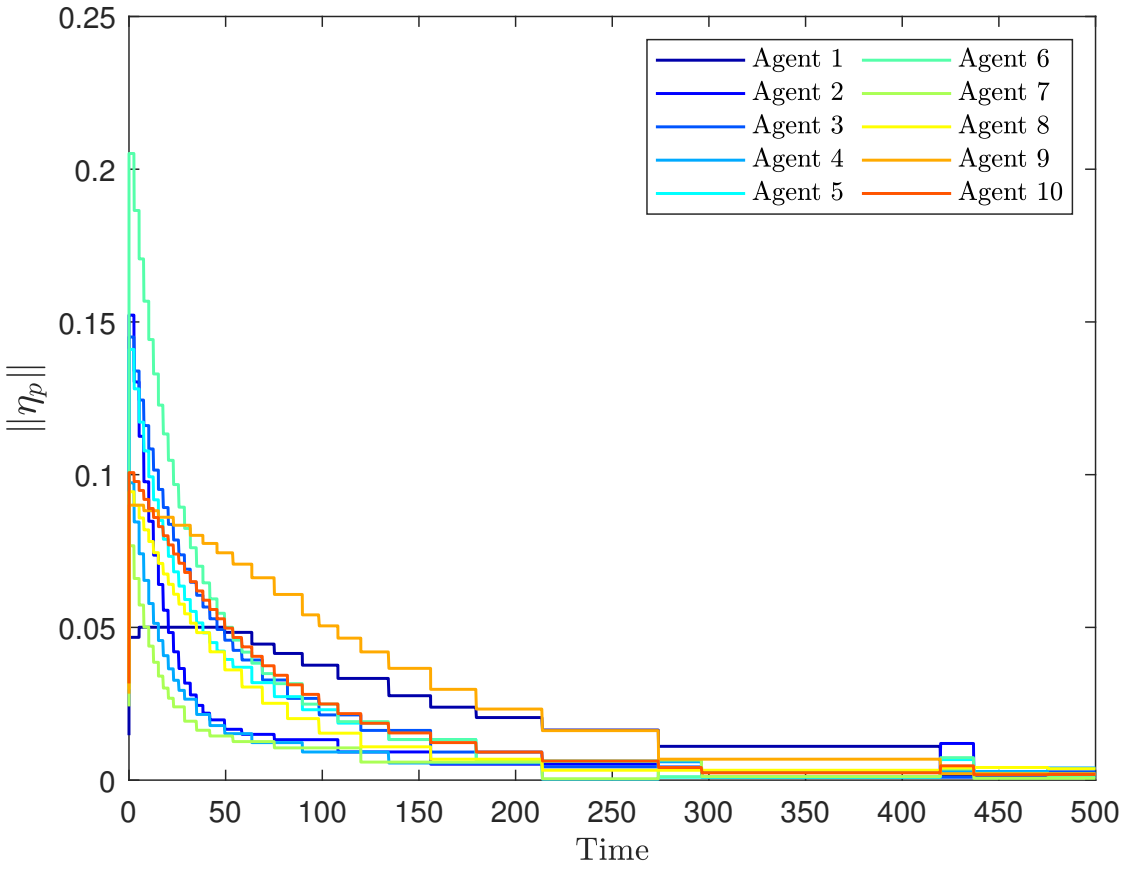

Figure 2-5. Depiction of the normed consensus variables $\{\eta_p\}_{p\in\mathcal{V}}$ versus time. The small magnitude values are due to a small value for k_1.

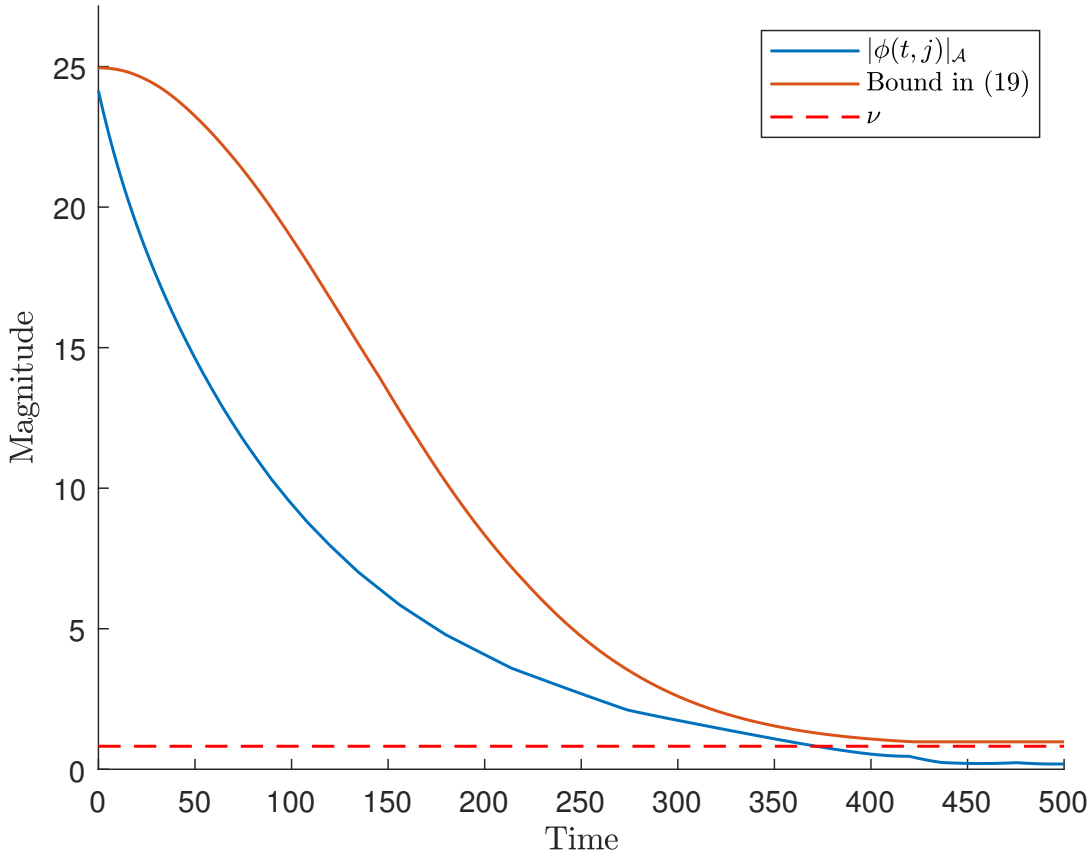

Figure 2-6. This plot depicts the distance of ϕ (solution of $\mathcal{H}$) to the desired attractor $\mathcal{A}$, the bound on $|\phi(t,j)|_{\mathcal{A}}$ presented in (2–19). This plot shows that the MAS achieves ν-approximate consensus.

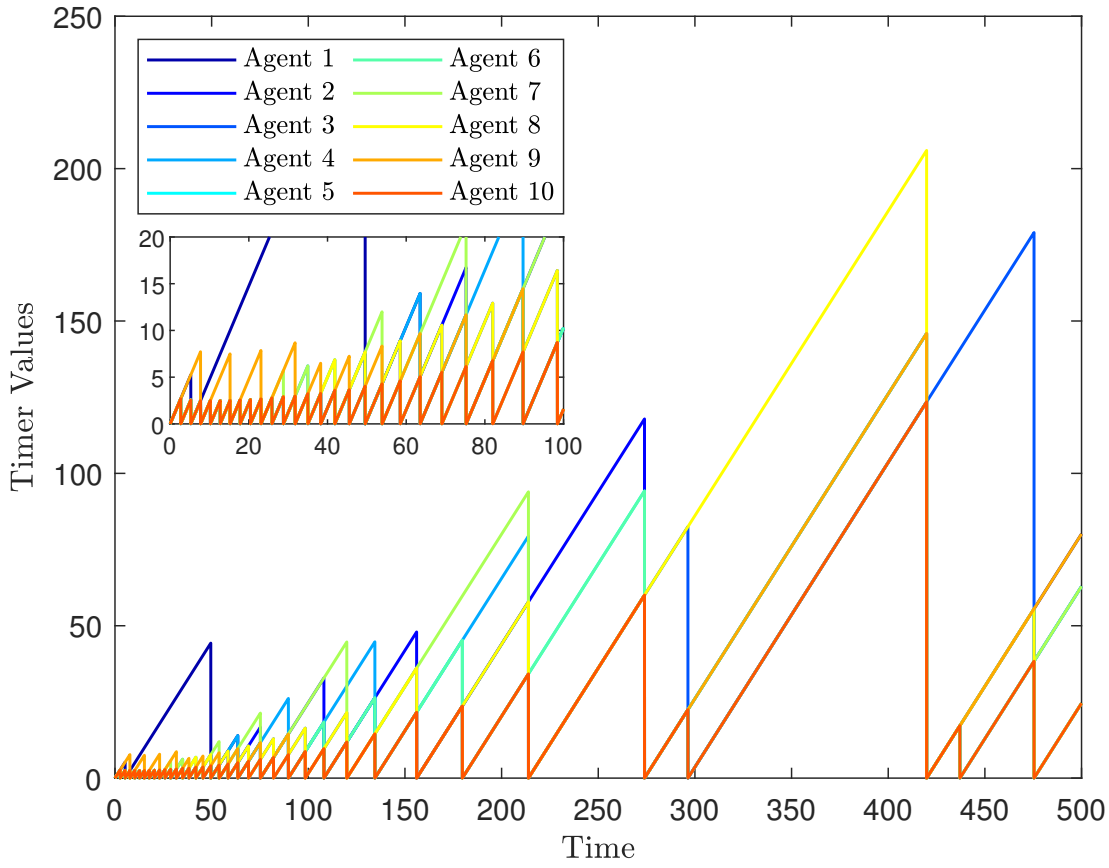

Figure 2-7. Illustration of the timer values $\{\beta_p\}_{p\in\mathcal{V}}$ versus time. The flow intervals of each agent are represented by the linear growth regions of the corresponding curve. Since no peak of any curve is smaller than $\tau_{\min} = 0.01$, every flow interval is longer than $\tau_{\min}$.

CHAPTER 3
EVENT-TRIGGERED INDIRECT CONTROL OF A COOPERATIVE AGENT

The work in this chapter is centered around developing control laws to regulate semi-cooperative or cooperative target agents. Cooperative agents are aware of the goal that the pursuer agents are regulating them towards. Semi-cooperative agents are agnostic to a goal and follow the control influence from the pursuer agent. Regardless of if a cooperative or semi-cooperative paradigm is adopted, the target agents in this work will move in a constant direction unless influenced by the pursuer. Typical indirect control results consider a continuous time influence function, typically a proximity based function, but in this work an event-triggered influence function is considered.

3.1 Problem Formulation

A single pursuer agent is tasked with regulating a single cooperative target agent to a goal location, which is unknown to the target agent. Unlike results such as [74, 77, 79], the target agent is only capable of measuring the relative direction to the pursuer agent when dictated by the pursuer agent's trigger function. Let $\eta_t(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ denote the target agent's position, and let $\eta_p(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ denote the pursuer agent's position. The dynamics of the target agent are given by

$$\dot{\eta}_t(t) \triangleq f(\eta_t(t)) + \mu_t(\eta_t(t_k), \eta_p(t_k)), \tag{3–1}$$

where the zero-order hold inter-agent influence is denoted $\mu_t(\eta_t(t_k), \eta_p(t_k)) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$, t_k denotes the k^{th} time instant when the pursuer agent influences the target agent, and $f : \mathbb{R}^n \to \mathbb{R}^n$ represents the target agent's unknown drift dynamics. The inter-agent influence function is defined by

$$\mu_t(\eta_t(t_k), \eta_p(t_k)) \triangleq \begin{cases} \nu_t\left(\frac{\eta_t(t_k)-\eta_p(t_k)}{\|\eta_t(t_k)-\eta_p(t_k)\|}\right), & \text{if } \|\eta_t(t_k) - \eta_p(t_k)\| \leq R_p \\ 0, & \text{otherwise,} \end{cases} \tag{3–2}$$

where $\nu_t \in \mathbb{R}_{>0}$ denotes the constant speed of the target agent and $R_p \in \mathbb{R}_{>0}$ denotes the maximum distance at which the pursuer agent can exert influence over the target agent. The influence function in (3–2) defines a unit vector from the pursuer to the target, scaled by the target's constant speed. This influence function denotes the effect of the pursuer's position on the target's velocity during influence events. Furthermore, the influence function in (3–2) is motivated by sensor applications in which sensing can only occur between discrete time intervals. The use of a relative direction influence is motivated by a sensor such as an underwater SONAR, which motivates the inclusion of a maximum influence range. The target agent's dynamics are not directly controllable; therefore, the pursuer agent will regulate the target agent's trajectory by positioning itself to update the zero-order hold influence function in (3–2). Unlike the target agent, the pursuer agent is directly controllable with dynamics

$$\dot{\eta}_p(t) \triangleq \mu_p(t), \tag{3–3}$$

where $\mu_p(t) : \mathbb{R}_{>0} \to \mathbb{R}^n$ denotes the control input of the pursuer agent.

Assumption 3.1. The target agent's drift dynamics, $f : \mathbb{R}^n \to \mathbb{R}^n$, are continuously differentiable and bounded as $\|f(\eta_t(t))\| \leq \overline{f}$.

Assumption 3.2. The pursuer agent can continuously measure its state, $\eta_p(t)$, and the target agent's state, $\eta_t(t)$ for all $t \in \mathbb{R}_{\geq 0}$.

Assumption 3.3. The pursuer agent can instantaneously impart influence on the target agent defined by the inter-agent influence function in (3–2).

3.2 Control Objective

The goal of the pursuer agent is to regulate the target agent's state to a static desired goal location, $\zeta_g \in \mathbb{R}^n$. To quantify the target agent regulation objective, the target agent position error $e_t(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is defined as

$$e_t(t) \triangleq \eta_t(t) - \zeta_g. \tag{3–4}$$

The pursuer agent error $e_p(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, is defined as

$$e_p(t) \triangleq \eta_d(t) - \eta_p(t), \tag{3–5}$$

where $\eta_d(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ represents the pursuer agent's desired trajectory, which is defined as

$$\eta_d(t) \triangleq \eta_t(t) + R_a S_0, \tag{3–6}$$

where $S_0 \triangleq -e_t(t) / \|e_t(t)\|$ denotes a direction vector equal to the normalized direction from the goal location to the target agent, and $R_a \in \mathbb{R}_{>0}$ represents the desired influence range, where R_a must be selected such that $R_a \leq R_p$. The minimum and maximum influence ranges are motivated by the pursuer agent's use of sensors to influence the target agent's control input. Based on the subsequent stability analysis, the pursuer agent's controller is designed as

$$\mu_p(t) \triangleq \dot{\eta}_d(t) + (\beta + 1) e_p(t), \tag{3–7}$$

where $\beta \in \mathbb{R}_{>0}$ is a user-selected constant. In the following subsection, we formulate a trigger function that determines conditions under which the pursuer agent exerts influence on the target agent's trajectory.

3.2.1 Trigger Design

The pursuer agent influences the target agent when dictated by the trigger function $T : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. The minimum and maximum IETs are denoted by $\delta_{\min}$ and $\delta_{\max}$, respectively. The trigger function is designed to incorporate a user-prescribed minimum IET to account for sensing or communication constraints. The trigger function employs a timer, $\tau \in \mathbb{R}_{>0}$, to enforce the minimum and maximum IETs while allowing elements of the system state to dictate events between the maximum and minimum IET limits. The timer is described by $\tau \triangleq t - t_k$, where t_k is the time the last influence event occurred.

The trigger function is defined as

$$T \triangleq \begin{cases} \gamma_s, & \tau < \delta_{\min}, \\ \gamma_s + \|e_p(t)\|^2 - (\alpha+1)\|e_t(t)\|^2, & \delta_{\min} \leq \tau < \delta_{\max}, \\ 0, & \text{otherwise}, \end{cases} \tag{3–8}$$

where $\gamma_s \triangleq \frac{1}{2}\overline{f}^2 + \frac{1}{2}{\nu_t}^2 + (\alpha+1)\|e_t(t_k)\|^2$ and $\alpha \in \mathbb{R}_{>0}$ is a user-defined constant. When $\tau < \delta_{\min}$, the trigger function is a positive constant equal to γ_s, preventing the pursuer agent from influencing the target agent before the minimum IET condition has been met. When the timer τ is between the minimum and maximum IETs, we have $\delta_{\min} \leq \tau < \delta_{\max}$ and the trigger function T is dependent on the system error signals $e_p(t)$ and $e_t(t)$. When the timer is equal to the maximum IET, the trigger function is set to zero. The trigger function generates an influence event when $T = 0$. This condition can occur when the timer reaches the maximum IET or if the norm of the target agent error, $\|e_t\|$, grows sufficiently large. From (3–8), since $T = \gamma_s$ when $\tau < \delta_{\min}$, the pursuer agent cannot influence the target agent when the IETs are less than the minimum IET. Therefore, Zeno behavior is prevented given the minimum IET is selected such that $\delta_{\min} \in (0, \delta_{\max}]$. When $T = 0$, the k^{th} influence time is updated as $t_k = t$ and the timer τ is reset such that $\tau = 0$. Additionally, the constant γ_s is reset to $\gamma_s = \frac{1}{2}\overline{f}^2 + \frac{1}{2}{\nu_t}^2 + (\alpha+1)\|e_t(t_k)\|^2$. The trigger function can be constructed to rely only on a timer variable; however, including the error signal component allows the trigger function to generate influence events more often if the target is not tracking the desired trajectory. That is, the error signal component of T enables influence events to occur before $\tau = \delta_{\max}$.

3.3 Stability Analysis

The stability analysis considers the system in two parts: a converging subsystem when the pursuer agent influences the target agent, and a diverging subsystem when

the pursuer agent does not influence the target agent. After each subsystem is considered independently, stability criteria are developed by considering the combined system as a switched system. For notational brevity, the time dependence of functions and variables is omitted in the following analysis, unless additional clarity is required. In the following subsections, t_k^s and t_k^u represent the k^{th} time instant the combined system enters its stable and unstable subsystem, respectively. Specifically, the sequence t_k^s, t_k^u, t_{k+1}^s represents an iteration of entering the stable subsystem for the k^{th} time, followed by the unstable subsystem for the k^{th} time, followed by stable subsystem for the $(k+1)^{\text{th}}$ time. From Assumption 3.3, $t_k^u - t_k^s = 0$. Therefore, in (3–8), $t_k = t_k^u$ and the trigger function evolves while in the unstable subsystem. Let $\xi = \left[e_p^\top, e_t^\top\right]^\top : \mathbb{R}_{\geq 0} \to \mathbb{R}^{2n}$ represent the concatenated vector of agent regulation errors. Consider the common candidate Lyapunov function, $V : \mathbb{R}^{2n} \to \mathbb{R}$,

$$V(\xi) = \frac{1}{2} e_p^\top e_p + \frac{1}{2} e_t^\top e_t. \tag{3–9}$$

3.3.1 State Convergence

The following stability analysis reveals that all system trajectories converge to the set $\mathcal{B} \triangleq \{\xi \in \mathbb{R}^{2n} : \|\xi(t)\|^2 \leq \frac{\gamma_s}{\lambda_s}\}$, where $\lambda_s \triangleq \min\{\beta, \alpha\}$ determines the rate of convergence, establishing globally uniformly ultimately bounded (UUB) regulation.

Theorem 3.1. *For the dynamical systems in (3–1) and (3–3) and any initial state $\xi(t_0)$, the control law given in (3–7) ensures $\xi(t)$ converges exponentially to the set $\mathcal{B}$, where*

$$\|\xi(t)\|^2 \leq \left(\|\xi(t_k^s)\|^2 - \frac{\gamma_s}{\lambda_s}\right) e^{-2\lambda_s\left(t - t_k^s\right)} + \frac{\gamma_s}{\lambda_s} \tag{3–10}$$

for all $t \in [t_k^s, t_k^u)$, given that $\alpha > 0$, $\beta > 0$, and Assumptions 3.1-3.3 are satisfied.

Proof. Taking the total derivative of (3–9) along the trajectories of ξ, and substituting the dynamics of the target agent from (3–1) and the time derivative of (3–5) yields

$$\dot{V}(\xi) = e_p^\top\left(\dot{\eta}_d - \dot{\eta}_p\right) - e_t^\top\left(f(\eta_t) + \mu_t\right). \tag{3–11}$$

Substituting (3–3) and (3–7) into (3–11) yields

$$\dot{V}(\xi) = -\beta e_p^\top e_p - e_t^\top \left(f(\eta_t) + \mu_t\right). \tag{3–12}$$

Using the Cauchy-Schwarz inequality and Young's inequality, (3–12) can be written as

$$\dot{V}(\xi) \leq -\beta \|e_p\|^2 + \|e_t\|^2 + \frac{1}{2}\|f(\eta_t)\|^2 + \frac{1}{2}\|\mu_t\|^2. \tag{3–13}$$

Given $\|f(\eta_t(t))\| \leq \overline{f}$ and $\|\mu_t\| \leq \nu_t$, we can upper bound (3–13) as

$$\dot{V}(\xi) \leq -\beta \|e_p\|^2 + \|e_t\|^2 + \frac{1}{2}\overline{f}^2 + \frac{1}{2}\nu_t^2. \tag{3–14}$$

Since $\gamma_s = \frac{1}{2}\overline{f}^2 + \frac{1}{2}{\nu_t}^2 + (\alpha+1)\|e_t\|^2$ while $t \in [t_k^s, t_k^u)$, then (3–14) can be expressed as

$$\dot{V}(\xi) \leq -\beta \|e_p\|^2 + \|e_t\|^2 + \gamma_s - (\alpha+1)\|e_t\|^2. \tag{3–15}$$

Therefore, (3–15) becomes

$$\dot{V}(\xi) \leq -\beta \|e_p\|^2 - \alpha \|e_t\|^2 + \gamma_s. \tag{3–16}$$

Using the fact that $\lambda_s = \min\{\beta, \alpha\}$ and substituting (3–9) into (3–16) yields

$$\dot{V}(\xi) \leq -2\lambda_s V(\xi) + \gamma_s. \tag{3–17}$$

Solving the differential inequality in (3–17) yields

$$V(\xi(t)) \leq V(\xi(t_k^s))\, e^{-2\lambda_s\left(t - t_k^s\right)} + \frac{\gamma_s}{2\lambda_s}\left(1 - e^{-2\lambda_s\left(t-t_k^s\right)}\right). \tag{3–18}$$

Using [115, Def. 4.6] shows that ξ is uniformly ultimately bounded by

$$\|\xi(t)\| \leq \sqrt{\|\xi(t_k^s)\|^2 e^{-2\lambda_s\left(t-t_k^s\right)} + \frac{\gamma_s}{\lambda_s}\left(1 - e^{-2\lambda_s\left(t-t_k^s\right)}\right)},$$

for all $t \in [t_k^s, t_k^u)$, which yields the result in (3–10). ■

3.3.2 State Divergence

Theorem 3.2. *For the dynamical systems in (3–1) and (3–3) and* $t \in \left[t_k^u, t_{k+1}^s\right)$*, the state* $\xi(t)$ *remains bounded as*

$$\|\xi(t)\|^2 \leq \left(\|\xi(t_k^u)\|^2 + \gamma_u\right) e^{2\left(t-t_k^u\right)} - \gamma_u, \tag{3–19}$$

where $\gamma_u \in \mathbb{R}_{>0}$ *is defined as* $\gamma_u \triangleq \frac{1}{2}\overline{f}^2 + \frac{1}{2}{\nu_t}^2 \in \mathbb{R}_{>0}$.

Proof. Taking the total derivative of (3–9) along trajectories of ξ, substituting the dynamics of the target agent from (3–1), and substituting the time derivative of (3–5) yields

$$\dot{V} = e_p^\top \left(\dot{\eta}_d - \dot{\eta}_p\right) + e_t^\top \left(f\left(\eta_t\right) + \mu_t\right). \tag{3–20}$$

Substituting (3–3) and (3–7) into (3–20) yields

$$\dot{V} = e_p^\top \left(\dot{\eta}_d - \left(\dot{\eta}_d + \beta e_p\right)\right) + e_t^\top \left(f\left(\eta_t\right) + \mu_t\right). \tag{3–21}$$

Using the Cauchy-Schwarz inequality, Young's inequality, and Assumption 3.1, (3–21) is upper-bounded as

$$\dot{V} \leq -\beta \|e_p\|^2 + \|e_t\|^2 + \frac{1}{2}\overline{f}^2 + \frac{1}{2}{\nu_t}^2. \tag{3–22}$$

Given $-\beta \|e_p\|^2 \leq \|e_p\|^2$ and $\gamma_u = \frac{1}{2}\overline{f}^2 + \frac{1}{2}{\nu_t}^2$, we can further bound (3–22) as

$$\dot{V} \leq 2V + \gamma_u. \tag{3–23}$$

Solving the differential inequality in (3–23) yields

$$V(t) \leq V\left(t_k^u\right) e^{2\left(t-t_k^u\right)} + \frac{1}{2}\gamma_u \left(e^{2\left(t-t_k^u\right)} - 1\right), \tag{3–24}$$

for all $t \in \left[t_k^u, t_{k+1}^s\right)$. Substituting (3–9) into (3–24) and simplifying the expression yields the result in (3–19). ■

3.3.3 Combined Stability Analysis

Theorem 3.3. *The combination of the trigger function in (3–8) and the control law in (3–7) guarantees that the state ξ remains bounded by*

$$\|\xi\left(t\right)\|^2 \leq \left(\|\xi\left(t_k^s\right)\|^2 - \frac{\gamma_s}{\lambda_s}\right) e^{-2\lambda_s\left(t-t_k^s\right)} + \frac{\gamma_s}{\lambda_s}$$

for all $t \in \mathbb{R}_{\geq 0}$ given that $t_{k+1}^s - t_k^u < \delta_k$ and

$$\delta_k \leq \frac{1}{2}\ln\left[\frac{\xi_s + \gamma_u}{\xi_u}\right], \tag{3–25}$$

where $\xi_s \triangleq \|\xi\left(t_k^s\right)\|^2$, and $\xi_u \triangleq \|\xi\left(t_k^u\right)\|^2$.

Proof. The stable subsystem is upper-bounded by (3–10), which can be written as

$$\|\xi\left(t\right)\| \leq \sqrt{\left(\xi_s - \frac{\gamma_s}{\lambda_s}\right) e^{-2\lambda_s\left(t-t_k^s\right)} + \frac{\gamma_s}{\lambda_s}}. \tag{3–26}$$

The unstable subsystem is bounded by (3–19), which can also be written as

$$\|\xi\left(t\right)\| \leq \sqrt{\left(\xi_u + \gamma_u\right) e^{2\left(t-t_k^u\right)} - \gamma_u}. \tag{3–27}$$

Since the influence updates occur instantaneously, $t_{k+1}^u - t_k^s = 0$, and the system predominantly remains in the unstable subsystem. The time interval $\delta_k = t_{k+1}^s - t_k^u$ is the time interval from when the system enters the unstable subsystem until it exits the unstable subsystem. Consider the unstable subsystem in δ_k and the stable subsystem in the preceding interval $\delta_{k-1} = t_k^u - t_k^s$. Given that the influence the pursuer imparts on the target agent occurs instantaneously, then $\delta_{k-1} = 0$. Let the unstable subsystem be bounded by the stable subsystem, then (3–26) and (3–27) can be expressed as

$$\xi_u e^{2\delta_k} \leq \left(\xi_s - \frac{\gamma_s}{\lambda_s}\right) e^{-2\lambda_s\delta_{k-1}} + \frac{\gamma_s + \lambda_s\gamma_u\left(1 - e^{2\delta_k}\right)}{\lambda_s}. \tag{3–28}$$

Since $\delta_{k-1} = 0$, (3–28) can be bounded by

$$\xi_u e^{2\delta_k} \leq \xi_s + \gamma_u. \tag{3–29}$$

Solving for δ_k in (3–29) yields the result in (3–25). ■

Remark 3.1. During the interval $t \in [t_k^s, t_k^u)$, the radius of the set $\mathcal{B}$ decreases because the target agent converges toward the goal location. Conversely, the radius of the set $\mathcal{B}$ expands when the target agent diverges from the goal location during this same interval. The evolution of the radius of $\mathcal{B}$ occurs because $\gamma_s = \frac{1}{2}\overline{f}^2 + \frac{1}{2}{\nu_t}^2 + (\alpha+1)\left\|e_t\left(t_k^s\right)\right\|^2$ is reset after each trigger event. Specifically, if $\left\|e_t\left(t_k^s\right)\right\| > \left\|e_t\left(t_{k-1}^s\right)\right\|$ then the radius of the set $\mathcal{B}$ increases, and if $\left\|e_t\left(t_k^s\right)\right\| < \left\|e_t\left(t_{k-1}^s\right)\right\|$ then the radius of the set $\mathcal{B}$ decreases. The trigger function given by (3–8) will generate influence events more frequently if the radius of $\mathcal{B}$ is growing due to target agent divergence from the goal location. When the target agent is diverging from the goal location, $\gamma_s + \left\|e_p\left(t\right)\right\|^2 - (\alpha+1)\left\|e_t\left(t\right)\right\|^2 = 0$ before $t - t_k = \delta_{\max}$, and the IETs will approach the minimum IET. Subsections 3.3.1 and 3.3.3 imply that with each successive sequence $\delta_k \to \delta_{k+1}$, the radius of $\mathcal{B}$ decreases as the pursuer agent regulates the target agent to the goal. Furthermore, from the stability result, the radius of the set $\mathcal{B}$ decreases with each successive sequence $\delta_k \to \delta_{k+1}$, and $\limsup_{t\to\infty} \left\|\xi\left(t\right)\right\|^2 \leq \frac{1}{2\lambda_s}\left(\overline{f}^2 + {\nu_t}^2\right)$ for all $t \in [0, \infty)$.

3.4 Results

3.4.1 Simulated Results

Numerical experiments demonstrate the efficacy of the controller in (3–7) and the trigger function in (3–8). The minimum IET $\delta_{\min}$ is determined by the sensing hardware constraints of the pursuer and target agents (e.g., the sampling time or processing time of the sensor). In this simulation study, the target agent drift dynamics are given by $f\left(\eta_t\left(t\right)\right) = \tanh\left(\eta_t\left(t\right)\right)$, satisfying Assumption 3.1. For this simulation, we select $\delta_{\min} = 0.1 \cdot \delta_{\max}$, $n = 2$, $R_p = 10.0$, $R_a = 5.0$, $\nu_t = 1.5$, $\alpha = 2.1$, and $\beta = 5.0$. Based on the parameter selection, γ_u was found to be $\gamma_u = 1.9639$. Given the initial configuration represented by the ○'s in Figure 3-1 and the value of γ_u, the maximum IET was found to be $\delta_{\max} = 0.178$ using 3–25. In Figure 3-3, T increases during the interval $t - t_k < \delta_{\max}$ because the target agent's tracking error e_t diminishes as it approaches the goal

location. If instead the tracking error were to increase during this period, T would rapidly decrease to zero closer to the minimum IET, $\delta_{\min}$. This relationship is critical for system efficiency—when T is driven to zero shortly after $\delta_{\min}$, the pursuer must intervene more frequently, indicating that the influence function is not effectively managing the target agent's tracking performance given the target agent's drift dynamics. The conservative nature of the result is evidence in Figure 3-2, where it can be seen that the target agent's error does not grow between influence events. Based on the fact that the target agent's error does not grow in between influence events, it is possible for the maximum dwell time to be increased while still ensuring the regulation of the target agent to the goal location.

3.4.2 Experiment

Experiments were performed at the University of Florida's Autonomy Park outdoor facility to validate the efficacy of the controller in (3–7) and the trigger function in (3–8). The results of a single experimental run are presented in Figures 3-5 and 3-6. The experiment was carried out using the Freefly Astro and Unitree Go1 platforms shown in Figure 3-4. In the experimental results, a pursuer agent (Freefly Astro quadcopter) was tasked with regulating a target agent (Unitree Go1 quadruped) to the desired coordinates $(-2.0\,\text{m}, -5\,\text{m}, 0\,\text{m})$ in the local Autonomy Park reference frame. The minimum IET, $\delta_{\min}$, was selected as 0.01s. Based upon the documented tolerances of onboard sensors and the maximum velocity of the target agent, $\overline{f}$ was conservatively estimated to be 6.0m/s. The target agent velocity, ν_t, was selected as 0.25m/s. From 3–25, and based on the initial values of ξ_u and ξ_s, the maximum IET, $\delta_{\max}$, was computed to be 0.0282s. The smallest and largest IETs from the experimental data were found to be 0.0154s and 0.0248s, respectively. Due to the small IETs, the influence and event markers shown in the simulation Figures 3-1-3-3 were omitted from Figures 3-5 and 3-6.

The experimental results demonstrate the effectiveness of the controller in (3–7) and the trigger function in (3–8), as the pursuer agent successfully regulated the target

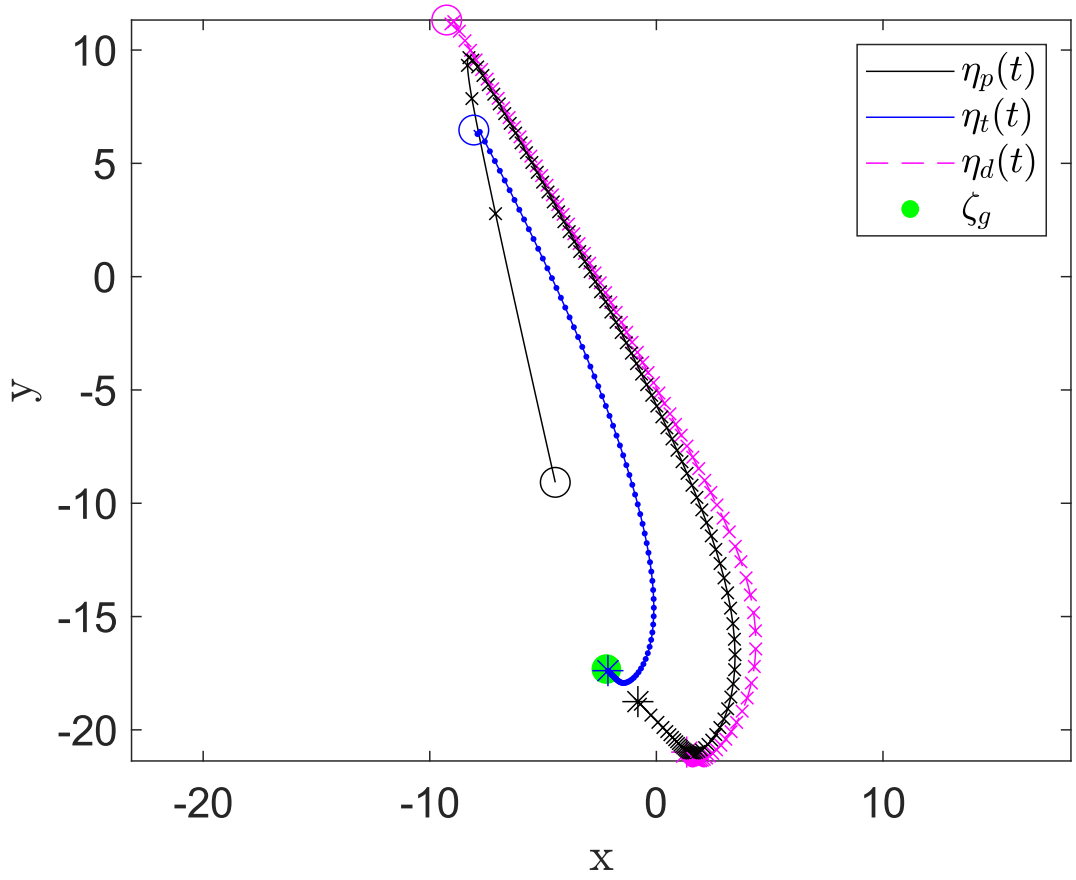

Figure 3-1. The trajectories of the pursuer agent, $\eta_p(t)$, and target agent, $\eta_t(t)$, are represented by the black and blue lines respectively. The desired trajectory, $\eta_d(t)$, is represented by the magenta dashed line. The black and magenta $\times$'s represent $\eta_p(t_k)$ and $\eta_d(t_k)$ when $T = 0$. The blue solid dots represent $\eta_t(t_k)$ when $T = 0$. The black, magenta, and blue $\bigcirc$'s represent $\eta_p(t_0)$, $\eta_d(t_0)$, and $\eta_t(t_0)$. The $*$'s represent $\eta_p(t_{\text{final}})$, $\eta_d(t_{\text{final}})$, and $\eta_t(t_{\text{final}})$. The green square represents the goal location.

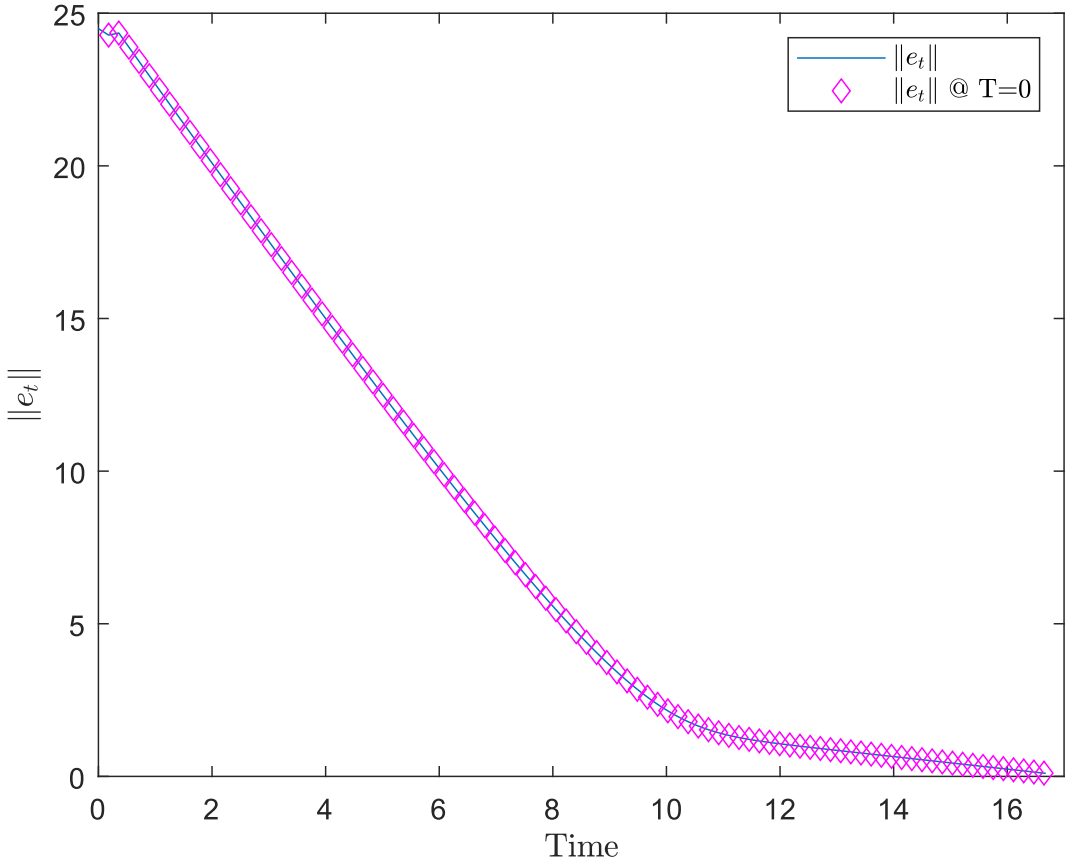

Figure 3-2. The value of $\|e_t\|^2$ over time. The magenta $\Diamond$'s represent when the pursuer agent influences the target agent.

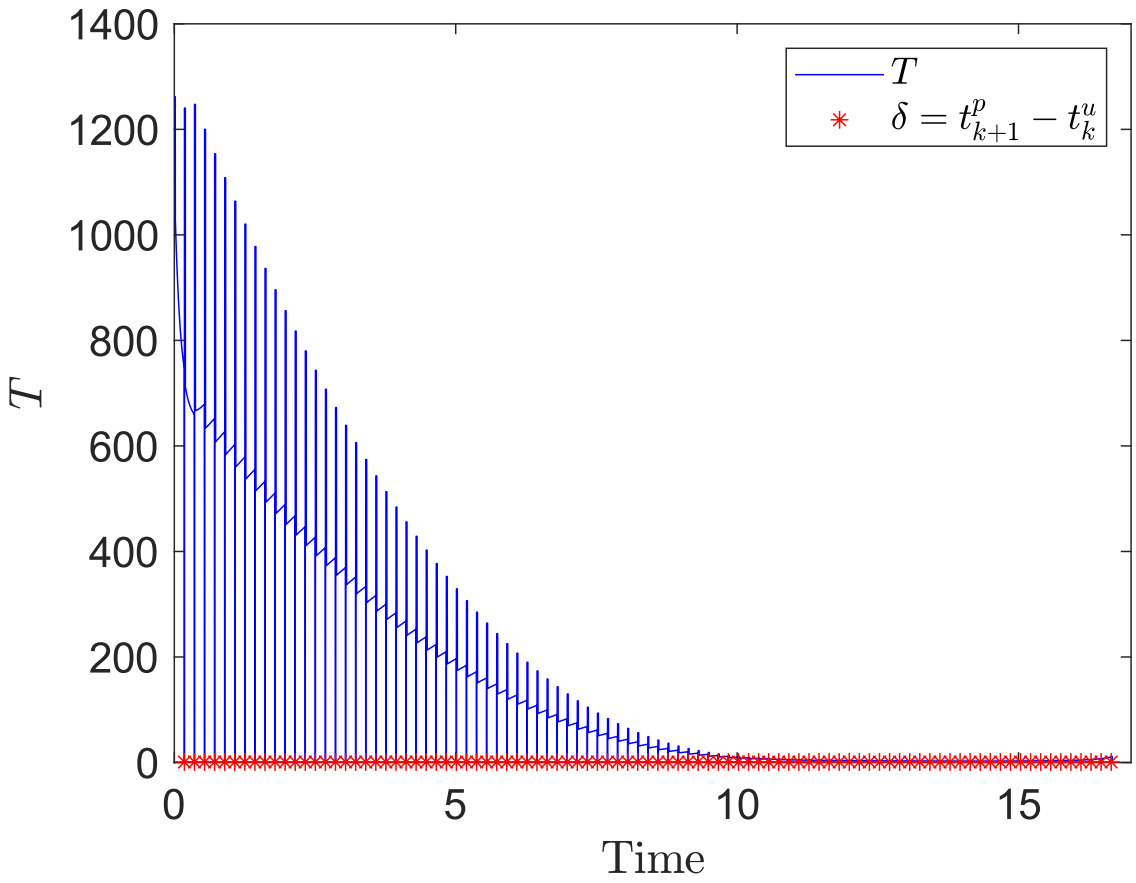

Figure 3-3. The evolution of the trigger function, denoted by the blue line over time. Event times when the pursuer agent exerts influence on the target agent are marked with red $*$'s. It can be seen that T resets to a constant and remains constant while $t^p_{k+1} - t^u_k \leq \delta_{\min}$. When $\delta_{\min} \leq t^p_{k+1} - t^u_k < \delta_{\max}$, T grows because $\|e_t\|^2$ gets smaller as the target agent is influenced in the direction of ζ_g. Finally, T is forced to zero when $t^p_{k+1} - t^u_k = \delta_{\max}$, which is evidenced by the downward spike at most red $*$'s.

Figure 3-4. The Freefly Astro quadcopter (left) was used as the pursuer agent and the Unitree Go1 quadruped (right) was used as the target agent. The quadruped uses an Emlid RS+ RTK GPS for precise position updates, which is fused with attitude and heading data from a Microstrain 3DM-GX5-AHRS using the ROS2 `robot − localization` package.

agent while maintaining IETs within the theoretically predicted bounds. In contrast to the simulated results in Section 3.4.1, experimental results revealed that trigger events were primarily initiated by state conditions rather than timer constraints, demonstrating the system's efficient adaptation to the target agent's dynamic movements.

3.5 Concluding Remarks

This work introduces a novel solution to the indirect control problem by leveraging an event-triggered relative bearing influence function to regulate a target agent toward a desired location. A rigorous connection is established between the maximum dwell

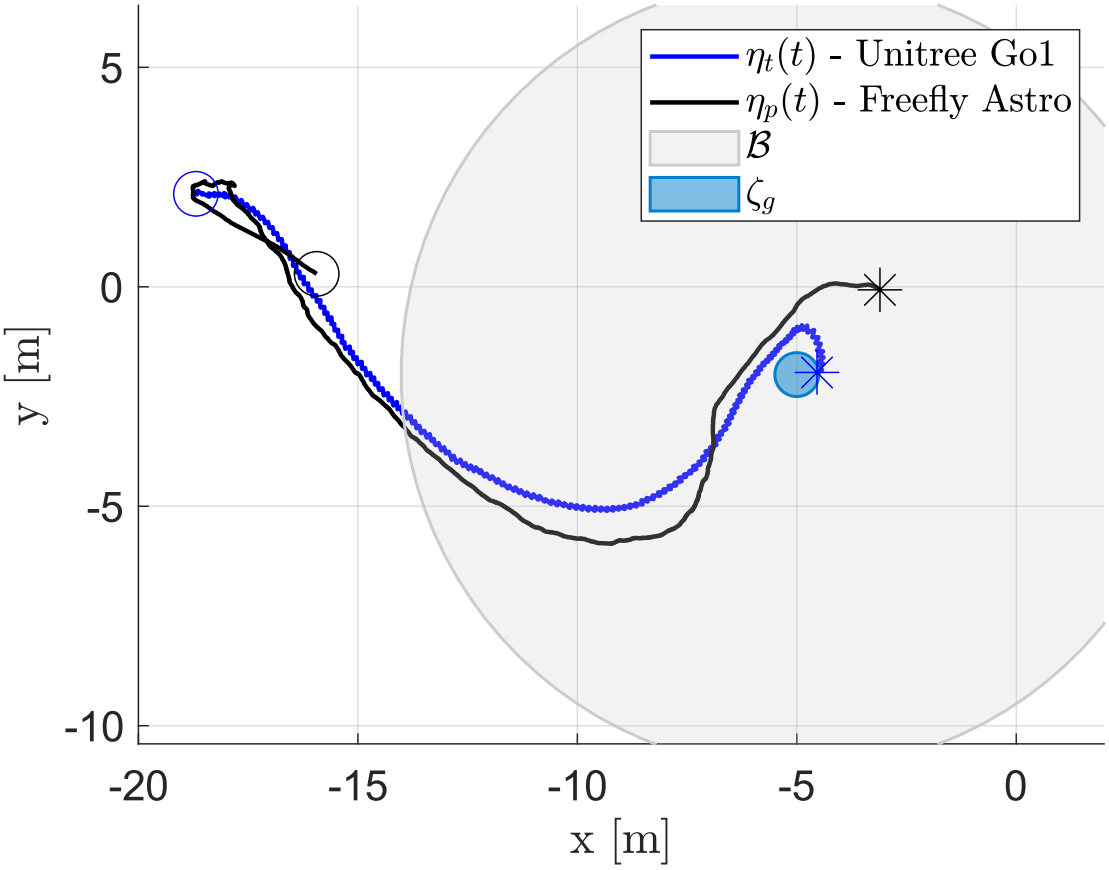

Figure 3-5. The trajectories of the pursuer agent, $\eta_p(t)$, and target agent, $\eta_t(t)$, are represented by the black and blue lines respectively. The $\bigcirc$'s represent $\eta_p(t_0)$ and $\eta_t(t_0)$. The $*$'s represent $\eta_p(t_{\text{final}})$ and $\eta_t(t_{\text{final}})$. The light blue circle represents the goal location and the ultimate bound of $\mathcal{B}$ described in Remark 3.1 is depicted by the light gray circle.

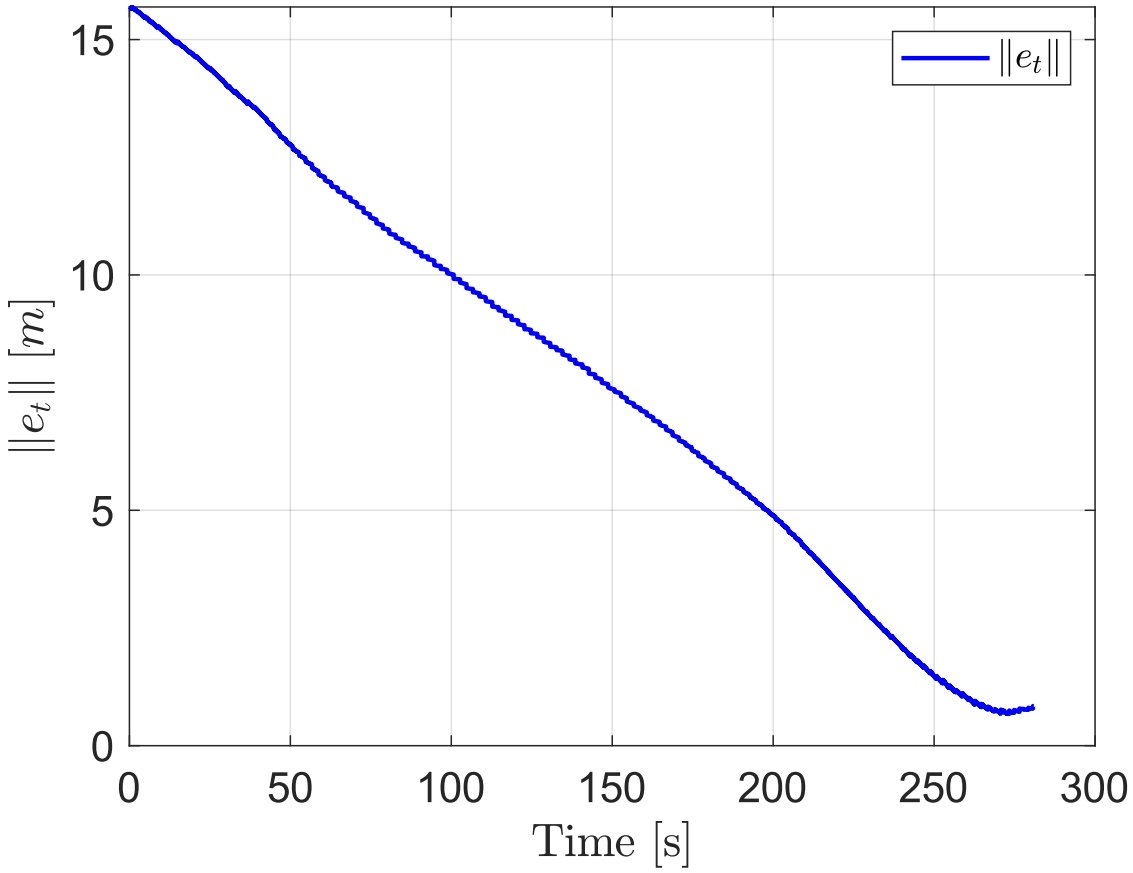

Figure 3-6. The value of $\|e_t\|^2$ during the experiment. Notable changes in the slope correspond to effective regulation of the target agent's trajectory by the pursuer agent's movement patterns.

time and IET bounds in an event-triggered indirect control framework. By formulating the system as a switched dynamical model with stable and unstable subsystems, this work develops a comprehensive dwell-time analysis to derive an upper bound on inter-event times for the trigger function. A piecewise continuous trigger function is proposed, incorporating both a maximum IET, determined through a dwell-time analysis, and a selectable minimum IET, providing enhanced control flexibility.

CHAPTER 4
INDIRECT CONTROL OF A COOPERATIVE AGENT THROUGH FIELD OF VIEW INTERACTIONS

The work in this chapter is centered around developing control laws to regulate a cooperative target agent along a desired trajectory. This work is motivated by the use of sensors that have a limited FOV, these sensors could include cameras or underwater SONAR. This chapter assumes that a target agent has a sensor that can detect the pursuer agent and can measure the resulting relative bearing to the pursuer agent. The target agent will move in a constant direction unless influenced by the relative bearing from the pursuer agent through an event-triggered influence function.

4.1 Problem Formulation

A single pursuer agent is tasked with regulating a single target agent to a desired trajectory which in unknown to the target agent. The target agent is not directly controllable but its dynamics are influenced by an inter-agent interaction. The target agent is only able to sense the pursuer agent when it is within the target agent's FOV, which is denoted $\mathcal{F}$. Many common FOVs can be modeled using a pair of circular arcs symmetric about the agent's orientation. In this work, the circular arcs are centered at $\pm\theta_t$ radians from the agent's orientation and the central angle of each circular arc is $\pm\psi_t$ radians, Figure (4-1) depicts the FOV model. The target agent is modeled using unicycle kinematics where $\eta_t(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ and $\phi_t(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}$ denote the target agent's position and orientation, respectively. Specifically, the target agent's unicycle kinematics are given by $f(\phi_t(t)) \triangleq \nu_t \cdot [\cos(\phi_t), \sin(\phi_t)]^\top$, where $\nu_t \in \mathbb{R}_{>0}$ is the known constant nominal speed of the target agent. The dynamics of the target agent are given by

$$\dot{\eta}_t(t) \triangleq f(\phi_t(t))(1 - \mathbb{1}_{\mathcal{F}}(\eta_p(t))) + k_t(\eta_t(t) - \eta_p(t))\mathbb{1}_{\mathcal{F}}(\eta_p(t)), \tag{4–1}$$

$$\dot{\phi}_t(t) \triangleq \mu_t(\eta_t(t), \eta_p(t)) \tag{4–2}$$

where $f : \mathbb{R} \to \mathbb{R}^n$ represents the unknown persistent vehicle dynamics, the indicator function $\mathbb{1}_{\mathcal{F}} : \mathbb{R}^n \to \{0, 1\}$ determines if the pursuer agent is within the target agent's

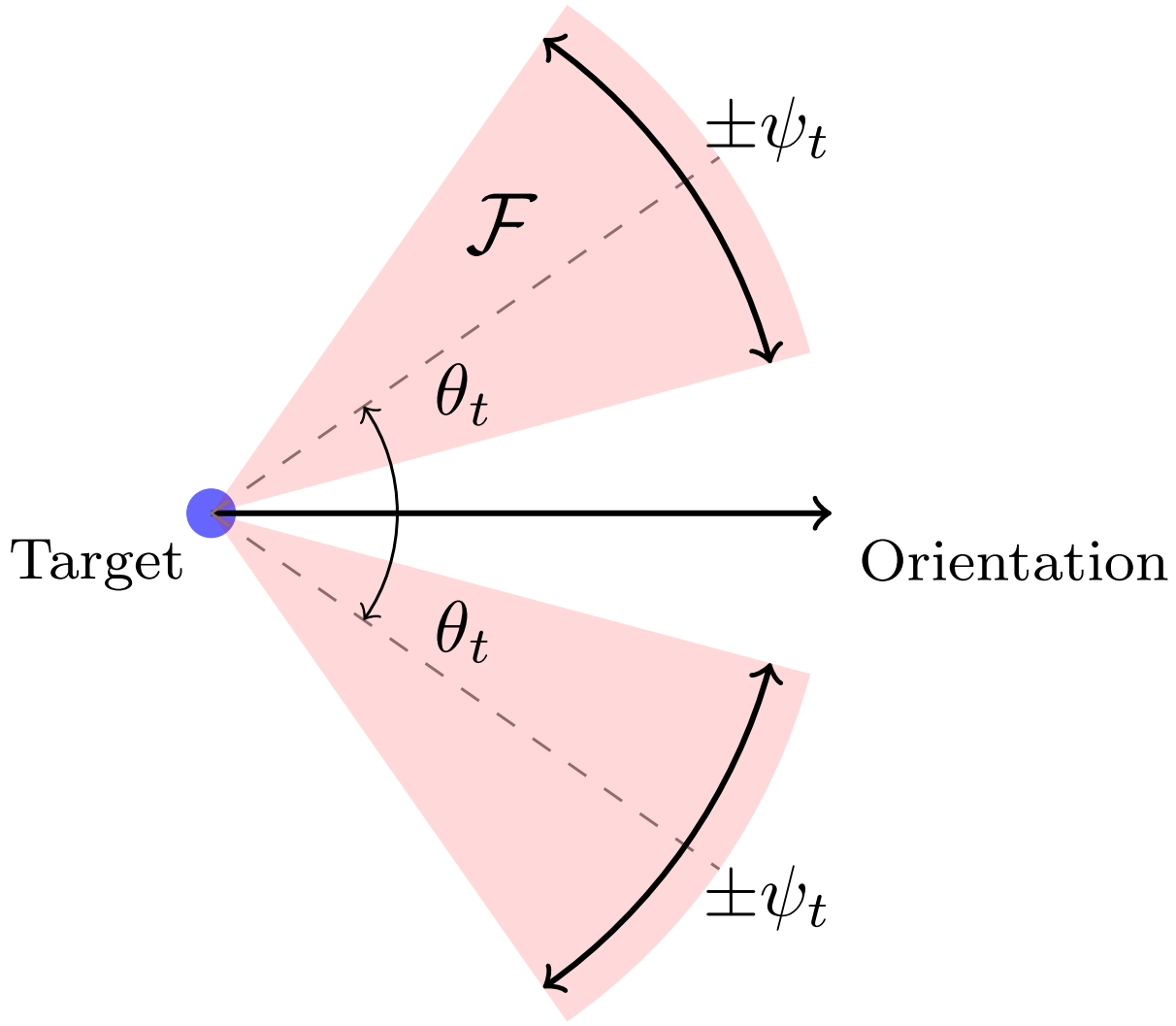

Figure 4-1. Target agent FOV model.

FOV, and $\mu_t : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ models the angular inter-agent influence between the pursuer and the target agent. The relative displacement inter-agent influence interaction is included in the dynamics of the target agent, whereas the angular inter-agent interaction function is defined by

$$\mu_t\left(\eta_t\left(t\right), \eta_p\left(t\right)\right) \triangleq \alpha \cdot \mathbb{1}_{\mathcal{F}}\left(\eta_p\left(t\right)\right), \tag{4–3}$$

where $\alpha \triangleq \left(\text{atan2}\left(\eta_t\left(t\right) - \eta_p\left(t\right)\right) - \phi_t\left(t\right)\right)$ is the difference of the relative bearing between the two agents and the target agent's orientation. The inter-agent interaction moves and rotates the target agent in a direction opposite to the direction of the relative

direction between the target agent and pursuer agent. The pursuer agent is modeled as a single-integrator system where $\eta_p(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ denotes the pursuer agent's position. Unlike the target agent, the pursuer agent is directly controllable with dynamics

$$\dot{\eta}_p(t) \triangleq \mu_p(t), \tag{4–4}$$

where $\mu_p(t) \in \mathbb{R}^n$ denotes the control input of the pursuer agent.

Assumption 4.1. The pursuer agent can continuously measure its state and the target agent's state.

Assumption 4.2. The pursuer agent can instantaneously impart influence on the target agent.

4.2 Control Objective

The goal of the pursuer agent is to regulate the target agent's state to a desired target agent trajectory, $\zeta_t(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$. To quantify the target agent regulation objective, the target agent position error $e_t(t) \in \mathbb{R}^n$ is defined as

$$e_t(t) \triangleq \eta_t(t) - \zeta_t(t). \tag{4–5}$$

The velocity of the desired target agent trajectory, $\dot{\zeta}_t : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, is bounded as $\left\| \dot{\zeta}_t(t) \right\| \leq \nu_\zeta$. The desired pursuer position $\eta_d(t)$ is designed as

$$\eta_d(t) \triangleq \zeta_t(t) + k_d e_t(t), \tag{4–6}$$

where $k_d \in \mathbb{R}_{>0}$ is a user-selected gain. The pursuer agent position error $e_p(t) \in \mathbb{R}^n$ is defined as

$$e_p(t) \triangleq \text{proj}\left(\eta_d(t), \eta_t(t)\right) - \eta_p(t), \tag{4–7}$$

where $\text{proj}\left(\eta_d(t), \eta_t(t)\right)$ projects the desired pursuer position into $\mathcal{F}$, and is defined as

$$\text{proj}\left(\eta_d(t), \eta_t(t)\right) \triangleq \begin{cases} \eta_d(t), & \text{if } \eta_d(t) \in \mathcal{F} \\ \eta_t(t) + k_r R_{\text{FOV}} \left[\cos(\phi'), \sin(\phi')\right]^\top & \text{otherwise.} \end{cases} \tag{4–8}$$

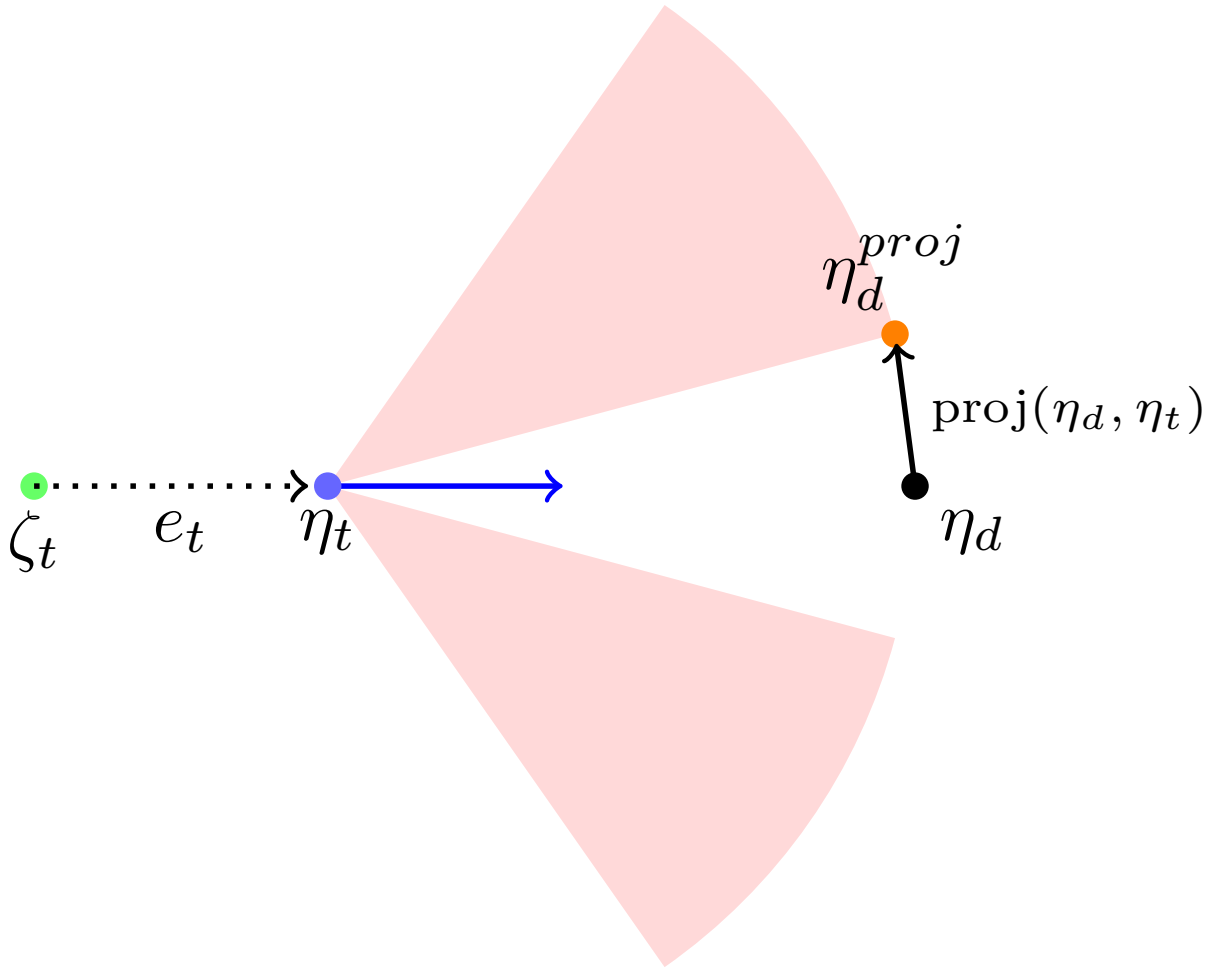

Figure 4-2. Visualization of (4–8), where $k_a = 1.0$ and $k_r = 1.0$, therefore η_d is projected onto the boundary of the FOV.

In (4–8), the angle $\phi' \in (-\pi, \pi]$ is defined by $\phi' \triangleq \phi_t + \text{sgn}(\alpha) \cdot \theta_t + \text{sgn}(\alpha + \text{sgn}(\alpha) \cdot (k_a \psi_t - \theta_t)) \cdot k_a \psi_t$, and $k_a, k_r \in (0, 1]$ are parameters that adjust where the projected point is angularly and radially within the FOV, respectively.

4.2.1 Switching Signal and Control Design

The pursuer agent influences the target agent when dictated by the switching signal $\sigma : \mathbb{R} \times \mathbb{R}^n \to \{0, 1\}$. The switching signal is defined as

$$\sigma \triangleq \begin{cases} 1, & \text{if } \|e_t(t)\|^2 > \gamma_s \text{ and } t - t_k \geq \delta_k, \\ 0, & \text{otherwise}, \end{cases} \tag{4–9}$$

where $\gamma_s \triangleq \frac{1}{2}\overline{\nu}_\zeta^2$, $t - t_k$ is the total elapsed time since η_p exited $\mathcal{F}$ the k^{th} time, and δ_k is the minimum dwell time developed in Subsection 4.3.3.

Remark 4.1. The switching signal allows the pursuer agent take advantage of the target agent's dynamics to reduce the target position regulation error. The FOV of the target agent could cause every trajectory correction caused by the pursuer agent to be an over-correction, causing $e_t(t)$ to oscillate and not approach zero. Employing the switching signal in (4–9) allows the target agent's constant velocity dynamics to drive $e_t(t)$ to zero when it is advantageous to do so.

When the switching signal indicates the target agent dynamics are sufficiently reducing $e_t(t)$, the pursuer agent will track an auxiliary desired position to avoid influencing the target agent arbitrarily. The auxiliary desired point, $\eta_d^*(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, is designed as

$$\eta_d^*(t) \triangleq \eta_t(t) + k_d R_{\text{FOV}} \begin{bmatrix} \cos(\phi_t(t) - \pi) \\ \sin(\phi_t(t) - \pi) \end{bmatrix},$$

where $k_d \in \mathbb{R}_{>1}$ is a scaling factor of R_{FOV}, this scaling factor keeps the pursuer agent from unintentionally entering $\mathcal{F}$. When $\sigma = 0$, (4–7) becomes $e_p(t) = \eta_d^*(t) - \eta_p$. Based on the switching signal in (4–9) and the subsequent stability analysis, the control input of the pursuer is designed as

$$\mu_p(t) \triangleq \begin{cases} \frac{d}{dt}\text{proj}(\eta_d(t), \eta_t(t)) + k_p e_p(t) + k_t e_t(t), & \text{if } \sigma = 1 \\ \dot{\eta}_d^*(t) + k_p e_p(t) + k_t e_t(t), & \text{otherwise,} \end{cases} \tag{4–10}$$

where $k_p \in \mathbb{R}_{>0}$ is a user-selected parameter.

4.3 Stability Analysis

The stability analysis considers the system in two parts: a converging subsystem when the pursuer agent is within $\mathcal{F}$ and influences the target agent, and a diverging subsystem when the pursuer agent does not influence the target agent, but does

follow a stand off point. After each subsystem is considered independently, a dwell-time analysis is used to develop stability criteria by considering the combined system as a switched system. For notational brevity, the time dependence of functions and variables is omitted in the following analysis, unless additional clarity is required. In the following subsections, t_k^s and t_k^u represent the k^{th} time instant the pursuer agent entered and exited $\mathcal{F}$, respectively. Specifically, the sequence t_k^s, t_k^u, t_{k+1}^s represents an iteration of entering $\mathcal{F}$ for the k^{th} time, followed by exiting $\mathcal{F}$ for the k^{th} time, and subsequently reentering $\mathcal{F}$ for the $(k+1)^{\text{th}}$ time. Let $\xi = \left[e_p^\top, e_t^\top\right]^\top : \mathbb{R}_{\geq 0} \to \mathbb{R}^{2n}$ represent the concatenated vector of agent regulation errors. Consider the candidate Lyapunov function, $V : \mathbb{R}^{2n} \to \mathbb{R}$, for the stable subsystem

$$V_s(\xi) = \frac{1}{2} e_p^\top e_p + \frac{1}{2} e_t^\top e_t. \tag{4–11}$$

The following stability analysis reveals that all system trajectories converge to the set $\mathcal{B} \triangleq \{\xi \in \mathbb{R}^{2n} : \|\xi(t)\|^2 \leq \frac{\gamma_s}{\lambda_s}\}$, where $\lambda_s \triangleq \min\left(k_p, k_d k_t - k_t - \frac{1}{2}\right)$ determines the rate of convergence, establishing global UUB regulation.

4.3.1 Target Convergence

Theorem 4.1. *For the dynamical systems in (4–1) and (4–4) and any initial state $\xi(t_0)$, the control law given in (4–10) ensures $\xi(t)$ converges exponentially to the set $\mathcal{B}$, where*

$$\|\xi(t)\|^2 \leq \left(\|\xi(t_k^s)\|^2 - \frac{\gamma_s}{\lambda_s}\right) e^{-2\lambda_s\left(t - t_k^s\right)} + \frac{\gamma_s}{\lambda_s} \tag{4–12}$$

for all $t \in [t_k^s, t_k^u)$, given that $k_p > 0$, $k_t > 1$, $k_d > \frac{2k_t+1}{2k_t}$, and Assumption 4.2 is satisfied.

Proof. Taking the total derivative of (4–11) along the trajectories of ξ, and substituting the dynamics of the target agent from (4–1) and the time derivative of (4–7) yields

$$\dot{V}_s(\xi) = e_p^\top \left(\frac{d}{dt}\text{proj}\left(\eta_d(t), \eta_t(t)\right) - \dot{\eta}_p\right) + e_t^\top \left(\dot{\eta}_t - \dot{\zeta}_t\right). \tag{4–13}$$

Substituting (4–4) and (4–10) into (4–13) yields

$$\dot{V}_s(\xi) = e_p^\top \left(\frac{d}{dt}\text{proj}\left(\eta_d(t), \eta_t(t)\right) - \left(\frac{d}{dt}\text{proj}\left(\eta_d(t), \eta_t(t)\right) + k_p e_p + k_t e_t \right) \right) + e_t^\top \left(\dot{\eta}_t - \dot{\zeta}_t \right). \tag{4–14}$$

Rewriting (4–14) as

$$\dot{V}_s(\xi) = -k_p e_p^\top e_p - k_t e_p^\top e_t + e_t^\top \dot{\eta}_t - e_t^\top \dot{\zeta}_t, \tag{4–15}$$

and substituting the target agent dynamics from (4–1) into (4–15) yields

$$\dot{V}_s(\xi) = -k_p e_p^\top e_p - k_t e_p^\top e_t + e_t^\top f\left(\phi_t(t)\right)\left(1 - \mathbb{1}_{\mathcal{F}}\right) + e_t^\top k_t \left(\eta_t - \eta_p\right)\mathbb{1}_{\mathcal{F}} - e_t^\top \dot{\zeta}_t. \tag{4–16}$$

When $t \in [t_k^s, t_k^u)$, $\eta_p \in \mathcal{F}$ and $\mathbb{1}_{\mathcal{F}} = 1$, therefore (4–16) becomes

$$\dot{V}_s(\xi) = -k_p e_p^\top e_p - k_t e_p^\top e_t + k_t e_t^\top \left(\eta_t - \eta_p\right) - e_t^\top \dot{\zeta}_t. \tag{4–17}$$

Substituting (4–6) and (4–7) into (4–17) yields

$$\dot{V}_s(\xi) = -k_p e_p^\top e_p - k_t e_p^\top e_t + k_t e_t^\top \left(e_p + (1 - k_d) e_t\right) + e_t^\top \dot{\zeta}_t. \tag{4–18}$$

Regrouping terms in (4–18) and using the Cauchy-Schwarz inequality and Young's inequality yields

$$\dot{V}_s(\xi) \leq -k_p \|e_p\|^2 - \left(k_d k_t - k_t - \frac{1}{2}\right)\|e_t\|^2 + \frac{1}{2}\left\|\dot{\zeta}_t\right\|^2. \tag{4–19}$$

Given $\left\|\dot{\zeta}_t\right\| \leq \nu_\zeta$ and $\gamma_s = \frac{1}{2}\overline{\nu}_\zeta^2$, then (4–19) can be expressed as

$$\dot{V}_s(\xi) \leq -k_p \|e_p\|^2 - \left(k_d k_t - k_t - \frac{1}{2}\right)\|e_t\|^2 + \gamma_s. \tag{4–20}$$

Given $\lambda_s = \min\left(k_p, k_d k_t - k_t - \frac{1}{2}\right)$ and substituting (4–11) into (4–20) yields

$$\dot{V}_s(\xi) \leq -2\lambda_s V(\xi) + \gamma_s. \tag{4–21}$$

Solving the differential inequality in (4–21) yields

$$V\left(\xi\left(t\right)\right) \leq V\left(\xi\left(t_k^s\right)\right) e^{-2\lambda_s\left(t-t_k^s\right)} + \frac{\gamma_s}{2\lambda_s}\left(1 - e^{-2\lambda_s\left(t-t_k^s\right)}\right). \tag{4–22}$$

Using [115, Def. 4.6] shows that ξ is UUB by

$$\left\|\xi\left(t\right)\right\| \leq \sqrt{\left\|\xi\left(t_k^s\right)\right\|^2 e^{-2\lambda_s\left(t-t_k^s\right)} + \frac{\gamma_s}{\lambda_s}\left(1 - e^{-2\lambda_s\left(t-t_k^s\right)}\right)},$$

for all $t \in [t_k^s, t_k^u)$, which yields the result in (4–12). ■

4.3.2 Target Divergence

Theorem 4.2. *For the dynamical systems in (4–1) and $t \in \left[t_k^u, t_{k+1}^s\right)$, the value of $\xi\left(t\right)$ remains bounded as*

$$\left\|\xi\left(t\right)\right\|^2 \leq \left(\left\|\xi\left(t_k^u\right)\right\|^2 + \frac{\gamma_u}{\lambda_u}\right) e^{2\lambda_u\left(t-t_k^u\right)} - \frac{\gamma_u}{\lambda_u}, \tag{4–23}$$

where $\gamma_u \in \mathbb{R}_{>0}$ is defined as $\gamma_u \triangleq \frac{1}{2}\overline{\nu}_\zeta^2 + \frac{1}{2}\overline{\nu}_t^2 \in \mathbb{R}_{>0}$.

Proof. Taking the total derivative of (4–11) along trajectories of ξ and substituting the dynamics of the target agent from (4–1) yields

$$\dot{V}_u = e_p^\top\left(\dot{\eta}_d^* - \dot{\eta}_p\right) + e_t^\top\dot{\zeta}_t - e_t^\top f\left(\phi_t\left(t\right)\right)\left(1 - \mathbb{1}_{\mathcal{F}}\right) - e_t^\top k_t\left(\eta_t - \eta_p\right)\mathbb{1}_{\mathcal{F}}. \tag{4–24}$$

When $t \in \left[t_k^u, t_{k+1}^s\right)$, $\eta_p \notin \mathcal{F}$, and $\mathbb{1}_{\mathcal{F}} = 0$, (4–24) can be written as

$$\dot{V}_u = e_p^\top\left(\dot{\eta}_d^* - \dot{\eta}_p\right) + e_t^\top\dot{\zeta}_t - e_t^\top f\left(\phi_t\left(t\right)\right). \tag{4–25}$$

Substituting (4–4) and (4–10), where $\sigma = 0$, into (4–25) and applying the Cauchy-Schwarz inequality and Young's inequality gives

$$\dot{V}_u \leq -\left(k_p + \frac{1}{2}\right)\left\|e_p\right\|^2 + \frac{3}{2}\left\|e_t\right\|^2 + \frac{1}{2}\left\|\dot{\zeta}_t\right\|^2 + \frac{1}{2}\left\|f\left(\phi_t\right)\right\|^2. \tag{4–26}$$

Given $\left\|f\left(\phi_t\right)\right\| \leq \overline{\nu}_t, \left\|\dot{\zeta}_t\right\| \leq \overline{\nu}_\zeta$, and $-k_p - \frac{1}{2} \leq k_p$, we can upper bound (4–26) as

$$\dot{V}_u \leq k_p\left\|e_p\right\|^2 + \frac{3}{2}\left\|e_t\right\|^2 + \frac{1}{2}\overline{\nu}_\zeta^2 + \frac{1}{2}\overline{\nu}_t^2. \tag{4–27}$$

Substituting (4–11) into (4–27), and given $\gamma_u \triangleq \frac{1}{2}\overline{\nu}_\zeta^2 + \frac{1}{2}\overline{\nu}_t^2$ and $\lambda_u = \max\left(k_p, \frac{3}{2}\right)$, (4–27) becomes

$$\dot{V}_u \leq 2\lambda_u V_u + \gamma_u. \tag{4–28}$$

Solving the differential inequality in (4–28) yields

$$V_u(t) \leq V_u\left(t_k^u\right) e^{2\lambda_u\left(t-t_k^u\right)} + \frac{\gamma_u}{2\lambda_u}\left(e^{2\lambda_u\left(t-t_k^u\right)} - 1\right), \tag{4–29}$$

for all $t \in \left[t_k^u, t_{k+1}^s\right)$. Substituting (4–11) into (4–29) and simplifying the expression yields the result in (4–23). ■

4.3.3 Combined Stability Analysis

Theorem 4.3. *The combination of the switching function in (4–9) and the control law in (4–10) guarantees that the state ξ remains bounded by*

$$\|\xi(t)\|^2 \leq \left(\|\xi\left(t_k^s\right)\|^2 - \frac{\gamma_s}{\lambda_s}\right) e^{-2\lambda_s\left(t-t_k^s\right)} + \frac{\gamma_s}{\lambda_s}$$

for all $t \in \mathbb{R}_{\geq 0}$ given that $t_{k+1}^s - t_k^u < \delta_k$ and

$$\delta_k \leq \frac{1}{2}\ln\left[\frac{\lambda_u \xi_s + \gamma_u}{\lambda_u \xi_u}\right], \tag{4–30}$$

where $\xi_s \triangleq \|\xi\left(t_k^s\right)\|^2$, and $\xi_u \triangleq \|\xi\left(t_k^u\right)\|^2$.

Proof. The stable subsystem is upper-bounded by (4–12), which can be written as

$$\|\xi(t)\| \leq \sqrt{\left(\xi_s - \frac{\gamma_s}{\lambda_s}\right) e^{-2\lambda_s\left(t-t_k^s\right)} + \frac{\gamma_s}{\lambda_s}}. \tag{4–31}$$

The unstable subsystem is bounded by (4–23), which can also be written as

$$\|\xi(t)\| \leq \sqrt{\left(\xi_u + \frac{\gamma_u}{\lambda_u}\right) e^{2\left(t-t_k^u\right)} - \frac{\gamma_u}{\lambda_u}}. \tag{4–32}$$

Since the influence updates occur instantaneously, $t_{k+1}^u - t_k^s = 0$, and the system predominantly remains in the unstable subsystem. The time interval $\delta_k = t_{k+1}^s - t_k^u$ is the time interval from when the system enters the unstable subsystem until it exits the unstable subsystem. Consider the unstable subsystem in δ_k and the stable subsystem

in the preceding interval $\delta_{k-1} = t_k^u - t_k^s$. Given that the influence the pursuer imparts on the target agent occurs instantaneously, then $\delta_{k-1} = 0$. Let the unstable subsystem be bounded by the stable subsystem, then (4–31) and (4–32) can be expressed as

$$\xi_u e^{2\delta_k} \leq \left(\xi_s - \frac{\gamma_s}{\lambda_s}\right) e^{-2\lambda_s \delta_{k-1}} + \frac{\gamma_s \lambda_u + \lambda_s \gamma_u \left(1 - e^{2\delta_k}\right)}{\lambda_s \lambda_u}. \tag{4–33}$$

Since $\delta_{k-1} = 0$, (4–33) can be bounded by

$$\xi_u e^{2\delta_k} \leq \xi_s + \frac{\gamma_u}{\lambda_u}. \tag{4–34}$$

Solving for δ_k in (4–34) yields the result in (4–30). ■

4.4 Simulation

Numerical experiments demonstrate the efficacy of the controller in (4–10) and the switching function in (4–9). For this simulation, we select $n = 2$, $R_{\text{FOV}} = 3.0$, $\nu_t = 1.0$, $\nu_\zeta = 0.625$, $k_t = 1.0$, $k_p = 5.0$ and $k_d = 5.0$. Given the initial configuration represented by the ○'s in Figure 4-3 and the value of γ_u, the minimum dwell time was found to be $\delta_{\min} = 0.01$ using 4–30. The minimum, maximum, and average intervals between influence actions $\left(t_{k+1}^s - t_k^u\right)$, denoted $\Delta t_{\min}$, $\Delta t_{\max}$, and Δt_{avg} was found to be $\Delta t_{\min} = 0.15$, $\Delta t_{\max} = 1.3$, and $\Delta t_{\text{avg}} = 0.49$.

4.5 Concluding Remarks

This work introduces a novel solution to the indirect control problem where inter-agent interactions can only occur in a limited target agent FOV. By formulating the system as a switched dynamical model with stable and unstable subsystems, this work develops a comprehensive dwell-time analysis to create a switching controller that can regulate a constant speed target agent toward a desired trajectory. A projection function is formulated to move the desired pursuer position into the target agent's FOV when necessary.

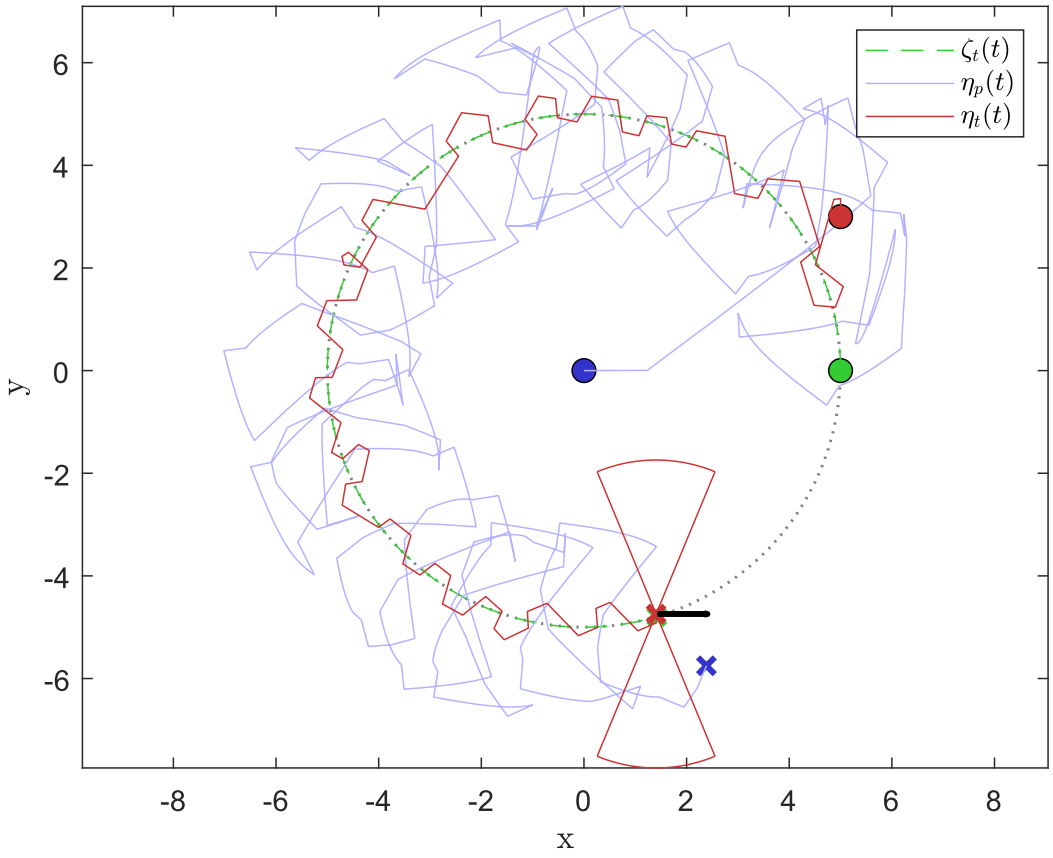

Figure 4-3. The trajectories of the pursuer agent, $\eta_p(t)$, and target agent, $\eta_t(t)$, are represented by the blue and red lines respectively. The desired target agent trajectory, $\zeta_t(t)$, is represented by the green dashed line. The blue, red, and green ○'s represent $\eta_p(t_0)$, $\eta_t(t_0)$, and $\zeta_t(t_0)$. The blue, red and green $\times$'s represent $\eta_p(t_{\text{final}})$, $\eta_t(t_{\text{final}})$, and $\zeta_t(t_{\text{final}})$.

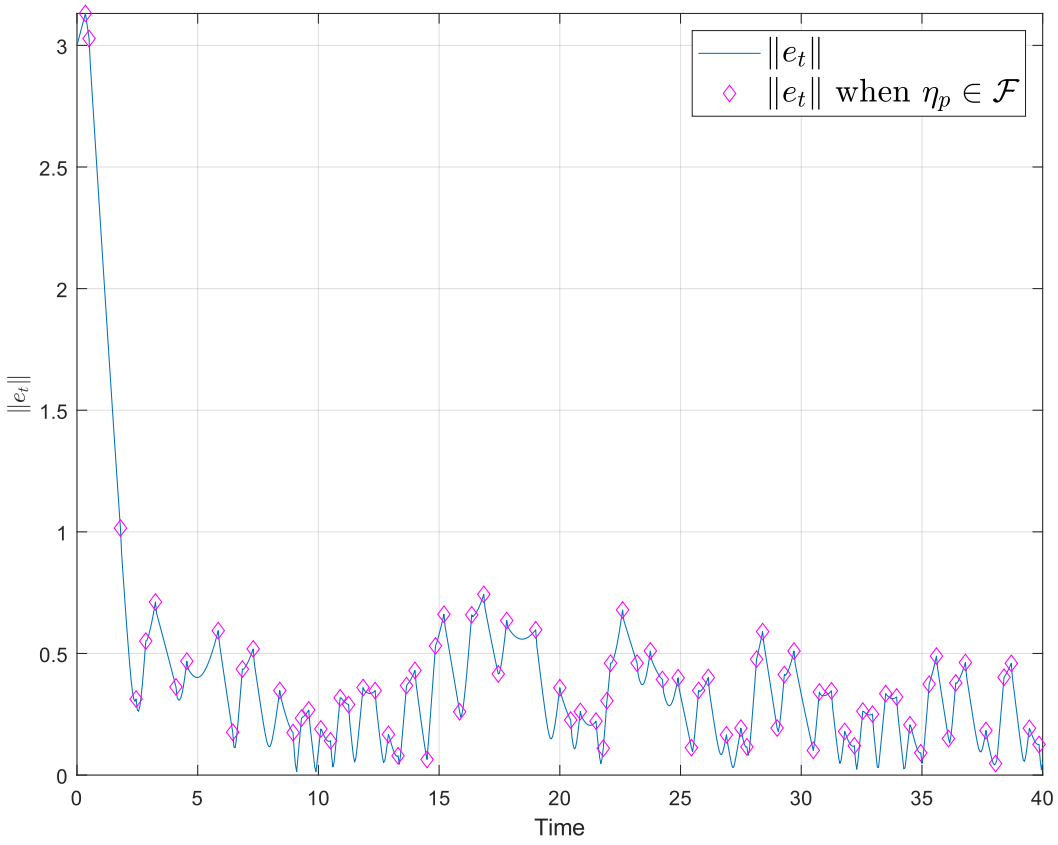

Figure 4-4. The value of $\|e_t\|$ over time. The magenta $\Diamond$'s represent when the pursuer agent entered $\mathcal{F}$ and influence the target agent.

CHAPTER 5
CONCLUSIONS

Indirect control has the potential to reduce the cost-capability gap and allow a team of highly-capable agents to guide many objective-capable agents to a goal or along a desired trajectory. The event-triggered indirect control framework allows for the constraints of sensor-based robotic systems to be considered when developing control laws and communication paradigms. When developing event-triggered or switched control solutions, it is imperative to show that the proposed design achieves a physically feasible trigger rate. The results presented in Chapters 2-4 are shown to achieve implementable event-triggered and state-dependent switching results that are shown to be Zeno-free.

Chapter 2 introduces a distributed event-triggered coordination strategy for multi-agent systems with homogeneous dynamics, ensuring that communication events occur no more frequently than a user-defined minimum interval. The resulting control method is proven to be PAS, and maintains the forward invariance of the agents' workspace, thereby ensuring safety. Furthermore, by appropriately selecting the event-triggering mechanism and bounding the control input, the desired minimum inter-event time can be guaranteed.

Future extensions of the work in Chapter 2 could focus on extending the event-triggered consensus strategy to heterogeneous multi-agent systems, where agents may have differing dynamics or capabilities. Another direction could involve relaxing the assumptions about network connectivity or allowing for time-varying communication delays, thereby making the approach more robust and applicable to real-world scenarios. Additionally, investigating adaptive mechanisms for dynamically tuning the minimum IET, to optimize performance while preserving the user-specified minimum IET, could further improve efficiency and safety in dynamic environments.

Chapter 3 presents an innovative approach to the indirect control problem by using an event-triggered relative direction influence function to guide a target agent toward a desired position. It establishes a rigorous link between maximum dwell time and inter-event time (IET) bounds within the event-triggered indirect control framework. By modeling the system as a switched dynamical system with both stable and unstable modes, a detailed dwell-time analysis is used to derive an upper bound on the IET for the triggering mechanism. A piecewise continuous trigger function is introduced, incorporating a maximum IET, which is determined through the dwell-time analysis, and a configurable minimum IET, offering flexibility in control design.

Future work for Chapter 3 includes the design of bounded control laws for pursuer agents to facilitate a closed-form characterization of the minimum inter-event time (IET), enhancing both theoretical clarity and practical implementation. Another key direction is the extension of the current analytical results to scenarios involving bearing-only measurements, which reflect more realistic sensing constraints in many applications. Furthermore, future research will investigate the scalability of the proposed event-triggered indirect control framework to systems with multiple pursuer and target agents, including the coordination and communication challenges that arise in such multi-agent environments. Additional efforts may also focus on robustness analysis under modeling uncertainties and external disturbances to further broaden the applicability of the approach.

Chapter 4 presents a solution to the indirect control problem where inter-agent interactions can only occur in a limited target agent FOV. Similar to Chapter 3, the pursuer and target agents are modeled as a switched dynamical system with stable and unstable subsystems, but in Chapter 4 a comprehensive dwell-time criteria enforces the switching controller to prevent Zeno behavior while regulating a constant speed target agent toward a desired trajectory. Chapter 4 also presents a projection function which moves the desired pursuer position into the target agent's FOV - creating a sub-optimal

desired pursuer position in terms of regulating the target agent position error, but is optimal in terms of effectively indirectly controlling the target agent.

The work in Chapter 4 can be characterized as primarily reactive to system errors, and future research will focus on developing more predictive control strategies—such as incorporating target motion forecasting or model-based estimation techniques—to improve responsiveness and control efficiency. Additionally, future work will explore the scalability of this approach to scenarios involving multiple target agents, including challenges related to FOV constraints, interaction scheduling, and decentralized coordination among multiple pursuers. Further extensions may also consider dynamic or adaptive FOV models, as well as robustness to sensor noise and communication delays, to enhance the applicability of the framework in real-world environments.

The work in Chapters 3-4 are impactful in addressing problems related to the indirect control of flocks or swarms of agents. Specifically, these chapters provide mechanisms by which a limited number of pursuer agents can strategically influence the movement and formation of larger groups of target agents by exploiting localized interactions, dwell-time-based switching logic, and visibility-constrained control laws to guide collective behavior toward desired global objectives. The work in Chapters 2-4 is relevant in scenarios where direct control of every agent is infeasible, such as in surveillance, environmental monitoring, or autonomous exploration tasks. The analytical tools and control frameworks developed in these chapters lay the groundwork for scalable and resource-efficient coordination strategies in multi-agent systems, where indirect influence must be both deliberate and adaptable to system variability.

LIST OF REFERENCES

[1] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[2] E. Panteley and A. Loría, "Synchronization and dynamic consensus of heterogeneous networked systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3758–3773, 2017.

[3] C. Sun and K. G. Vamvoudakis, "Continuous-time safe learning with temporal logic constraints in adversarial environments," in *Proc. Am. Control Conf.*, 2020, pp. 4786–4791.

[4] D. V. Dimarogonas and K. H. Johansson, "Event-triggered cooperative control," in *Proc. European Control Conf.*, Aug. 2009, pp. 3015–3020.

[5] W. Heemels, K. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proc. IEEE Conf. Decis. Control*, Dec. 2012, pp. 3270–3285.

[6] M. Mazo and P. Tabuada, "Decentralized event-triggered control over wireless sensor/actuator networks," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2456–2461, 2011.

[7] M. Mazo and M. Cao, "Decentralized event-triggered control with asynchronous updates," in *Proc. IEEE Conf. Decis. Control Eur. Control Conf.*, pp. 2547–2552.

[8] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440.

[9] F. M. Zegers, D. P. Guralnik, and W. E. Dixon, "Event/self-triggered multi-agent system rendezvous with graph maintenance," in *Proc. IEEE Conf. Decis. Control*, pp. 1886–1891.

[10] F. Zegers, M. Hale, J. Shea, and W. E. Dixon, "Event-triggered formation control and leader tracking with resilience to byzantine adversaries: A reputation-based approach," *IEEE Trans. Netw. Syst.*, vol. 8, no. 3, pp. 1417–1429, 2021.

[11] F. Zegers, M. Hale, J. M. Shea, and W. E. Dixon, "Reputation-based event-triggered formation control and leader tracking with resilience to byzantine adversaries," in *Proc. Am. Control Conf.*, 2020, pp. 761–766.

[12] N. Mathew, S. L. Smith, and S. L. Waslander, "A graph-based approach to multi-robot rendezvous for recharging in persistent tasks," in *IEEE Int. Conf. Robot. Automat.*, pp. 3497–3502.

[13] J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, "Collaborative multi-robot search

and rescue: Planning, coordination, perception, and active vision," *IEEE Access*, vol. 8, pp. 191 617–191 643.

[14] Z. Zhou, J. Liu, and J. Yu, "A survey of underwater multi-robot systems," *IEEE/CAA J. Autom. Sin.*, vol. 9, no. 1, pp. 1–18.

[15] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1680–1685, Sep. 2007.

[16] D. P. Borgers and W. Heemels, "On minimum inter-event times in event-triggered control," in *Proc. IEEE Conf. Decis. Control*, 2013, pp. 7370–7375.

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

[18] W. Heemels, J. Sandee, and P. Van Den Bosch, "Analysis of event-driven controllers for linear systems," *Int. J. Control*, vol. 81, no. 4, pp. 571–590.

[19] W. Ren and R. W. Beard, *Distributed Consensus in Multi-Vehicle Cooperative Control*. New York: Springer-Verlag, 2008.

[20] D. Dimarogonas, E. Frazzoli, and K. Johansson, "Distributed self-triggered control for multi-agent systems," *Proc. IEEE Conf. Decis. Control*, pp. 6716–6721.

[21] 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.

[22] J. Yang, "A consensus control for a multi-agent system with unknown time-varying communication delays," *IEEE Access*, vol. 9, pp. 55 844–55 852.

[23] M. Ajina, D. Tabatabai, and C. Nowzari, "Asynchronous distributed event-triggered coordination for multiagent coverage control," *IEEE Trans. Cybern.*, vol. 51, no. 12, pp. 5941–5953.

[24] J. Berneburg and C. Nowzari, "Distributed dynamic event-triggered coordination with a designable minimum inter-event time," in *Proc. Am. Control Conf.*, pp. 1424–1429.

[25] ——, "Robust dynamic event-triggered coordination with a designable minimum interevent time," *IEEE Trans. Autom. Contr.*, vol. 66, no. 8, pp. 3417–3428, 2021.

[26] C. Chen, F. L. Lewis, and X. Li, "Event-triggered coordination of multi-agent systems via a lyapunov-based approach for leaderless consensus," *Automatica*, vol. 136, p. 109936.

[27] Y. Liu, C. Nowzari, Z. Tian, and Q. Ling, "Asynchronous periodic event-triggered coordination of multi-agent systems," *Proc. IEEE Conf. Decis. Control*, pp. 6696–6701.

[28] H. Liu, L. Cheng, M. Tan, and Z.-G. Hou, "Exponential finite-time consensus of fractional-order multiagent systems," *IEEE Trans. Syst. Man Cybern. Syst.*, 2018.

[29] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[30] X. Liu, S. S. Ge, C.-H. Goh, and Y. Li, "Event-triggered coordination for formation tracking control in constrained space with limited communication," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1000–1011.

[31] C. Nowzari and J. Cortés, "Self-triggered coordination of robotic networks for optimal deployment," *Automatica*, vol. 48, no. 6, pp. 1077–1087.

[32] R. Postoyan, R. G. Sanfelice, and W. M. H. Heemels, "Inter-event times analysis for planar linear event-triggered controlled systems," in *Proc. IEEE Conf. Decis. Control*, pp. 1662–1667.

[33] A. Rajan and P. Tallapragada, "Analysis of inter-event times for planar linear systems under a general class of event triggering rules," in *Proc. IEEE Conf. Decis. Control*, pp. 5206–5211.

[34] M. Ghodrat and H. Marquez, "Event-triggered design with guaranteed minimum interevent times and $\mathcal{L}_p$ performance," *IEEE Trans. Autom. Control*, vol. 65, no. 4, pp. 1668–1675.

[35] M. Li, Y. Long, T. Li, H. Liang, and C. P. Chen, "Dynamic event-triggered consensus control for input constrained multi-agent systems with a designable minimum inter-event time," *IEEE/CAA J. Autom. Sin.*, vol. 10, pp. 1–12.

[36] M. Li, Y. Long, T. Li, and C. P. Chen, "Consensus of linear multi-agent systems by distributed event-triggered strategy with designable minimum inter-event time," *Inf. Sci.*, vol. 609, pp. 644–659.

[37] H. Xie, B. Wu, and F. Bernelli-Zazzera, "High minimum inter-execution time sigmoid event-triggered control for spacecraft attitude tracking with actuator saturation," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 2, pp. 1349–1363.

[38] F. Zegers, D. Guralnik, S. Edwards, C.-L. Lee, and W. E. Dixon, "Event-triggered consensus for second-order systems: A hybrid systems perspective," in *Proc. IEEE Conf. Decis. Control*.

[39] J.-M. Li, C.-W. Chen, and T.-H. Cheng, "Motion prediction and robust tracking of a dynamic and temporarily-occluded target by an unmanned aerial vehicle," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 4, pp. 1623–1635, 2020.

[40] D. Dimarogonas and E. Frazzoli, “Distributed event-triggered control strategies for multi-agent systems,” *Proc. Annu. Allerton Conf. Commun. Control Comput.*, pp. 906–910.

[41] D. V. Dimarogonas and K. H. Johansson, “Event-triggered control for multi-agent systems,” in *Proc. IEEE Conf. Decis. Control*, Dec. 2009, pp. 7131–7136.

[42] Y. Fan, L. Liu, G. Feng, and Y. Wang, “Self-triggered consensus for multi-agent systems with Zeno-free triggers,” *IEEE Trans. Autom. Control*, vol. 60, no. 10, pp. 2779–2784, 2015.

[43] W. Hu, L. Liu, and G. Feng, “Consensus of linear multi-agent systems by distributed event-triggered strategy,” *IEEE Trans. Cybern.*, vol. 46, pp. 148–157.

[44] A. A. Soderlund and S. Phillips, “Hybrid systems approach to autonomous rendezvous and docking of an underactuated satellite,” *J. Guid. Control Dyn.*, vol. 46, no. 10, pp. 1901–1918, 2023.

[45] J. A. V. Trejo, D. Rotondo, M. A. Medina, and D. Theilliol, “Robust observer-based leader-following consensus for a class of nonlinear multi-agent systems: application to UAV formation control,” in *Intl. Conf. Unmann. Aircraft Syst.*

[46] H. Wang, “Flocking of networked uncertain Euler-Lagrange systems on directed graphs,” *Automatica*, vol. 49, no. 9, pp. 2774–2779, 2013.

[47] M. Maghenem, E. Panteley, and A. Loria, “Singular-perturbations-based analysis of dynamic consensus in directed networks of heterogeneous nonlinear systems,” *IEEE Trans. Autom. Control*.

[48] K. Astrom and B. Bernhardsson, “Comparison of periodic and event based sampling for first-order stochastic systems,” in *Proc. IFAC World Congress*, Beijing, China, 1999, pp. 301–306.

[49] X. Wang and M. Lemmon, “Event-triggering in distributed networked systems with data dropouts and delays,” *Int. Workshop Hybrid Syst. Comput. Control*.

[50] ——, “Event-triggering in distributed networked control systems,” *IEEE Trans. Autom. Control*, vol. 56, pp. 586–601.

[51] J. Lunze and D. Lehmann, “A state-feedback approach to event-based control,” *Automatica*, vol. 46, pp. 211–215.

[52] R. Olfati-Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[53] G. S. Seyboth, D. V. Dimarogonas, and K. H. Johansson, “Event-based broadcasting for multi-agent average consensus,” *Automatica*, vol. 49, pp. 245–252, Jan. 2013.

[54] W. Zhu, Z.-P. Jiang, and G. Feng, “Event-based consensus of multi-agent systems with general linear models,” *Automatica*, vol. 50, pp. 552–558.

[55] G. Guo, L. Ding, and Q.-L. Han, “A distributed event-triggered transmission strategy for sampled-data consensus of multi-agent systems,” *Automatica*, vol. 50, no. 5, pp. 1489–1496.

[56] X. Meng and T. Chen, “Event based agreement protocols for multi-agent networks,” *Automatica*, vol. 49, pp. 2125–2132, Jul. 2013.

[57] D. Dimarogonas and K. Johansson, “Bounded control of network connectivity in multi-agent systems,” *Control Theory Appl.*, vol. 4, no. 8, pp. 1330–1338, Aug. 2010.

[58] A. Anta and P. Tabuada, “To sample or not to sample: Self-triggered control for nonlinear systems,” *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2030–2042.

[59] Y. Fan and G. Hu, “Connectivity-preserving rendezvous of multi-agent systems with event-triggered controllers,” in *Proc. IEEE Conf. Decis. Control*, 2015, pp. 234–239.

[60] X. Yi, K. Liu, D. V. Dimarogonas, and K. H. Johansson, “Dynamic event-triggered and self-triggered control for multi-agent systems,” *IEEE Trans. Autom. Control*, 2018.

[61] W. Tabib, K. Goel, J. Yao, C. Boirum, and N. Michael, “Autonomous cave surveying with an aerial robot,” *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 1016–1032.

[62] K. D. Listmann, M. V. Masalawala, and J. Adamy, “Consensus for formation control of nonholonomic mobile robots,” in *IEEE Int. Conf. Robot. Automat.* IEEE, pp. 3886–3891.

[63] D. Lapandić, L. Persson, D. V. Dimarogonas, and B. Wahlberg, “Aperiodic communication for mpc in autonomous cooperative landing,” *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 113–118.

[64] J. Qin, Q. Ma, Y. Shi, and L. Wang, “Recent advances in consensus of multi-agent systems: A brief survey,” *IEEE Trans. Indus. Electron.*, vol. 64, pp. 4972–4983.

[65] F. Zegers, S. Phillips, and W. E. Dixon, “Consensus over clustered networks with asynchronous inter-cluster communication,” in *Proc. Am. Control Conf.*, pp. 4249–4254.

[66] J. P. Hespanha, Z. Dodds, G. D. Hager, and A. S. Morse, “What tasks can be performed with an uncalibrated stereo vision system?” *Int. J. Comput. Vision*, vol. 35, no. 1, pp. 65–85, 1999.

[67] J. P. Hespanha, M. Prandini, and S. Sastry, “Probabilistic pursuit-evasion games: A one-step Nash approach,” in *Proc. IEEE Conf. Decis. Control*, vol. 3, 2000, pp. 2272–2277.

[68] R. Vidal, O. Shakernia, H. Kim, D. Shim, and S. Sastry, “Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation,” *IEEE Trans. Robot. and Autom.*, vol. 18, no. 5, pp. 662–669, Oct. 2002.

[69] J. Chen, W. Zha, Z. Peng, and D. Gu, “Multi-player pursuit–evasion games with one superior evader,” *Automatica*, vol. 71, pp. 24–32, 2016.

[70] D. W. Oyler, P. T. Kabamba, and A. R. Girard, “Pursuit–evasion games in the presence of obstacles,” *Automatica*, vol. 65, pp. 1–11.

[71] M. V. Ramana and M. Kothari, “Pursuit-evasion games of high speed evader,” *J. Intell. Rob. Syst.*, vol. 85, no. 2, pp. 293–306, 2017.

[72] T. H. Chung, G. A. Hollinger, and V. Isler, “Search and pursuit-evasion in mobile robotics,” *Autonomous Robots*, vol. 31, no. 4, p. 299, 2011.

[73] W. Zha, J. Chen, Z. Peng, and D. Gu, “Construction of barrier in a fishing game with point capture,” *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1409–1422.

[74] R. Licitra, Z. I. Bell, E. Doucette, and W. E. Dixon, “Single agent indirect herding of multiple targets: A switched adaptive control approach,” *IEEE Control Syst. Lett.*, vol. 2, no. 1, pp. 127–132, January 2018.

[75] P. Deptula, Z. I. Bell, E. Doucette, J. W. Curtis, and W. E. Dixon, “Data-based reinforcement learning approximate optimal control for an uncertain nonlinear system with partial loss of control effectiveness,” in *Proc. Am. Control Conf.*, 2018, pp. 2521–2526.

[76] P. Deptula, Z. Bell, F. Zegers, R. Licitra, and W. E. Dixon, “Approximate optimal influence over an agent through an uncertain interaction dynamic,” *Automatica*, vol. 134, pp. 1–13, Dec. 2021.

[77] R. Licitra, Z. Hutcheson, E. Doucette, and W. E. Dixon, “Single agent herding of n-agents: A switched systems approach,” in *IFAC World Congr.*, 2017, pp. 14 374–14 379.

[78] R. Licitra, Z. Bell, and W. Dixon, “Single agent indirect herding of multiple targets with unknown dynamics,” *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 847–860, 2019.

[79] A. Pierson and M. Schwager, “Bio-inspired non-cooperative multi-robot herding,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 1843–1849.

[80] A. Pierson, Z. Wang, and M. Schwager, “Intercepting rogue robots: An algorithm for capturing multiple evaders with multiple pursuers,” *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 530–537, 2017.

[81] A. Pierson and M. Schwager, “Controlling noncooperative herds with robotic herders,” *IEEE Trans. Robot.*, vol. 34, no. 2, pp. 517–525, 2018.

[82] B. Bennett and M. Trafankowski, “A comparative investigation of herding algorithms,” in *Proc. Symp. Underst. Model. Collect. Phenom.*, 2012, pp. 33–38.

[83] N. K. Long, K. Sammut, D. Sgarioto, M. Garratt, and H. A. Abbass, “A comprehensive review of shepherding as a bio-inspired swarm-robotics guidance approach,” in *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 4, no. 4, 2020, pp. 523–537.

[84] J. Zhi and J. Lien, “Learning to herd agents amongst obstacles: Training robust shepherding behaviors using deep reinforcement learning,” *IEEE Robot. Autom. Lett.*, vol. 6, pp. 4163–4168, 2020.

[85] ——, “Learning to herd amongst obstacles from an optimized surrogate,” *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 2954–2961, 2022.

[86] D. Strömbom, R. P. Mann, A. M. Wilson, S. Hailes, A. J. Morton, D. J. T. Sumpter, and A. J. King, “Solving the shepherding problem: heuristics for herding autonomous, interacting agents,” *J. of the R. Society Interface*, vol. 11, no. 100, 2014.

[87] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, and S. Cameron, “Experiments in automatic flock control,” *Robot. Autom. Syst.*, vol. 31, no. 1, pp. 109–117, 2000.

[88] C. W. Reynolds, “Flocks, herds and schools: A distributed behavioral model,” in *Proc. 14th Annu. Conf. Comput. Graph. Interact. Tech.*, pp. 25–34.

[89] O. B. Bayazit, J.-M. Lien, and N. M. Amato, “Roadmap-based flocking for complex environments,” in *10th Pac. Conf. Comput. Graph. Appl.* IEEE, pp. 104–113.

[90] V. Chipade and D. Panagou, “Herding an adversarial attacker to a safe area for defending safety-critical infrastructure,” in *Proc. Am. Control Conf.*, 2019, pp. 1035–1041.

[91] V. S. Chipade and D. Panagou, “Herding an adversarial swarm in an obstacle environment,” *Proc. IEEE Conf. Decis. Control*, pp. 3685–3690.

[92] A. Pierson, A. Ataei-Esfahani, I. Paschalidis, and M. Schwager, “Cooperative multi-quadrotor pursuit of an evader in an environment with no-fly zones,” *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 320–326, 2016.

[93] E. Sebastián and E. Montijano, “Multi-robot implicit control of herds,” in *IEEE Int. Conf. Robot. Autom.*, 2021, pp. 1601–1607.

[94] E. Sebastián, E. Montijano, and C. Sagues, “Adaptive multirobot implicit control of heterogeneous herds,” *IEEE Trans. Robot.*, vol. 38, no. 6, pp. 3622–3635, 2022.

[95] K. Fujioka and S. Hayashi, "Effective shepherding behaviours using multi-agent systems," in *Proc. IEEE Region 10 Conf.* IEEE, pp. 3179–3182.

[96] A. A. Paranjape, S.-J. Chung, K. Kim, and D. H. Shim, "Robotic herding of a flock of birds using an unmanned aerial vehicle," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 901–915, 2018.

[97] J.-M. Lien, O. B. Bayazit, R. T. Sowell, S. Rodriguez, and N. M. Amato, "Shepherding behaviors," in *Proc. IEEE Int. Conf. on Robot. and Autom.*, vol. 4, Apr. 2004, pp. 4159–4164.

[98] Y. Tsunoda, Y. Sueoka, Y. Sato, and K. Osuka, "Analysis of local-camera-based shepherding navigation," *Advanced Robotics*, vol. 32, pp. 1217 – 1228.

[99] A. Parikh, R. Kamalapurkar, and W. E. Dixon, "Integral concurrent learning: Adaptive control with parameter convergence using finite excitation," *Int. J. Adapt. Control Signal Process.*, vol. 33, no. 12, pp. 1775–1787.

[100] R. Kamalapurkar, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for approximate optimal regulation," *Automatica*, vol. 64, pp. 94–104.

[101] W. A. Makumi, Z. I. Bell, and W. E. Dixon, "Approximate optimal indirect regulation of an unknown agent with a lyapunov-based deep neural network," *IEEE Control Syst. Lett.*

[102] V. S. Chipade and D. Panagou, "Multiagent planning and control for swarm herding in 2-d obstacle environments under bounded inputs," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1956–1972, 2021.

[103] R. A. Licitra, A. J. Neale, E. A. Doucette, and J. W. Curtis, "Adversarial aircraft diversion and interception using missile herding techniques," in *Micro-and Nanotechnology Sensors, Systems, and Applications XI*, vol. 10982. SPIE, 2019, pp. 335–343.

[104] J.-M. Lien, S. Rodriguez, J.-P. Malric, and N. M. Amato, "Shepherding behaviors with multiple shepherds," in *Proc. IEEE Int. Conf. on Robot. and Autom.*, 2005, pp. 3402–3407.

[105] H. Song, A. Varava, O. Kravchenko, D. Kragic, M. Wang, F. T. Pokorny, and K. Hang, "Herding by caging: a formation-based motion planning framework for guiding mobile agents," *Autonomous Robots*, vol. 45, pp. 613 – 631.

[106] A. Varava, K. Hang, D. Kragic, and F. T. Pokorny, "Herding by caging: a topological approach towards guiding moving agents via mobile robots." *Robotics: Science and Systems*, pp. 1–9, 2017.

[107] T. Miki and T. Nakamura, "An effective simple shepherding algorithm suitable for implementation to a multi-mobile robot system," in *Proc. of the First Int. Conf.*

on Innov. Computing, Inf., and Control, vol. 3, Washington, DC, USA, 2006, pp. 161–165.

[108] F. Auletta, D. Fiore, M. J. Richardson, and M. di Bernardo, "Herding stochastic autonomous agents via local control rules and online target selection strategies," *Autonomous Robots*, vol. 46, pp. 469 – 481.

[109] M. Bacon and N. Olgac, "Swarm herding using a region holding sliding mode controller," *J. Vib. Control*, vol. 18, no. 7, pp. 1056–1066, 2012.

[110] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems*. Princeton University Press, 2012.

[111] D. Liberzon, *Switching in Systems and Control*. Birkhauser, 2003.

[112] R. Goebel, R. G. Sanfelice, and A. R. Teel, "Hybrid dynamical systems," *IEEE Control Syst. Mag.*, vol. 29, no. 2, pp. 28–93, 2009.

[113] Z. Cai, M. S. de Queiroz, and D. M. Dawson, "A sufficiently smooth projection operator," *IEEE Trans. Autom. Control*, vol. 51, no. 1, pp. 135–139, Jan. 2006.

[114] T.-H. Cheng, Z. Kan, J. R. Klotz, J. M. Shea, and W. E. Dixon, "Event-triggered control of multiagent systems for fixed and time-varying network topologies," *IEEE Trans. Autom. Control*, vol. 62, no. 10, pp. 5365–5371.

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

BIOGRAPHICAL SKETCH

Patrick McGill Amy received his bachelor's degree in ocean engineering from the Florida Institute of Technology in 2017. The following year, he continued his academic career at the Johns Hopkins University. Patrick finished his master's degree in robotics at the Johns Hopkins University while working at the Naval Research Laboratory in 2019. Patrick moved to Florida and started working at the Naval Surface Warfare Center Panama City Division in 2019. In 2021, Patrick joined the Nonlinear Controls and Robotics Laboratory to conduct research and pursue a Ph.D. under the advisement of Dr. Warren Dixon. Patrick received his Ph.D. from the University of Florida in August 2025. Patrick's research focuses on developing stability methods and stability criteria to control nonlinear sensor-based multi-agent systems.