A TRIDENT SCHOLAR PROJECT REPORT

NO. 450

Communication Dependent Control of Multi-Vehicle Formations

by

Midshipman 1/C Aaron M. Sims, USN

UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND

USNA-1531-2

U.S.N.A. --- Trident Scholar project report; no. 450 (2016)

COMMUNICATION DEPENDENT CONTROL OF MULTI-VEHICLE FORMATIONS

by

Midshipman 1/C Aaron M. Sims
United States Naval Academy
Annapolis, Maryland

_____
(signature)

Certification of Adviser(s) Approval

Assistant Professor Levi D. DeVries
Systems Engineering Department

_____
(signature)

_____
(date)

VADM (Ret.) Charles I. Leidig, USN
Mechanical Engineering Department

_____
(signature)

_____
(date)

Acceptance for the Trident Scholar Committee

Professor Maria J. Schroeder
Associate Director of Midshipman Research

_____
(signature)

_____
(date)

USNA-1531-2

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
05-11-2016		

4. TITLE AND SUBTITLE
Communication Dependent Control of Multi-Vehicle Formations

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Sims, Aaron M.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Naval Academy
Annapolis, MD 21402

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
Trident Scholar Report no. 450 (2016)

12. DISTRIBUTION / AVAILABILITY STATEMENT

This document has been approved for public release; its distribution is UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The expanded role of unmanned systems in military and civilian applications has introduced interesting new questions in multi-vehicle coordination and communication. The goal of this research is to derive steering algorithms utilizing event- and self-triggered control, capable of driving vehicles to a particular formation while simultaneously coordinating inter-agent communication instances (i.e. surfacing times). This goal is broken up into two segments: multi-vehicle control and communication control. By leveraging previous work in tracking control, we show the ability to combine kinematic and vehicle dynamic controllers to create a model that easily adapts to changing vehicle dynamics and time-varying desired configurations. Target and trajectory tracking are demonstrated, both of which are useful in conducting autonomous missions. Additionally, we present a permutation algorithm to optimally assign vehicles to formation positions based on proximity.

This work combines the multi-vehicle and communication-coordinating control to present a framework capable of steering a multi-vehicle system to a time-varying configuration without relying on all-to-all communication. The contributions of this work may enhance the capability of underwater sampling by increasing formation precision subject to intermittent communication.

15. SUBJECT TERMS
Consensus, Tracking, Multi-vehicle Control, Laplacian, Coordination, Self-Triggered Control

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**		33	**19b. TELEPHONE NUMBER** *(include area code)*

Communication Dependent Control of Multi-Vehicle Formations

MIDN Aaron M. Sims, USN

The expanded role of unmanned systems in military and civilian applications has introduced interesting new questions in multi-vehicle coordination and communication. Specifically, in communication-limited or denied environments, there exists a need for control algorithms that drive agents to desired formation in space while coordinating the time instances in which they communicate. In the undersea domain, communication is limited underwater since it compromises stealth and signals attenuate at relatively short spatial scales compared to the air domain. Consequently, there exists a need for multi-vehicle control algorithms capable of driving vehicles to a dynamic configuration subject to directed, time delayed communication topologies amongst the vehicles.

To combat these issues and deliver a system that allows underwater vehicles to communicate without reliance on an underwater network, coordination of vehicle surfacing is essential. The goal of this research is to derive steering algorithms utilizing event- and self-triggered control, capable of driving vehicles to a particular formation while simultaneously coordinating inter-agent communication instances (i.e. surfacing times). This goal is broken up into two segments: multi-vehicle control and communication control.

By leveraging previous work in tracking control, we show the ability to combine kinematic and vehicle dynamic controllers to create a model that easily adapts to changing vehicle dynamics and time-varying desired configurations. Target and trajectory tracking are demonstrated, both of which are useful in conducting autonomous missions. Additionally, we present a permutation algorithm to optimally assign vehicles to formation positions based on proximity.

This work also leverages theories from self-triggering control and builds upon them by accounting for the time delayed communication. In this work, each agent computes its own control based on the last known information of its communicating neighbors. The information includes the time, position, and velocity of the neighbor vehicles at their last known communication time. Using the provided information, agents calculate their own control and the next interval update time. Each agent uses the last shared information from its communicating neighbors to determine its own next execution time independent of the other agent's execution times. The results guarantee convergence to a desired configuration. We compare the results of self-triggered control strategies in which all agents update their control at each time instance to our algorithm which relies on delayed information to update the control of individual agents based on their unique computed next execution time.

In total, this work combines the multi-vehicle and communication-coordinating control to present a framework capable of steering a multi-vehicle system to a time-varying configuration without relying on all-to-all communication. The contributions of this work may enhance the capability of underwater sampling by increasing formation precision subject to intermittent communication. This may be especially beneficial for applications such as underwater surveillance, environmental sampling, or underwater threat detection.

Keywords- Consensus, Tracking, Multi-vehicle Control, Laplacian, Coordination, Self-Triggered Control

1 INTRODUCTION

Both the military and private sector have expressed interest in unmanned multi-vehicle systems to conduct missions currently requiring manned craft. As a result, there exists a need to control multi-vehicle systems subject to realistic communication limitations. This paper addresses this need by deriving an algorithm capable of steering vehicles to desired configurations subject to practical limitations on communication between agents.

Consider a multi-vehicle system in which each agent computes its steering control based on information communicated to it by its network of neighbors. This system of agents, called a decentralized system of agents, is subject to practical limitations in how or when it can communicate information. The goal of this research is to design a controller for each agent to steer the vehicles to a desired configuration subject to the communication constraints imposed by operating in a realistic environment. Specifically, we aim to leverage precious work in self-triggered control to derive new steering algorithms that steer vehicles to desired configurations while simultaneously coordinating the timing at which they communicate.

The project is broken down into two segments. First, we derive a decentralized algorithm to steer a set of N vehicles to a desired configuration subject to a pre-determined communication network. To achieve the configuration with a realistic vehicle model, we pursue a two-part control approach. First, we generate reference trajectories assuming simple kinematic agents. Then, we track the reference trajectories with higher fidelity dynamic vehicle models. The desired configuration can be a specific, spatial position, an orbit in a desired parametric shape around a target, or a path that tracks a moving target. Additionally, the results of the research include implementation of a network flow optimization routine to demonstrate the benefits of vehicle communication in agreeing on vehicle assignment to a set of end-state positions.

The second phase of the project focuses on implementing self triggering control to allow for relaxed communication constraints among the vehicles. The results of the research include the derivation of a new self-triggering condition that allows for individual vehicles to update their control as opposed to current strategies that require all agents to communicate at each time instance. Further, the consensus algorithm, which steers all vehicles to a common position, was adapted to allow for the steering of the vehicles to any desired formation utilizing the self triggered control strategy.

The time delayed communication aspect is addressed by applying theories from event and self-triggered control from [1] and [2]. However, the focus of the control law presented is for a multi-agent systems where the agents compute their individual control law based on the last known information of the other agents. The distributed self-triggered control strategy is used to calculate the next time for each individual vehicle to update its control.

The contributions of this work are 1) a kinematic controller that steers vehicles to any desired configuration, 2) a tracking controller that steers high fidelity vehicle dynamics to track the kinematic reference trajectories, and 3) a distributed self-triggering control that coordinates communication instances between neighboring agents.

The paper is organized as follows. The following subsection outlines the specific problem this work addresses. Section II provides a mathematical introduction and overview of the key components of the research. Section III introduces and derives a graph transformation for steering vehicles to a desired formation. Section IV derives the theoretical results

beginning with the kinematic controller and working up to self-triggering and tracking control. Section V illustrates a numerical simulation of the self-triggering and tracking control applied to tracking a moving target. Section VI concludes the paper by summarizing the accomplishments and proposing future areas for research focus.

1.1 Problem Statement

In an ideal case, the vehicles will operate under all-to-all (each vehicle communicates directly with every other vehicle) instantaneous communication. However, real applications impose constraints that may make all-to-all communication unrealistic. For example, in applications with obstacles between vehicles that require line of sight data transmission, all-to-all communication is not achievable.

Take for example, an undersea application where stealth is paramount. In this case, the unmanned underwater vehicles (UUV) would not want to communicate underwater if the acoustic communication would subject them to counter detection. As a result, surfacing to communicate with one another through a receiver such as a satellite could be one method for reducing the reliance on underwater communication.

Another example could be a multi AUV system clearing a building. If one vehicle wants to clear a room, it too must drop out of the formation and may be subjected to intermittent communication with the formation.

In a UUV application, the objective is to stay on target as long as possible while conducting the specified mission. However, there is a need to surface and communicate in order to receive information from other agents. As a result, the vehicles will not be able to communicate every time instance. Prior work has shown that if we assume each vehicle communicates at every time step, the vehicles converge to a desired formation of positions as shown in Figure 1. However, in reality, each vehicle will spend time underwater without

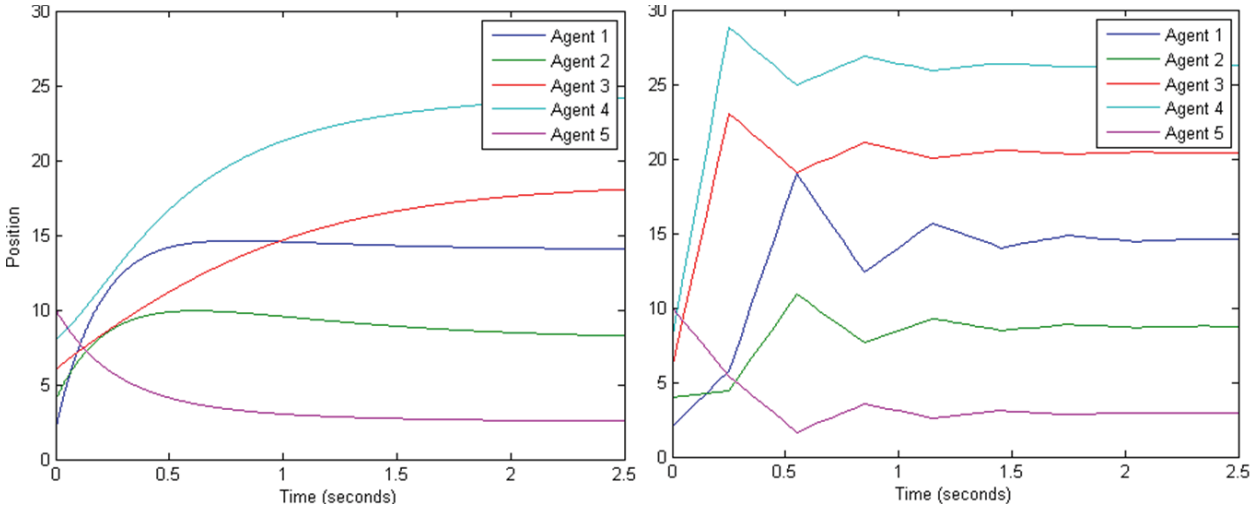

Figure 1: Convergence to desire positions assuming the vehicles communicate in accordance with defined communication topology every time step

communication. As a result, communication occurs at intervals of τ, which affects the rate

of formation convergence.

Figure 1 illustrates a control update delay of $\tau = 0.3$ seconds. Notice, the system takes slightly longer to converge to the desired configuration and reached the formation in a jagged fashion. The trajectories of the vehicle show the effects updating the control law every $\tau = 0.3$ seconds instance. Since the vehicles have no new information for 0.3 time instances, they are forced to rely on outdated information during that time, thus explaining the linear motion between update instances. When they communicate again, they are able to update their control law and adjust their motion. However, they will maintain the same linear motion until they are able to communicate again. If the communication intervals become too large, the system will become unstable and the desired configuration will not be achieved.

If the goal is to keep the agents in a prescribed configuration as long as possible, then the communication intervals of the vehicles should vary to ensure that some of them are always conducting the mission when others drop out to communicate. As a result, we investigate the behavior for a group of vehicles who communicate with individual surfacing intervals, meaning $\tau_1 \neq \tau_2 \neq \tau_n$. In this case, each vehicle gains information from its neighbors from the time when they last surfaced. Figure 2 exemplifies this scenario for the case of 3 UUVs conducting a sampling mission. Moving from left to right, the vehicles have surfacing intervals of $\tau = 3$ seconds, $\tau = 4$ seconds, and $\tau = 5$ seconds respectively. The diagram depicts the knowledge of each vehicle at time step 9. It shows the latest information each vehicle has about every other vehicle. For example, in Figure 2 (left), vehicle one knows

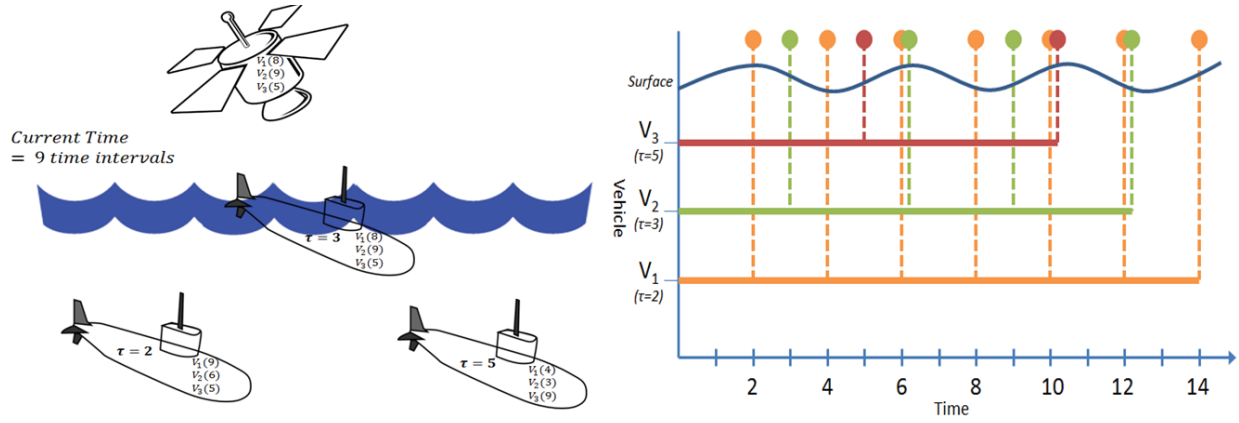

Figure 2: At 9 time units (left), the second UUV is surfaced. All of the UUVs are annotated with the time of their last known information about every other vehicle. The graphical representation (right) shows the effects of individual discrete surfacing intervals on the information available to each agents when it surfaces to update its control.

its own position at time 9, denoted $V_1(9)$, and the position of vehicles two, $V_2(6)$, and three, $V_3(5)$, respectively. Figure 2 (right) shows a time lapse graphical representation of the information available to each vehicle when it surfaces to calculate its control. Note, in between surfacing instances, the vehicles are unaware of any changes to their neighbors and therefor maintain their last computed control for the entire submerged interval. It is evident that vehicle one's information not only depends on its own surfacing intervals, but the surfacing interval of the other vehicles. When vehicle one surfaces, it only receives the information from the other vehicles the last time they surfaced. If the vehicle surfaces in the

shortest intervals, then it is relatively simple. However, in the case like vehicle three, the known vehicle information is derived as follows:

$$t_{surf,n} = mod(t, \tau_n) \tag{1}$$

$$t_{known,n,i} = floor(t_{surf,n}/\tau_n)\tau_n \tag{2}$$

Once an equation defining the information each vehicle knows is derived, the goal is to derive an algorithm that allows each vehicle to coordinate its interval time to maintain the formation.

2 MATHEMATICAL BACKGROUND

This section provides a brief overview of the mathematical concepts that are utilized to derive the results of this paper. Section 2.1 discusses basic concepts and terminology from graph theory that are used to model a multi-agent communication network. Section 2.2 introduces lyapunov based control which will be used to guarantee convergence of the multi-vehicle system.

2.1 Graphical Representation of Agent Communication

Consider a multi-vehicle system consisting of N agents with the goal being to steer each vehicle to a position in a desired configuration. Each agent is represented as a vertex such that the group is given by the set of vertices, ν. A communication channel between i and j is represented as an edge between vertice ν_i and ν_j. The set of edges representing all agent communication is defined as ϵ. The communication topology is defined by the graph, $G = (\nu, \epsilon)$ [3]. For example, the graph in Figure 3 illustrates communication between $N = 5$ agents. Agent one communicates with agents four and five as represented by the bi-directional arrows representing edges $\epsilon_{1,4}$ and $\epsilon_{1,5}$. Agent one is informed of agent three's behavior via information gathered through agent four's behavior.

The graph, G, admits a matrix representation known as the graph Laplacian, L. To compose L, we define the degree matrix, D, a diagonal matrix whose i^{th} diagonal element represents the number of edges connecting vehicle i. $D(i, j)$ is equal to the number of junctions at node i when $i = j$. The Adjacency matrix, A, is a matrix representation of the set of edges ϵ. The element $A(i, j) = 1$ if (n_i, n_j) is a set of ϵ and zero otherwise. The degree and adjacency matricies for the digraph in Figure 3, are shown below in Equations (3) and (4)

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} \tag{3}$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{4}$$

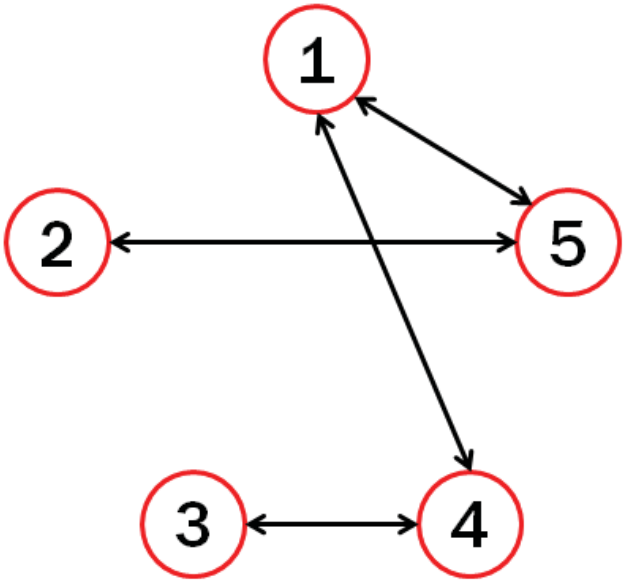

Figure 3: The digraph is one example of a possible digraph for $N = 5$ agents. Only the agents directly connected with an arrow are able to communicate with one another. Other agents must rely on indirect communication to determine their final positions.

The graph Laplacian is defined

$$L = D - A, \tag{5}$$

and can be used to describe the communication between the vehicles [3]. The graph laplacian for the graph in Figure 3 is found from equations (1)-(3) as

$$L = \begin{bmatrix} 2 & 0 & 0 & -1 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 \\ -1 & 0 & -1 & 2 & 0 \\ -1 & -1 & 0 & 0 & 2 \end{bmatrix}. \tag{6}$$

The graph laplacian has a few important properties that are utilized in future analysis. First, the second smallest eigenvalue, λ_2 approximates the time required to achieve the desired configuration from the random initial conditions [4]. This value plays a significant role in further work where communication assumptions are relaxed to determine the effects of the system behavior. Second, $L\mathbf{x} = 0$ if $\mathbf{x} = [1, 1, ..., 1]^T$. This property becomes significant when lyapunov based control is implemented since a potential function based on this property is minimum when all of the agents have the same position. Third, previous work has shown that in order to guarantee convergence, the graph, G, must be connected [3]. A connected graph is one in which there exists a path from node ν_i to ν_j for any (i, j). The path does not have to be a direct link between the two nodes since information can be shared throughout the system. A connected graph is essential because it guarantees that information can travel between any two agents.

2.2 Lyapunov Control

In order to achieve the desired configuration, this research leverages previous work from Lyapunov-based control, most notably the results in references [5], [6], [7], [8], [9], [10], [11], [12], [13], and [14].

The Lyapunov-based control approach seeks to derive a control that minimizes a potential function derived based on the desired agent formation. Define the vector of agent positions $\mathbf{x} = [x_1, x_2, ..., x_N]^T$ and consider a potential function defined such that

$$V(\mathbf{x}) \geq 0 \quad \forall \mathbf{x}, \tag{7}$$

and $V(\mathbf{x}_f) = 0$ for the desired agent configuration vector $\mathbf{x}_f$. The goal is to design a control such that

$$\dot{V}(\mathbf{x}) < 0 \quad \forall \mathbf{x} \neq \mathbf{x}_f. \tag{8}$$

Equation 7 guarantees that the potential function representing the current vehicle formation, V, is always positive, while Equation 8 guarantees that the potential will be decreasing for all possible formations, $\mathbf{x}$, except the desired formation, $\mathbf{x}_f$ [15]. Combining these two principles guarantees that the potential function eventually reaches its minimum where $V = 0$, corresponding to the desired final configuration. The concept of potential function-based control is illustrated in Figure 4. The state $\mathbf{x}_f = [x_1, x_2]^T$ represents the current configuration

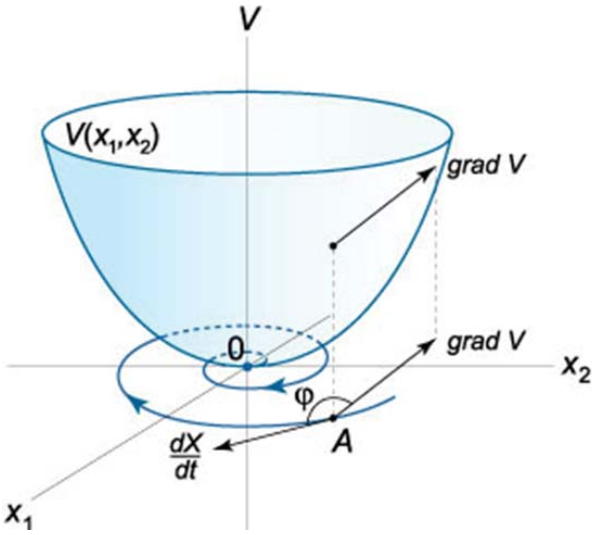

Figure 4: The diagram represents Lyapunov Based Control where the goal is to drive the potential, V, to a minimum (the bottom of the bowl).

of the two-vehicle system, while the bowl represents the potential function over all possible configurations, with the minimum being the desired configuration, $\mathbf{x} = [0, 0]^T$. If the gradient of the state motion is always negative, meaning it is moving downward in the bowl, it will eventually arrive at the bottom, or minimum, of the bowl. At that point, any motion of the particle will have a positive gradient, thus violating Equation (8). Therefore, at that point, the particle has reached the minimum corresponding to the desired final configuration, called the steady-state of the system.

In its simplest form, the potential function for multi-vehicle control applications, V, is defined as

$$V(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T L\mathbf{x}, \tag{9}$$

where L is the graph laplacian. Recalling the property of the laplacian, $L\mathbf{x} = 0$, if $\mathbf{x} = [1, 1, ..., 1]$, we see that $V(\mathbf{x})$ is zero only when all of the vehicles have reached consensus, meaning all agents lie at the same position. In section 3, we introduce a transformation of the graph laplacian, L, that allows us to steer the vehicles to any desired configuration instead of only allowing for vehicle consensus to a common position.

3 GRAPH TRANSFORMATION FOR MULTI-VEHICLE FORMATION CONTROL

Currently, the potential function from Equation (9) ensures the vehicles steer to consensus based on the initial conditions. In order to achieve the overall goal of steering vehicles to a desired formation, we transform the the graph laplacian, L.

As a result, let $\mathbf{c}$ be the desired configuration for the vehicles. The goal is to design a transformed laplacian, P, such that $P\mathbf{c} = 0$. In this case, if the vehicle positions are equal to the desired final position, $\mathbf{c}$ or some multiple, then the potential will be at its minimum indicating that all of the vehicles are in their desired positions. The derivation of the transformed graph laplacian is as follows, using a similarity transformation between matrices:

$$\mathbf{c} = \frac{\mathbf{x_f}}{|\mathbf{x_f}|} \tag{10}$$

$$\mathbf{R} = \text{diag}(c_1, c_2, ...c_n) \tag{11}$$

$$P = RLR^{-1}. \tag{12}$$

In a 1-dimensional, numerical example where $N = 3$ vehicles and the desired linear configuration of $\mathbf{c} = [2, 3, 6]$, R is calculated in accordance with the following equations.

$$R = \begin{bmatrix} \frac{2}{7} & 0 & 0 \\ 0 & \frac{3}{7} & 0 \\ 0 & 0 & \frac{6}{7} \end{bmatrix} \tag{13}$$

$$L = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \tag{14}$$

$$P = \begin{bmatrix} 2 & \frac{-2}{3} & \frac{-1}{3} \\ \frac{-3}{2} & 2 & \frac{-1}{2} \\ -3 & -2 & 2 \end{bmatrix} \tag{15}$$

Note that the transformed laplacian does not guarantee convergence to the desired configuration. Rather it will steer the vehicles to some multiple of the configuration. In order

to account for this change and ensure the vehicles reach the exact desired formation, a term is added to the potential which takes on the form

$$V = \frac{1}{2}\mathbf{x}^T RLR^{-1}\mathbf{x} + \frac{1}{2}\sum_{k=1}^{N} a_0(x_k - x_{f,k})^2 \tag{16}$$

The first term on the right hand side in Equation (16) is the relative position term which defines the relative position between the two agents. After the similarity transform shown in Equation (15), the P matrix weights the graph lapliacian, L, according to the desired final positions. Additionally, the similarity transform shown in Equation (12) is used to ensure the new matrix, P, has the same eigenvalues as L. While the graph laplacian, L, does not affect the desired shape of the configuration, it does effect the robustness of convergence. If you calculate the eigenvalues of the graph laplacian, you will have one 0 eigenvalue since L is not linearly independent. The second smallest eigenvalue, which is the first non-zero value, is proportional to the speed of response to a step disturbance [4]. Specifically, the larger λ_2, the quicker the convergence of the system. Additionally, L and P share the same eigenvalues. In accordance with Equation (7), V is positive semi-definite.

Multiplying x and x^T in Equation (7), ensures $V \geq 0 \forall x$. Further, the transformation ensures that the magnitude of V is not affected by the square. In addition, a factor of $\frac{1}{2}$ is introduced to account for the factor of 2 appearing when a derivative is taken.

The second term in Equation (16) is the absolute position term which defines each agents absolute position in space. This term allows to ensure the vehicles converge to the desired formation as opposed to some scaled multiple. The term $a_{0,k}$ is a vector of length N which is assigned based on the following [14]:

$$a_{0,k} = \begin{cases} 0, & \text{if } k_{th} \text{ agent is unaware of its desired position} \\ 1, & \text{if } k_{th} \text{ agent knows its desired final position.} \end{cases} \tag{17}$$

In the case where $a_{0,k} = 0$, the k_{th} vehicle must rely on communicated information from the other agents to determine its control. This work has shown that as long as

$$||a_{0,k}|| \neq 0 \text{foratleastonevehicle} \tag{18}$$

the system will converge since other vehicles can rely on inter-vehicle communication to determine their appropriate relative position [14]. Additionally, the positions error, $x - x_f$, is squared in the absolute position term to guarantee that V is always positive ensuring that Equation 16 holds true.

Once the potential function, V is defined, the focus then shifts to developing the control to ensure the vehicles convergence in accordance with V.

4 THEORETICAL RESULTS

This section presents solutions to achieve our three contributions. Inter-vehicle communication must be defined in order to develop a steering control algorithm for multi-vehicle configurations subject to intermittent communication. Once complete, we are able to derive

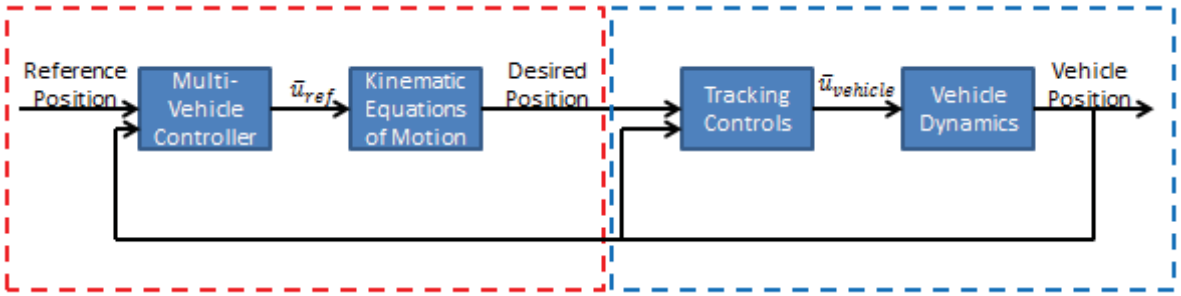

Figure 5: The red box outlines the kinematic controller, while the blue box outlines the tracking controller. The blue box encapsulates the more realistic steering dynamics that are used to track the kinematic reference trajectories.

a kinematic steering control. This project then derives more robust and realistic controls such as trajectory and target tracking to build upon the kinematic controller.

Once the control is established, the communication assumptions are relaxed through the implementation of self-triggering control to allow agents to calculate their next communication instance. This section addressed the various facets of the final implementation through a methodical derivation of the final control.

4.1 Control Approach

Prior to considering the communication constraints of the multi-vehicle system, we first addressed the control aspect. Specifically, how can a multi-agent system be steered to any number of configurations with more complex dynamics? A two component approach with reference and actual vehicles, governed by simple kinematics and high fidelity vehicle dynamics respectively. This approach is easier to implement higher fidelity dynamics since we can derive the complex trajectory motion utilizing a kinematic controller.

Figure 5 illustrates a block diagram of the control structure for a single vehicle. The block diagram demonstrates the basic control structure for our algorithm, which we built incrementally as we increased the complexity and capability of the simulations. At a high level, the controller works by generating desired trajectories and steering the vehicles to track the desired trajectories. The reference trajectories are governed by kinematic equations of motion allowing for free range of motion, while the actual vehicle dynamics are governed by the equations of motion of a self-propelled steering vehicle. For perspective, think of kinematic dynamics as a hockey puck since the x and y velocities can be directly controlled, while the particle dynamics would be comparable to a bicycle which must steer to change direction.

Simply put, we design the reference trajectories with a simple kinematic controller that can be easily applied to any desired trajectory. The actual vehicle dynamics are used to chase the reference trajectories bounded by their steering limitations.

The diagram changes slightly when we shift to incorporate the effects of the self-triggering control and communication assumption. The new block diagram for our desired final system is shown in Figure 6, which differed from Figure 5 in that it includes the self-triggering control as a part of the kinematic controller. The self-triggering control is built into the

kinematic control and is used to govern the behavior of the reference trajectories.

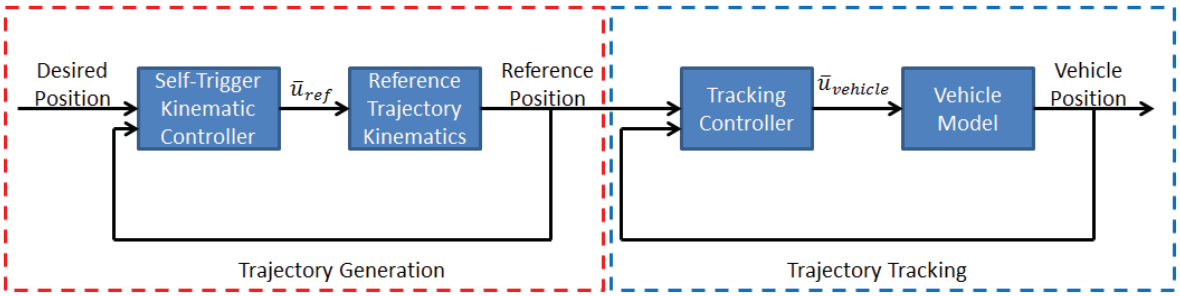

Figure 6: Although the broad concept of tracking control still applies when the communication constraints are added, the overall control diagram must be amended to account for the self-triggering control implementation.

In the new control set-up, the self-triggering control is effectively integrated into the kinematic control block. Essentially, the reference trajectories are governed by the kinematic controller and the self-triggering control, while the actual vehicles, governed by the vehicle dynamics are unchanged and continue to chase after the reference trajectories.

Note that surfacing, control update, and communication are synonymous in this paper. Each time an agent surfaces, not only do we assume it is instantaneous, but also, we assume that the agents communicate with an information router and calculate their new control.

4.2 Kinematic Control for Reference Trajectory Generation

In order to implement tracking control, we must first define the dynamics of the actual and reference trajectories. Kinematic control is the most basic. In two dimensions, the kinematic equations of motion are

$$\dot{x} = u_1 \tag{19}$$

$$\dot{y} = u_2. \tag{20}$$

In Equations (19) and (20), we have direct control over both the x and y velocity. Using a control derived from the potential function (16) results in

$$u_k = -KP_x x_k - Ka_{0,k}(x_k - xf). \tag{21}$$

Figure 7 shows the functionality of kinematic control by simulating a set of N one dimensional agents without loss of generality. The control in Equation (21), $\mathbf{u}$, is designed in order to guarantee the agents reach the desired locations specified by the dashed lines. By designing the controller in accordance with the requirements, it is possible to guarantee convergence to the final desired positions.

Figure 7 shows the convergence of a five agent system using kinematic control. By applying lyapunov based control, the vehicles move to their preassigned desired final positions. The results provide the foundation for future work with trajectory tracking to simulate more realistic vehicle dynamics.

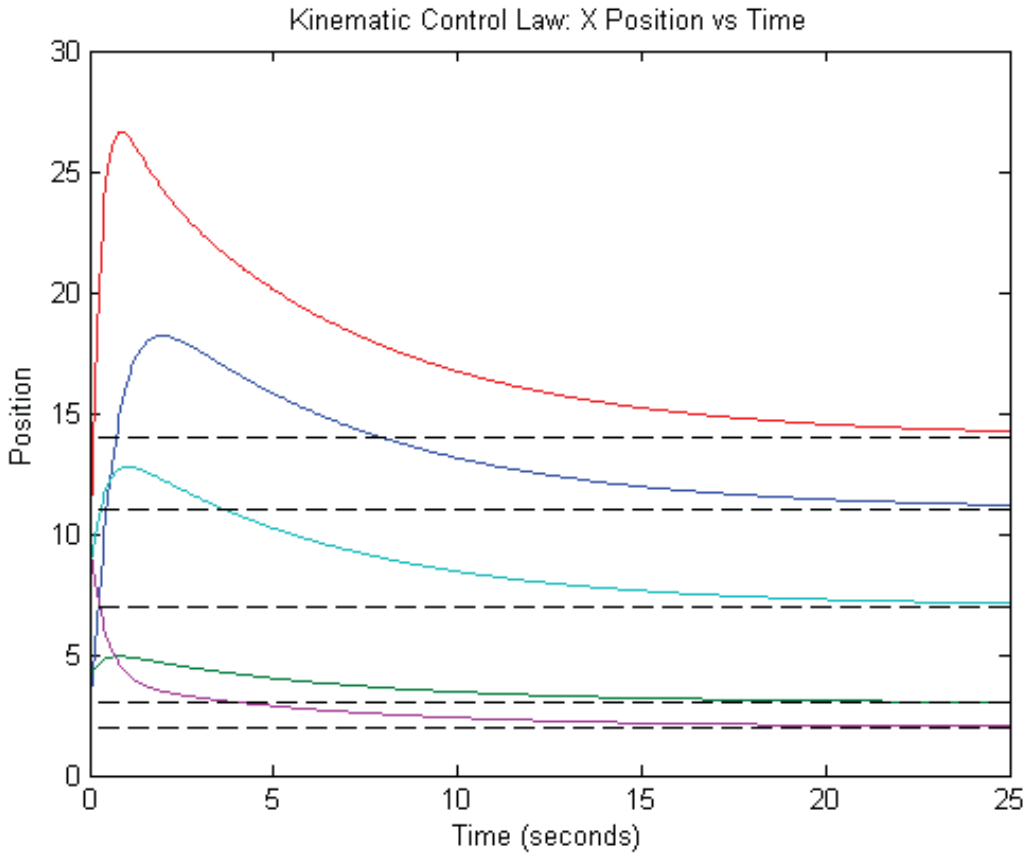

Figure 7: The plot show the position versus time of 5 agents using kinematic control. Their initial positions are $x_0 = [2, 4, 6, 8, 10]$ and final positions are $x_f = [1, 2, 3, 4, 5]$. The gain is 0.5.

Kinematic control is excellent due to its simplicity and ease of design; however, it does not realistically represent vehicle dynamics. The position plot of the vehicles shows that all of the vehicles reach their desired final position, thus validating the use of lyapunov-based control. However, this steering method is crude and assumes a particle is able to move about in any orientation, hence the direct control of both the x and y velocities. As a result, a control law that more closely represents actual vehicle dynamics is desired. Referring to the previous analogy, vehicles that behave like bicycles are preferred to those that behave like hockey pucks since most vehicles use steering to change their position as opposed to directly adjusting their velocity. Consequently, a more complex controller capable of more closely mimicking a true system is desired.

We desire higher fidelity dynamics to more closely mimic the behavior of actual vehicles. However, higher fidelity dynamics make derivation of coordinated control algorithms difficult and requires new derivations for each vehicle model. To combat this issue, we can utilize a trajectory tracking approach in which the reference vehicles are defined by the kinematic control and the actual vehicles are governed by steering dynamics. The actual vehicles then 'chase' the reference trajectories. As a result, this work leverages the work from Ren and

Beard to implement more complex dynamics [9]. For the actual vehicles, we use the self propelled particle dynamics

$$\begin{aligned} \dot{x} &= s\cos\theta \\ \dot{y} &= s\sin\theta \\ \dot{\theta} &= u_1 \\ s &= u_2 \end{aligned} \tag{22}$$

where θ is the vehicle orientation, s is vehicle speed, $\mathbf{u} = [u_1, u_2]$ are the turn rate and speed controls, respectively, and x and y are the x and y positions, respectively.

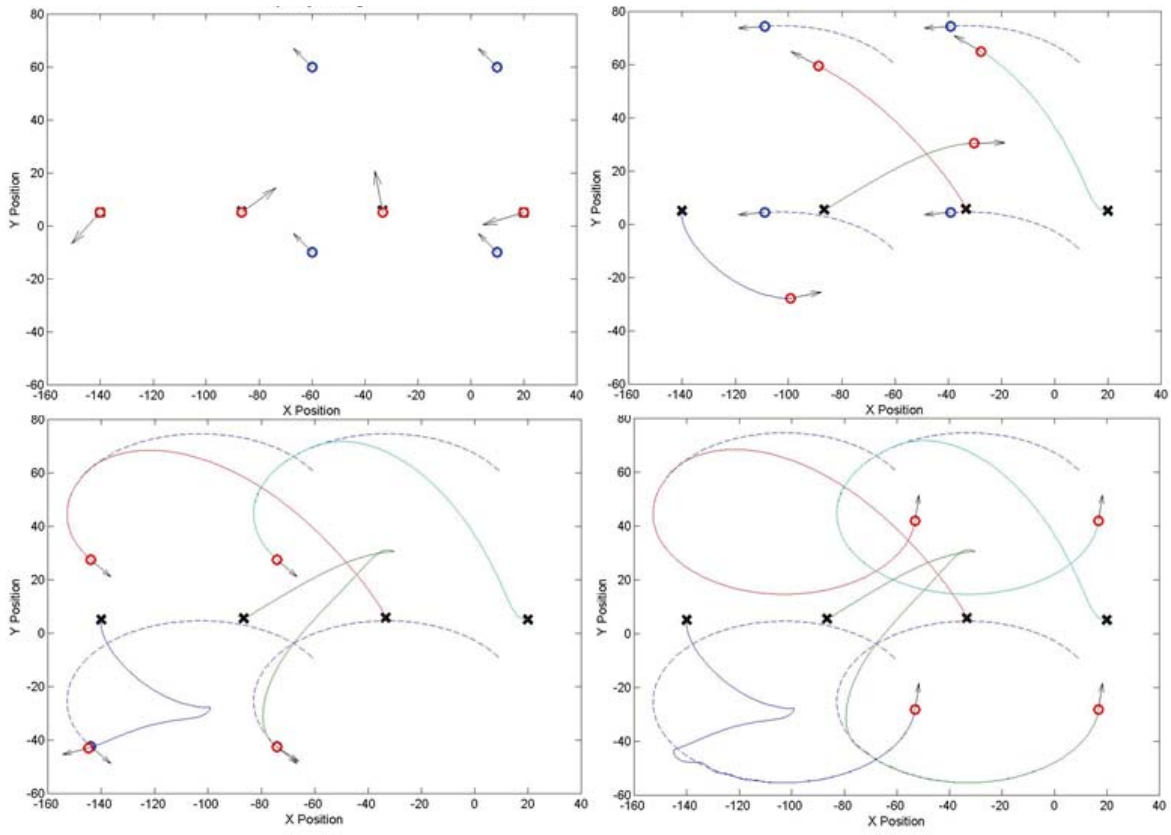

Figure 8: The plot shows the interaction between the reference trajectories (red) and actual vehicles (blue) as derived from Ren and Beard [9]. The actual vehicles chase, catch, and then follow the oval pattern of the reference trajectories.

Figure 8 demonstrates the trajectory tracking control as the actual vehicles (blue circles) chase and catch the reference trajectories (red circles).

The functional block diagram in Figure 5 represents the structure of our final control design. It features a combination of kinematic control, contained within the red dashed box, and self-propelled particle dynamics, contained within the blue dashed box, to drive the vehicles to their desired final positions. More specifically, this work draws on the simplicity of kinematic control which allows us to define a multitude of shapes. For example, its

is possible to define any parametric equation such as circles, ovals, sine waves, etc. with relative simplicity.

Self-propelled particle dynamics are used to act as our actual vehicles. Between the two separate controllers, this work creates a tracking control in which the actual vehicles follow the reference trajectories defined by the kinematic controller. The advantage is that we can define a large number of trajectories without significantly changing the control law, and the actual vehicles will follow and catch the reference trajectories with their own steering and speed limits imposed.

Take for example, Figure 8. In this simulation, tracking control with the reference trajectories denoted by blue circles and the actual vehicle denoted by red circles is implemented. As apparent in the figure, the actual vehicles must catch the reference trajectories for their respective path. Once they catch it, they are able to continue following the reference trajectory which serves as the desired configuration. To achieve the desired result, the control derived in Ren and Beard is implemented by including reference and actual vehicles. The reference vehicles are subject to particle dynamics and track the desired position without a steering constraint. Note, the reference vehicles are simply phantom trajectories that are used as targets for the actual vehicles.

From Figure 8, it is evident that the reference vehicles (blue) traveled in an ellipse, and the actual vehicles (red) followed them before eventually meeting. In the fourth panel in Figure 8, note that the actual vehicles track the reference vehicles. The strength of our controller is evident in Figures 8 and 9. Figure 8 illustrates how the kinematic equations follow the exact desired shape. However, the sharp turns are not possible for the particle dynamics to track with the given steering constant. As a result, the actual vehicles track the reference vehicles as best they can, but are not always identical. The robustness of this control is shown by implementing a target tracking scenario. In the simulation in Figure 9, the same combination of dynamics is implemented. Instead the control is modified to have the vehicle follow the target vehicle, denoted by a blue x.

4.3 Self-Triggered Control for Consensus

While all-to-all communication is both desirable and ideal, many applications require either directed topologies or time-delayed communication between agents. Self and event-triggered control approaches have proven successful in addressing the demand for robust control despite non instantaneous vehicle communication in [16], [17], [18], [19], [20], [21], and [22]. However, these application present limiting assumptions that do not fit every possible application.

While many applications are possible, framing the communication problem in the context of an undersea surveillance mission provides a clear understanding of the advantages and results, where triggering instances represent surfacing of the unmanned underwater vehicles (UUVs) including communication with a central information router such as a satellite. Our goal is to maximize the UUV time on target while maintaining a time-bounded formation error. In our algorithm, instead of each UUV surfacing when one triggers to surface, only the triggered vehicle will surface. The remaining UUVs will continue conducting the mission underwater. The surfacing UUV will share its current information with the satellite and compute its control based on the last known position, velocity, and surfacing time of its neighboring UUVs. We assume that the surfacing of the vehicle is instantaneous relative to

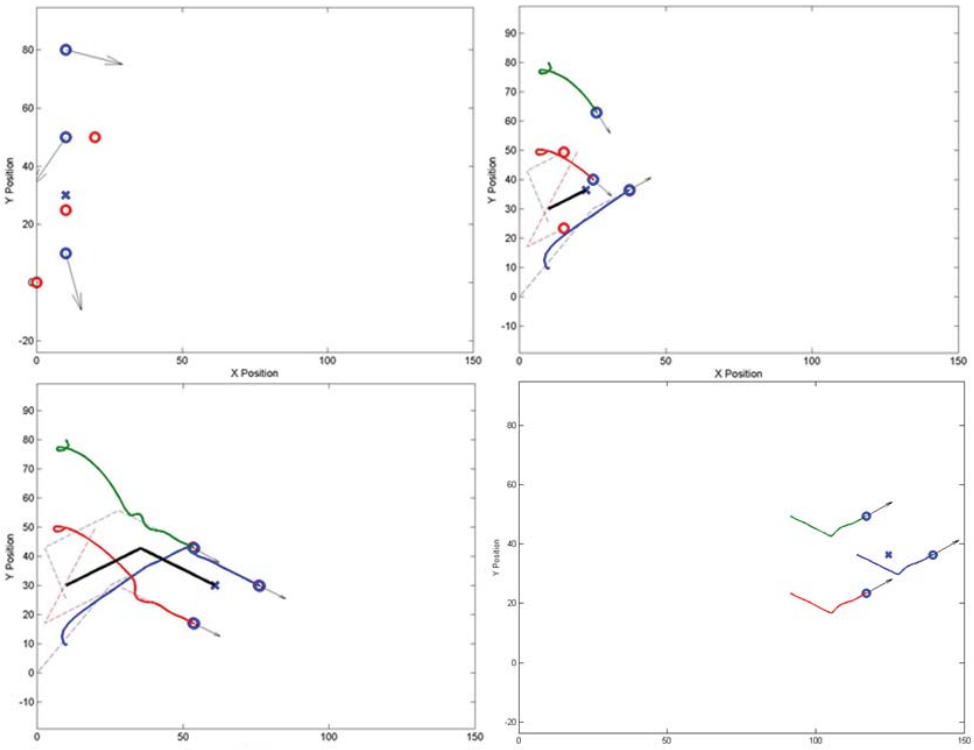

Figure 9: Four instances in time from the target tracking simulation are shown. It is evident that the actual vehicles are constrained in their motion as they steer and change velocities in order to catch the reference vehicles moving in an oval.

the time scales associated with the problem.

The time delayed communication aspect is addressed by applying theories from event and self-triggered control from [1] and [2]. However, the focus of our control law presented is for a multi-agent systems where the agents compute their individual control law based on the last known information of the other agents. The distributed self-triggered control strategy is used to calculate the next communication execution time for each individual vehicle.

In the control presented in [1], every time a vehicle triggers, all other vehicles communicate as well. While this is ideal as there is no distributed time delay amongst the vehicles and the control laws for each can be computed based on current information, it is not practical in our application as it would require all neighbors to surface with the triggered vehicle. As a result, our algorithm excludes the need for all of the vehicles to communicate at each triggering instance. Instead only the triggering vehicle recomputes its control at its respective triggering instance. In theory, this will reduce the overall recomputing instances.

The distributed-self triggering algorithm leverages previous work including [1] and [2]. In our case, instead every agent communicating when one vehicle triggers, we created a central information hub that allows individual vehicles to compute their control. The central information hub contains the last known position, velocity, and update time of all other vehicles, which is used to calculate the triggering vehicles information. Essentially, instead

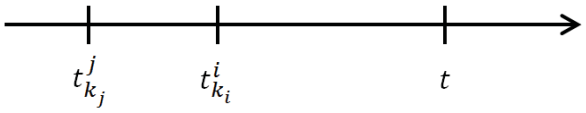

Figure 10: The time line indicates the different surfacing instances of agents i and j. This representation allows for a visual understanding of how the control is derived and how the integration of $\dot{x}$ is derived.

of having all current information at each triggering time, this algorithm relies on dated information from the other agents to calculate its control over its next interval.

The control for agent i is based on the position error between vehicle i and its neighboring agents. Since consensus is desired, the control law is the sum of the errors between an agent and its communicating neighbors. This error is zero when agent i and its communicating neighbors all share the same position. As a result, the control law for agent i is

$$u_i(t) = -K \sum_{j \in N_i} (x_i(t^i_{k_i}) - x_j(t^j_{k_j})) = -KL_i\mathbf{x}, \tag{23}$$

where L_i is the i^{th} row of the laplacian, L, and $x_j(t^j_{k_j})$ is the last known event time of x_j. The control for i is updated only at its event times $t^i_{0_i}, t^i_{1_i}, ..., t^i_{\infty_i}$. The control for j is updated only at its event times $t^j_{0_j}, t^j_{1_j}, ..., t^j_{\infty_j}$. From Figure 10, if agent i chooses to update its control at time t, it may only use information available at that time. Since agent i has not communicated since t_i, the information passed at t_j is unknown to agent i. As a result, this information must be excluded from the control. Hence, $x_j(t_{ki})$ is used for the position of agent j since i was last aware of j's position at time t_i. This clearly illustrates the time delayed nature of the control. As a result of the known information, the control is updated to be

$$u_i(t) = -\sum_{j \in N_i} (x_i(t^i_{k_i}) - x_j(t^i_{k_i})).$$

However, when i surfaces, it receives information about j timestamped at j's last surface time, $t^j_{k_j}$. So if j shares its last known position and computed velocity then x_j's position can be computed using the integral form

$$x_j(t^i_{k_i}) = x_j(t^j_{k_j}) + \dot{x}_j(t^j_{k_j})(t^i_{k_i} - t^j_{k_j}), \tag{24}$$

so substituting for $x_j(t^i_{k_i})$ the control becomes

$$u_i(t) = -K \sum_{j \in N_i} x_i(t^i_{k_i}) - [x_j(t^j_{k_j}) + \dot{x}_j(t^j_{k_j})(t^i_{k_i} - t^j_{k_j})],$$

for $t \in [t^i_{k_i}, t_{k+1}{}^i_i)$. Since $u_i(t) = \dot{x}_i(t)$, the kinematics of agent i are

$$\dot{x}_i(t) = -\sum_{j \in N_i} (x_i(t^i_{k_i}) - x_j(t^i_{k_i})). \tag{25}$$

Define the position error of each agent as the difference between the last known position (i.e. the position of its last trigger/control update time) and the position at the current time. Thus

$$e_i(t) = x_i(t^i_{k_i}) - x_i(t), \qquad \text{and} \qquad e_j(t) = x_j(t^j_{k_j}) - x_j(t).$$

Then rearranging both equations gives

$$x_i(t^i_{k_i}) = x_i(t) + e_i(t), \qquad \text{and} \qquad x_j(t^j_{k_j}) = x_j(t) + e_j(t).$$

Also, from (24) we know $x_j(t^i_{k_i}) = x_j(t^j_{k_j}) + \dot{x}_j(t^j_{k_j})(t^i_{k_i} - t^j_{k_j})$. Since $x_j(t^j_{k_j})$ is unknown, we substitute the error which gives

$$x_j(t^i_{k_i}) = x_j(t) + e_j(t) + \dot{x}_j(t^j_{k_j})(t^i_{k_i} - t^j_{k_j}). \tag{26}$$

Substituting (24) and (26) into (25) gives the updated control

$$\dot{x}_i(t) = -\sum_{j\in N_i}[x_i(t) - x_j(t) + e_i(t) - e_j(t) - \dot{x}_j(t^j_{k_j})(t^i_{k_i} - t^j_{k_j})]. \tag{27}$$

In order to eventually write in vector form, we break the kinematics down into three terms by rewriting the kinematics as

$$\dot{x}_i(t) = -\sum_{j\in N_i}(x_i(t) - x_j(t)) - \sum_{j\in N_i}(e_i(t) - e_j(t)) + \sum_{j\in N_i}\dot{x}_j(t^j_{k_j})(t^i_{k_i} - t^j_{k_j}).$$

Letting $\gamma_j = \sum_{j\in N_i}\dot{x}_j(t^j_{k_j})(t^i_{k_i} - t^j_{k_j})$, then

$$\dot{\bar{x}} = -L\bar{x} - L\bar{e} + \bar{\gamma}.$$

Switching focus, we define the lyapunov potential as

$$V = \frac{1}{2}\mathbf{x}^T L\mathbf{x},$$

and let $\mathbf{z} = L\mathbf{x}$. Then, taking the derivative of the potential and substituting z yields

$$\dot{V} = \mathbf{x}^T L\dot{\mathbf{x}} = \mathbf{x}^T L(-L\mathbf{x} - L\mathbf{e} + \gamma) = -\mathbf{x}^T LL\mathbf{x} - \mathbf{x}^T LL\mathbf{e} + \mathbf{x}^T L\gamma = -\mathbf{z}^T\mathbf{z} - \mathbf{z}^T L\mathbf{e} + \mathbf{z}^T\gamma. \tag{28}$$

Expressing the potential as a summation we have

$$\begin{aligned}\dot{V} &= -\sum_i z_i^2 - \sum_i\sum_{j\in N_i} z_i(e_i - e_j) + \sum_i\sum_{j\in N_i} z_i\dot{x}_j(t^j_{k_j})(t^i_{k_i} - t^j_{k_j}) \\ &= -\sum_i z_i^2 - \sum_i |N_i| z_i e_i + \sum_i\sum_{j\in N_i} z_i e_j + \sum_i\sum_{j\in N_i} z_i\dot{x}_j(t^j_{k_j})(t^i_{k_i} - t^j_{k_j})\end{aligned}$$

Using the inequality $\mathbf{xy} \leq \frac{a}{2}x^2 + \frac{1}{2a}y^2$ for $a > 0$ on $z_i e_i$ and $z_j e_j$, we use algebra to rearrange the potential as shown, noting that a property of summations allows $|N_i|$, the neighbors of i, to replace the summation if no j terms are present. We have

$$\dot{V} \leq -\sum_i z_i^2 + \sum_i |N_i|(\frac{a}{2}z_i^2 + \frac{1}{2a}e_i^2) + \sum_i \sum_{j \in N_i} (\frac{a}{2}z_i^2 + \frac{1}{2a}e_j^2) + \sum_i \sum_{j \in N_i} z_i \dot{x}_j(t_{k_j}^j)(t_{k_i}^i - t_{k_j}^j),$$

which yields

$$\begin{aligned} \dot{V} \quad \leq \quad & -\sum_i z_i^2 + \sum_i |N_i| \frac{a}{2} z_i^2 + \sum_i |N_i| \frac{1}{2a} e_i^2 + \sum_i |N_i| \frac{a}{2} z_i^2 \\ & + \sum_i \sum_{j \in N_i} \frac{1}{2a} e_j^2 + \sum_i \sum_{j \in N_i} z_i \dot{x}_j(t_{k_j}^j)(t_{k_i}^i - t_{k_j}^j) \end{aligned}$$

$$\dot{V} \leq -\sum_i z_i^2 + \sum_i |N_i| a z_i^2 + \sum_i |N_i| \frac{1}{2a} e_i^2 + \sum_i \sum_{j \in N_i} \frac{1}{2a} e_j^2 + \sum_i \sum_{j \in N_i} z_i \dot{x}_j(t_{k_j}^j)(t_{k_i}^i - t_{k_j}^j) \quad (29)$$

Since the graph is symmetric, meaning for some matrix M, $M = M^T$, we can interchange indices to get the e_j term in terms of i allowing for future simplification of the control

$$\sum_i \sum_{j \in N_i} \frac{1}{2a} e_j^2 = \sum_i \sum_{j \in N_i} \frac{1}{2a} e_i^2 = \sum_i \frac{1}{2a} |N_i| e_i^2. \quad (30)$$

Substituting (30) into (29) gives

$$\dot{V} \leq -\sum_i z_i^2 + \sum_i |N_i| a z_i^2 + \sum_i |N_i| \frac{1}{2a} e_i^2 + \sum_i |N_i| \frac{1}{2a} e_i^2 + \sum_i \sum_{j \in N_i} z_i \dot{x}_j(t_{k_j}^j)(t_{k_i}^i - t_{k_j}^j$$

$$\dot{V} \leq -\sum_i z_i^2 + \sum_i |N_i| a z_i^2 + \sum_i \frac{|N_i|}{a} e_i^2 + \sum_i \sum_{j \in N_i} z_i \dot{x}_j(t_{k_j}^j)(t_{k_i}^i - t_{k_j}^j)$$

$$\dot{V} \leq -\sum_i (1 - |N_i| a) z_i^2 + \sum_i \frac{|N_i|}{a} e_i^2 + \sum_i \sum_{j \in N_i} z_i \dot{x}_j(t_{k_j}^j)(t_{k_i}^i - t_{k_j}^j)$$

where $\gamma_j = \dot{x}_j(t_{k_j}^j)(t_{k_i}^i - t_{k_j}^j)$. Invoking $\mathbf{xy} < \frac{a}{2}x^2 + \frac{1}{2a}y^2$ on the term $z_i \gamma_j$ we have

$$\dot{V} \leq -\sum_i (1 - |N_i| a) z_i^2 + \sum_i \frac{|N_i|}{a} e_i^2 + \sum_i \sum_{j \in N_i} (\frac{a}{2} z_i^2 + \frac{1}{2a} \gamma_j^2)$$

which simplifies to

$$\dot{V} \leq -\sum_i (1 - \frac{3|N_i|}{2} a) z_i^2 + \sum_i \frac{|N_i|}{a} e_i^2 + \sum_i \sum j \in N_i \frac{1}{2a} \gamma_j^2. \quad (31)$$

Requiring $1 - \frac{3|N_i|}{2} a > 0$, stipulates $a < \frac{2|N_i|}{3}$ for each i. Then $\dot{V} \leq 0$ if

$$-\sum_i (1 - \frac{3|N_i|}{2} a) z_i^2 + \sum_i \frac{|N_i|}{a} e_i^2 + \sum_i \sum_{j \in N_i} \frac{1}{2a} \gamma_j^2 \leq 0$$

$$-\sum_i[(\sum_{j\in N_i}\frac{1}{2a}\gamma_j^2)+\frac{|N_i|}{a}e_i^2-(1-\frac{3|N_i|}{2}a)z_i^2]\leq 0.$$

Rearranging the inequality and solving for e_i we achieve

$$e_i^2\leq\frac{\sigma_i a}{|N_i|}[(1-\frac{3|N_i|}{2}a)z_i^2-\sum_{j\in N_i}\frac{1}{2a}\gamma_j^2]. \tag{32}$$

To simplify the notation, let

$$\beta_i\triangleq\frac{\sigma_i a(1-\frac{3}{2}a|N_i|)}{|N_i|}, \tag{33}$$

which gives

$$e_i^2\leq\beta_i z_i^2-\frac{\sigma_i}{2|N_i|}\sum_{j\in N_i}\frac{1}{2a}\gamma_j^2. \tag{34}$$

Expanding the error allows the condition to be rewritten as

$$|x_i(t_{k_i}^i)-x_i(t)|^2\leq\beta z_i^2-\frac{\sigma_i}{2|N_i|}\sum_{j\in N_i}\frac{1}{2a}\gamma_j^2 \tag{35}$$

Recalling (25), the kinematics are

$$\dot{x}_i(t)=-\sum_{j\in N_i}x_i(t_{k_i}^i)-x_j(t_{k_i}^i).$$

Integrating $\dot{x}_i(t)$ gives

$$x_i(t)=-\sum_{j\in N_i}[x_i(t_{k_i}^i)-x_j(t_{k_i}^i)](t-t_{k_i}^i)+x_i(t_{k_i}^i). \tag{36}$$

which can then be substituted into (35) to give

$$|x_i(t_{k_i}^i)-(-\sum_{j\in N_i}[x_i(t_{k_i}^i)-x_j(t_{k_i}^i)](t-t_{k_i}^i)+x_i(t_{k_i}^i))|^2\leq\beta z_i^2-\frac{\sigma_i}{2|N_i|}\sum_{j\in N_i}\frac{1}{2a}\gamma_j^2.$$

Combining like terms and simplifying gives the left hand side of (37) in terms of agent i variables

$$|\sum_{j\in N_i}[x_i(t_{k_i}^i)-x_j(t_{k_i}^i)](t-t_{k_i}^i)|^2\leq\beta z_i^2-\frac{\sigma_i}{2|N_i|}\sum_{j\in N_i}\frac{1}{2a}\gamma_j^2. \tag{37}$$

Since $x_j(t)$ will follow the same form as $x_i(t)$, we get

$$x_j(t)=-\sum_{l\in N_j}[x_j(t_{k_j}^j)-x_l(t_{k_j}^j)](t-t_{k_j}^j)+x_j(t_{k_j}^j), \tag{38}$$

where l are the neighbors of j as defined by the graph laplacian, L. From (36) and (38), denote now

$$\rho_i=-\sum_{j\in N_i}(x_i(t_{k_i}^i)-x_j(t_{k_i}^i)),$$

and

$$\rho_j = -\sum_{l\in N_j}(x_j(t^j_{k_j}) - x_l(t^j_{k_j})).$$

Also, let the discrete execution times be $\xi_i = t^* - t^i_{k_i}$ where t^* is a vector of candidate future trigger times. Now recalling $\mathbf{z} = L\mathbf{x}$, we know $z_i = L_i x_i$ which is equivalent to

$$z_i(t) = -\sum_{j\in N_i}(x_i(t) - x_j(t)). \tag{39}$$

Substituting (36) and (38) gives

$$\begin{aligned} z_i(t) &= \sum_{j\in N_i}[(-\sum_{j\in N_i}(x_i(t^i_{k_i}) - x_j(t^i_{k_i}))(t - t^i_{k_i}) + x_i(t^i_{k_i})] \\ &\quad -\sum_{j\in N_i}[-\sum_{l\in N_j}(x_j(t^j_{k_j}) - x_l(t^j_{k_j}))(t - t^j_{k_j}) + x_j(t^j_{k_j})]. \end{aligned}$$

Making substitutions for ρ_i, ρ_j,and ξ_i and simplifying gives

$$\begin{aligned} z_i(t) &= \sum_{j\in N_i}(-\rho_i\xi_i + x_i(t^i_{k_i})) - \sum_{j\in N_i}(-\rho_j(t - t^j_{k_j}) + x_j(t^j_{k_j})) \\ &= -|N_i|\rho_i\xi_i + |N_i|x_i(t^i_{k_i}) + \sum_{j\in N_i}(\rho_j((t - t^i_{k_i}) + t^i_{k_i} - t^j_{k_j}) + x_j(t^j_{k_j})) \\ &= -|N_i|\rho_i\xi_i + \sum_{j\in N_i} x_i(t^i_{k_i}) + \sum_{j\in N_i}\rho_j\xi_i + \sum_{j\in N_i}\rho_j\xi_i(t_{k^i_i} - t^j_{k_j}) - \sum_{j\in N_i} x_j(t^j_{k_j}) \end{aligned}$$

$$z_i(t) = [-|N_i|\rho_i + \sum_{j\in N_i}\rho_j]\xi_i + \sum_{j\in N_i}\rho_j(t^i_{k_i} - t^j_{k_j}) + |N_i|x_i(t^i_{k_i}) - \sum_{j\in N_i} x_j(t^j_{k_j}). \tag{40}$$

Let $P_i = -|N_i|\rho_i + \sum_{j\in N_i}\rho_j$ and $\Phi_i = \sum_{j\in N_i}\rho_j(t^i_{k_i} - t^j_{k_j}) + |N_i|x_i(t^i_{k_i}) - \sum_{j\in N_i} x_j(t^j_{k_j})$ in order to facilitate simplification of the triggering expression. Recall from (37), we have

$$|\sum_{j\in N_i}[x_i(t^i_{k_i}) - x_j(t^i_{k_i})](t - t^i_{k_i})|^2 \le \beta z_i^2 - \frac{\sigma_i}{2|N_i|}\sum_{j\in N_i}\frac{1}{2a}\gamma_j^2.$$

Expanding z_i from (40), we have

$$\begin{aligned} |\sum_{j\in N_i}[x_i(t^i_{k_i}) - x_j(t^i_{k_i})](t - t^i_{k_i})|^2 &\le \beta[(-|N_i|\rho_i + \sum_{j\in N_i}\rho_j)\xi_i + \sum_{j\in N_i}\rho_j(t^i_{k_i} - t^j_{k_j}) \\ &\quad +|N_i|x_i(t^i_{k_i}) - \sum_{j\in N_i} x_j(t^j_{k_j})]^2 - \frac{\sigma_i}{2|N_i|}\sum_{j\in N_i}\frac{1}{2a}\gamma_j^2 \end{aligned} \tag{42}$$

Simplifying by substituting ρ_i, ξ_i, P_i, and Φ_i gives the triggering expression

$$|\rho_i\xi_i|^2 \le \beta(P_i\xi_i + \Phi_i)^2 - \frac{\sigma_i}{2|N_i|}\sum_{j\in N_i}\frac{1}{2a}\gamma_j^2. \tag{43}$$

The trigger time, t^* is found by determining the value of ξ_i that establishes equivalence because it ensures $\dot{V} \leq 0$ in (28). The results of the self-trigger control simulation for a N=4 vehicle system is shown below. The control guarantees convergence due to the decreasing potential from lyapunov-based control. Asterisks represent the time in which the control is updated. Figure 11, compares the behavior of the system with the old and new triggering conditions applied, respectively.

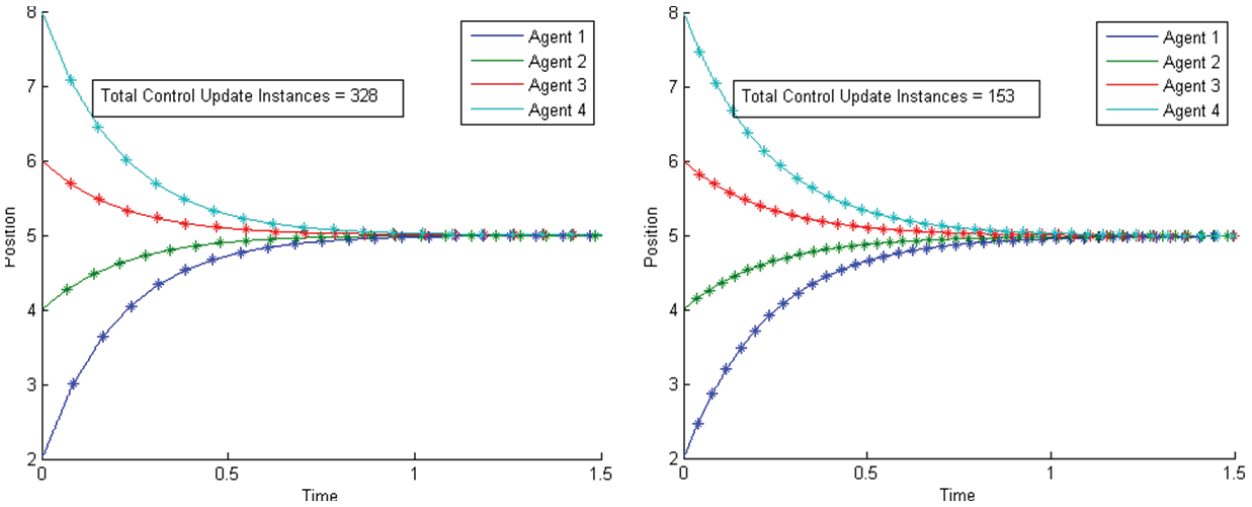

Figure 11: The left figure shows the system behavior with the old implementation requiring all neighbor vehicles to surface at an agent's triggering instance. The right figure shows the implementation of the new condition 43 which allows only one vehicle to surface at its own triggering instance.

It is important to note that there are more individual triggering instances in the new application. However, since the new triggering condition requires only one vehicle to surface at each trigger instance, while the old condition requires all vehicles to surface at each instance, there are less total control updates in the new control.

4.4 Self-Triggered Kinematic Formation Control

The previous derivation works when consensus is desired. However, our end goal desires steering vehicles to formations. As a result, we must incorporate the transformation into our control. Since the transformed laplacian is scaled based on the desired final formation, it is not necessarily a symmetric matrix. Consequently, the assumptions made in the previous proof do not hold and a new derivation is required.

We must derive the transformed graph laplacian to account for the changes in our control law. In order to resolve this issue, we redefined the control, $u_i(t)$, by including the desired final positions to scale the control. While much of the derivation follows from section 4.3, the triggering condition is slightly different. The key steps of the derivation are shown below.

Start by redefining the control law based on the last known event times of the vehicles

including the desired final positions, c, as

$$u_i(t) = -K \sum_{j \in N_i} (c_j x_i(t^i_{k_i}) - c_i x_j(t^j_{k_j})) = -KH\mathbf{x}, \tag{44}$$

where

$$H = \text{diag}(S^* * \mathbf{1} - S) and S = A \circ \mathbf{c1}^T,$$

where $x_j(t^j_{k_j})$ is the last known event time of x_j. The control for i is updated only at its event times $t^i_0, t^i_1, ..., t^i_\infty$. The control for j is updated only at its event times $t^j_0, t^j_1, ..., t^j_\infty$. However, in the case where not all vehicles communicate at each triggering instance, the control is updated to be

$$u_i(t) = -K \sum_{j \in N_i} (c_j x_i(t^i_{k_i}) - c_i x_j(t^i_{k_i})).$$

However, when i surfaces, it receives information about j timestamped at j's last surface time, $t^j_{k_j}$. So if j shares its last known position and computed velocity then

$$x_j(t^i_{k_i}) = x_j(t^j_{k_j}) + \dot{x}_j(t^j_{k_j})(t^i_{k_i} - t^j_{k_j}), \tag{45}$$

so substituting for $x_j(t^i_{k_i})$ the control is

$$u_i(t) = -\sum_{j \in N_i} c_j x_i(t^i_{k_i}) - c_i[x_j(t^j_{k_j}) + \dot{x}_j(t^j_{k_j})(t^i_{k_i} - t^j_{k_j})],$$

for $t \in [t^i_{k_i}, t_{k+1}{}^i_i)$. Next, the kinematics of agent i are

$$\dot{x}_i(t) = -\sum_{j \in N_i} (c_j x_i(t^i_{k_i}) - c_i x_j(t^i_{k_i})). \tag{46}$$

Following from page 17, the error does not change. As a result, we can perform the same substitution into the kinematic equation (46) to give

$$\dot{x}_i(t) = -\sum_{j \in N_i} [c_j(x_i(t) + e_i(t)) - c_i((x_j(t) + e_j(t) + \dot{x}_j(t^j_{k_j})(t^i_{k_i} - t^j_{k_j}))]$$

$$\dot{x}_i(t) = -\sum_{j \in N_i} [c_j x_i(t) - c_i x_j(t) + c_j e_i(t) - c_i e_j(t) - c_i \dot{x}_j(t^j_{k_j})(t^i_{k_i} - t^j_{k_j})]. \tag{47}$$

Rewriting (47) gives

$$\dot{x}_i(t) = -\sum_{j \in N_i} (c_j x_i(t) - c_i x_j(t)) - \sum_{j \in N_i} (c_j e_i(t) - c_i e_j(t)) + \sum_{j \in N_i} c_i \dot{x}_j(t^j_{k_j})(t^i_{k_i} - t^j_{k_j}).$$

Let $\gamma_j = \sum_{j \in N_i} c_i \dot{x}_j(t^j_{k_j})(t^i_{k_i} - t^j_{k_j})$, then

$$\dot{\mathbf{x}} = -H\bar{x} - H\bar{e} + \bar{\gamma}.$$

Let

$$V = \frac{1}{2}\mathbf{x}^T H\mathbf{x},$$

and let $\mathbf{z} = H\mathbf{x}$. Then,

$$\dot{V} = \mathbf{x}^T H\dot{\mathbf{x}} = \mathbf{x}^T H(-H\mathbf{x} - H\mathbf{e} + \gamma) = -\mathbf{x}^T HH\mathbf{x} - \mathbf{x}^T HH\mathbf{e} + \mathbf{x}^T H\gamma = -\mathbf{z}^T\mathbf{z} - \mathbf{z}^T H\mathbf{e} + \mathbf{z}^T\gamma.$$

Now

$$\dot{V} = -\sum_i z_i^2 - \sum_i \sum_{j\in N_i} z_i(c_j e_i - c_i e_j) + \sum_i \sum_{j\in N_i} z_i c_i \dot{x}_j(t_{k_j}^j)(t_{k_i}^i - t_{k_j}^j).$$

Let $C_i = \sum_{j\in N_i} c_j$ then

$$\dot{V} = -\sum_i z_i^2 - \sum_i C_i z_i e_i + \sum_i \sum_{j\in N_i} z_i c_i e_j + \sum_i \sum_{j\in N_i} z_i c_i \dot{x}_j(t_{k_j}^j)(t_{k_i}^i - t_{k_j}^j).$$

Using the inequality $\mathbf{xy} \le \frac{a}{2}x^2 + \frac{1}{2a}y^2$ for $a > 0$ on $z_i e_i$, $(z_i c_i)e_j$, and $z_i\gamma_i$,

$$\begin{aligned}\dot{V} \;\le\; & -\sum_i z_i^2 - \sum_i C_i\frac{a}{2}z_i^2 - \sum_i C_i\frac{1}{2a}e_i^2 \\ & + \sum_i \sum_{j\in N_i}[\frac{a}{2}(c_i z_i)^2 + \frac{1}{2a}e_j^2] + \sum_i \sum_{j\in N_i}(\frac{a}{2}z_i^2 + \frac{1}{2a}\gamma_j^2)\end{aligned}$$

$$\begin{aligned}\dot{V} \;\le\; & -\sum_i z_i^2 - \sum_i C_i\frac{a}{2}z_i^2 - \sum_i C_i\frac{1}{2a}e_i^2 \\ & + \sum_i \sum_{j\in N_i}\frac{a}{2}(c_i z_i)^2 + \sum_i \sum_{j\in N_i}\frac{1}{2a}e_j^2 + \sum_i \sum_{j\in N_i}(\frac{a}{2}z_i^2 + \frac{1}{2a}\gamma_j^2).\end{aligned}$$

We combine like terms and simplify to give

$$\dot{V} \le -\sum_i 1 - \frac{a}{2}(C_i - |N_i|c_i^2 + 1)z_i^2 - \sum_i C_i\frac{1}{2a}e_i^2 + \sum_i \sum_{j\in N_i}\frac{1}{2a}e_j^2 + \sum_i \sum_{j\in N_i}\frac{1}{2a}\gamma_j^2$$

assume $1 - \frac{a}{2}(C_i - |N_i|c_i^2 + 1) > 0$, then $a < \frac{2}{C_i - |N_i|c_i^2 + 1}$. Then $\dot{V} \le 0$ if

$$-\sum_i [(1 - \frac{a}{2}(C_i - |N_i|c_i^2 + 1))z_i^2 - C_i\frac{1}{2a}e_i^2 + \sum_j \frac{1}{2a}(e_j^2 + \gamma_j^2)] \le 0,$$

or

$$e_i^2 \le \frac{\sigma_i 2a}{C_i}[(1 - \frac{a}{2}(C_i - |N_i|c_i^2 + 1))z_i^2 - \sum j\frac{1}{2a}(e_j^2 + \gamma_j^2)].$$

To simplify the notation, let

$$\beta_i = \frac{\sigma_i 2a}{C_i}(1 - \frac{a}{2}(C_i - |N_i|c_i^2 + 1))z_i^2.$$

As expected, the β_i value changes to account for the c_i and c_j terms which will eventually scale the control to steer the vehicles to their individual desired positions in the formation.

$$e_i^2 \leq \beta_i z_i^2 - \frac{\sigma_i 2a}{C_i} \sum j \frac{1}{2a}(e_j^2 + \gamma_j^2)]$$

From (46), we have

$$\dot{x}_i(t) = -\sum_{j \in N_i} [c_j x_i(t_{k_i}^i) - c_i x_j(t_{k_i}^i)].$$

Integrating $\dot{x}_i(t)$ gives

$$x_i(t) = -\sum_{j \in N_i} (c_j x_i(t_{k_i}^i) - c_i x_j(t_{k_i}^i))(t - t_{k_i}^i) + x_i(t_{k_i}^i). \tag{48}$$

Plugging in for $x_i(t)$ and simplifying the left hand side of the equation renders

$$|\sum_{j \in N_i} c_j x_i(t_{k_i}^i) - c_i x_j(t_{k_i}^i)(t - t_{k_i}^i)|^2 \leq \beta z_i^2 - \frac{\sigma_i 2a}{C_i} \sum j \frac{1}{2a}(e_j^2 + \gamma_j^2).$$

Let

$$z_i(t) = -\sum_{j \in N_i} (c_j x_i(t) - c_i x_j(t)).$$

Integrate x_j the same way. From these denote, now

$$\rho_i = -\sum_{j \in N_i} (x_i(t_{k_i}^i) - x_j(t_{k_i}^i)),$$

and

$$\rho_j = -\sum_{l \in N_j} (x_j(t_{k_j}^j) - x_l(t_{k_j}^j)).$$

Also, let the discrete execution times be $\xi_i = t^* - t_{k_i}^i$ where t^* is a vector of candidate future trigger times. We follow the same procedure starting with equation (37) on page 19 to complete the derivation.

$$|\rho_i \xi_i|^2 \leq \beta(P_i \xi_i + \Phi_i)^2 - \frac{\sigma_i}{C_i} \sum_{j \in N_i} (e_j^2 + \gamma_j^2) \tag{49}$$

The trigger condition is when equivalence is established in (49).

As shown in Figure 12, the new derivation allows the vehicles to steer to desired final positions which in this simulation are located at $\mathbf{x}_f = [1, 2, 4, 5]$. There is no significant change in the number of triggering instance. In order to achieve desirable convergence results, we are able to manipulate the constant a from the triggering derivation. Manipulating a is comparable to changing the triggering gain of the system and ultimately affects the number of triggering instances and thus the convergence of the agents. The smaller we make a, the greater the number of triggering instances, and the more closely the trajectory models a first order differential equation. Figure 13 below illustrates the difference between

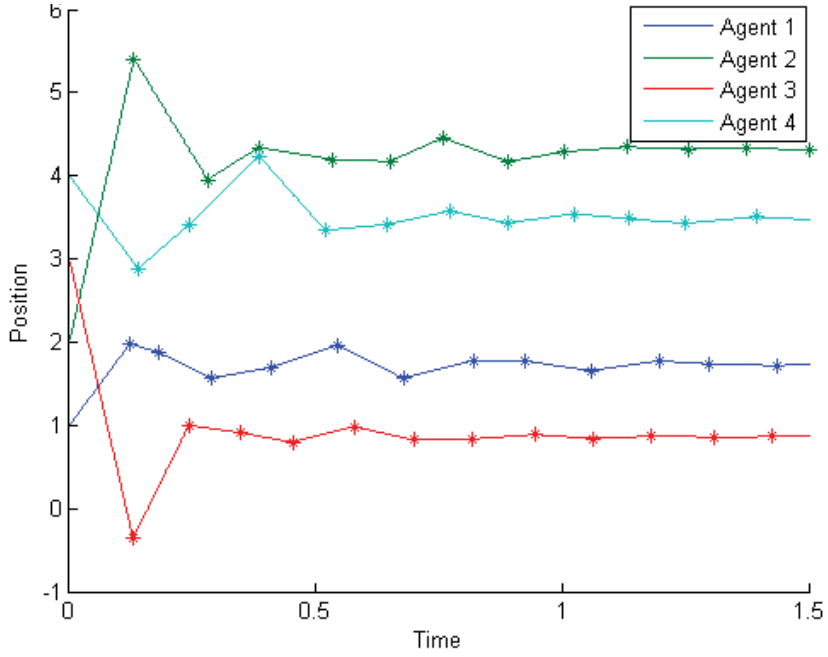

Figure 12: The new implementation allows the vehicles to be steered to a desired formation in 2-D. Note, since there is no absolute position term in our algorithm, we only guarantee that convergence is to some multiple of the desired formation, not to the absolute formation. The desired configuration is $\mathbf{x}_f = [1, 2, 4, 5]$.

the responses for an identical system (same initial and final desired positions) when the a value is changed. As apparent in the figure, there is a distinct tradeoff between the convergence behavior and the number of triggering conditions that can be optimized based on the desired behavior of the system. If a system of agents needs to be extremely accurate and converge quickly, then a smaller a value is more appropriate. However, these vehicles will be required to communicate more frequently. The idea is to increase a to the point where the convergence behavior just meets the desired convergence parameters. At this point, you will have the minimum communication and triggering instances necessary to achieve your desired configuration parameters for the given mission.

In order to deliver an algorithm capable of steering a multi-vehicle system to a a desired formation subject to time-delayed communication, the multi-vehicle control discussed in section 4.2 must be integrated with the distributed self-triggered control. To achieve this, we must first transition our simulations from being one dimensional to two dimensional. Since the entire derivation was based on regrouping terms and additive properties, we achieve the same results for both the x and y positions of the vehicles. As a result, the only change occurs with our graph laplacian. We simply add the potential from the x component to the y component to get a new potential. This equation becoming less than or equal to 0 indicated a triggering instance for the specific agent.

$$\dot{V} = \dot{V}_x + \dot{V}_y \tag{50}$$

By changing our definition of V, we create two terms for each term in the current control,

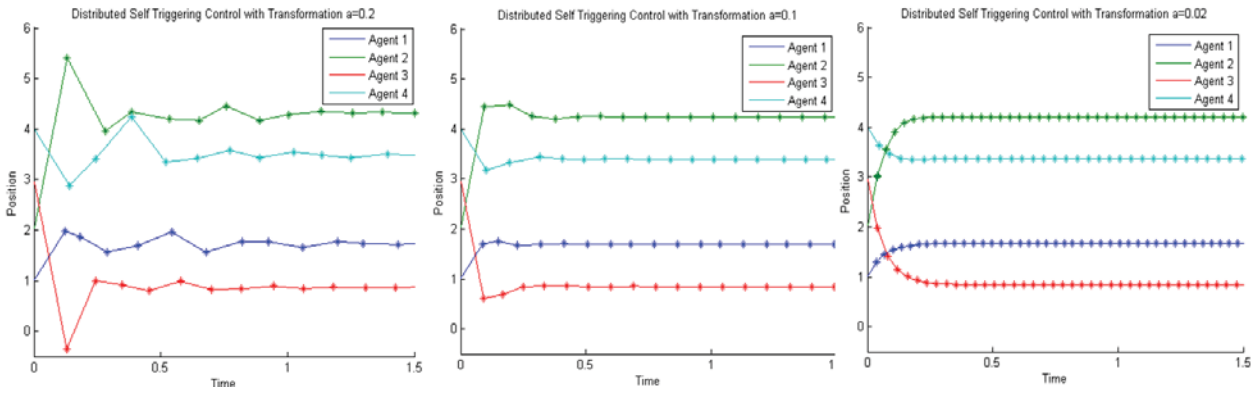

Figure 13: The constant a decreases in the figures from left to right. As the value decreases, the response more closely resembles a first order response, but the tradeoff is the increased triggering instances.

one for each dimension of travel. This update would account for the x and y dimensions that will allow for trajectory tracking. Further, we adapted the kinematics of the distributed-self triggering control to account for both the reference and actual vehicle trajectories. Following from the discussion of the trajectory tracking in Section 4.2, we have the reference vehicles establish the desired, ideal trajectory dictated by the kinematic controllers. The distributed self-triggering control is implemented with the actual vehicles which affects how they chase their desired trajectory.

4.5 Assignment of Optimal Formation Locations

In prior control literature, each vehicle is preassigned a position. If the desired configuration position is denoted by $\mathbf{x}_d = [x_{1,d}, x_{2,d}, ...]$, then the vehicle designated as x_2 will end up and the final position $x_{2,d}$. This assignment protocol is not efficient because the pre-assignment does not account for vehicle positions compared to their desired final positions. For example, vehicle three may be closer to $x_{2,d}$ than vehicle two. As a result, we implemented a permutation matrix to change the assignments of the final positions to the closest agent. A network flow optimization routine is used to calculate the permutation matrix. The cost is based on distance, meaning the new assignment protocol worked to minimize the total distance traveled between all vehicles. The set of all vehicle positions is defined by $\mathbf{x}$ and the set of all target positions as $\mathbf{x}_d$.

$$c_{ij} = \sqrt{(x_{d,x_i} - x_{x_i})^2 + (x_{d,y_i} - x_{y_i})^2}. \tag{51}$$

The permutation matrix is calculated from the the cost matrix.

$$Z = \min \sum_{i,j=1}^{N} Q_{ij} c_{ij} \tag{52}$$

Subject to the following constraints:

$$\sum_{i=1}^{N} Q_{ij} \in T \tag{53}$$

$$\sum_{j=1}^{N} Q_{ij} \in X \tag{54}$$

$$Q_{ij} = \{0, 1\}. \tag{55}$$

The constraints imposed ensure that every vehicle is assigned to one target and every target is assigned to one vehicle. Currently, this assumes that all of the vehicle know their desired positions, meaning $a_{0,n} = 1 \forall n$ but this can be relaxed to the subset of vehicles with knowledge of the desired positions. This equation sets to define the permutation as Q_{ij}. The permutation is implemented in according to the following equation.

$$\mathbf{x}_f^* = Q\mathbf{x}_f. \tag{56}$$

Since the permutation matrix, Q is simply a transformation of the identity matrix, I. Equation (56) reorders the desired positions to the closest agents. A simple simulation is used to show the difference between traditional and permutation assignments. Figure 14 shows the

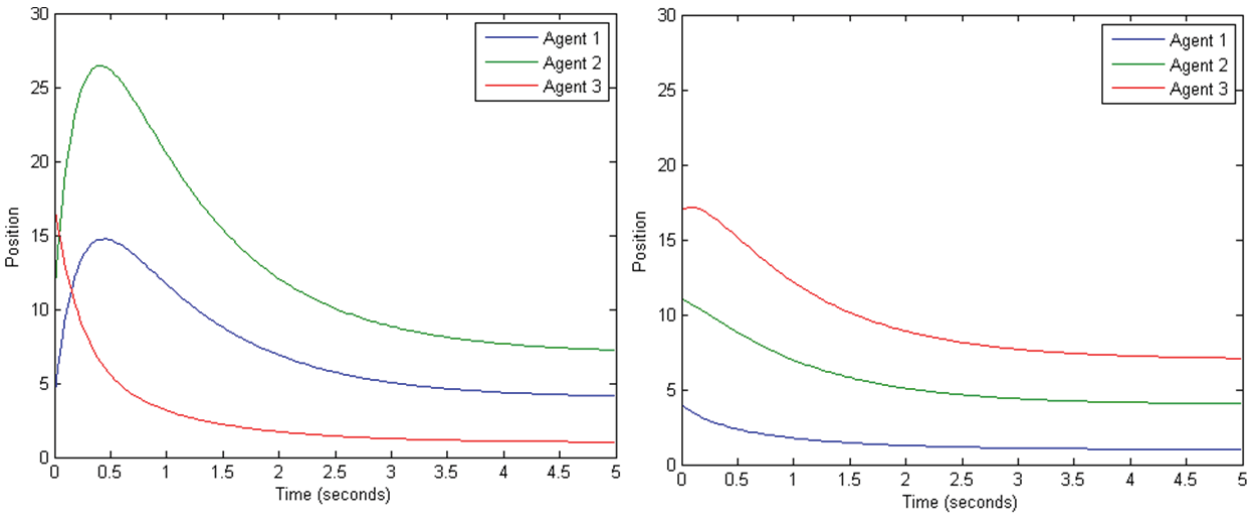

Figure 14: Without permutation (left), the vehicles drive to their preassigned final positions resulting in a large overshoot and aggressive actuation for some of the vehicles. With permutation implemented (right), the vehicle trajectories are much more controlled and the vehicle motion in smoother.

effects before and after the implementation of the permutations matrix, respectively as three agents converge to the configuration $\mathbf{x}_d = [1, 4, 7]^T$. The corresponding control u_i producing the trajectories in shown in Figure 15. As evident in Figure 14, the permutation matrix causes much smoother responses as opposed to the overshoot that occurs with traditional assignment. Although the settling time is approximately the same, the permutation implementation results in less aggressive control, which can reduce fuel consumption and models more realistic vehicle behavior as shown in Figure 15. Also, in this one dimensional example, it prevents the vehicles from crossing paths which eliminates the potential for collision.

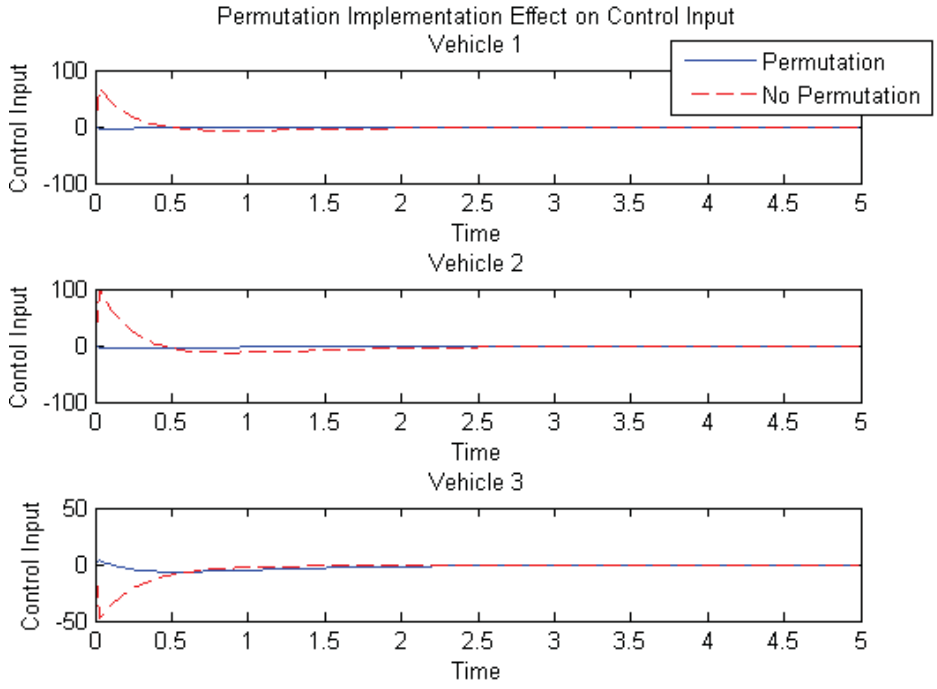

Figure 15: When the permutation is implemented, the control does not have a significant spike at the beginning indicating a less aggressive and more realistic control method.

5 APPLICATION

In order to deliver an algorithm capable of steering vehicles to a time-varying formation subject to self-triggered control, we must augment the results from the tracking control and the self triggering control. As shown in the control in Figure 6, the self triggering controller is implemented as a part of the reference trajectory kinematics. The self-triggering control is used to generate the desired trajectories which are then tracked by the actual vehicles governed by steering control. Figure 16 shows the convergence of the vehicles in a diamond formation and moving in a sawtooth pattern utilizing the self triggering control. The red circles represent the actual vehicles while the blue dashed line represents the reference trajectory defined by the self triggering kinematic controller.

Figure 16 shows the error over time of the formation. Due to the time-delayed nature of the communication, the error never reaches zero. Although it is greatly reduced, when the formation turns, the vehicles are unaware until their next update time, which accounts for the increase in error at each corresponding turning time.

The control can be applied to N number of vehicles. To show the versatility of the control, the $N = 6$ vehicles are steered into a hexagonal formation in Figure 17. The vehicles converge to the desired configuration. Again, the error temporarily increases at each turning instance of the formation due to the lag caused by the time-delay inherent to self triggered control.

The results from the aggregate simulation illustrate the ability to steer vehicles to desired configurations subject to time-delayed, directed communication. It theoretically justifies the ability to implement a system capable of conducting underwater missions without the reliance on underwater communication.

6 CONCLUSION

A multi-vehicle steering algorithm utilizing self-triggered control capable of driving vehicles to a desired formation subject to time-delayed communication is proposed in this paper. We demonstrated the ability to drive vehicles to a wide range of configurations, given that the configurations and paths can be defined by a parametric equation. We furthered the applicability of the research by applying an optimization routine which allows the vehicles to make informed decisions about their final positions. We also derived a distributed self-triggering control strategy capable of convergence in the presence of time-delayed, directed communication. We were able to minimize the agents communication instances which maximized their time conducting their given mission while still ensuring they converge to the desired formation. In the end, this project delivers a robust system that incorporates advanced control with a distributed self-triggering control strategy to steer a multi vehicle system to a desired, time-varying formation with time-delayed, directed communication.

The final simulation theoretically justifies the ability to implement a system to conduct underwater missions without relying on direct inter-agent communication. Next steps include implementing the system for ground vehicles and eventually underwater vehicles.

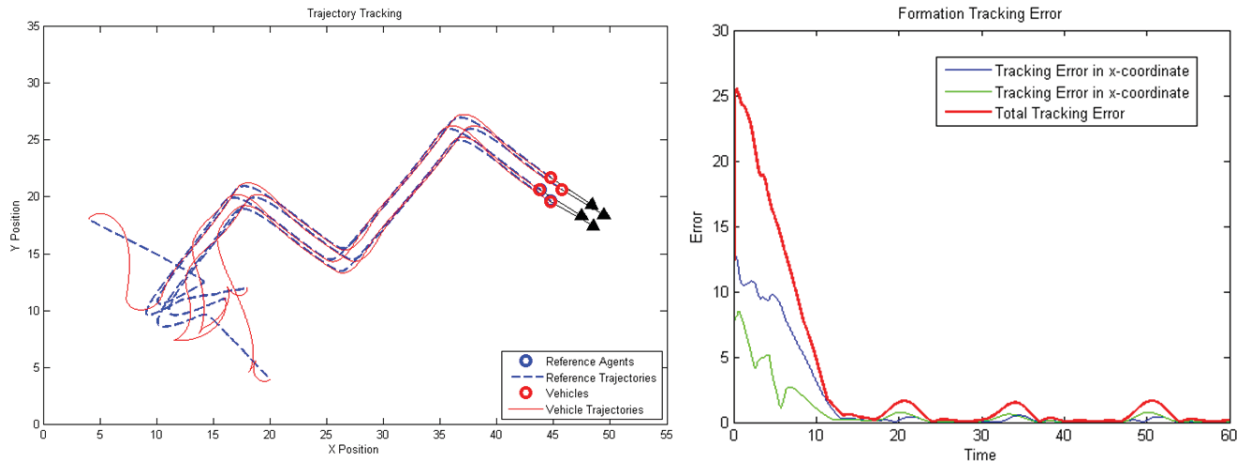

Figure 16: Implementation of target tracking with self-triggered control. The agents converge about the desired diamond formation and move about in the desired sawtooth path (left). As time passes, the formation error approaches zero (right). There are local maximums for the error at each turn in the formation. Because of the self-triggered control, the delayed nature of the communication will cause initial lag before the agents trigger to correct their position.

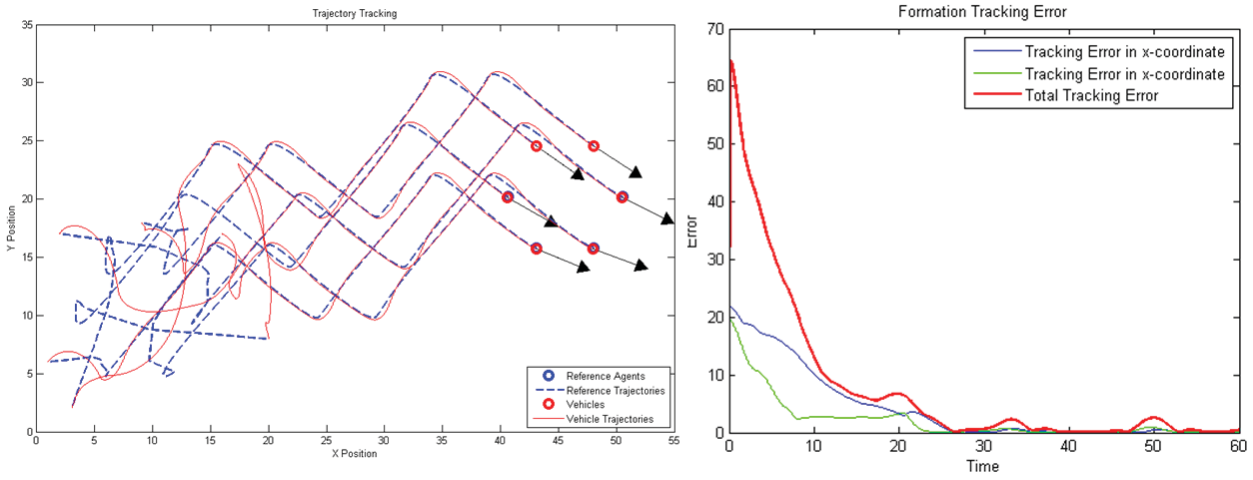

Figure 17: A hexagonal formation with N=6 agents is shown (left). The agents are able to use the algorithm to reach the desired, time varying formation. The error eventually approaches zero (right). There are spikes corresponding to the turning of the formation as the vehicles momentarily lag before their control is updated.

References

[1] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," *IEEE 51st Annual Conference on Decision and Control (CDC)*, pp. 3270–3285, 2012.

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

[3] D. B. West, *Introduction to Graph Theory*, 2nd ed. Pearson, 2001.

[4] Y. Kim and M. Mesbahi, "On Maximizing the Second Smallest Eigenvalue of a State-Dependent Graph Laplacian," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, 2006. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1576862

[5] L. DeVries and D. a. Paley, "Multivehicle Control in a Strong Flowfield with Application to Hurricane Sampling," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. June 2012, pp. 794–806, 2012.

[6] G. Shi and K. H. Johansson, "Randomized optimal consensus of multi-agent systems," *Automatica*, vol. 48, no. 12, pp. 3018–3030, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.automatica.2012.08.018

[7] R. Sepulchre, D. A. Paley, and N. E. Leonard, "Stabilization of Planar Collective Motion With Limited Communication," *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 706–719, 2008. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4484210

[8] R. Sepulchre, D. a. Paley, and N. E. Leonard, "Stabilization of Planar Collective Motion: All-to-All Communication," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 811–824, 2007.

[9] W. Ren and R. W. Beard, "Trajectory tracking for unmanned air vehicles with velocity and heading rate constraints," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 5, pp. 706–716, 2004.

[10] C. Peterson and D. a. Paley, "Multivehicle Coordination in an Estimated Time-Varying Flowfield," *Journal of Guidanance, Contol, and Dynamics*, vol. 34, no. 1, pp. 177–191, jan 2011. [Online]. Available: http://arc.aiaa.org/doi/abs/10.2514/1.50036

[11] D. A. Paley and C. Peterson, "Stabilization of Collective Motion in a Time-Invariant Flowfield," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 3, pp. 771–779, 2009. [Online]. Available: http://arc.aiaa.org/doi/abs/10.2514/1.40636

[12] A. Olshevsky and J. N. Tsitsiklis, "Degree fluctuations and the convergence time of consensus algorithms," *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pp. 2626–2631, 2013.

[13] M. Mesbahi, "On State-dependent dynamic graphs and their controllability properties," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 387–392, 2005. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1406134

[14] D. A. Paley, N. E. Leonard, and R. Sepulchre, "Stabilization of symmetric formations to motion around convex loops," *Systems and Control Letters*, vol. 57, no. 3, pp. 209–215, 2008.

[15] H. K. Khalil, *NonLinear Systems*, 3rd ed. Prentice Hall, 2002.

[16] E. Garcia, Y. Cao, and D. W. Casbeer, "Model-based event-triggered multi-vehicle coordinated tracking control using reduced order models," *Journal of the Franklin Institute*, vol. 351, no. 8, pp. 4271–4286, 2014. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0016003214001537

[17] E. Garcia, Y. Cao, H. Yu, P. Antsaklis, and D. Casbeer, "Decentralised event-triggered cooperative control with limited communication," *International Journal of Control*, vol. 86, no. 9, pp. 1479–1488, 2013. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/00207179.2013.787647

[18] E. J. Rodriguez-Seda, "Self-Triggered Collision Avoidance Control for Multi-Vehicle Systems," 2015.

[19] M. Mazo Jr., A. Anta, and P. Tabuada, "On Self-Triggered Control for Linear Systems: Guarantees and Complexity," *European Control Conference*, pp. 1–6, 2009. [Online]. Available: http://www.mmazojr.net/Work/Publications.html

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

[21] R. Postoyan, A. Anta, W. P. M. H. Heemels, P. Tabuada, and D. Nešić, "Periodic event-triggered control for nonlinear systems," *IEEE 52nd Annual Conference on Decision and Control*, vol. 58, no. 4, pp. 7397–7402, 2013.

[22] M. Donkers, P. Tabuada, and W. Heemels, "On the minimum attention control problem for linear systems: A linear programming approach," *IEEE Conference on Decision and Control and European Control Conference*, pp. 4717–4722, 2011.