NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

EXPLORATION OF FORMATION CONTROL ALGORITHMS FOR UNMANNED SURFACE VEHICLES

by

Jennifer T. Nguyen

December 2023

Thesis Advisor: Sean P. Kragelund
Second Reader: Isaac I. Kaminer

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2023	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
EXPLORATION OF FORMATION CONTROL ALGORITHMS FOR UNMANNED SURFACE VEHICLES

5. FUNDING NUMBERS
N0001422WX00617

6. AUTHOR(S) Jennifer T. Nguyen

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research (NPS CRUSER, Monterey, CA 93943)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

Unmanned surface vehicles (USV) and small combatant craft swarms are in increasing demand as adversarial forces develop their capabilities. The United States Navy is exploring the use of USV swarms for intelligence, surveillance, and reconnaissance (ISR) missions, hydrographic data collection, and surface vessel force protection. Currently, there are several different algorithms for controlling swarms consisting of the same platform. The discussion of the implementation of formation control algorithms on heterogeneous swarms is scarce. Heterogeneous swarms are formations consisting of vehicles with dissimilar dynamics. Example algorithms include the rigid formation control (RFC) algorithm and virtual bodies and artificial potential fields (VBAP). Through simulation testing, this thesis analyzes the strengths and weaknesses of the different formation control algorithms based on complexity of the software, ability to maintain a given formation shape, and overall robustness. The algorithms are also evaluated on their ability to command a heterogeneous swarm.

14. SUBJECT TERMS
unmanned surface vehicles, USV, autonomous systems, formation control, rigid formation control, RFC, virtual bodies and potential fields, VBAP, intelligence, surveillance, and reconnaissance, ISR

15. NUMBER OF PAGES
73

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

EXPLORATION OF FORMATION CONTROL ALGORITHMS FOR UNMANNED SURFACE VEHICLES

Jennifer T. Nguyen
Ensign, United States Navy
BS, United States Naval Academy, 2022

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2023

Approved by: Sean P. Kragelund
Advisor

Isaac I. Kaminer
Second Reader

Brian S. Bingham
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Unmanned surface vehicles (USV) and small combatant craft swarms are in increasing demand as adversarial forces develop their capabilities. The United States Navy is exploring the use of USV swarms for intelligence, surveillance, and reconnaissance (ISR) missions, hydrographic data collection, and surface vessel force protection. Currently, there are several different algorithms for controlling swarms consisting of the same platform. The discussion of the implementation of formation control algorithms on heterogeneous swarms is scarce. Heterogeneous swarms are formations consisting of vehicles with dissimilar dynamics. Example algorithms include the rigid formation control (RFC) algorithm and virtual bodies and artificial potential fields (VBAP). Through simulation testing, this thesis analyzes the strengths and weaknesses of the different formation control algorithms based on complexity of the software, ability to maintain a given formation shape, and overall robustness. The algorithms are also evaluated on their ability to command a heterogeneous swarm.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

APF	Artificial Potential Fields
DOD	Department of Defense
GPS	Global Positioning System
HVU	High Value Unit
ISR	Intelligence, Surveillance, and Reconnaissance
NOAA	National Oceanic and Atmospheric Administration
NPS	Naval Postgraduate School
RFC	Rigid Formation Control
ROS	Robot Operating System
USN	U.S. Navy
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
USV	Unmanned Surface Vehicle
UUV	Unmanned Underwater Vehicle
UxV	Unmanned Vehicle
VBAP	Virtual Bodies and Artificial Potentials

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

I would like to thank my advisor, Sean Kragelund, for your guidance, patience, and help over the past year. Although out time was short, I've learned so much and am grateful for the opportunity to continue working on our shared interest in autonomous swarms.

I would also like to thank CDR Paul Frontera and Dr. Matt Feemster at USNA for making the connection with Dr. Kragelund early on and their continued support post-academy.

Finally, I'd like to thank my mom, dad, brother, and fiancé for their support throughout my post-graduate education from the other side of the country. I am and will continue to better myself for you guys.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

There in an increasing need for autonomous systems in the United States Navy and military as a whole. The development of Unmanned Vehicle (UxV) systems in the military, whether it be ground, air, surface, or underwater, is slowly being integrated and will continue to be incorporated into military operations. To remain competitive with adversarial countries, research and development in unmanned technology is vital to maintaining a leading edge.

1.1 Motivation

As sensor and processing capabilities continue to grow, the ability to deploy autonomous vehicles also increases. Networks have developed to become stronger and much more secure, and developments in battery technology have allowed them to travel farther. When the Navy discusses autonomous vehicles, the two most common platforms mentioned and researched are Unmanned Aerial Vehicles (UAVs) and Unmanned Underwater Vehicles (UUVs). Unmanned Surface Vehicles (USVs) is growing topic of interest in the research and development community, more specifically USV swarms [1].

A USV is a vehicle that operates on the surface of a body of water without any crew or personnel onboard. They provide a wide array of capabilities and services because they are equipped to carry customizable payloads. Applications vary from commercial to military operations. On the commercial side, applications include climate monitoring, bathymetric data collection, offshore oil and gas pipeline maintenance, and hydrographic surveying [2]. Hydrographic surveys collect data from multi-beam sonar readings which measure the depth of the sea floor and other waterways using sound waves paired with global positioning system (GPS) data. The National Oceanic and Atmospheric Administration (NOAA) relies on hydrographic surveys to create and update nautical charts. These navigational charts serve commercial, military, and recreational mariners alike. They inform sailors of navigational hazards such as shallow features and shorelines.

Figure 1.1. Kings Bay, GA – Commercial tugs escort the nuclear-powered strategic missile submarine USS Tennessee (SSBN 734). Source: [3].

For naval applications, USVs can be used for escort missions for high-value assets. To the submarine and surface warfare community, USVs can offer force protection or guide transits to and from ports. In the fleet, there are currently U.S. Coast Guardsmen, sailors, or commercial workers driving small craft to conduct this task (Figure 1.1). Being constrained by their draft in the shallower waterways near ports, ships and submarines are restricted in maneuverability and speed. If it were to be attacked during this time of transit, then the force protection vehicles are trained to take the hit to ensure the submarine's safety (prevents damage and exposure to the nuclear reactor). USVs are also deployed to conduct Intelligence, Surveillance, and Reconnaissance (ISR) and search and rescue operations. The mobility and flexibility of these robots allow for a wide range of use cases with an easily interchangeable payload depending on the mission set at hand.

Figure 1.2. Tug boats escort the aircraft carrier USS John C. Stennis (CVN 74) as the ship pulls into Everett, Washington, for a scheduled port visit. Source: [4].

An advantage of using USV swarms are the increased breadth in a desired search area. It also saves time, cost, and energy when multiple smaller autonomous vehicles are conducting the same mission set as opposed to one manned vessel. Another advantage of using USVs are to keep personnel onboard safe rather than putting them at risk performing dangerous or time-consuming tasks. Approximately 80% of marine vessel accidents are caused by human error, fatigue, and/or distraction [5]. By implementing multiple USVs as an assured layer of safety, risk of collision or damage to the vessel(s) and its crew is mitigated. Utilizing a swarm allows operators to extend their range, minimize error, and/or shorten the time it takes to complete the task than if one vessel was utilized at a time.

To enable USVs to carry out these missions, there needs to be a program capable of commanding each vehicles' speed and bearing while remaining resilient against environmental factors (i.e., current or wind). In addition, there have not been many advancements to USV research compared to those in the unmanned aerial vehicle or unmanned underwater vehicle communities.

The typical naval fleets consists of dissimilar platforms: carriers, destroyers, cruisers, and even submarines. As the Navy produces more large and medium USVs, they will slowly be

integrated into conventional fleet formations [1]. The need for heterogeneous USV swarms will increase as there are multiple models being procured, such as the medium and large USV. In a report on Navy large unmanned surface and undersea vehicles, the Congressional Research Service states:

> Large USVs will complement the Navy's manned combatant force by delivering increased readiness, capability and needed capacity at lower procurement and sustainment costs and reduced risk to sailors. While unmanned surface vehicles are new additions to the fleet units, LUSV will combine robust and proven commercial vessel specifications with existing military payloads to rapidly and affordably expand the capacity and capability of the surface fleet. [1]

Future implementation of autonomous platforms in the Navy increase the chance of having to work with USV swarms of dissimilar dynamic properties. A robust algorithm that maintains a formation with a fixed-distance between agents is needed when traveling from one point to another.

1.2 Literature Review

There are several methods that researchers around the world are using to program USVs for performing point-to-point and/or obstacle avoidance maneuvers.

A common path-planning algorithm used for unmanned robots is the A* algorithm which utilizes graph theory to find the path that accumulates the least amount of cost. For path-planning algorithms, cost can vary depending on the desired outcome of the user or system, from the shortest amount of time of travelled, to the least power consumption, to the shortest distance from the start to the final point. The traditional A* algorithm sums the heuristic cost of neighboring nodes from the starting node to the end node.

$$f(n) = g(n) + h(n) \tag{1.1}$$

Equation 1.1 represents the formula to calculate the heuristic cost where $g(n)$ is the actual cost from the starting node to node n, $h(n)$ is the heuristic cost or heuristic value from

node n to the final node, and $f(n)$ is the estimated total cost from the starting node to the final node through node n. The A* algorithm computes all combinations of paths from the starting node to the final node until the optimal path is found. Once found, the agents in formation follow the coordinates of the optimal path. One downside to this approach is that the more nodes the graph contains, the longer it will take the algorithm to search. Another downside is that the normal and heuristic costs between each node needs to be pre-calculated or known beforehand in order to find the optimal solution. Figure 1.3 depicts an example of a pre-made cost map. Finding those values such as the distance, fuel emitted, or travel time between every node on a real map can be time consuming.

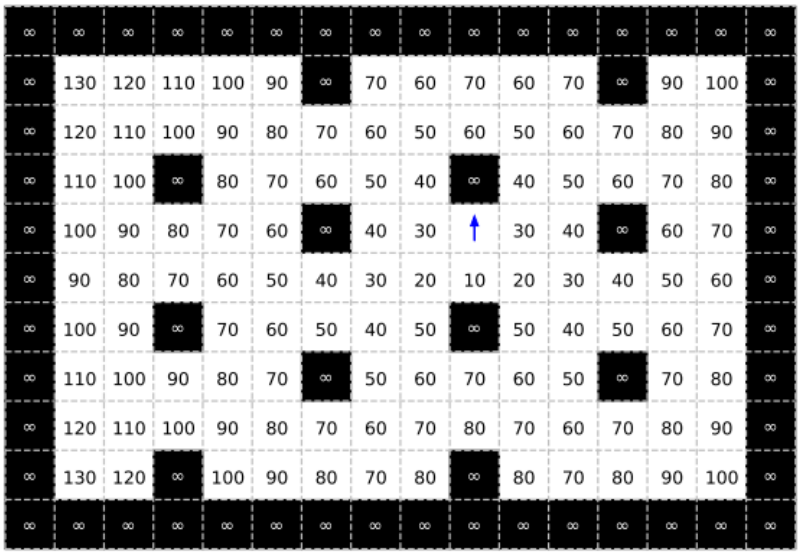


Figure 1.3. Sample heuristic map. Source: Sang et al. [6].

Sang et al. [6] created an improved A* algorithm created to accommodate a formation of multiple vehicles. First, a raster map is created to discretize the areas the USV can reach (Figure 1.4). Secondly, a minimum trajectory element between two states, ζ_{Pi}, was added to represent the areas the USV cannot reach. The disadvantage to this approach is that the user needs to create an in-depth environment ahead of time, prior to running the algorithm.

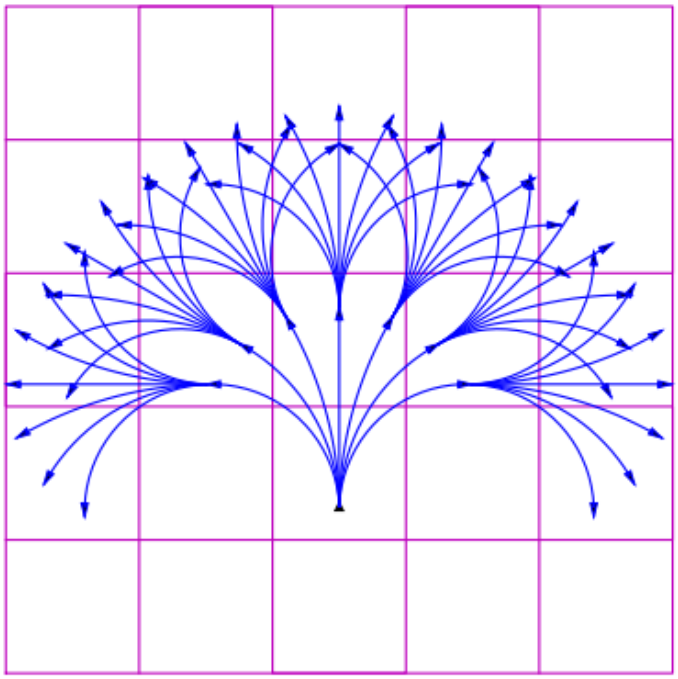


Figure 1.4. Sample raster map with a USV's reach overlay. Source: Sang et al. [6].

Artificial Potential Fields (APF) is another commonly used algorithm used for inter-agent and obstacle avoidance. It consists of vector fields that closely mimic a magnetic field. Attractive and repulsive forces are assigned to different features in the environment. The attractive forces are applied to objects such as the goal or endpoint of a system. The repulsive forces are applied to obstacles, walls/boundaries, and/or fellow robots to avoid collision (Figure 1.5). The agent operating with an artificial potential fields algorithm would move based on the summation of these attractive and repulsive forces enacted on it [7]. Figure 1.5 presents a visual representation of a APF vector map, with repulsive vectors pointing away from obstacle 1 (O_1), and the rest of the field attracting towards the goal.

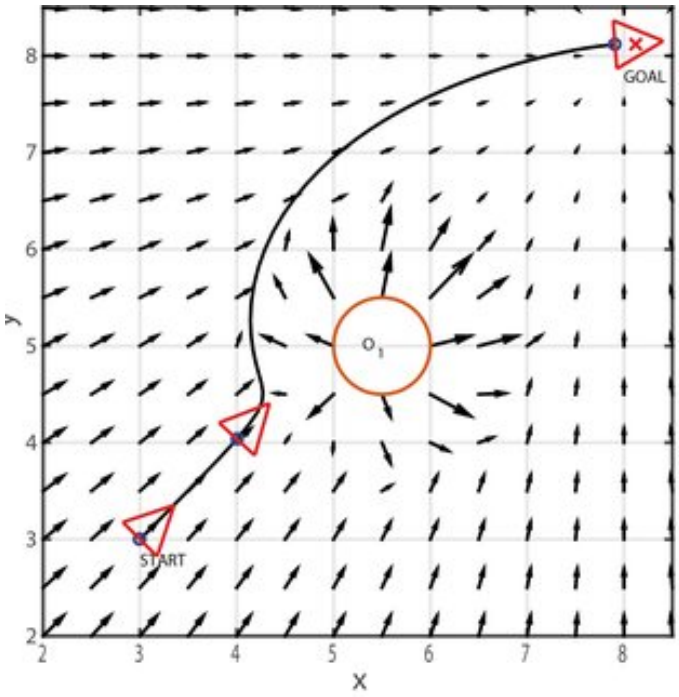


Figure 1.5. Sample artificial potential field vector map. Source: Sang et al. [6].

1.3 Problem Statement

The objective of this research is to evaluate which of two chosen algorithms will successfully guide a heterogeneous USV swarm (a formation consisting of different USV platforms) and maintain a desired separation distance. Different USVs maneuver using dissimilar dynamic characteristics and modes of propulsion/steering [8]. These different characteristics correlate to how they perform with each implemented algorithm. Heterogeneous swarms by nature possess a lower level of coordination than homogeneous swarms. The ultimate objective is to modify and evaluate pre-existing algorithms to control heterogeneous swarms.

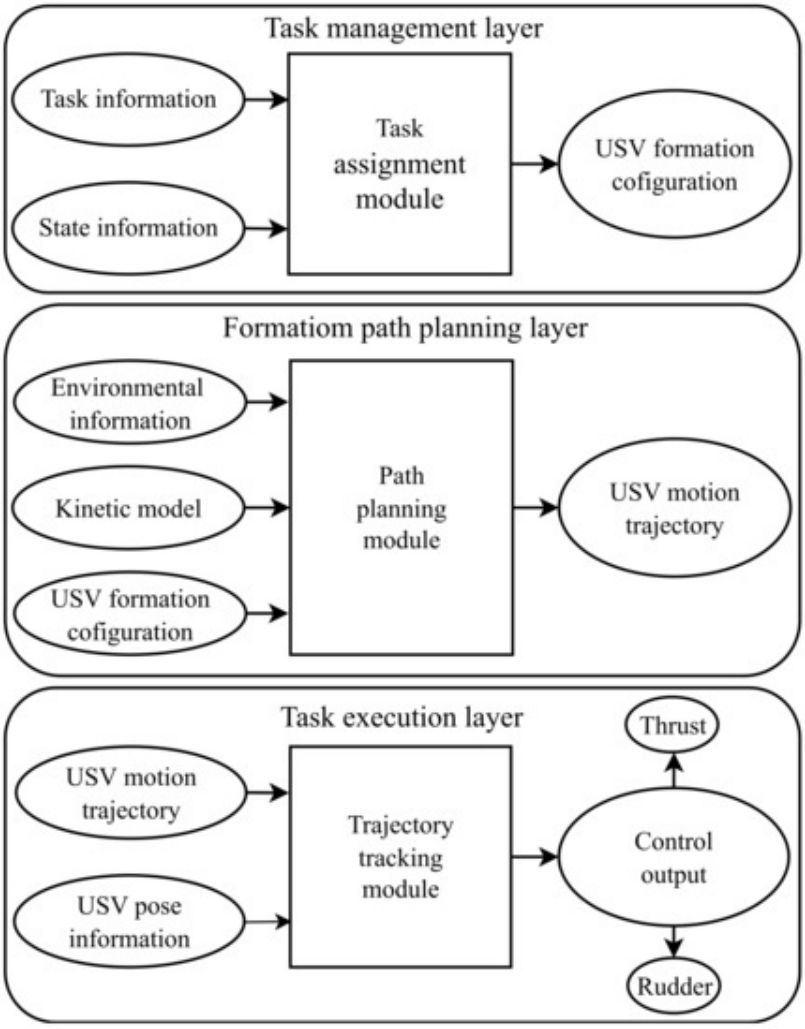


Figure 1.6. The layered software task infrastructure of unmanned surface vehicle control. Source: [6].

In a paper written by Sang [6], the operations that go into controlling a USV can be grouped into three categories: (i) the "Task management layer," (ii) the "Formation path planning layer," and (iii) the "Task execution layer." The research conducted in this paper focuses mainly on the "Formation Path Planning Layer" and the "Task Execution Layer" of the layered software task infrastructure model in Figure 1.6. The algorithms and vehicles used will be described in depth in Chapter 2.

1.4 Contributions

There is currently little to no work being done on heterogeneous USV formations for military applications. This thesis covers:

Heterogeneous USV swarms for future naval applications: As the Navy continues to produce multiple variants of USVs, the need for formations that are able to adapt to different agent dynamics increases. This research explores the performance of two of many formation-keeping algorithms with a heterogeneous swarm.

Side-by-side comparison of USV algorithms: Many current USV scholarly articles focus on one algorithm. This thesis compares two different types of algorithms and places them in a scenario that allows them to compete with each other.

Joint autonomy effort: The algorithms listed in the following section are those created and tested at the Naval Postgraduate School and the United States Naval Academy. To maintain a joint effort on autonomy research, this thesis helps build the bridge between the institutions.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Background

Multiple combinations of swarm configurations and algorithms will be tested. The performances of each test will then be analyzed to determine which algorithm is best suited heterogeneous USV swarms.

2.1 Algorithms

Two different algorithms will examined and tested: (i) Virtual Bodies and Artificial Potentials (VBAP) and (ii) Rigid Formation Control (RFC). Each algorithm maintains a multi-agent formation in different ways. The VBAP algorithm is an algorithm used by Naval Postgraduate School (NPS) to control small USV swarms for thesis research and field experimentation. VBAP has been implemented and adapted to operate on a low-cost USV, platform used at NPS described in Section 2.2.2. The simulated closed-loop model of this platform was developed from the aforementioned field experiment. The model takes in both heading and turn rate commands. Conversely, the Kingfisher USVs model, seen in Section 2.2.1, was derived from actuator inputs and sensed motion. An additional turn rate, heading, and velocity controller needed to be added to the Kingfisher model in order to use either formation algorithm. This information was used in order to standardize each algorithm and platform model to be able to later command them.

2.1.1 VBAP

VBAP is a variation of the APF algorithm discussed in Section 1.2. To avoid the complications that traditional potential field algorithms encounter, Fiorelli et al. [9] implemented an additional simulated *virtual leader* [9]. The virtual leader is a pointless mass, dynamical agent with an attractive force placed on it to maintain the desired velocity and heading of the formation. A constantly moving attractive potential field also minimizes the chance of the formation being caught in a local minima deadlock. The virtual leader's dynamics can be customized to more closely represent the agents it's leading. Repulsive forces are also implemented into the system. To prevent agent-to-agent collisions, repulsive forces are added to each simulated USV.

Figure 2.1 represents the VBAP algorithm created by Fiorelli et al. [9]. x_{ij} in Figure 2.1 represent the distance in between agents that will be evaluated against d_0 and d_1. For example, a repelling force is produced when a pair of vehicles is too close, or when $||x_{ij}|| < d_0$; an attracting force is produced when the vehicles are too far, or when $||x_{ij}|| > d_0$; and no force is induced when the vehicles are very far apart or when $||x_{ij}|| \geq \ \mathrm{d}_1 > d_0$.

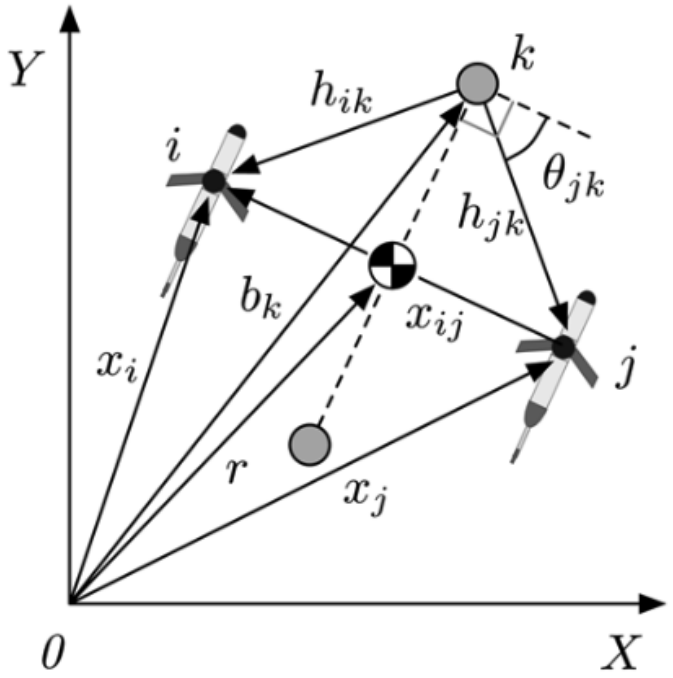


Figure 2.1. Notional diagram of VBAP algorithm. Source: [9].

2.1.2 Rigid Formation Control

One of the algorithms being evaluated in this thesis is the RFC algorithm [10], [11]. The RFC algorithm is a decentralized algorithm based on graph theory and the spacing or distance between agents. The desired distance values can be declared in the algorithm. Given the current position of each agent, the RFC algorithm produces a command velocity and heading for the agents to follow to maintain inter-agent spacing as well as an overall formation (Figure 2.2). It utilizes a proportional-integral (PI) controller to maintain the desired speed and heading. A PI controller contains a control loop that allows the current

state of the system to be sent as feedback and helps dictate the future performance of the system. The proportional (P) aspect of this controller increases as the error between the desired value and the actual value increases. The desired value in this case can be either speed or heading. The integral (I) aspect relates to the summation of the previous error values. The proportional and integral gains can be adjusted to reduce steady-state error in the system.

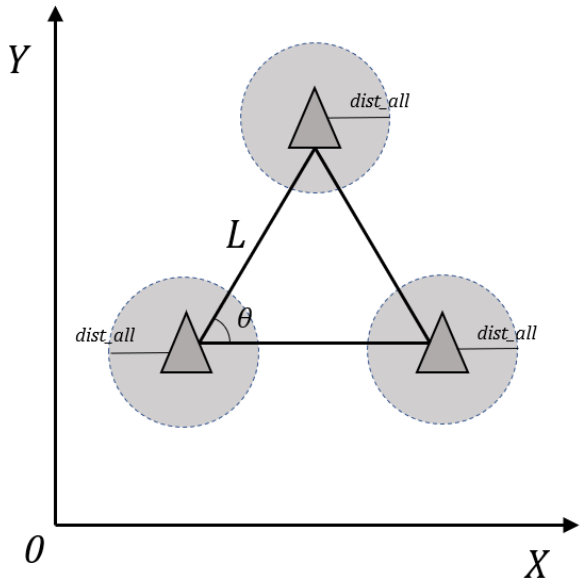


Figure 2.2. Notional diagram of RFC algorithm.

One unique addition from the original RFC algorithm was a distance error allowance [10]. RFC was originally created for Unmanned Ground Vehicles (UGVs), a platform that is much more maneuverable and is much less prone to disturbances from environmental factors on its performance. Compared to USVs, UGVs are not affected by current, wind, or side-slip velocities. The distance error allowance value created a buffer from the RFC algorithm to initiate any speed or heading correction to the agents. It kept vehicles from over-correcting and turning too far. This is significant because once one vehicle over-corrects, all of the other vehicles in the formation over-correct to maintain the desired inter-vehicle spacing. The RFC algorithm prioritizes maintaining distance between agents over following the given

speed and heading commands. Over-corrections destabilize the formation to a greater extent because USVs cannot maneuver backwards as well as UGVs. The addition of a distance error allowance was crucial in maintaining stability of RFC.

2.2 Vehicle Dynamics

Multiple platforms are simulated to evaluate the performance of the different control algorithms listed in section 2.1.

2.2.1 Kingfisher M200

The Kingfisher M200 is a catamaran-style USV created by Clearpath Robotics, Inc (Figure 2.3). The following set of dynamic equations are based off of the research and work from Nicholas Manzini in 2017 who modeled the Kingfisher USV in Gazebo [12]. The state variables used in this model are as shown in Equation 2.1.

$$\bar{x} = \begin{bmatrix} x \text{ [m]} \\ y \text{ [m]} \\ \psi \text{ [rad]} \\ u \text{ [m/s]} \\ v \text{ [m/s]} \\ r \text{ [rad/s]} \end{bmatrix} \tag{2.1}$$

The variables x and y represent the east and northward position of each agent. The heading angle is represented as ψ, u is the surge velocity, v is the sway velocity, and r is the turning rate or yaw about the z-axis.

Figure 2.3. Clearpath Robotics Kingfisher M200 on shore.

The velocities in the east and northward directions are represented as $\dot{x}$ and $\dot{y}$ (Equations 2.2 and 2.3). The turning rate is denoted as $\dot{\psi}$ (Equation 2.4). The nonlinear surge acceleration is shown as $\dot{u}$ (Equation 2.5). The nonlinear sway acceleration and the nonlinear yaw acceleration are represented as $\dot{v}$ and $\dot{r}$ respectively (Equations 2.6 and 2.7). The control variables are the port and starboard thruster commands, T_{port} and T_{stbd}. These two variables are grouped into a matrix, $control$, since u already represents surge. The thrusters have a maximum output of 41.5 N and a minimum output of -12.5 N [12].

$$\dot{x} = u\cos\psi - v\sin\psi \tag{2.2}$$

$$\dot{y} = u\sin\psi + v\cos\psi \tag{2.3}$$

$$\dot{\psi} = r \tag{2.4}$$

$$\dot{u} = \frac{(X_{|u|u}|u|u) + T_{port} + T_{stbd}}{(m - X_{\dot{u}})} \tag{2.5}$$

$$\dot{v} = \frac{(Y_v v) + (Y_r - m)ur}{(m - Y_{\dot{v}})} \tag{2.6}$$

$$\dot{r} = \frac{(N_{|r|r}|r|r) + (T_{port} - T_{stbd})(d/2)}{(I_{zz} - N_{\dot{r}})} \tag{2.7}$$

Note that the state variables u and r in Equations 2.5 and 2.7 are multiplied in the numerator by the absolute values of themselves. This becomes a problem when trying to propagate through these equations to evaluate the performance of the dynamical system. The changes made were to turn $|u|u$ into u^2 as surge should not be negative under the assumption that the USVs do not maneuver in reverse. The second change was changing $|r|r$ into a hyperbolic sine equation since the yaw rate depended on the sign of r (a negative vs. a positive r could mean the difference between a left or right turn). If r was just squared like u was, then the USV would only be limited to right turns. After trial and error, coefficients were added to the initial $sinh$ function to accurately match the function of $|r|r$ (Equation 2.8).

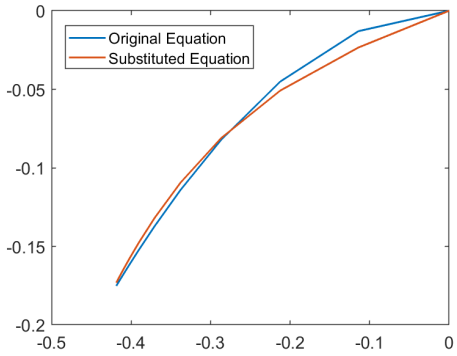
$$0.035(\sinh(5.5r)) \tag{2.8}$$

Equation2.8 smoothly approximates the original function, as shown in Figure 2.4.

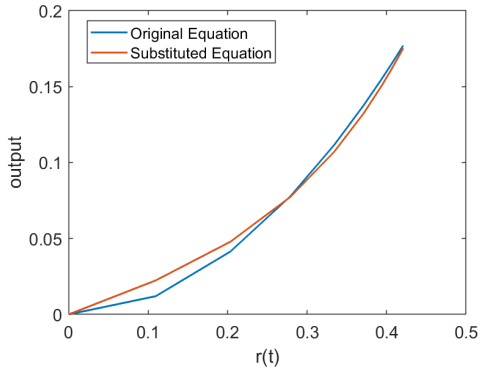
The new state equations for the surge and yaw rates are represented in Equations 2.9 and 2.10.

$$\dot{u} = \frac{(X_{|u|u}u^2) + T_{port} + T_{stbd}}{(m - X_{\dot{u}})} \tag{2.9}$$

$$\dot{r} = \frac{(N_{|r|r}(0.035\sinh 5.5r) + (T_{port} - T_{stbd})(d/2)}{(I_{zz} - N_{\dot{r}})} \tag{2.10}$$



(a) Negative values of r.



(b) Positive values of r.

Figure 2.4. The negative and positive values of r using the $|r|r$ (original) vs. $0.035(\sinh 5.5r)$ (substituted) over time.

The constant parameters in Equations 2.2 to 2.4 are defined in Table 2.1. An additional speed and heading controller were implemented into the Kingfisher model since this was an open-loop model. The current Kingfisher model performed given specific heading and velocity commands from the user. These additional controllers were implemented in order for the model to read the velocity and heading commands and produce the correct output in their own using a proportional-integral-derivative (PID) controller. The PID controller is much like the PI controller with an additional gain, the derivative gain. This value dictates what the future value is going to be based on the current rate of change. The PID controller was implemented into the speed and heading controller of the Kingfisher model.

Table 2.1. Kingfisher USV model constant parameters. Adapted from [12].

Label	Variable	Value [units]
Rigid Body Mass	m	36.0 kg
Added Mass (Surge)	$X_{\dot{u}}$	61.7 kg
Linear Drag (Surge)	$X_u u$	0.0 $\frac{N}{(m/s)}$
Quadratic Drag (Surge)	$X_{\|u\|u}$	16.9 $\frac{N}{(m/s)^2}$
Added Mass (Sway)	$Y_{\dot{v}}$	-34.5 kg
Linear Drag (Sway)	Y_v	-9.3 $\frac{N}{(m/s)}$
Coupled Linear Drag (Sway)	Y_r	1.0 N $\cdot s$
Rigid Body Inertia (Yaw)	I_{zz}	8.35 kg $\cdot m^2$
Added Mass Inertia (Yaw)	$N_{\dot{r}}$	12.4 kg $\cdot m^2$
Linear Drag (Yaw)	$N_r r$	0.0 $\frac{N \cdot m}{(rad/s)}$
Quadratic Drag (Yaw)	$N_{\|r\|r}$	139.0 $\frac{N \cdot m}{(rad/s)^2}$
Distance between Thrusters	d	0.9 m

2.2.2 Sandwich Boat

The "Sandwich Boat" is another USV platform developed by Naval Surface Warfare Center (NSWC) Carderock. It is a low-cost test platform that earned its namesake because of the watertight food storage container used to house its battery and electronic components. Unlike the Kingfisher, the sandwich boat only contains one thruster; however, it is also equipped with a rudder (2.5). The rudder is attached to a servo motor that responds to steering controller outputs. This means that it will take in one thruster and one rudder command as opposed to two thruster commands that the Kingfisher takes in.

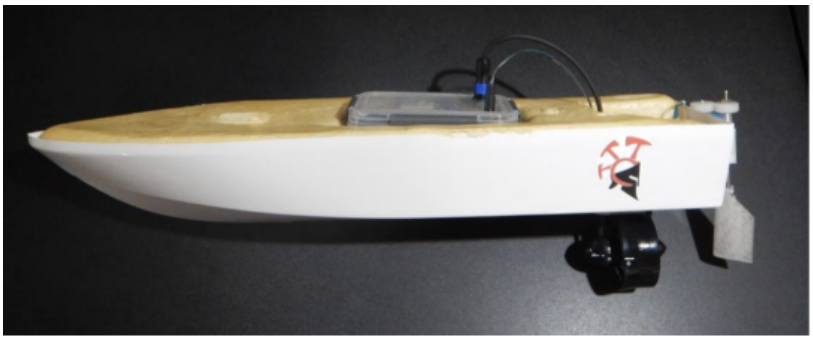


Figure 2.5. NPS sandwich boat USV. Source: [13]

Simple transfer functions for the sandwich boat speed and turning dynamics are shown in Equation 2.11, modeled as a transfer function.

$$u = \frac{1}{(s+1)}\, u_c \tag{2.11}$$

The speed command is u_c [m/s], and the surge velocity output is u [m/s].

The closed-loop transfer function for the sandwich boat turning dynamics was identified from sea trial data and is represented in Equation 2.12.

$$r = \frac{15.68}{(s^2 + 5.44s + 16.0)}\, r_c \tag{2.12}$$

The turn rate command is r_c [m/s], and the turn rate output is r [m/s].

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Simulation Framework

The USV dynamics described in section 2.2 were simulated using Robot Operating System (ROS). Additional ROS tools were utilized for simulated visualization, controller tuning, and data collection.

3.1 ROS Infrastructure

ROS packages were developed to implement i) speed, turn rate, and heading controllers for the Kingfisher USV; ii) the VBAP algorithm and its virtual leader; and iii) the RFC algorithm. Launch files were also constructed to run the dynamics for the Kingfisher and sandwich boats as well as the environments to run single-USV and triple-USV simulations.

Each simulated test contains three USVs, each labeled with a unique identification number: 9995, 9997, and 9999. The initial conditions for each agent are listed in Table 3.1. All three agents in their initial start points in the virtual environment are represented in Figure 3.1.

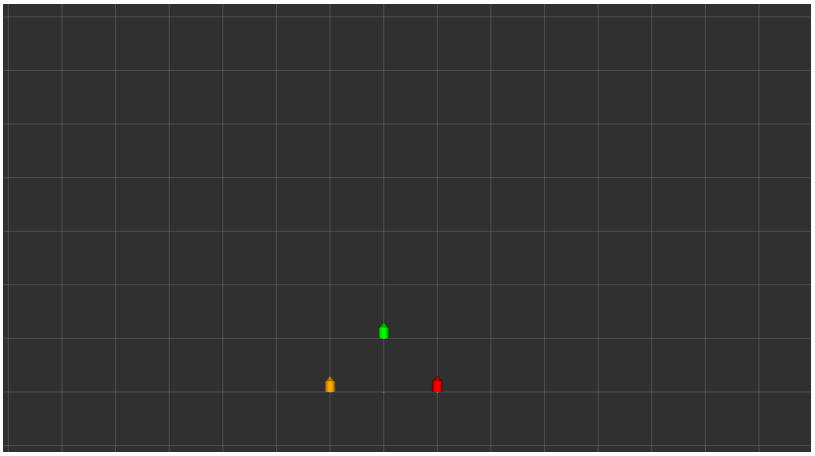


Figure 3.1. Three simulated USVs in their initial positions in RQT.

Table 3.1. Simulation parameters.

Label	Value [units]
Desired Separation Distance	10 m
Simulation Duration	90 sec
USV 9995 Initial Position	(10 m, 0 m)
USV 9997 Initial Position	(0 m, -10 m)
USV 9999 Initial Position	(0 m, 10 m)
Command Update Frequency	(10 Hz)

3.2 Testing Metrics

The two algorithms listed in Chapter 2 are evaluated under pre-determined metrics to observe which performs the best. The four metrics selected for analyzing the performance of each algorithm are:

- Mean Inter-Vehicle Distance (m)
- Minimum Inter-Vehicle Distance (m)
- Inter-Vehicle Distance Error (m)
- Time to Converge (m)

The mean inter-vehicle distance is the average distance between each USV over the duration of the simulated test. This value is taken from taking the Euclidean distance between each agent (Equation 3.1) at each time step, taking the collective sum of distances between all agents, and dividing that value by the number of time steps, s (Equation 3.2).

$$dist_{ij}(t) = \sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2} \tag{3.1}$$

$$mean\ ivd = \frac{\sum_{i=\Delta t}^{t_f} dist_{ij}(t)}{s} \tag{3.2}$$

The mean inter-vehicle distance shows how well the formation maintains a given desired separation distance. The minimum inter-vehicle distance is the shortest distance that any two USVs get to each other throughout the duration of the simulation. This value identifies how close the agents get to each other and ensures that there is no risk of collision. If the minimum inter-vehicle distance gets to zero, that indicates that a collision occurred. The inter-vehicle distance error is the summed error between all agents. The error is considered the difference between the current inter-vehicle distance and the desired separation distance, L. The error is calculated by summing the difference in distance between all agents in formation at each time step in the simulation.

$$error_{ij}(t) = \sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2} - L \tag{3.3}$$

The time to converge is recorded at the time all three agents reach the desired separation distance from each other from the initial condition. The time to converge shows how the algorithms prioritize the overall formation speed and heading commands from the inter-vehicle distance.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Results and Analysis

Simulations are run with different combinations of the algorithms and platforms described in Chapter 2. The first two tests evaluate a homogeneous swarm of USVs guided by the algorithm used in previous field experimentation: (i) the VBAP algorithm operating on sandwich boats and (ii) the RFC algorithm operating on Kingfishers. The third and fourth tests are the algorithm executed on a homogeneous swarm of different USVs: (iii) the VBAP algorithm operating on Kingfishers and (ii) the RFC algorithm operating on sandwich boats. The final two tests examined the performance of each algorithm when used to guide a heterogeneous swarm. The simulated heterogeneous swarm consists of one Kingfisher and two sandwich boats.

4.1 Three Sandwich Boats Using VBAP

The first test consists of a homogeneous swarm of three sandwich boats following a virtual leader executing a waypoint box pattern. The gains associated with this algorithm had to be tuned in order for the sandwich boats maintain its formation. The tuning process was conducted via trial and error, going through each gain value one-by-one and running the simulation. The four test metrics described in Section 3.2 were used in evaluating whether the change in gains improved or worsened. Table 4.1 lists the values for each parameter when implementing VBAP onto the sandwich boats. The values that induced the most change were the gains for the agent interactive forces (α) and the turn rate command output (K_r). α affects the attractive and repulsive forces that the agents have on each other. This keeps the agents from spreading too far apart or getting too close together. K_r affects the influence of the turn rate command output. Because the sandwich boats are equipped with a rudder, it is able to make sharper turns and K_r can be lower. A higher value of K_r would cause the sandwich boats to overshoot their turns and cause oscillations when course correcting.

Table 4.1. VBAP variable and parameter inputs for sandwich boat implementation.

Parameter	Variable	Value
Gain for dissipation force	K_d	6.0
Gain for High Value Unit (HVU) attraction force	K_{hvu}	6.0
Gain for norm of velocity (m/s)	K_{vNorm}	1.3
Gain for turn rate command output	K_r	0.3
Gain for speed command output	K_v	0.1
Desired swarm separation distance	d_0	10 m
Distance Threshold to ignore other agents	d_1	25 m
Gain for agent interactive forces	α	5.0
Value to prevent undefined condition	ϵ	0.01
Minimum USV speed command	v_{min}	0.0 $\frac{m}{s}$
Maximum USV speed command	v_{max}	1.6 $\frac{m}{s}$
Maximum USV turn rate	r_{max}	0.785 $\frac{rad}{s}$
Desired virtual leader speed	v_{leader}	1.5 $\frac{m}{s}$

Figure 4.1 shows the paths that each USV took from the initial condition to its final position after 90 seconds. The virtual leader is traveling in a 30 m by 30 m box with an initial position of (0 m, 10 m). Because the agents are attracted to the virtual leader, they cut the corners of the square in order to remain close to the virtual leader. One notable characteristic of the VBAP algorithm is that the initial positions of the USVs are not initially assigned with one virtual leader. Throughout the simulation, the agents will shift and reposition themselves in order to maintain inter-vehicle separation distance. In this case, USV 9995 is initially in the front of USVs 9997 and 9999. At the end of the simulation, USV 9997 and 9995 are in front of USV 9999. The VBAP algorithm does not enforce specific formation positions with only one virtual leader.

Figures 4.3 and 4.2 provide insight on the speed and turn rate commands to each USV. The speed reaches an average of approximately 1 (m/s) for each USV. Compared to the virtual

leader speed command of 1.5 (m/s), the USVs operate at a speed closer to 1 because they are cutting the corners of the box and are travelling a shorter distance than the path the virtual leader is taking. To prevent themselves from passing the virtual leader, the USVs move slightly slower. There are increases in the turn rate for every leg in the desired box which can be seen every 20 seconds.

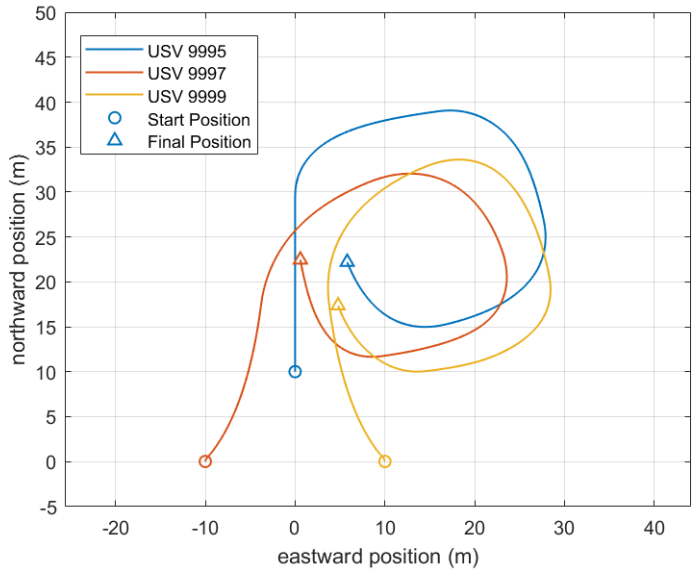


Figure 4.1. The northward vs. eastward position (m) of each simulated agent and their trajectories over 90 sec.

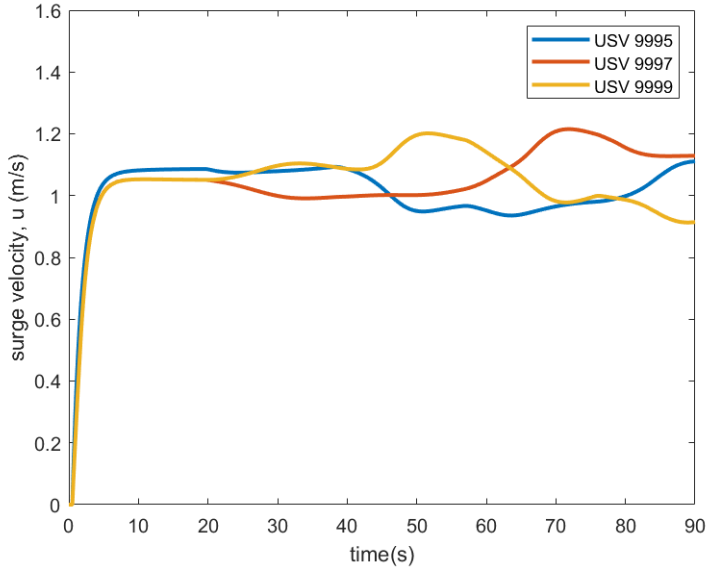


Figure 4.2. Each USV's velocity (m/s) over 90 sec.

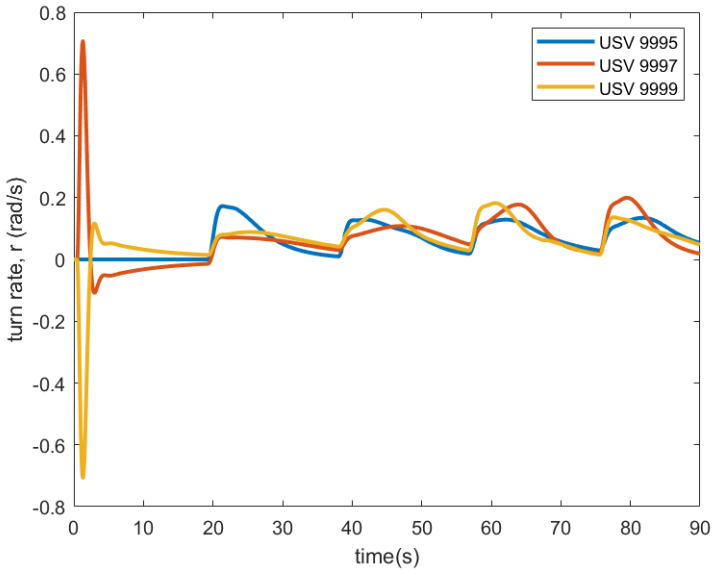


Figure 4.3. Each USV's turn rate (rad/s) over 90 sec.

The distance between each USV over time is presented in Figure 4.4. The important things to

note from the separation distance graphs are how close the trends are to the desired separation distance and to ensure that the value is not zero. A zero value in separation distance indicates a crash between two agents. In this test, the USVs maintain approximately 5 to 6.5 m of separation.

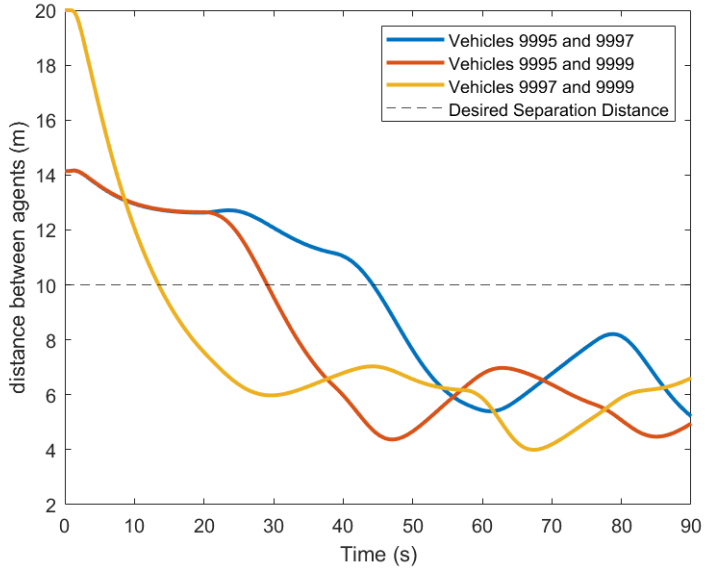


Figure 4.4. The distance (m) between each simulated agent over 90 sec. The desired separation distance is 10 m (dashed line).

4.2 Three Kingfishers Using RFC

The second test consists of a homogeneous swarm of three sandwich boats utilizing the RFC algorithm. When implementing RFC, the only parameter that is adjusted is the distance maintenance gain, K_d. Through thorough simulation testing, it was discovered that RFC is highly sensitive to the distance maintenance gain. If the gain value is too high, the agents will continuously over-correct, deeming the formation unstable. K_d is set equal to 0.0001 for this simulation. This extremely low value suggests that additional investigation is required to verify the algorithm has been properly adapted to this problem; however, previous testing conducted by Frontera et al. [10] also used a small value for this gain.

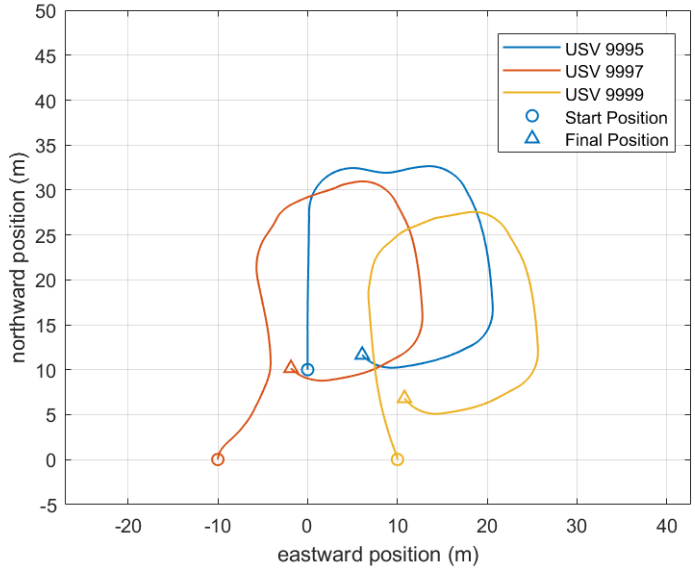


Figure 4.5. The northward vs. eastward position (m) of each simulated agent and their trajectories over 90 sec.

Figure 4.5 shows the trajectories of each USV took from the initial condition to its final position after 90 seconds. To approximate the VBAP virtual leader's waypoint box pattern, 90-degree heading changes were commanded every 20 seconds by adding $\pi/2$ radians to the current heading. The RFC algorithm maintains ample separation distance. One interesting characteristic about the rigid formation control algorithm is that the "shape" of the formation is maintained throughout the whole simulation. The agent at the top of the triangle initially remains in a position 10 meters greater than the other two agents in the northward direction at all times. Even when the formation is commanded to change heading and move laterally, the agent in the "front" does not rotate about the center of the formation to remain in front. All agents maintained the desired triangle formation independent of the specified bearing (Figure 4.5).

The Kingfishers oscillate in speed as they traverse through the simulated environment and adjust to maintain separation distance (Figure 4.5). The distance maintenance gain is close to zero by an order of magnitude of 10^{-4}. Figures 4.6 and 4.7 provide insight on the oscillatory behavior through the speed and turn rate commands sent to each USV. This could partially be due to the PID controller gains that were used for the speed and steering controllers in the Kingfisher model.

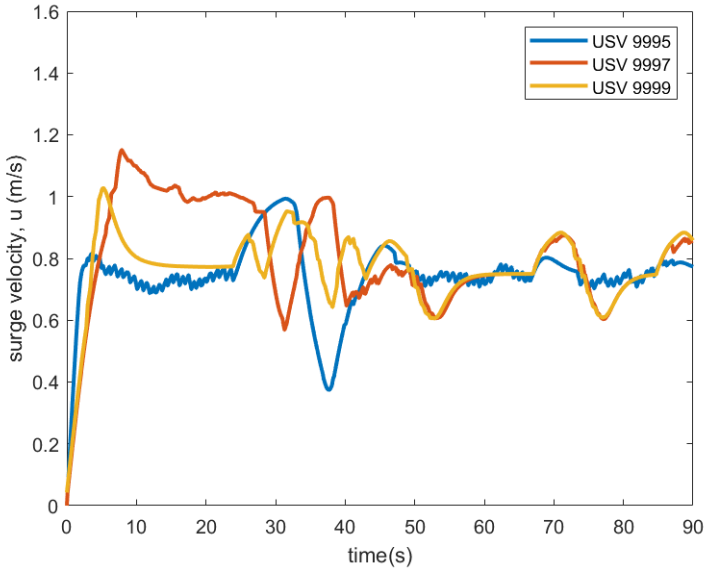


Figure 4.6. Each USV's velocity (m/s) over 90 sec.

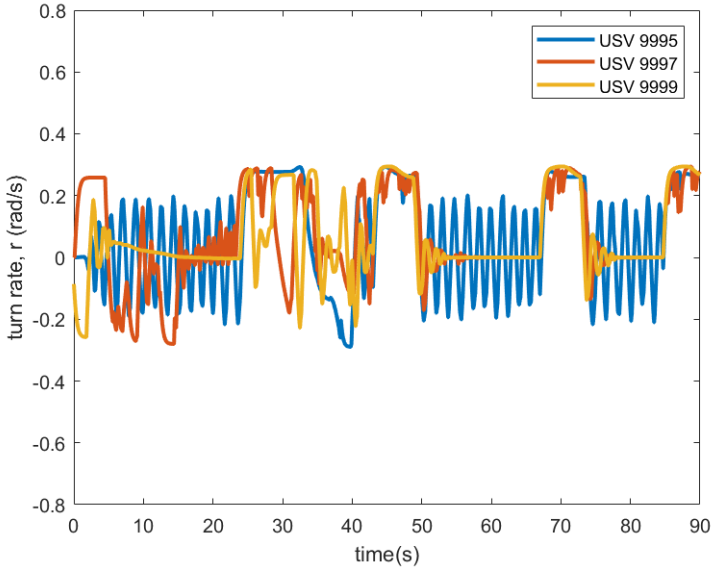


Figure 4.7. Each USV's turn rate (rad/s) over 90 sec.

The distance between each USV over time is presented in Figure 4.8. The distance between USVs 9997 and 9999 remain significantly larger than that of their separation between USV 9995. The probable cause is the initial separation distance. USVs 9997 and 9999 start 20 meters apart whereas USVs 9997 and 9999 from USV 9995 start 14.2 meters apart.

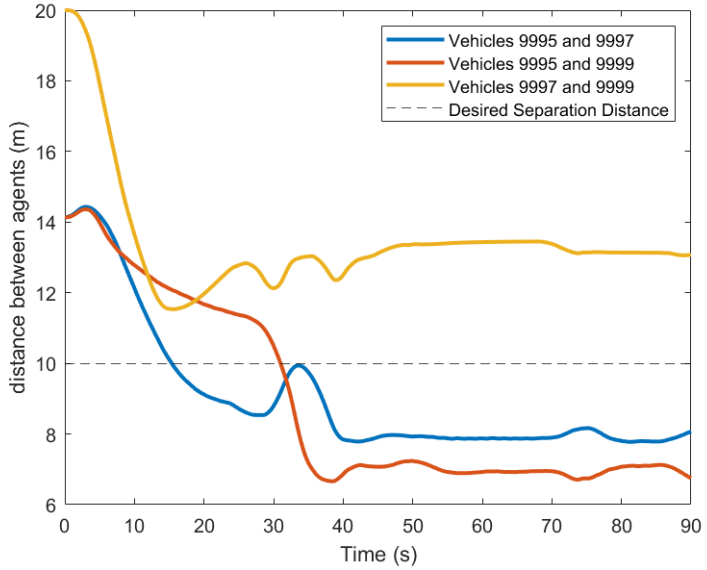


Figure 4.8. The distance (m) between each simulated agent over 90 sec. The desired separation distance is 10 m (dashed line).

4.3 Three Kingfishers Using VBAP

The third test implements VBAP onto the Kingfisher model. Because this model is slower and accounts for sway dynamics, whereas the sandwich boat model does not, several parameter adjustments were required to tailor the VBAP algorithm to the Kingfisher platform. The gains for dissipation, HVU attraction force, and ϵ value had to be increased while the norm of the velocity gain, turn rate command output gain, α, and maximum virtual leader speed had to be decreased (Table 4.2).

Table 4.2. VBAP variable and parameter inputs for Kingfisher USV implementation.

Parameter	Variable	Value
Gain for Dissipation Force	K_d	7.5
Gain for HVU Attraction Force	K_{hvu}	8.0
Gain for Norm of Velocity (m/s)	K_{vNorm}	1.0
Gain for Turn Rate Command Output	K_r	0.15
Gain for Speed Command Output	K_v	0.1
Desired Swarm Separation Distance	d_0	10 m
Distance Threshold to Ignore Other Agents	d_1	25 m
Gain for Agent Interactive Forces	α	1.0
Value to Prevent Undefined Condition	ϵ	0.5
Minimum USV speed command	v_{min}	0.0 $\frac{m}{s}$
Minimum USV speed command	v_{min}	0.0 $\frac{m}{s}$
Maximum USV speed command	v_{max}	1.6 $\frac{m}{s}$
Maximum USV turn rate	r_{max}	0.785 $\frac{rad}{s}$
Desired virtual leader speed	v_{leader}	1.5 $\frac{m}{s}$

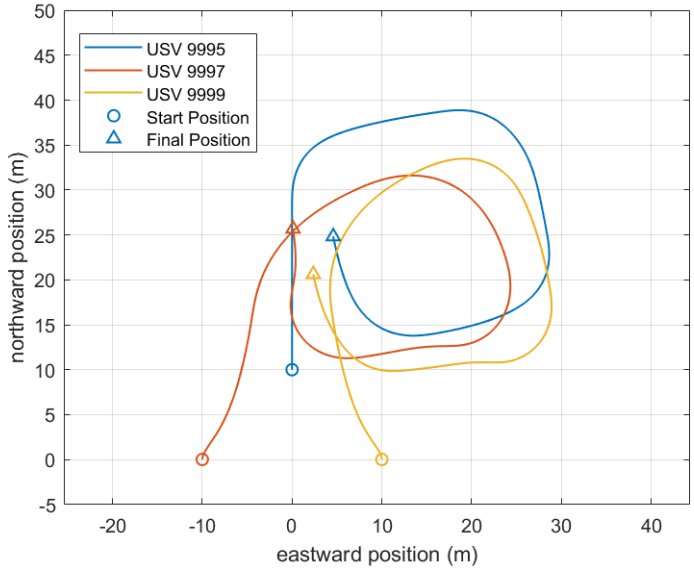


Figure 4.9. The northward vs. eastward position (m) of each simulated agent and their trajectories over 90 sec.

The trajectory that each USV took from the initial condition to its final position after 90 seconds are shown in Figure 4.9. Like the behavior of the VBAP in Section 4.1, the initial position of the USVs differ from the final positions. The VBAP algorithm does not enforce specific formation positions.

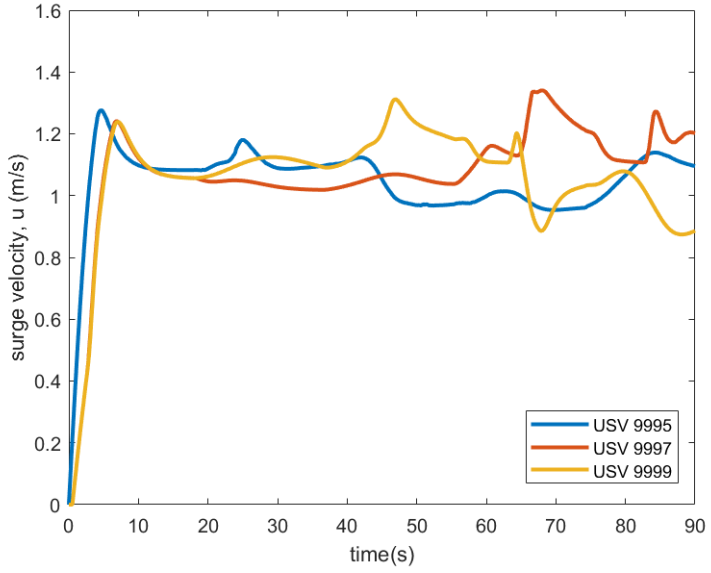


Figure 4.10. Each USV's velocity (m/s) over 90 sec.

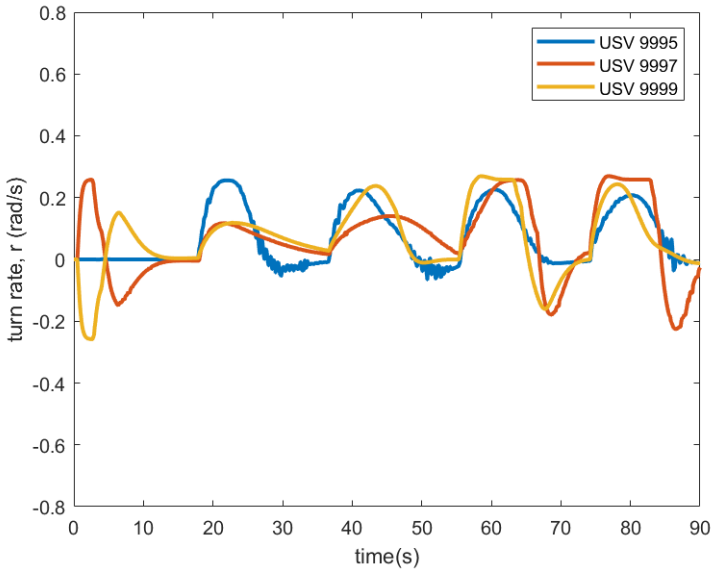


Figure 4.11. Each USV's turn rate (rad/s) over 90 sec.

The distance between each USV over time is presented in Figure 4.12. Unlike the Kingfishers using the RFC algorithm, the formation reaches the desired formation convergence distance and does so within 46 seconds. VBAP tends to outperform RFC at formation convergence from the initial conditions.

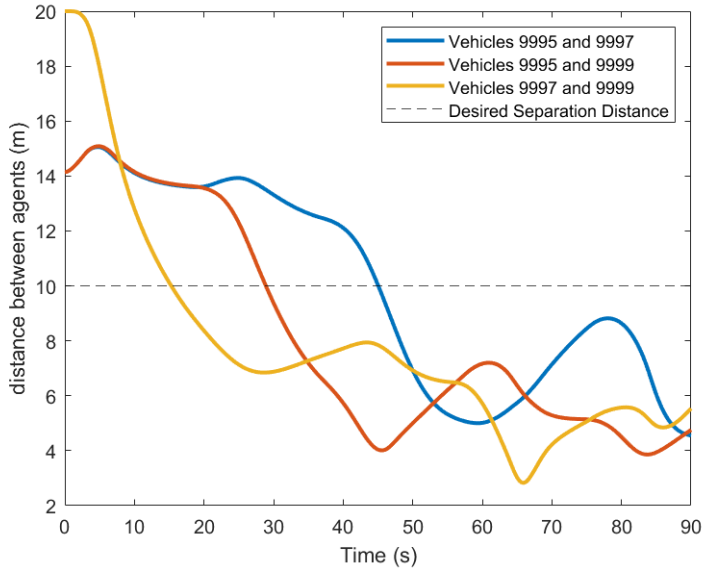


Figure 4.12. The distance (m) between each simulated agent over 90 sec. The desired separation distance is 10 m (dashed line).

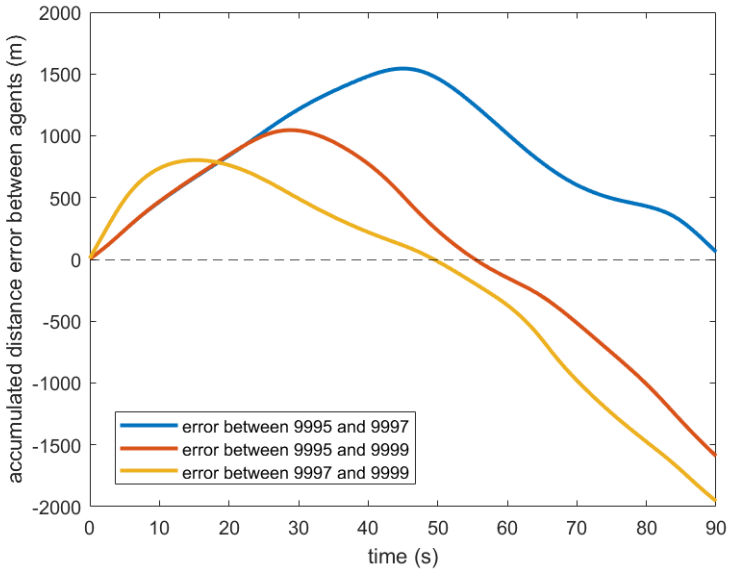


Figure 4.13. The accumulated distance error (m) between the simulated agents over 90 sec.

4.4 Three Sandwich Boats Using RFC

The fourth test consists of a homogeneous swarm of three sandwich boats utilizing the RFC algorithm. Because the sandwich boat's closed-loop steering model does not incorporate sway dynamics, simulated sandwich boats do not exhibit side slip. Because the performance of the system is much better than that of the Kingfisher, K_d is increased slightly to 0.0005 for this simulation.

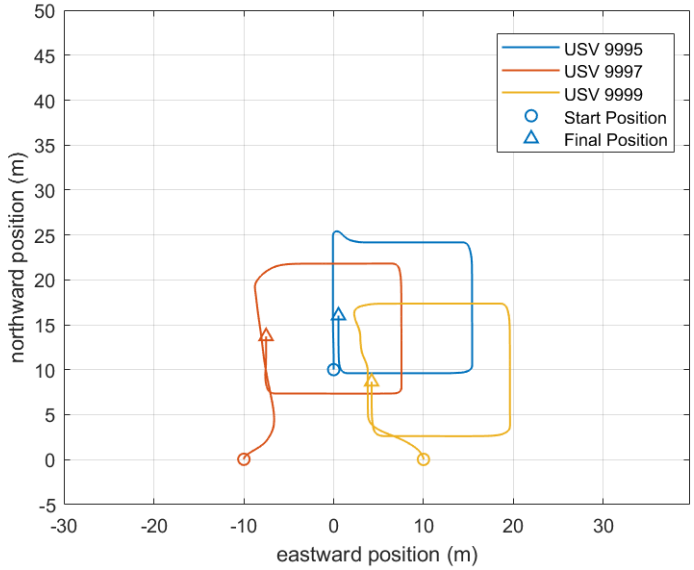


Figure 4.14. The northward vs. eastward position (m) of each simulated agent and their trajectories over 90 sec.

The trajectory that each USV took from the initial condition to its final position after 90 seconds are shown in Figure 4.14. Although there was a slight increase in the distance maintenance gain, there was no increase in oscillations (Figures 4.15 and 4.16).

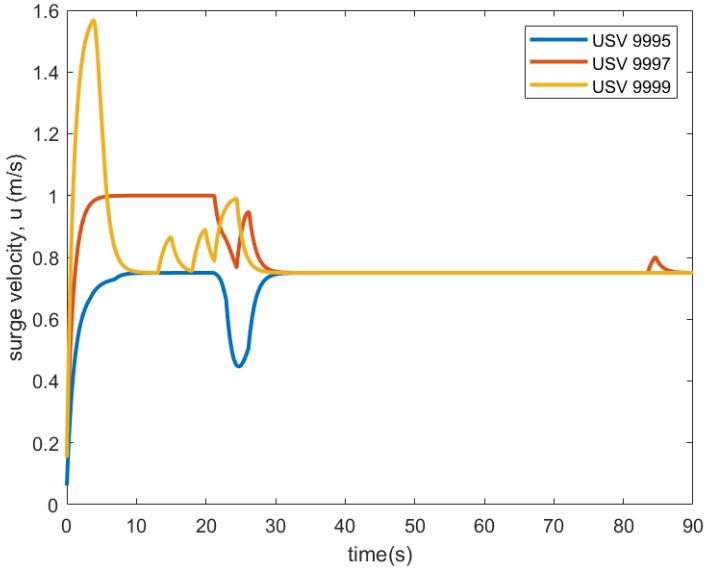


Figure 4.15. Each USV's velocity (m/s) over 90 sec.

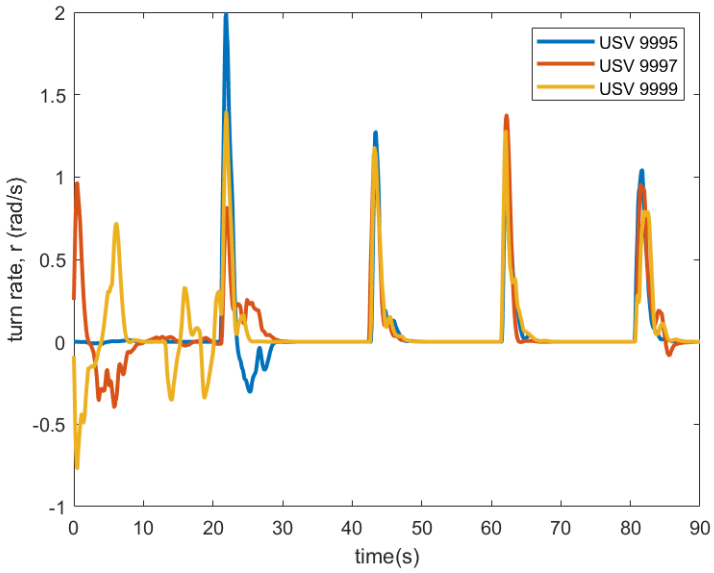


Figure 4.16. Each USV's turn rate (rad/s) over 90 sec.

The distance between each USV over time is presented in Figure 4.17. Again, the distance between USVs 9997 and 9999 remain significantly larger than that of their separation between USV 9995. The RFC algorithm seems to prevent USVs 9997 and 9999 from fully converging to 10 m. Although there's no full convergence, the RFC algorithm maintains fairly constant separation distance, even during changes in heading.

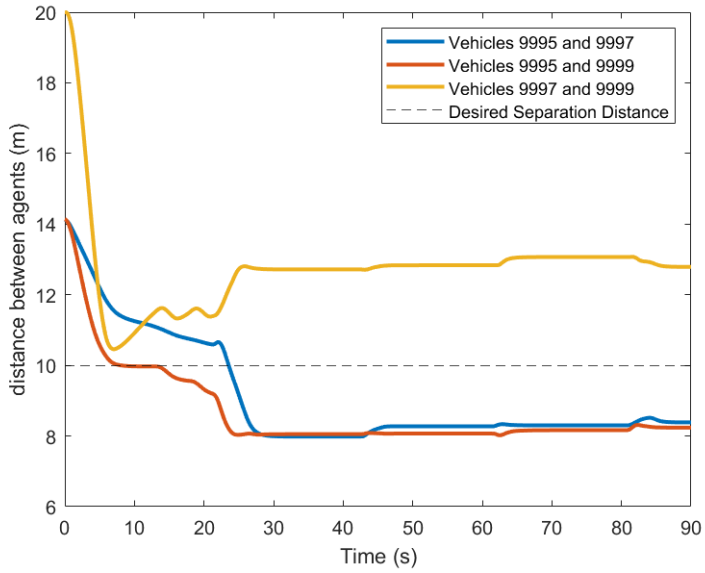


Figure 4.17. The distance (m) between each simulated agent over 90 sec. The desired separation distance is 10 m (dashed line).

4.5 Mixed Swarm Using VBAP

For the mixed swarm simulations, the parameter values from the Kingfisher VBAP algorithm were used to accommodate the slower speeds and maximum turn rate of the Kingfisher platform (Table 4.2) since the Kingfisher model is a more accurate model than the sandwich boat's. USV 9995 is initialized with Kingfisher dynamics while USV 9997 and 9999 are initialized with sandwich boat dynamics.

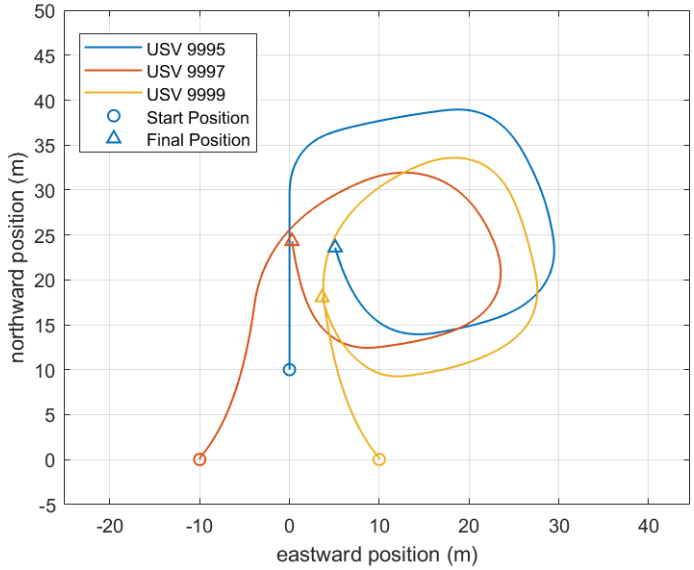


Figure 4.18. The northward vs. eastward position (m) of each simulated agent and their trajectories over 90 sec.

Figure 4.18 shows the paths that each USV took from the initial condition to its final position after 90 seconds. Figures 4.19 and 4.20 provide insight on the speed and turn rate commands to each USV. In previous tests, the sandwich boats tended to have less oscillations in heading than the Kingfisher. Because the Kingfisher is a higher-fidelity model and is more prone to corrections, the oscillations in USV 9997 and 9999 could be caused by the slight oscillation in the USV simulating Kingfisher dynamics.

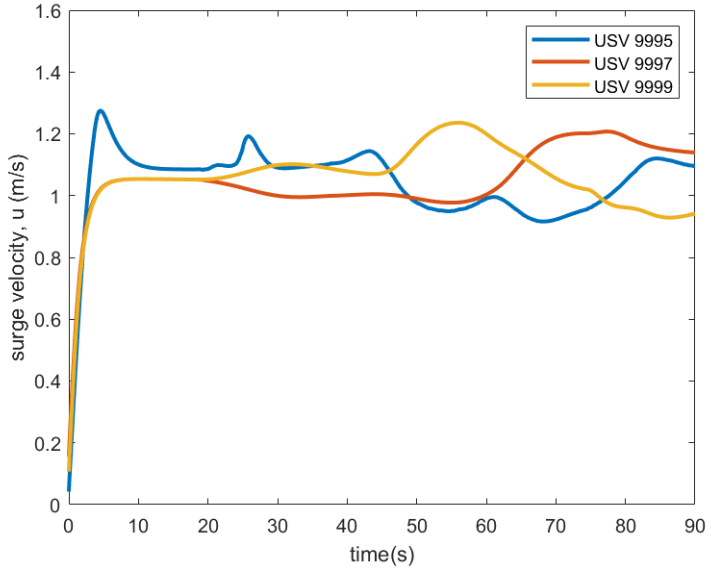


Figure 4.19. Each USV's velocity (m/s) over 90 sec.

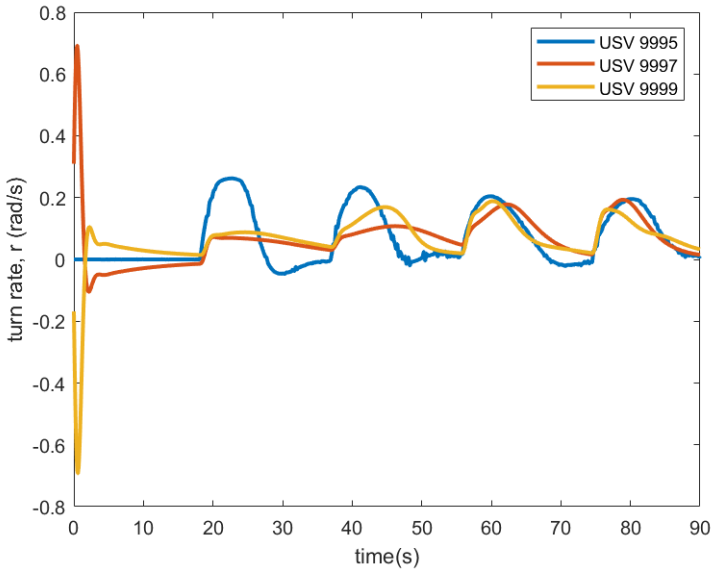


Figure 4.20. Each USV's turn rate (rad/s) over 90 sec.

The distance between each USV over time is presented in Figure 4.21. The formation reaches the desired formation convergence distance and does so within 46 seconds and has a minimum separation distance of 3.5 m.

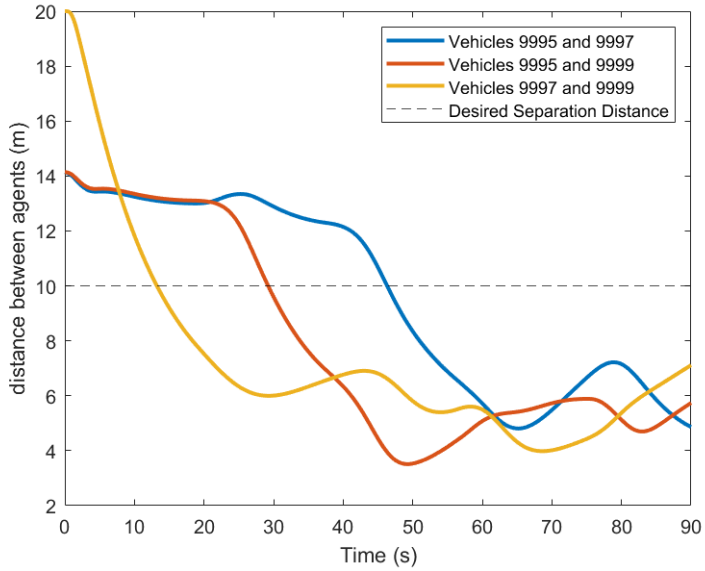


Figure 4.21. The distance (m) between each simulated agent over 90 sec. The desired separation distance is 10 m (dashed line).

4.6 Mixed Swarm Using RFC

The same dynamic setup as well as parameter values are implemented from the previous test to ensure an equal final evaluation. Figure 4.22 shows the paths that each USV took from the initial condition to its final position after 90 seconds. Although not a testable metric, the heterogeneous swarm using the RFC algorithm was the only simulation that did not complete a whole box trajectory within the allotted 90 seconds.

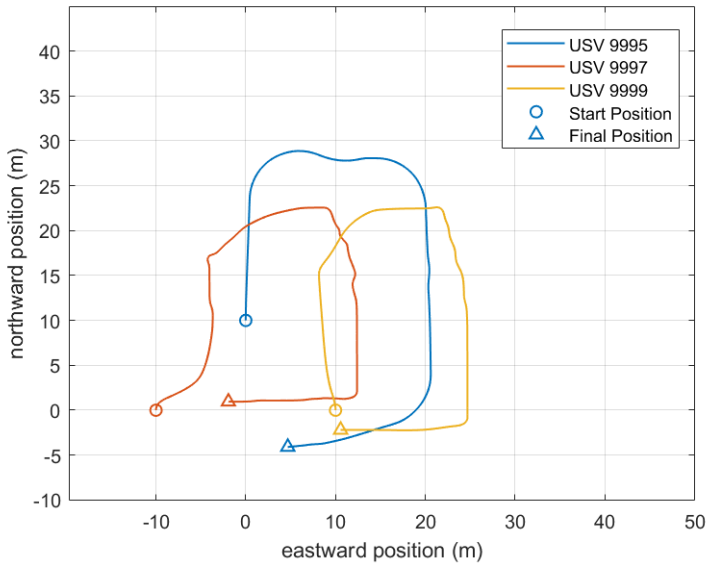


Figure 4.22. The northward vs. eastward position (m) of each simulated agent and their trajectories over 90 sec.

The mixed swarm using RFC produced state trajectories with more oscillations in both surge velocity and turn rate (Figures 4.23 and 4.24). The velocity and turn rate commands have increased oscillations possible due to the PID controllers in the Kingfisher model. The difference in dynamics and propulsion methods in each USV cause the oscillatory behavior, a characteristic to note when working with future heterogeneous swarms.

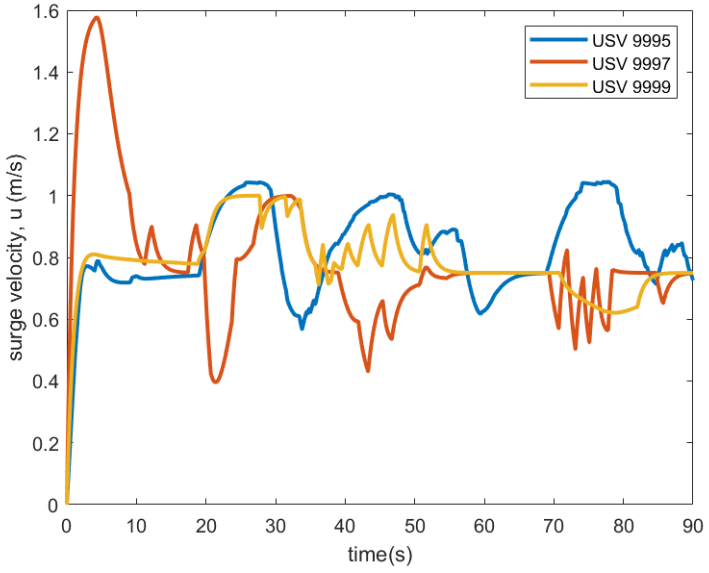


Figure 4.23. Each USV's velocity (m/s) over 90 sec.

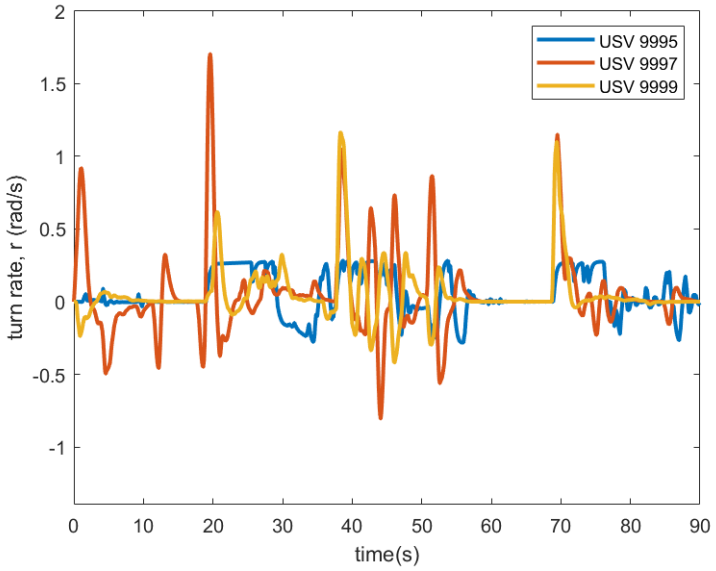


Figure 4.24. Each USV's turn rate (rad/s) over 90 sec.

The distance between each USV over time is presented in Figure 4.25. The formation does not converge and has a minimum separation distance of 3.2 m.

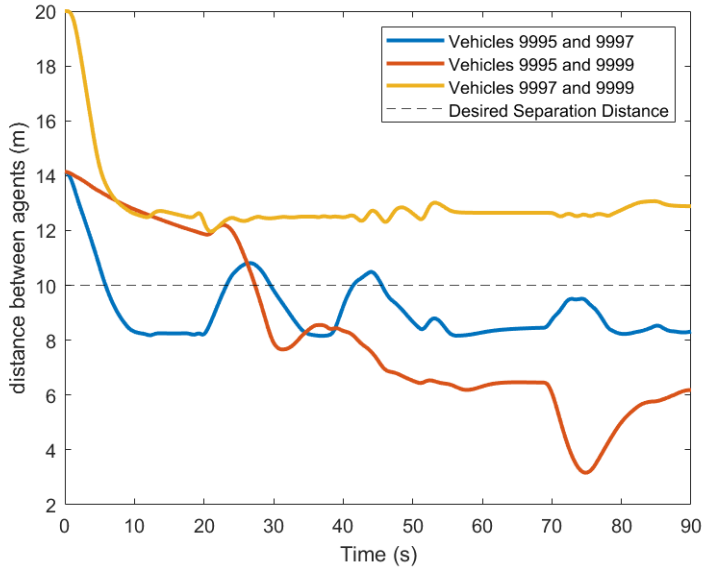


Figure 4.25. The distance (m) between each simulated agent over 90 sec. The desired separation distance is 10 m (dashed line).

4.7 Comparison of Metrics

The performance metrics described in Section 3.2 are listed along with the data from all six tests (Table 4.3). Some tests in the "Time to Converge" category did not end up closing its separation distance of 10 m therefore, never completely closing the distance to the desired formation separation value. The tests using the RFC algorithm didn't converge to the desired separation value. Typically, USV 9997 and USV 9999 tended to be the two USVs that failed to move to the desired separation distance. This could be due to the distance error allowance value implemented into the RFC algorithm.

Tests also had varying inter-vehicle distance error values (Figure 4.26). Negative error values show that the agents were closer than the desired separation distance while positive values represent a larger separation distance. No test had a collision or a minimum inter-vehicle distance of 0 m, and the mean inter-vehicle distance for all tests was 8.3 m or greater.

Table 4.3. Simulation performance metrics.

USV Type and Algorithm	Time to Converge (sec)	Inter-Vehicle Distance Error (m)	Minimum Inter-Vehicle Distance (m)	Mean Inter-Vehicle Distance (m)
Three Kingfishers with VBAP	45 sec	-3,484.6 m	2.8 m	8.7 m
Three Kingfishers with RFC	N/A	1,258.1 m	6.7 m	10.5 m
Three Sandwich Boats with VBAP	45 sec	-4,260.0 m	4.0 m	8.4 m
Three Sandwich Boats with RFC	N/A	487.5 m	8.0 m	10.2 m
Mixed Swarm with VBAP	46 sec	-4,612.0 m	3.5 m	8.3 m
Mixed Swarm with RFC	N/A	302.3 m	3.2 m	10.1 m

Of all six tests, the combination that had the least amount of accumulated separation error and had the mean separation distance closest to the specified value of 10 m was the mixed swarm using the RFC algorithm. The close second was the three sandwich boat formation using RFC. Both had low error and desirable mean inter-vehicle distance values; however, the sandwich boat and RFC test had a significantly higher minimum inter-vehicle distance value meaning that the RFC algorithm had a lower risk of collision. Figure 4.26 depicts the

total accumulated error for each test. Again, the mixed swarm using the RFC algorithm had the lowest total error at t_f. The tests using VBAP maintained a tighter formation, explaining the accumulated error being negative. The agents drifted closer together as time went on.

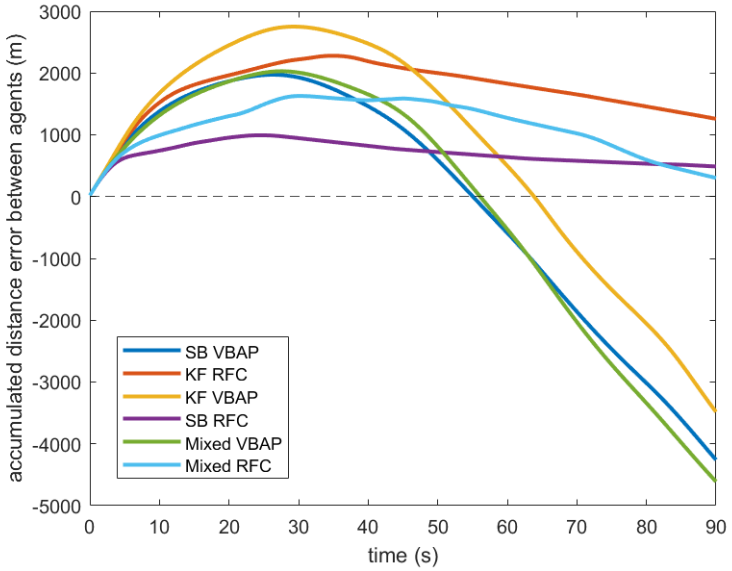


Figure 4.26. The inter-vehicle distance error accumulated over time between all six algorithms.

The sandwich boat performance was likely owing to better steering control, since this model did not incorporate realistic sway dynamics. From these tests, RFC has proven to maintain an inter-vehicle distance closer to the desired inter-agent separation value and produce less error. VBAP's strength is that the formation converges fully to the desired separation distance instead of creating and maintaining space greater than that distance.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5: Conclusion

Two different formation control algorithms studied in this thesis were found suitable for both homogeneous as well as heterogeneous swarms of USVs. Of these algorithms, the RFC algorithm performed better than VBAP when controlling a swarm of different platforms. RFC had an overall trend of not allowing the formation to fully converge to the desired separation distance and was significantly more prone to instability because of the sensitive gain, K_d; however, it produced the tests with lower cumulative error and a mean inter-vehicle distance closer to the specified value of 10 m. VBAP produced more stable results, meaning that the agents' heading oscillated much less, but produced significantly greater error. Another drawback of the VBAP algorithm, as implemented for this USV application, is its large number of constant parameters, which had to be tuned in order to achieve good performance. Once tuned, however, these values did not generalize well to different USV platforms. For both algorithms, the parameters had to be tuned to accommodate the "weakest" characteristic of each boat in the formation (i.e., slowest velocity, smallest turn rate).

For heterogeneous USV swarms, RFC is the more sound choice. In terms of naval application, the RFC algorithm had the best performance in maintaining the desired separation distance between agents. The reduction of the risk of collision and destruction of hardware is crucial in performing tasks such as search and rescue, ISR missions, and force protection operations out at sea. The goal was to mitigate risk and increase the safety of sailors out at sea. An algorithm that is reliable in maintaining desired inter-vehicle distances is integral to naval operation.

5.1 Future Work

Further exploration into the algorithms, specifically the effects of the gain values on the algorithm outputs, need to be tested. The sensitivity of the distance maintenance gain from the RFC algorithm specifically needs to be investigated. Another area of work is to increase the amount of agents in the formation. Environmental forces can be added to the simulated

boats to see how each algorithm performs under these disturbances. Stochastic Formation Control, another algorithm created by Dr. Mwaffo et al. [14] at the United States Naval Academy, is another algorithm that can be implemented and evaluated in future testing. In order to validate simulation results, these algorithms can be implemented on actual USVs for in-water testing. The performance of the algorithms on larger platforms can be assessed. The transition from software to hardware testing would be beneficial for examining the performance of each algorithm in the face of external environmental factors.

List of References

[1] R. O'Rourke, "Navy large unmanned surface and undersea vehicles: background and issues for Congress," CRS Report No. R45757, Washington DC, USA, 2023 [Online]. Available: https://crsreports.congress.gov/product/pdf/R/R45757

[2] Z. Liu, Y. Zhang, X. Yu, and C. Yuan, "Unmanned surface vehicles: an overview of developments and challenges," *Annual Reviews in Control*, vol. 41, May 2016 [Online]. Available: https://doi.org/10.1016/j.arcontrol.2016.04.018

[3] T. E. Witham, "Commercial tugs escort the nuclear-powered strategic missile submarine USS TENNESSEE (SSBN 734) as the vessel departs from Kings Bay." National Archives Catalog, Mar. 24, 1990 [Online]. Available: https://catalog.archives.gov/id/6456178

[4] J. L. Escobosa, "Tug boats escort the aircraft carrier USS John C. Stennis (CVN 74)," Wikimedia Commons, June 2009 [Online]. Available: https://commons.wikimedia.org/wiki/File:US_Navy_090629-N-9928E-366_Tug_boats_escort_the_aircraft_carrier_USS_John_C._Stennis_%28CVN_74%29_as_the_ship_pulls_into_Everett,_Wash.,_for_a_scheduled_port_visit_as_the_ship_prepares_to_pick_up_family_and_friends_for_a_Tiger_Crui.jpg#filehistory

[5] Embention, "USV (unmanned surface vehicle), applications and advantages," Sep. 2015 [Online]. Available: https://www.embention.com/news/usv-unmanned-surface-vehicle-applications-and-advantages/

[6] T. Sang, J. Xiao, J. Xiong, H. Xia, and Z. Wang, "Path planning method of unmanned surface vehicles formation based on improved A* algorithm," *Journal of Marine Science and Engineering*, vol. 11, no. 1, 2023 [Online]. Available: https://doi.org/10.3390/jmse11010176

[7] E. Kagan, N. Shvalb, and I. Ben-Gal, *Autonomous Mobile Robots and Multi-Robot Systems: Motion-Planning, Communication, and Swarming*. Hoboken, NJ, USA: John Wiley & Sons [Online]. Available: https://www.researchgate.net/publication/341203824_Autonomous_mobile_robots_and_multi-robot_systems_Motion-planning_communication_and_swarming

[8] G. Tan, H. Sun, L. Du, J. Zhuang, J. Zou, and L. Wan, "Coordinated control of the heterogeneous unmanned surface vehicle swarm based on the distributed null-space-based behavioral approach," *Ocean Engineering*, vol. 266, p. 112928, Dec. 2022 [Online]. Available: https://doi.org/10.1016/j.oceaneng.2022.112928

[9] E. Fiorelli, N. E. Leonard, P. Bhatta, D. A. Paley, R. Bachmayer, and D. M. Fratantoni, "Multi-AUV control and adaptive sampling in Monterey Bay," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 4, pp. 935–948, Oct. 2006 [Online]. Available: https://doi.org/10.1109/JOE.2006.880429

[10] P. Frontera, M. Feemster, and J. Nguyen, "Application of rigid formation control to marine vessels," *Ocean Engineering*, vol. 277, p. 114274, June 2023 [Online]. Available: https://doi.org/10.1016/j.oceaneng.2023.114274

[11] X. Cai and M. de Queiroz, "Multi-agent formation maintenance and target tracking," in *2013 American Control Conference*, June 2013, pp. 2521–2526 [Online]. Available: https://doi.org/10.1109/ACC.2013.6580213

[12] N. A. Manzini, "USV path planning using potential field model," M.S. thesis, Dept. of Natl. Sec. Aff., NPS, Monterey, CA, USA, 2017 [Online]. Available: https://hdl.handle.net/10945/56152

[13] Matthew Heubach J., "Use of artificial fiducial markers for USV swarm coordination," M.S. thesis, Dept. of Natl. Sec. Aff., NPS, Monterey, CA, USA, 2022 [Online]. Available: https://hdl.handle.net/10945/71478

[14] V. Mwaffo, P. DeLellis, and J. S. Humbert, "Formation control of stochastic multi-vehicle systems," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 6, pp. 2505–2516, Nov. 2021 [Online]. Available: https://doi.org/10.1109/TCST.2020.3047422

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California