

NETWORK OF UNMANNED SURFACE VEHICLES: DESIGN AND APPLICATION TO  
TARGET TRACKING

By

Chandler J. Panetta

A THESIS

Submitted to  
Michigan State University  
in partial fulfillment of the requirements  
for the degree of

Electrical and Computer Engineering – Master of Science

2021

## ABSTRACT

### NETWORK OF UNMANNED SURFACE VEHICLES: DESIGN AND APPLICATION TO TARGET TRACKING

By

Chandler J. Panetta

Unmanned surface vehicles (USVs) have gained increased attention in environmental monitoring, navigation assistance, search-and-rescue, and other fields over the past three decades. USVs provide an effective platform for mobile sensing applications and offer flexibility in specific capabilities. This work presents a network of compact USVs that are capable of deploying underwater sensors using an automated winch. The small, maneuverable nature of each USV is ideal for inland bodies of water, and the relatively large payload capacity allows for surveys that last multiple hours.

Motivated by the application of acoustic telemetry-based fish movement tracking, this work focuses on using a network of USVs to localize an underwater acoustic tag by exploiting the time-difference-of-arrival (TDOA) of the emitted signal. A distributed TDOA-based particle filter (PF) algorithm is proposed for localizing a moving target modeled by a discrete-time correlated random walk (DCRW). Furthermore, an online model learning method is explored, where target position estimates are used to update the unknown probability distributions of the target's movement model. Through numerical simulations, the distributed PF is shown to result in effective estimation of the target position when a node is connected to a network that collectively has an adequate number of TDOA measurements. Additionally, the efficacy of online model learning in handling model uncertainties is demonstrated in simulation studies.

TDOA-based localization algorithms are further validated in field experiments using a network of four USVs carrying acoustic telemetry equipment. In particular, TDOA and GPS data are collected and used to assess the target estimation performance for the distributed TDOA-based PF and a distributed TDOA-based extended Kalman filter (EKF) under different settings for the network topology.

To my family and friends, whose love, support, and guidance have shaped me into the person I am today.

## ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation (IIS 1734272, IIS 1848945, ECCS 2030556).

First and foremost, I would like to thank my research advisor, Dr. Xiaobo Tan, for his trust, support, and mentoring throughout the latter part of my time at MSU. Dr. Tan positively motivated me to push the limits of my own academic and creative ability, and I will be forever grateful for the positive impact that working in the Smart Microsystems Lab has had on my life. I would also like to acknowledge the United States Air Force, Air Force Institute of Technology, and the cadre at AFROTC Detachment 380 for the opportunities I had during my undergraduate and graduate studies to advance my education. The leadership lessons, character development, and academic opportunities that I have experienced as a member of these organizations is invaluable to me.

I extend thanks to my academic committee members at Michigan State University: Dr. Shaunak Bopardikar and Dr. Vaibhav Srivastava. They have inspired me to continue to pursue the field of multi-agent systems, and I have thoroughly enjoyed attending lectures by both of these great professors. I would also like to thank Dr. Osama Ennasr, who collaborated with me on this research. His assistance, guidance, and outreach made this project possible. Additionally, I would like to thank Dr. Christopher Holbrook and Dr. Darryl Hondorp from the U.S. Geological Survey for their help running acoustic telemetry experiments with the USVs on Higgins Lake.

I would like to acknowledge Danton Thon who was instrumental in the mechanical design, development, and construction of the vehicles used in this research. His ingenuity and craftsmanship consistently surpassed all expectation, and I am grateful to have worked alongside him. Additionally, I am thankful to the past and present members of the SML who directly helped with this project, including Hanish Mehta, Robert Billette, Paul Schulman, Demetris Coleman, Eric Gaskell, and John Thon, as well as all of the undergraduate senior design students who have helped along the way. Besides those listed, I am grateful for the opportunity to have worked with all the members of SML, and I can't thank them enough for all the various ways they have helped me become a better

student and engineer.

Lastly, I am extremely grateful for the love and support of my family and friends who have helped me along the way. In particular, my mom for her guidance navigating my academic and career goals as well as letting me set up shop at home during COVID-19 stay-at-home measures, my dad for his patience and help running all of the field experiments discussed in later sections, and my brothers and roommates for being a shoulder to lean on during my academic journey. Additionally, I send a special thanks to my girlfriend for the motivation she has given me and for all her help editing my written works. I would not be where I am today without all of them.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the U.S. Government.

## TABLE OF CONTENTS

LIST OF TABLES . . . . .	ix
LIST OF FIGURES . . . . .	x
CHAPTER 1 INTRODUCTION . . . . .	1
1.1 Motivation . . . . .	1
1.2 Research Objectives . . . . .	3
1.3 Summary of Contributions . . . . .	3
1.3.1 Design of Unmanned Surface Vehicle Network . . . . .	4
1.3.2 Distributed Particle Filter for TDOA-based Localization . . . . .	4
1.3.3 Experimental Validation . . . . .	5
1.4 Thesis Organization . . . . .	6
CHAPTER 2 DESIGN OF UNMANNED SURFACE VEHICLE (USV) . . . . .	7
2.1 USV Hull Design . . . . .	7
2.2 Component Selection . . . . .	8
2.2.1 PixHawk Controller and ArduPilot . . . . .	9
2.2.2 Ground Control and Vehicle Software . . . . .	9
2.3 First-Generation USV . . . . .	10
2.3.1 Tag Detection Performance. . . . .	11
2.4 Second-Generation USV . . . . .	12
2.4.1 Automated Winch . . . . .	13
2.4.2 Sensor Hub . . . . .	14
2.4.3 Network of USVs . . . . .	15
2.5 Future Work . . . . .	16
CHAPTER 3 DISTRIBUTED TDOA-BASED LOCALIZATION . . . . .	17
3.1 Problem Setup . . . . .	18
3.2 Distributed Particle Filter Algorithm with Online Model Learning . . . . .	19
3.2.1 Distributed Particle Filter Algorithm . . . . .	20
3.2.2 Online Learning of Target Model Parameters . . . . .	22
3.3 Simulation Results . . . . .	23
3.3.1 Distributed Particle Filter . . . . .	23
3.3.2 Online Model Learning . . . . .	25
CHAPTER 4 FIELD EXPERIMENTAL RESULTS ON TDOA-BASED TARGET LO- CALIZATION . . . . .	28
4.1 Experiment Setup . . . . .	28
4.1.1 Waypoint Tracking . . . . .	28
4.1.2 LAN Configuration . . . . .	30
4.2 Coordinate Transform . . . . .	31
4.3 Distributed TDOA Localization - Passively Drifting Target . . . . .	33

4.3.1	Data Analysis . . . . .	34
4.3.2	Distributed Localization with a Fully Connected Topology . . . . .	36
4.3.3	Distributed Localization under Other Network Topologies . . . . .	39
4.4	Distributed PF Localization - DCRW Target . . . . .	44
CHAPTER 5 SUMMARY AND FUTURE WORK . . . . .		47
5.1	Summary . . . . .	47
5.2	Future Work . . . . .	48
BIBLIOGRAPHY . . . . .		49

## LIST OF TABLES

Table 2.1: Tag detection efficiency for stationary receivers ("Stationary") and USV-mounted receiver with two configurations of the receiver orientation ("Vertical" and "Horizontal"). . . . .	12
Table 4.1: Approximate weather conditions for the stationary target experiment. . . . .	33

## LIST OF FIGURES

Figure 1.1: Acoustic telemetry tag (a) and surgical implantation in fish (b) [25]. . . . .	2
Figure 2.1: iXBlue DriX USV [21, 31]. . . . .	8
Figure 2.2: ClearPath Heron USV [20]. . . . .	8
Figure 2.3: Picture of first-generation USV developed in the MSU Smart Microsystems Lab.	10
Figure 2.4: Mounted acoustic receiver. . . . .	10
Figure 2.5: Map of the acoustic receiver range test for the mounted and stationary acoustic receivers. The blue arrow and line show the USV’s path, the red pin marks the target, and the green pins mark the stationary receivers. . . . .	11
Figure 2.6: Automated winch and sensor hub unit: (A) Ethernet slip ring and housing, (B) Kevlar reinforced Ethernet tether cable, (C) winch motor and gearbox, (D) Vemco VR2C acoustic receiver, and (E) sensor hub unit. . . . .	13
Figure 2.7: A collection of USVs used for TDOA tests. (A) Waterproof Pelican case housing USV electronics, (B) automated winch, (C) sensor hub, and (D) Vemco VR2C acoustic receiver. Note: the USV to the far right does not incorporate an automated winch and sensor hub unit. . . . .	14
Figure 2.8: Deployed USV network. (T) represents the USV carrying the acoustic tag, and (R) represents the receiving nodes. . . . .	15
Figure 2.9: An USV prototype developed by the Smart Microsystems Lab based on an alternative design concept. . . . .	16
Figure 2.10: Demonstration of the self-righting capabilities of the USV based on the alternative design. . . . .	16
Figure 3.1: Simulation setup. Here the diamonds represent the nodes’ locations, the red circle represents the moving target, the green, blue, and orange circles represent the estimates by nodes 1, 4, and 5, respectively, and lines connecting the nodes represent communication links. . . . .	23
Figure 3.2: Average norm errors for nodes 1 and 5 at each time-step for the setting of $\mathcal{N}_1 = \{2, 3\}$ , $\mathcal{N}_4 = \{5\}$ , and $\mathcal{N}_5 = \{4\}$ . Note: $\mathcal{N}_5$ has an insufficient number of TDOA measurements. . . . .	24

Figure 3.3:	Average norm errors for node 1 and node 5 at each time-step for the setting of $\mathcal{N}_1 = \{2, 3, 4\}$ , $\mathcal{N}_4 = \{1, 5\}$ , and $\mathcal{N}_5 = \{4\}$ . The PF estimate of $\mathcal{N}_5$ converges despite having an insufficient number of TDOA measurements. . . . .	25
Figure 3.4:	Average norm error, $E_4(k)$ , with and without the consensus step. . . . .	26
Figure 3.5:	Cumulative norm error, $\sum_1^k E_5(k)$ , for each movement model compared to the distance the target moved from the center of the nodes. The center of the nodes is at $(0, 0)$ . . . . .	27
Figure 3.6:	PDF estimates of the target's movement model compared to the true PDFs. . . . .	27
Figure 4.1:	USV waypoint tracking after controller tuning. Slight steady-state oscillations along the trajectory can be observed. . . . .	30
Figure 4.2:	Screen view of the host computer monitoring each device on the network through SSH connection. . . . .	31
Figure 4.3:	USVs deployed on Crockery Lake, MI. The colored pins show the initial location of the receiving USVs for the passive target experiment. . . . .	33
Figure 4.4:	75 network TOA measurements plotted on a linear x-axis. The vertical lines indicate the TOA of a ping given by the number of seconds after 00:00:00 UTC. . . . .	34
Figure 4.5:	Initial formation of the USV network. Diamonds represent receiving nodes, and the circle represents the target. . . . .	35
Figure 4.6:	Final formation of the USV network. The solid line shows the drift trajectory of the target USV. . . . .	35
Figure 4.7:	Fully Connected Topology – node EKF estimate's trajectories (a) and accompanying estimation errors (b) as well as node PF estimate's trajectories (c) and accompanying estimation errors (d). In figures (a) and (c) the diamonds represent sensing nodes, the lines connecting the diamonds represent communication links, the solid trajectory shows the target's true trajectory, and each dashed trajectory shows the node estimate's trajectory. . . . .	38
Figure 4.8:	Three connected topologies where only one node is connected to two neighbors. . . . .	39
Figure 4.9:	Topology A – node EKF estimate's trajectories (a) and accompanying estimation errors (b) as well as node PF estimate's trajectories (c) and accompanying estimation errors (d). In figures (a) and (c) the diamonds represent sensing nodes, the lines connecting the diamonds represent communication links, the solid trajectory shows the target's true trajectory, and each dashed trajectory shows the node estimate's trajectory. . . . .	40

Figure 4.10: Topology B – node EKF estimate’s trajectories (a) and accompanying estimation errors (b) as well as node PF estimate’s trajectories (c) and accompanying estimation errors (d). In figures (a) and (c) the diamonds represent sensing nodes, the lines connecting the diamonds represent communication links, the solid trajectory shows the target’s true trajectory, and each dashed trajectory shows the node estimate’s trajectory. . . . . 41

Figure 4.11: Topology C – node EKF estimate’s trajectories (a) and accompanying estimation errors (b) as well as node PF estimate’s trajectories (c) and accompanying estimation errors (d). In figures (a) and (c) the diamonds represent sensing nodes, the lines connecting the diamonds represent communication links, the solid trajectory shows the target’s true trajectory, and each dashed trajectory shows the node estimate’s trajectory. . . . . 42

Figure 4.12: Topology B – Target and node EKF estimate’s trajectories (a) and corresponding node error (b) with the addition of the virtual node (VN). . . . . 43

Figure 4.13: Topology C – Target and node EKF estimate’s trajectories (a) and corresponding node error (b) with the addition of the virtual node (VN). . . . . 44

Figure 4.14: QGroundControl window showing the DCRW GPS waypoint trajectory loaded to the target USV. Circled numbers indicate sequential GPS waypoints. . . . . 46

Figure 4.15: Distributed PF estimate trajectories for a target modeled by a DCRW. The solid trajectory shows the true path of the target, and the dashed trajectory shows each node’s PF estimate. . . . . 46

# CHAPTER 1

## INTRODUCTION

Approximately seventy-one percent of the Earth's surface is covered by water, and a relatively small portion of that percent has been studied. Exploring the Earth's waterways has historically proven to be a tedious, expensive, and often dangerous task [13]. These bodies of water, whether it be oceans, lakes, or rivers, are in a constant state of vicissitude. With the recent availability of affordable and compact GPS units, long-range telemetry systems, and high-powered single-board computers, unmanned surface vehicles (USVs) have gained increased attention for remote sensing applications [30].

In the past two decades, the utility of USVs has been demonstrated by hobbyists, academic institutions, private companies, and government agencies [7, 39]. Flexibility in physical construction and specific capabilities allows for a wide range of missions including search-and-rescue, environmental monitoring, bathymetric mapping, and defense applications [27, 30]. This thesis focuses on the construction of a network of USVs capable of traversing inland bodies of water and deploying sensing payloads to a desired depth. To put the device in context, we focus on the specific application of localizing an underwater acoustic tag using multiple sensing receivers mounted on USVs.

### 1.1 Motivation

The Great Lakes Region of North America holds enough freshwater to cover the surface of the continental U.S. in three meters of water and accounts for over twenty-percent of the world's freshwater [17, 41]. The ecosystems of these lakes, rivers, and streams are under constant pressure from human and environmental impacts including landscape changes, chemical imbalances, climate change, introduction of invasion species, over-fishing, and many other factors. One of the ways researchers monitor the health of freshwater ecosystems is to track the movement patterns of fish in their natural landscape [41].

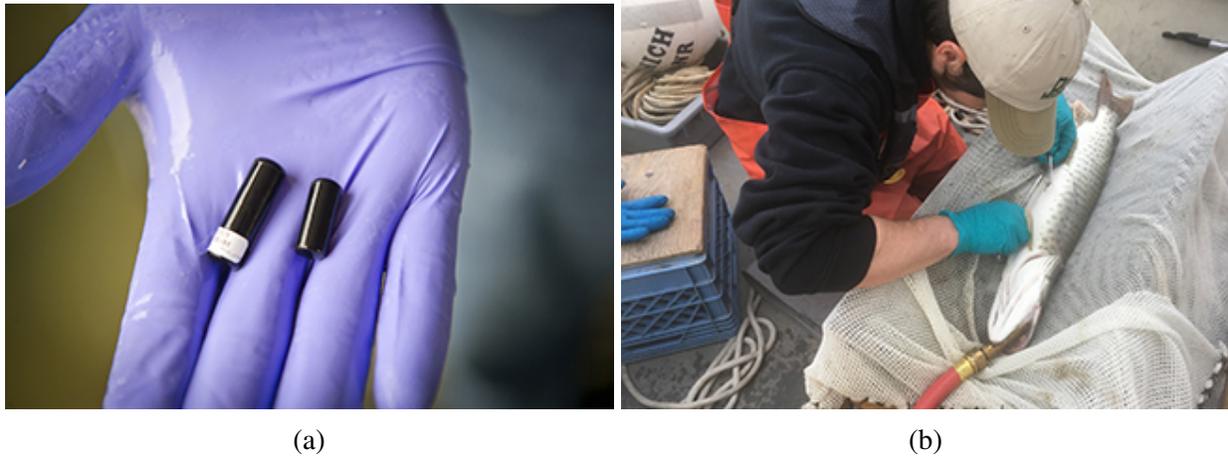


Figure 1.1: Acoustic telemetry tag (a) and surgical implantation in fish (b) [25].

Traditionally, fish movement patterns are recorded using a mark-and-recapture technique. This method involves encoding a fish with a microscopic wire and clipping one of its fins as an identifier to anglers. When the fish is captured, anglers are encouraged to self-report the fish's location. To avoid complications and inconsistencies with this method, researchers have started using acoustic telemetry to locate tagged fish. This method involves surgically implanting an acoustic tag into the fish's body cavity, as shown in Figure 1.1 [25], then measuring acoustic pings from the tag using an array of deployed sensing receivers. Due to the large scale success of acoustic telemetry systems, the Great Lakes Acoustic Telemetry Observation System (GLATOS) was established by the Great Lakes Fishery Commission [19].

GLATOS is a collaborative, bi-national effort put forth by researchers to develop an acoustic telemetry system to monitor fish in the Great Lakes. Projects and equipment are deployed into lakes to detect fish that have been surgically implanted with an acoustic tag. The receiver network is shared amongst researchers to encourage collaboration and outreach by federal, provincial, tribal, and university organizations [19]. Fine-scale positioning of acoustic tags underwater can be achieved using an array of closely located receivers and is able to give insight to fish movement behaviour in specific environments [19, 42, 43]. Using an array of stationary receivers requires heavy equipment, large-scale deployment, and substantial manpower [42]. Mounting each receiver

to an unmanned mobile robot for fish localization and tracking is a promising way to reduce the cost and resources required to capture fine-scale acoustic telemetry data [33, 37].

## 1.2 Research Objectives

This research focuses on designing a network of USVs with the ability to carry and deploy aquatic sensing equipment semi-autonomously. The platform must provide an open-architecture capable of carrying a wide array of sensors. It is desired that each USV be designed in such a way that it could be deployed by a single person and be able to conduct surveys extending multiple hours without needing to be recharged. Creating a cost-effective mechanical design that is easy to replicate out of readily available components allows for quick production. An individual outside of the engineering landscape could also build and utilize it in independent research. To evaluate the utility of the USV system for acoustic telemetry applications, a field experiment demonstrating the capacity to localize a moving underwater target using time-difference-of-arrival (TDOA) based algorithms will be investigated.

### **Summary of Objectives:**

- Construct a network of (four) unmanned surface vehicles
- Propose and verify a distributed TDOA-based particle filter localization algorithm using numerical simulation
- Conduct field experiments using the USV network and acoustic telemetry equipment
- Evaluate the distributed TDOA-based particle filter (PF) and a distributed extended Kalman filter (EKF) for underwater localization of a moving target based on data from the field experiments

## 1.3 Summary of Contributions

The contribution of this work highlights the construction of an USV network capable of deploying underwater sensing payloads to a desired depth, an investigation of distributed TDOA-based

localization of a moving target, and evaluation of TDOA-based localization algorithms with data from field experiments involving the USV network.

### **1.3.1 Design of Unmanned Surface Vehicle Network**

Initially, the hull design of the USV concentrated on navigating shallow rivers, lakes, and streams. Previous versions of the USV focused on validating open-source hardware and software components to simplify repeated construction outside of the academic landscape. The first iteration of the USV embraced a catamaran hull with two fixed waterproof electric motors (thrusters) with differential steering, a commercially purchased controller, and a static acoustic receiver mount on the underside of the hull. This design was used to test receiver detection efficiency and gain insight into the limitations of the acoustic telemetry system on a mobile robot. A second version of the USV embraced a single hull design that was self-righting, waterproof, and incorporated vectored thrust steering.

Motivated by the pitfalls of the first two designs, a third version of the USV was developed. This iteration again adopted the catamaran hull, utilized vectored thrust steering, and included the addition of an automated winch. This automated winch allows sensing electronics to be deployed to various depths and transfer data via a reinforced Ethernet tether cable. Additional aspects of notable improvements to the USV design are discussed in later sections.

### **1.3.2 Distributed Particle Filter for TDOA-based Localization**

Previous work utilizing TDOA-based localization algorithms has considered both centralized and distributed approaches [45]. A minimum of two (three) TDOA measurements are required to localize a target in two (three) dimensions. Potential issues for the centralized approach, including single node failure and high communication traffic at the reference node, may arise when implementing this method. These issues are addressed when using a distributed TDOA-based algorithm where each node assumes itself as the reference node, gathers time-of-arrival (TOA) measurements from

its neighbors to derive TDOA measurements, estimates the target’s location based on available measurements, and uses information available from its neighbors to update the estimate [36].

In this thesis, a distributed particle filter (PF) for TDOA-based localization is proposed. In addition, online model learning is utilized by the PF to estimate the unknown parameters of the target dynamics. To put the study in context, we focus on the case where the target is modeled by a discrete-time correlated random walk (DCRW). Random walk models have been widely used for modeling animal movement behavior [2, 22], interactions with animals and their environment [16], cell movement [15], and other applications [9, 12]. This method is compared with a non-adaptive approach through numerical simulation to show the efficacy of incorporating online model learning in improving target localization performance.

### **1.3.3 Experimental Validation**

The third and final major contribution of this work details the open-water field experiments conducted in November 2020 to explore the feasibility of using a USV network carrying acoustic telemetry equipment for fine-scale positioning of an underwater tag. These field experiments were conducted on Crockery Lake in Chester Township, Michigan and focused on two objectives: (1) to verify the simulation results of the distributed TDOA-based PF algorithm and the distributed extended Kalman filter (EKF) [34, 35, 38] for a passively drifting underwater target; (2) verify the PF localization algorithm for a target modeled by a discrete-time correlated random walk (DCRW).

This field experiment was conducted using a network of four semi-autonomous USVs: three carrying a Vemco VR2C acoustic receiver and one carrying a Vemco Range Test Tag. These USVs were deployed on open-water, and time-of-arrival (TOA) measurements were gathered at each of the receiving nodes. An attempt was made to calculate real-time target estimates during the test, but the majority of data analysis was accomplished through post-processing. Data from the field experiments were used to validate key findings in [10, 36], and they revealed opportunities for future work to improve real-time target localization and tracking.

## 1.4 Thesis Organization

The remainder of this thesis is as follows. In Chapter 2, the design, construction, and validation of the USV for remote sensing applications is discussed in detail. This chapter highlights notable improvements to the USV, including the automated winch, the underwater sensor hub, and integration of an acoustic receiver. Chapter 3 outlines the proposed method for a distributed TDOA-based particle (PF) estimator for localizing a moving target. In particular, a discrete-time correlated random walk (DCRW) is considered for the target movement. Data collected from the PF's estimate is used to recursively update the estimator's movement model, and localization performance is verified through numerical simulation. Lastly, Chapter 4 outlines the open-water field tests conducted with a network of four USVs. Data collected from the field tests was analyzed through post-processing, and key findings from the simulation results were evaluated using the experimental data.

## CHAPTER 2

### DESIGN OF UNMANNED SURFACE VEHICLE (USV)

The physical specifications for a USV are nearly endless. The size, shape, payload capacity, and on-board electronics are largely dependent on the specific use for the device and the waterway in which it navigates. One of the first documented USVs built for research purposes, named ARTEMIS, was developed by MIT Sea Grant in 1993 for unmanned bathymetric sampling. Since then, there have been a multitude of designs for defense, search-and-rescue, remote surveying, cooperative control, and many other implementations [27].

The design of the USV in this work centers around improving the methods for mobile sensing in the Great Lakes region, and more specifically, collecting fine-scale fish movement data by localizing a tagged fish using acoustic telemetry data. The USV must adopt a highly maneuverable body with minimal draft and provide a cost-effective platform for acoustic telemetry. In order to localize a target in real-time using TDOA-based algorithms, a minimum of three USVs for two-dimensions and four USVs for three-dimensions are necessary. Thus, the design must be repeatable and be constructed from commercially available components to simplify manufacturing.

#### 2.1 USV Hull Design

Much of the previous development of small USVs focused on deploying the devices in oceanic environments. These devices were built to withstand the corrosiveness of saltwater and often experience harsher conditions than the Great Lakes region. Past development involved primarily on two different hull designs: single hull designs and double hull designs. Single hull designs provide a robust structure that can easily be enclosed for complete waterproof protection. Notable examples include the US Navy's MUSCL USV [32], Liquid Robot's Wave Glider [28], and the iXBlue DriX USV shown in Figure 2.1 [21]. Despite these advantages, single hull USVs typically require advanced composite construction, have narrow deck-space, and need to have a high unloaded draft to remain stable in the water.



Figure 2.1: iXBlue DriX USV [21, 31].



Figure 2.2: ClearPath Heron USV [20].

A double hull, or catamaran, USV addresses the stability issue by providing two separate hulls with a deck that spans the middle. This design allows the USV to remain stable even when it is unloaded, and it is able to maintain an extremely low draft. Additionally, the elevated and relatively large deck provides ample space for payloads, electronics, and USV drive systems. Examples of catamaran USVs include the ClearPath Heron (shown in Figure 2.2) [20], the SeaWASP [14], and MIT's AutoCat [24]. After weighing design aspects for each configuration, it was determined that a catamaran style hull would be advantageous for this application. This design allows for a variety of payloads, can operate in both shallow depths and open water, and provides a construction that can be easily repeated. The size would be constrained to be manageable by a single user, and the material selection would minimize the amount of special fabrication required.

## 2.2 Component Selection

One of the main objectives in the development of this USV was to create an open-architecture that is easily repeatable and could be built outside of the academic landscape. Recently, there has been an abundance of affordable, compact, and accurate robotic vehicle components that provide immense resources and community support [3, 5].

### **2.2.1 PixHawk Controller and ArduPilot**

The PixHawk controller is a general-purpose vehicle controller and hardware set built for hobby and commercial robotic vehicle applications. The controller comes equipped with built-in handheld remote control support, I2C support, a CAN bus, an analog-to-digital converter, and four general-purpose UART connections [3]. Additionally, the design features a servo-rail built for direct plug-in support for common servo connectors which can be externally powered to support servos up to a 10 volt, 10 amp rating. Built around the ARM Cortex M4 processor, the PixHawk has been widely used in commercial and academic robotic vehicle research [1, 8, 26].

To avoid the cost of developing embedded firmware, the USV was designed to use ArduPilot open-source firmware to provide control input to the vehicle. ArduPilot is a firmware suite that includes an assortment of vehicle classes depending on the dynamics and actuation method of the robot. The USV utilizes the ArduPilot-Rover class and its associated parameters and vehicle options to provide manual remote control, autonomous waypoint tracking, real-time GPS data, parameter tuning, and telemetry logging. Communication between ground control software, firmware, and embedded hardware is achieved using built-in MAVlink protocol. MAVlink is a lightweight serial communication protocol that allows the controller to receive input from open-source ground control software, or through programming classes such as the pymavlink Python class [6].

### **2.2.2 Ground Control and Vehicle Software**

Ground control software provides a means for the user to monitor the status of the USV. It is the link between the user and the device and acts as the constructor for MAVlink messages. QGroundControl and APM Mission Planner are popular open-source software packages that are compatible with the ArduPilot and PixHawk system. Both of these options were used extensively to tune the control parameters of the USV, and each had its own strengths. Despite the success of these programs for a single USV, neither one of them has validated support for a multi-vehicle architecture. To include multiple USVs, Python-based ground control and on-board vehicle software was developed for communicating with and monitoring the network. The ground control sends messages that



Figure 2.3: Picture of first-generation USV developed in the MSU Smart Microsystems Lab.



Figure 2.4: Mounted acoustic receiver.

are parsed by the vehicle software and converted to MAVlink using the pymavlink Python class. MAVlink commands are sent to the PixHawk using one of the available UART ports. The developed vehicle software is also responsible for registering, updating, and storing neighboring agents' data for running the proposed localization algorithms.

## 2.3 First-Generation USV

Figure 2.3 shows the first-generation USV developed in the MSU Smart Microsystems Lab. The parallel hulls were left hollow to allow for battery storage, and the waterproof Pelican case held all the remaining vehicle electronics. Two fixed BlueRobotics T200 thrusters were used to power the craft, and steering was achieved via differential drive of the motors. A PixHawk controller loaded with ArduRover firmware was used to control the USV, and all of the interfacing electronics were commercially purchased to allow for repetition in the design.

The underside of the hull was constructed with a fixed acoustic receiver mounting bracket as shown in Figure 2.4. This custom bracket allowed for a Vemco VR2 receiver to be mounted in one of two fixed positions: vertical (perpendicular to the surface of the water) or horizontal (parallel to the surface of the water). In both instances, the receiver was fully submerged in the water when the USV was deployed. A test of the USV's capacity for remote sensing was first demonstrated using a range test of the acoustic telemetry system at Higgins Lake in June 2019.



Figure 2.5: Map of the acoustic receiver range test for the mounted and stationary acoustic receivers. The blue arrow and line show the USV's path, the red pin marks the target, and the green pins mark the stationary receivers.

### 2.3.1 Tag Detection Performance.

The tag detection performance for various receiver configurations and USV operating modes were tested and compared to the traditional method of deploying static receivers on anchored buoys. Five stationary receivers were deployed at 75, 150, 225, 300, and 375 meters away from a stationary acoustic tag. The USV carrying the mounted acoustic receiver made four passes along this route (two passes with the receiver horizontal and two passes with it vertical), stopping for two-minutes at each stationary receiver's location.

Detection data for both the stationary and USV mounted receivers was analyzed using the known ping rate of the acoustic tag. The efficiency was calculated using the following equation:

$$Efficiency = \frac{\text{number of pings detected}}{\text{expected number of pings transmitted}} \times 100\% \quad (2.1)$$

Table 2.1 shows the receiver efficiency for each configuration, and Figure 2.5 shows a map of the range test. We observed that the mounted receiver performs better in the vertical position when compared to the horizontal position, noting that 62% of the pings received in the horizontal position

were when the device pointed towards the target, leaving only 38% of the pings recorded when the receiver was pointed away from the target. Additionally, receiver efficiency for all configurations drops significantly at and further than 225 meters. We note that one of the major differences between the mounted receiver in the vertical position and the stationary receivers was the depth at which the receiver was submerged. Surface back-scatter and electronic noise from the thruster could interfere with the receiver's transducer and cause missed pings. Lowering the receiver to a deeper depth could improve the performance of the mounted configuration.

Table 2.1: Tag detection efficiency for stationary receivers ("Stationary") and USV-mounted receiver with two configurations of the receiver orientation ("Vertical" and "Horizontal").

Receiver Efficiency (%)						
Orientation	0m	75m	150m	225m	300m	375m
Stationary	-	89	60	25	0	0
Vertical	100	77	70	0	0	0
Horizontal	90	74	0	0	0	0

## 2.4 Second-Generation USV

The second iteration of the catamaran USV design adopted many of the same features as the first while addressing some key issues. First, the dual thruster design provided an unnecessary amount of power and limited the longevity of the vehicle for long surveys. It was determined that incorporating a single thruster mounted to a servo actuator for steering would provide sufficient power and mobility to the device, while still allowing the option for a second thruster to be fitted for heavy payloads. Secondly, mounting the sensing electronics close to the surface of the water proved to be a limiting factor of the device's capabilities, and electronic interference from the thrusters caused sensor efficiency to drop significantly. Extending the TDOA-based localization algorithms to three-dimensions would require the acoustic receivers to be deployed at a known depth. This problem would be overcome by employing a winch to raise and lower the sensor to a desired depth.

### 2.4.1 Automated Winch

The proposed automated winch can be seen in Figure 2.6. A BlueRobotics Kevlar-reinforced Ethernet cable acts as both a support cable for the device and as a means to transfer data to the surface. The cable-wheel, motor, and mounting bracket were re-purposed from a commercially available Minn Kota Deckhand 25 electronic boat anchor, and the data cable was fed through a slip ring connector to allow for the cable-wheel to spin freely while maintaining network connection. Ethernet permits long range data transfer with minimal losses, supports IP network protocol between devices, and allows for a wide range of sensor options to be connected. This project sponsored an MSU ECE 480 Senior Design team in the fall of 2019 and was expanded upon to be compatible with the USV network.

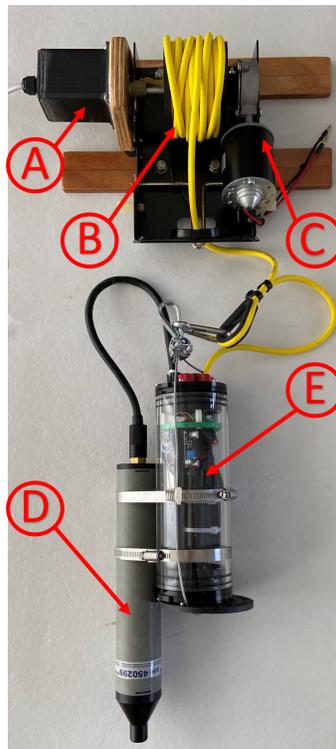


Figure 2.6: Automated winch and sensor hub unit: (A) Ethernet slip ring and housing, (B) Kevlar reinforced Ethernet tether cable, (C) winch motor and gearbox, (D) Vemco VR2C acoustic receiver, and (E) sensor hub unit.

## 2.4.2 Sensor Hub

Real-time localization estimates require that TOA measurements from the deployed receiver be sent to the surface computer on the boat to be shared amongst the nodes of the network. To keep the open-architecture of the USV, a submersible device was designed and built to act as a relay between the underwater sensor and the surface computer. This project sponsored an MSU ECE 480 Senior Design team in the spring of 2020 and was built and modified for the USV system. Figure 2.6 shows the sensor hub device attached to the automated winch. The device holds a depth sensor, Raspberry Pi, battery, and serial converters to which the sensors and Ethernet cable are connected. The Raspberry Pi is connected via Ethernet to a second Raspberry Pi on the surface for data processing using the developed Python software.

It is imperative that the clock on each receiver be in sync in order to accurately localize the target using TDOA-based algorithms. As a simple example, if we assume that the speed of the signal is 1500 m/s, then a clock that is out of sync by 1 ms results in approximately 1.5 m of distance error in the estimate. The sensor hub is used to address this issue by providing a means for GPS data from the surface boat to pass quickly through the network cable to update the receiver's internal clock.

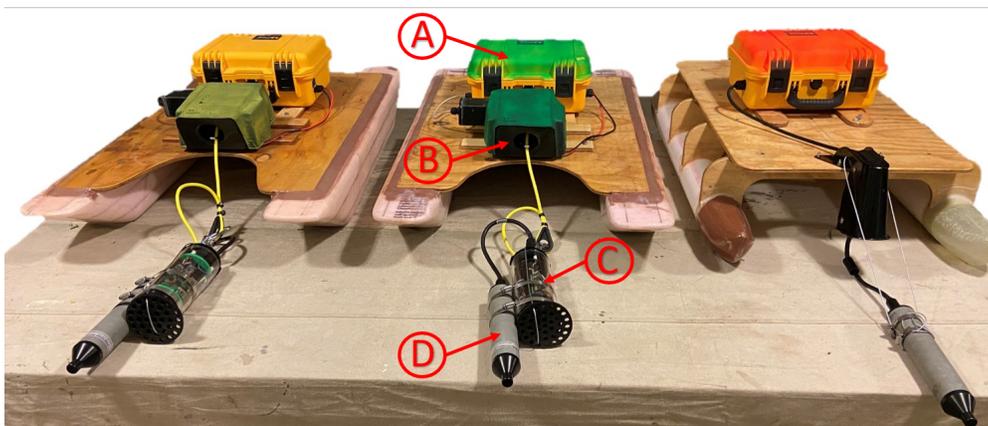


Figure 2.7: A collection of USVs used for TDOA tests. (A) Waterproof Pelican case housing USV electronics, (B) automated winch, (C) sensor hub, and (D) Vemco VR2C acoustic receiver. Note: the USV to the far right does not incorporate an automated winch and sensor hub unit.



Figure 2.8: Deployed USV network. (T) represents the USV carrying the acoustic tag, and (R) represents the receiving nodes.

### 2.4.3 Network of USVs

As previously mentioned, repeatable construction of the USV is a top priority. In order to run TDOA-based localization in two-dimensions, a minimum number of three sensing nodes, or USVs, are required. These nodes must be able to communicate GPS data and TOA measurements to each other, and each must be able to be controlled independently by the ground station. The Raspberry Pi on the surface boat has a USB connected GPS device, as well as an XBee Pro S3B utilizing ZigBee protocol for long-range telemetry. Nodes broadcast relevant data to the network by applying round-robin scheduling to time the messages between nodes and the ground station. Additionally, a fourth USV carrying the acoustic tag is required to simulate target movement. This USV is programmed with a waypoint trajectory generated by a discrete-time correlated random walk (DCRW) discussed in later sections. Figure 2.8 shows the network of USV deployed on open-water with sensor hubs and receivers submerged.

## 2.5 Future Work

Despite the success of the catamaran hull, the design lacked features to be used in more austere environments. For example, a large wave could capsize the boat causing the winch to be fully submerged, leaving no option to self recover. Figure 2.9 and 2.10 show an alternative design of USV, which was also prototyped by the Smart Microsystems Lab. The vehicle was constructed with a fully waterproof, submersible shell and has internal ballast to self-right in the event that the boat capsizes. In addition, the multilayered carbon fiber and fiberglass shell was durable, and the electronics were fixed securely in the inside of the hull. Future work includes designing a hull similar to this design but with the addition of the automated winch.



Figure 2.9: An USV prototype developed by the Smart Microsystems Lab based on an alternative design concept.



Figure 2.10: Demonstration of the self-righting capabilities of the USV based on the alternative design.

## CHAPTER 3

### DISTRIBUTED TDOA-BASED LOCALIZATION

In centralized TDOA localization algorithms, a reference node is assigned and gathers time-of-arrival (TOA) measurements from neighboring nodes. The reference node subtracts TOA measurements from its own, converts the resulting TDOA measurements to ranges using the known propagation speed of the signal, and estimates the location of the target using the geometric intersection of the measurements [18, 40, 46]. In distributed TDOA algorithms, each node assumes itself as the reference node, obtains TDOA measurements through communication with its neighbors, estimates the target's location based on available measurements, and uses information available from its neighbors to update the estimate [38].

The location of the target is estimated using the TDOA measurements and the known locations of sensor nodes. In the presence of measurement noise, TDOA-based localization algorithms for both the centralized and distributed approaches adopt varying state estimation techniques for target localization, e.g., the extended Kalman filter (EKF) [38], unscented Kalman filter (UKF) [40], and particle filter (PF) [18, 46]. It has been shown that, under a distributed EKF scheme, if the target dynamics are modeled as a constant-velocity model with white Gaussian noise as acceleration, a given node with an insufficient number of TDOA measurements from its immediate neighbors can still estimate the location of the target if it is connected to a network that has a sufficient number of TDOA measurements [35, 36]. However, little work has been done to explore distributed TDOA-based algorithms when the movement model is non-Gaussian, or the target's movement model is not known before hand. For example, in animal tracking applications, the movement model is dependent on the animal's interaction with the environment and may not be precisely known prior to localization and tracking [2, 9, 16]. Instead, a stochastic process is used to model the animal's movement, and a particle filter (PF) can be used to handle such models.

In this chapter we present a distributed PF for TDOA-based localization of a target modeled by a discrete-time correlated random walk (DCRW). In addition, we propose an online model learning

approach in the distributed PF, in the presence of model uncertainty for the target dynamics. We briefly explain the model update process for the distributed PF here. Prior to execution, a uniform distribution is assumed by the PF to propagate between time-steps. Following each iteration of the algorithm, a posterior distribution is generated using PF estimates to recursively update the PF movement model for the next iteration. It is shown with simulation that the PF can estimate the unknown parameters for the distributions of the target's movement model, which can then be used to update the PF's own propagation model.

### 3.1 Problem Setup

First, we present a DCRW model for the moving target. The DCRW model is based on two random variables that represent the turning angle,  $\phi$ , and step length,  $\delta$ , respectively. The turning angle,  $\phi$ , is randomly selected at each time-step from a wrapped Cauchy distribution with probability density function (PDF)

$$p(\phi; \mu, \gamma) = \frac{1 - \gamma^2}{2\pi(1 + \gamma^2 - 2\gamma \cos(\phi - \mu))}, \quad (3.1)$$

for  $-\pi < \phi \leq \pi$ , where  $\gamma$  is the clustering parameter, and  $\mu$  is the mean turning angle. Similarly, the step length,  $\delta$ , is randomly selected at each time-step from a Weibull distribution with PDF

$$p(\delta; \kappa, \lambda) = \frac{\kappa}{\lambda} \left(\frac{\delta}{\lambda}\right)^{\kappa-1} e^{-\left(\frac{\delta}{\lambda}\right)^\kappa}, \quad (3.2)$$

for  $\delta \geq 0$ , where  $\kappa$  and  $\lambda$  are the shape and scale parameters, respectively. We define the heading angle,  $\theta(k)$ , and step length,  $\Delta(k)$ , at time  $k$  as

$$\theta(k) = \theta(k-1) + \phi, \quad (3.3)$$

$$\Delta(k) = \delta, \quad (3.4)$$

for  $k \geq 0 \in \mathbb{Z}$ , with initial heading  $\theta(0)$ . For the purpose of this work, we consider a moving target in the 2D plane where  $p(k) = \begin{bmatrix} p^x(k) & p^y(k) \end{bmatrix}^T$  denotes the Cartesian coordinates of the target at time-step  $k$ . The propagation of the target is given by

$$p(k) = p(k-1) + \Delta(k) \begin{bmatrix} \cos(\theta(k)) \\ \sin(\theta(k)) \end{bmatrix}, \quad (3.5)$$

for  $k \geq 0 \in \mathbb{Z}$ , where  $p(0) = \begin{bmatrix} p^x(0) & p^y(0) \end{bmatrix}$  is the starting location of the target.

The target emits a signal at regular time intervals that is detected by a group of  $N$  receivers, referred to as nodes. For a given node  $i$ , the set of nodes that can communicate with node  $i$  are referred to as the *neighbors* of  $i$ , denoted  $\mathcal{N}_i$ . When the signal is received at node  $i$ , the time of arrival (TOA) is recorded, and the node acquires TOA measurements from all of its neighbors. The TOA of the received signal is dependent on each node's relative distance from the target. Assuming that the propagation speed of the signal is known, the measurements at each node can be expressed as

$$y_i(k) = h_i(k) + w_i(k) \quad (3.6)$$

where  $w_i(t) \in \mathbb{R}^{N_i}$  is assumed to be zero-mean, white Gaussian noise,

$$h_i(k) = \begin{bmatrix} h_{i,1}(k) \\ \vdots \\ h_{i,N_i}(k) \end{bmatrix}, \quad (3.7)$$

and

$$h_{i,j}(k) = \|p(k) - \rho_i(k)\| - \|p(k) - \rho_{i,j}(k)\|. \quad (3.8)$$

Here,  $N_i = |\mathcal{N}_i|$  (cardinality of the set  $\mathcal{N}_i$ ) denotes the number of nodes that can communicate with node  $i$ ,  $\rho_i(k) = \begin{bmatrix} \rho_i^x & \rho_i^y \end{bmatrix}^T$  is the position of node  $i$ , and  $\rho_{i,j}(k) = \begin{bmatrix} \rho_{i,j}^x & \rho_{i,j}^y \end{bmatrix}^T$  is the position of the  $j$ -th neighbor of node  $i$ .

We investigate two problems in this work. First, we propose a distributed particle filter for localizing a moving target modeled by DCRW. Second, we present a method for estimating the unknown parameters for the movement model of the target.

### 3.2 Distributed Particle Filter Algorithm with Online Model Learning

In this section, we define a particle filter (PF) estimator using TDOA measurements for the distributed PF approach, and we discuss our method for updating the unknown parameters for the DCRW model. The goal for the distributed approach is to obtain localization estimates without requiring a single reference node to process all TDOA measurements for the system. Instead,

each node generates TDOA measurements from its available neighbors, runs its own particle filter algorithm, shares information with its neighbors, and determines a location estimate for the target. Researchers in [11, 29, 44, 47] have developed various methods for distributing particle filtering across multiple nodes of a sensor network. Our approach utilizes a consensus step at each node that shares a portion of the highest weighted particle data between neighbors to update individual estimates at each node. The particle filter algorithm is summarized by the following.

### 3.2.1 Distributed Particle Filter Algorithm

*Initialization:* At each node, generate  $P$  random particles (each representing a potential location for the target) uniformly distributed in the space of possible target locations.

1. *Particle Measurement Update:* For each particle  $n = 1, 2, 3, \dots, P$ , compute TDOAs for every node  $i$ . Let  $\{c_i^n(k)\}_{n=1}^P$  be the set of particles for node  $i$ , where  $c_i^n(k) = [c_i^{nx}(k) \ c_i^{ny}(k)]$  denotes the  $n$ -th particle location of node  $i$  at time-step  $k$ . This particle induces a *particle measurement*  $d_i^n(k) = [d_{i,1}^n(k), \dots, d_{i,N_i}^n(k)]^T$ , where  $d_{i,j}^n$  is given by

$$d_{i,j}^n(k) = \|c_i^n(k) - \rho_i(k)\| - \|c_i^n(k) - \rho_{i,j}^n(k)\|. \quad (3.9)$$

2. *Weight Update:* Update particle weights based on the assumed noise model for the sensor. Let  $y_i(k)$  denote the target TDOA measurement from (3.8), and  $\epsilon_i^n(k)$  denote the difference between the target TDOA measurement and the particle TDOA measurement,

$$\epsilon_i^n(k) = \|y_i(k) - d_i^n(k)\|. \quad (3.10)$$

The particle weight,  $w_i^n(k)$ , can be updated based on the sensor noise model. In this work, we assume a zero-mean Gaussian distribution with standard deviation  $\sigma$  for the sensor noise. The weight  $w_i^n(k)$  is then computed as the posterior probability,

$$w_i^n(k) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\epsilon_i^n(k)}{\sigma}\right)^2}, \quad (3.11)$$

and the weights are normalized to obtain:

$$\bar{w}_i^n(k) = \frac{w_i^n(k)}{\sum_{n=1}^P w_i^n(k)}. \quad (3.12)$$

Normalized particle weights are used for the following *Consensus*, *Estimation*, and *Resampling* steps.

3. *Consensus*: Let  $\{s_i^n(k)\}_{n=1}^m$  be a subset containing the highest weighted particles for node  $i$ , such that  $m < \frac{P}{N}$ . Here,  $s_i^n(k) = [\bar{w}_i^n(k) \quad c_i^{nx}(k) \quad c_i^{ny}(k)]$  denotes the  $n$ -th particle weight and location of node  $i$  at time-step  $k$ . Each node  $i$  broadcasts its highest weighted particles,  $\{s_i^n(k)\}_{n=1}^m$ , to all neighbors,  $j \in \mathcal{N}_i$ . Node  $i$  receives node  $j$ 's highest weighted particles and replaces the same portion of its lowest weighted particles. Next, we redefine the set of particle locations,  $\{c_i^n(k)\}_{n=1}^P$ , and their associated normalized weights,  $\bar{w}_i^n(k)$ , calculated from the particles received at node  $i$ . The number of particles,  $m$ , shared between nodes is dependent on the assumed communication constraints of the network.
4. *Estimation*: Estimate the target's location at time-step  $k$  using a weighted average of a portion of highest weighted particles. Let  $S_i = [W_i(k) \quad C_i(k)]$  be an array defined as the top 20% of highest weighted particles and the associated locations, where  $W_i(k)$  is a vector of particle weights and  $C_i(k)$  is a matrix of particle locations. Let  $\chi_i(k)$  be the PF's estimate of the target's location, which is obtained as

$$\chi_i(k) = W_i(k)^T C_i(k). \quad (3.13)$$

5. *Resampling*: At each time-step  $k$ , take  $P$  samples (with replacement) of the set  $\{c_i^n(k)\}_{n=1}^P$  where the probability of a given sample  $n$  is  $\bar{w}_i^n(k)$ .
6. *Time-Update*: Propagate particles according to the knowledge of the DCRW model for the target movement dynamics. Let  $\theta_i^n(k)$  denote the heading of each particle,  $\phi_i^n(k)$  be a randomly sampled angle from the assumed turning angle PDF,  $\delta_i^n(k)$  be a randomly sampled step length from the assumed step length PDF, and  $\Delta_i^n(k)$  be the step-length of the particle

at  $k$ . For  $n = 1, 2, 3, \dots, P$

$$\theta_i^n(k+1) = \theta_i^n(k) + \phi_i^n(k), \quad (3.14)$$

$$\Delta_i^n(k+1) = \delta_i^n(k), \quad (3.15)$$

$$c_i^n(k+1) = c_i^n(k) + \Delta_i^n(k+1) \begin{bmatrix} \cos(\theta_i^n(k+1)) \\ \sin(\theta_i^n(k+1)) \end{bmatrix}. \quad (3.16)$$

The algorithm then goes back to step (1) and repeats the process.

### 3.2.2 Online Learning of Target Model Parameters

A key objective of this work is to use the estimate from the TDOA-based particle filter algorithm to construct the unknown probability distributions of the target's DCRW movement model. Let  $\chi_i(k)$  be the PF's estimated target location from (3.13). We can obtain an estimate  $\hat{\theta}_i(k)$  of the heading, and the change in heading angle  $\hat{\phi}_i(k)$ , of the particle filter's estimate from  $k-1$  to  $k$ , for  $k \geq 2$ , as

$$\hat{\theta}_i(k) = \text{atan2} \left( \frac{\chi_i^y(k) - \chi_i^y(k-1)}{\chi_i^x(k) - \chi_i^x(k-1)} \right), \quad (3.17)$$

$$\hat{\phi}_i(k) = \hat{\theta}_i(k) - \hat{\theta}_i(k-1), \quad (3.18)$$

where  $\text{atan2}$  is the four-quadrant tangent function. Similarly, let  $\hat{\delta}_i(k)$  be the change in step-length for the PF estimate from  $k-1$  to  $k$ :

$$\hat{\delta}_i(k) = \|\chi_i(k) - \chi_i(k-1)\|. \quad (3.19)$$

Define  $\{\hat{\phi}_i(k)\}$  and  $\{\hat{\delta}_i(k)\}$  as the set of observed turning angles and step-lengths, respectfully. We assume that the PDF functions for the turning angle and step-length, given by (3.1) and (3.2), are known and use maximum likelihood estimation to estimate the unknown parameters of the PDFs based on the observed turning angles,  $\{\hat{\phi}_i(k)\}$ , and step-lengths,  $\{\hat{\delta}_i(k)\}$ .

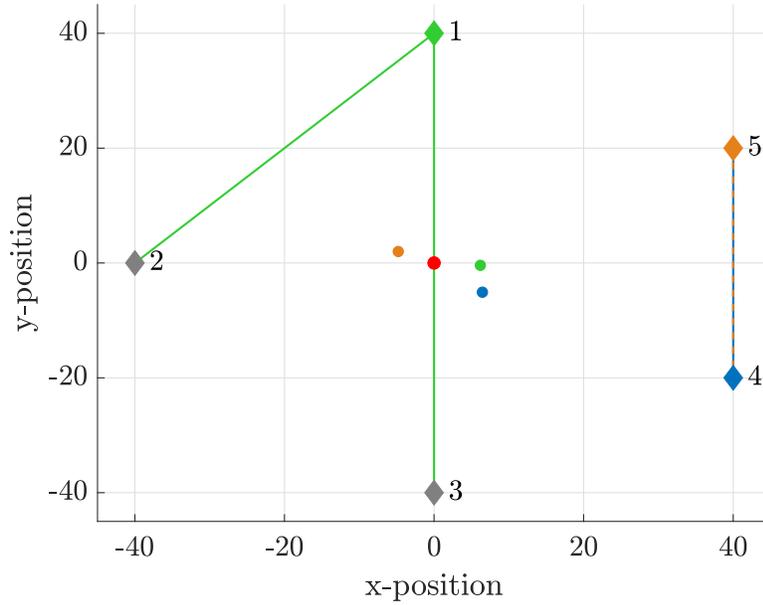


Figure 3.1: Simulation setup. Here the diamonds represent the nodes’ locations, the red circle represents the moving target, the green, blue, and orange circles represent the estimates by nodes 1, 4, and 5, respectively, and lines connecting the nodes represent communication links.

### 3.3 Simulation Results

The simulation defines five stationary nodes and initializes  $P$  uniformly distributed particles  $\{c_i^n(0)\}_{n=1}^P$  for each node  $i$ . The initial position of the target is randomly selected uniformly over the workspace. For each set of parameters tested, the target moves a given number of time-steps according to the DCRW model, and the simulation is repeated for  $r = 50$  iterations. The average norm error is used to assess the performance of each algorithm.

#### 3.3.1 Distributed Particle Filter

In this section, we investigate the performance and utility of the distributed particle filter (PF) algorithm for  $P = 100$  particles,  $k = 100$  time-steps, and  $r = 50$  simulation iterations where the PF assumes that it knows the target’s true movement model. For nodes running the distributed PF algorithm, the top 10% ( $m = 10$ ) of highest weighted particles are shared between distributed nodes, and the top 20% of particles are used for state estimation. Let  $\chi_i(k)$  denote node  $i$ ’s estimate

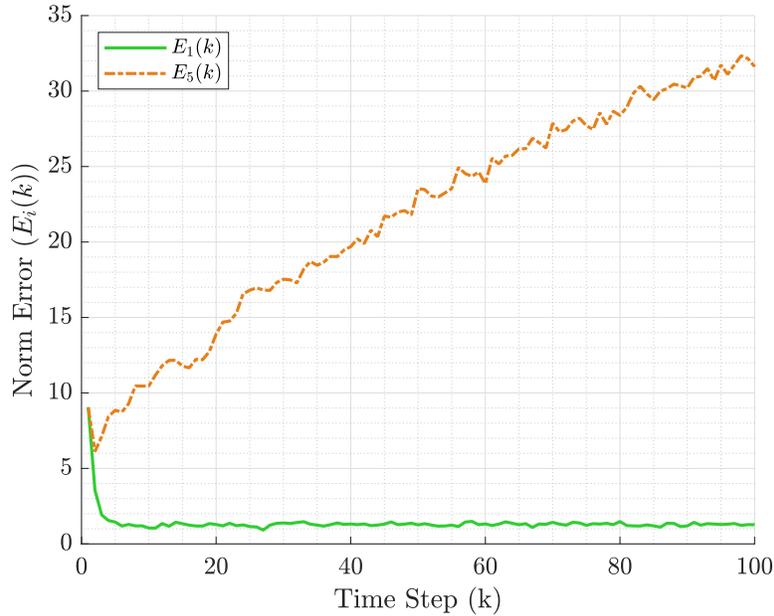


Figure 3.2: Average norm errors for nodes 1 and 5 at each time-step for the setting of  $\mathcal{N}_1 = \{2, 3\}$ ,  $\mathcal{N}_4 = \{5\}$ , and  $\mathcal{N}_5 = \{4\}$ . Note:  $\mathcal{N}_5$  has an insufficient number of TDOA measurements.

of the target's location at time-step  $k$ . We define the *norm error*,  $E_i(k)$ , as the Euclidean distance between the PF's estimate and the target's true location, i.e.,

$$E_i(k) = \|\chi_i(k) - p(k)\|. \quad (3.20)$$

Figure 3.1 shows the configuration of the nodes where node 1 has two neighbors,  $\mathcal{N}_1 = \{2, 3\}$ , while nodes 4 and 5 have only one neighbor each, i.e.,  $\mathcal{N}_4 = \{5\}$  and  $\mathcal{N}_5 = \{4\}$ . In this configuration, nodes 1, 4 and 5 utilize the distributed PF algorithm to estimate the target's location. Node 1 has access to a sufficient number of TDOA measurements by receiving TOA measurements from nodes 2 and 3, while nodes 4 and 5 have an insufficient number of TDOA measurements ( $N = 1$ ). Figure 3.2 shows the average norm error at nodes 1 and 5. We see that the PF estimate at node 1 converges, while the estimate from node 5 diverges as expected. Next, we assume that node 1 establishes a connection to node 4 so that  $\mathcal{N}_1 = \{2, 3, 4\}$ ,  $\mathcal{N}_4 = \{1, 5\}$ , and  $\mathcal{N}_5 = \{4\}$ . Figure 3.3 shows the norm error for the PF estimates at nodes 1 and 5. We observe that node 5 is connected to a network with a sufficient number of TDOA measurements through node 4, and as a result, the estimate of node 5 converges. This result suggests that exchanging of particles, through the

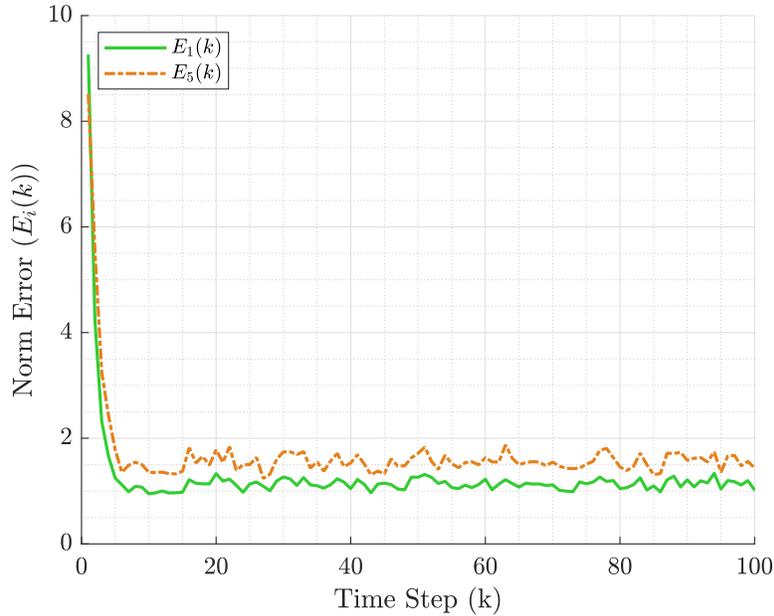


Figure 3.3: Average norm errors for node 1 and node 5 at each time-step for the setting of  $\mathcal{N}_1 = \{2, 3, 4\}$ ,  $\mathcal{N}_4 = \{1, 5\}$ , and  $\mathcal{N}_5 = \{4\}$ . The PF estimate of  $\mathcal{N}_5$  converges despite having an insufficient number of TDOA measurements.

consensus step, allows other agents to successfully localize the target even with an insufficient number of TDOA measurements.

In addition to the previous observation, the exchange of particles in the proposed distributed PF approach can further improve the estimates of neighbors with sufficient measurements. To show that, we consider the case when the network is connected such that  $\mathcal{N}_1 = \{2, 3, 4\}$ ,  $\mathcal{N}_4 = \{1, 5\}$ , and  $\mathcal{N}_5 = \{4\}$ . Figure 3.4 shows a comparison of the norm error at node 4 for a distributed PF utilizing the consensus step and one without. We observe that, while the estimates converge to their true values in both cases, the norm error decreases with the addition of the consensus step. This is facilitated by information fusion and particle exchange among nodes, as information from more TDOA measurements are available at each node.

### 3.3.2 Online Model Learning

In this section, we no longer assume that the PF has access to the target’s true parameters for the PDF distributions of the DCRW model. The change in heading angle and step length for  $k > 3 \in P$  of

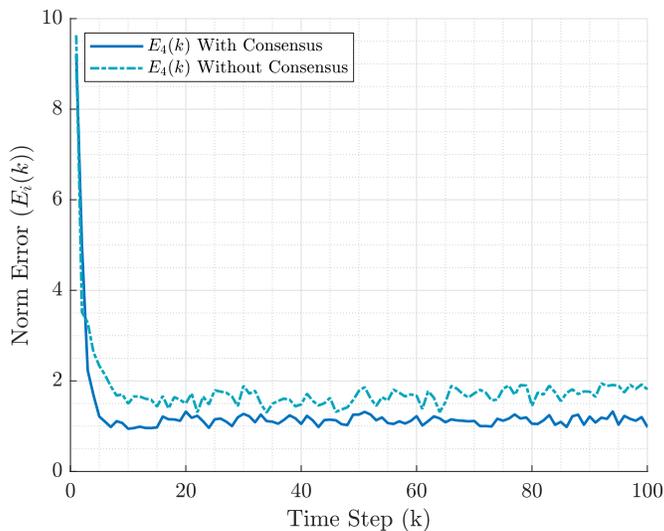


Figure 3.4: Average norm error,  $E_4(k)$ , with and without the consensus step.

the particle filter's estimate is used to recursively update the target's movement model in real-time. Each PF calculates the change in heading and step-length from the location estimate according to (3.18) and (3.19) and uses the sets  $\{\hat{\phi}_i(k)\}$  and  $\{\hat{\delta}_i(k)\}$  to estimate the parameters of (3.1) and (3.2) using MATLAB's maximum likelihood estimation (mle) solver.

Consider a network of  $N = 5$  nodes configured such that  $\mathcal{N}_1 = \{2, 3, 4\}$ ,  $\mathcal{N}_4 = \{1, 5\}$ , and  $\mathcal{N}_5 = \{4\}$ . Assume that each node utilizes the distributed PF algorithm with consensus, and the performance at node 5 is considered. In this section, we compare the distributed PF with adaptive online model learning to a distributed PF using non-adaptive Gaussian distributions for the particle propagation model. For the non-adaptive Gaussian model, the turning angle PDF is assumed to have  $\mu = 0$ ,  $\sigma = \frac{\pi}{4}$  for  $-\pi < \phi \leq \pi$  and the step length PDF is assumed to have  $\mu = 2$ ,  $\sigma = 1$  for  $\delta \geq 0$ .

Figure 3.5 shows the average norm error of node 5's estimate at each time-step for the adaptive, Gaussian, and known PDF propagation models. As time increases, the target moves away from the optimal configuration of the estimators, and the lower bound of the error increases [34]. Assuming the wrong movement model (Gaussian) does significantly worse than when we utilize the online learned model. We observe that the average norm error for the adaptive model is consistent with

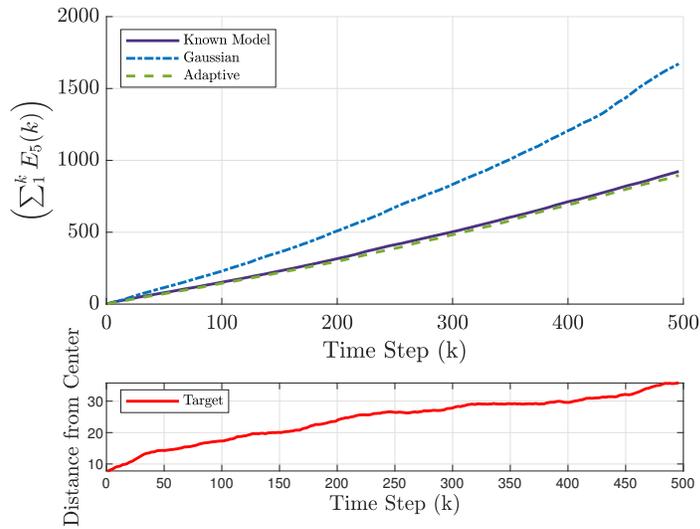


Figure 3.5: Cumulative norm error,  $\sum_1^k E_5(k)$ , for each movement model compared to the distance the target moved from the center of the nodes. The center of the nodes is at  $(0, 0)$ .

knowing the true parameters of the DCRW model. Figure 3.6 shows an example of the online learned model as time increases. The particle propagation model starts with a uniform distribution at  $k = 0$  and updates each time-step according to the estimated parameters. The estimated PDFs center around the target's true PDFs for the turning angle and step length distributions.

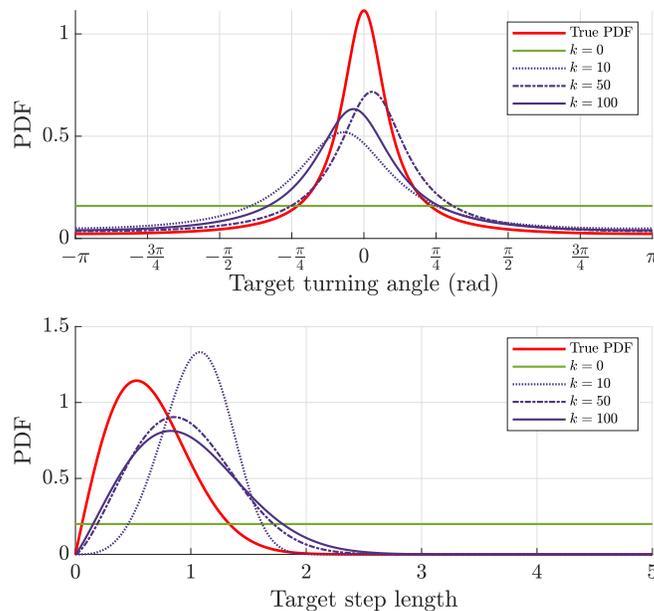


Figure 3.6: PDF estimates of the target's movement model compared to the true PDFs.

## CHAPTER 4

### FIELD EXPERIMENTAL RESULTS ON TDOA-BASED TARGET LOCALIZATION

This chapter discusses the field experiments conducted to assess the distributed TDOA-based localization algorithms. The experiments were designed in two parts. First, a test of the localization algorithm for a passively drifting target was conducted. In this test, receiving USVs (nodes) were deployed in a desirable formation after network initialization and passively collected data. Nodes were not actively controlled to hold position; this would minimize noise from the thruster turning on. The target USV was deployed in the center of the formation, and each node continually updated its GPS data, time-of-arrival (TOA) measurements, and stored data from neighboring nodes. The second experiment was designed to verify the distributed PF algorithm for a target trajectory modeled by a discrete-time correlated random walk (DCRW). To achieve this, a fifty time-step DCRW trajectory was randomly generated according to the methods outlined in the previous chapter. The x-y coordinates of the trajectory were converted to GPS points and loaded as a waypoint trajectory in the target USV's autopilot. Receiving nodes were again placed in a desired formation around the target to passively collect network data.

These field experiments were conducted on Crockery Lake in Chester Township, Michigan over the month of November 2020. The data was post-processed using MATLAB, and used to evaluate the distributed TDOA-based PF algorithm and distributed TDOA-based extend Kalman filter (EKF) algorithm [36].

#### 4.1 Experiment Setup

##### 4.1.1 Waypoint Tracking

Precise position control for each node in the network is not required for the purpose of TDOA-based localization of a moving target. As long as each node knows the precise location of itself and each of its neighbors at any time-step, then localization can be achieved. That being said,

for two-dimensional localization, the three co-planar nodes *must* be in a non-collinear formation to accurately locate the target, and the lower bound for the variance of the estimation error is dependent on the location of each node relative to each other and the target. The optimal formation to minimize the Cramer-Rao bound for the estimation error is given by a uniform circular array, with each node equidistant on the circumference of a circle of given radius around the target [34]. For three nodes, this formation was achieved using an equilateral triangle with the target in the middle. To mitigate issues with network deployment and to minimize measurement noise caused by the electric motor engaging, receiving nodes were manually deployed in a triangular formation around the target to passively collect data without motor control input.

Each iteration from  $k - 1$  to  $k$  of the DCRW was given by a straight line. Therefore, the target USV was only required to complete a trajectory from one waypoint to the next waypoint at any given time-step. ArduPilot open-source firmware, along with QGroundControl host computer software, has built-in support for autonomous waypoint tracking and a manual override by a handheld remote controller which was validated using the first-generation USV and the USV with an alternative hull design. Waypoint tracking using ArduPilot was achieved using two separate controllers: a PI velocity controller and a feed-forward PID heading controller. These controllers were tuned to meet the desired performance for the target USV on open-water at Lake Lansing in August 2020. Figure 4.1 shows the performance of the USV waypoint tracking. The steady-state oscillations along the desired trajectory could not be settled despite multiple efforts to tune the built-in controllers. This issue highlights one of the biggest disadvantages of open-source firmware packages, as there are limited design options available to the controller. However, despite these slight oscillations, the target USV was successfully able to navigate from one waypoint to the next regardless of initial orientation. It was determined that the tracking performance was sufficient for the time being and improvements would be made in future implementations.



Figure 4.1: USV waypoint tracking after controller tuning. Slight steady-state oscillations along the trajectory can be observed.

#### 4.1.2 LAN Configuration

As discussed in Chapter 1, Python-based ground control and vehicle control software was developed to support the multi-agent network. This software was extensively bench-tested prior to running TDOA-based experiments but had limited capacity to monitor the devices in real-time. Real-time device system access allows for improvements in system initialization, emergency code modification, troubleshooting, and data analysis. Additionally, having the sensor hub connected over Ethernet (TCP/IP) allows for remote login to the underwater computer connected directly to the Vemco VR2C receiver.

The LAN network was configured by assigning one of the USVs as a WiFi hotspot and DHCP server. All the USVs, as well as the host computer, connected to the network with an assigned static IP address. In this case, the target USV was configured as the DHCP server since it would likely stay in the middle of the USV formation. Each device automatically connected to the network when they were powered on, and the host computer was used to remotely login to each device using an SSH connection. Once connected, the SSH shells were used to view real-time data and program execution during each experiment. Figure 4.2 shows an example window of the host computer after logging into each device. Having remote access to each computer on the USV network proved invaluable for troubleshooting in the field during experiments. The LAN at each node was also used

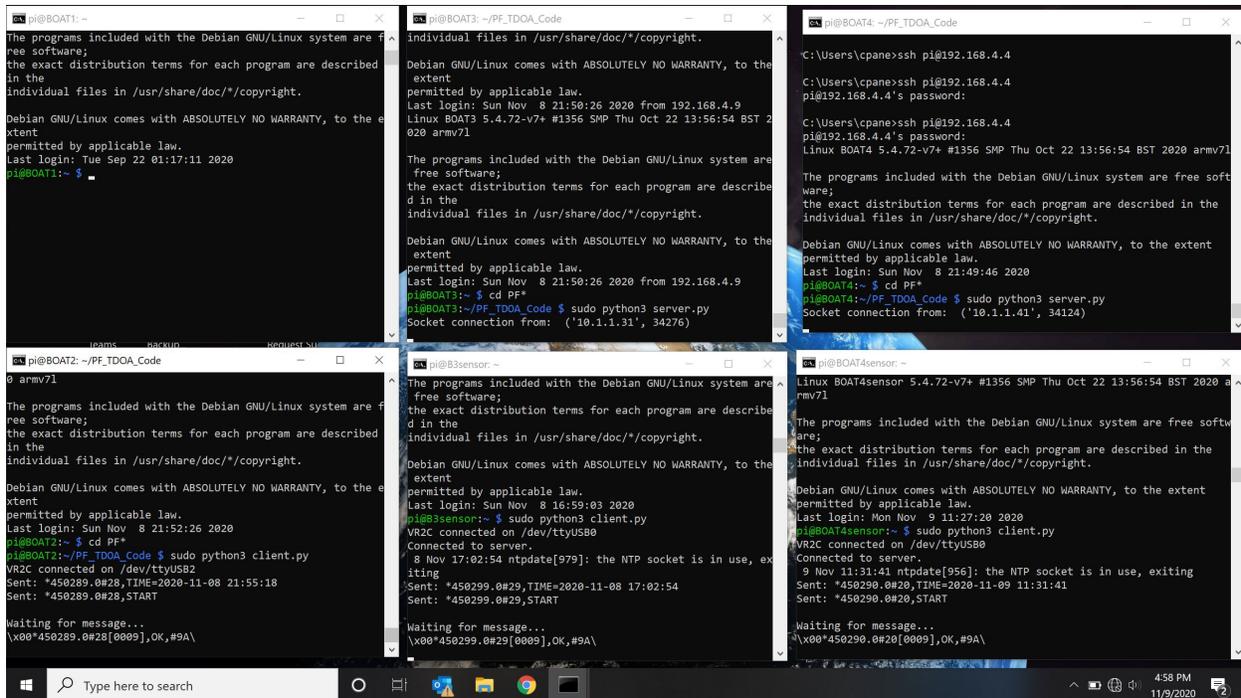


Figure 4.2: Screen view of the host computer monitoring each device on the network through SSH connection.

to transfer TOA measurements from the sensor hub Raspberry Pi computer (RPI) to the surface and to update receiver clocks based on GPS. Receiver clock synchronization is imperative to accurately localize a target using TDOA-based algorithms. A method to update receiver clocks was designed in which GPS data was continually sent from the surface RPI to the sensor RPI through an IP socket; then the sensor RPI updated the receiver clock every four minutes to account for clock drift at each node. A dry test of the acoustic telemetry equipment was performed before the USVs were deployed on water to ensure their proper function. Receiver transducers were placed a few centimeters from the acoustic tag in air, and TOA measurements were monitored using the LAN. TOA and GPS updates between the nodes of the network were still accomplished using the XBee radios instead of the LAN due to its limited range.

## 4.2 Coordinate Transform

The PF algorithm relies on the sensing nodes and target operating in a Cartesian coordinate system in the two-dimensional space. Since the nodes and the target USV localize using GPS

coordinates in a spherical coordinate space, it is required that each GPS point be converted to a Cartesian point where one unit of the Cartesian plane represents one meter of linear distance. We assume that the USVs operate in close enough proximity to neglect the curvature of the Earth and calculate their relative position using a known GPS *origin point*. The origin point is shared to each node during network initialization to ensure all nodes have the same frame of reference.

We denote the GPS coordinates of node  $i$  as  $l_i(k) = \begin{bmatrix} l_i^x(k) & l_i^y(k) \end{bmatrix}$ , where  $l_i^x(k)$  and  $l_i^y(k)$  correspond to the longitude and the latitude, respectively, of node  $i$  at time-step  $k$ , and let  $l_0$  be the GPS coordinates of the reference frame. Then the position of node  $i$ , denoted  $\rho_i(k) = \begin{bmatrix} \rho_i^x(k) & \rho_i^y(k) \end{bmatrix}^T$ , with respect to the reference frame, is given by

$$\rho_i^x(k) = \cos(l_i^y) \sin(l_i^x(k) - l_0^x) \quad (4.1)$$

$$\rho_i^y(k) = \cos(l_0^y) \sin(l_i^y(k)) - \sin(l_0^y) \cos(l_i^y(k)) \cos(l_i^x(k) - l_0^x), \quad (4.2)$$

where  $l_i^x(k)$ ,  $l_i^y(k)$ , and  $l_0$  are measured in decimal degrees [4]. Once an estimate of the target's location is calculated, each node must convert the estimate's Cartesian position, given by  $p(k) = \begin{bmatrix} p^x & p^y \end{bmatrix}^T$ , to a GPS coordinate for reference to the environment. We define the distance,  $d$ , and relative bearing,  $\beta$ , as

$$d = \sqrt{(p^x(k))^2 + (p^y(k))^2} \quad (4.3)$$

$$\beta = \arctan\left(\frac{p^y(k)}{p^x(k)}\right). \quad (4.4)$$

Given the radius of the Earth,  $r$ , the target's longitude  $l_t^x(k)$  and latitude  $l_t^y(k)$  are given by

$$l_t^y(k) = \arcsin\left(\sin(l_0^y) \cos\left(\frac{d}{r}\right) + \cos(l_0^y) \sin\left(\frac{d}{r}\right) \cos(\beta)\right) \quad (4.5)$$

$$l_t^x(k) = l_0^x + \text{atan2}\left(\frac{\sin(\beta) \sin\left(\frac{d}{r}\right) \cos(l_0^y)}{\cos\left(\frac{d}{r}\right) - \sin(l_0^y) \sin\left(l_t^y(k)\right)}\right). \quad (4.6)$$

These equations were also used to generate the DCRW GPS waypoint trajectory discussed in later sections [4].

### 4.3 Distributed TDOA Localization - Passively Drifting Target

The first successful distributed TDOA test occurred on Saturday, November 28, 2020 on Crockery Lake in Chester Township, Michigan shown in Figure 4.3. Table 4.1 shows the weather on the day of the test. The experiment setup followed the procedure listed in previous sections, and the USVs were loaded on a boat for deployment. Network initialization using the developed Python vehicle software took place prior to open-water deployment to verify system status while in range of the WiFi LAN.

A five-pound barbell weight was attached to the end of a fifteen-foot rope mounted to the bottom of each USV to help mitigate surface drift during the duration of the experiment. Note that these

Table 4.1: Approximate weather conditions for the stationary target experiment.

Crockery Lake, MI				
Weather Conditions - November 28, 2020				
Air (°F)	Water (°F)	Wind Speed (mph)	Wind Direction	Time (24-hr EST)
34-37	40-45	8-13	WSW	11:56-12:35

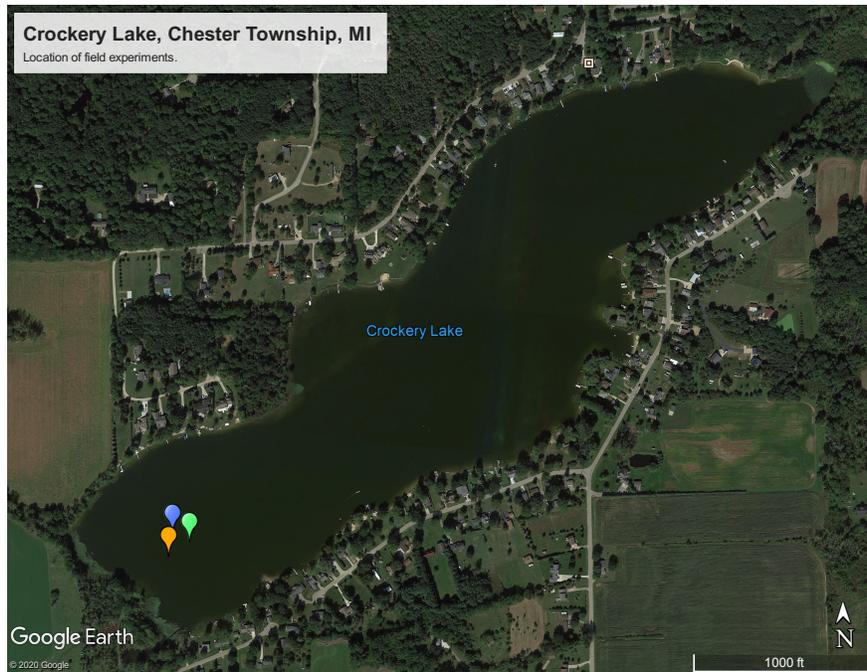


Figure 4.3: USVs deployed on Crockery Lake, MI. The colored pins show the initial location of the receiving USVs for the passive target experiment.

weights did not reach the bottom of the lake to anchor the USVs in a stationary position due to the depth at the location of the test, but they did help to slow drift. Three receiving USVs were placed in a triangular formation, with the fourth target USV placed at the approximate centroid. Acoustic receivers were attached to the USVs using stainless steel cables, and the acoustic tag was securely attached on the weighted rope at a depth lower than the receivers' transducers. The formation was placed at the southwest corner of the lake to allow wind drift to push the USVs across the largest available stretch of open-water. The USV network passively collected data for twenty-minutes, and the acoustic tag (target) sent a ping every seven seconds.

In this section, the distributed TDOA particle filter (PF) algorithm outlined in Chapter 3 and the distributed extended Kalman filter (EKF) covered in [36] are used to localize the underwater target based on the collected field data. All of the data analysis in the following sections was post-processed using MATLAB.

### 4.3.1 Data Analysis

During the twenty-minute duration of the test, 77 measurements were received by node 1, 77 measurements by node 2, and 79 measurements by node 3. The TOA measurements at each node and the known asynchronous receiver clock offset were used to index which measurements were received by all nodes for the same acoustic ping instance. It was determined that 75 network TOA measurements were received, with the first TOA recording at 17:05:59 UTC and the last recording at 17:26:03 UTC. Figure 4.4 show that the majority of the 75 network TOA measurements

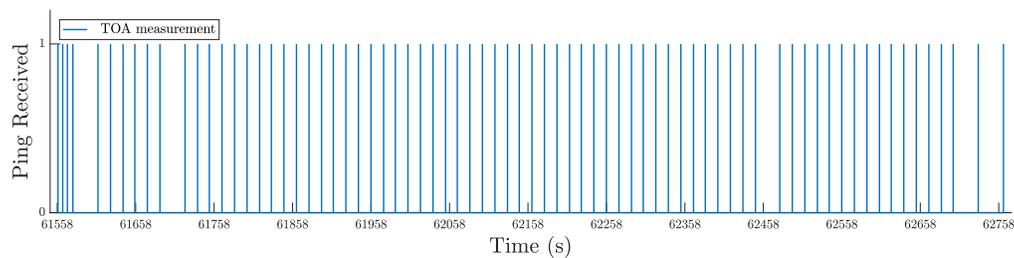


Figure 4.4: 75 network TOA measurements plotted on a linear x-axis. The vertical lines indicate the TOA of a ping given by the number of seconds after 00:00:00 UTC.

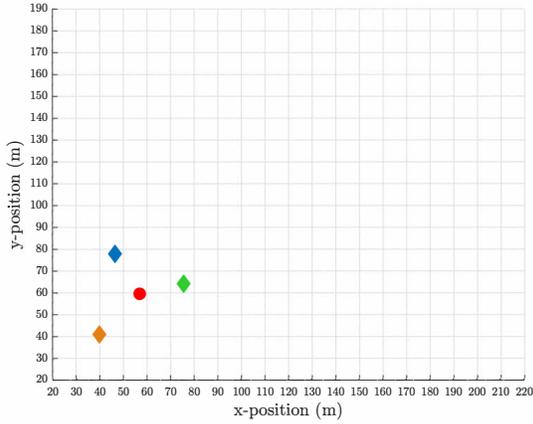


Figure 4.5: Initial formation of the USV network. Diamonds represent receiving nodes, and the circle represents the target.

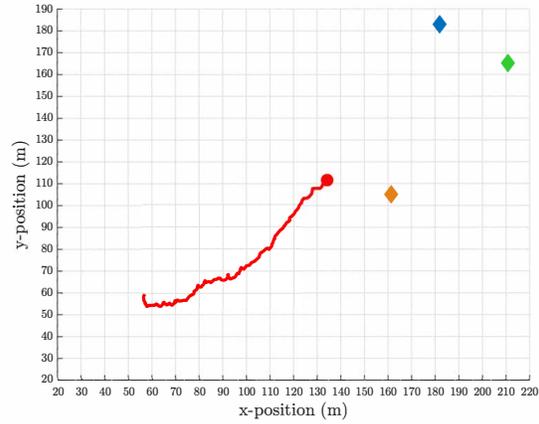


Figure 4.6: Final formation of the USV network. The solid line shows the drift trajectory of the target USV.

were received at a consistent time interval. For each TOA measurement, the corresponding GPS data for each node was indexed, converted to Cartesian coordinates using the origin point  $l_0 = \begin{bmatrix} 43.16436 & -85.85882 \end{bmatrix}$ , and used by the nodes to compute target location estimates using the distributed TDOA-based PF and EKF algorithms.

Figure 4.5 displays the initial GPS locations of the USVs which were converted to Cartesian coordinates using the origin point, and Figure 4.6 shows their final locations. We observe that all of the nodes drifted significantly over the course of the test despite the added weight, with the target USV trailing behind the receiving nodes. In implementing the PF and EKF algorithms, we assumed that the target drifts with a constant velocity corrupted with additive Gaussian noise. One of the largest sources of error for TDOA-based localization comes from asynchronous receiver clocks. The method developed for synchronizing the receiver clocks using GPS was determined to be inaccurate prior to the start of the test, and the receiver clocks varied up to 20 ms from each other, which resulted in large estimation error. To account for the offset in post-processing, a calibration method was used to calibrate the receiver TOA measurements. The calibration process is listed on the following page.

### **Clock Calibration:**

1. Choose node  $i$  as the *control*. The TOA measurement from this node will not be adjusted.
2. Calculate the TDOA measurements of the given node  $i$  at time-step  $k = 1$ .
3. Calculate the corresponding TDOA “ground truth” measurements at node  $i$  using the range difference between each node and the target (based on the GPS data at time-step  $k = 1$ ), divided by the propagation speed of the acoustic signal.
4. Determine a desired constant for each node  $j$ ’s TOA measurement so that node  $i$ ’s measured TDOA at  $k = 1$  is equivalent to node  $i$ ’s calculated TDOA “ground truth” at  $k = 1$ .
5. Add or subtract the desired constants to *all* TOA measurements at each node  $j$ .

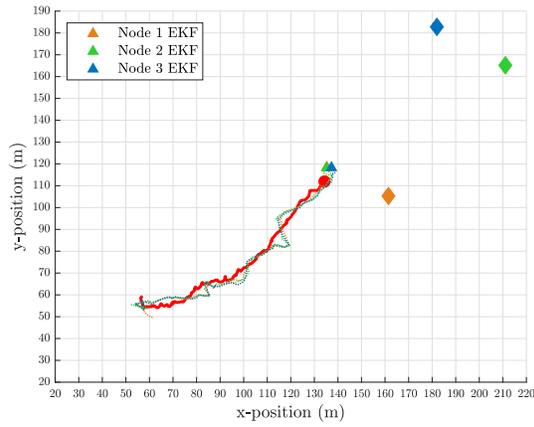
The receivers were programmed to update their internal clocks using GPS every four minutes during the experiment. This process resulted in fluctuations of the time offset between receivers due to inaccuracy of the GPS update method. To address this, the clock calibration method listed above was implemented once at the beginning of the experiment and was repeated every four minutes. For the initial calibration, node 3 was chosen as the control node, node 1 had 27 ms subtracted from every TOA measurement, and node 2 had 54 ms added to every TOA measurement.

### **4.3.2 Distributed Localization with a Fully Connected Topology**

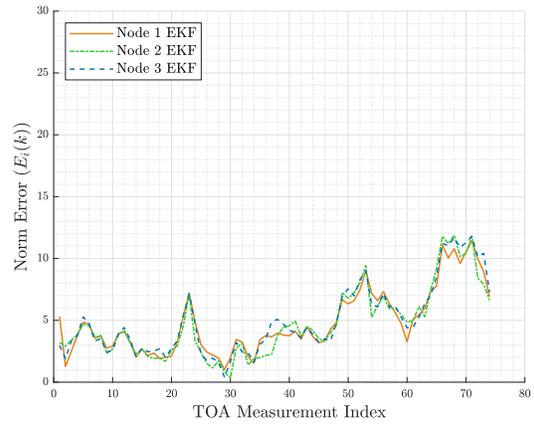
This subsection compares the localization performance of the distributed TDOA-based PF and EKF localization algorithms when different network topology conditions are considered – note the available TOA data from the experiments correspond to a complete graph. The target’s true starting location was at  $p(0) = [56.96, 59.53]$ . We first assume that the network is fully connected and that each node has a sufficient number of TDOA measurements to localize the target.

The PF initialized  $P = 1000$  particles distributed uniformly with  $c_i^{nx}(0), c_i^{ny}(0) \in [40, 70]$ . During the consensus step, each shared its top 5% of particles with neighboring nodes. The EKF randomly selected its initial position from a Gaussian distribution such that  $x_i^x(0), x_i^y(0) \sim \mathcal{N}(\mu =$

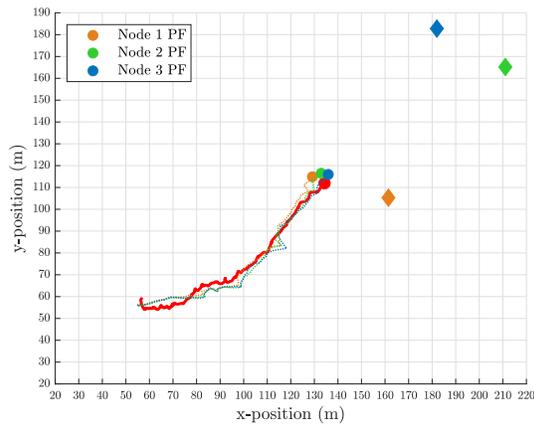
55,  $\sigma^2 = 4$ ) and initial velocity as  $\dot{x}_i^x(0), \dot{x}_i^y(0) = 0.1$ . We assume that the process noise of the target's movement model and the measurement noise are Gaussian with mean  $\mu_p = 0$ , standard deviation  $\sigma_p = 5$  and mean  $\mu_m = 0$ , standard deviation  $\sigma_m = 9$ , respectively. Figure 4.7 shows that the trajectory of the localization estimate for both the EKF and PF converges at each node. We observe that both TDOA-based algorithms accurately estimate the location of the target for the duration of the test. From Figure 4.7c and Figure 4.7d, we see that the estimation errors for the EKF and PF algorithms are mostly consistent between the two methods for the fully connected network topology. For each algorithm, the average execution time for a single iteration was recorded in MATLAB, and the experiment was repeated ten times to determine an average execution time. The average execution time for one iteration of the distributed EKF was  $2.68 \times 10^{-4}$  s, while the average execution time of the distributed PF was  $2.73 \times 10^{-1}$  s. Both algorithms were sufficiently fast for this particular application when taking into consideration that the acoustic tag transmits a ping every 7 s, despite the distributed EKF executing one iteration of the algorithm approximately 1000 times faster.



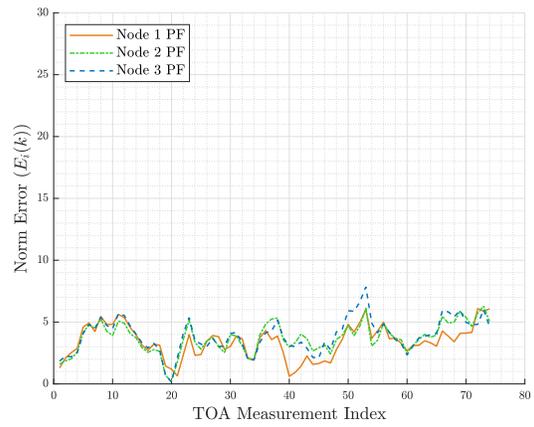
(a)



(b)



(c)



(d)

Figure 4.7: Fully Connected Topology – node EKF estimate’s trajectories (a) and accompanying estimation errors (b) as well as node PF estimate’s trajectories (c) and accompanying estimation errors (d). In figures (a) and (c) the diamonds represent sensing nodes, the lines connecting the diamonds represent communication links, the solid trajectory shows the target’s true trajectory, and each dashed trajectory shows the node estimate’s trajectory.

### 4.3.3 Distributed Localization under Other Network Topologies

Next, we investigated the distributed PF and EKF algorithms for different assumptions of the network's topology. In real-time estimation, the topology would have been dependent on the physical communication constraints of the network, but for the purpose of this analysis, the topology was predetermined prior to testing the algorithms. In particular, the performance of the distributed algorithms was evaluated for connected undirected graphs where at least one node did not have sufficient number of measurements directly.

Three nodes were used during the field experiment to collect measurements, which led to three different undirected graphs that satisfied the condition above, as shown in Figure 4.8. The estimation error for each node was evaluated for each of the possible configurations. Figure 4.9 shows the estimates' trajectory and each node's distributed PF and EKF estimation error for the given network topology. We observe that the estimate converged at each node for topology A, despite nodes 2 and 3 having an insufficient number of TDOA measurements. However, for topology B and topology C, as shown in Figure 4.10 and Figure 4.11, the distributed PF converged and the distributed EKF diverged. This led to the observation that the convergence of the TDOA-based distributed PF and EKF algorithms in two-dimensions, for minimal graph connectivity, may be dependent on the geometric formation of the nodes relative to the target. We note that in both cases where the distributed EKF estimates diverged, node 1's estimate followed a similar trajectory. The EKF estimate of node 1 could be driving the estimates of node 2 and node 3 away from the target's true location and towards a different equilibrium point [23].

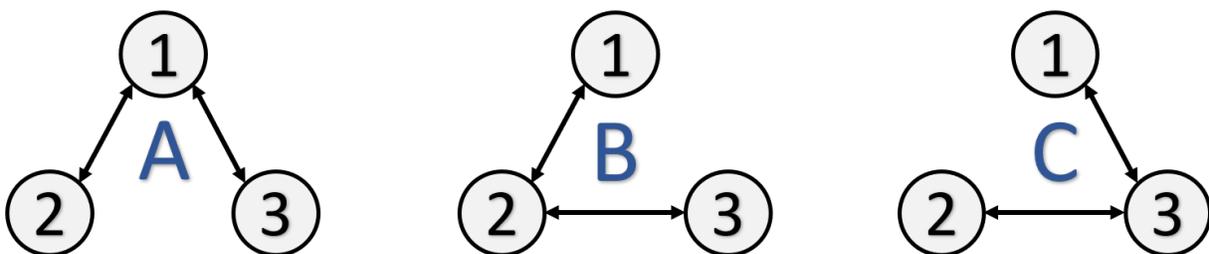
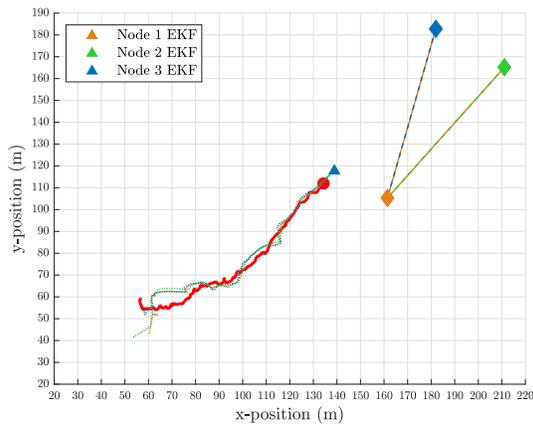
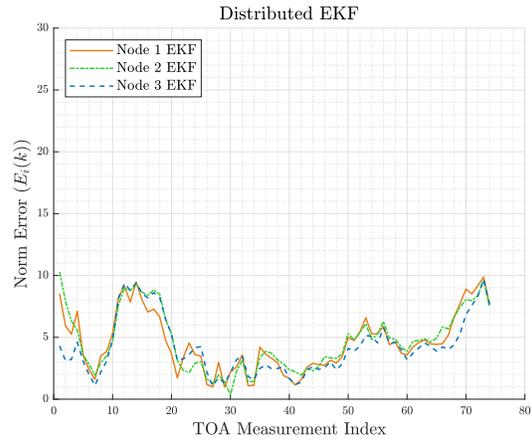


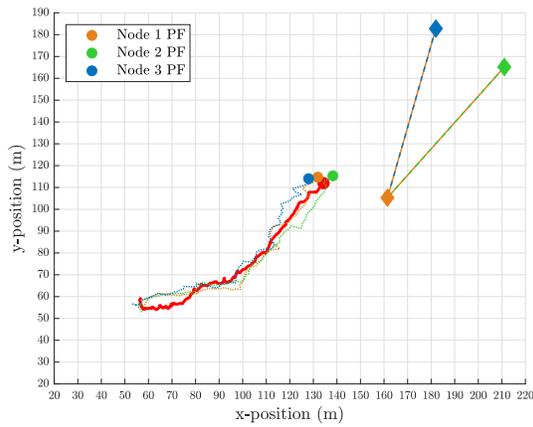
Figure 4.8: Three connected topologies where only one node is connected to two neighbors.



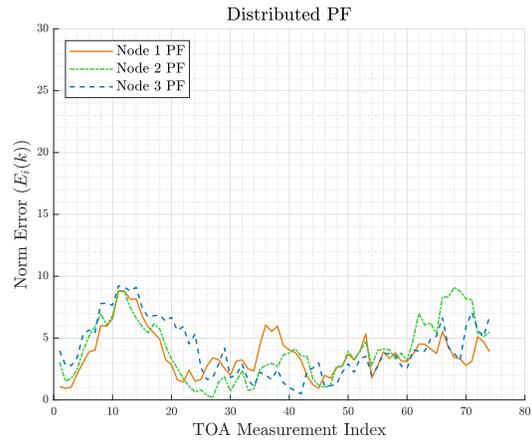
(a)



(b)

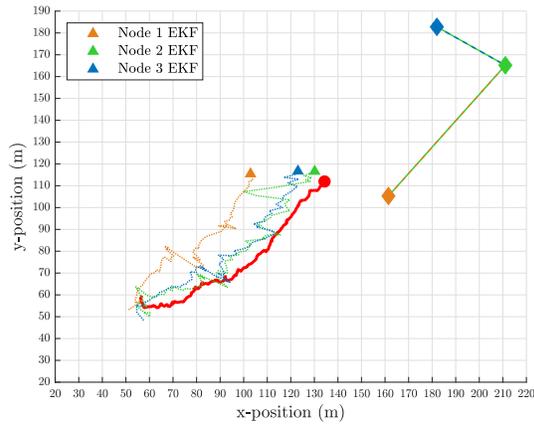


(c)

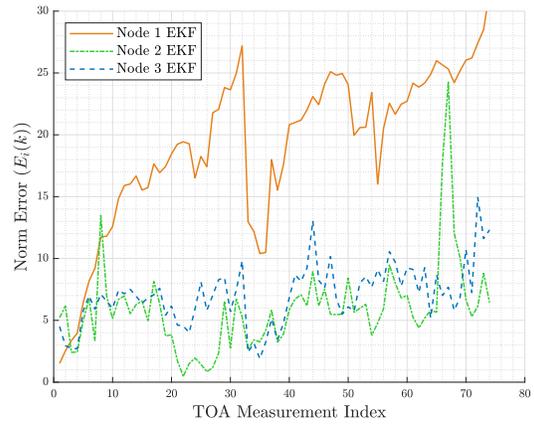


(d)

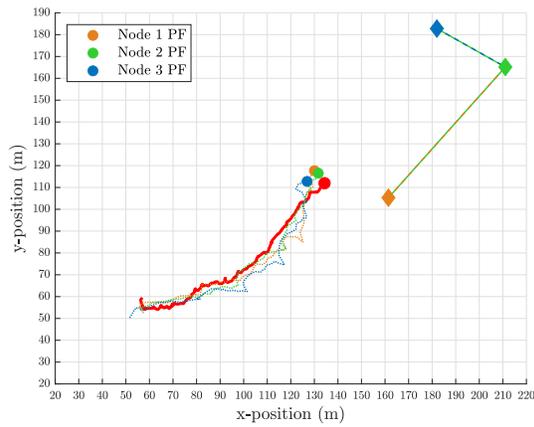
Figure 4.9: Topology A – node EKF estimate’s trajectories (a) and accompanying estimation errors (b) as well as node PF estimate’s trajectories (c) and accompanying estimation errors (d). In figures (a) and (c) the diamonds represent sensing nodes, the lines connecting the diamonds represent communication links, the solid trajectory shows the target’s true trajectory, and each dashed trajectory shows the node estimate’s trajectory.



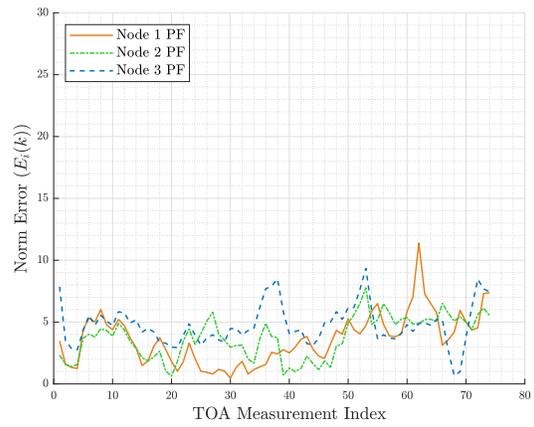
(a)



(b)

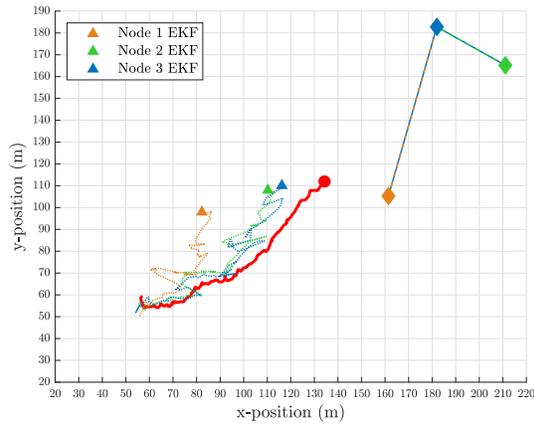


(c)

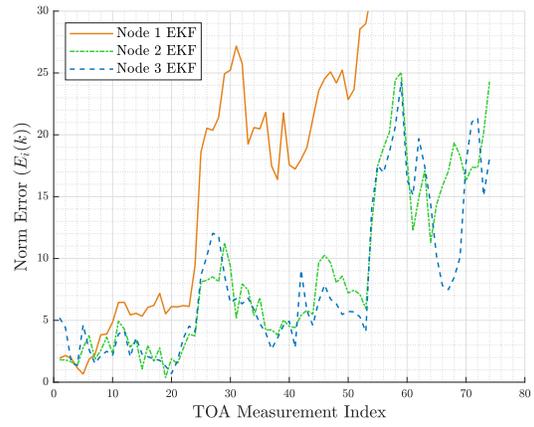


(d)

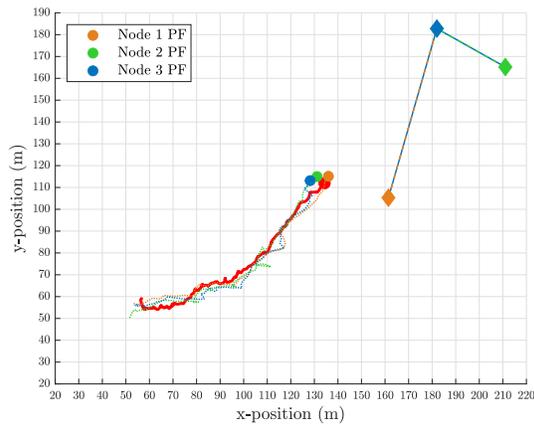
Figure 4.10: Topology B – node EKF estimate’s trajectories (a) and accompanying estimation errors (b) as well as node PF estimate’s trajectories (c) and accompanying estimation errors (d). In figures (a) and (c) the diamonds represent sensing nodes, the lines connecting the diamonds represent communication links, the solid trajectory shows the target’s true trajectory, and each dashed trajectory shows the node estimate’s trajectory.



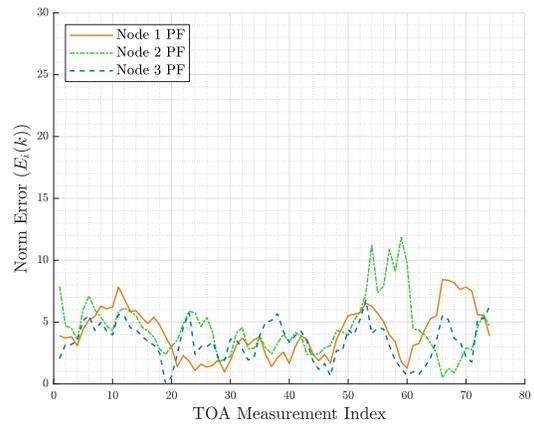
(a)



(b)



(c)



(d)

Figure 4.11: Topology C – node EKF estimate’s trajectories (a) and accompanying estimation errors (b) as well as node PF estimate’s trajectories (c) and accompanying estimation errors (d). In figures (a) and (c) the diamonds represent sensing nodes, the lines connecting the diamonds represent communication links, the solid trajectory shows the target’s true trajectory, and each dashed trajectory shows the node estimate’s trajectory.

To further test the convergence properties of the TDOA-based EKF using the experimental data, we established a *virtual node* that supplied an additional TDOA measurement to the network. The virtual node was created using a translation of node 2’s GPS data, and we assumed that its TDOA measurement was given by the range difference of the node and the target with additive noise. The measurement noise for the virtual node was assumed to be Gaussian with mean  $\mu = 0$  and variance  $\sigma^2 = 4$ . The virtual node established a connection with node 1 for topology B and C, and the estimation error at each node was analyzed. Figure 4.12 shows the trajectory and estimation error for topology B, and Figure 4.13 shows the trajectory and estimation error for topology C, with the addition of the virtual node connected to node 1.

We observed that the addition of the virtual node and the subsequent additional TDOA measurement caused each node’s estimate in the given topologies to converge, regardless of whether that node independently had a sufficient number of TDOA measurements. In this case, only two nodes in the network had a sufficient number of TDOA measurements. This shows that the addition of a fourth sensing node could improve the robustness of the estimate at each node even if the additional node only establishes one new connection with the network.

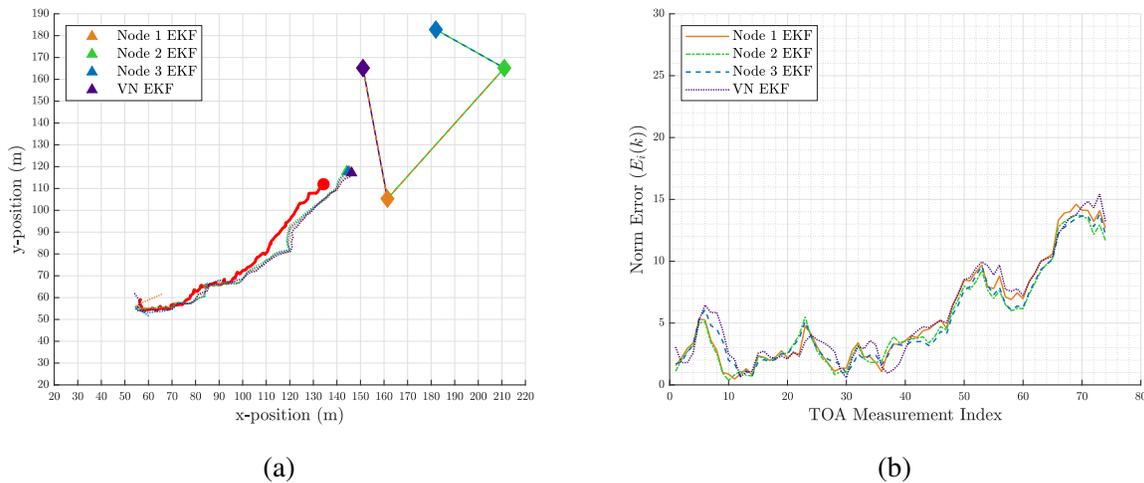


Figure 4.12: Topology B – Target and node EKF estimate’s trajectories (a) and corresponding node error (b) with the addition of the virtual node (VN).

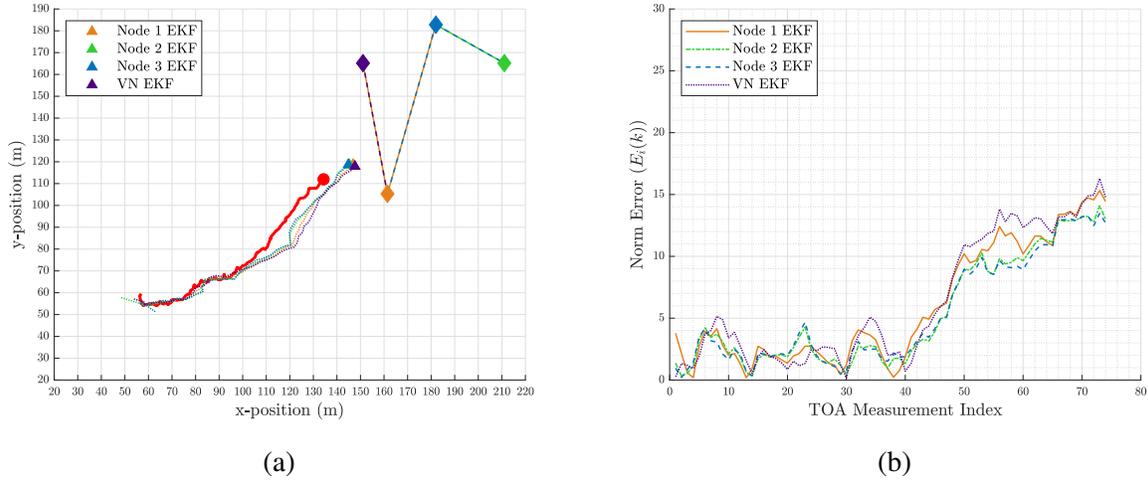


Figure 4.13: Topology C – Target and node EKF estimate’s trajectories (a) and corresponding node error (b) with the addition of the virtual node (VN).

#### 4.4 Distributed PF Localization - DCRW Target

In this section, the field test involving a target modeled by a discrete-time correlated random walk (DCRW) is described, and the evaluation of the distributed PF algorithm for target localization is presented. Prior to the experiment, a fifty time-step DCRW trajectory was computed using the methods outlined in Chapter 3. The x-y coordinates of the generated walk were converted to GPS coordinates using a GPS origin point on the lake, and the corresponding waypoint trajectory was loaded onto the PixHawk controller using QGroundControl. The target USV was programmed to loiter at each waypoint for fifteen seconds to allow for adequate time to receive an acoustic ping from the target, and the target USV navigated the waypoint trajectory autonomously. Receiving nodes were deployed in a triangular formation around the expected trajectory of the target USV to passively collect measurements during the duration of the walk.

The open-source autopilot traversed the waypoint trajectory continuously, without stopping at each time-step of the walk, despite efforts to control the wait-time at each waypoint. This resulted in only 14 consistent TOA measurements to be recorded at each node over the course of the fifty time-step DCRW. Figure 4.14 shows the QGroundControl window with the waypoint trajectory loaded, and Figure 4.15 shows the target’s true trajectory and the distributed PF estimate generated

from the TDOA measurements. The PF initialized  $P = 1000$  particles uniformly distributed such that  $c_i^{nx}(0), c_i^{ny}(0) \in [50, 90]$ , and we assumed a fully connected undirected graph.

The performance of the distributed TDOA-based PF for localizing a target modeled by the DCRW is hard to assess given the limited amount of TOA measurements recorded during the experiment. In order to properly judge the results of the simulation in Chapter 3, the target USV would have had to loiter at each waypoint long enough for the receiving nodes to collect a TOA measurement from the acoustic tag. Further experiments to address this problem would have been conducted, but the weather conditions in early December proved to be unfavorable for additional attempts. Despite the lack of TDOA measurements, the results from this experiment show that the USV is capable of autonomously navigating a waypoint trajectory modeled by a DCRW. Additionally, Figure 4.15 shows that the PF estimate at each node roughly follows the direction of the target USV and indicates promise for future DCRW target tracking and real-time animal tracking applications.

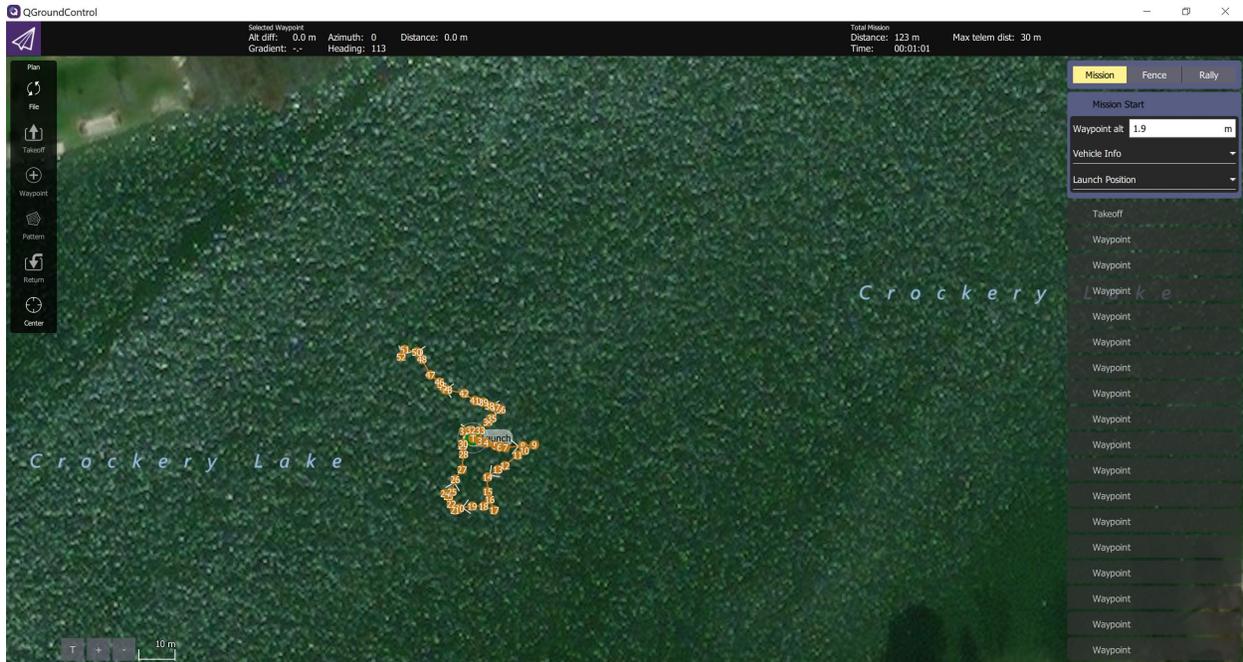


Figure 4.14: QGroundControl window showing the DCRW GPS waypoint trajectory loaded to the target USV. Circled numbers indicate sequential GPS waypoints.

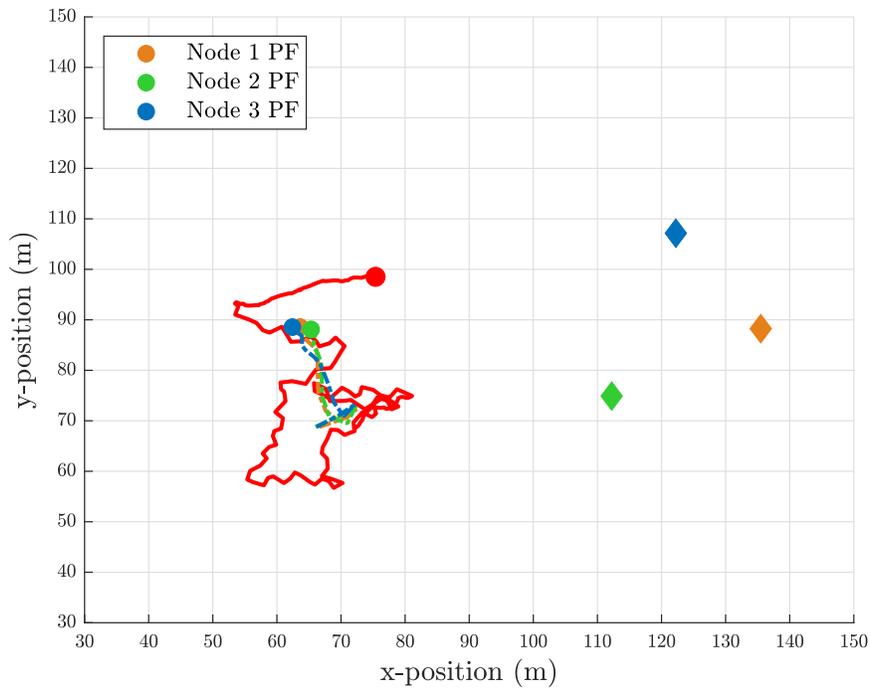


Figure 4.15: Distributed PF estimate trajectories for a target modeled by a DCRW. The solid trajectory shows the true path of the target, and the dashed trajectory shows each node's PF estimate.

## CHAPTER 5

### SUMMARY AND FUTURE WORK

#### 5.1 Summary

In this thesis, we presented a network of compact USVs capable of navigating a GPS waypoint trajectory autonomously. Each USV was designed to be low cost and have components that are easily sourced through commercial means. The design was based on an open-source platform that could be built and operated outside of the academic landscape. The USV incorporated a novel autonomous winch and sensor hub unit, allowing for a wide range of payloads to be deployed to a desired depth. We focused on the specific application of tracking an acoustic tag using acoustic telemetry equipment, and we proposed and conducted preliminary validation of receiver deployment using the USVs to improve fine-scale fish tracking in the Great Lakes region.

The natural movement of fish is governed by highly stochastic processes, and movement models are often difficult to determine for tracking. To address this issue, a distributed TDOA-based particle filter (PF) algorithm was proposed to localize a moving target modeled by a discrete-time correlated random walk (DCRW). Through numerical simulations, we showed that a node with an insufficient number of TDOA measurements could accurately localize the target as long as it was connected to a node with a sufficient number of measurements. We then presented a method for online model learning of the unknown parameters for the PDF distributions of the target's DCRW model. We observed that using the learned model to update the distributed PF algorithm improves target localization and provides real-time parameter estimates for the unknown target movement model.

Lastly, we presented the results on TDOA-based localization algorithms based on data collected from field experiments involving a network of USVs. Receivers were integrated with the USVs, and the USVs were deployed on open water to passively collect data. Through post-processing, we compared the performance of the distributed PF algorithm to that of the distributed EKF. We

compared different network topologies to show that convergence at all of the nodes is possible even if a node has an insufficient number of TDOA measurements, and we showed through an example that the addition of a fourth measurement node could improve the robustness of the distributed EKF algorithms. Additionally, we proposed a method for testing the distributed PF for a model governed by a DCRW, and we were able to show preliminary results to highlight the potential of the algorithm for nonlinear, non-Gaussian target tracking.

## **5.2 Future Work**

Future development of the USV includes advancing the design to create a single-hull, self-righting, fully waterproof USV capable of carrying and deploying sensors using the automated winch. This design would be able to handle harsher conditions of the Great Lakes and other coastal environments. In addition, with further development, a camera located in the sensor hub would enable other applications (e.g., search and rescue) for the USV.

The distributed PF can be extended to three-dimensions, and the optimal PF parameters to minimize computation cost shall be investigated. Additionally, it will be of interest to extend the results of online model learning to other random walk models and apply the methods discussed here to track live acoustically-tagged fish. These results could provide a framework to identify real-time movement models of various types of fish within their environment. A distributed control scheme for the USV network could be designed based on the target estimate to maintain an optimal formation and improve estimation performance.

Lastly, improved methods for testing the distributed PF algorithm for a target modeled by a DCRW shall be developed. The target USV could be programmed with a waypoint trajectory modeled by a DCRW walk and will loiter at the desired waypoint for a sufficient amount of time before executing further iterations of the walk. Online model learning will be employed to validate simulation results and to develop real-time estimates of the target's movement model. Further development of the network's communication protocol and technique for clock synchronization shall be investigated to allow for real-time estimates of the target's location.

## **BIBLIOGRAPHY**

## BIBLIOGRAPHY

- [1] 3RD. 2020. 3DR Drones. <https://www.3dr.com/>.
- [2] A. Nicosia, T. Duchesne, L. Rivest, and D. Fortin. 2017. A general hidden state random walk model for animal movement. *Computational Statistics & Data Analysis* 105. 76–95.
- [3] A. Thiercelin, I. Galloway, C. Schwarzmiller, and A. Klimaj. 2020. The Open Standards for Drone Hardware. <https://pixhawk.org/>.
- [4] A. Upadhyay. 2019. Formula to Find Bearing or Heading Angle Between Two Points: Latitude Longitude. <https://www.igismap.com/formula-to-find-bearing-or-heading-angle-between-two-points-latitude-longitude/>.
- [5] ArduPilot Dev Team. 2020. ArduPilot. <https://ardupilot.org/ardupilot/index.html>.
- [6] ArduPilot Developers. 2020. ArduPilot/pymavlink. <https://github.com/ArduPilot/pymavlink>.
- [7] Basement Engineering. 2020. Building a Self-Driving Boat (ArduPilot Rover). <https://www.instructables.com/Building-a-Self-Driving-Boat-ArduPilot-Rover/>.
- [8] BlueRobotics. 2020. BlueROV2 - Affordable and Capable Underwater ROV. <https://bluerobotics.com/store/rov/bluerov2/>.
- [9] C. Albertsen. 2019. Generalizing the first-difference correlated random walk for marine animal movement data. *Scientific Reports* 9(1). 1–14.
- [10] C. Panetta, O. Ennasr, and X. Tan. 2020. Distributed particle filter with online model learning for localization using time-difference-of-arrival (TDOA) measurements. In *ASME 2020 Dynamic Systems and Control Conference* 3305, 1–7.
- [11] D. Gu. 2007. Distributed particle filter for target tracking. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 3856–3861. IEEE.
- [12] D. Johnson, J. London, M. Lea, and J. Durban. 2008. Continuous-time correlated random walk model for animal telemetry data. *Ecology* 89(5). 1208–1215.
- [13] D. Laffoley, S. Hulme, B. Francis, B. Frinault, J. Baxter, and M. von Lindenfels. 2017. Exploring Ocean Risk: Hazards, Vulnerabilities, Global Priorities and Regional Resilience. Tech. rep. Swedish Ministry of the Environment and Energy.
- [14] E. Beck, W. Kirkwood, D. Caress, T. Berk, P. Mahacek, K. Brashem, J. Acain, V. Reddy, C. Kitts, J. Skutnik, and G. Wheat. 2008. SeaWASP: A small waterplane area twin hull autonomous platform for shallow water mapping. In *2008 IEEE/OES Autonomous Underwater Vehicles*, 1–7.
- [15] E. Codling, M. Plank, and S. Benhamou. 2008. Random walk models in biology. *Journal of the Royal Society Interface* 5(25). 813–834.

- [16] E. Gurarie. 2008. *Models and Analysis of Animal Movements: From Individual Tracks to Mass Dispersal*: University of Washington dissertation.
- [17] EPA. 2019. Physical Features of the Great Lakes. <https://www.epa.gov/greatlakes/physical-features-great-lakes>.
- [18] F. Gustafsson and F. Gunnarsson. 2003. Positioning using time-difference of arrival measurements. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)*, vol. 6, VI–553. IEEE.
- [19] GLATOS. 2020. <https://glatos.glos.us/glatos>.
- [20] Heron. 2020. Heron Unmanned Surface Vessel for Aquatic Research. <https://clearpathrobotics.com/heron-unmanned-surface-vessel/>.
- [21] iXblue. 2020. DriX: Unmanned Surface Vehicle. <https://www.ixblue.com/products/drix>.
- [22] J. Bailey, J. Wallis, and E. Codling. 2018. Navigational efficiency in a biased and correlated random walk model of individual animal movement. *Ecology* 99(1). 217–223.
- [23] J. Díez-González, R. Álvarez, L. Sánchez-González, L. Fernández-Robles, H. Pérez, and Castejón-Limas, Manuel. 2019. 3D TDOA Problem solution with four receiving nodes. *Sensors* 19(13). 2892.
- [24] J. Manley, A. Marsh, W. Cornforth, and C. Wiseman. 2000. Evolution of the autonomous surface craft AutoCat. In *OCEANS 2000 MTS/IEEE Conference and Exhibition. Conference Proceedings (Cat. No.00CH37158)*, vol. 1, 403–408 vol.1.
- [25] J. Pepin. 2020. Showcasing the DNR: Tracking Fish Movements with Acoustic Telemetry. <https://content.govdelivery.com/accounts/MIDNR/bulletins/2981905>.
- [26] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys. 2011. PIXHAWK: A system for autonomous flight using onboard computer vision. In *2011 IEEE International Conference on Robotics and Automation, 2992–2997*.
- [27] L. Zhixiang, Z. Youmin, Y. Xiang, and Y. Chi. 2016. Unmanned surface vehicles: An overview of developments and challenges. *Annual Reviews in Control* 41. 71 – 93.
- [28] Liquid Robotics. 2020. The Wave Glider: How It Works. <https://www.liquid-robotics.com/wave-glider/how-it-works/>.
- [29] M. Coates. 2004. Distributed particle filters for sensor networks. In *IPSN '04: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, 99–107. Association for Computing Machinery.
- [30] Manley, J. 2008. Unmanned surface vehicles, 15 years of development. In *OCEANS 2008*, 1–4.
- [31] Mtc. 2020. Subsea, Offshore and Life Support Solutions. <https://www.uniquegroup.com/autonomous-survey-vessels>.

- [32] Naval Drones. 2017. Modular Unmanned Surface Craft Littoral (MUSCL). <http://www.navaldrones.com/MUSCL.html>.
- [33] O. Ennasr. 2020. *Gliding Robotic Fish: Design, Collaborative Estimation, and Application to Underwater Sensing*; Michigan State University dissertation.
- [34] O. Ennasr and X. Tan. 2017. Distributed estimation and tracking using time-difference-of-arrival (TDOA) measurements. In *ASME 2017 Dynamic Systems and Control Conference*, 1–8.
- [35] O. Ennasr and X. Tan. 2018. Distributed localization of a moving target: Structural observability-based convergence analysis. In *2018 Annual American Control Conference (ACC)*, 2897–2903.
- [36] O. Ennasr and X. Tan. 2020. Time-difference-of-arrival (TDOA)-based distributed target localization by a robotic network. *IEEE Transactions on Control of Network Systems* 1–1. doi:10.1109/TCNS.2020.2979864.
- [37] O. Ennasr, C. Holbrook, D. Hondorp, C. Krueger, D. Coleman, P. Solanki, J. Thon, and X. Tan. 2020. Characterization of acoustic detection efficiency using a gliding robotic fish as a mobile receiver platform. *Animal Biotelemetry* 8(1). 1–13.
- [38] O. Ennasr, G. Xing, and X. Tan. 2016. Distributed time-difference-of-arrival (TDOA)-based localization of a moving target. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2652–2658.
- [39] Office of Naval Research. 2020. Unmanned Surface Vehicle and Small Combatant Craft. <https://www.onr.navy.mil/en/Science-Technology/Departments/Code-33/All-Programs/331-advanced-naval-platforms/unmanned-surface-vehicle>.
- [40] P. Wu, X. Li, and Y. Bo. 2013. Iterated square root unscented Kalman filter for maneuvering target tracking using TDOA measurements. *International Journal of Control, Automation, and Systems* 11(4). 761–767.
- [41] R. Moll, C. Sellinger, E. Rutherford, J. Johnson, M. Fainter, and J. Gannon. 2013. *The Great Lakes: An Overview of Their Formation, Geology, Physics, and Chemistry* 3–30. Michigan State University Press.
- [42] T. Binder, S. Farha, H. Thompson, C. Holbrook, R. Bergstedt, S. Riley, C. Bronte, J. He, and C. Krueger. 2018. Fine-scale acoustic telemetry reveals unexpected lake trout, *Salvelinus namaycush*, spawning habitats in northern Lake Huron, North America. *Ecology of Freshwater Fish* 27(2). 594–605.
- [43] Water Linked. 2020. Underwater GPS G2. <https://waterlinked.com/underwater-gps-g2/>.
- [44] X. Sheng and Y. H. Hu. 2005. Distributed particle filters for wireless sensor network target tracking. In *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 4, iv/845–iv/848 Vol. 4. IEEE.

- [45] X. Su, I. Ullah, X. Liu, and D. Choi. 2020. A review of underwater localization techniques, algorithms, and challenges. *Journal of Sensors* 2020(6403161). 24.
- [46] X. Zhong and J. Hopgood. 2014. Particle filtering for TDOA based acoustic source tracking: Nonconcurrent multiple talkers. *Signal Processing* 96. 382–394.
- [47] Y. Huang, W. Liang, H. Yu, and Y. Xiao. 2008. Target tracking based on a distributed particle filter in underwater sensor networks. *Wireless Communications and Mobile Computing* 8(8). 1023–1033.