Fall 2023

Framework for Implementing Advanced Radar Plotting Aid Capability for Small Maritime Vessels

Jason Stark Harris
Old Dominion University, jason_s_harris@outlook.com

FRAMEWORK FOR IMPLEMENTING ADVANCED RADAR PLOTTING AID CAPABILITY FOR SMALL MARITIME VESSELS

by

Jason Stark Harris
M.S. August 2016, Old Dominion University
B.S. December 2014, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
December 2023

Approved by:

Dimitrie C. Popescu (Director)

Lee Belfore (Member)

Otilia Popescu (Member)

Chungsheng Xin (Member)

ABSTRACT

FRAMEWORK FOR IMPLEMENTING ADVANCED RADAR PLOTTING AID CAPABILITY FOR SMALL MARITIME VESSELS

Jason Stark Harris
Old Dominion University, 2023
Director: Dr. Dimitrie C. Popescu

Every year in the United States many people are killed or injured when maritime vessels collide with other vessels or fixed objects. According to the United States Coast Guard, the top contributing factors to these collisions are operator inattention, operator inexperience and an improper lookout. Larger commercial vessels are required to have RADAR systems which support Automatic RADAR Plotting Aid (ARPA) which can automatically detect collisions and alert an operator to change course. These systems can be very expensive which put them out of reach of the average recreational boater. It is however possible to implement a low cost ARPA like system which is enabled by commercial marine RADAR and open source software. This dissertation presents a framework which can be used to develop an ARPA like system. There are two main problems that would be encountered that this dissertation addresses. The first problem is automatically extracting the targets from a standard commercial RADAR. Most modern RADAR systems are network enabled and send data back to a display using a standard Ethernet interface. All of the main vendors transfer the data using proprietary formats. Some vendors offer software developer kits and some open source projects, such as OpenCPN, have the ability to communicate with the RADAR systems and decode the data streams. Once the data is received from the RADAR, open source computer vision software such as OpenCV can be used to perform the target extraction. A discussion about instrumentation needed to make sure that those targets are appropriately converted to geographic coordinates is also performed. The second main problem is how to implement the tracking algorithm. The state of the art for general purpose tracking is the Multiple Hypothesis Tracking (MHT) algorithm. The MHT algorithm requires a state estimator so that it can predict where a target will be in the future. The predominate state estimator used by MHT has been the Kalman filter. This dissertation explores the use of a Particle Filter along with the Kalman Filter. A scenario where two boats pass by an observer vessel is conducted and results are analyzed and discussed.

In memory of my parents, Norman and Cindy

ACKNOWLEDGEMENTS

I technically met and was mentored by Dr. Dimitrie Popescu while I was working on my associates degree before I started at Old Dominion University. When I finally started at ODU, I thought that getting a Bachelor's degree was going to be a huge opportunity for me and the thought of completing a PhD was a dream that I never thought I would achieve. I am extremely grateful for Dr. Popescu giving me the opportunity and guidance to complete my Bachelors, Masters and PhD at ODU. I am also extremely grateful for the gratuitous amount of time and encouragement that Dr. Robert Ash and Dr. Lee Belfore have given me over the many years I have been a student at ODU. I would to thank Dr. Chungsheng Xin for the support he has given me, the equipment he has let me borrow and space he has let me occupy in the lab while I have been in graduate school. I would like to thank Dr. Otilia Popescu for her mentorship and the time that she has spent co-authoring papers with me.

I have been privileged to have the support of many friends and colleagues throughout my academic journey. Two that must be mentioned are Michael Hayden and Dr. Stephanie Cronin who have endured listening to me ramble about my studies for longer than anyone should have. I am also very grateful to Paul Buckwalter, K4PRB, and Barry Priddy, K5VIP, for originally convincing me to pursue a degree in Electrical Engineering and encouraging me throughout the years.

Lastly, there is no way that I could have completed this dissertation without the support of my lovely and patient wife Jennifer. Working full time while in graduate school has been very stressful, but together we have endured.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Operating a vessel on the water can be dangerous. High speeds combined with poor situational awareness lead to many accidents. In 2022, the United States Coast Guard stated that there were 1,085 accidents where a vessel collided with a recreational vessel leading to 39 deaths and 512 injuries[3]. There were also 477 collisions between a vessel and a fixed object leading to 57 deaths and 314 injuries[3]. The top three known primary contributing factors for accidents were operator inattention, operator inexperience, and an improper lookout[3]. In total, these three primary factors were the cause of 1,453 accidents leading to 136 deaths and 791 injuries[3]. While there is no 100% solution to reduce accidents on the water, technology can be used to augment the skills and capabilities of boat operators.

A marine RADAR for small craft is a very important tool when it comes to safety. Small Marine RADAR can be used to navigate on the water in adverse conditions such as fog or torrential downpours when navigation by sight becomes no longer possible. One feature that has been missing in small marine RADAR is the Automatic RADAR Plotting Aid (ARPA) which is implemented on much larger ships and are typically called ARPA RADAR. ARPA enables a RADAR to automatically detect nearby obstacles, plot their course, and warn an operator in the event that it detects that a collision may occur. If this technology was more available and affordable, it is theorized that more vessels would be equipped with them and accidents on the water would be reduced.

Small maritime vessels, such as personal watercraft, as defined in this dissertation are approximately 60ft in length or less. These types of watercraft are generally owned by individuals for recreational purposes and are able to be trailer-ed around the country. Commercial off-the-shelf RADAR available for these types of watercraft, are much smaller, and are only a few thousand dollars. What is missing with these RADAR is they do not have the ability to perform the ARPA functionality to automatically detect targets and warn the operator if a collision may occur. The purpose of this thesis is to describe a framework by which this capability can be implemented.

1.1 SMALL VESSEL DYNAMICS

Implementing an ARPA capability on marine RADAR for small vessels is faced with several challenges. One of the main challenges is being able to provide an accurate platform heading to the tracking algorithm. The dynamic nature of the vessel operating on the water leads to uncertainties regarding the vessel's orientation with respect to the yaw axis. These uncertainties end up being represented as noise in the tracking process which if not suppressed can lead to failure of the tracking algorithm to successfully track targets.

Once targets are detected, maintaining tracks with a RADAR mounted on a vessel can prove to be challenging. The reason for this is that movements of the craft on the water due to wave action increase the co-variance matrix surrounding observations leading to larger uncertainties about the precise location of detected targets. These effects are much more problematic on smaller vessels and those that do not have high quality instruments to take vessel bearing measurements. Solving these problems would lead to the ability to put affordable small marine RADAR with collision detection capability on more vessels leading

to a safer environment both inshore and at sea.

If the yaw rate rotation on a vessel is not adequately measured and incorporated correctly into the system, this rotation can cause the RADAR system on board the craft to measure targets relatively far away from their actual location. If this error becomes very large, the tracking system will ultimately start to degrade resulting in extremely poor or unusable performance.

There is very little published literature concerning the yaw rate of vessels on the water. The primary reason for this is that Naval Architecture is primarily concerned with sea keeping ability of a craft. A vessel has appropriate sea keeping ability if it is able to make way in the desired maximum sea state and is more of a qualitative assessment. How a vessel's heading is changing at the sub second scale is considered irrelevant, but that is exactly what needs to be known when working with a RADAR tracking system. One of the few papers published with real world measurements was written by Nana Abankwa *et al.* and during a particular wave slam their vessel experienced a yaw rate change of 45° per second [4]. Unfortunately, they didn't describe any characteristics of the vessel which would have been very useful, but it can be assumed that it was on the smaller size. Yaw measurements of a buoy with an HF RADAR system installed on it were taken and discussed with respect to correction for direction of arrival [5]. However, these measurements were taken over a period of 5 minutes and could not be used as a basis to determine the motion of a vessel under way.

To explore the effect of yaw rate variation on RADAR tracking performance this dissertation simulates a vessel experience different yaw rate rotations. To do so a system will

be developed using a Multiple Hypothesis Tracker. The yaw rate will be varied in degrees and a Monte Carlo simulation will be employed to determine where the tracker is no longer capable of maintaining a track.

1.2 HEADING SENSORS

While there is little empirical data which shows the yaw rate variations of a small vessel operating on the water, this dissertation will discuss that inexpensive consumer grade MEMS sensors are at the limit of being useable even on a non maneuvering platform.

It is important to note that this dissertation will focus primarily on micro-electromechanical systems (MEMs) based sensors and not more advanced sensor types such as Fiber Optic Gyroscope and Laser Ring Gyroscopes. This is due to the fact that MEMs based devices are affordable to the point where they are in almost every cell phone. The latter type devices are extremely expensive and would make the cost of implementation unreasonable.

MEMS sensors can be constructed using several different techniques and are considered to be low cost devices[6]. MEMS technology started to become popular in the 1990s due to their use in the automotive industry[6]. MEMS technology can be used to construct accelerometers, gyroscopes, and magnetometers.[7]. The most basic diagrams of the construction of these types of sensors is shown in Fig-1. The accelerometer is constructed by using a mass supsended between two small springs. The mass acts as a capacitive plate and as its relationship changes with respect to a fixed plate, the change in capacitance can be converted to an acceleration value[8]. Two accelerometers can be used to form a gyroscope where the acclerometer furthest from the center of rotation will register a larger acceleration. A torsional magnetometer can be formed by building a plate with a coil suspended by a

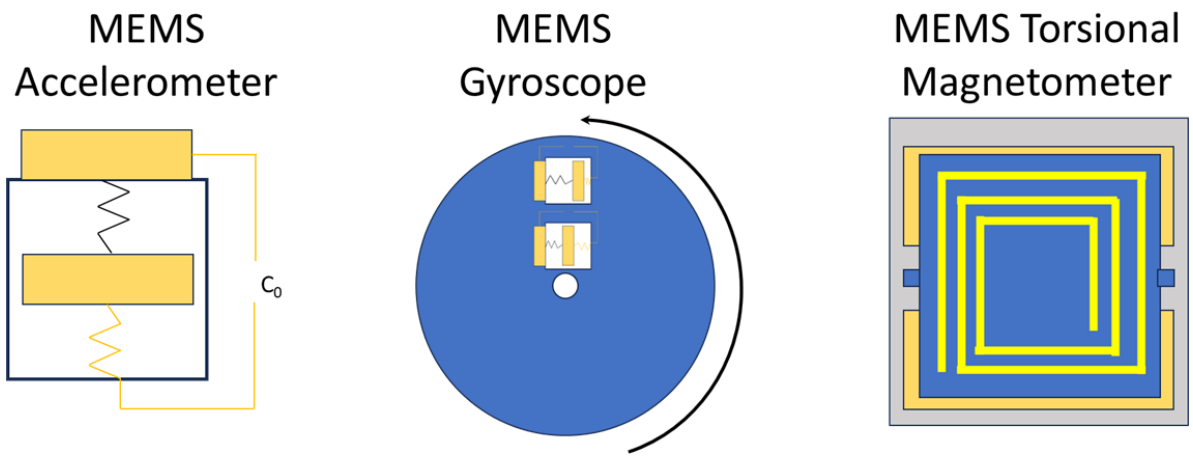

Fig. 1: Basic Diagrams of MEMS Accelerometers, Gyroscopes and Torsional Magnetometers

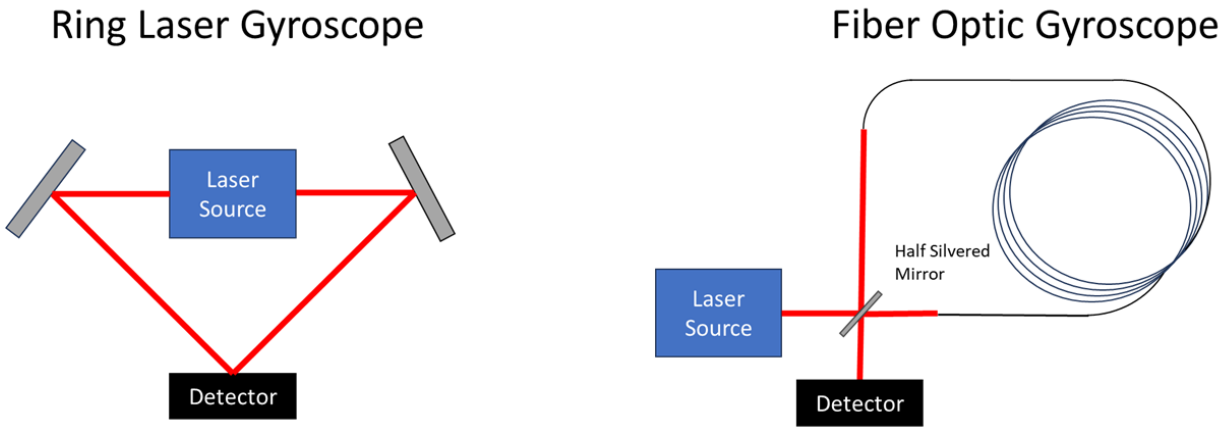

Fig. 2: Diagram of a Ring Laser Gyroscope and Fiber Optic Gyroscope

torsion bar. When a current is passed through the coil of wire, the Lorentz force will apply a torsion to the plate altering the capacitance with respect to two fixed capacitive plates underneath [9].

To determine orientation of an object such as a vessel or a gaming controller, low cost Inertial Measurement Units (IMUs) are starting to reach relatively good performance [10]. An IMU is a device that packages together magnetometers, gyroscopes, and accelerometers

Device	Magnetometer		Gyroscope	
	Update Rate	Resolution	Update Rate	Resolution
MPU-9250	100 Hz	0.59 uT	8000 Hz	$0.0076°/s$
ICM-20948	100 Hz	0.15 uT	9000 Hz	$0.0076°/s$
BNO055	30 Hz	0.3 uT	400 Hz	$0.0038°/s$

TABLE 1: Popular consumer grade Inertial Measurement Units and their Magnetometer and Gyrsocope specifications.

together which are able to complement each other for determining orientation. Magnetometers are able to determine true measurement with respect to magnetic North, but they are relatively slow and not very precise. Gyroscopes are incapable of determining where magnetic North is, but they are faster and more precise. Accelerometers can be used to determine orientation of the sensor platform. The specifications for some popular IMUs are shown in Table-1.

Sensor Fusion algorithms are able to fuse the magnetometer, gyroscope, and accelerometer data together to achieve an absolute heading resolution of approximately 1° rms[11]. The main limiting factor for getting accurate heading data is calibration of the sensors and non proprietary open fusion algorithms such as Madgwick's algorithm perform almost as well as proprietary algorithms [11].

High end MEMS based IMUs are able to achieve static heading resolutions of 0.05° with dynamic heading resolutions of around 1°[7]. Ring Laser Gyroscopes (RLGs) and Fiber Optic Gyroscopes (FOGs) have much better performance, but at a much higher cost[12].

Fiber optic gyroscopes can have dynamic heading resolutions of 0.1°[13].

RLGs and FOGs work on the same basic principle which is the Sagnac effect demonstrated by Georges Sagnac in 1913 [14]. The Sagnac effect states that a rotation will induce a phase change in two counterpropagating beams of light which can be measured using an interferometer[14]. The first RLG was demonstrated in 1963 and the first FOG was demonstrated in 1976 [14].

1.3 SIGNAL PROCESSING

In order to perform target detection on data from small marine RADAR and to display it there are three functional blocks. The first function is the RADAR which works by sending electromagnetic pulses out into the environment and listening for their echos. The second function is the processing system which takes the received information from the RADAR and analyzes it to find targets. The last function is the display function which would present the results to the user. This functional architecture is shown in Fig. 3. Implementation of these functions would allow the design of a system which is capable of detecting other vessels and alerting the operator. A comparison of using Kalman Filtering vs Particle Filtering for state estimation will also be performed to determine which may perform better for this application.

Particle Filtering is a technique where a state estimate is formed by using a particle model. Statistical calculations can be performed based on the particles characteristics to determine the estimate of a state and to determine the uncertainty of a state. Each particle is propagated through a state transition model to predict what the next state may be. These propagated particles are then compared when a new real world observation is available. Some

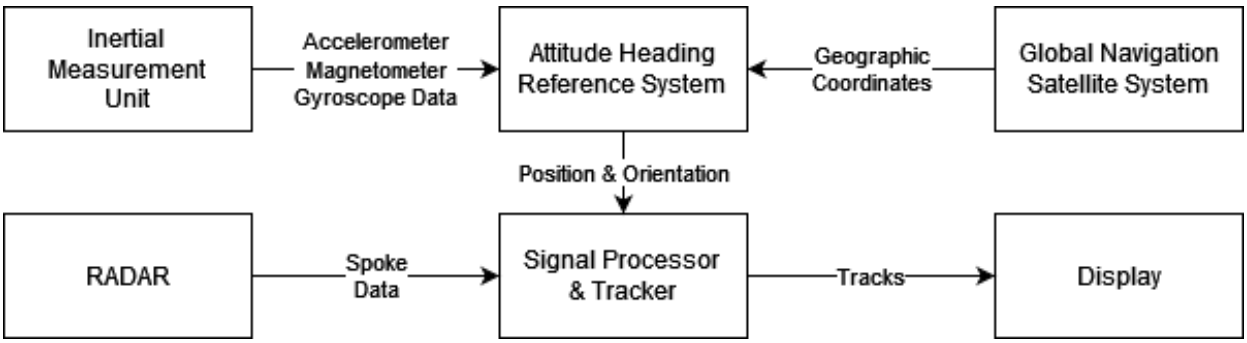

Fig. 3: Functional Architecture of the Proposed System

of the particles are removed if they likely no longer represent the state and new particles are generated based off particles which more likely represent the state.

While Particle Filtering is discussed somewhat widely in literature, there are very few discussions about how best to appropriately implement an advanced version of it. *Elfring et. al.* does an excellent job in describing the steps that need to be taken and also the issues that may arise when implementing a particle filter [15]. There are four main issues identified that impact the analysis in this dissertation. They are Degeneracy, Sample Impoverishment, Divergence, and selecting an proper Importance Density.

Degeneracy is where as the particle filter is operated, only one particle ends up being effective and the rest of the particles become irrelevant. The solution to the degeneracy problem is resampling the particles in the particle filter. The issue with resampling too rapidly is that the filter may take awhile to converge and could fall apart. An example of degeneracy is shown in Fig-4.

After resampling, the particle filter ends up with duplicate samples meaning that there

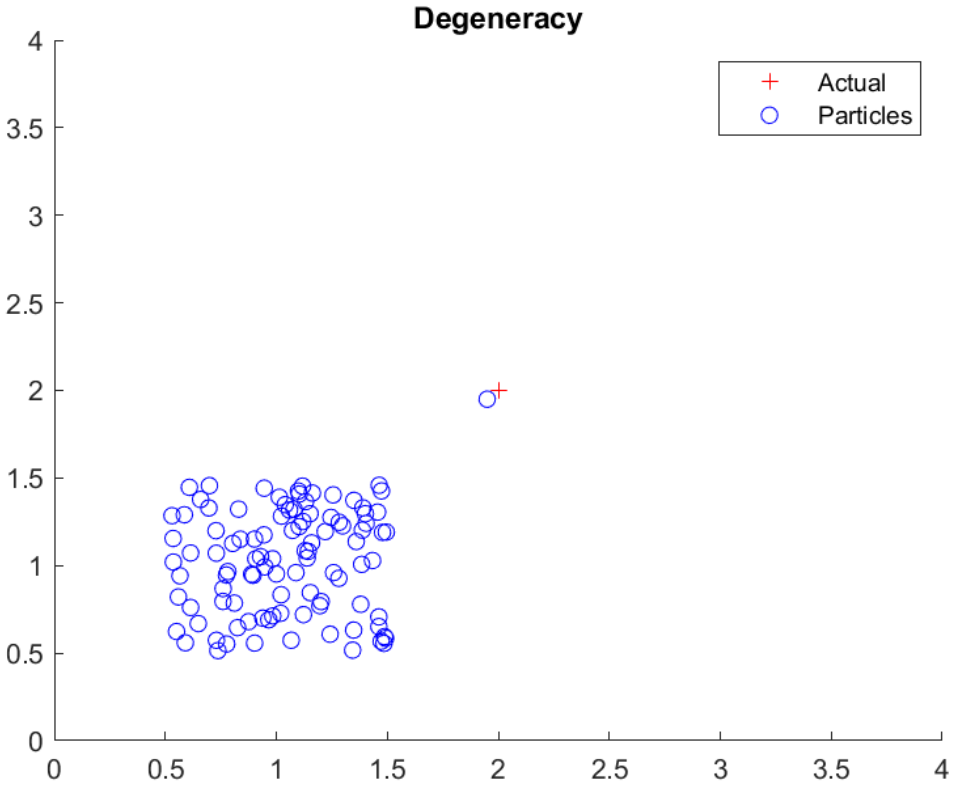

Fig. 4: A Particle Filter Implementation Exhibiting Degeneracy

are again fewer opportunities to express the possible state which is called Sample Impoverishment. The solution to Sample Impoverishment is to apply process noise to all of the particles in the particle filter on ever iteration. However, too much noise will cause the particle filter to diverge. An example of this scenario is shown in Fig-5.

Divergence is where particles grow apart from the true state and then none of the particles are able to be representative. This is due to too much noise or due to an improper state transition equation. An illustration of a scenario where divergence occurs is Fig-6.

The Importance Density function determines how representative a particle represents the

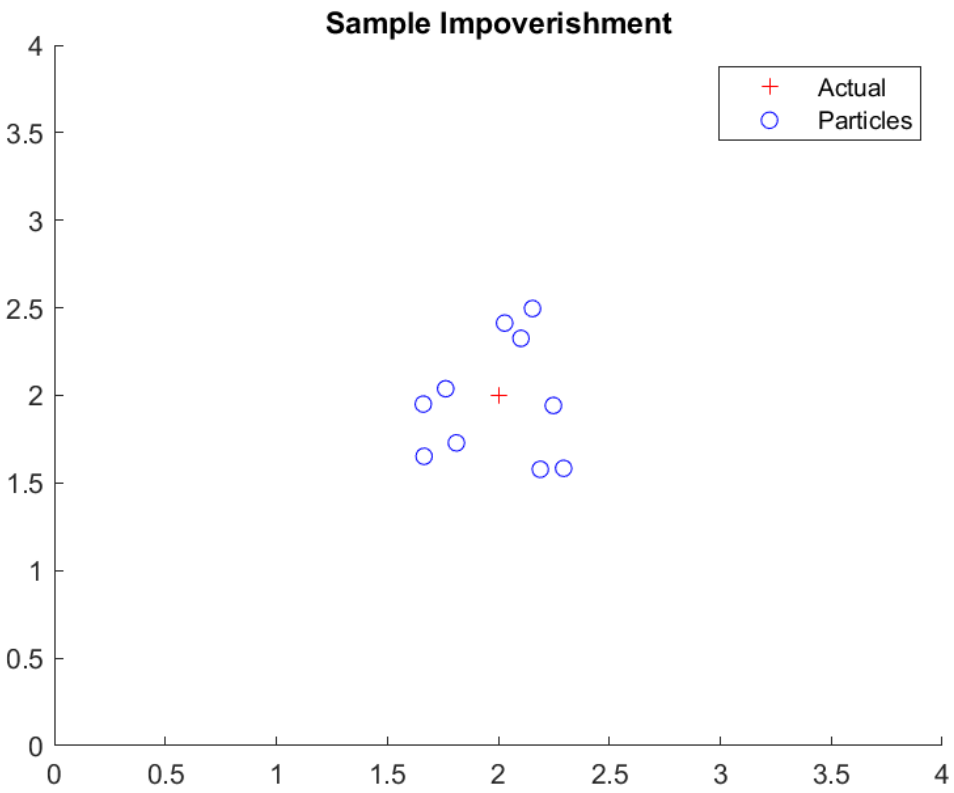

Fig. 5: A Particle Filter Implementation Exhibiting Sample Impoverishment

state. Choosing a sub-optimal importance density function leads to a lack of convergence of the filter and ultimately the filter will collapse.

1.4 DISSERTATION ORGANIZATION AND CONTRIBUTIONS

This dissertation is composed of proposed solutions to two main problems encountered when implementing ARPA style RADAR for small marine vessels. There is very little literature discussing some of these topics and it is a goal to make this dissertation to be able to be used as a reference for others who wish to expand and improve upon this area of work.

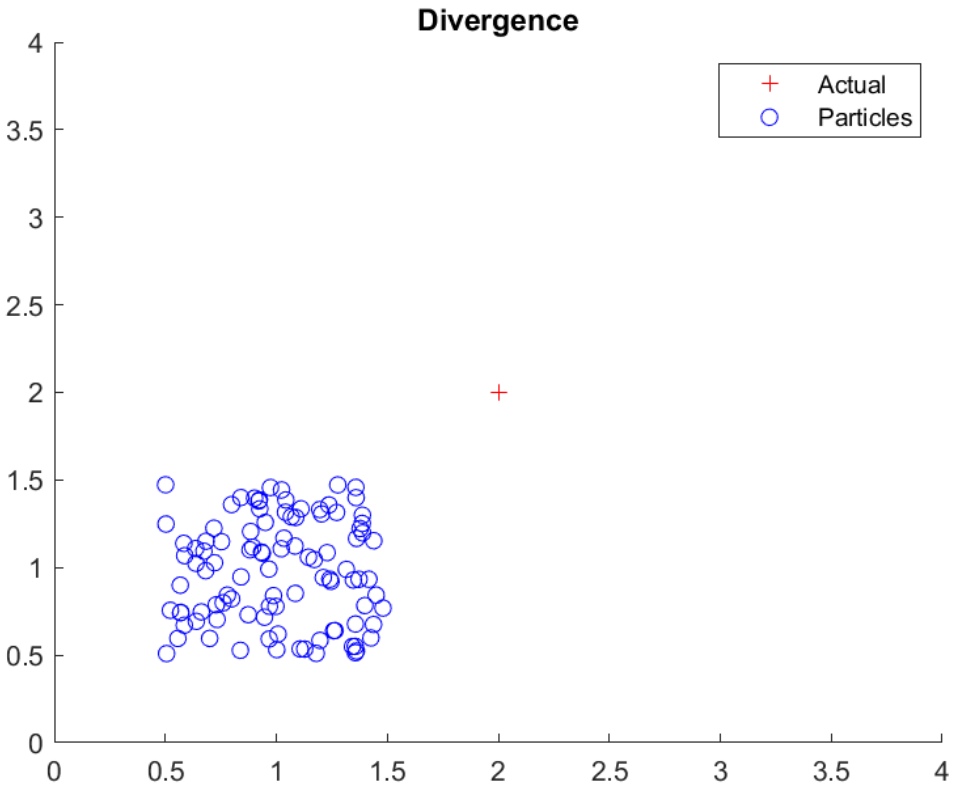

Fig. 6: A Particle Filter Implementation Exhibiting Divergence

The first problem that is encountered when trying to implement this system is how to extract targets from the RADAR. Most marine RADAR present a raster display of their output. This means instead of getting raw RADAR signal data, the data is output in spoke format where an amplitude is given for every range possibility in a particular direction. This becomes a computer vision problem and a proposed solution to extracting the targets from the raster display is given in Chapter 3 which has been published[16].

The second problem that is encountered is how to track the extracted targets from the RADAR data. When it comes to RADAR tracking, one of the most advanced tracking

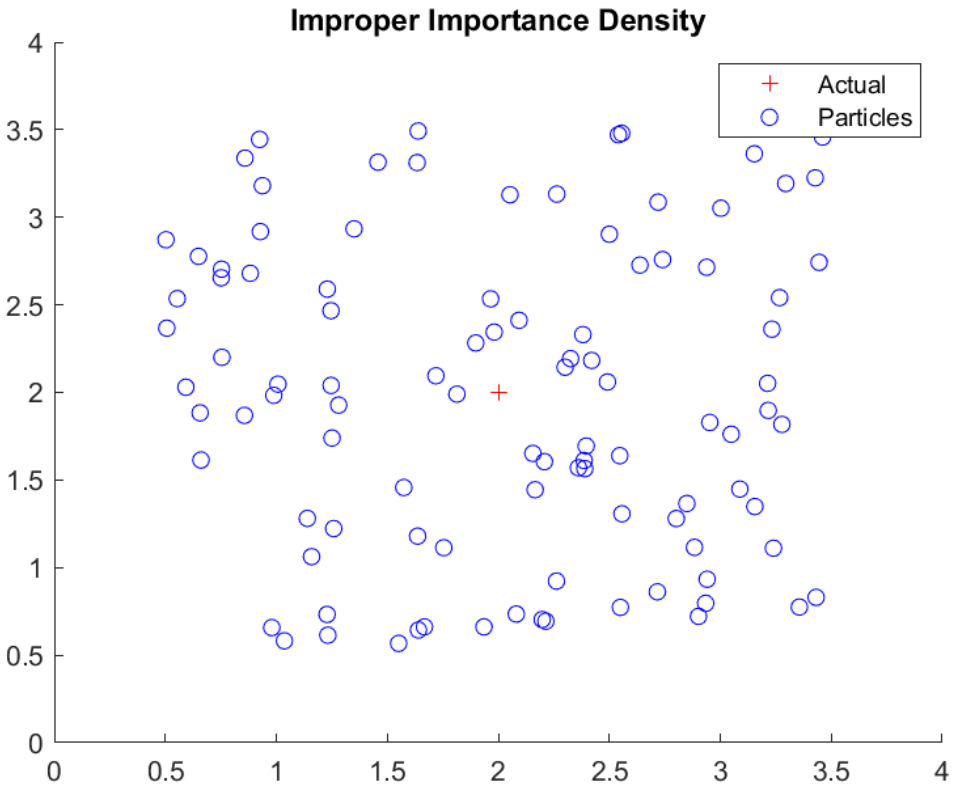

Fig. 7: A Particle Filter Implementation Exhibiting Issues with Improper Importance Density Function

algorithms is based on Multiple Hypothesis and is called Multiple Hypothesis Tracking. This means that the tracker considers that every observation could be noise, could be the start of a new track or could be part of an existing track. Multiple Hypothesis Tracking requires the ability to be able to propagate tracks into the future. There are two techniques that are considered more advance for performing these state estimations. The first is Kalman filtering and the second is Particle filtering. The problem becomes which estimator is more accurate and which estimator requires the least amount of computational complexity in order to be

implemented at the lowest financial cost. The proposed solution to this second problem which is how to do the target detection and tracking has been published[17] and is discussed in Chapters 2, 4, and 5. Chapter 2 discusses how physical phenomena introduct noise into the RADAR system. Chapter 4 discuses the Multiple Hypothesis Tracking algorithm and underlying state estimation systems. Chapter 5 discusses the numerical results and simulations. This dissertation concludes in Chapter 6 where a summary of the dissertation contributions is made. Chapter 6 also discusses future work that could be built off of this dissertation.

CHAPTER 2

RADAR PRINCIPLES

To understand this dissertation, there are a few fundamental concepts that need to be understood. The first concept is that by which a RADAR operates. This starts with an understanding of how a RADAR is constructed physically and how it processes the signals it receives. This fundamental concept is best understood in a perfect simulation. Once that is understood there needs to be a discussion about the noise that will be present in all RADAR systems.

Most modern RADAR systems for small craft today employ a solid state design instead of a magnetron based design. In order to increase the probability of detection due to the lower power levels, pulse compression is employed instead of a standard continuous wave pulse which increases the coherent integration time. The particular method used is Frequency Modulated Continuous Wave (FMCW) modulation. A FMCW RADAR transmits a signal from frequency F_1 to Frequency F_2 over a period of time T. This would present itself on a spectrum histogram similar to what is shown in Fig. 8. The Pulse Repetition Frequency is calculated as

$$PRF = \frac{1}{T} \tag{1}$$

and the bandwidth β can be calculated as

$$\beta = |F_1 - F_2| \tag{2}$$

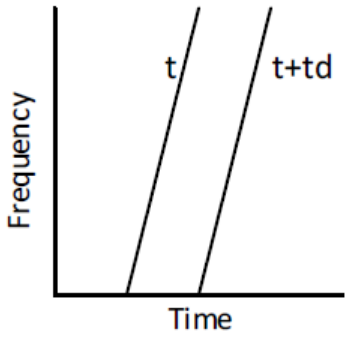

Fig. 8: RADAR Transmitted Signal (t) and Reflection (td) as would be shown on a waterfall spectrogram

A FMCW RADAR determines the range to the target by mixing its transmitted signal (t) with its received signal (td). A target that is down range will reflect a signal that is delayed by a certain amount of time (td). This delay manifests itself as a frequency offset when the transmitted and received pulses are mixed together and corresponds to the range of the target.

The offset frequency is a function of the chirp slope (K) and the distance to the target. This offset frequency resembles a simple sine wave and can be calculated as

$$\Delta f = K\frac{2d}{c} \tag{3}$$

Where K is the chirp slope and is defined as the product of the PRF and Bandwidth, d is the distance to the target and c is the speed of light.

It is important to note that the range resolution is a function of bandwidth, so the bandwidth must be increased to increase range resolution. In order to not violate Nyquist, the resolution bandwidth (RBW) is equivalent to the PRF. This can be shown as

$$RBW = K\frac{2d}{c} \tag{4}$$

$$PRF = PRF * \beta\frac{2d}{c} \tag{5}$$

$$1 = \beta\frac{2d}{c} \tag{6}$$

$$\frac{c}{2d} = \beta \tag{7}$$

$$d = \frac{c}{2 * \beta} \tag{8}$$

where d in Equation-8 becomes the range resolution.

The analytical expression for representing the waveform is

$$x(t) = sin\left(\theta_0 + 2\pi\left(\frac{Kt^2}{2} + F_1 t\right)\right) \tag{9}$$

where θ_0 is the starting phase of the waveform. K is the chirp slope and F_1 is the starting frequency.

Range processing can simply be performed by taking the Discrete Fourier Transform (DFT) of the received signal. RADAR targets are then represented as impulses and the frequency at which the impulse occurs is correlated with distance to the target. It is important to note that some RADAR systems choose to use the Chirp-Z transform. The Chirp-Z transform allows the RADAR to perform the range processing while decimating the output so that it is more appropriate for further processing [18].

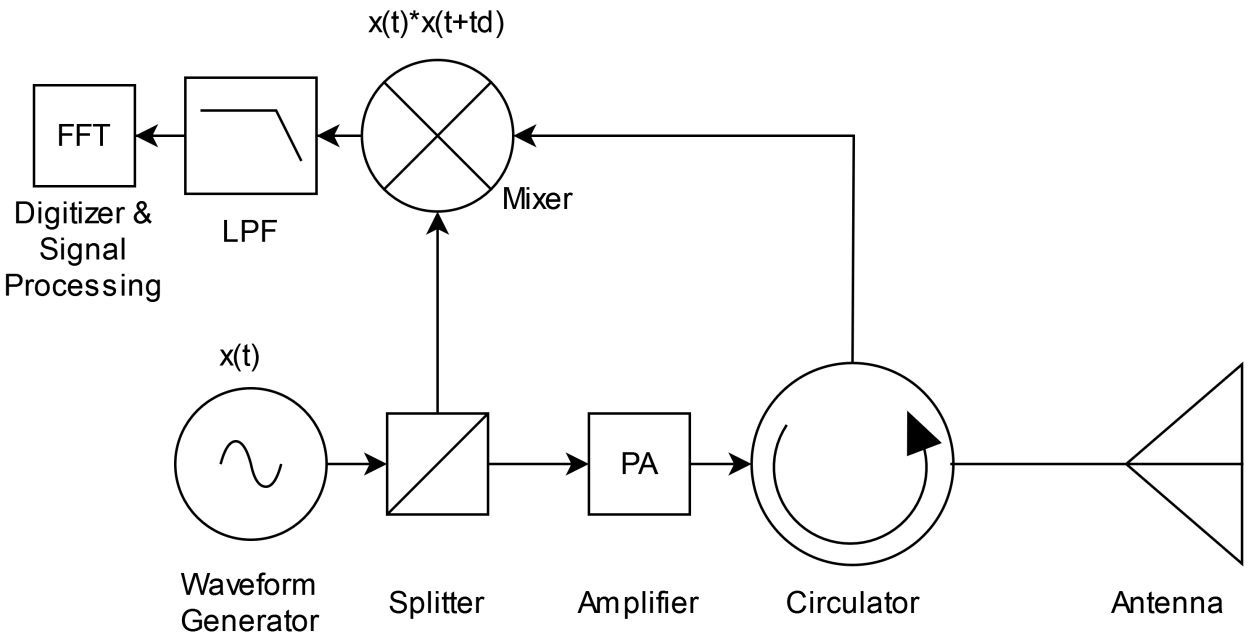

Fig. 9: Block Diagram of a Frequency Modulated Continuous Wave (FMCW) Marine RADAR.

An example block diagram of a marine RADAR is shown in Fig. 9. A signal generator creates the FMCW waveform. The waveform is sent to a splitter. One output of the splitter connects to the power amplifier which amplifies the signal before sending it through the circulator to the antenna. The other side of splitter connects to a mixer which mixes the received signal with the generated signal. The output of the mixer is sent through a low pass filter where it is then digitized and processed using an FFT.

2.1 ARRAY CONSTRUCTION

RADAR ultimately working by sending a signal out into the world and receiving the reflection. In a perfect world, an almost infinite amount of power could be transmitted from

the RADAR guaranteeing that a reflection would be received and processed. Unfortunately, that is not possible so when designing a RADAR system it is important to design it properly. The most cost effective way of increasing the probability of detection is by having a larger RADAR aperture. The aperture of a RADAR ultimately dictates its size and is the amount of collection area that the RADAR has for receiving reflections. Small marine RADAR typically have an aperture which is designed out of a phased array. A phased array is a structure which takes the inputs from multiple antennas and ensures those inputs are added constructively allowing the RADAR to ultimately receive and signal with a higher amplitude.

Phased arrays can be implemented via analog means and digital means. The simplest phased arrays can be constructed out of monopole antennas and are called end fired arrays. These types of phased arrays point in a specific direction and the larger the number of antennas in the array, the more directional it becomes. They are constructed out of monopoles spaced at quarter wavelength in a straight line. Delays are added to the feedlines so that the signals add constructively and destructively to point in one direction. The resulting signals from the antenna elements plus their delays are sent to a combine which adds them together producing one resulting signal. An illustration of an end fire array is shown in Fig-10.

This basic structure can be adapted by changing the delays so that the array pattern points broadside to the physical array. This type of array is simply called a broadside array. In this case, the delays that are introduced between the elements and the combiner are set to zero. A beam is formed which points both towards the front of the array and behind it. An illustration of a broadside array is shown in Fig-11.

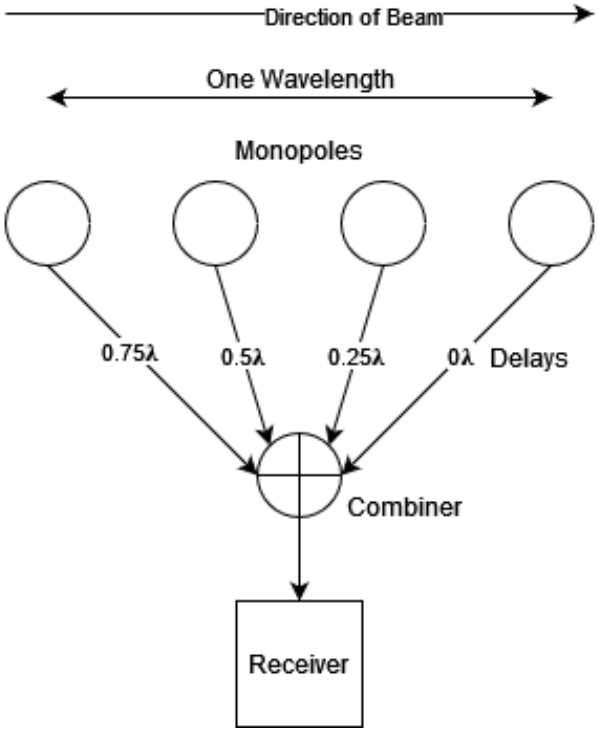

Fig. 10: Simple End Fire Array

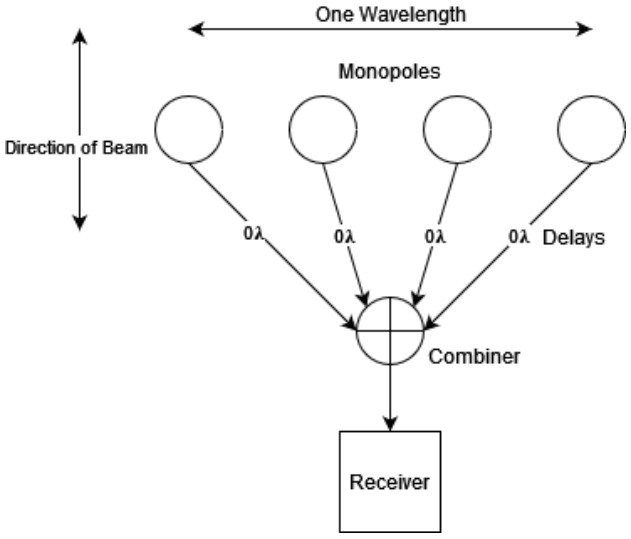

Fig. 11: Simple Broadside Array

The broadside array is physically easier to implement when designing small marine RADAR. The monopoles can also be swapped out for directional antennas such as slotted waveguides which makes the array unidirectional.

The directivity D at a given angle for a uniform linear array of length N with elements spaced d meters apart at a frequency with wavelength λ can be determined using Equation-10[19, 20, 21].

$$D = \sum_{n=1}^{N} e^{\left(\frac{\sin(\theta) n d 2\pi j}{\lambda}\right)} \tag{10}$$

The International Telecommunications Union has restricted X-Band radio navigation to operate between 9200 MHz and 9500 MHz [22]. It appears most operate towards the higher frequency portion of the spectrum around 9.4 GHz [23, 24].

The wavelength at 9.4 GHz is therefore approximately 3.19cm. A typical uniform linear array has elements spaced half a wavelength apart. Performance of the system can be significantly increased if sea-clutter returns can be reduced. These returns are typically vertically polarized. Using elements that are horizontally polarized and increasing the spacing between elements decreases the cross-polarization level and increases isolation [25].

A wavelength spacing of 0.75λ was chosen for this model. Given a diameter of 24" which most commercial small marine RADAR systems follow, 24 patch antenna elements can be used for the array with an aperture size of around 22.6" and a half power array beam width θ_{bw} of approximately 4°. A plot of this implementation is shown in Figure-12. It is important to note that many commercial marine RADAR use slotted waveguide antennas, however they can be more complicated to design and analyze than an array made out of

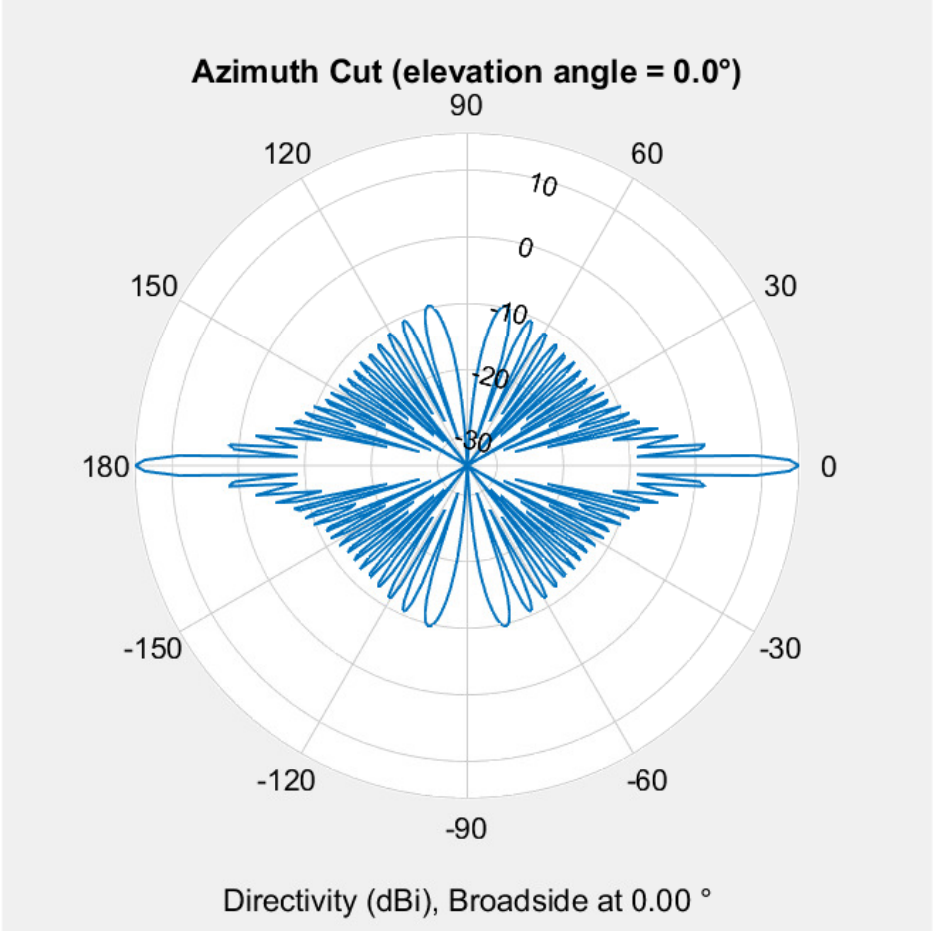

Fig. 12: Antenna Pattern of the Marine RADAR antenna.

patch antennas [26].

Small Marine RADAR systems typically have a vertical beam width of approximately 20° [24]. This becomes important because it, in some cases, can cause blind spots if the RADAR is not mounted properly on the vessel and its view is obstructed.

2.2 RADAR EQUATION

The RADAR equation is used to understand how a RADAR system will radiate a target and what the resulting received signal will be. If we assume in the simplest case that the

RADAR is emitting power from an isotropic antenna which radiates equally in all directions, then the power density is simply the transmitted power divided by a sphere of a certain radius as shown in Eq-11[1].

$$\text{Power density from isotropic antenna} = \frac{P_t}{4\pi R^2} \tag{11}$$

From there, we can extend the formula to add a gain factor G_t if the antenna is designed to radiate in a particular direction as shown in Eq-12[1].

$$\text{Power density from directional antenna} = \frac{P_t G_t}{4\pi R^2} \tag{12}$$

When the transmitted RADAR wave hits the target, it is going to reflect some of that power back. The amount of power reflected back is the targets RADAR cross section which uses the unit σ which has units power per square meter. Putting this together with the power transmitted from a directional antenna results in the basic RADAR equation which determines power received (P_r) given a transmitted power level, transmitted antenna gain (G_t), receive antenna gain (G_r), distance (R), and radar cross section of the target (σ) as shown in Eq-13[1].

$$P_r = \frac{P_t G_t G_r \sigma}{(4\pi)^2 R^4} \tag{13}$$

The RADAR cross section of a target can theoretically be calculated by solving Maxwell's equations. However, in practice it is usually much easier to measure the RADAR cross section [1]. For complex targets, such as boats, the RADAR cross section changes depending upon its orientation with respect to the RADAR[1]. Some examples of RADAR cross

Object	Square meters
Bird	0.01
Small open boat	0.02
Human	1
Small pleasure boat	2
Cabin cruiser	2
Automobile	100

TABLE 2: RADAR Cross Sections at Microwave Frequencies as determined by *Skolnik*[1]

sections at microwave frequencies as determined by *Skolnik* are shown in Table-2.

The RADAR equation can be manipulated to determine the maximum range (R_{max})that a given target can be detected. This is accomplished by setting the received power to the minimum detected power level of the RADAR system and solving for the range which is shown in Eq-14[1].

$$R_{max} = \left(\frac{P_t G_t G_r \sigma}{(4\pi)^2 P_{min}} \right)^{\frac{1}{4}} \tag{14}$$

The probability of detection of the RADAR system (P_d) is partially correlated to the maximum range equation. Environmental factors, such as external noise, also influence the probability of detection.

2.3 SYSTEM NOISE

There are two main sources of system noise that are present in RADAR systems. This

noise causes uncertainties about where exactly detected targets are at any point in time and ultimately must be properly presented to the user. The first main source of noise is from the RADAR itself. This noise is generated from the signal processing and antenna system. The second main source of noise is from the craft's movement on the ocean.

The two components of noise in the RADAR system are noise in range and noise in azimuth. The resolution bandwidth ultimately influences noise in the range and thus it is important to have it as low as possible. Noise in azimuth is a result from the beam width of the antenna. Performance of the system can be significantly increased if sea-clutter returns can be reduced. These returns are typically vertically polarized. Thus using antenna elements that are horizontally polarized and increasing the spacing between elements decreases the cross-polarization level and increases isolation [25]. Most small marine RADAR antennas are made out of slotted waveguide antennas which are described in [26] and have a beam width of only a few degrees [23, 24].

2.4 OCEAN MOVEMENT NOISE

Sea state is the term used to describe the dynamics of the ocean during a certain period of time. Sea states can be determined by measuring the average height of the waves. There are several different techniques for defining the sea state, however this dissertation will be using the Beaufort Scale to specify sea state which is described in Table-3[2].

The sea state by itself is meaningless without knowing the wave period. A sea state of 5 with a period of 5 seconds would result in a very rough environment. However, a sea state of 5 and a period of 30 seconds would not be anywhere near as bad. Ocean waves are generated as a function of the wind and the depth of the water. Shallow seas are unable to

Sea State	Wave Height (ft)	Wind Description
0	0ft	Calm
1	0-1ft	Light Air
2	1-2ft	Light Breeze
3	2-4ft	Gentle Breeze
4	3.5-6ft	Moderate Breeze
5	6-10ft	Fresh Breeze
6	9-13ft	Strong Breeze
7	13-19ft	Near Gale
8	18-25ft	Gale
9	23-32ft	Severe Gale
10	29-41ft	Storm
11	37-52ft	Violent Storm
12	46ft	Hurricane

TABLE 3: Beaufort Sea State Scale as described by the United States National Oceanic and Atmospheric Administration[2]

generate large waves.

It is important to note that the characteristics of the sea state can have a major effect on RADAR performance. In 1984 the National Oceanic and Atmospheric Administration (NOAA) performed high sea state testing on a 29 foot boat [27]. During this test low sea state 3 conditions were experienced and maximum pitch of the vessel reached 17.5°. It is important to note that if the vessel were to be smaller or the sea state larger, that is it possible that the RADAR's probability of detect would decrease due to not being able to see the targets due to pointing either into the water or up to the sky.

This phenomena could be simulated. Ocean Waves cannot be adequately represented as pure sine waves. Ocean Waves are best represented as trochoidal waves (also called Gertsner waves) [28]. These types of waves have wider troughs and narrower peaks than standard sine waves as shown in Fig. 13. The result for a small craft in larger seas is that the effect on pitch (or roll if the craft is perpendicular to the waves) will be greater than that simulated by a pure sine wave.

These waves are best simulated by setting up a grid of coordinates which become the sampling location. From those sampling locations, a sine wave at a specific phase is added to modify the X and Y coordinates as shown in Eq-15 and Eq-16. d is the distance between the grid coordinates in unit less length and P is the wave period in unit less length. The distance of the grid coordinates must not violate the Nyquist Theorem.

$$X[k] = \frac{A}{2} cos(2\pi \frac{kd}{P}) + kd \tag{15}$$

$$Y[k] = \frac{A}{2} sin(2\pi \frac{kd}{P}) \tag{16}$$

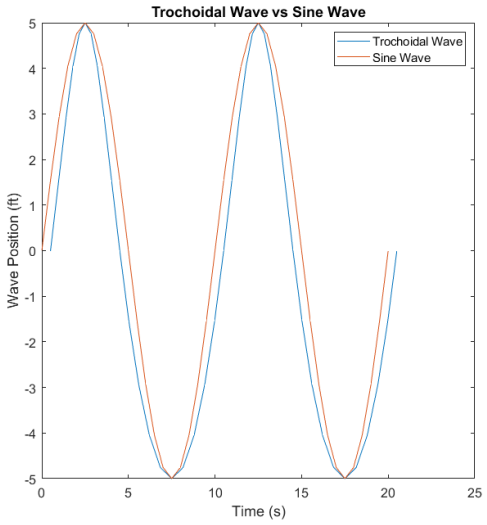

Fig. 13: Comparison of a Sine Wave vs Trochoidal Wave with a Period of 10 seconds and Height of 10 feet

The ocean can also contribute to noise in azimuth of the RADAR system. This is due to rotations in the yaw axis of the RADAR on the vessel. A system of very accurate gyroscopes must be used in order for the system to know exactly where spokes were pointing when the signals were received. There is currently very little published research on this, however there is some research related to High Frequency RADAR on marine platforms [29]. It is the authors intentions to research this further and to do further analysis.

2.5 FUNDAMENTAL TRACKING ISSUES

In order for a RADAR tracker to track properly, it needs to be able to have an accurate location of the object that it is tracking. RADAR tracking can be done using only an angle from the observer to target, which is called bearings-only tracking. RADAR tracking

Fig. 14: The uncertainties about the vessel's heading add to the noise inherent in the RADAR system itself which reduces RADAR tracking performance

can also be performed using a bearing angle and range which is what is most commonly implemented.

RADAR tracking is less complex when the platform doing the sensing is fixed such as a platform fixed on the ground. When the RADAR is fixed, the only noise in the system with respect to determining a targets location is due to the noise of the RADAR itself. When the RADAR platform is on a maneuvering platform such as a vessel or an airframe, then there is also noise present due to uncertainties about the platforms orientation. A representation of these two scenarios is shown in Figure-14.

If the noise that is generated inside of the RADAR tracking system is too great, then the tracking algorithms will fail. Tracking algorithms have to make basic assumptions about their targets and if their tracks exceed those assumptions they become discarded. One of the most basic assumptions for a tracking algorithm is the speed of a target. If the tracking algorithm detects an observation at one location and then at some later time detects another

observation in the vicinity of the first, it will calculate the resulting speed. If the speed is above a certain gating threshold, it will not associate the observations into a track. In the case of a standard RADAR setup, that erroneous speed could have been generated by noise a fraction of a degree in the azimuth to the target.

CHAPTER 3

AUTOMATIC TARGET DETECTION

There has been much research done on marine RADAR and most of this work has been done in various niches such as Ocean Wave analysis [30, 31] various types of target detection [32, 33, 34, 23] and tracking algorithms[35, 36]. One of the topics which has been mostly missing from previous literature is how to implement a system which can take RADAR data from a commercial off the shelf (COTS) small marine RADAR so that the data may be used for future research. One of the goals of this dissertation is to discuss a process by which targets can be extracted from small marine RADAR to be used for further research involving the development and implementation of novel tracking algorithms.

3.1 TARGET REPRESENTATION

In a perfect world, the RADAR would detect other craft and know their position precisely. Unfortunately, the noise described in the previous section prevents this. The best method for presenting this information to the operator (or tracking algorithm) is as an ellipse.

The major axis of the ellipse is the width of the beam at the specified range plus the associate noise. The minor axis is dependent on the number of occupied range bins and the resolution of the RADAR determined by the bandwidth plus the noise. The equation for an ellipse is given as

$$\frac{x^2}{a} + \frac{y^2}{b} = 1 \tag{17}$$

The variables **h** represent the translation in the X coordinate system and **k** represent the translation in the Y coordinate system as shown.

$$\frac{(x-h)^2}{a} + \frac{(y-k)^2}{b} = 1 \tag{18}$$

The next step is to be able to rotate the ellipse. To do so, a change of variable variables can be implemented and **u** and **v** will be used for this

$$\begin{aligned} u &= x - h \\ v &= y - k \end{aligned} \tag{19}$$

Then a counter clockwise rotation matrix is used to rotate the ellipse as follows

$$\begin{aligned} \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} &= \begin{bmatrix} u' \\ v' \end{bmatrix} \\ &= \begin{bmatrix} u\cos(\theta) - v\sin(\theta) \\ u\sin(\theta) + v\cos(\theta) \end{bmatrix} \end{aligned} \tag{20}$$

Then back substitute the change of variables into the original equation to arrive at the derivation for a ellipse that has been translated and rotated

$$\begin{aligned} &\frac{((x-h)\cos(\theta) - (y-k)\sin(\theta))^2}{a} \\ &+ \frac{((x-h)\sin(\theta) + (y-k)\cos(\theta))^2}{b} = 1 \end{aligned} \tag{21}$$

The equation can now be modified with parameters from the RADAR so that it can be used for drawing ellipses which represent the target plus the noise. a which is the major axis of of the ellipse can be substituted by $d\tan(\theta_{bw})$ which represents the beam width at the specified distance from the RADAR. b which is the minor axis is equivalent to the range resolution of the radar and would therefore be determined using Equation-8. Putting this all together, the resulting equation is

$$\begin{aligned} &\frac{((x - d\cos(\theta))\cos(\theta) - (y - d\sin(\theta))\sin(\theta))^2}{d\tan(\theta_{bw})} \\ &+\frac{((x - d\cos(\theta))\sin(\theta) + (y - d\sin(\theta))\cos(\theta))^2}{\frac{c}{2*\beta}} \\ &= 1 \end{aligned} \tag{22}$$

If the two targets that were analyzed earlier are plotted using this equation, the result is shown in Fig. 15. They are both 45° from the RADAR with a range of 1 nautical mile and 3.5 nautical miles. a represents the targets location in range including the noise. b represents the targets location in azimuth plus the noise. d_1 and d_2 represent the respective distances to target. θ_1 and θ_2 represent the respective azimuths. This could easily be prototyped into a display that an operator could look at for situational awareness.

As far as implementing collision avoidance measures, these ellipses could be put through a standard tracking system which could generate course and speed for each target. The system could then warn the operator if a collision is possible and warn the operator with an alarm.

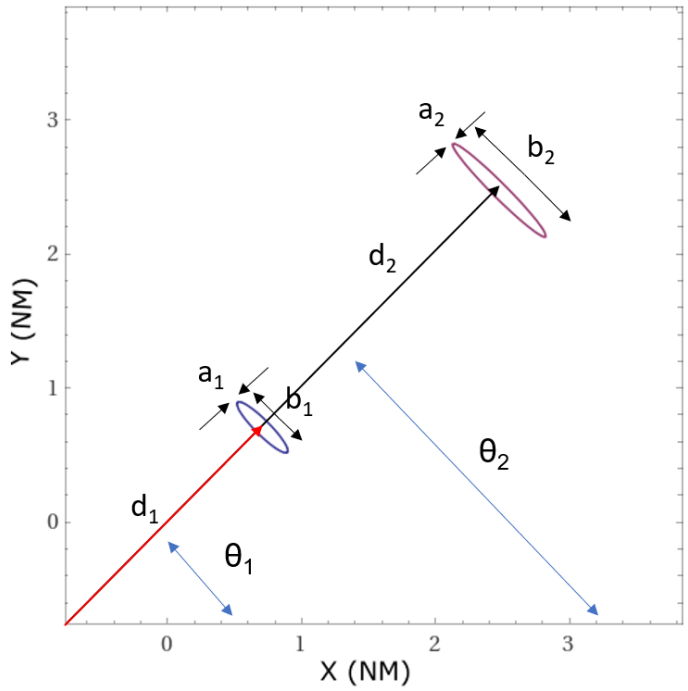

Fig. 15: The extracted targets from the Target Detection routine plotted using ellipses given by Equation-22.

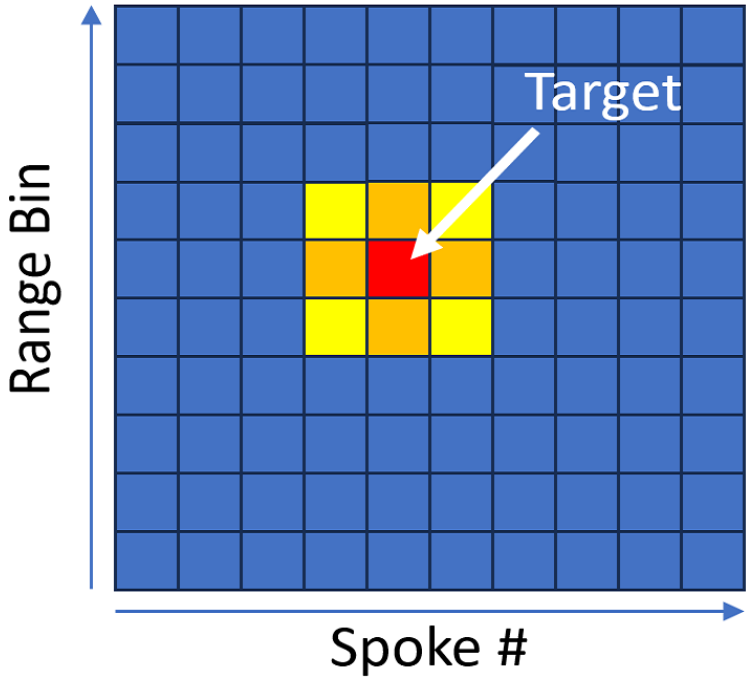

Fig. 16: RADAR spoke data is returned from the RADAR where a spoke corresponds to a fraction of a degree in azimuth and range bins are used to specify distance. The value of the cell indicates returned signal strength.

3.2 TARGET EXTRACTION

A RADAR for small maritime vessels is typically only a few thousand dollars and offers decent performance. Modern ones are typically network connected and are capable of communicating over a standard Ethernet interface using standard Internet Protocol (IP) packets. Most RADAR vendors support software development kits which can be used to easily connect to the RADAR and access all of its capabilities[37]. Some of these RADAR systems are supported by open source software packages such as the RADAR Pi plugin for OpenCPN which is capable of communicating with some RADAR to retrieve its data[38]. The data is returned in the form of spoke data which is a vector at a given azimuth from the vessel. The indices in the vector represent a distance from the vessel and the value for each index represents the returned signal strength from the RADAR.

A RADAR scan is a full 360° of spoke data. A basic illustration of spoke data is shown in Fig-16. Modern small RADAR systems return in some cases several thousand spokes per revolution resulting in an angular resolution which can be on the order of a fraction of a degree. As the signal processing system quantizes the returns into the spoke data (typically as a result of a Fourier transform), power can sometimes be split between cells reducing the overall signal-to-noise (SNR) ratio of a target [39, 18].

It is important to note that as a target gets further away from the radar, the target appears larger on a polar display due to distortions from the polar projection. This is demonstrated in Fig. 22 where two targets were generated with the same size, however, the first target was 1 nautical mile away while the second target was 3.5 nautical miles away. In Fig. 23, which is displayed in Cartesian, both targets appear to be the same size.

Therefore, when processing the data for automatic target detection it is important to due so in Cartesian format.

Once each RADAR scan is received, analysis needs to be done on the data to determine targets. This can be performed using standard digital image processing techniques such as blob detection. In order to do this detection, a toolkit such as OpenCV can be used. OpenCV supports a function called *SimpleBlobDetector* which uses the $findContours$ function which is an implementation of the algorithm described by Satoshi Suzuki et. al [40]. Ultimately, this algorithm works by detecting the change in gradient of an image to detect borders. It then follows the border to see if the border forms an enclosed region. Once a blob is detected, OpenCV can filter the results to only return blobs that meet area, threshold, circularity, inertia or convexity requirements. To implement a target extraction method, filtering based on threshold would allow OpenCV to only return blobs with a certain signal strength. The $findContours$ function can only work with grayscale images. So, the first step in detecting the targets is to convert the image into a grayscale image. Once the image has been converted to grayscale, the *SimpleBlobDetector* function operates by detecting darker blobs.

The problem becomes how to set the thresholds for blob detection. If the threshold is too high, then targets will not be detected in the RADAR data. However, if the threshold is too low, then there will be too many targets detected which would increase the false alarm rate to unacceptable levels. A 1dB change in threshold can result in three orders of magnitude change in the probability of false alarm, thus automatic target detection systems can typically only handle less than a 1dB increase in noise level[1]. Systems that automatically

adjust their thresholds are called Constant False Alarm Rate (CFAR) detectors[1].

CFAR detectors are typically implemented using a cell averaging technique. The value of each cell in the RADAR display is compared against the surrounding cells. There is typically a perimeter of guard cells immediately around the pixel under test, meaning that the distance between the reference cell and the cell under test is typically at least 2. This accommodates situations where the power may be split across two cells.

The OpenCV *SimpleBlobDetector* function requires two inputs for thresholding, the minimum and maximum values. The function starts at the minimum threshold value and steps its way up to the maximum threshold value. It then prunes detected blobs that were inside of other blobs that were detected at a higher threshold value. This is very similar to a gradient descent algorithm. The thresholding method proposed in this chapter is to set the minimum threshold value to zero and set the maximum threshold value to two standard deviations below the mean of the cell values in the RADAR return.

To simulate the system, a simulated target on a RADAR raster display was generated. Spoke data was generated for a complete revolution of the RADAR up to 360° with a resolution of 1°. To enhance the legibility of this chapter, there were only 64 range bins simulated which is considerably lower than would be found in an actual system. However, the fundamental techniques would remain the same. A singular simulated RADAR return used in this simulation is shown in Fig-17. The reflected power of the simulated target is approximately 20 dB above the noise floor.

The raw RADAR image is then converted to an 8 bit grayscale image as shown in Fig-18. This is accomplished using OpenCV's built in image loading library. The next step is to

invert the image colors. The *SimpleBlobDetector* function attempts to find blobs that are darker. OpenCV provides a function called *bitwise_not* which can do this conversion and the result is shown in Fig-19.

The final step is to call the *SimpleBlobDetector* function which detects the blobs and returns the pixel coordinates and other pieces of information about the blobs. OpenCV provides an easy routine called *drawKeypoints* which allows for an easy display of the resulting blobs as shown in Fig-20. An example of multiple targets being detected is shown in Fig-21.

To determine how effective this technique is for automatic target extraction, a divide and conquer algorithm was implemented which reduced the SNR of the simulated target. A simulation was run and it was determined this technique works effectively when the target signal is approximately 12 dB or more.

While this technique will work for basic cases, there are many improvements that can be done. Work done by D. Yulian *et. al.* shows that accuracy of ship detection within a range of 2 nautical miles can approach 97% [32].

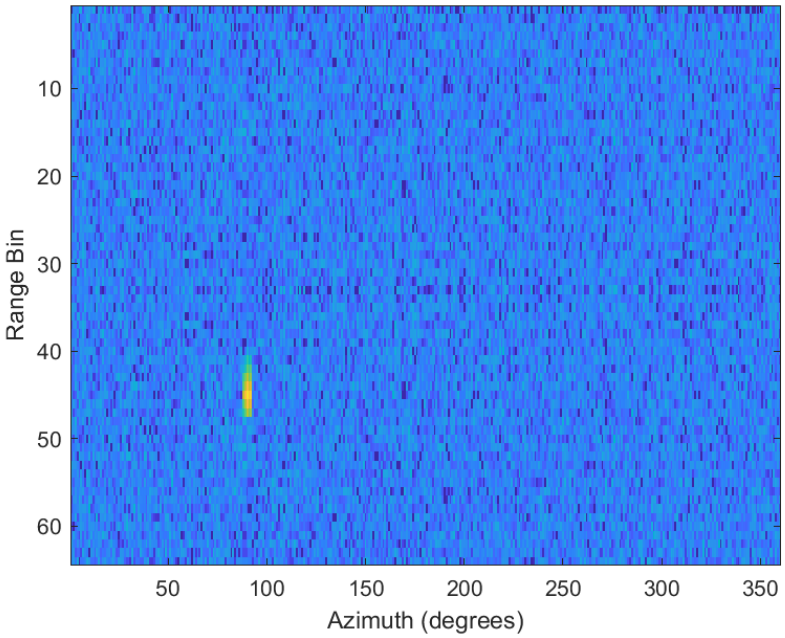

Fig. 17: Simulated RADAR target with full color display.

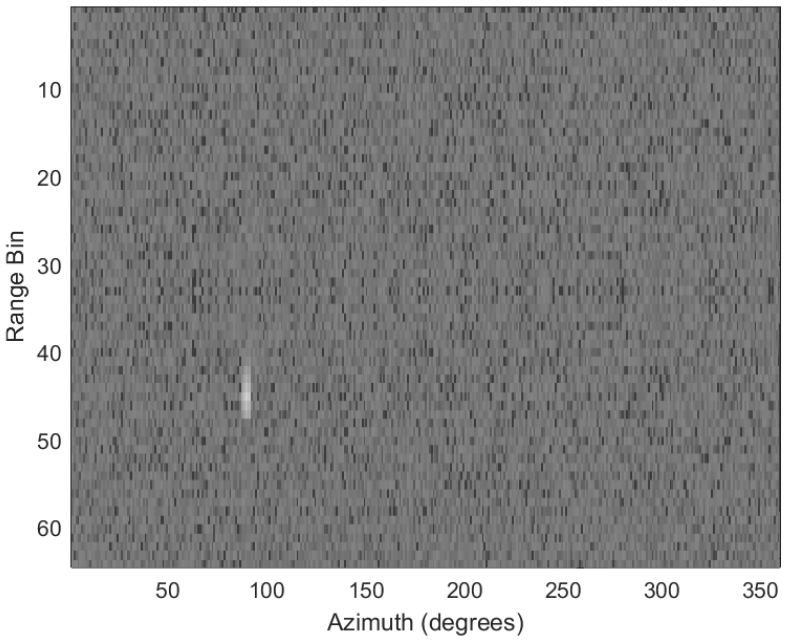

Fig. 18: The simulated RADAR target after is has been loaded using OpenCV and converted to greyscale.

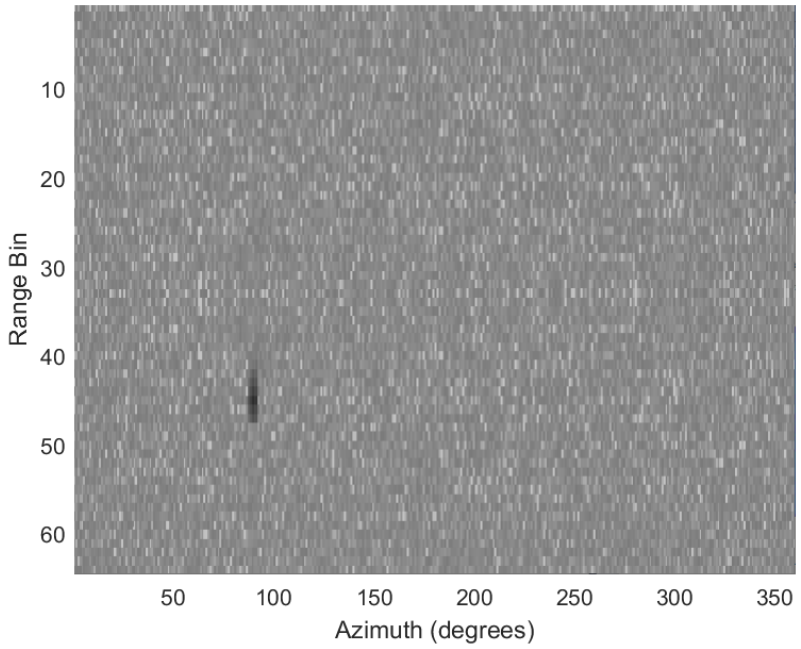

Fig. 19: The simulated RADAR target after its colors have inverted immediately before calling the *SimpleBlobDetector* function.

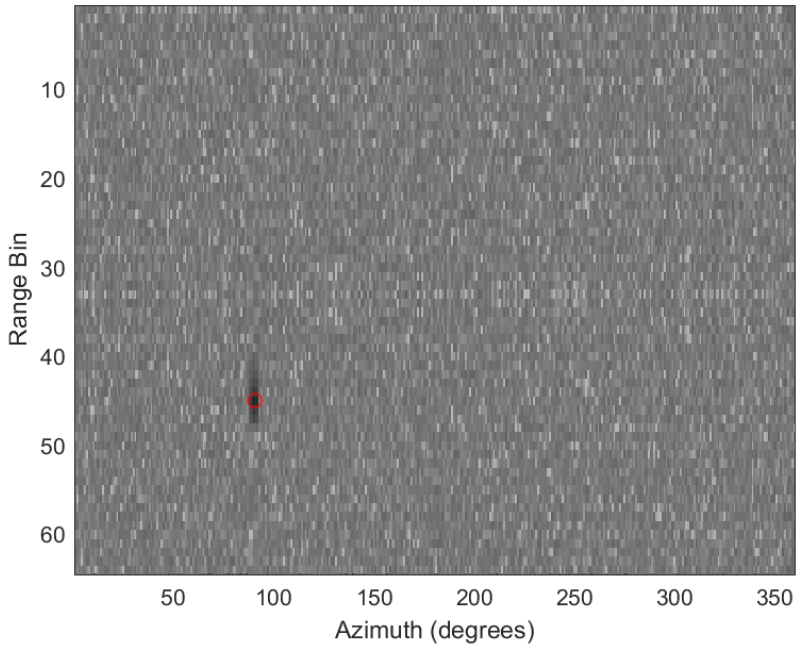

Fig. 20: Example of a single target being detected and being annotated with OpenCV's *drawKeypoints* function.

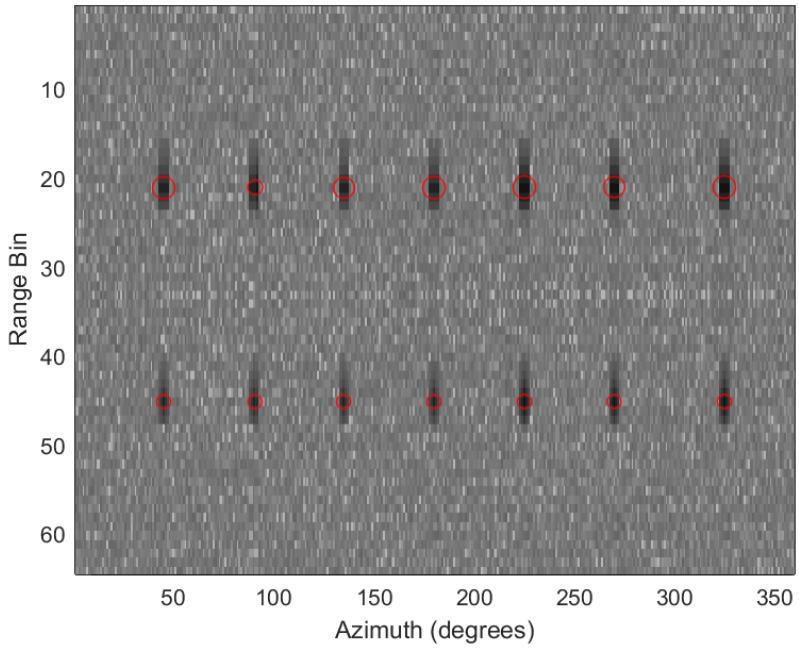

Fig. 21: Example of multiples targets being detected and being annotated with OpenCV's *drawKeypoints* function.

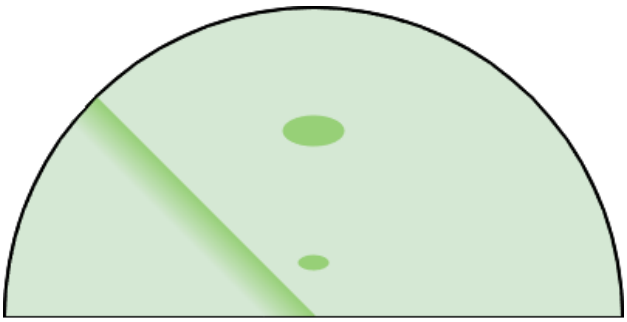

Fig. 22: This figure shows two targets of the same size, but at different distances from the RADAR. When plotted in polar format, they appear as different sizes.

Fig. 23: When displayed in Cartesian format, RADAR targets look the same size regardless how far they are from the RADAR.

CHAPTER 4

MULTIPLE HYPOTHESIS TRACKING

The Multiple Hypothesis Tracking Algorithm was originally described by Donald B. Reid in his paper "An Algorithm for Tracking Multiple Targets". Multiple Hypothesis Tracking (MHT) is one of the best algorithms for target tracking due to the fact that it is one of the few algorithms that works for multiple targets, missing measurements, false alarms, track initiation, sensor data sets, multiple-scan correlation, clustering, and is recursive [41].

When it comes to tracking algorithms, one of the most advanced and modern is the MHT algorithm. This tracking algorithm is capable of tracking multiple maneuvering targets. To do so, it relies on a state estimation and prediction algorithm to estimate where a target is based off of observations and where that target may be going. Most implementations of the MHT algorithm rely on Kalman Filtering which is relatively computationally efficient. A goal of this dissertation is to explore whether Particle Filtering performs as well as Kalman Filtering and if the additional computational complexity is worth it.

The purpose for using a Multiple Hypothesis Tracker over simpler methods such as nearest neighbor correlation algorithms is that nearest neighbor approaches experience frequent miscorrelations which result in a serious degradation of track quality [42]. A Multiple Hypothesis Tracker performs significantly better when dealing with dense target environments and false alarms. This makes it an ideal algorithm for marine applications since it will work in the dense cluttered target environments of a harbor as well as the open ocean.

When a vessel is operating on the water, there are several motions that it experiences. One of those motions is Directional Stability which is partially related to the yaw rate of the vessel. As hydrodynamic effects act on the vessel, the heading of the vessel will change. If these changes in heading are not recorded at a fast enough rate, then the vessel orientation input into the Multiple Hypothesis Tracking algorithm will result in poor quality target locations being used. This will ultimately lead to the tracker no longer being able to track targets.

A RADAR mounted on a vessel detects targets at a specified bearing (η) and range (r). These measurements need to be converted to measurements that the tracking algorithm can use. Inside of η there are actually two measurements added together. One measurement is the measured bearing to target η_m which has known variance and the other measurement is the unknown offset η_u which has unknown variance.

η_u is created by the fact that the vessel may not know exactly where it is pointing at a given instant. Most recreational craft have GPS systems that are only capable of updating once per second (i.e. a 1Hz GPS). If they were to experience a large wave slam, this would mean that the RADAR and tracking algorithm could be processing targets that are very far away from where they actually are due to the unmeasured movement of the vessel. This variance is Gaussian and can only be reduced by using a sensor with a higher and more accurate update rate for vessel orientation.

4.1 MULTIPLE HYPOTHESIS TRACKING IMPLEMENTATION

The tracker is implemented as a Track-Oriented Multiple Hypothesis Tracker using either a Kalman Filter or Particle Filter for target location estimation. A block diagram of this

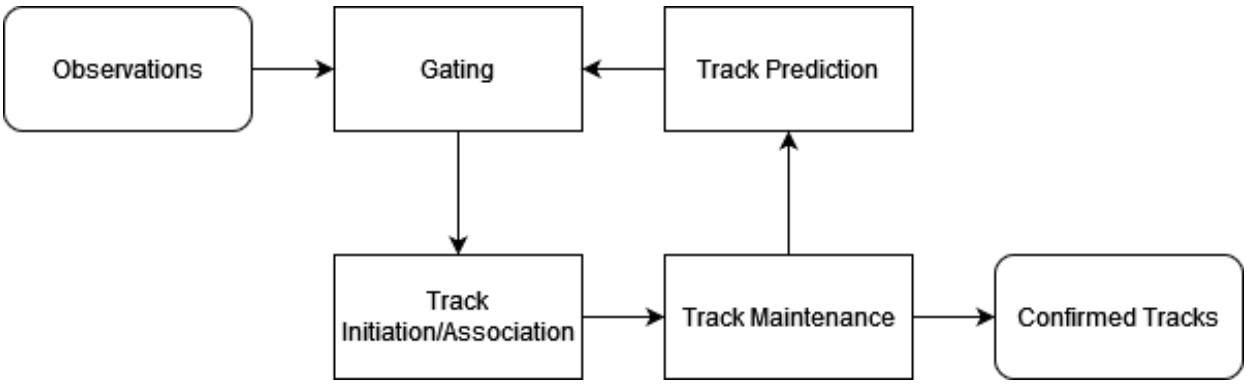

Fig. 24: Block Diagram of the Tracking Algorithm

system is shown in Fig. 24 where the observations are first given to the gating process. The gating process informs the track initiation/association process. Then the track maintenance process is performed where surviving tracks are given to the track prediction stage and the confirmed tracks are output to the user.

When it comes to implementing Multiple Hypothesis Trackers, there are two main approaches. The first approach is the system that Reid described in his paper which is termed Hypothesis-Oriented Multiple Hypothesis Tracking (HO-MHT) [43]. This method carries hypothesis over from scan to scan until ultimately the hypothesis are pruned due to not being associated with a track.

The second approach is termed Track-Oriented Multiple Hypothesis Tracking (TO-MHT). This method converts the tracks from a previous scan into hypothesis, analyzes them given the information from the latest scan and then converts the data back into tracks which is much more efficient [43].

The first step in the process is gating. In this step, observations whose distances fall below a certain threshold compared to a track's predicted location are assigned to that

track[44]. Measurements above the threshold are recorded as possible new tracks. The distance is calculated as the Mahalanobis distance $D_M(\bar{x})$.

$$\bar{x} = \begin{bmatrix} y_{k_x} \\ \\ y_{k_y} \end{bmatrix} \tag{23}$$

$$\bar{u} = \begin{bmatrix} \hat{x} \\ \\ \hat{y} \end{bmatrix} \tag{24}$$

$$S = \begin{bmatrix} P_k^{-11} & P_k^{-12} \\ \\ P_k^{-21} & P_k^{-22} \end{bmatrix} \tag{25}$$

$$D_M(\bar{x}) = \sqrt{(\bar{x} - \bar{u})^T S^{-1} (\bar{x} - \bar{u})} \tag{26}$$

It is possible for an observation to be assigned to more than one track. In this case, a cluster is formed. The cluster contains all intersecting tracks. This means that if tracks *T1* and *T2* share an observation and *T2* and *T3* share an observation then a cluster is formed containing *T1*, *T2* and *T3*. These clusters form an assignment problem which can be solved using an algorithm such as the Munkres Algorithm [42].

The track score L_i at time k can be computed recursively as [43]

$$L(k) = L(k-1) + \Delta L(k) \tag{27}$$

where the initial track score $L(1)$ is calculated as the natural logarithm of the probability of a new track over the probability of a false alarm. For tracks that are in progress, updates are calculated as [43]

$$\Delta L(k) = \begin{cases} ln[1 - P_D] & \text{no detection} \\ ln\left[\frac{P_D}{(2\pi)^{M/2}\beta_{FT}\sqrt{|S|}}\right] - \frac{D_M(x)^2}{2} & \text{detection} \end{cases} \tag{28}$$

where P_D is the probability of detection and β_{FT} is the false target density calculated as the probability of false alarm over the measurement volume. M is the number of dimensions which for this simulation is set to two since the tracker is only operating on two dimensions.

4.2 KALMAN FILTERING

To predict where a measurement for a track should be, a Kalman filter can be used for state estimation and prediction. A Kalman Filter is the ideal estimator when dealing with linear systems that have Gaussian noise. The Kalman filter takes in observations and using a recursive algorithm is able to make an assessment of the noise in the system. The Kalman filter can then adjust how much it weights the feedback from the observations vs the feedback from the state estimation equations to provide a filtered output. The notation the authors used for this section was taken from [45].

The state estimation and estimated co-variance matrix is calculated as

$$x_k^- = F_{k-1}x_{k-1}^+ + w_{k-1} \tag{29}$$

$$P_k^- = F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \tag{30}$$

Where in this application both w_k and Q_k are set to zero.

The update equations are calculated as

$$K_k = P_k^- H_k^T (H_k P_{k-1}^- H_k^T + R_k)^{-1} \tag{31}$$

$$x_k^+ = x_k^- + K_k(y_k - H_k x_k^-) \tag{32}$$

$$P_k^+ = (I - K_k H_k) P_k^- \tag{33}$$

The state equations are

$$x_k = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{\dot{x}} \\ \hat{\dot{y}} \end{bmatrix} \tag{34}$$

where $\hat{x}$ and $\hat{y}$ are the coordinates in Cartesian format and $\hat{\dot{x}}$ and $\hat{\dot{y}}$ are the respective velocities. The state transition matrix is

$$F = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{35}$$

The observation model H is only capable of measuring location and is therefore

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{36}$$

The measurement noise in polar form would be given as

$$R_k = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_{\eta_m}^2 \end{bmatrix} \tag{37}$$

Where $\sigma_{\eta_m}^2$ is the variance due to the measurement of azimuth in the RADAR and σ_R^2 is the uncertainty in the range resolution.

The issue then becomes how to convert the co-variance matrix in polar form to a co-variance matrix in Cartesian form. The naive expansion is [42]

$$R_c = \begin{bmatrix} \sigma_{x_0}^2 & \sigma_{x_0 y_0}^2 \\ \sigma_{x_0 y_0}^2 & \sigma_{y_0}^2 \end{bmatrix} \tag{38}$$

where

$$\sigma_x^2 = \sigma_{r_0}^2 \cos^2 \eta_m + r^2 \sin(\eta)^2 \sigma_{\eta_m}^2 \tag{39}$$

$$\sigma_y^2 = \sigma_{r_0}^2 \sin^2 \eta_m + r^2 \cos(\eta)^2 \sigma_{\eta_m}^2 \tag{40}$$

$$\sigma_{xy}^2 = \frac{1}{2} sin2\eta_m \left[\sigma_r^2 - r^2 \sigma_{\eta_m}^2\right] \tag{41}$$

However, this leads to bias. Research done by Y. Bar-Shalom *et al.* has shown that these measurements can be converted without introducing bias [46]. When this is done, the co-variance matrix becomes

$$R_k^{11} = -\lambda_{\eta_m}^2 r^2 cos^2 \eta_m + \frac{1}{2}(r^2 + \sigma_r^2)(1 + \lambda_{\eta_m}^{'} cos2\eta_m) \tag{42}$$

$$R_k^{22} = -\lambda_{\eta_m}^2 r^2 \sin^2 \eta_m + \frac{1}{2}(r^2 + \sigma_r^2)(1 - \lambda_\eta^{'} cos2\eta_m) \tag{43}$$

$$R_k^{12} = -\lambda_{\eta_m}^2 r^2 cos\eta_m \sin \eta_m + \frac{1}{2}(r^2 + \sigma_r^2)\lambda_{\eta_m}^{'} \sin 2\eta_m \tag{44}$$

$$R_k^{21} = R_k^{12} \tag{45}$$

where

$$\lambda_{\eta_m} = E\left[\cos \upsilon_{\eta_m}\right] = e^{-\sigma_{\eta_m}^2/2} \tag{46}$$

$$\lambda_{\eta_m}^{'} = E\left[\cos 2\upsilon_{\eta_m}\right] = e^{-2\sigma_{\eta_m}^2} \tag{47}$$

when the bearing measurement error is Gaussian.

4.3 PARTICLE FILTERING

There are several different names that are synonymous with Particle Filtering. Some of these names are sequential importance sampling, bootstrap filtering, the condensation algorithm, interacting particle approximations, Monte Carlo Filtering, sequential Monte Carlo (SMC) filtering, and survival of the fittest [45]. The method of particle filtering implemented in this dissertation is sequential importance resampling. Particle Filtering can be used to replace the Kalman Filter in the Track Prediction process of the Multiple Hypothesis Tracker.

Particle Filtering works by first designing a system equation x_k where v_k is an independent white noise process with a known probability density functions.

$$x_{k+1} = f_k(x_k) + v_k \tag{48}$$

The first step is to generate N initial particles based off of the PDF of the initial state $p(x_0)$. The particles are denoted $x^+_{0,i}(i = 1, .., N)$ where the complexity of the system is dictated by N where the higher the value the more accurate the estimate.

The particles are originally generated based off of an estimate of the PDF of the noise process. The estimate doesn't have to be exact because perturbations are added to reduce the degeneracy problem where only a few particles end up representing the PDF.

Once the particles are generated, they are then passed through $f_x()$. The particles are then compared to the measured value y_k to determine their likelihood. The calculation $\hat{N}_{eff}$ is computed to determine the number of effective particles [47].

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N}(w_k^i)^2} \tag{49}$$

A starting point for basic particle filters is to resample the particles if $\hat{N}_{eff}$ is less than $N/2$. In that case, a basic technique called multinomial resampling can be used to perform the resampling of the particles.

The first step in multinomial resampling is to use an importance density function to assign probabilistic weights w_k^i. From there, the Cumulative Sum of Weights (CSW) vector c is generated. For each N, a uniform random value between 0 and 1 is selected. These are looked up in the vector c to select the corresponding particles out of x_k. Ultimately

what this does is selects particles that have a higher importance than those that do not. The resulting resampled vector then statistically only includes particles that have a higher weight and is used from that point on.

At any time, the expected value of particle filter can be taken by calculating the expected value of x_k. This is done by taking the summation of the particles x_k multiplied by their weights w_k^i to determine the state estimate.

When an observation is made, the first thing that needs to be done is to generate a set of particles that represent possibilities for the actual location of the observation. To do so, Eq-42 and Eq-43 can be used to estimate the standard deviations for the particles. For this simulation a simple Gaussian random number can be generated with the mean equivalent to the observation's location.

For each estimate, the particles are passed through $f_k()$ where the state transition matrix is the same as in Eq-35 and a noise vector v_k is added. The particles are then measured against new correlated observations to determine an error. In this implementation, the importance density function is the normal probability distribution function of the observation against each particle. The result is then normalized to form the weights.

4.3.1 EXAMPLE PARTICLE FILTERING

To develop an understanding of how to implement a particle filter, we will work through an extremely basic example. Suppose that there is a target which is traveling at a 45° angle from the origin at 50 kilometers per hour. This equates to approximately 14 meters per second. A plot showing the actual position of the target is shown in Fig-25.

Now, each of those observations is observed. Let's say that these observations have noise

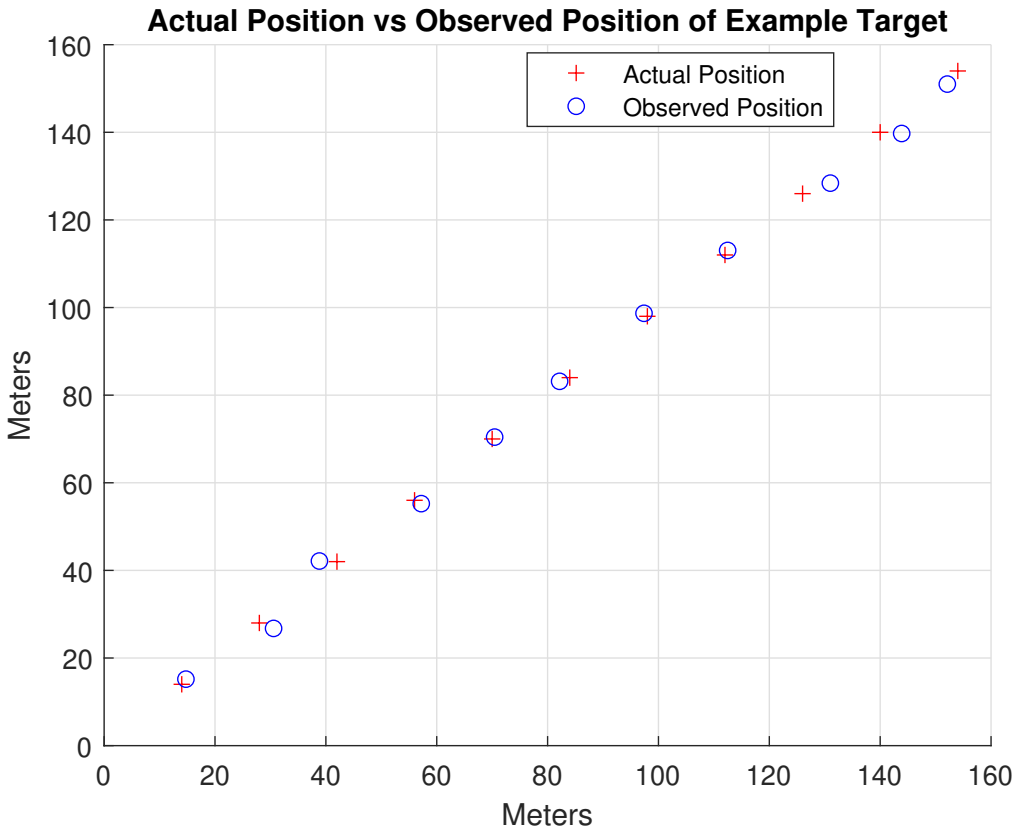

Fig. 25: Results from the particle filter attempting to estimate the position of a moving target.

in a Gaussian distribution of approximately 5 km/h which equates to around 1.7 meters per second. A plot showing the actual position of the target and the measured observations of the target is shown in Fig-25.

To build a particle filter estimator, we need to come up with the state transfer equations. For this basic example, we will just state that the next state is the current state plus the estimated velocity plus a position error. We can define this easily as

$$x^+ = x + v_x + n_x \tag{50}$$

$$y^+ = y + v_y + n_x \tag{51}$$

Inherent in the velocities v_x and v_y is some noise. So we will try to estimate those velocities combined with the noise. To do so, we should generate some particles. If we imagine that this target can move at a speed between 0 km/h and 100 km/h , then we could generate 100 particles between those speeds as

$$\text{velocity\_particles} = [0, 0.2278, 0.5556, ..., 27.78]_{1\times100} \tag{52}$$

We now have a 100 element vector ranging from 0 to 27.78 m/s. Since we are initializing the particle filter, we need to set the particle weights. At this point, we do not have any ability to state which particles represent the system the most, so we set them all to be the same probability which is 1/100.

$$\text{velocity\_particle\_weights} = [0.01, ..., 0.01]_{1\times100} \tag{53}$$

We can then make a decision to generate particles to represent the position error. We can design the system so that we have 100 particles uniformly distributed between -5 meters and 4.9 meters. This would be written as

$$\text{position\_particles} = [-5, -4.9, ..., 4.9]_{1\times100} \tag{54}$$

$$\text{position\_particle\_weights} = [0.01, ..., 0.01]_{1\times100} \tag{55}$$

The next step is to choose our importance density function. For this example, we will choose to evaluate the particles against a normal distribution compared to the measurement values. We will choose the standard deviation to be 1.4 m/s.

When we receive the second observation, we can subtract its position from the original observation to determine the measured velocity.

$$x_{m_v} = x_m[2] - x_m[1] \tag{56}$$

$$y_{m_v} = y_m[2] - y_m[1] \tag{57}$$

From there, we can start to update the weights. We can do so by evaluating the vector containing the weights against the equation for calculating the Guassian probability distribution.

$$\text{x\_velocity\_particle\_importance} = \frac{1}{1.4\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\text{x\_velocity\_particles} - x_{m_v}}{1.4}\right)^2} \tag{58}$$

$$\text{y\_velocity\_particle\_importance} = \frac{1}{1.4\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\text{y\_velocity\_particles} - x_{m_v}}{1.4}\right)^2} \tag{59}$$

We can then multiply the velocity particle importance density by the values of the velocity particles.

$$\text{x\_velocity\_particles} = \text{x\_velocity\_particles} \times \text{x\_velocity\_particle\_importance} \tag{60}$$

$$\text{y\_velocity\_particles} = \text{y\_velocity\_particles} \times \text{y\_velocity\_particle\_importance} \tag{61}$$

Now, we can normalize the weights as follows

$$\text{x\_velocity\_weights} = \frac{\text{x\_velocity\_weights}}{\sum_{a=1}^{100} \text{x\_velocity\_weights}[a]} \tag{62}$$

$$\text{y\_velocity\_weights} = \frac{\text{y\_velocity\_weights}}{\sum_{a=1}^{100} \text{y\_velocity\_weights}[a]} \tag{63}$$

Once the weights are normalized, we can then estimate the velocities by multiplying the values of the particles by their weights.

$$\text{x\_estimated\_velocity} = \sum_{a=1}^{100} \text{x\_velocity\_particles}[a] \times \text{x\_velocity\_weights}[a] \tag{64}$$

$$\text{y\_estimated\_velocity} = \sum_{a=1}^{100} \text{y\_velocity\_particles}[a] \times \text{y\_velocity\_weights}[a] \tag{65}$$

Once we determine the latest estimated velocity, we can estimate the position noise and estimate a new position following the same process. We first start with the state transfer equation

$$\text{x\_estimate} = \text{x\_estimate} + \text{x\_estimated\_velocity} + \sum_{a=1}^{100} \text{x\_position\_particles}[a] \times \text{x\_position\_weights}[a] \tag{66}$$

$$\text{y\_estimate} = \text{y\_estimate} + \text{y\_estimated\_velocity} + \sum_{a=1}^{100} \text{y\_position\_particles}[a] \times \text{y\_position\_weights}[a] \tag{67}$$

From there, we then proceed with the standard particle filtering process to come up with a new estimate as

$$\text{x\_position\_particle\_importance} = \frac{1}{1.4\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\text{x\_position\_particles}-\text{x\_estimate}}{1.4}\right)^2} \tag{68}$$

$$\text{y\_position\_particle\_importance} = \frac{1}{1.4\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\text{y\_position\_particles}-\text{y\_estimate}}{1.4}\right)^2} \tag{69}$$

$$\text{x\_position\_weights} = \text{x\_position\_weights} \times \text{x\_position\_particle\_importance} \tag{70}$$

$$\text{y\_position\_weights} = \text{y\_position\_weights} \times \text{y\_position\_particle\_importance} \tag{71}$$

$$\text{x\_position\_weights} = \frac{\text{x\_position\_weights}}{\sum_{a=1}^{100} \text{x\_position\_weights}[a]} \tag{72}$$

$$\text{y\_position\_weights} = \frac{\text{y\_position\_weights}}{\sum_{a=1}^{100} \text{y\_position\_weights}[a]} \tag{73}$$

We can now proceed with trying to estimate the system. For this example, the measurements for the first 10 observations are shown in Table-4. As the particle filter iterates, the filter weights particles that represent the system with higher probabilities. This is illustrated in Figures 26,27,28, and Fig-29 where you can see the distribution of the particles with higher weights decreases at observation 10 compared to the 2nd observation.

Observation	X	Y
1	14.7527	12.1102
2	30.5674	32.2489
3	38.8376	43.0156
4	57.2070	55.9117
5	70.4463	71.0006
6	82.1692	83.7130
7	97.3930	97.8262
8	112.4797	114.0856
9	131.0098	127.9726
10	143.8772	141.9841

TABLE 4: Observations for particle filtering example

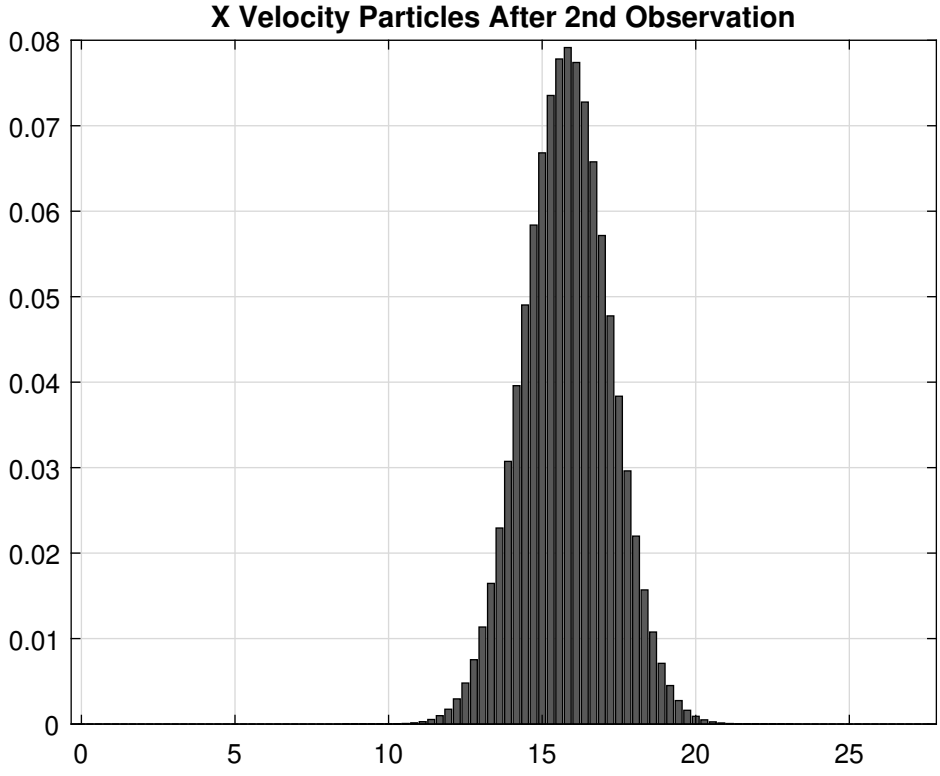

Fig. 26: Bar chart showing the X particle values and weights after the 2nd iteration.

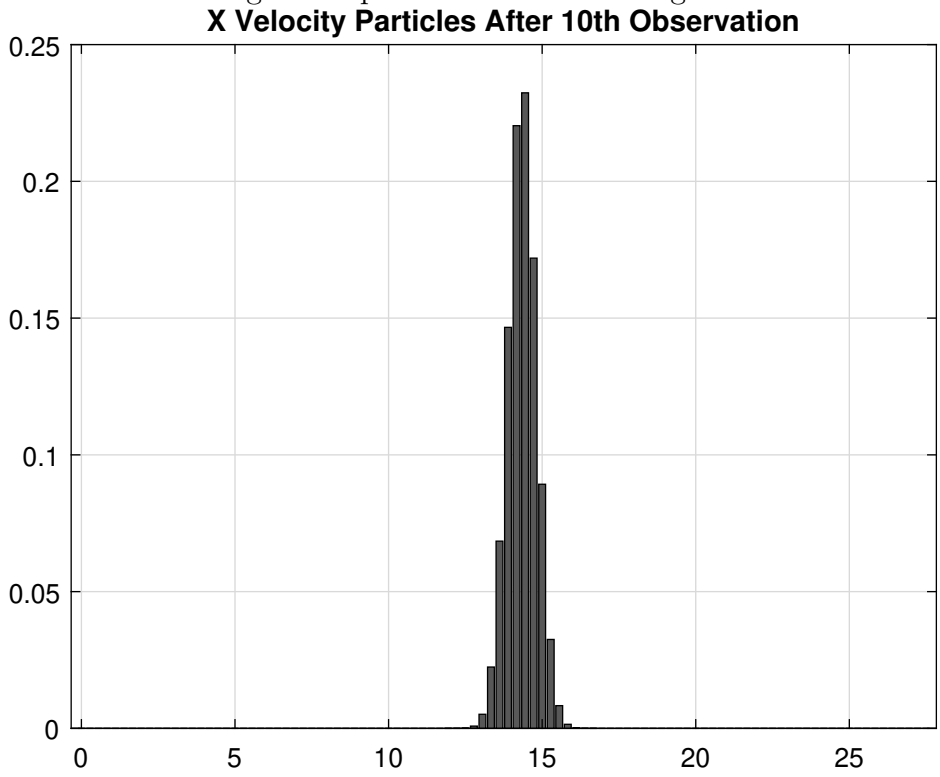

Fig. 27: Bar chart showing the X particle values and weights after the 10th iteration.

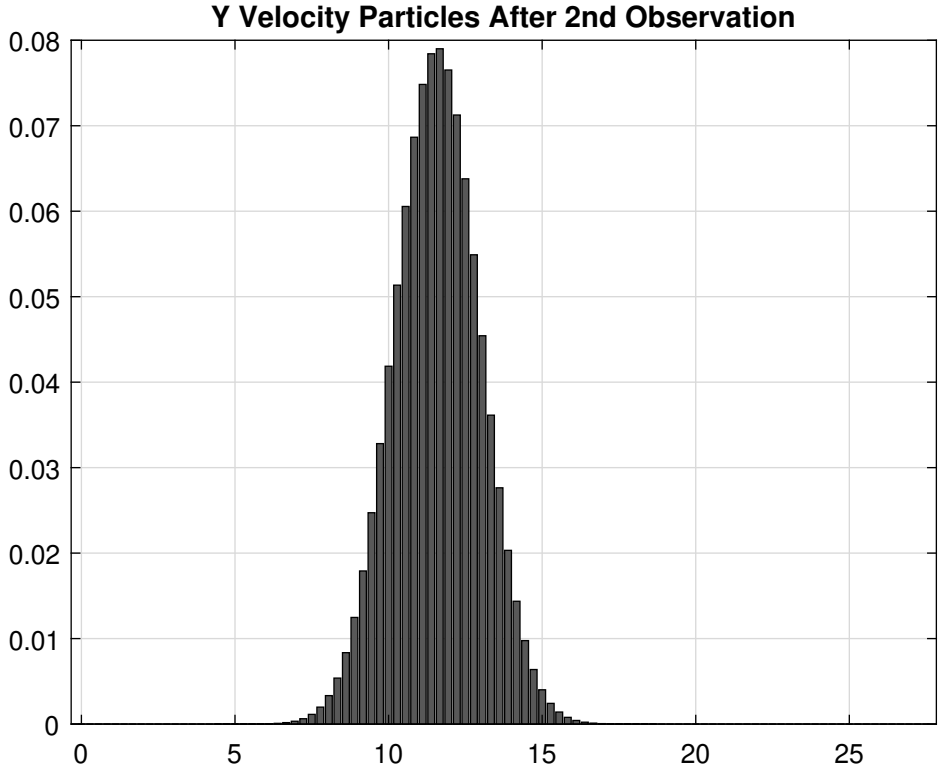

Fig. 28: Bar chart showing the Y particle values and weights after the 2nd iteration.

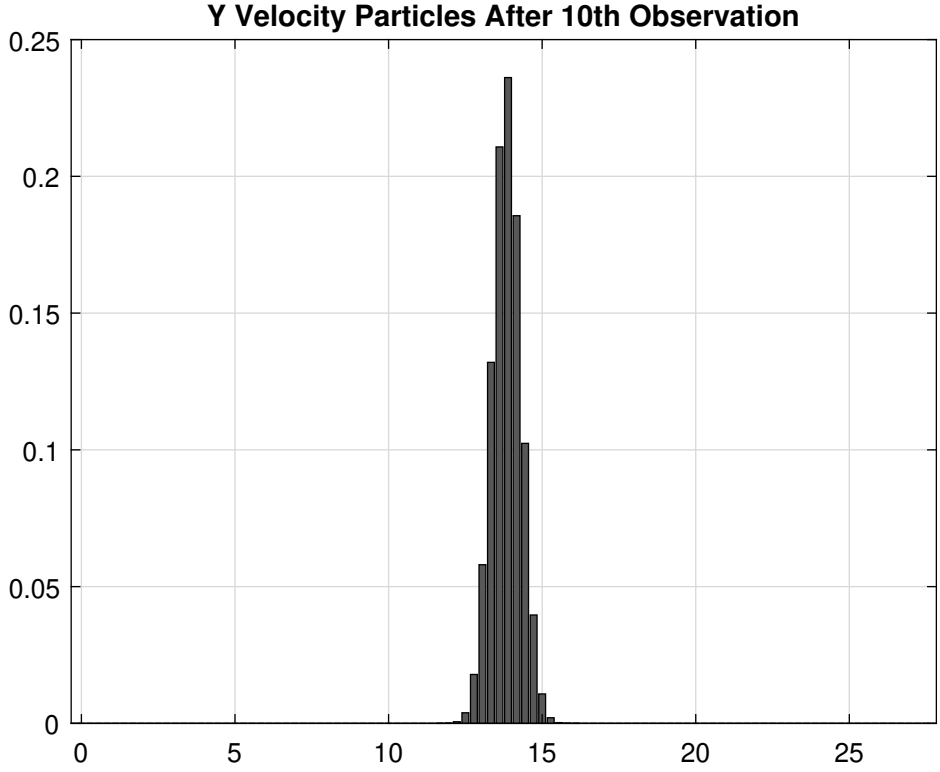

Fig. 29: Bar chart showing the Y particle values and weights after the 10th iteration.

CHAPTER 5

NUMERICAL RESULTS & SIMULATIONS

To test the effect of the noise (η_u) introduced by uncertainties regarding the vessels yaw position on performance, a model was set up using three vessels. One vessel served as the observation vessel and the other two vessels served as target craft as shown in Figure-30. The observation vessel moved straight up the center of the simulated environment. Vessel 1 moved across from the port side to the starboard side of the observer vessel. Vessel 2 passed by the observer vessel on the starboard side.

In the simulation, the observer vessel transited North at 20 knots, Vessel 1 transited to the East at 20 knots and Vessel 2 transited to the South at 20 knots. The tracks for the two different vessels were chosen so that they would generate different test cases for the tracker. With respect to vessel 1, it is on a course which intersects with the observer vessel. The bearing from the observer vessel to Vessel 1 remains the same until they intersect when it changes as they then move away from each other. The bearing from the observer vessel to Vessel 2 is constantly changing as they approach and then pass each other.

The revolutions per minute of the simulated RADAR on the observer vessel is was set to be 48 RPM. The simulation time was 7.5 minutes which resulted in 563 observations of the targets. A divide and conquer algorithm was used to try to find the maximum variance in yaw rate before the tracker failed using both the Kalman Filter and Particle Filter for state estimation. Each simulation had one hundred sub-simulations were run and the mean value

of variances that didn't break the track were recorded and calculated as one final simulation value.

The way the divide and conquer algorithm in this simulation was implemented was to first start with a yaw rate variation of 1°. If the multiple hypothesis algorithm was not able to maintain a track, the yaw rate variation was multiplied by half and tried again. If the multiple hypothesis tracker successfully maintained the track, then it would be multiplied by 150 percent. The Particle Filtering and Kalman Filtering state estimation aglorithms were run using this divide and conquer algorithm to achieve 100 successful runs. The yaw rate variations for those runs were logged and then the mean was taken on the results to indicate how much yaw rate variation would cause the tracking algorithm to fail 50% of the time.

To perform this analysis and determine the absolute best case scenarios the probability of detection of the RADAR system was set to be 100%. This means that every simulated observation was given to the tracker so that there would be no track propagations based off of a pure estimation. Since in a frequency modulated continuous wave RADAR system bandwidth determines range resolution, a bandwidth of 10 MHz was selected giving a range resolution of approximately 15 meters. The beam width of the RADAR antenna was simulated to be approximately 4° which matches the beam width of an X-band RADAR with about a 24 inch antenna aperture. For the particle filter implementation, the importance density function was implemented as the probability distribution function with a standard deviation set to the gating threshold. The gating threshold for this simulation was 30 meters which would essentially gate any craft going over 60 knots.

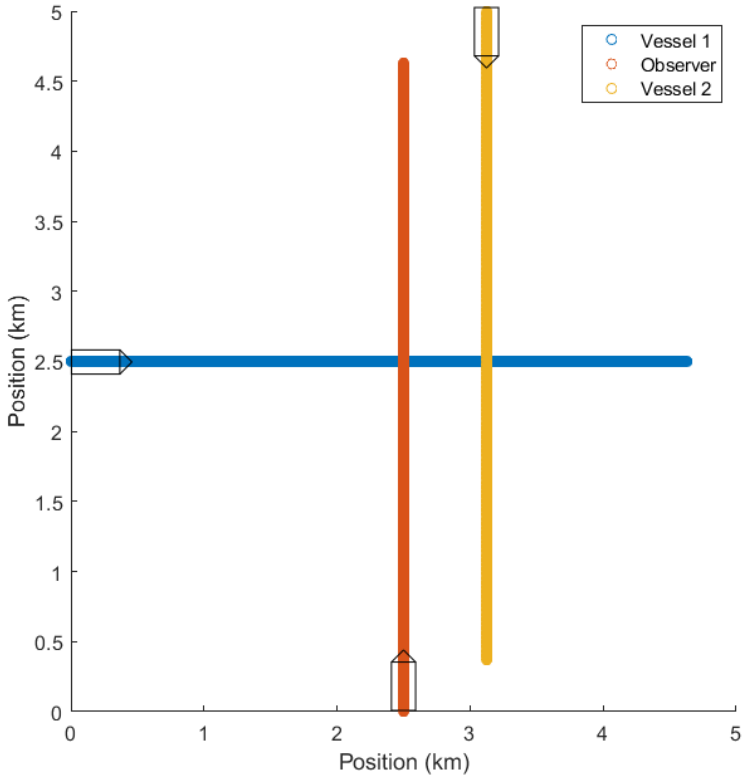

Fig. 30: Vessel starting positions and tracks throughout the simulated environment.

5.1 ANALYSIS OF RESULTS

When analyzing the performance of a tracking system, one of the most limiting factors is the gating threshold. Typically the gating threshold is selected to rule out targets that are moving too fast or too slow. The problem when there are uncertainties in bearing is that the noise itself can cause targets to appear to be moving much faster which then puts them outside of the gate. This leads to the tracker constantly creating new tracks and then dropping them. The following equation can be used to determine maximum theoretical variance given a gate threshold velocity V_g and expected velocity of target V_t at a given

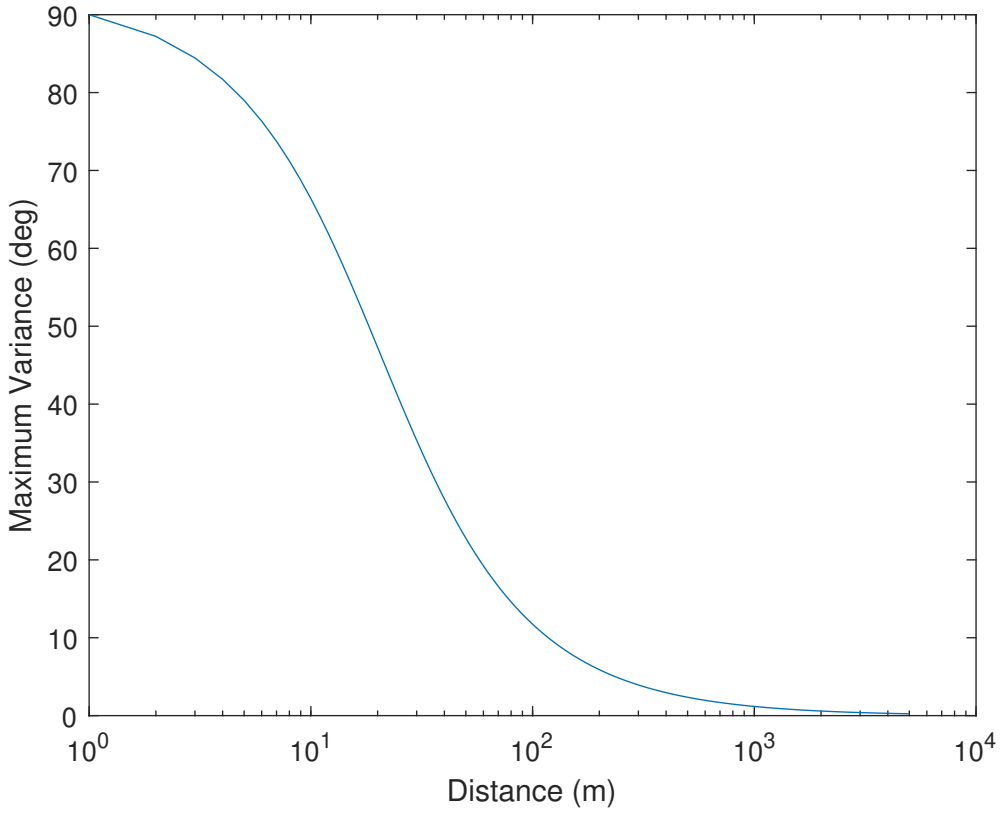

Fig. 31: Maximum Theoretical Variance allowed in Vessel Yaw Uncertainty for the simulation

range r.

$$\sigma^2_{\eta_{MAX}} = \arctan \frac{V_g - V_t}{r} \tag{74}$$

When using the parameters in this simulation, which are 20 knots expected target velocity and 60 knots gate threshold velocity, the maximum variance allowed quickly becomes small as range increases as shown in Figure-31.

Four total simulations were run for the Kalman Filter and the Particle Filter as the state

Track	Theoretical Limit
Vessel 1	0.3335°
Vessel 2	0.2340°

TABLE 5: Theoretical limit of yaw uncertainty compared to the mean of the simulated results where the tracker was still able to maintain a track and the tracker was using a particle filter for estimation.

	Kalman Filter		Particle Filter	
Simulation	Vessel 1	Vessel 2	Vessel 1	Vessel 2
1	0.0656°	0.0167°	0.0263°	0.00265°
2	0.0608°	0.0164°	0.0252°	0.00157°
3	0.0626°	0.0147°	0.0266°	0.00232°
4	0.0599°	0.0160°	0.0262°	0.00402°

TABLE 6: Simulation results for showing the maximum variance in yaw for the Multiple Hypothesis Tracker when using a Kalman Filter or Particle Filter for state estimation

estimation algorithms. These results are shown in Table-6.

The performance of the tracker is worse than the theoretical performance from the gating process. The performance of vessel 2 appears to be significantly worse than that of vessel 1. This can be attributed to the fact that the changes in angle to the observational vessel are more rapid than those of vessel 1 due to the closer distance as it is passing by. Ultimately what this means is that to track targets given a similar environment, orientation of the

vessel must be known at all times and kept less than the limits that were measured during the simulation. The only way that this can be achieved is to use a high accuracy gyroscope that is capable of producing updates much faster than the 1Hz update rate typically found with consumer GPS.

Fig-32 shows a comparison of the performance of the two filters showing the track of Vessel 1. It takes several observations for both of the filters to start converging. In this implementation, the Kalman Filter converges more quickly. The Particle Filter converges very well when the observer vessel is close to Vessel 1, but has more noise the farther away it is. Similar results are shown for Vessel 2 in Fig-33. Once the Kalman Filter converges it stays converged through the entire duration where in this implementation the Particle Filter grows more noisy the farther away the observer vessel is from the target. Ultimately, in this specific set of simulations, the Kalman Filter was able to track Vessel 1 when the vessel's yaw uncertainty was about 2.5 times higher than what the Particle Filter could track. The Kalman Filter was able to track Vessel 2 at rates of vessel yaw uncertainty being 6 times worse than what the particle filter could track.

Performance of the Particle Filter in this implementation was not as good as expected. This is likely due to sub-optimal resampling which could be improved using a technique which has improved sample diversity such as the regularized particle filter or Markov Chain Monte Carlo (MCMC) particle filter [47]. It is also possible that several of the common challenges with particle filtering such as Sample Impoverishment, Particle Filter Divergence and issues with the Importance Density may have contributed to the suboptimal performance [15].

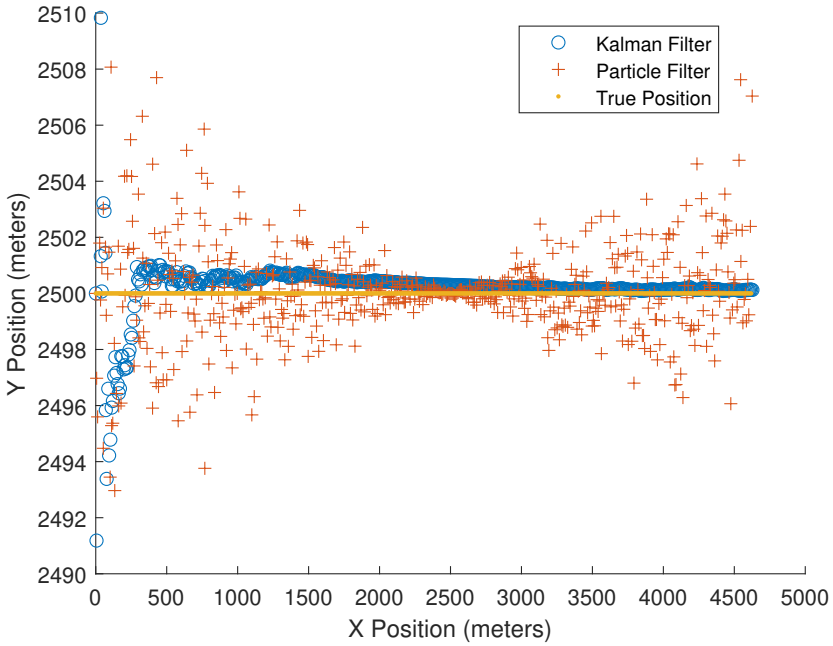

Fig. 32: Comparison of the track history of the Kalman Filter and Particle Filter when observing Vessel 1.

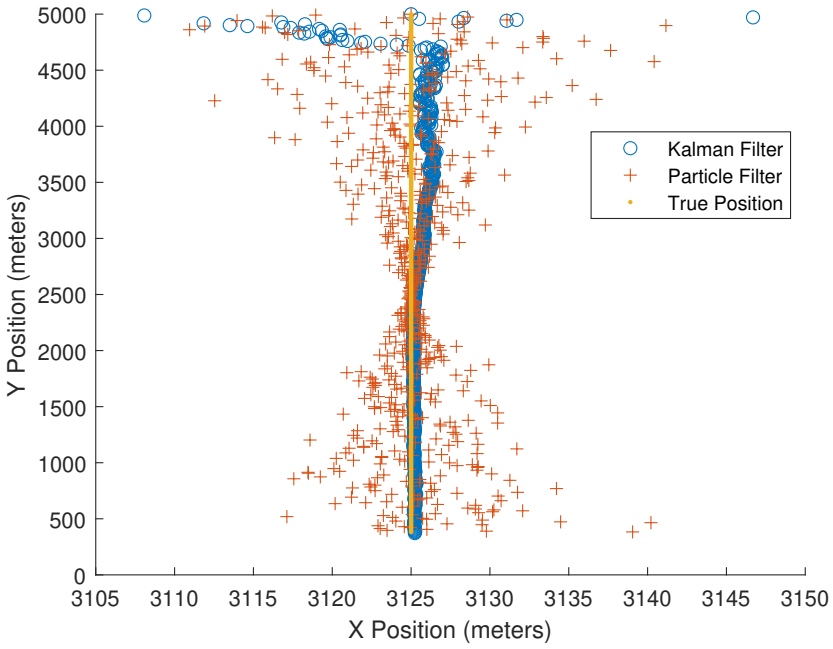

Fig. 33: Comparison of the track history of the Kalman Filter and Particle Filter when observing Vessel 2.

5.2 COMPUTATIONAL ANALYSIS

Kalman Filtering and Particle Filtering have different computational requirements. When dealing with large numbers of tracks, such as being in a congested harbor, a RADAR tracking system may encounter issues. This section will discuss the algorithms from a computational complexity standpoint and from the minimum number of floating point operations.

It is important to note that the amount of processing power required for these tracking algorithms is ultimately tied to the number of tracks the system is processing. The following analysis focuses on the computational complexity for one track, but that computational complexity would be then be multiplied by the total number of tracks. For a vessel far out at sea, the number of tracks would be very minimal and could possibly be zero. For dense environments, such as congested harbors, it is possible that the number of tracks is enough to overwhelm the processing system. Thus a system would need to be designed which could operate in the most dense track environments.

The absolute maximum worse case scenario for the number of tracks is the number of range bins times the number of spokes. In this case, which would never be encountered in the real world, every range bin in every spoke would be a possible target. Some of the COTS marine RADAR systems currently on the market return up to 2048 spokes with 512 range bins resulting in over a million possible tracks. Again, it is highly unlikely this would occur in real world operation due to the fact that the automatic target detection process would likely assume that the noise level is very high and would not work appropriately due to not being able to find an edge to perform the target detection.

5.2.1 COMPUTATIONAL COMPLEXITY

Kalman Filtering consists mainly of matrix multiplications performed in a sequence. Regardless of what the input values are, the Kalman filter will always perform the same routine. The Kalman filter has a computational complexity expressed in Big O notation as $O(1)$.

The Particle filter if implemented as its most basic concept would also have a computational complexity of $O(1)$. However, this system would rather quickly collapse. For realistic implementations, it is going to perform some sort of resampling when the number of effective particles have fallen below a certain value. If the importance density function is poorly chosen or the dynamics of the system cause the number of effective particles to drop below the threshold frequently, resampling may occur very often. The multinomial resampling algorithm is described as pseudocode in Appendix A where the cumulative summation function creates a vector corresponding to the summation of the weights. The last element in the vector is thus one. Random numbers are chosen between zero and one and are searched into the cumulative summation vector to determine the matching particle. This means that particles that have higher weights are chosen more and end up being in the new vector created to store the new particles chosen after resampling. This is illustrated in Fig-34.

As can be seen from the pseudocode in Appendix A, it is possible for the sampling function to be expressed as $O(N^2)$ in the worst case scenario.

5.2.2 NUMBER OF MULTIPLICATIONS

Modern processors typically require multiple clock cycles to multiply two floating point

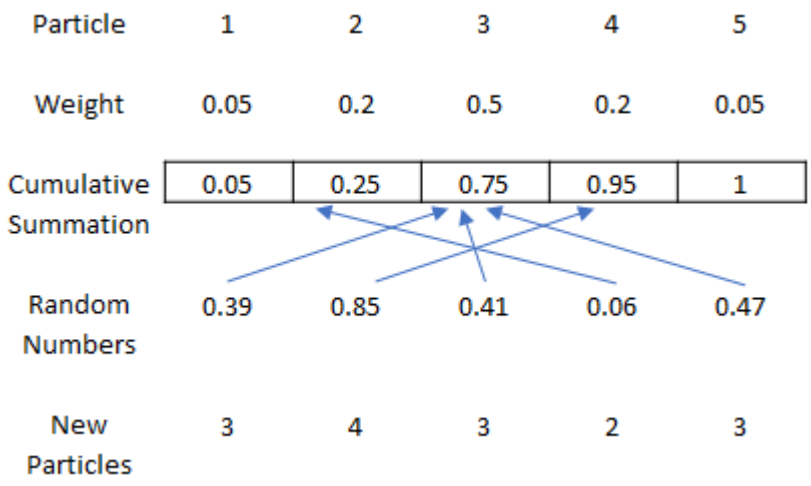

Fig. 34: The resampling algorithm works by using a cumulative summation function to build a vector. Random numbers are generated and searched inside the cumulative summation vector to determine the new particles.

numbers even if they support hardware floating point units. As an example, the AMD Zen4 architecture (used in processors such as the Ryzen 5 7600) requires 7 clock cycles to multiply two floating point numbers [48]. Low end processors may not possess a hardware floating point unit which would require multiplication of floating point numbers to be emulated using the supported integer operations. An example of this is the Raspberry Pi Pico which does not have a hardware floating point unit and requires approximately 7,687 clock cycles to multiply two floating point numbers[49] through a software technique. Ultimately, this means that the larger the number of multiplications required, the more clock cycles are required, and the more concerns there are regarding real time performance.

The implementation of the Kalman Filter for this two dimensional tracking implementation requires 376 multiplications in order to take a new observation, incorporate it, and

generate a new estimate. For a similar three dimensional tracking implementation, this number grows to 1,028 multiplications. Some of the matrices used by the Kalman Filter will be sparse in real world implementations, which means that the number of multiplications can be reduced. Since the Kalman filter always performs the same number of steps, it is easy to plan for the amount of computational time needed on a real time system.

The particle filter implementation for this dissertation requires 14 multiplications per particle per dimension for track estimation purposes. This means a two dimensional tracking implementation where 1,000 particles are used, such as in this dissertation, would require approximately 28,000 multiplications. Using 1,000 particles in a similar three dimensional implementation would require 42,000 multiplications per track update.

If we take the amount of time needed for a two dimensional state estimation filter to run, we can start to estimate the number of tracks updates that a processor can compute per second as shown in Table-7. At 48 revolutions per minute, new spoke data would be received every 1.25 seconds. A system implemented on either an AMD Ryzen^{TM} 5 7600 (which uses the Zen4 architecture) or a Raspberry Pi Pico would be capable of operating real time with only one track. The Raspberry Pi Pico would not be able to keep up with real time execution of the Particle Filtering implementation with more than one track and the Kalman Filter implementation would run into issues when dealing with around 54 tracks. It is important to note that these are actually the most optimistic numbers due to the fact that there will be overhead for program control and other operations.

Processor	Clock Speed	Multiplication Time	Kalman Filter updates/second	Particle Filter updates/second
AMD Ryzen^{TM} 5 7600	3800MHz	1.84ns	1,443,769	38,775
Raspberry Pi Pico	125MHz	61.5us	43.25	1.16

TABLE 7: Estimates for the number of track updates per second that two selected processors can perform using either the Kalman Filter or Particle Filter for state estimation and the implementation described in this dissertation. It is important to note that this assumes that floating point multiplication is the limiting factor and that time for process control and other mathematical operations are negligible. The current price for a Ryzen 5 7600 processor only is around $200 and the current price for a Raspberry Pi Pico is $4.

CHAPTER 6

CONCLUSIONS

This dissertation discusses how small marine RADAR with an automatic target detection capability could be used to improve safety on the water. There are many accidents that occur due to collisions between vessels and other objects. Using commercial off the shelf RADAR, low cost sensors for vessel attitude determination, and open source software, an affordable system could be built to automatically detect and warn operators of impending collisions. This dissertation presents a framework for how to accomplish this task. It starts by describing how RADAR data can be received from a commercial off the shelf (COTS) small marine RADAR and processed in the OpenCV computer vision library to detect targets. It then discusses how there is noise inherent in the system which arises from both the RADAR signal processing as well as from the ocean itself which causes there to be uncertainties related to where exactly targets may be. These uncertainties can be very large on smaller vessels which can experience yaw rate variations due to wave slams. The dissertation discusses how those targets plus their respective noises could be plotted as ellipses and presented to the operator.

A discussion of the Multiple Hypothesis Tracking (MHT) algorithm was performed. MHT requires the use of a state estimator which is implemented in the vast majority of systems as a Kalman Filter due to its relatively low computational complexity. The state estimator is used to predict the future location of a target so that in the event that an observation is lost, the Multiple Hypothesis Tracker can resume tracking if a future observation

is made. Analysis was performed to see if the much more computationally complex particle filtering state estimation algorithm would perform better. In the implementation used by this dissertation, the particle filtering algorithm did not perform as well as the Kalman Filtering algorithm. The computational complexity of the Kalman Filter and the Particle Filter was discussed and some estimates of performance on a modern mid-range processor and extreme low end processor were presented.

6.1 FUTURE WORK

The next steps for this work would be to actually implement the system. This could begin by purchasing a commercial off the shelf marine RADAR and mounting it near a waterway to start collecting data. Commercial vessels broadcast their locations using the Automatic Identification System (AIS) at regular intervals on a radio-frequency channel. This information could be used with the RADAR data to provide a ground truth location for the received targets. This collected data could be used to determine better thresholds for the automatic target detection process. An analysis of the detected targets and the state estimation filters could lead to performance improvements. In the Kalman Filter implementation, this would likely be a better selection for the initial value of the covariance matrix. For the Particle Filter implementation, real world data could lead to a better selection of the number of particles, the importance density functions as well as how much process noise should be added. This resulting data could be published for others to analyze and use in their own simulations. Once the performance parameters have been tuned, the next step would be to move the system to a vessel.

Once the system is moved onto a vessel, one of the first things that should be analyzed

is the yaw rate variation due to wave slams. There is currently almost no literature which discusses these effects on small vessels. The data should be measured using low cost MEMs devices such as the ones used in cell phones as well as tactical grade optical based systems which would serve as a baseline. This data could be used to build a model which would describe what the dynamic heading accuracy of different orientation filters. While in this implementation, the Particle Filter didn't perform as well as the Kalman Filter for state estimation for MHT, it is possible that Particle Filtering may have benefits in navigation filters due to the larger prevalence of nonlinearities.

Finally, newer types of navigation technology could also be explored. It is possible in the near future that MEMS technology may be replaced with other types of technology. New fabrication techniques may make Hemispheric Resonator Gyroscopes more available which may provide better performance and reliability than traditional MEMS devices [50]. Others are also exploring AI/ML techniques for inertial navigation which may lead to increased performance for determination of vessel orientation[51].

REFERENCES

[1] M. I. Skolnik, *Introduction to RADAR Systems*. Singapore: McGraw-Hill Book Co, 1981.

[2] US Department of Commerce, NOAA, "Beaufort wind scale," Sep 2016.

[3] United States Coast Guard, "2022 recreational boating statistics," 2023.

[4] N. O. Abankwa, S. J. Johnston, M. Scott, and S. J. Cox, "Ship motion measurement using an inertial measurement unit," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pp. 375–380, 2015.

[5] X. Yi, X. Wu, X. Yue, L. Zhang, Z. Chen, and B. Wan, "Ocean surface current inversion with anchored floating high-frequency radar: Yaw compensation," *IEEE Journal of Oceanic Engineering*, vol. 46, no. 3, pp. 927–939, 2021.

[6] N. Yazdi, F. Ayazi, and K. Najafi, "Micromachined inertial sensors," *Proceedings of the IEEE*, vol. 86, no. 8, pp. 1640–1659, 1998.

[7] Z. Lin, Y. Xiong, H. Dai, and X. Xia, "An experimental performance evaluation of the orientation accuracy of four nine-axis mems motion sensors," in *2017 5th International Conference on Enterprise Systems (ES)*, pp. 185–189, 2017.

[8] K. Rao, X. Wei, S. Zhang, M. Zhang, C. Hu, H. Liu, and L.-C. Tu, "A mems micro-g capacitive accelerometer based on through-silicon-wafer-etching process," *Micromachines*, vol. 10, no. 6, 2019.

[9] L. Wu, Z. Tian, D. Ren, and Z. You, "A miniature resonant and torsional magnetometer based on lorentz force," *Micromachines*, vol. 9, no. 12, 2018.

[10] P. Patonis, P. Patias, I. N. Tziavos, D. Rossikopoulos, and K. G. Margaritis, "A fusion method for combining low-cost imu/magnetometer outputs for use in applications on mobile devices," *Sensors*, vol. 18, no. 8, 2018.

[11] G. Tomasch and K. Winer, "Limits of absolute heading accuracy using inexpensive mems sensors," *Hackaday Journal of What You Don't Know*, vol. 1, no. 2, 2019.

[12] S. Recouvreur, "Mems vs fog: What inertial system should you choose?," Sep 2023.

[13] Y. N. Korkishko, V. A. Fedorov, V. E. Prilutskii, V. G. Ponomarev, I. V. Morev, S. F. Skripnikov, M. I. Khmelevskaya, A. S. Buravlev, S. M. Kostritskii, I. V. Fedorov, A. I. Zuev, and V. K. Varnakov, "Strapdown inertial navigation systems based on fiber-optic gyroscopes," *Gyroscopy and navigation (Online)*, vol. 5, no. 4, pp. 195–204, 2014.

[14] H. C. Lefevre, *The Fiber-Optic Gyroscope, Third Edition*. Norwood: Artech House, 3 ed., 2022.

[15] J. Elfring, E. Torta, and R. van de Molengraft, "Particle filters: A hands-on tutorial," *Sensors*, vol. 21, no. 2, 2021.

[16] J. S. Harris and D. C. Popescu, "Process for automatic target detection using small marine radar," in *IEEE SOUTHEASTCON 2022*, 2022.

[17] J. S. Harris and D. C. Popescu, "Evaluation of vessel yaw uncertainties on multiple hypothesis tracker performance," in *OCEANS 2022, Hampton Roads*, pp. 1–6, 2022.

[18] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, New Jersey: Prentice Hall, 1975.

[19] W. L. Stutzman and G. A. Thiele, *Antenna Theory and Design*. Hoboken, New Jersey: John Wiley and Sons, Inc, 3rd ed., 2013.

[20] C. A. Balanis, *Antenna Theory Analysis and Design*. Hoboken, New Jersey: John Wiley and Sons, Inc, 4th ed., 2016.

[21] C. A. Balanis, *Advanced Engineering Electromagnetics*. Hoboken, New Jersey: John Wiley and Sons, Inc, 2nd ed., 2012.

[22] "Fcc online table of frequency allocations," tech. rep., Federal Communications Commission, 2021.

[23] C. Alexandrov, A. Draganov, and N. Kolev, "An application of automatic target recognition in marine navigation," in *Proceedings International Radar Conference*, pp. 250–255, 1995.

[24] J. Han, S. Y. Kim, and J. Kim, "Enhanced target ship tracking with geometric parameter estimation for unmanned surface vehicles," *IEEE Access*, vol. 9, pp. 39864–39872, 2021.

[25] F.-Y. Kuo and R.-B. Hwang, "High-isolation x-band marine radar antenna design," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 5, pp. 2331–2337, 2014.

[26] H. M. El Misilmani, M. Al-Husseini, K. Y. Kabalan, and A. El-Hajj, "A design procedure for slotted waveguide antennas with specified sidelobe levels," in *2014 International Conference on High Performance Computing Simulation (HPCS)*, pp. 828–832, 2014.

[27] J. W. Lewis, "Shock and vibration environmental test of noaa 29-foot hydrographic survey launch," 1984.

[28] J. Tessendorf, "Simulating Ocean Water." Available online at `https://people.cs.clemson.edu/~jtessen/reports/papers_files/\coursenotes2004.pdf`. Accessed: January 8, 2022.

[29] X. Yi, X. Wu, X. Yue, L. Zhang, Z. Chen, and B. Wan, "Ocean surface current inversion with anchored floating high-frequency radar: Yaw compensation," *IEEE Journal of Oceanic Engineering*, vol. 46, no. 3, pp. 927–939, 2021.

[30] I. R. Young, W. Rosenthal, and F. Ziemer, "A three-dimensional analysis of marine radar images for the determination of ocean wave directionality and surface currents," *Journal of Geophysical Research*, vol. 90, pp. 1049–1059, 1985.

[31] D. R. Lyzenga and D. T. Walker, "A simple model for marine radar images of the ocean surface," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 12, pp. 2389–2392, 2015.

[32] D. Yulian, R. Hidayat, H. A. Nugroho, A. A. Lestari, and F. Prasaja, "Automated ship detection with image enhancement and feature extraction in fmcw marine radars,"

in *2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, pp. 58–63, 2017.

[33] J. Ryan, "Detection of small radar cross section targets at sea using coherent radar," in *International Conference on Radar Systems (Radar 2017)*, pp. 1–6, 2017.

[34] Y. Lei, L. Si, L. Sun, and F. Xu, "Ship detection in marine radar images based on a modified yolov3-tiny," in *IET International Radar Conference (IET IRC 2020)*, vol. 2020, pp. 1554–1560, 2020.

[35] Y. Shijun, C. Jinbiao, and S. Chaojian, "A data fusion algorithm for marine radar tracking," in *2009 WRI Global Congress on Intelligent Systems*, vol. 1, pp. 234–238, 2009.

[36] M. Hu, C. Zheng, and L. Sun, "Multi-task based marine radar tracking network," in *2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS)*, pp. 616–620, 2021.

[37] Raymarine, "Marine radar software development kit."

[38] OpenCPN, "Radar pi plugin."

[39] M. A. Richards, *Fundamentals of Radar Signal Processing.* New York, New York: McGraw Hill, 2005.

[40] S. Suzuki and K. be, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.

[41] D. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[42] S. Blackman, *Multiple-Target Tracking with Radar Applications*. Artech House, 1986.

[43] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Boston: Artech House, 1999.

[44] J. R. Werthmann, "Step-by-step description of a computationally efficient version of multiple hypothesis tracking," in *Signal and Data Processing of Small Targets 1992*, vol. 1698, pp. 288–300, International Society for Optics and Photonics, 1992.

[45] D. Simon, *Optimal State Estimation*. Hoboken, New Jersey: John Wiley & Sons, Inc, 2006.

[46] Z. Duan, C. Han, and X. Rong Li, "Comments on "unbiased converted measurements for tracking"," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 4, pp. 1374–, 2004.

[47] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Boston, MA: Artech House, 2004.

[48] A. Fog, "Lists of instruction latencies, throughputs and micro-operation breakdowns for intel, amd, and via cpus," 2022.

[49] Raspberry Pi Ltd., "Raspberry pi pico c/c++ sdk," 2023.

[50] A. R. Ranji, V. Damodaran, K. Li, Z. Chen, S. Alirezaee, and M. J. Ahamed, "Recent advances in mems-based 3d hemispherical resonator gyroscope (hrg)-a sensor of choice," *Micromachines*, vol. 13, no. 10, 2022.

[51] S. Recouvreur, "How is ai revolutionising inertial navigation?," June 2023.

APPENDIX A

PARTICLE RESAMPLING PSEUDOCODE

```
c = CUMSUM(weights)                          ▷ Cumulative Summation Function
newParticles = ZEROS(1, N)                   ▷ Generate an all zero vector to store results
for i = 1, ..., N do
    rn = RAND(1)                             ▷ Generate one random number between 0 and 1
    index ← 1
    for j = 1, ..., N do                     ▷ Search for the index
        if rn <= c[j] then
            index ← j
            break
        end if
    end for
    newParticles[i] = oldParticles[index]
end for
```

VITA

Jason Stark Harris
Department of Electrical and Computer Engineering
Old Dominion University
Norfolk, VA 23529

Education

- Ph.D. Electrical and Computer Engineering, December 2023, Old Dominion University
- Master of Business Administration, May 2021, University of Massachusetts Amherst
- M.S. Electrical and Computer Engineering, August 2016, Old Dominion University
- B.S. Electrical Engineering, December 2014, Old Dominion University

Select Publications

- J. S. Harris and D. C. Popescu, "Evaluation of Vessel Yaw Uncertainties on Multiple Hypothesis Tracker Performance," OCEANS 2022, Hampton Roads, VA, USA, 2022
- J. S. Harris and D. C. Popescu, "Process for Automatic Target Detection Using Small Marine RADAR," SoutheastCon 2022, Mobile, AL, USA, 2022
- A. G. Cappiello, D. C. Popescu, J. S. Harris and O. Popescu, "Radio Link Design for CubeSat-to-Ground Station Communications Using An Experimental License," 2019 International Symposium on Signals, Circuits and Systems (ISSCS), Iasi, Romania, 2019
- M. W. O'Brien, J. S. Harris, O. Popescu and D. C. Popescu, "An Experimental Study of the Transmit Power for a USRP Software-Defined Radio," 2018 International Conference on Communications (COMM), Bucharest, Romania, 2018
- O. Popescu, J. S. Harris and D. C. Popescu, "Designing the communication sub-system for nanosatellite CubeSat missions: Operational and implementation perspectives," SoutheastCon 2016, Norfolk, VA, USA, 2016

Typeset using LaTeX.