Target Tracking in Multi-Static Active Sonar Systems
Using Dynamic Programming and Hough Transform

by

Mohammad El-Jaber

A thesis submitted to the Department of Electrical and Computer Engineering
In conformity with the requirements for
the degree of
Master of Science (Engineering)

Queen’s University
Kingston, Ontario, Canada
(July, 2009)

Copyright ©Mohammad El-Jaber, 2009
Abstract

Tracking multiple targets in a high cluttered environment where multiple receivers are used is a challenging task due to the high level of false alarms and uncertainty in the track hypothesis. The multi-static active sonar scenario is an example for such systems where multiple source-receiver combinations are deployed. Due to the nature of the underwater environment and sound propagation characteristics, tracking targets in the underwater environment becomes a complex operation.

Conventional tracking approaches (such as the Kalman and particle filter) require a predetermined kinematic model of the target. Moreover, tracking an unknown and changing number of targets within a certain search area requires complex mathematical association filters to identify the number of targets and associate measurements to different target tracks. As the number of false detections increases, the computational complexity of conventional tracking system grows introducing further challenges for real-time target tracking situations.

The methodology presented in this thesis provides a rapid and reliable tracking system capable of tracking multiple targets without depending on a kinematic model of the target movement. In this algorithm, Self Organizing Maps, Dynamic Programming and the Hough transform are combined to produce tracks of possible targets’ paths and estimate of targets’ locations. Evaluation of the performance of the tracking algorithm is performed using three types of simulations and a set of real data obtained from a sea trial. This research documents the results of experimental testing and analysis of the tracking system.
Acknowledgements

I would like to extend my gratitude to my thesis supervisors, Dr. Aboelmagd Noureldin and Dr. Mohamed Bayoumi of Queen’s University. Their support and guidance over the course of the last two years have been unwavering. Without their advices, patience and optimism, this research would not have been completed.

The sea acoustic simulation experiments used in this research would not have been possible if it were not for the collaboration of the members of the Navigation and Instrumentation Research Group at the Royal Military College of Canada. I would especially like to thank Mr. Abdalla Osman for offering his assistance.

Finally, I would like to thank the Mr. Scott Campbell, Mr. Jim Theriault and Dr. Garfield Mellema from Defence Research and Development Canada (DRDC) for their guidance through the different phases of the research and providing data from field trials (SEABAR’07). Without their collaboration, the results of this research could not have been achieved. The SEABAR’07 sea trial was held by the NATO Undersea Research Centre (NURC) and was made available under agreement with NURC and Defence Research and Development Canada – Atlantic.
Table of Contents

Abstract............................................................................................................................................ ii
Acknowledgements......................................................................................................................... iii
Table of Contents ............................................................................................................................ iv
List of Figures ................................................................................................................................ vii
List of Abbreviations and Acronyms ............................................................................................... x
Chapter 1 Introduction ..................................................................................................................... 1
  1.1 Background ............................................................................................................................ 1
    1.1.1 Anti Submarine Warfare (ASW)..................................................................................... 2
    1.1.2 Sonobuoy Devices in ASW............................................................................................. 3
    1.1.3 Multistatic Active Sonobuoy fields................................................................................. 6
  1.2 Problem Statement ................................................................................................................. 7
    1.2.1 Multiple Target Tracking using Dynamic Programming ................................................ 9
    1.2.2 Experimental Validation using Simulation Data............................................................. 9
    1.2.3 Experimental Validation using Real Data....................................................................... 9
  1.3 Scope of Research ................................................................................................................ 10
  1.4 Thesis Outline ...................................................................................................................... 11
Chapter 2 Literature Review .......................................................................................................... 13
  2.1 Introduction .......................................................................................................................... 13
  2.2 Active Sonar ........................................................................................................................ 13
    2.2.1 Monostatic Active Sonar ............................................................................................... 14
    2.2.2 Bistatic Active Sonar .................................................................................................... 23
    2.2.3 Multistatic Active Sonar ............................................................................................... 25
  2.3 Target Tracking .................................................................................................................... 33
    2.3.1 Problem Formulation .................................................................................................... 34
    2.3.2 Single-Target state estimation filters: ........................................................................... 35
      2.3.2.1 Kalman Filter ......................................................................................................... 35
      2.3.2.2 Extended Kalman filter ........................................................................................ 36
      2.3.2.3 Unscented Kalman filter ......................................................................................... 37
      2.3.2.4 Particle Filter .......................................................................................................... 37
    2.3.3 Multi target tracking and data association .................................................................... 38
      2.3.3.1 Joint Probability Data Association Filter (JPDAF)................................................ 40
6.4 Recommendations and Future work ................................................................. 120
References .............................................................................................................. 122
List of Figures

Figure 1.1 Sonobuoy loaded on aircraft................................................................. 4
Figure 1.2 Aircraft launched sonobuoy................................................................. 4
Figure 1.3 GPS sonobuoy concept....................................................................... 6
Figure 1.4 Multistatic Active Sonobuoy scenario................................................. 7
Figure 2.1 Monostatic Active Sonar system......................................................... 15
Figure 2.2 Target localization in monostatic active sonar systems................. 15
Figure 2.3 Aspect variation of Submarine Target Strength.................................. 18
Figure 2.4 Echo, Noise and Reverberation as a Function of Range................. 21
Figure 2.5 Range Limiting Variations............................................................... 22
Figure 2.6 Bistatic Active Sonar system............................................................ 23
Figure 2.7 The bistatic geometry........................................................................ 25
Figure 2.8 Equal-Time-of-Arrival Ellipse......................................................... 26
Figure 2.9 Specular Reflection Angle in MSA fields......................................... 28
Figure 2.10 Target localization in MSA sonobuoy fields................................. 29
Figure 2.11 Range and Bearing errors............................................................ 31
Figure 2.12 Localization error due to sonobuoy positioning error.................. 32
Figure 2.13 Block Diagram of track-oriented MHT algorithm...................... 42
Figure 3.1 SOM Node Activation by sonobuoys readings............................. 48
Figure 3.2 Using SOMs to discretize data obtained by sonobuoys................. 49
Figure 3.3 Hierarchal SOMs for discretizing two-dimensional data points......... 50
Figure 3.4 Labelling connected clusters......................................................... 51
Figure 3.5 Deciding the size of the search matrix.......................................... 54
Figure 3.6 Directions used in DP tracking....................................................... 56
Figure 3.7 Penalty Matrices for Directions 1:9............................................... 58
Figure 3.8 Using Hough transform for identifying straight target-paths........... 67
Figure 4.1 Contact level Simulation - Targets' paths....................................... 69
Figure 4.2 Contact level Simulation - Ideal contact data............................... 70
Figure 4.3 Contact level simulation - Inaccurate contact data....................... 70
Figure 4.4.a Contact level simulation - Scenario 1........................................ 71
Figure 4.4.b Contact level simulation - Scenario 2......................................... 71
Figure 4.5.a Transmission loss versus range and depth for a source of frequency 200 Hz and at depth of 100 m ............................................................................................................................... 73
Figure 4.5.b Transmission loss versus range for a source of frequency 200 Hz and at depth of 100 m .................................................................................................................................................... 73
Figure 4.6 Scenario of 1 transmitter and 4 x 4 array of receivers .................................................. 76
Figure 4.7.a Seabar data analysis - Transmitter position ............................................................................. 78
Figure 4.7.b Seabar data analysis - Transmitter position – zoom in ............................................................ 78
Figure 4.8 Speed of sound propagation in water - RX2 ........................................................................ 79
Figure 4.9 Speed of sound propagation in water - RX3 ........................................................................ 80
Figure 4.10 SEABAR A56 CW data - SNR threshold = 15 db .......................................................... 81
Figure 4.11.a SNR levels vs time of arrival - RX3 ..................................................................................... 82
Figure 4.11.b SNR levels vs time of arrival - RX3 – zoom in ................................................................... 82
Figure 4.12.a SNR levels vs time of arrival - RX2 ..................................................................................... 83
Figure 4.12.b SNR levels vs time of arrival - RX2 – zoom in ................................................................... 83
Figure 5.1.a Input to the tracker containing 6 targets ............................................................................ 85
Figure 5.1.b Tracker output for 6 targets ................................................................................................. 85
Figure 5.2 Tracker output for ideal data ................................................................................................. 87
Figure 5.3 Tracker output for inaccurate data .......................................................................................... 87
Figure 5.4 Ideal data - Two false contacts per Sonobuoy – Frame 29 .................................................. 89
Figure 5.5 Ideal data - Two false contacts per Sonobuoy – Frame 41 .................................................. 90
Figure 5.6 Ideal data - Two false contacts per Sonobuoy – Frame 60 .................................................. 90
Figure 5.7 False Alarams - Ideal data - Two false contacts per sonobuoy .................................................. 92
Figure 5.8 Target position error - Ideal data - Two false contacts per sonobuoy ........................................... 92
Figure 5.9 Ideal data - Five false contacts per Sonobuoy – Frame 24 ...................................................... 93
Figure 5.10 Ideal data - Five false contacts per Sonobuoy – Frame 41 ...................................................... 94
Figure 5.11 Ideal data - Five false contacts per Sonobuoy – Frame 60 ...................................................... 94
Figure 5.12 False alarms - Ideal data - Five false contacts per Sonobuoy .................................................. 95
Figure 5.13 Target position error - Ideal data - Five false contacts per Sonobuoy ........................................... 95
Figure 5.14 Inaccurate data - Two false contacts per Sonobuoy – Frame 29 ........................................... 96
Figure 5.15 Inaccurate data - Two false contacts per Sonobuoy – Frame 45 ........................................... 97
Figure 5.16 Inaccurate data - Two false contacts per Sonobuoy – Frame 60 ........................................... 98
Figure 5.17 False alarms - Inaccurate data - Two false contacts per Sonobuoy ........................................... 99
Figure 5.18 Target position error - Inaccurate data - Two false contacts per Sonobuoy ............... 99
Figure 5.19 Inaccurate data - Five false contacts per Sonobuoy – Frame 29 ...................... 100
Figure 5.20 Inaccurate data - Five false contacts per Sonobuoy – Frame 41 ...................... 100
Figure 5.21 Inaccurate data - Five false contacts per Sonobuoy – Frame 60 ...................... 101
Figure 5.22 False alarms - Inaccurate data - Five false contacts per Sonobuoy ................. 102
Figure 5.23 Target position error - Inaccurate data - Five false contacts per Sonobuoy ...... 102
Figure 5.24 Acoustic level simulation - Scenario 1 .............................................................. 103
Figure 5.25 Tracker output - Acoustic level simulation - Scenario 1 ..................................... 104
Figure 5.26 Acoustic level simulation - Scenario 2 .............................................................. 105
Figure 5.27.a Tracker output for Scenario 2: \( p=0.4 \) – Frame 16 ...................................... 106
Figure 5.27.b Tracker output for Scenario 2: \( p=0.4 \) – Frame 41 ...................................... 106
Figure 5.27.c Tracker output for Scenario 2: \( p=0.4 \) – Frame 53 ...................................... 106
Figure 5.28.a Tracker output for Scenario 2: \( p=0.8 \) – Frame 14 ...................................... 109
Figure 5.28.b Tracker output for Scenario 2: \( p=0.8 \) – Frame 48 ...................................... 109
Figure 5.28.c Tracker output for Scenario 2: \( p=0.8 \) – Frame 60 ...................................... 109
Figure 5.29 Acoustic level simulation - Maximum SNR levels in dB acquired by sonobuoys... 110
Figure 5.30 Acoustic level simulation - Number of false contacts acquired by sonobuoys ...... 110
Figure 5.31 Acoustic level simulation - Error in tracker position estimation .......................... 111
Figure 5.32 Acoustic level simulation - Number of false alarms .......................................... 111
Figure 5.33 Run A56 data after thresholding ........................................................................ 112
Figure 5.34 Tracking output for run A56 ............................................................................. 113
Figure 5.35 Tracker output for run A56 compared with target GPS position ...................... 114
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASW</td>
<td>Anti Submarine Warfare</td>
</tr>
<tr>
<td>MSA</td>
<td>Multi-Static Active sonar</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>SE</td>
<td>Signal Excess (echo excess) (dB)</td>
</tr>
<tr>
<td>SL</td>
<td>Source Level (dB)</td>
</tr>
<tr>
<td>TS</td>
<td>Target Strength (dB)</td>
</tr>
<tr>
<td>PL</td>
<td>Propagation Loss (dB)</td>
</tr>
<tr>
<td>RD</td>
<td>Recognition Differential (dB)</td>
</tr>
<tr>
<td>RL</td>
<td>Reverberation Level (dB)</td>
</tr>
<tr>
<td>AN</td>
<td>Ambient Noise (dB)</td>
</tr>
<tr>
<td>DI</td>
<td>Receiver Directivity Index (dB)</td>
</tr>
<tr>
<td>TT</td>
<td>Target Tracking</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Density Function</td>
</tr>
<tr>
<td>KF</td>
<td>Kalman Filter</td>
</tr>
<tr>
<td>EKF</td>
<td>Extended Kalman Filter</td>
</tr>
<tr>
<td>UKF</td>
<td>Unscented Kalman Filter</td>
</tr>
<tr>
<td>PF</td>
<td>Particle Filter</td>
</tr>
<tr>
<td>MTT</td>
<td>Multiple Target Tracking</td>
</tr>
<tr>
<td>JPDAF</td>
<td>Joint Probability Data Association Filter</td>
</tr>
<tr>
<td>MHT</td>
<td>Multi Hypothesis Tracking</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>PMHT</td>
<td>Probabilistic Multi Hypothesis tracking</td>
</tr>
<tr>
<td>DMHT</td>
<td>Distributed Multi Hypothesis Tracking</td>
</tr>
<tr>
<td>PHD</td>
<td>Probability Hypothesis Density</td>
</tr>
<tr>
<td>CPHD</td>
<td>Cardinalized Probability Hypothesis Density</td>
</tr>
<tr>
<td>GM – CPHD</td>
<td>Gaussian Mixture Cardinalized Probability Hypothesis Density</td>
</tr>
<tr>
<td>DP</td>
<td>Dynamic Programming</td>
</tr>
<tr>
<td>SOM</td>
<td>Self Organizing Map</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background

In many applications where signals are propagated in the underwater environment, utilizing sound is favored over other radiation sources such as electromagnetic waves. Sound is known to have a robust performance when undergoing the harsh underwater conditions and attenuation profiles. Due to its relative ease of use and propagation, sound has been applied to a variety of purposes and for the exploration of the seas. The technique that uses sound propagation under water to navigate, communicate or to detect objects is called SONAR which is an acronym for sound navigation and ranging [1].

Based on the functionality of transmitting or receiving, sonar systems can be classified as active or passive. Sonar systems, equipments, and devices are said to be active when sound is purposely generated by one of the system components called the projector. Active sonar systems are said to echo-range on their targets. Passive or listening sonar systems use sound radiated (usually unwittingly) by the target. In this case, only one-way transmission through the sea is involved, and the system centers around the hydrophone used to listen to the target sounds. Communication, telemetry, and control applications employ a hybrid form of sonar system using a projector and hydrophone at both ends of the acoustic communication path [1]. The research documented in this thesis will be focusing on active sonar systems.
Based on their applications, systems utilizing sonar can be classified into civilian and military applications. Civilian applications of underwater sound include acoustic devices for navigation and localization, remote control and monitoring of underwater equipments, location and identification of fish, and routine and emergency communications. These devices are used for scientific, commercial and recreational exploitation of the oceans [2, 3]. Military applications of underwater sound include acoustic mines (mines that explode when the acoustic level in their pass-band reaches a certain value) and detection of underwater objects (ships and submarines) [1]. This research focuses on the application of underwater target tracking in antisubmarine warfare.

1.1.1 Anti Submarine Warfare (ASW)

With the evolution of submarines, interest in the development of sonar systems and techniques for underwater target detection has expanded. Early attempts for developing of sonar systems started at the beginning of the twentieth century, followed by several attempts which were highly encouraged by the World wars. Active (asdic) sonar was the basis for many British and American developed systems. Meanwhile, the German systems tended to be based on passive sonar [4].

In 1940, the research in the area of ASW and the need to find sonar system that can detect German submarines led to the introduction of Omni-directional and directional sonar systems called “sonobuoys” [4]. The motivation for the development of sonobuoys was to have an expendable sonar system that can be deployed from ships behind convoys
and detect German submarines. In 1942 sonobuoys were redesigned to be dropped from airplanes and by 1943 these devices were available for deployment in large quantities [Holler06].

Over the next decades, sonobuoy systems were enhanced and new types of sonobuoys were developed. This research emphasizes on the applications of utilizing active sonar systems where at least one transmitter sonobuoys and multiple receiver sonobuoys are deployed in what is known as the multistatic active sonar scenario [5].

1.1.2 Sonobuoy Devices in ASW

A sonobuoy is mainly a sonar system that is relatively small, and can be deployed from a platform to become submerged in the water and that are used for echo ranging or for listening and providing information about local sound amplitude, as a function of frequency and time. Figure 1.1 [6] shows the loading of an aircraft with sonobuoys.

Sonobuoys can be classified into three categories according to their functionality:

1. Active Sonobuoy which emit a sound pulse (ping) to generate an echo from the target.
2. Passive Sonobuoy that listen and acquire acoustic energy emitted by possible targets.
3. Special Purpose Sonobuoy that monitors measurements and information of the underwater environment such as the water temperature.

This research will focus on applications where combinations of active and passive sonobuoys are used, as will be illustrated later.
Sonobuoys are usually deployed from helicopters or airplanes (as shown in Figure 1.2 [7]), but they can also be deployed from surface ships. In order to determine the position of a submarine, it should be within the detection range of at least three passive sonobuoys (preferably four to achieve better accuracy). However, the bearing of the submarine can be detected if the submarine lies in the detection range of one sonobuoy that is equipped with a compass. The advantages of using passive sonobuoys over other acoustic measuring systems include its relative low cost, ease of deployment and the fact that they are not disturbed by the noise from the deploying platform. The main factors that affect the operation of sonobuoys are interference from surrounding sources and background noise [8].

Figure 1.1 Sonobuoy loaded on aircraft  Figure 1.2 Aircraft launched sonobuoy
Knowing the location of the sonobuoys is essential information especially in applications where the target positions are determined relative to the position of the sonobuoys. Present buoy geolocation schemes involve a variety of radio direction finding schemes to home on the radio frequency (RF) transmission of the sonobuoy. These methods require the ASW aircraft to maneuver to obtain a sonobuoy position, and also in most cases require the use of some form of directional antenna. These methods also mean the aircraft cannot stand-off at any distance from the buoy pattern without degrading localization accuracy [9].

Recently global positioning systems (GPS) were introduced to sonobuoys for accurate knowledge of their geographic locations. The Sonobuoys system employing GPS for positioning of deployed sonobuoys is known as “GPS Sonobuoys” [9]. Figure 1.3 shows a GPS sonobuoys system [10]. The floating part of the sonobuoy carries a GPS receiver which is used with another GPS receiver mounted on the platform (ship in this case) to apply differential GPS for the sonobuoy position. This will lead to highly accurate position of the sonobuoy and hence the detected source can be accurately localized.
1.1.3 Multistatic Active Sonobuoy fields

The number of sonobuoys used for tracking targets in the underwater environment might vary. Based on the number of sonobuoys used, sonar systems utilizing sonobuoys can be categorized as follows [11]:

1. Monostatic systems, where only one sonobuoy is used
2. Bistatic systems, where two sonobuoys are used.
3. Multistatic systems, where more than two sonobuoys are deployed.

Active sonar systems where multiple receivers are deployed in separate locations with at least one transmitter active sonobuoy are known as multistatic active sonar (MSA) systems. A basic diagram of a multistatic active sonar system is shown Figure 1.4.
Multistatic Active sonar involves actively searching an area for submerged targets. Obviously, utilizing multiple receivers can enhance the accuracy of target detections and increase the coverage area of active sonar pings [11].

1.2 Problem Statement

Utilizing multistatic active sonobuoy fields has more advantages over deploying monostatic systems where one sonobuoy is used as a transmitter and a receiver, and bistatic systems where only two sonobuoys are used where one is a transmitter and the other is a receiver. Some of these advantages are:

- Wider range of detection
- Higher probability of target detection
- Increased accuracy in case of multiple detections
• More efficient utilization of the high cost transmitter sonobuoy by deploying low cost receiver sonobuoys.

However, introducing multiple receivers in active sonar systems requires more complex methodologies to analyze the data acquired as its quantity increases. Analysis and development of algorithms to deal with the complexity of the multistatic system is drawing a lot of attention in order to exploit the full capabilities of the deployed system [12]. Providing a tracking system that can detect and track unknown number of targets in the harsh underwater environmental conditions is one of the main challenges in designing tracking systems for MSA fields. Moreover, the high rate of false contacts provided by the multistatic active sonobuoy fields is another difficulty that should be overcome by the tracking system [1, 12, 13, 14], while maintaining a realistic computational time taking into consideration the nature of application. Therefore, tracking targets in the MSA fields is an ongoing research.

The primary objective of this research is to suggest a tracking method that is capable of dealing with the data provided by a multistatic active sonobuoy system. The tracking system is to be able to efficiently detect an unknown number of targets appearing at different time instances of the tracking period. Moreover, the algorithm should be able to keep track of previous positions of the targets with reasonable accuracy. Reduced computational complexity and low time consumption are important issues that the tracking algorithm should properly solve in order to make the tracking algorithm useable for real-time application. If successful, this research could be extended to
produce an effective and fast tracking algorithm for other target-tracking systems. Detailed thesis objectives are described in the following sub-sections.

1.2.1 Multiple Target Tracking using Dynamic Programming

This research will examine the benefits of applying a tracking algorithm based on Dynamic Programming for multiple target tracking. Utilizing dynamic programming as the major block in the tracking algorithm along with other auxiliary artificial intelligence techniques can facilitate breaking down the multiple targets tracking problem to simpler hierarchical set of steps. It is hypothesized that by applying an efficient dynamic programming algorithm with sufficient data reduction, the tracking problem would require less complexity and an effective multi-target tracking system can be applied when multistatic active sonobuoy fields are deployed.

1.2.2 Experimental Validation using Simulation Data

The algorithm developed in this research will be tested using various simulations in order to analyze the performance of the tracking algorithm in different scenarios. A total of seven simulation data sets will be generated with different number of targets and different types of data provided. The simulation datasets increase in complexity at every testing phase in order to verify the ability of the tracking algorithm to deal with challenging scenarios and erroneous readings provided by the sensors.

1.2.3 Experimental Validation using Real Data

After assessing the performance of the tracking algorithm developed using simulation data, a data set collected from a multistatic active sonobuoy sea trial
(SEABAR07) is used to perform a final testing of the algorithm. The goal of this test is to validate the ability of the tracking algorithm to deal with real data collected using sonobuoys, and analyze the performance of the algorithm. Moreover, a comparison between the method developed in this thesis and other tracking methods is discussed.

1.3 Scope of Research

The research documented in this paper will cover the following:

1. Obtaining a general background on active sonar systems and target tracking techniques.

2. Reducing the amount of data provided by the sonobuoys using Self-Organizing Maps.

3. Developing a dynamic-programming algorithm to detect, track, and provide the position of multiple targets using the output data of the Self-Organizing Maps.

4. Utilizing the Hough transform for line detection to provide more accurate and robust tracks of possible targets.

5. Generating simulation data where the data provided represent targets’ locations as image pixels.

6. Generating simulation data where the data provided represent targets’ locations as range and bearing (contact-level data).

7. Generating acoustic level simulation raw-data where the data provided by the simulation represent the reflection of a ping of the surface of a target, where the ping is transmitted from an active sonobuoy and received at passive sonobuoys and
undergoes attenuation profiles as based on a module for normal mode generation which is a part of a simulation program developed at the Centre for Marine Science and Technology at Curtin University of Technology (Actup version 2.2)

8. Testing the tracking algorithm using the three types of simulation.

9. Testing the tracking algorithm using SEABAR07 active sonar sea trial, run A056 data set provided by NURC (NATO Undersea Research Centre).

10. Comparing to other methods to assess the merits and limitation of the tracking algorithm developed in this thesis.

1.4 Thesis Outline

Chapter 2 introduces the relevant background information required to gain an appreciation for the work conducted throughout this thesis. Active sonar systems are presented and discussed in details in order to provide understanding of the nature of the problem of underwater target tracking. Information regarding the path losses factors that affect detections of targets is provided. Moreover, Chapter 2 also includes a general overview of the problem of tracking multiple targets using multiple sensors. Current techniques for detecting a target, estimating their locations and tracking them are presented as well. Moreover, the problem of data association when multiple targets exist within a search area, in addition to a brief overview of common approaches to solve the data-association problem is provided.

Chapter 3 discusses the tracking method developed in this research. The first block of the tracking algorithm to be discussed in Chapter 3 is the data reduction using
Self Organizing Maps. The next block discussed is the major block in the tracking system which relies on the dynamic programming algorithm where the input data are the reduced data obtained by the Self organizing maps, and the output is the locations of possible targets and their previous tracks. The final step of the tracking algorithm is the Hough transform for line detection which serves as a method to provide more accurate tracking.

Chapter 4 presents the validation data to be used to test the tracking algorithm and analyze its performance. Simulation data is first used to examine the tracking algorithm. Three types of simulations are introduced in Chapter 4: Pixel-Target simulation, Contact Level simulation, and Acoustic Level simulation. Moreover, Chapter 4 discusses the data set obtained from the sea trial SEABAR07 and presents inherited errors in the readings acquired.

Chapter 5 discusses the results of applying the tracking algorithm developed in this research to the simulation and real data. Analysis of the performance of the tracker is discussed in details, including the effect of various parameters on the targets’ position error and the number of false alarms.

Chapter 6 summarizes the work completed and the conclusions made out of this research. It highlights the contributions of this thesis and provides recommendations for future work.
Chapter 2

Literature Review

2.1 Introduction

As the number of sensors increases in a tracking system, the complexity of the tracking system required increases in order to deal with the increased level of data-complexity. In multistatic active sonar systems, dealing with the complexity of the data requires a deep understanding of the nature of the system and the conditions that limit tracking in the underwater environment. Moreover, it is essential to obtain knowledge about tracking systems used for tracking multiple targets in multi-sensor systems. This chapter will provide an overview of active sonar systems and the different types of sonar systems. Factors that affect the performance of each sonar system as well as the limitations of each system will be discussed. In addition, this chapter will discuss the problem of target tracking and techniques used for tracking and estimating target positions. Moreover, an overview of the problem of tracking multiple targets and associating data for each target will be presented. It should be noted that, although the focus of this research is to provide a tracking algorithm for multiple targets in the Multistatic Active Sonar systems, the tracking algorithm can be applied for other sonar and radar tracking applications.

2.2 Active Sonar

In applications where active sonar is utilized, a pulse of acoustic energy is transmitted into the water by an acoustic projector. This projector can be installed on a
ship, another submarine, or a sonobuoy. The energy pulse (ping) propagates through the water. Portions of this ping get reflected when they hit the surface of a target, a clutter, the water surface or the sea ground. This reflected echo is the signal of interest at the receiver location. Range information is then determined by measuring the time interval between transmitted and received pulses [1]. Based on the number of devices used for transmission and receiving a ping, sonar systems can be categorized into monostatic, bistatic and multistatic active sonar systems.

2.2.1 Monostatic Active Sonar

A monostatic active sonar system is a system where the receiver and the transmitter are at the same location [1] as shown in Figure 2.1. When the ping is transmitted, propagation loss is experienced both as the pulse propagates through water to the target and when it reflects back as the echo travels toward the receiver hydrophone. Since the signal travels the same distance and path, transmission loss is equal in each direction. By determining the time delay between transmitting a ping and receiving its reflection, the target location can be expressed as the intersection of the bearing of the arrived signal and the circle circumference that represents the distance between the target and the sonar system. The process of localizing a target given the time delay between transmitting a ping and receiving its reflection is illustrated in Figure 2.2.
Figure 2.1 Monostatic Active Sonar system

Figure 2.2 Target localization in monostatic active sonar systems
Active sonar systems’ performance is affected by the nature of background interference. Depending on the dominant interfering background, the performance of a sonar system can be either noise-limited or reverberation-limited. In a noise-limited situation, the performance of the system is limited by the high level of ambient noise present at the frequency of the echo. In a reverberation-limited situation, the high level of reverberation present limits the performance of the system.

When the dominant background is ambient noise, the active sonar equation can be written as follows [1]

\[
SE = SL + TS - 2PL - AN - RD + DI
\]  

(2.1)

where

SE = signal excess (echo excess) (dB)
SL = source level (dB)
TS = target strength (dB)
PL = propagation loss (1 way) (dB)
AN = ambient noise (dB)
RD = recognition differential (dB)
DI = receiver directivity index (dB)

When the dominant background is reverberation, the active sonar equation may be written as follows

\[
SE = SL + TS - 2PL - RD - RL
\]  

(2.2)

where
SE = signal excess (echo excess) (dB)
SL = source level (dB)
TS = target strength (dB)
PL = propagation loss (1 way) (dB)
RD = recognition differential (dB)
RL = reverberation level (dB)

It can be noticed that the difference in the equations between the reverberation-limited case and the noise-limited case is replacing the ambient noise level and the receiver directivity index in the equations of the noise limited scenario with the reverberation level in the reverberation-limited scenario. The following is clarification of the terms used in the sonar equations [1].

- **Signal Excess (SE)**

  Signal Excess, or the Echo Excess, refers to the received signal level available for detection. The goal in an active sonar system is to detect the portion of the emitted acoustic energy pulse reflected from a target’s surface and propagating back to a sonar receiver.

- **Source Level (SL)**

  The source level is the level of the acoustic energy pulse (in dB) transmitted from the projector at a reference distance of 1 yard from the projector in the direction of the target.
• **Target Strength (TS)**

The target strength of a reflecting object (in dB) refers to the intensity of sound reflected by the target at a distance of 1 yard from its acoustic center in some direction. The value of TS depends on the size, shape, construction, type of material, roughness and aspect of the target, as well as the angle, frequency and waveform of the incident sound energy.

For submarines, the target strength measurements are highly correlated with the aspect, thus forming a “butterfly” pattern as shown in Figure 5-5 [1]. It can be noticed that the smallest values of target strength measurements of a submarine are found for bow and stern aspect, while the largest values are normally seen on beam aspect.

![Figure 2.3 Aspect variation of Submarine Target Strength](attachment:image.png)
• **Propagation Loss (PL)**

Propagation loss, or the transmission loss, is the attenuation in signal intensity (in dB) as it travels through the ocean medium from 1 yard of the acoustic center of the sound source to a point in the sea. In active sonar equations, the distance is between the sonar source and the target. Propagation loss is a complex function of several environmental conditions and signal parameters.

• **Ambient Noise (AN)**

Ambient noise is the steady-state level (in dB) of the total noise background existing at the receiving sensor, as measured by a nondirectional hydrophone. It is the level of the steady-state background noise of the sea, turbulences, surface noise and other sources of noise which tend to mask the desired signal.

• **Recognition Differential (RD)**

Recognition differential is the signal to noise ratio (in dB) required at the sonar processor to enable an operator to recognize the presence of a signal 50 percent of the time.

• **Directivity Index (DI)**

The receiver directivity index is a measure (in dB) by which an array, through its beam pattern and characteristics, is able to filter out isotropic noise.
• **Reverberation Level (RL)**

It is a measure of the level of reverberation (in dB) present at the receiver output terminals. Reverberation refers to the sum of unwanted scattered projected energy pulse to the receiver, and therefore it differs from noise in that the sonar system causes it by itself.

While the level of ambient noise remains the same with respect to the level of projector source level, the reverberation intensity increases as the source level increases. Moreover, the reverberation intensity decreases with increasing range of the reflective surface while the level of ambient noise remains unaffected as can shown in Figure 2.4. Therefore, an active sonar system can either be reverberation limited at short ranges, with the target masked by reverberation, or it can be noise limited because the target is too far away for enough energy to get reflected from its surface and produce a detectable echo above the level of ambient noise [1].
While both of the echo level and reverberation level decreases with range, the echo level decreases more rapidly as a function of range than the reverberation level. The range at which the reverberation level is equal to the echo level is known as the reverberation-limited range. Similarly, the range at which ambient noise level is equal to the echo level is the noise-limited range as illustrated in Figure 2.4. The shorter of these two ranges determines the limitation of the active sonar system.

Due to several factors and changes in the surrounding environment, the level of ambient noise and reverberation might vary and thus the system limitation might alter between being reverberation limited and noise limited as shown in Figure 2.5. When low level of ambient noise is present, the performance of the sonar system is limited by the
reverberation level. Therefore, the system is effective when the target is within a maximum of the reverberation-limited range, before the echo level falls below the reverberation level. When high level of ambient noise is present, the performance of the system becomes limited by the high ambient noise level at the frequency of the echo. Therefore, the system becomes effective when the target is within a maximum of the noise-limited range before the echo level falls below the ambient noise level [1].

Figure 2.5 Range Limiting Variations
2.2.2 Bistatic Active Sonar

It is a common practice in sonar systems to use different devices placed at different locations as sonar transmitter and receiver [1], as shown in Figure 2.6. One advantage of deploying bistatic active sonar systems is expanding the range of detection by decreasing the total distance a reflected ping propagates.

![Bistatic Active Sonar system](image)

Figure 2.6 Bistatic Active Sonar system

Introducing the adjustment of having the sonar receiver at a different geographical location of the transmitter requires modifying the active sonar equations as follows [1]:

23
Noise-Limited scenario:

\[ SE = SL + TS - (PL1 + PL2) - RD - AN + DI \] (2.3)

Reverberation-Limited scenario:

\[ SE = SL + TS - (PL1 + PL2) - RD - RL \] (2.4)

It can be noticed that in a bistatic active sonar system, the propagation loss is no longer identical between the source-target and the target-receiver paths. Since the propagation loss in each path is different, both propagation losses must be added to obtain the total propagation loss affecting the system. The term (2PL) used in the monostatic active sonar equations is replaced with (PL1 + PL2), where PL1 represents the propagation loss between the source and the target and PL2 represents the propagation loss between the target and the receiver.

Moreover, the target strength can no longer be generally described in terms of the target aspect as shown previously in Figure 2.9. The target strength in the bistatic active sonar scenario will depend on the geometry of the source-receiver locations. The bistatic theorem states that for large smooth objects, the target strength is equal to the monostatic target strength taken at the bisector of the “bistatic angle” between the direction to the source and the receiver [1]. In other words, the target strength in a bistatic scenario where source (S), target (O) and receiver (R) form a bistatic angle (SOR) is the same as the monostatic target strength where the source/receiver device (P) forms a direction (OP) with the target as illustrated in Figure 2.7.
2.2.3 Multistatic Active Sonar

As discussed in Chapter 1, section 1.3, active sonar systems that employ multiple receivers located in separate locations are known as multistatic active sonar systems. Each pairing of a particular source and receiver has its own applicable sonar equation as in the bistatic case described earlier. In the Multistatic scenario, sound is emitted from a source and received at time $T_R$ at one of the sensors, as illustrated in Figure 2.8. The emission of sound and subsequent processing for an echo is referred to as an event [12]. The time interval separating events depends on both the size of the search area and the ocean environment.

Figure 2.7 The bistatic geometry
Due to the geometry of each source-receiver combination in the MSA scenario, the time information related to a ping reflection of the surface of a target results into obtaining ellipsoidal range information of the target location. This ellipse, shown in Figure 2.8, is referred to as *equal-time of arrival (ETA) ellipse*. As in any mathematical ellipse, the oval shape is formed about two points, called *foci*. For the MSA scenario, the foci are the source and the receiver that performed the transmission and detection of the ping. The ping that arrives directly from the active sonar source at the receiver is referred to as *direct arrival*, which is usually much higher in intensity than other reflections of the same ping. All the points on an ellipse have an equal total distance from one focus point,
to that point on the ellipse, to the other focus point. In the multistatic scenario, pings’
reflections arriving at the same time represent targets that fall on the same ellipse [15], as
shown in Figure 2.8.

The concept of the specular reflection angle is critical to the success of an MSA
scenario. In a specular reflection, the direction of incoming ping, and the direction of the
reflected ping from the surface of a target make the same angle with respect to the surface normal
[1, 16]. This concept is illustrated in Figure 2.9. MSA systems rely on having a large number
of sensors distributed over an area in a pattern that maximizes the probability that one or
more sensors will be positioned in the echo’s beam-pattern. In an MSA scenario, non-
specular reflections will unlikely exceed the Target-Strength level required to reach the
detection threshold. However, acquiring a specular reflection of a target doesn’t mean
that the reflection is immune against reverberation masking and noise limitations that
might adversely affect the detection [1].
One important characteristic of the specular reflection in MSA scenarios is that the Doppler shift affecting the ping reflected from a target at the specular reflection angle will be equal to zero [16]. Moreover, the target’s line of travel will be nearly tangential to the ellipse from the strongest detection. Therefore, the line of travel can be determined by determining the tangent line to the ellipse for the strongest detection from a single ping.

When multiple detections of the same target are available via multiple sensors in an MSA system, the ellipses obtained from the measurements acquired can be used to obtain an accurate location of the target by using the crossings of those ellipses. As with a single directional sonobuoy, a contact could be localized with a single detection by using...
the single ellipse and the associated bearing line. However, this could lead to an unacceptably high area of uncertainty (AOU) due to bearing inaccuracies and the size of the ellipse. The crossing of multiple ellipses is considered to be a more accurate localization technique when compared to the individual bearing-range information obtained at each individual sonobuoy. When more than one crossing point is detected in the ellipses-crossing approach, the bearing acquired from the directional sonobuoys can be used to roughly point towards the correct intersection [15]. Figure 2.10 illustrates the methodology of using ellipses intersection point to determine the location of a target.

![Figure 2.10 Target localization in MSA sonobuoy fields](image)

Figure 2.10 Target localization in MSA sonobuoy fields

Although the ellipses-crossing approach is capable of providing high accuracy measurements of the target location, it is not considered to be a very practical approach.
Active sonar systems are known to have a high rate of false detections, and utilizing all the data available to obtain ellipses-crossings for a large number of sensors is computationally very expensive. Moreover, to obtain tight ellipse convergence, the locations of the source and the detecting sensors must be accurate, and the processor must be able to compensate for errors in sonobuoy placement or relative location drift [15].

There are several sources of error that contribute to inaccuracy in determining the target location in MSA scenarios. In addition to the noise, reverberation and multipath reflections that result into acquiring false detections, errors in detections of pings reflected from targets can occur. Inaccuracy in determining the time of arrival of a reflected ping will affect range computations of the target location. Moreover, inaccuracy in bearing estimation of the arrived signal at the receiver sonobuoy will result into inaccuracy in determining the target location on the ETA ellipse as illustrated in Figure 2.11 [15].
Additionally, errors in determining the location of the source or the receiver might alter the shape of the ellipse and the range information resulting into another type of inaccuracy as illustrated in Figure 2.12.
Along with previous sources of errors and data complexity, an MSA system reveals its presence as an active sonar system by emitting the acoustic energy pulses. Moreover, since the detected energy pulses are not emitted by the targets themselves, no information is available about the nature of targets.

However, interest in MSA systems is increasing as they provide relatively good detection in extreme environmental conditions when compared to other passive and active sonar systems. Moreover, with the advancing technology used in submarines industry, targets are becoming more silent and harder to detect, and active sonar provides a detection mechanism for stealthy targets.
2.3 Target Tracking

The term “Target Tracking” refers to the process in which the task of estimating the state of a target both at the current time (filtering) and at any point in the future (prediction) is executed. In any target tracking system, the user is interested in a descriptive output indicating the current target location, direction of movement, next possible location, speed of movement and possibly a measurement of the target acceleration. Moreover, the tracking system should provide some measure of the accuracy of these state estimates. The state estimation is computed under two types of uncertainty: target model uncertainty and measurement uncertainty. Target model uncertainty exists as a result of the fact that most of the targets of interest are typically noncooperating targets (i.e. the targets do not follow a predefined trajectory and their behavior cannot be modeled with absolute confidence). Nevertheless, target tracking systems require one or more models of motion for the targets of interest. Therefore, when the motion of a target deviates from the predefined motion model, significant errors can occur in estimating the target trajectory. The second type of uncertainty, measurement uncertainty, exists because of the inherited inaccuracy of the measurement devices and sensors and the noise existing in the measurements fed to the tracking algorithm. As a consequence of these two sources of inaccuracy, errors can be made in associating measurements to existing tracks, and this type of error can result in rather significant errors in estimating the state of a target [17].

There are many algorithms for target tracking (TT) which vary in the level of their complexity [18, 19]. However, all target tracking algorithms use two main types of data
processing: data association and track evaluation. A typical tracking process consists of acquiring repetitive sets of measurements (e.g. scans) from some sensors (e.g. radars), which produce measurement data (usually called observations), originating from unknown quantity of real objects (targets) or clutters (false alarms). The second step is associating the observations acquired to predefined set of trajectories (tracks) and some probability measures of these tracks are evaluated. The aim of this process is to preserve a set of tracks that effectively resemble with an acceptable level of accuracy the real set of targets [20].

2.3.1 Problem Formulation

In general, the information representing a target as its location is defined by a sequence of states $X_k: k = 1, 2, ..., M$. The change in the target state over time can be expressed by the following equation [20]:

$$X_k = f_k(X_{k-1}) + W_k$$  \hspace{1cm} (2.5)

where $W_k: k = 1, 2, ..., M$ is the noise vector. The relationship between the measurement and the state is specified by the measurement equation as follows

$$Z_k = h_k(X_k, N_k)$$  \hspace{1cm} (2.6)

where $N_k$ is a noise vector that is independent of $W_k$. The aim of a tracking algorithm is to estimate the state $X_k$ given all the measurements up to the current moment of computing the state estimation, or equivalently to construct the probability density function $p(X_k|Z_{1,...,k})$. The recursive Bayesian filter provides a theoretically optimal solution which breaks the tracking problems into two parts: Prediction and Correction. In
the prediction stage, prior probability density function (PDF) of the current state $p(X_k|Z_{1,...,k-1})$ is computed using equation 2.5 and the PDF of the state $k-1$ available from the previous iteration. In the correction step, the likelihood function $p(Z_k|X_k)$ is used to determine the posterior PDF $p(X_k|Z_{1,...,k})$. When the measurements available through the sensors correspond to only one target, the state estimation of the target can be simply found as illustrated. However, when the measurements correspond to multiple objects, an algorithm to associate the measurements with each object is required. [21].

2.3.2 Single-Target state estimation filters:

Single-Target tracking filters are capable of estimating the target state when the measurements represent one target only. For most single-target tracking problems, the Kalman filters or the particle filter is used depending on the nature of the dynamic equation (i.e. linear or nonlinear).

2.3.2.1 Kalman Filter

Kalman filtering is an optimal estimation tool that provides a sequential recursive algorithm for the optimal least mean variance estimation of the system states [22, 23, 24]. The Kalman filter is capable of providing a real-time statistical estimation of the accuracy of the system states and thus providing a very useful quantitative error analysis tool in addition to its functionality as an optimal estimator [25]. The filter generates its own error analysis with the computation of the error covariance matrix, which gives an indication of the estimation accuracy.
The KF uses a form of feedback control by which the filter estimates the process states at some time and then obtains feedback in the form of noisy measurements. Therefore, the equations for the Kalman Filter (KF) fall into two groups: prediction and update. In the prediction phases, the KF uses the state estimate from previous iteration to provide an estimate of the current system state. In the update stage, the Kalman filter uses the measurements available at the current iteration to provide a more accurate and refined current state estimate [26].

The basic Kalman filter is limited to a linear assumption for the state transition and observation models. However, when the functions $f_k$ or $h_k$ are nonlinear, other variations of the Kalman filter are used such as the Extended Kalman filter and the Unscented Kalman filter.

### 2.3.2.2 Extended Kalman filter

The Extended Kalman filter is a variation of the Kalman filter where the state transition and observation functions are not necessarily linear but rather differentiable functions. However, $f$ and $h$ cannot be applied to the covariance directly. Therefore, $f$ and $h$ are linearized using the Taylor series expansion. A matrix of partial derivatives (the Jacobian) is computed, and at each iteration, this matrix is evaluated using the current predicted states. For this approximation to be valid, this linearization should be a good approximation of the non-linear model in the entire uncertainty domain associated with the state estimate [27].
2.3.2.3 Unscented Kalman filter

The Unscented Kalman filter (UKF) is an attractive methodology to apply for target tracking when the state transition and observation models show high level of nonlinearity. While the Extended Kalman filter can perform poorly in such scenarios because the mean and covariance are propagated through linearization of the underlying nonlinear model, the UKF gives particularly good performance. The UKF uses an unscented transformation to perform state estimation of nonlinear systems without linearizing the system and measurement models by applying a deterministic sampling used to pick a minimal set of sample points (sigma points) around the mean. The system function is applied to each sample, which results in a group of transformed points. The mean and covariance of this group of points are the propagated mean and covariance. Since there is no linearization involved in the propagation of the mean and covariance, the Jacobians of the system and measurement model do not have to be calculated [28].

2.3.2.4 Particle Filter

Particle filters, also known as Sequential Monte Carlo methods (SMC), are very common alternative for the Kalman filter, Extended Kalman filter and the Unscented Kalman filter when very high nonlinearities exist in the state and observation models. Particle filter has been a successful numerical approximation technique for Bayesian sequential estimation. Starting with a weighted set of samples approximately distributed according to $p(X_{k-1}|Z_{k-1})$, new samples are generated from a suitably designed proposal
distribution, which may depend on the old state and new measurements. From time to time, it is necessary to resample the particles to avoid degeneracy of the importance weights. The resample procedure essentially multiplies particles with high importance weights, and discards those with low importance weights. With sufficient samples, the performance of particle filters approaches the Bayesian optimal estimate; therefore the particle filter can outperform the EKF or UKF. The approaches can also be combined by using a version of the Kalman filter as a proposal distribution for the particle filter [21, 29].

Although the particle filter performs well with highly nonlinear models where the Kalman filters would usually fail, a major disadvantage of the Particle filter is the heavy computational nature of the algorithm which makes it time consuming and very demanding from a hardware aspect [21].

2.3.3 Multi target tracking and data association

The term multiple-target tracking (or MTT) refers to the process of formulating the trajectories of a generally unknown number of moving objects from a set of space-dependent or time-dependent signals that they trigger in a common detection region [30]. Several applications in real-life apply multiple-target tracking process as surveillance systems that typically involve using large arrays of sensors interfaced with some extensive data-processing computational infrastructure. The process of reconstructing the tracks multiple targets using sets of measurements can be broken down to the following steps [21, 30]:

a. Reconstruction of coordinate points (or observations) from raw detection signals.

b. Association of the observations into tracks (commonly referred to as pattern recognition).

c. The estimation of the track parameters.

d. Consistency checks on reconstructed trajectories.

The first step in the process described earlier is usually performed by applying various signal processing techniques. The second step requires techniques drawn from control systems theory, artificial intelligence, and global optimization to associate the data with the correct tracks. The third step requires evaluating the track parameters by means of static or dynamic statistical models, and the last step is generally based on application-dependant criteria. External constraints such as real-time requirements, may apply to some (or all) of the process steps especially when time consumption is critical as in most of the military application of target tracking [30].

The observation data acquired from a tracking system may describe kinematic properties of the targets’ motion such as positions, velocity, or acceleration. Additionally, the observation data may include more information such a signal magnitude, which helps in determining other attributes such as the target size, and the level of confidence associated with an observation. The existence of at most one true measurement acquired from each sensor is generally assumed without loss of generality since observations from the same source may easily be combined in the data reconstruction step. Existence of
false contacts and spurious observations in the data acquired from the sensors is very common in most tracking environments due to several factors as the existence of clutters [17, 30].

When tracking multiple targets using Kalman filters or particle filters, the main challenge is to associate the most likely measurements for each target's state. A trivial approach is to perform a nearest-neighbor approach to assign the measurements to each target. However, this approach would fail if the targets are relatively close to each other. Incorrect data association would lead the filter to fail in the objective of tracking [17]. Several approaches were introduced to solve the problem of data association. In the following sections, three of the most common approaches to deal with the data association task are briefly discussed.

2.3.3.1 Joint Probability Data Association Filter (JPDAF)

JPDAF is one of the early statistical approaches to solve the data association dilemma in multiple-target tracking problems. The JPDAF enforces a kind of exclusion principle that prevents two or more trackers from latching onto the same target by calculating target-measurement association probabilities jointly [31]. A key notion in the JPDAF is that of a joint event $\Theta$ or conjunction of association events $\Theta_{t_j}$ (where the subscript $t_j$ denotes which target measurement $j$ is matched to). The probability of a particular $\Theta$ depends on the distances between each target's predicted measurement and the actual measurement it is associated with in $\Theta$. 
The major limitation of JPDAF is its inability to cope with new targets appearing within the field of view (FOV) or already tracked objects exiting the FOV since it assumes a fixed number of targets [31]. Therefore, a change in the number of targets might lead to significant tracking errors. This makes this method impractical for ASW applications.

2.3.3.2 Multiple Hypothesis Tracking (MHT)

To avoid the problem of data association to incorrect targets, motion correspondence decision in the MHT is delayed until several frames of measurements are examined. Figure 2.13 shows the basic block diagram of an MHT tracking algorithm. The MHT algorithm preserves several correspondence hypotheses for each target at each time frame. The final tracking output produced by the MHT algorithm consists of the most likely set of correspondences over the time period of its observation. An important advantage of the MHT over the JPDAF is its ability to create new tracks for objects entering the FOV and to stop tracking targets exiting the FOV. Moreover, the MHT algorithm is capable of handling the continuation of a track even if some measurements from an object are missing. Furthermore, unlike the JPDAF, a measurement may not be assigned to an object because the object may have exited the FOV, or a measurement corresponding to an object may not be obtained. The latter happens because either the object is occluded or it is not detected due to noise [21, 31].

The MHT makes associations in a deterministic sense and exhaustively enumerates all possible associations. A probabilistic MHT (PMHT) introduced by Streit
and Luginbuhl in 1994 [32] reduces the computational complexity by assuming the associations are statistically independent random variables and thus there is no requirement for exhaustive enumeration of associations. However, even with the recent developments on MHT, the MHT algorithm is still considered one of the most complex tracking algorithms that is computationally exponential both in memory and time. The Distributed MHT filter (DMHT), a product of NURC, is claimed to provide a robust, high performance and computationally efficient for tracking multiple target in comparison with other MHT approaches [33].

![Figure 2.13 Block Diagram of track-oriented MHT algorithm](image-url)

*Figure 2.13 Block Diagram of track-oriented MHT algorithm*
2.3.3.3 Probability Hypothesis Density (PHD) Tracking

Multiple-Target detection and tracking using PHD filters is relatively a new approach that is being applied to various real-time tracking applications. The PHD filter is based on a multitarget first-moment approximation. Instead of the full multitarget probability distribution $f_{k|k}(X|Z(k))$, the PHD propagates a multitarget statistical first moment: the PHD $D_{k|k}(X|Z(k))$. The PHD filter has computational complexity $O(m \cdot n)$ where $m$ is the current number of measurements and $n$ is the current number of detected targets [34].

The PHD filter provides a less computationally expensive approach to the problem of optimal multiple-target tracking than the MHT. However, PHD filtering requires evaluation of multiple integrals that have no closed form in general, which results into making the PHD filter a very complex and difficult approach. Recently, approximating of the PHD recursion using a Sequential Monte Carlo (SMC) method has been proposed [35]. The PHD filter provides only estimations of the states of targets that are present within the FOV at any time frame $k$. However, the PHD filter does not keep records of the target identities. Therefore, the PHD filter by itself is not capable of producing tracks for the targets detected. However some data-association functionalities of MHT can be incorporated with the PHD filter to produce the desired tracks. Moreover, other variations of the PHD such as the Cardinalized PHD (CPHD) provide more accurate estimates of target number than the probability hypothesis density (PHD) filter. Therefore, the CPHD is capable of providing better estimation of the states of targets. However, the CPHD is
mathematically much more expensive compared with the PHD filter, where the CPHD mathematical complexity is \( O(m^3 \cdot n) \) where \( n \) is the number of targets and \( m \) is the cardinality of measurement set at each time index. Techniques in literature reduce the computation complexity of CPHD filters with preserving track continuity under linear/Gaussian conditions such as the Gaussian Mixture CPHD (GM-CPHD) filter [36]. The PHD provides promising results in comparison with MHT and JPDAF algorithms, with the tradeoffs between accuracy of measurement and computational complexity still holding.

2.3.4 Dynamic Programming in Multi-Target tracking

Dynamic programming approach in multi-target tracking could be more explicitly called a solution strategy rather than an algorithm, and employs a staged optimization policy for solution development, and the specifics vary considerably in any given application [30]. DP breaks down the problem into phases or subsets of the problem space (e.g., a set of arcs in a large graph network). In each stage of the problem, several states associated are determined. These states refer to the values of the variables of interest, and decisions are made at each stage that are dependent only on the state within the stage (i.e., the processes are assumed memory-less). The decision made at each stage (e.g., the assignments for one scan of data) requires computing a return (cost or value). The return is the current optima solution to the local or stage-wise problem. When the return is determined, the solution moves to the next stage. DP solutions thus involve recursive equations for the returns between stages. A DP algorithm can be designed as
required to have forward recursions and backward recursions, depending on the nature of
the problem targeted and the decision of the analyst. [30, 37].
Chapter 3

Methodology

In this chapter, a detailed description of the tracking system methodology will be provided. The method used for target tracking in this research is a multistage method that can be divided into three phases: Data discretization, dynamic programming, and Line detection using Hough transform.

In the first stage, the system simplifies the data and reduces its amount by using Self Organizing Maps as a data discretization method that is followed by a clustering step. The output of the first stage is a binary image frame that is fed to the second stage of the system. In the second stage, the dynamic programming algorithm assigns a score to each pixel representing the similarity between a target behavior and each pixel’s pattern of movement throughout the previous set of frames. Moreover, the dynamic programming keeps a track of previous positions of pixels representing targets’ locations. The third and final step in the tracking system applies the Hough transform to detect possible straight lines of movements which provides better position estimation. In the following section, detailed description of the three main stages of the tracking methodology is presented.

3.1 Data Discretization using Self Organizing Maps

Kohonen Self Organizing Map (SOM) is a sheet-like type of artificial neural networks that is frequently used in applications where reducing data dimensionality
while learning and preserving possible patterns is required [40]. A SOM consists of a number of nodes where each node is associated with a weight vector of the same dimension as the input data vectors and a certain position among the other nodes. The nodes of a SOM can be tuned through a learning process to get activated to specific patterns or inputs. The easy visualization of data obtained by a 2-dimensional SOM in addition to the reduction of data complexity is utilized in the tracking method proposed in this research [40, 41].

SOMs are usually trained using pre-collected datasets so that the ordering of the nodes will map the data distribution [40]. However, the interest here is to simplify data representation through the SOM nodes rather than using the data points. Therefore, a static version of SOMs is used to discretize the readings acquired by sonobuoys. The weights associated with each node are set so that each reading from the sonobuoys will activate the closest node as shown in Figure 3.1. By inputting all the x-y co-ordinates of the data points acquired from the sonobuoys to the SOM, an array containing the indices of all nodes activated can be formed. The number of nodes of the SOM should balance the accuracy of the algorithm’s output, the size of the area to be searched for possible targets and the computation complexity level in addition to other factors to be discussed later. The data discretized using a SOM can be represented in a binary image format where the sonobouys readings are shown as pixels referring to the nodes activated. Figure 3.2 shows the representation of the received signals in an image format using a SOM.
Figure 3.1 SOM Node Activation by sonobuoys readings
In order to cover large areas with a reliable accuracy and relatively low amount of computation, it is beneficial to use a hierarchal set of SOMs. The first layer of the SOM consists of few nodes that completely cover the search area but with very poor resolution. The second layer is a high resolution map that covers only the area represented by one node in the first SOM layer. For example, to cover an area of 10 Km$^2$ with a resolution of 200 m$^2$ per node, one high resolution SOM of size 50×50 nodes can be used. Alternatively, a low resolution SOM 4×4 can be used to divide the search area to 4 quarters each of a size of 5 Km$^2$. Each node of the low resolution SOM will be mapped.
to a high resolution SOM of size 25×25 nodes. Using such hierarchal sets of SOM can reduce the number of calculations needed to discretize large search areas. Figure 3.3 illustrates using hierarchal SOMs to discretize two-dimensional data points.

![Hierarchical SOMs Diagram](image.png)

**Figure 3.3 Hierarchal SOMs for discretizing two-dimensional data points**

Due to the various sources of errors reflected in the sonobuoys readings as well as the geometry of the path pursued by the target, several adjacent nodes could be activated rather than one node. It is assumed that connected pixels are activated by readings referring to the same target. Therefore, connected components in the binary images need to be clustered and labeled. A simple union-find algorithm [42] is applied to find
connected components of the binary image as shown in Figure 3.4. Each connected cluster of pixels is labeled, and a trimmed-mean of the data points that activates each cluster is computed. A final binary image version is constructed using the nodes activated by these averages.

![Figure 3.4 Labelling connected clusters](image)

**Figure 3.4 Labelling connected clusters**

### 3.2 Dynamic Programming

In [38], the authors suggest a track-before-detect Dynamic Programming (DP) algorithm based on the Viterbi algorithm [43] in addition to a methodology to adjust the algorithm’s parameters to optimize its performance. The algorithm is a further development on the dynamic programming technique developed by Barniv [44, 45]. The
technique was developed to track dim, punctiform targets in a sequence of infrared (IR) images, thus it needed to be modified to fit the application of MSA sonar tracking. The binary image (constructed in the discretization phase using SOMs and then reassembled after clustering connected pixels) is the basis of the data model (a first-order hidden Markov model). The sequence of binary images formed after each ping is the observed data, while the target track is the hidden sequence of events [38, 39].

The algorithm assigns a score to each pixel in the current binary image indicating the similarity between a target’s behavior and the pixel pattern of movement throughout the previous sequence of data sets. Additionally a track of each pixel in the previous datasets is calculated and updated after each ping. The methodology to compute the score matrix and track matrix as proposed by Nichtern et al. [38] is illustrated and adjusted to fit the MSA scenario in Sections 3.2.1 to 3.2.4, and an approach to determine an appropriate threshold for the target alarm is proposed in Section 3.2.5.

3.2.1 Score matrix

The score matrix for frame $n$ updates the pixels of value 1 with an additional score. To compute the score for a pixel $k$ in frame $n$ the following variables should be determined:

a. The source pixel in frame $n-1$

b. The score of source pixel in frame $n-1$

c. The direction of source pixel in frame $n-1$
The last two parameters should be available in frame $n$ once the source pixel has been determined as they are computed in frame $n-1$. To find the source pixel, a search area has to be specified. For the MSA sonar scenario, the size of the search area will depend on the following criteria:

a. The size of area covered by each node in the SOM

b. The maximum speed to be tracked

c. The time gap between two consecutive pings

In this research, the following assumptions are taken into consideration: an area of 200 m$^2$ covered by each node in the SOM, a target speed ranging up to 5 m/sec (9.7 knots), a ping interval of 1 minute and an error of ±150 m. Since a target might move up to 450 m (4 pixels) between frame $n$ and frame $n-1$, a search area of 9 x 9 pixels will be needed, with the pixel $k$ under analysis in the center of the matrix as shown in Figure 3.5. It is worth mentioning that the above numbers are very close to reality and represent typical MSA tracking problems.
Finding the source pixel requires examining all the possible scores that the pixel under study can have using the scores in the search area of the score matrix in frame $n-1$. The pixel in frame $n-1$ which assigns the highest score to the pixel $k$ studied in frame $n$ is considered the source pixel. A technique to compute the score of a pixel $(x + i, y + j)$ introduced by Nichtern et al. [38] as follows:

$$\text{Source Pixel Score}(x + i, y + j) = SF_{n-1}(x + i, y + j).Var \quad (3.1)$$

where $(x, y)$ are the indices of pixel $k$ in frame $n$ while $i$ and $j$ vary to cover the search area, $SF_{n-1}$ is the score matrix for frame $n-1$, and $Var$ is a parameter that reflects the effect of change in direction on choosing the source pixel. $Var$ should have a maximum value when the direction computed between pixel $k$ and possible source pixel $k_s$ in frame...
$n-1$ is identical to the saved direction of pixel $k_s$ in the previous frame but $Var$ shouldn’t be so heavily weighted that it would emphasize directional consistency on the cost of the source pixel score. An appropriate parameter $Var$ for a certain pixel $k$ and a possible source pixel $k_s$ can be determined by

$$Var = (1 - p) \cdot \frac{|\text{Direc}_k - \text{Direc}_{k_s}|}{180^\circ} + p; 0 \leq p \leq 1$$  \hspace{1cm} (3.2)$$

where $\text{Direc}_k$ and $\text{Direc}_{k_s}$ are the directions of movement for pixel $k$ and pixel $k_s$ in degrees. If $\text{Direc}_k$ or $\text{Direc}_{k_s}$ cannot be determined, an appropriate assumption of $\text{Direc}_k - \text{Direc}_{k_s}$ can be assumed to be $90^\circ$. The value of $p$, the maneuvering tolerance factor, determines the level of maneuvering to be tolerated by focusing the decision making of choosing the score pixel on the score of the pixel $k_s$ in frame $n-1$ when choosing a small $p$. Conversely, when choosing a small value of $p$, emphasis will be on directional consistency and thus less maneuvering would be tolerated.

After determining the source pixel $k_s$ in frame $n-1$ for pixel $k$ in frame $n$, a score can be computed for pixel $k$ using the following equation, as proposed by Nichtern et al. [38]:

$$Score_n(x_k, y_k) = 1 + b \cdot Score_{n-1}(x_{k_s}, y_{k_s}) \cdot W(x_{k_s} - x_k, y_{k_s} - y_k)$$  \hspace{1cm} (3.3)$$

where $b$ is the effective memory coefficient (EMC) which regulates the propagation of scores throughout the sequence of frames and $W$ is the penalty matrix which will be discussed in Section 3.2.3.
3.2.2 Direction computation

Determining the direction of movement of a pixel $k$ from its source $k_s$ is of great importance to compute the parameter $Var$ and the correct penalty matrix as will be discussed later. To simplify the computations, directions of movements are rounded to the nearest $45^\circ$, so that only the first eight directions shown in Figure 3.6 are considered. Direction 9 is reserved for an indeterminable direction of movement, such as the scenario where the target is not moving or moving slowly, or the source pixel in the previous frame cannot be identified.

![Figure 3.6 Directions used in DP tracking](image)

Figure 3.6 Directions used in DP tracking
Clearly, no directions can be determined after transmitting the first ping as no previous data is available. Therefore all pixels for the first frame with value 1 are given direction 9. Direction 9 is also used when the search area for a pixel in frame $n$ is empty. When receiving a new frame, the direction of movement of each pixel is determined by rounding up the angle between the pixel and its source pixel in the previous frames to one of the 9 directions. These directions are saved in a Direction matrix which is propagated to the successive frame.

3.2.3 Probability matrix

The probability matrix $W$ is a set of weights that regulate the propagation of the score from a pixel in frame $n-1$ to frame $n$. A strong deviation in the direction of a pixel in frame $n$ from the direction of its source in frame $n-1$ will cause the probability matrix to attenuate the score propagated to the pixel in frame $n$ relative to the severity of the change in direction.

The number of penalty matrices is equal to the number of directions to be considered since a unique penalty matrix is associated with each direction as shown in Figure 3.7.
The penalty matrix to be used to propagate the score for pixel $k$ in frame $n$ will be chosen depending on the direction of the source pixel $k_s$ in frame $n-1$. Nichtern et al. [38] suggest using a penalty matrix composed of a weighted sum of the penalty matrix associated with the direction of the source pixel and the one associated with direction 9,

$$W = (1 - p) \cdot W_{\text{direct}k_s} + p \cdot W_{\text{direct}9}$$  \hspace{1cm} (3.4)

where $p$ is the maneuvering tolerance factor as shown in Section 2.2.1. Choosing a small $p$ would correspond to a reduced maneuvering tolerance for a possible target, thus raising the probability of losing track of the target. Assigning a high value of $p$ would result in
higher probability of false alarm as more noise and more false detections will be tolerated.

In this research, a technique to create the penalty matrices is introduced. For directions 1 to 8, equations (5) to (7) can be used to find the components of the penalty matrix for direction direc and size m×m:

\[
w_{direc_{k,l}} = 1 + \cos\varphi_{k,l}
\]

(3.5)

\[
\varphi_{k,l} = \theta_{direc} - \arctan\left(\frac{-k + m + 1}{-l + m + 1}\right)
\]

(3.6)

\[
W_{direc} = W_{direc} \cdot \frac{1}{\sum_{k,l} w_{direc_{k,l}}}
\]

(3.7)

For Direction 9, a simple function that decreases the weight as the distance of the pixel from the centre of the matrix increases can be used as long as the weights are normalized.

3.2.4 Tracking matrix

The tracking matrix is updated after processing each ping. After finding the sources for the pixels in binary frame n, the tracking matrix n will contain, at each location of a pixel of value 1, the sequence of sources during the entire series of frames processed.
This version of the tracking matrix will consume more storage space than the version proposed by Nichtern et al. [38] since the tracking matrix saved for each frame in this scenario will contain the whole track for each pixel if available. Computationally, however, since the length of the track will be used later as one of the threshold parameters, it will be easier to have the whole track as it exists after processing each frame. Moreover, future work on connecting broken tracks will be facilitated if the complete possible track at any frame $n$ is available directly rather than tracing it back.

### 3.2.5 Target alarm threshold

A preliminary step to determine an appropriate alarm threshold is computing the maximum score value a pixel can acquire when it represents a moving target under ideal conditions. The score propagating through a sequence of frames from frame 1 through frame $n$ can be modeled using the series

$$Score_n = 1 + \sum_{i=1}^{n} \prod_{k=1}^{i} (b \cdot w_k)$$

(3.8)

where $w_k$ is the penalty weight at frame $k$ and $b$ is the effective memory coefficient as stated in Section 3.2.1. If the target is moving with a consistent velocity, this would result in having a similar penalty weight at each frame $k$. In this scenario, the previous series can be simplified to

$$Score_n = 1 + \sum_{i=1}^{n} (b \cdot w)^i$$

(3.9)
This illustrates the importance of choosing \( b \cdot w < 1 \) in order for the series to converge to:

\[
Score_n = 1 + \frac{1}{1 - b \cdot w}
\]  

(3.10)

In more realistic scenarios the score of a pixel representing a moving target would range only to part of the score found by equation 3.10. Hence, the operator will have to choose a percentage \( d \) of the value obtained by 3.10 to be a first parameter in the alarm threshold. This percentage will control the false positive rate and false negative rate of target detection. Another useful parameter to include in the decision making of target detection is the length of track, as it will reduce false detections when choosing a small \( d \). The target detection process can be shown as:

\[
Target_n(x, y) = \begin{cases} 
1, & \text{if } Score_n(x, y) > d \cdot \left( 1 + \frac{1}{1 - b \cdot w_{max}} \right), \\
0, & \text{otherwise}
\end{cases} 
\]  

(3.11)

where \( Target_n(x, y) \) is the target decision matrix with value 1 at pixels representing targets, and \( L \) is the threshold for the track length.
3.2.6 Determining appropriate Maneuvering Tolerance Factor

In section 3.2.4, a systematic methodology of determining a threshold for target detection alarm was introduced. However, the penalty weight \( w_k \) in equation 3.8 consists of both the directional penalty weight \( w_{dir} \) accounting to the change in direction from the previous frame, and \( w_{dir=9} \) which accounts for maneuvers. The maneuvering tolerance factor \( p \) scales both values as stated in equation 3.4.

Assuming a pixel representing a target moving in a straight path holds a saturated score as in equation 10, a change in the direction of the target will result into a new score that can be computed as follows:

\[
Score_{k,new} = 1 + b \cdot \left( 1 + \frac{1}{1 - b \cdot w_{dir,old}} \right) \cdot \left( p \cdot w_{dir=9} + (1 - p)w_{dir,new} \right) \tag{3.12}
\]

The algorithm will be able to keep tracking the target which performs a change in direction that maintains \( Score_{k,new} \) above the predetermined threshold; therefore the following inequality should be maintained

\[
Score_{k,new} > d \cdot Score_{k,old} \tag{3.13}
\]

Substituting from equations 11 and 13 leads to the following:

\[
p \cdot w_{dir=9} + (1 - p)w_{dir,new} > \frac{1}{b} \cdot \left( d - \left( 1 + \frac{1}{1 - b \cdot w_{dir,old}} \right)^{-1} \right) \tag{3.14}
\]

At this point, it is essential to decide on the maximum change of direction to be tolerated taking into consideration criteria such as the mechanical limitations of the target motion.
capabilities. Determining the maximum change of direction will lead into an easy computation of $w_{dir, new}$ using the predetermined penalty weight matrices, and thus leaving the maneuvering tolerance factor the only variable to consider.

3.2.7 Updating lost tracks

Losing measurements’ updates of a target location occurs frequently in active sonar systems. A data-dependant algorithm as the one documented in this research would lose track of a target when no updates are available for more than one ping. To overcome this deficiency of the algorithm, a simple expectation of the target location can be calculated based on previous target locations. The following procedure can be followed to update a target location when track loss occurs:

a. Search previous frame’s tracking matrix for tracks with length $> L_{lost}$ that didn’t get updated at current frame.

b. Use last $N$ samples of the track to model the motion path of the target.

c. Use the model to estimate target location at current frame.

d. Update tracking matrix and propagate score to next frame.

e. Update estimation counter to trace the duration for which the track was lost.

The motion model to be estimated here is a $1^{st}$ degree polynomial that fits the x-y coordinates for the last N samples in a least mean square sense. Using large N or a higher degree polynomial wouldn’t necessary improve the performance of the expectation model. This is because the track-history has less influence on next target location as the
data points used get older. In this research, a maximum of N=5 is used to predict the next location of the target when track-loss occurs.

In order to properly approach the motion modeling problem, it is essential to make some basic assumptions about the dynamics of the targets. The following assumptions will be considered:

1. The target is assumed to continue moving in the same direction prior to losing track.
2. The target is assumed to travel at approximately the same velocity when the track is lost.

Thus, the motion model specified as a straight line

\[ y = A \cdot x + B \]  \hspace{1cm} (3.15)

Where

\[ B = \frac{\sum_{i=1}^{N} x_i^2 \sum_{i=1}^{N} y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i x_i}{N \sum_{i=1}^{N} x_i^2 - (\sum_{i=1}^{N} x_i)^2} \]  \hspace{1cm} (3.16)

\[ A = \frac{N \sum_{i=1}^{N} y_i x_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{N \sum_{i=1}^{N} x_i^2 - (\sum_{i=1}^{N} x_i)^2} \]  \hspace{1cm} (3.17)

Since it is assumed that the target is moving in a straight line with the same magnitude velocity and with a consistent direction, the velocity on the x-axis and the y-axis should be consistent as well. A good estimation of the velocity can be obtained by substituting
the x and y coordinates of the start and end points of the last segment of the track with length N. From that

\[ x_{new} = \frac{x_{i-1} - x_{i-N-1}}{N} + x_{i-1} \]  

(3.18)

\[ y_{new} = \frac{y_{i-1} - y_{i-N-1}}{N} + y_{i-1} \]  

(3.19)

It is essential before attempting to form a motion model to identify the special cases where the target is not moving, or moving in a horizontal or vertical line. When the target is not moving, it is natural to assume that the next expected location would be the same as the last point. When the path of motion is changing along only one of the axis, the target is assumed to keep moving along the same axis with the average speed calculated using the last N positions.

3.3 Hough transform

The Hough transform is a feature extraction technique used commonly in digital image processing. It is practical in applications where the readings are noisy or the features to be extracted suffer from gaps between the pixels [46, 47]. This is of great importance in the application discussed in this research as the track of the target, even in the straight line scenario, will usually consist of sparse unconnected pixels due to the time separation between the pings. This time gap might be large enough to allow the target to move several pixels before it is detected again by a subsequent ping. The width of the gap depends essentially on the speed of the target, the time separation between the
pings and the number of nodes in the SOM (since it determines the size of the cell each pixel represents). Errors in the readings and possible missed detections of the target can also contribute to the length of the gaps.

For a point \((x_i, y_i)\), any line crossing this point can be represented by

\[
x_i \cdot \cos(\theta) + y_i \cdot \sin(\theta) = \rho
\]

(3.20)

where \(\rho\) represents the distance between the line and the origin, while \(\theta\) is the angle of the vector from the origin to this closest point. Thus all the lines passing through the point \((x_i, y_i)\) can be represented in the \(\rho-\theta\) plane with a sinusoidal shaped curve as indicated in equation 3.20. A line crossing two data points is represented in the \(\rho-\theta\) plane by the intersection point of the curves belonging to those two points. Repeating for all data points and keeping track of the intersection points in an accumulator matrix makes it possible to choose which lines to show based on the gap between the points and the number of points it passes through [48] as shown in Figure 3.8.
While an easy choice for the minimum number of pixels required to be crossed by one line and the maximum gap between those pixels would be the same as discussed in section 3.2.1 and 3.2.5, it might be useful to consider choosing lower thresholds for the Hough transform detection. For a target following a straight path, while its detected path track might satisfy the length threshold $L$, inaccuracies in determining the time of arrival of a ping reflection and its bearing might result in having the pixels representing the target’s path to be deviating of the straight line.
Chapter 4

Simulation and Field Test Data

The method developed in this research is verified by both simulation and real field test data to assess the merits and the limitations. Three types of simulation were used to analyze the performance of the tracking algorithm. The level of complexity of data representation increases through the different types of simulations.

4.1 Type 1: Pixel targets

In this type of simulation, a preliminary test of the performance of the algorithm was conducted. Multiple targets moving at variable speeds were simulated as moving pixels. The targets appeared at different time instances to test the algorithms ability to adapt to multiple and new targets. Moreover, stationary point-targets were also simulated to investigate the ability of the tracker to identify targets with zero velocity. Results of the performance of the tracking algorithm with pixel targets are illustrated in chapter 5.

4.2 Type 2: Contact level simulation

In this simulation, a scenario where 4 by 4 array of receiver sonobuoys with the transmitter in the middle of the array was simulated. Two targets with different paths are visible to the arrays of sonobuoys. It is assumed that the closest 3-6 sonobuoys will have readings corresponding to the target location, giving a probability of detection equal to 1. The number of sonobuoys to detect the target varies within the range specified at each ping. Figure 4.1 shows the layout of the sonobuoys and the target paths.
The input to the simulation is generated range information and bearing. A Gaussian error with variance 20 meters was introduced to the range readings and a Gaussian error with variance 2° was introduced to the bearing. Figure 4.2 and Figure 4.3 show the data generated for the target location for the ideal and the inaccurate measurements scenarios respectively.

Figure 4.1 Contact level Simulation - Targets' paths
Figure 4.2 Contact level Simulation - Ideal contact data

Figure 4.3 Contact level simulation - Inaccurate contact data
In addition to the inaccuracy introduced to the measurements indicating the target position, false contacts were introduced to increase the level of complexity. In one scenario, 2 false contacts per sonobuoy were generated giving a total of 32 false contacts per ping. In the second scenario, 5 false contacts per sonobuoy were generated giving a total of 80 false contacts per ping. The data generated for the targets’ positions for 60 pings in addition to the false contacts for the two scenarios mentioned earlier are illustrated in Figure 4.4.a and Figure 4.4.b respectively.

![Contact Level Simulation - Scenario 1](image)

Figure 4.4.a Contact level simulation - Scenario 1
4.3 Type 3: Acoustic level Simulation

An MSA scenario was simulated by assuming one transmitter sonobuoy centered in the middle of a 4 x 4 array of receiver sonobuoys. The source and the receivers were at a depth of 100 m. Every 60 seconds, the transmitter emitted a continuous wave (CW) ping of 1 second duration at a frequency of 200 Hz. The receiver sonobuoys are capable of estimating the bearing to the origin of the ping reflection detected by the sonobuoy. The locations of the sonobuoys were assumed to be known and relatively stable.

The ping transmitted and reflected back to one of the receivers will go through propagation losses simulated based on a module for normal mode generation which is a part of a simulation program developed at the Centre for Marine Science and Technology.

Figure 4.4.b Contact level simulation - Scenario 2
at Curtin University of Technology (Actup version 2.2). The selection of normal modes
takes into consideration different environmental conditions including propagation path,
sound speed profile, diffraction and reflection coefficients of different levels of the ocean
and other factors affecting the propagation loss profile of sound [49, 50, 51, 52]. The
sound propagation profile is shown in Figures 4.5.a and 4.5.b. shows the
variation of propagation loss with depth and range for a source at a frequency of 100 Hz
and a depth of 100 m while Figure 5b shows propagation loss versus range for a 200 Hz
source at a depth of 100 m with a receiver at a depth of 100 m.

Figure 4.5.a Transmission loss versus range and depth for a source of frequency 200 Hz and
at depth of 100 m
Figure 4.5.b Transmission Loss versus range for a source of frequency 200 Hz at depth of 100 m and receiver depth of 100 m

During the 60 seconds time interval between emitting two consecutive pings, the DIFAR sonobuoy would receive a direct ping arrival from the transmitter and a possible reflection from the target. This reflection was modeled based on the simple assumption of perfectly spherical target which reflects incident wave in all directions. Furthermore, erroneous readings were simulated by high level additive Gaussian noise which would misguide the matched filter into generating false readings of ping reflections.

Bearing estimation of the data points generated by the matched filter was done by utilizing the construction of the DIFAR sonobuoy which allows it to acquire signal
information at three channels referred to as the omni, sine, and cosine channels. The
relation between the acquired signal and angle of arrival is given as follows:

\[ x_{ok} \quad \text{Omnidirection Channel} \]  
\[ x_{sk} = x_{ok} \cdot \sin \theta \quad \text{Sine Channel} \]  
\[ x_{ck} = x_{ok} \cdot \cos \theta \quad \text{Cosine Channel} \]

where \( k \) represents the index of the series, \( k=1\ldots N \). The Bartlett beamforming technique
was used to estimate the direction of arrival \( \theta \). An estimate, \( \hat{Q} \), of the cross-spectral
matrix needs to be formed [53]

\[ \hat{Q} = \begin{bmatrix} \Phi_{oo} & \Phi_{os} & \Phi_{oc} \\ \Phi_{os}^* & \Phi_{ss} & \Phi_{sc} \\ \Phi_{oc}^* & \Phi_{cs}^* & \Phi_{cc} \end{bmatrix} \]  

(4.4)

where o, s, and c refer to omni, sine, and cosine channels, and the asterisk denotes the
complex conjugate. Knowing the carrier frequency of the ping transmitted, the
conventional beamforming technique relies on an iterative approach by which it goes
through a range of angles in predefined steps to form a steering vector,

\[ \beta(\theta) = \begin{bmatrix} 1 \\ \sin(\theta) \\ \cos(\theta) \end{bmatrix} \]  

(4.5)

At each angle, \( \theta \), an estimate of the conventional beam power is computed from the
steering vector as follows:
\[ \hat{E}_{CB} = \beta^T \hat{Q} \beta \]  

(4.6)

The angle which maximizes the beam power \( \hat{E}_{CB} \) was considered the angle of arrival of the received signal.

The time-bearing information and the sound speed were used to construct an ellipse with the transmitter and receiver locations as foci. The bearing obtained for that data point from the receiver sonobuoy pointed to its location on the ellipse. Figure 14 shows the 16 DIFAR sonobuoys and the bearing acquired for the target in ideal conditions.

Figure 4.6 Scenario of 1 transmitter and 4 x 4 array of receivers
4.4 Real Field Test Data [Seabar’07]

A multistatic active sonar system was deployed and tested in the Seabar07 sea trial on the Malta Plateau, South of Sicily, 10 -22 October, 2007\(^1\) [54, 55]. The dataset to be analyzed in this paper is called A56, combining the two events A05 and A06. Run A56 took place on October the 18th, 2007 during the time period 8:41:00 to 11:38:00. In this run, one transmitter was used as well as two receivers (RX2 and RX3). A total number of pings equaling 139 were transmitted during this time period. The testing and analysis documented in this research is done using the CW contacts acquired without attempting to identify target contacts using Doppler shift.

In order to be able to fairly evaluate the performance of the tracking system discussed earlier with the real data from the Seabar07 – A56 trial, sources of error in the data is investigated. A first measure of the data accuracy is performed by trying to obtain the source location using the time arrival information and bearing of the direct blast at the receiving sonobuoys, speed of sound in water provided (1500 m/sec) and the GPS positions of the receiver sonobuoys. Figures 4.7.a and 4.7.b show the data obtained from the receiving sonobuoys indicating the location of the transmitter. It can be clearly observed that there is an offset in the GPS location of the transmitter and the best approximation of the transmitter location indicated by the intersection of the arcs of the data points obtained from the receiver sonobuoys. The source of this error can’t be confidently determined using the information available. It might be due to an error in the

\(^1\) The SEABAR’07 sea trial was held by the NATO Undersea Research Centre (NURC). The SEABAR’07 data was made available under agreement with NURC and Defence Research and Development Canada – Atlantic.
GPS readings of the transmitter of the receiver sonobuoys, or due to a miscalculation of the speed of sound in water.

Figure 4.7.a Seabar data analysis - Transmitter position

Figure 4.7.b Seabar data analysis - Transmitter position – zoom in
Using the average speed of sound in water obtained by dividing the distance between the transmitter and receiver using their GPS locations by the time duration needed for the transmitted ping to arrive at the receiver can be shown in Figure 4.8 and Figure 4.9. However, utilizing the average speed obtained in determining the path of the target did not yield to better results. This implies that the source of error might be more related to a GPS localization error or a varying speed of sound in water.

Figure 4.8 Speed of sound propagation in water - RX2
Analysis of the contact data obtained from the receivers suggest the occurrence of a time-delay in the arrival of reflections from the target represented by pings transmitted from an echo repeater as shown in Figure 4.10. A better insight of the delay in time information associated with some of the data points can be obtained by observing the SNR levels of the detections collected by sonobuoys with respect to the time of arrival. Figure 4.11.a and Figure 4.11.b illustrate this information for receiver RX3, while Figure 4.12.a and Figure 4.12.b represent the same type of information for receiver RX2. It can be clearly seen that a time shift is occurring at both sonobuoys for the trace of detections of the target. A probable explanation is that the time delay is occurring at the echo-repeater representing the target, and thus the error is occurring at both of the receiver sonobuoys at the same pings. The data analyzed in this research is not corrected for the delay discussed, and the error is dealt with as an inherited inaccuracy in the data.
Figure 4.10 SEABAR A56 CW data - SNR threshold = 15 db
Figure 4.11.a SNR levels vs. time of arrival - RX3

Figure 4.11.b SNR levels vs. time of arrival - RX3
Figure 4.12.a SNR levels vs. time of arrival - RX2

Figure 4.12.b SNR levels vs. time of arrival - RX2 – zoom in

83
Chapter 5

Results and Discussions

5.1 Pixel Point Targets

The first testing of the algorithm was carried by simulating multiple targets directly as pixels moving in a consistent path throughout the sequence of frames. In this scenario, 3 targets were stationary throughout the 30 frames, and 3 targets were moving, making different numbers of maneuvers at different angles. At each frame, 200 erroneous readings were introduced in addition to the correct positions of the targets. For this scenario, all of the targets were detected by the tracker by ping number 7 and tracked through frame number 30. Figure 5.1.a shows the readings collected in 30 frames with the target locations emphasized for easier visualizations and Figure 5.1.b shows the tracks detected by the algorithm as well as the output after performing line detection using Hough transform.
Figure 5.1.a Input to the tracker containing 6 targets

Figure 5.1.b Tracker output for 6 targets
The goal from testing the tracking method using targets represented as pixels is to focus on analyzing the performance of the dynamic programming stage given ideal input. By choosing to represent the targets and false contacts as pixels, the stage where data discretization is performed using SOMs is skipped because this is the simplest form of data that can be obtained. The output of the tracking method verifies that the dynamic programming algorithm developed is effective in tracking multiple targets moving at different speeds. Moreover, the tracker was able to detect all the targets with no error in positional estimation.

However, analysis of the tracker using pixel targets pointed out the problem of false targets birth when stationary targets exist within the search area. Since false readings can take place close to a stationary target location, the tracker might falsely identify that false reading as an indication that the target started to move. However, this is not considered a severe problem since the original stationary target will remain detected given availability of data representing its location. Moreover, the false targets disappear rapidly as no updates are available for their path hypothesis.

5.2 Contact Level Simulation

Analysis of the contact level simulation was initiated by testing the tracking algorithm with the ideal contact data and the inaccurate contact data obtained from the receivers as described in section 4.2. Figure 5.2 and Figure 5.3 show the final tracking output of the two scenarios respectively.
Figure 5.2 Tracker output for ideal data

Figure 5.3 Tracker output for inaccurate data
It is noticed from the output of the tracker in the case where the inaccurate contact data is fed into the algorithm, that the Hough transform failed to connect the second part of the path of target 1 with a line that gives a better approximation of the target movement. This is a result of the deviation of the activated nodes off the straight line of movement the target is supposed to follow. This yielded to exceeding the thresholds of the output of the Hough transform of ignoring discontinuity and deviation of the pixels from a straight line. Although the straight line estimation is not available as an output of the tracker for the last segment of the path of the target 1, it is still possible to track and estimate previous positions of the target using the activated pixels nodes.

Results of the analysis of the different scenarios for the contact level simulation for different number of false contacts combined with ideal and inaccurate contact data as described in section 4.2 is illustrated in Figures 5.2 to 5.23. The figures show the tracker output at different instances, number of false alarms at each ping as well as the error in estimating the target position. It can be noticed that the number of false alarms increases as the number of false contact data introduced to the tracker increases. However, the target-position accuracy of the tracking algorithm does not significantly decrease as the number of false contacts increase since the output of the Hough transform provides a target path with a relatively better accuracy than the one provided by activated nodes. A detailed analysis of the results obtained for each scenario of the contact level simulations is provided in the following sections.
5.2.1 1st Scenario: Ideal measurements and 2 false contacts per sonobuoy

In this scenario, ideal measurements of the target position were fed into the tracking algorithm along with 2 false measurements per sonobuoy. Since the measurements were error-free, the tracking algorithm was able to trace the movement of both targets at all time instances starting ping 6 as illustrated in Figure 5.4, Figure 5.5 and Figure 5.6. Although the tracks of both targets did not have gaps or deviations, the lines detected of the Hough transform made it easier to visualize the paths followed by the targets.

![Figure 5.4 Ideal data - Two false contacts per Sonobuoy – Frame 29](image-url)
Figure 5.5 Ideal data - Two false contacts per Sonobuoy – Frame 41

Figure 5.6 Ideal data - Two false contacts per Sonobuoy – Frame 60

90
The number of false tracks generated by the tracking algorithm did not exceed 2 false tracks at any time instance as illustrated in Figure 5.7, and these false tracks disappeared rapidly as no updates for their tracks could be obtained. Low number of false alarms is mainly due to the relatively low number of false contacts generated for the simulated 60 pings (1920 false contacts). Additionally, the error in estimating the target position did not exceed 5 meters for Target 1 and 100 meters for Target 2 as shown in Figure 5.8. Although the measurements of the target position that are fed into the algorithm are ideal, the error in estimating the target position is mainly due to the discretization process. Since each node in the SOM is to cover an area of 200 meters, a maximum error of 100 meters is to be expected regardless of the accuracy of the data if the representation of the target position is to be done using the activated nodes of the SOM. Although this can be overcome by considering the data points that activated the nodes and thus a zero localization error can be obtained in this scenario, the position estimation mechanism in the tracker obtained for a target gives priority to the output of the Hough transform when available.
Figure 5.7 False Alarms - Ideal data - Two false contacts per sonobuoy

Figure 5.8 Target position error - Ideal data - Two false contacts per sonobuoy
5.2.2 2nd Scenario: Ideal measurements and 5 false contacts per sonobuoy

Ideal measurements of the target position were fed into the tracking algorithm in this scenario similar to the 1st scenario; however, 5 false measurements per sonobuoy were introduced to the algorithm in order to test the robustness of the system. The algorithm tracked both targets at all time instances starting at ping 6 although more false contacts were introduced in this scenario as illustrated in Figure 5.9, Figure 5.10 and Figure 5.11. The Hough transform for line detection facilitated observing the tracks generated for the real targets among other false tracks generated at different time instances.

Figure 5.9 Ideal data - Five false contacts per Sonobuoy – Frame 24
Figure 5.10 Ideal data - Five false contacts per Sonobuoy – Frame 41

Figure 5.11 Ideal data - Five false contacts per Sonobuoy – Frame 60
As the number of false contacts introduced to the tracking algorithm increased to a total of 4800 false contacts, the number of false tracks produced by the tracker increased accordingly reaching a maximum of 7 false tracks as shown in Figure 5.12. However, the error in estimating the target position remained the same as in the previous scenario in which less false contacts were introduced as shown in Figure 5.13. This emphasizes the ability of the tracking algorithm to connect measurements representing a consistency in movement as the number of false contacts increases.

![Figure 5.12 False alarms - Ideal data - Five false contacts per Sonobuoy](image-url)
5.2.3 3rd Scenario: Inaccurate measurements and 2 false contacts per sonobuoy

Instead of the ideal measurements of the target position, inaccurate measurements as discussed in section 4.2 were fed into the tracking algorithm along with 2 false measurements per sonobuoy. Although an error was introduced in the measurements of actual locations of the targets, the tracker was able to detect and track both targets at all time instance starting ping 8 as illustrated in Figure 5.14, Figure 5.15 and Figure 5.16. However, it can be noticed that the tracker was following wrong measurements in few segments of the tracks as the error introduced to the measurements of the target location indicated a false change in direction. The Hough transform provided a more reliable estimation of the target path when deviations in the measurements occurred.
Figure 5.14 Inaccurate data - Two false contacts per Sonobuoy – Frame 29

Figure 5.15 Inaccurate data - Two false contacts per Sonobuoy – Frame 45
The number of false tracks generated by the tracking algorithm remained within the boundaries of 2 false tracks at any time instance as shown in Figure 5.17. However, the error in estimating the target position increased since the position estimation process is data dependant and the measurements obtained are inaccurate. It can be noticed from the error in estimation illustrated in Figure 5.18 that when the Hough transform failed to connect the activated nodes, the position error increased dramatically compared with previous scenarios, reaching a maximum of 435 meters.
5.2.4 4th Scenario: Inaccurate measurements and 5 false contacts per sonobuoy

The contacts fed into the tracking algorithm in this scenario are inaccurate measurements of the actual locations of the targets mixed with 5 false contacts per
sonobuoy. Figures 5.19, 5.20 and 5.21 illustrate the ability of the tracking algorithm to identify and track the targets at all time instances starting ping 8.

Figure 5.19 Inaccurate data - Five false contacts per Sonobuoy – Frame 29

Figure 5.20 Inaccurate data - Five false contacts per Sonobuoy – Frame 49
The tracking algorithm produced more false tracks as the number of false contacts increased, reaching a maximum of 10 false contacts at any time instance as shown in Figure 5.22. The position estimation error for Target 2 was not affected in this scenario as the Hough transform was able to provide reliable estimation of the target position. For Target 1, the error margin for the position estimation was decreased dramatically when the Hough transform was able to detect lines of movement. However, when a position estimation of the Hough transform wasn’t available; the error increased dramatically reaching a maximum of 300 meters as shown in Figure 5.23. Moreover, it can be noticed that the last line of movement detected while the target was moving north-east was based on erroneous measurements, resulting into an increasing error in estimating the target position for the last 6 pings.
Figure 5.22 False alarms - Inaccurate data - Five false contacts per Sonobuoy

Figure 5.23 Target position error - Inaccurate data - Five false contacts per Sonobuoy
5.3 Acoustic level simulation

For the acoustic level simulation, two scenarios were considered for analysis each with a target path of different level of complexity. In the first scenario shown in Figure 5.24, a simple trajectory of a target moving at a speed of 3 m/sec in a straight line was simulated as discussed in section 4.3. The transmitter produced 15 pings at 60 second intervals. The noise level at the receiving end of the system varied with each ping at the different receivers depending on the total distance traveled by the ping and the attenuation profiles discussed previously. A measurement of the number of erroneous readings at each ping is a more realistic measure of the performance of the algorithm. The target was detected by the tracker at ping number 8; however, estimation of earlier positions of the target is available in the tracking matrix.
The output of the tracking algorithm including the line detection using Hough transform is shown in Figure 5.25. It can be noticed from the output of the tracking algorithm, that due to the inaccuracy of target positions obtained and the false contacts generated by the high noise, the target path presented by the activated nodes does not show precisely the straight line of movement of the target. However, with the aid of the Hough transform, the gaps and deviations of the target positions detections were overcome and a more accurate estimation of the target path could be obtained.

Figure 5.25 Tracker output - Acoustic level simulation - Scenario 1
The second scenario, shown in Figure 5.26, was an attempt to test the algorithm and illustrate its limitations. The target in this scenario made two sharp maneuvers while moving at a speed of 3 m/sec. 60 pings of data were collected.

![Acoustic level simulation - Scenario 2](image)

**Figure 5.26 Acoustic level simulation - Scenario 2**

The algorithm is applied twice with two different *maneuvering tolerance factor* values. The first run shown in Figure 5.27.a, 5.27.b and 5.27.c, used $p = 0.4$ while the second run, shown in Figure 5.28.a, 5.28.b and 5.28.c, used $p = 0.8$. When using small $p$, a cleaner track can be obtained; however, the algorithm will be less tolerant of sharp maneuvers. Thus, the track detected the target at ping 13 but lost it at the first maneuver. The algorithm detected the target again later, but failed to connect the two segments, and lost it again after ping 53 due to noisy data.
Figure 5.27.a Tracker output for Scenario 2: $p=0.4$ – Frame 16

Figure 5.27.b Tracker output for Scenario 2: $p=0.4$ – Frame 41
As shown in Figure 5.28, when choosing large $p$, the tracker is able to follow the target detected at ping 7 even after a sharp maneuver. However, this happens at the expense of picking up additional noise and identifying it as a target as shown in the tracker output at ping 60.
Figure 5.28.a Tracker output for Scenario 2: \( p = 0.8 \) – Frame 14

Figure 5.28.b Tracker output for Scenario 2: \( p = 0.8 \) – Frame 48
For scenarios 1 and 2, Figure 5.29 and Figure 5.30 show the maximum value of SNR obtained at the receivers and the number of false contacts for the runs discussed previously. Figure 5.31 shows the error in the target position estimate. The number of false alarms for each scenario within the 10 Km$^2$ search area is illustrated in Figure 5.32. When the algorithm indicates that a line was detected, it is assumed that the line is a more accurate representation and thus the error is computed by calculating the distance between the line and the target position at ping $n$. 
Figure 5.29: Acoustic level simulation - Maximum SNR levels in dB acquired by sonobuoys

Figure 5.30: Acoustic level simulation - Number of false contacts acquired by sonobuoys
Figure 5.31: Acoustic level simulation - Error in tracker position estimation

Figure 5.32: Acoustic level simulation - Number of false alarms
5.4 Real data analysis

The A56 run in the SEABAR07 trial introduces interesting tracking challenges such as maneuvers and fading detection [57]. In order to limit the number of contacts acquired by the receivers, the following thresholds were introduced:

- SNR threshold = 0 dB.
- Contact number threshold = 10

The contacts that satisfy the thresholds mentioned above shown in Figure 5.33 are fed into the tracking algorithm. Since detections from the target aren’t always available and thus the probability of detection POD<1, the tracking algorithm will have to use the estimation mechanism imbedded in the algorithm when tracks are lost.

![Figure 5.33: Run A56 data after thresholding](image-url)
The following parameters were used for the tracking algorithm with run A56:

- Maneuvering tolerance factor = 0.6
- Minimum track length to start estimating tracks = 9
- Maximum number of estimations for lost tracks = 6

In addition to that, the Hough transform is tolerating a gap of 6 pixels in the line detection step. The lines that achieved a score within the maximum 10% in the accumulator were considered as possible tracks. The output of the tracker after the line detection step is illustrated in Figure 5.34, and track of the target compared with the GPS position of the target is shown in Figure 5.35.

![Figure 5.34: Tracking output for run A56](image-url)
The results point up that the best estimation of the tracker was the one obtained by the Hough transform. The ability to skip gaps and consider the majority of the pixels connected to form a line were very useful criteria as they helped overcome the error introduced by the shifted segments in the horizontal part of the target path. The mean square error was 103.646 meters, which could be reduced to 57.547 meters if the data were corrected for the shift in the last segment of the track. The total number of false tracks at ping 139 generated by tracking algorithm was 67 tracks.

The dataset of run A56 was examined by other groups using other filters and tracking algorithm. In [56], the same threshold criteria was followed for the analysis of run A56 using the Distributed Multi-Hypothesis Tracker (DMHT). The results achieved
were a position mean square error ranging from 50 to 150 meters, and the number of false tracks generated by the algorithm ranged between 72 and 77 tracks. The Gaussian Mixture cardinalized probability hypothesis density GM-CPHD [57] was experimented with the A05 dataset achieving a position mean square error of 348.72 and generating 37 false tracks. The results achieved by the tracking algorithm documented in this paper are comparable to the ones achieved by other filters. Moreover, since the tracking algorithm is based on discretization and clustering data as a first step in the algorithm using SOMs, the amount of computations would decrease as the number of data points to be considered decreases through the discretization phase.
Chapter 6

Conclusions, Contributions and Future Work

6.1 Summary

The primary objective of this research was to suggest a tracking algorithm capable of dealing with the complexity of the Multistatic Active Sonobuoy tracking systems while providing efficient tracking solutions for multiple targets with unknown number. This research overviewed existing techniques in multiple target tracking and their limitations, and provided an efficient tracking system with competitive performance and relatively low complexity.

The self organizing maps (SOMs) were used in order to simplify the representations of the large number of detections by the sonobuoys in the MSA scenario. The task of the SOM was to discretize the data while maintaining indexing of the nodes activated in the map. By applying a labeling algorithm to the connect nodes of the SOM, each labeled cluster was treated as a group of measurements belonging to one object, and an approximation of the readings were computed. The output of the SOM is a binary image frame that is forwarded to the next block in the tracking algorithm.

The second task carried on by the tracking algorithm is identifying targets’, their locations, and their previous tracks. This task was carried on by a Dynamic programming track-before-detect algorithm. The dynamic programming breaks the tracking problem into a set of smaller problems, making the task easier and lower in complexity. With the current binary image frame as an input, the dynamic programming algorithm computes a
score for each pixel that indicates the likelihood of each pixel to be representing a target based on motion analysis propagated through each iteration. Once a pixel exceeds a certain threshold, it is declared to be representing a target, and the user is alarmed about the existence of a possible target with its current and previous locations shown.

The final task of the tracking algorithm is to provide a relatively more accurate localization of the target. This step is carried on by the Hough transform. By trying to detect lines of motion in the tracks of possible targets, the algorithm can overcome some errors in measurements and provide an easier visualization of target tracks. The Hough transform provides an efficient solution to connect the gaps between the target locations represented by disconnected pixels.

The remainder of this chapter outlines the main conclusions of this research. It also discusses the contributions of the research and provides some recommendations for future work.

6.2 Conclusions

In this research, a method for tracking multiple targets in an MSA sonar system is described and evaluated using various sets of simulated data and a set of real-data obtained from a sea trial. The following conclusions were drawn from the experimental work conducted in this research.

a. The Self Organizing Map used to discretize the data as a first step before inputting the measurements set to the dynamic programming algorithm is a powerful tool in
reducing the amount of data provided by the measurements. By applying a clustering technique to connected nodes and obtaining a meaningful representative measurement of the cluster data, the amount of data can be reduced dramatically which in turn reduces the amount of computations needed in the other tracking algorithm operations.

b. Choosing an efficient spacing between the nodes of the SOM is a critical issue. While choosing relatively large spacing can reduce the computational complexity, it will reduce the accuracy of the tracking algorithm and will increase the risk of associating false measurements within clusters containing target position measurements. On the other hand, relatively small spacing between the nodes of the SOM can increase the accuracy of the tracking algorithm. This will result into increasing the amount of data propagated in the tracking algorithm and thus increasing the computational time.

c. Determining a correct search area for possible previous locations of targets depends on several factors such as speed limitations of the target, tolerated error in measurements and spacing between the nodes in the SOM. Calculating the size of the search area is a critical task as it serves as a gating for the measurements to be considered as previous locations of possible targets in a previous frame.

d. A well designed dynamic programming algorithm simplifies the tracking problem while maintaining efficient target localization performance. Moreover, by using predefined matrices and with the reduction of amount of data obtained by the SOM, the amount of calculations was reduced dramatically. While the algorithm provided is measurement dependent and does not incorporate motion model, it can still demonstrate
effective results when tracking multiple targets by relying on directional consistency in movement of possible targets. Moreover, optional target location estimation can be incorporated in the algorithm to keep track of targets when measurements are lost when target tracks meet predefined thresholds.

e. Applying Hough transform to tracks of detected targets reduced errors in targets’ location occurring due to measurements error, false measurements or caused by the tracking algorithm mainly in the discretization phase. The lines of movements detected by the Hough transform provided more accurate position estimation of the targets.

f. The performance of the tracker is comparable to other tracking techniques as shown when analyzing the performance of the tracking algorithm with the sea trial data set. The algorithm is independent of the number of targets and requires a low amount of computations. However, the algorithm is data-dependent and thus noisy readings will degrade the performance of the tracker. Failure to detect the target will result into broken track segments, which can be overcome by a linear estimation model based on previous track data.

6.3 Thesis contributions

Based on the research objectives and the results presented, the goals established in chapter 1 have been achieved. To state that the research completely suggests an optimum tracking algorithm for multiple targets for the Multistatic Active Sonobuoy scenario would be incorrect, as the algorithm provided has its shortcomings in terms of accuracy and when dealing with very noisy measurements. However, the algorithm suggested is an
efficient tracking algorithm that is capable of providing a fast, computationally inexpensive tracking solution for unknown and varying number of targets.

A major success in this research is the ability of the tracking algorithm to identify and track targets without depending on a motion model, which drastically lowers the system complexity of system. Even with noisy data, the tracking algorithm was able to provide competitive results depending only on the measurements provided by the sensors. This research confirms that using a well designed dynamic programming algorithm aided with other auxiliary tools can provide an efficient solution for the problem of tracking multiple targets in scenarios where multiple sensors are used.

6.4 Recommendations and Future work

With any suggested method or technique, there is always potential for related research with the goals of enhancing the performance of the algorithm or researching a new approach inspired by the methodology used. Some recommendations and suggestions for future work are provided below.

a. While the current algorithm utilizes the Hough transform for line detection only, this limits the improvements in target localization to scenarios where targets move in straight lines. Although this is very useful in many scenarios as moving in a straight direction is a basic assumption used in several tracking algorithms, it would be beneficial to examine using the Hough transform to detect other patterns of motions such as arcs. Although this might increase the level of complexity, it could provide significant
improvement in target location estimation especially when the target is trying to maneuver.

b. Although the tracking method proposed in this research is a measurement based approach, combining the algorithm with other traditional tracking techniques shouldn’t be excluded. When dealing with a high level of false measurements, it might be useful to incorporate an additional estimation technique based on target-motion limitations and dynamics as well as a study of the noise characteristics at the sensors in order to provide a more robust solution.
References


