A Thesis

entitled

Target Tracking Via Marine Radar

by

Nishatha Nagarajan

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the

Master of Science Degree in Electrical Engineering

Dr. Mohsin M. Jamali, Committee Chair

Dr. Peter V. Gorsevski, Committee Member

Dr. Richard G. Molyet, Committee Member

Dr. Patricia R. Komuniecki, Dean
College of Graduate Studies

The University of Toledo

December 2012
Copyright 2012, Nishatha Nagarajan

This document is copyrighted material. Under copyright law, no parts of this document may be reproduced without the expressed permission of the author.
An Abstract of
Target Tracking Via Marine Radar

by

Nishatha Nagarajan

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the Master of Science in Electrical Engineering

The University of Toledo
December 2012

The growing energy needs have eventually increased the development of wind turbines. The constructions of wind turbines have several potential impacts of which the most significant factor is the increasing bird mortality rates due to collision and habitat loss. Since then, radars have been deployed to study the behavior of birds towards wind turbines. Radars employ target tracking for identifying the targets (birds) accurately and efficiently. Several methods of tracking were developed to improve the tracking efficiency of the radars over the years. Most widely used tracking techniques are Kalman filter and particle filter. These filters use data with random errors and estimate accurate values for the current state of the system. Kalman filter is a linear estimator which does not depend on a set of past observations and hence efficient in real time applications. The particle filter also known as sequential Monte Carlo method is a nonlinear estimator which uses a set of particles with various weights for estimation. However, particle filters have high computation time. Kalman and particle filters were developed over the years creating various models for various types of systems. A block version of Compressive Matching Pursuit (CoSaMP) algorithm used in signal reconstruction called BCoSaMP was employed in tracking. It was seen to give a similar or better performance than
particle filter with less computation time. The BCoSaMP algorithm with Kalman filter estimation was developed which reduces the mean square error as compared to other models in certain cases.

This thesis focuses on developing tracker models in radR. Kalman filter tracking model based on linear data and Gaussian noise that operates over a variety of target motions and velocities is developed. Particle filter is designed for nonlinear target motion with non-Gaussian noise. BCoSaMP model that assumes data as sparse is applied for target tracking and a modified BCoSaMP which replaces least square estimation with Kalman filter estimation are also implemented. These models were tested with different data sets and a comparative analysis is performed. The algorithms are tested on simulated data and marine radar data in radR to compare the effects of the developed tracker models with the conventional methods in radR. The hybrid algorithm is shown to have better performance over the other models in the case of simulated track for some targets. Particle filter has the highest detection rate with marine radar data.
Acknowledgements

I am very thankful to the almighty God for everything that I am today

I am most grateful to my advisor Dr. Mohsin Jamali for giving me the opportunity to work in this project which has been a life cherishing experience. I would like to also thank Dr. Peter Gorsevski and Dr. Richard Molyet for taking time to serve on my committee. I am grateful for the continuous support given by Dr. Peter Gorsevski from Department of Geospatial Sciences at BGSU, Dr. Joseph Frizado from Department of Geology at BGSU and Dr. Verner Bingman from Department of Psychology at BGSU for this project which is partially funded by the Department of Energy (Contract #DE-FG36-06G086096). I would also like to thank all my lab mates especially Todd Schmuland and Edris Amin for their help and support. I would also like to thank Dr. Jeremy Ross for contributions to this work.

I am most grateful to know Shavin Thaddeus Shahnawaz and Vivek Sreedhar for all their guidance and for being there to help me always. I am very lucky to have very special friends in my life who have always been with me like Praneeth Nelapati, Sakina Junaghadwala, Sreenadh Reddy, Gubbala Uday Kishore and Midde Vijaya Kumar.

My loving family has been the greatest blessing in my life. I am most thankful to my loving mother Gajalakshmi and the coolest dad Nagarajan. I am very fortunate to have the most wonderful sister Yamini Padma and my little brother Akul Akshit.
# Table of Contents

Abstract .................................................................................................................................................. iii

Acknowledgements .................................................................................................................................. v

Table of Contents ..................................................................................................................................... vi

List of Tables ........................................................................................................................................... x

List of Figures ............................................................................................................................................ xii

1. Introduction .......................................................................................................................................... 1

   1.1 Methods for Bird Monitoring ........................................................................................................ 2

       1.1.1 Visual Methods ......................................................................................................................... 2

       1.1.2 Acoustic Data ............................................................................................................................ 2

       1.1.3 Radio Telemetry ......................................................................................................................... 3

       1.1.4 Cameras ....................................................................................................................................... 3

       1.1.5 Radars ......................................................................................................................................... 4

           1.1.5.1 Weather Radars .................................................................................................................. 5

           1.1.5.2 Marine Radars .................................................................................................................... 6

       1.2 Target Tracking ............................................................................................................................... 6

       1.3 Current Research .............................................................................................................................. 7

2. Radars in Avian Study ............................................................................................................................. 8

    2.1 Introduction ....................................................................................................................................... 8

    2.2 Marine Radars in Avian Study ......................................................................................................... 8
2.3 Methods of Reducing Ground Clutter..................................................9
2.4 Analysis of Marine Radar Data............................................................9
2.5 Previous Radar Studies.................................................................10
2.6 Summary......................................................................................14

3. Marine Radar and Radar Data Processing Software..............................15

3.1 Introduction................................................................................15
3.2 Marine Radar.............................................................................16
3.3 Digitizing Card............................................................................19
3.4 radR.........................................................................................24
  3.4.1 Radar Data Processing............................................................25
  3.4.2 Blip Processing.........................................................................28
3.5 Summary....................................................................................31

4. Tracker Models in radR.......................................................................32

4.1 Introduction................................................................................32
4.2 Tracking Techniques......................................................................33
4.3 Tracker Models in radR.................................................................34
4.4 Tracker Model..............................................................................35
  4.4.1 Nearest Neighbor Model........................................................37
  4.4.2 Multi Frame Correspondence Model........................................40
4.5 Summary....................................................................................46

5. Tracking Algorithms...........................................................................47

5.1 Introduction................................................................................47
5.2 Literature Review of Target Tracking Algorithms............................49
## List of Tables

3.1 Models of Furuno Mark 3 ........................................................................................................ 17
3.2 Types of radiator .................................................................................................................... 18
3.3 Specifications of the radar ..................................................................................................... 18
3.4 Blip processing parameters .................................................................................................. 28
3.5 Plugins in radR ..................................................................................................................... 30
4.1 Functions in tracker model in radR ..................................................................................... 37
4.2 Parameters in the nearest neighbor model .......................................................................... 40
4.3 Parameters in multi-frame correspondence model .............................................................. 44
4.4 Parameters in tracks.csv file ............................................................................................... 44
4.5 Sample data in tracks.csv file ............................................................................................ 45
5.1 Input and output parameters for Kalman filter ..................................................................... 64
5.2 Input data frame of targets in Kalman filter ........................................................................ 65
5.3 Input and output parameters for particle filter ..................................................................... 75
5.4 Input data frame of targets in particle filter ......................................................................... 75
5.5 Input and output parameters for BCoSAMP ......................................................................... 87
5.6 Input data frame of targets in BCoSAMP ............................................................................ 87
6.1 Optimized blip processing parameters ................................................................................ 129
6.2 Computation time of tracker models .................................................................................... 136
List of Figures

1-1 WSR88D data in identification of birds.......................................................5
1-2 Target detection using marine radar data...................................................6
3-1 Experimental setup.......................................................................................15
3-2 Radar trailer and the slotted array antenna at Bowling Green.......................16
3-3 Parabolic antenna at University of Toledo....................................................17
3-4 Block diagram of XIR3000C......................................................................19
3-5 Heading and bearing signals in radar data...................................................21
3-6 Digitizing card setup....................................................................................21
3-7 Radar data in radar sample application.......................................................22
3-8 Snapshot of parameters in radar sample application....................................23
3-9 Processing of data in radR..........................................................................25
3-10 Blip processing............................................................................................27
3-11 Sample radar data in radR..........................................................................29
3-12 Sample radar data filtered using blip processing in radR..........................30
3-13 Enabling plugin in radR..............................................................................31
4-1 Tracking flight paths of birds using radar...................................................33
4-2 Types of bird tracks......................................................................................33
4-3 Target tracking..............................................................................................34
4-4 Tracker models in radR................................................................................35
4-5 Tracking in radR………………………………………………………………………………..36
4-6 Algorithm for tracker model……………………………………………………………………37
4-7 Nearest neighbor model with track 1 created using minimum distance ..................38
4-8 Sample data in radR………………………………………………………………………………..39
4-9 Snapshot of tracks created by nearest neighbor plugin in radR.................................39
4-10 Digraph with five vertices……………………………………………………………………..41
4-11 Sample data in radR………………………………………………………………………………..43
4-12 Snapshot of tracks created by multi-frame correspondence plugin in radR...........43
5-1 Factors influencing tracking……………………………………………………………………..49
5-2 Track 1 &2 have the same possible target in the next state…………………………..53
5-3 Example of target recognition by multiple radars………………………………………54
5-4 Kalman filter design………………………………………………………………………………..61
5-5 Flowchart of Kalman filter design……………………………………………………………..63
5-6 Input and output parameters for Kalman filter………………………………………………64
5-7 Prediction stage in Kalman filter………………………………………………………………65
5-8 Update stage of Kalman filter…………………………………………………………………66
5-9 Association of parameters within two stages………………………………………………67
5-10 Implementation of Kalman filter algorithm in radR………………………………………68
5-11 Flowchart of Kalman filter tracker model……………………………………………………69
5-12 Kalman filter design in radR……………………………………………………………………70
5-13 Sequential importance sampling particle filter……………………………………………74
5-14 Input and output parameters for the particle filter……………………………………….74
5-15 Prediction for N particles…………………………………………………………………………76
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-16</td>
<td>Update in particle filter</td>
<td>77</td>
</tr>
<tr>
<td>5-17</td>
<td>Association of parameters within two stages</td>
<td>78</td>
</tr>
<tr>
<td>5-18</td>
<td>Implementation of particle filter algorithm in radR</td>
<td>78</td>
</tr>
<tr>
<td>5-19</td>
<td>Flowchart of particle filter tracker model</td>
<td>79</td>
</tr>
<tr>
<td>5-20</td>
<td>Particle filter design in radR</td>
<td>80</td>
</tr>
<tr>
<td>5-21</td>
<td>Flowchart of BCoSaMP algorithm</td>
<td>86</td>
</tr>
<tr>
<td>5-22</td>
<td>Input and output parameters for BCoSaMP</td>
<td>87</td>
</tr>
<tr>
<td>5-23</td>
<td>Prediction of target position</td>
<td>88</td>
</tr>
<tr>
<td>5-24</td>
<td>Update in BCoSaMP algorithm</td>
<td>89</td>
</tr>
<tr>
<td>5-25</td>
<td>Association of parameters within two stages</td>
<td>90</td>
</tr>
<tr>
<td>5-26</td>
<td>Implementation of BCoSaMP algorithm in radR</td>
<td>90</td>
</tr>
<tr>
<td>5-27</td>
<td>Flowchart of BCoSaMP Model</td>
<td>91</td>
</tr>
<tr>
<td>5-28</td>
<td>Modified BCoSaMP model for tracking</td>
<td>92</td>
</tr>
<tr>
<td>5-29</td>
<td>Algorithm of tracker model</td>
<td>93</td>
</tr>
<tr>
<td>5-30</td>
<td>Layout of tracker model in radR</td>
<td>94</td>
</tr>
<tr>
<td>5-31</td>
<td>Developed tracker models in radR</td>
<td>95</td>
</tr>
<tr>
<td>5-32</td>
<td>Implementation of new algorithm in radR</td>
<td>95</td>
</tr>
<tr>
<td>5-33</td>
<td>Comparison of current tracker models with Kalman filter</td>
<td>97</td>
</tr>
<tr>
<td>5-34</td>
<td>Comparison of current tracker models with particle filter</td>
<td>99</td>
</tr>
<tr>
<td>5-35</td>
<td>Comparison of current tracker models with BCoSaMP</td>
<td>101</td>
</tr>
<tr>
<td>5-36</td>
<td>New tracker models in radR</td>
<td>102</td>
</tr>
<tr>
<td>6-1</td>
<td>Study area-Ottawa National Wildlife Refugee</td>
<td>105</td>
</tr>
<tr>
<td>6-2</td>
<td>Single simulated track at constant velocity</td>
<td>106</td>
</tr>
</tbody>
</table>
6-3 Simulated multi-target tracks……………………………………………………………106
6-4 Simulated multi-target tracks in radR environment…………………………………107
6-5 Snapshot of marine radar data…………………………………………………………108
6-6 Kalman filter single target tracking……………………………………………………109
6-7 Error rate of Kalman filter……………………………………………………………109
6-8 Multi target tracking using Kalman filter…………………………………………….110
6-9 Error rates for the four tracks…………………………………………………………110
6-10 Kalman filter tracker model output in radR…………………………………………111
6-11 Kalman filter tracker model output with noise in radR……………………………112
6-12 Particle filter single target tracking…………………………………………………113
6-13 Error rate of particle filter……………………………………………………………113
6-14 Multi target tracking using particle filter…………………………………………114
6-15 Error rates for the four tracks…………………………………………………………114
6-16 Particle filter tracker model output in radR…………………………………………115
6-17 Particle filter tracker model output with noise in radR……………………………116
6-18 BCoSaMP single target tracking……………………………………………………117
6-19 Error rate of BCoSaMP tracking……………………………………………………117
6-20 Multi target tracking using BCoSaMP………………………………………………118
6-21 Error rates for the four tracks…………………………………………………………118
6-22 BCoSaMP tracker model output in radR……………………………………………..119
6-23 BCoSaMP tracker model output with noise in radR………………………………120
6-24 Modified BCoSaMP single target tracking…………………………………………121
6-25 Error rate of modified BCoSaMP tracking…………………………………………121
6-26 Multi target tracking using modified BCoSaMP
6-27 Error rates for the four tracks
6-28 Modified BCoSaMP tracker model output in radR
6-29 Modified BCoSaMP tracker model output with noise in radR
6-30 Error rates of four tracker models for a single track
6-31 Error rates of track 1 for four tracker models in multi target environment
6-32 Error rates of track 2 for four tracker models in multi target environment
6-33 Error rates of track 3 for four tracker models in multi target environment
6-34 Error rates of track 4 for four tracker models in multi target environment
6-35 Track of the balloon moving away from the radar
6-36 Kalman filter tracking of the balloon
6-37 Particle filter tracking of the balloon
6-38 BCoSaMP tracking of the balloon
6-39 Modified BCoSaMP tracking of the balloon
6-40 Multi-frame correspondence tracking of the balloon
6-41 Track of the balloon moving towards the radar
6-42 Kalman filter tracking of the balloon
6-43 Particle filter tracking of the balloon
6-44 BCoSaMP tracking of the balloon
6-45 Modified BCoSaMP tracking of the balloon
6-46 Multi-frame Correspondence tracking of the balloon
6-47 Number of targets detected
Chapter 1

Introduction

One of the major environmental issues of today is the increasing bird mortality rate due to increase in large scale installation of wind turbines for generation of green energy. There are 836 bird species in Migratory Bird Treaty Act of which 78 species are endangered and 14 threatened in U.S. There are also various other reasons for the increase in bird mortality rate such as cats, tall buildings, power transmission lines and cell towers. The threat due to wind turbines increases exponentially with the increasing need for energy [1]. The American Bird Conservancy has reported millions of bird deaths each year and the deaths due to wind turbines alone contributing to 440,000 per year. The deaths are caused due to collision or loss of habitat at the constructed site [2]. Due to these reasons, there have been restrictions to build wind turbines and the necessity to recognize the low risk sites before construction [3].

The potential wind farm sites generally are avoided if they happen to be stopover areas used by birds for migration. Appropriate measures are needed to prevent potential collision with wind turbines before construction. Thus it can be said that the increase in the need for construction of wind turbines has increased the growth of avian study [1]. Avian study at wind turbines project [4] site help in selecting low risk sites based on:
• Bird activity.
• Migration parameters such as passage rates, altitudes, direction, speed etc.
• Type of species and habitat.

1.1 Methods for Bird Monitoring:

Bird monitoring studies may be based on visual methods [5], acoustic signal recording [6], cameras [7], radio telemetry [8], radars [9] and multiple sensors [10]. An overview of these methods is provided in the following sections.

1.1.1 Visual Methods:

Direct observation is the oldest methods used for observing bird migrations [11]. A visual method of monitoring bird densities in an area is called the point count method. Birds are identified by visual observation or their song. A point count method includes counting the number of birds in an area of certain radius around the observer. A radius of 20 m is generally used. The point count method helps in differentiating the bird species and their abundance in an area. The procedure is repeated in the same area for a few days to improve the accuracy of observations [5]. Bird studies were also conducted by moon watching. In this method the observations were based on reflections of birds crossing the disk of the moon [12].

1.1.2 Acoustic Data:

Acoustic data in avian studies were used to identify the species. With the availability of high end recorders and software for identification, this became a popular method. There are various software tools available for identifying species based on the spectral characteristics. The acoustic data helps in providing the number of birds, call patterns and peak passage rates. A recording system is set up on the wind turbine to collect data. The
data consists of bird sounds, turbine noise and other environmental noise. A recognition system is capable of detecting bird songs and removing all other noise. If a bird is detected then there is a possibility of occurred collision [6]. Acoustic data is preprocessed to remove noise and feature extraction techniques are applied for bird detection. Features may be extracted using Mel-Frequency Cepstral Coefficients (MFCC) [13], wavelet transforms [14] and other techniques. More recently feature extraction methods such as Spectrogram-based Image Frequency Statistics (SIFS) and Mixed MFCC and SIFS (MMS) technique have been used. Extracted features are combined with various classification methods for identification of birds [15]. A commercially available bat recognition system such as Sonobat [16] and others [17] can be used for detection of bats.

1.1.3 Radio Telemetry:

Another method of tracking bird flight is by telemetry in which a device is attached to the bird. In this method a radio transmitter is attached to the bird and the signal is tracked to study the bird behavior. The radio transmitter consists of a high frequency transmitter, power supply, antenna and material for mounting. Nowadays Platform Terminal Transmitters (PTT) and Global Positioning System (GPS) transmitters which use satellites to relay signals are also used. These transmitters are highly efficient for monitoring birds. However birds have low survival rate when attached to such devices [8].

1.1.4 Cameras:

A series of webcams have been used to track birds. The cameras are initially tested to find limitations with respect to size, velocity and contrast of objects. The data obtained from the camera is processed with background subtraction, stereo vision and lens
distortion techniques for target identification [7]. IR cameras are widely used because the images are independent of lighting conditions and also provide the temperature information. In IR cameras, the targets are detected by applying background subtraction and connectivity of components helps in tracking. The background subtraction techniques used to detect moving objects are Running Average (RA) [18], Running Gaussian Average (RGA) [18] and Mixture of Gaussian (MOG) [18]. Thresholding method is applied to detect birds using an adaptive thresholding of Otsu method called EOtsu which is an extended version of Otsu method [19]. Then tracking is applied based on the connectivity of the object. The objects with same area are given same label numbers and morphology closure operator is used to create a track [20].

1.1.5 Radars:

Discovery of unwanted targets in radar data called as angels led to the evolution of the study of ornithology [9]. Radars were first used for wind energy related avian study during mid-1970s. Birds migrate mostly during the night where visual techniques cannot be applied for detection and hence radars are used. It is also useful when there is less visibility due to fog or clouds and also for detection over a wider range. Weather and surveillance radars were used over large areas however high resolution images cannot be obtained at wind sites and coverage area may not be available near the required wind site. Marine radars which can provide high resolution image can be very useful. Tracking radars are also used to track targets by locking the targets position. In the tracking mode, it gives information on bird flight. In surveillance mode, it uses angle tracking for multiple targets. Tracking radars are not widely available, expensive and requires operator training. On the other hand marine radars are commercially available, less
expensive and comparatively easy to operate. Therefore marine radars are widely used in avian study [9].

1.1.5.1 Weather Radars:

The weather radars can also be used to study the bird migration paths. Weather Surveillance Radars (WSR88D) were established in 1988 for meteorological purposes all across US. WSR88D is an S band radar and the maximum range coverage is 250 Km [21]. WSR88D data is widely available and can be exploited to extract biological activity. The signal processor provides three types of data which is the reflectivity, radial velocity and spectrum width. The reflectivity and radial velocity data is filtered to detect birds [22]. Low reflectivity in the image is removed and the radial velocity data is filtered based on the bird velocity which is greater than insects. The velocity of birds are also affected by winds that is if the bird is flying along the direction of wind then its velocity is greater than the speed of tailwind and vice versa. The final targets obtained after filtering are quantified to study bird densities and flight direction. Figure 1-1 shows the target identification system used with the weather radar data [23]. However, weather radars cannot provide a high resolution image at wind energy sites and hence marine radars were used [9].

![Figure 1-1. WSR88D data in identifying birds](image-url)
1.1.5.2 Marine Radars:

Marine radars are widely used in avian study due to their commercial availability, higher resolution, easy maintenance, dependable, less expensive and provides altitude information [9] [24].

The target detection in marine radars is difficult due to high amount of clutter and noise. The ground clutter can be reduced in the marine radar by using protection shield around the radar beam or by elevating the antenna mount [25]. To study the behavior of flight of birds it is important to implement tracking in radars. Various steps in processing marine radar data are shown in Figure 1-2. The efficiency of target detection depends on the accuracy of the tracking method used as it helps in removal of unwanted targets [26].

![Figure 1-2. Target detection using marine radar data](image)

1.2 Target Tracking:

Birds are tracked based on their detection on multiple sweeps of the radar to understand their flight behavior. Tracking also helps in identifying the bird migration paths. radR is an open source platform for studying biological targets [27]. radR is used for processing the marine radar data. Many algorithms have been implemented for tracking birds and are discussed in the following chapters.
1.3 Current Research:

In this research four tracker models have been developed in radR and their performances with different data sets are compared. Tracker models are Kalman filter, particle filter, BCoSaMP and modified BCoSaMP. Various chapters in this thesis are structured as:

Chapter 2: Radars in Avian Study

Chapter 3: Equipment specifications and radR software used for data processing.

Chapter 4: Current tracking models in radR.

Chapter 5: Kalman filters, particle filters and BCoSaMP and modified BCoSaMP are discussed.

Chapter 6: Simulation of tracker models.

Chapter 7: Conclusions and future work.
Chapter 2

Radars in Avian Study

2.1 Introduction

Radars are extensively used in the field of ornithology due to high detection ranges and ability to detect birds during low visibility conditions. Originally Doppler radars were used to detect the speed of bird flight. The tracking radars in the non-tracking mode can help in detecting the density and the altitude of a single target or a flock of birds. Later studies of tracking birds from reflectivity data were developed. Satellite tracking radars were also used for wide range target tracking. However, in recent times weather and marine radars are very popular for the study of bird migration patterns for quantification of their activity [28].

Radars have the advantage of surveying larger areas beyond the detection capabilities of visual techniques. They are used for identifying direction and speed of birds. However a radar cannot recognize bird species or distinguish small birds flying at closer vicinity from a large bird. Thus radars are used in studies where species identification is not required [29].

2.2 Marine Radars in Avian Study:

Marine radars are popular in avian study due to their commercial availability, low cost and higher resolution. Marine radars can be mounted on trailers to study the bird behavior
at required locations and clutter reduction screen is placed around the antenna. Shorter pulse lengths have better target discrimination while long pulse lengths have greater detection range. Depending on the range the pulse length can be selected. Simultaneous visual and radar interpretations can be performed for accurate target recognition. Visual observations are made with the help of binoculars and spotting scope. The target detection by the radar depends on the settings. The settings of the radar are adjusted depending on size of target such as large or smaller birds. Based on the sensitivity of the radar the targets are identified [30]. Initially visual monitoring of the radar data was done to identify targets and later on image analysis was used to analyze the collected radar data. The radar data was collected by a personal computer and later image analysis techniques were applied to analyze and quantify the bird information [29].

Data from the X-band marine radar with T bar antenna in the vertical mode can be combined with the WSR88D data for the same time period to see any correlation between two data sets [31]. The T bar antenna can be replaced by a parabolic antenna which can provide 3D information [32].

2.3 Methods of Reducing Ground Clutter:

Ground clutter in marine radars can be reduced by using a clutter reducing screen around the antenna. The screen is made of aluminum and is inserted on the lower end of the antenna at 90° angle. Another method of reducing clutter is by elevating the antenna by inserting a 50 mm wooden shim which increases the angle of elevation by 10°. Twist the waveguide for further increasing the elevation angle [32].

2.4 Analysis of Marine Radar Data:

Huansheng et al. has used the marine radar for target detection using filtering
techniques. The data obtained from the marine radar is the reflectivity information and bird detection is performed using filtering techniques. Filtering of radar data is implemented by applying background subtraction, median filtering, segmentation and morphology [26].

Background subtraction can be applied to remove stationary objects from an image. There are many background subtraction techniques such as running Gaussian average, temporal median filter, mixture of Gaussians, Kernel Density Estimation (KDE), sequential KD approximation, co-occurrence of image variations and Eigen backgrounds [18]. Principal Component Analysis (PCA) belongs to the Eigen background method and helps in identifying regular patterns in a data [33]. Random noise can be removed by applying noise removal techniques such as median filtering, segmentation and image morphology. Median filtering reduces noise while preserving edges as described by Ziv Yaniv [34]. Image segmentation and morphology are given by Huang and Wu in [35] and Wayne, Lin Wei-Cheng and Wu Ja-Ling in [36].

2.5 Previous Radar Studies:

Commercially available Mobile Avian Radar System (MARS) by Geo-Marine uses X-band marine radar with T-bar antenna operating in vertical and horizontal modes. The horizontal mode can provide range, direction and speed of the target. Radar in vertical mode can provide altitude [38].

The types of radars used in the field of ornithology are fan beamed radars and pencil beamed radars [38]. In 1998 Clemson University Radar Ornithology Laboratory (CUROL) developed a bird detecting system called BIRDRAD. This consists of FURUNO 2155BB radar. The T-bar antenna in vertical mode can only provide altitude
information of targets within the radar beam. Parabolic antenna was used for three dimensional tracking of the bird targets [32]. Walls has used two marine radars simultaneously. He has used X-band in vertical mode and S-band in the horizontal scanning mode to study bird movements [39]. A custom built converter was used to digitize the radar pulses and FORTRAN programs were used to obtain pulses that have recognized targets. The data was stored in an ASCII and processing was done using R 1.9.0 [40].

The radar system was connected to a personal computer and radar data was stored. Image analysis was then used to remove ground clutter and the birds were categorized based on the parameters. The clutter can also be removed using the anti-sea clutter which helps in detecting small targets. In vertical mode when the antenna points to the ground the energy transmission was disabled using a blind sector [41]. Since slow flying birds cover the radar viewing range in 30 seconds, therefore the area is scanned by the radar for every 30 seconds. The birds are classified based on the size using Small-Bird Equivalents (SBE) and also helps in determining the bird numbers [29]. Radar is calibrated to reduce sensitivity to remove clutter at closer ranges without affecting the detection of birds [38].

Bertram et al has used FURUNO FR810D (940 MHz, 10 KW, 2 m antenna) marine radar and has captured radar video using VHS video recorder. Image acquisition system from the Play Technologies Snappy Video Snapshot has been used. The images were converted to binary images and a histogram for the image was created. Bird activity was defined by subtracting the percentage of white pixels from the image when no birds were present. The bird numbers were estimated based on the area covered by birds by the area of a known bird [42]. Burger [43] has classified birds into different species based on the
speed of flight, path and size. Weber [32] has used track information of targets and has implemented classification algorithms.

DeTect Inc [44] [45] is developing software to determine wing beat frequency for identification of bird species. If the bird is tracked by the radar beam then the fluctuations in the target echo can be used to obtain the wing beat frequency. The wing beat frequencies help in identifying the species and differentiating birds from bats. Fixed vertical beam radar with beamwidth large enough to allow large as well as slow moving birds within the beam for a few seconds is used to measure its wing beat frequency.

Mabee et al. [46] removed insect contamination using air speed. Small targets within 500 m of the radar range and targets with a velocity of <6 m/second has been filtered. The airspeed is calculated using the formula given in [46]:

\[ V_a = \sqrt{V_g^2 + V_w^2 - 2V_gV_w \cos \theta} \]  

(2.1)

Where \( V_a \) is the airspeed

\( V_g \) is the groundspeed

\( V_w \) is the wind velocity

\( \theta \) is the angle between the target direction and wind vector direction

The target trails is turned on for 30 seconds which is just long enough to find out direction and speed. Post processing of data is done using SAS V.8 and passage rates are obtained. The average flight direction is given as [46]:

\[ \bar{x} = \frac{\sum_{i=1}^{n} \sin(\theta_i)}{n} \]  

(2.2)
\[
\bar{y} = \frac{1}{n} \sum_{i=1}^{n} \cos(\theta_i)
\]

Where \( \theta_i \) is the flight direction at \( i^{th} \) observation

\[
\bar{\mu} = \tan^{-1}\left(\frac{\bar{y}}{\bar{x}}\right)
\]

Dispersion is calculated as:

\[
r = (\bar{x}^2 + \bar{y}^2)^{1/2}
\]

Where \( r = 1 \), when the direction is the same for all observations and \( r = 0 \), when there is uniform distribution [41].

Marine radars have been used for bird detection as well as quantification by McFarlane et al. [47]. The bird densities are calculated based on the probability of the target being a bird based on distance between observer and target. Three assumptions used in this method are certainty of detecting objects at observation point, detecting objects at initial positions and exact measurements. The densities are quantified as birds per square kilometer [47]. The wind effects on bird migration are studied as it has greater influence on bird flights compared to other variables. The winds increase the ground speed of birds and also causes drift in the migration path as reported by Kerlinger et. al. [48].

Gauthreaux et al. [28] has equipped radar with a Geographical Positioning System (GPS) to locate the latitude and longitude of the target. The marine radar results are correlated with the WSR 88D data [28]. Gauthreaux et al. [49] have developed another method of validation by comparing vertically pointing fixed beam radar with the thermal camera data.
2.6 Summary:

Radars used in avian study and the extensive usage of marine radars in the field of ornithology were discussed. Various techniques implemented to reduce clutter and the data processing methods for bird identification in the marine radar data is also discussed. The 3-D information of targets is obtained using a parabolic antenna and the t-bar antenna can be used in the vertical mode to extract the altitude information of birds. In the following chapter the experimental setup to collect data using marine radar and the data processing software radR is discussed in detail. The processing of data in radR to extract bird targets relies on various filtering parameters.
Chapter 3

Marine Radar and Radar Data Processing Software

3.1 Introduction

Marine radar is used for bird observation and quantification of their activity. The radar data is collected using a digitizing card XIR3000B from Russell Technologies as the slave display. The data collected is processed using open source radR software for target detection and tracking. The experimental setup of the entire system is shown in Figure 3-1. The marine radar is run all through the night from evening civil twilight to morning civil twilight.

![Experimental setup diagram](image)

**Figure 3-1.** Experimental setup

The marine radar is connected to the digitizing card and as the antenna rotates the signal is transmitted to the digitizer. The received signal is digitized and transmitted to a
laptop/PC through the USB and data is collected using radar sample application provided by SDK of the XIR3000 card. radR is used for real time processing or post processing of the saved files.

3.2 Marine Radar:

Furuno 1500 Mark 3 (X band) marine radar is used in this project for bird detection. The radar and the antennas used are shown in Figures 3-2 and 3-3.

![Radar trailer and the slotted array antenna at Bowling Green](image)

**Figure 3-2.** Radar trailer and the slotted array antenna at Bowling Green

Advantages of using marine radar in avian study are discussed in Chapter 2. Some of the features of Furuno are automatic tracking aid, target trail, target alarms, navigation plotting, clutter sweep, adjustable gain and interference rejector. This radar is easy to operate and extensive training is not required.
There are different models of Furuno marine radars available which are given in Table 3.1. These models are classified based on the input power and the radiator model.

**Table 3.1: Models of Furuno Mark 3**

<table>
<thead>
<tr>
<th>Radar Model</th>
<th>Power (KW)</th>
<th>Radiator Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR- 1505 Mark 3</td>
<td>6</td>
<td>XN12AF,XN20AF</td>
</tr>
<tr>
<td>FR- 1510 Mark 3</td>
<td>12</td>
<td>XN12AF,XN20AF</td>
</tr>
<tr>
<td>FR- 1525 Mark 3</td>
<td>25</td>
<td>XN20AF</td>
</tr>
</tbody>
</table>

Radiators are available in various lengths and appropriate length should be selected based on application. Specifications for different types of slotted waveguide are given in Table 3.2.
Table 3.2: Types of radiator

<table>
<thead>
<tr>
<th>Radiator Type</th>
<th>XN12AF</th>
<th>XN20AF</th>
<th>XN24AF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length (Ft.)</td>
<td>4</td>
<td>6.5</td>
<td>8</td>
</tr>
<tr>
<td>Beamwidth (H)</td>
<td>1.8°</td>
<td>1.23°</td>
<td>0.95°</td>
</tr>
<tr>
<td>Beamwidth (V)</td>
<td></td>
<td>20°</td>
<td></td>
</tr>
<tr>
<td>Sidelobe ±10°</td>
<td>-28 dB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polarization</td>
<td>Horizontal</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Specifications of Furuno Mark 3 radar are given Table 3.3. It has an Automatic Tracking Aid (ATA) which can plot up to 20 targets and the Electronic Plotting Aid (EPA) allows plotting tracks for 10 targets. The radar map provides the geographical information of the area.

The antennas used were slotted array and the parabolic antenna. The slotted array antennas are widely used in navigation system due to emission of linear polarized waves and greater efficiency.

Table 3.3: Specifications of the radar

<table>
<thead>
<tr>
<th>Specifications of Marine Radar</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Frequency</strong></td>
</tr>
<tr>
<td><strong>IF</strong></td>
</tr>
<tr>
<td><strong>Noise Figure</strong></td>
</tr>
<tr>
<td><strong>Range Accuracy</strong></td>
</tr>
<tr>
<td><strong>Bearing Accuracy</strong></td>
</tr>
<tr>
<td><strong>EPA</strong></td>
</tr>
<tr>
<td><strong>ATA</strong></td>
</tr>
</tbody>
</table>

During the migratory seasons the data is collected using the acquired marine radar. The data collection is divided into three time slots which are evening civil twilight, morning civil twilight and night time observations. The slotted array antenna is used during the morning and evening civil twilights with the antenna rotation around the horizontal axis, it is the time when the birds ascend and descend. The horizontal beam
width is 1.23° and provides greater target discrimination, detectability and higher resolution. This mode helps in detecting the height of targets and hence the ascend and descend of targets can be detected accurately. The night time observations are performed using the parabolic antenna which covers wider area and target are detected along with heights and angle of arrival [50] [51].

3.3 Digitizing Card:

The digitizing card is used in the slave mode and the files are saved in the form of .REC files in the form of image frames. The block diagram of XIR3000C is shown in Figure 3-4.

![Block Diagram of XIR3000C](image)

**Figure 3-4.** Block diagram of XIR3000C

In the slave mode the digitizing card requires four input signals from the transceiver which are azimuth, heading marker, trigger and video signals. The XIR3000 has:

- Hi-speed USB bus
- 100 MHz sample rate
- Compatible with all PCs and transceivers
- Software Development Kit (SDK) for customization
- Automatic Radar Plotting Aid (ARPA) tracking
- Ethernet with TCP/IP

The Russell Technologies Inc. Software Development Kit (RTI SDK) allows flexibility in scale, image and display options. The SDK has following features:

- Header, RTI DLL and library files
- Radar sample application which allows viewing and saving of radar images
- MS VC++ sample files
- Clutter and interference suppression
- Antenna Control Module (ACM)

The radar sample helps in displaying, saving and playback of files in the form of .REC files. The .REC files can be processed using radR software developed for detecting biological targets [52]. In this project XIR3000B digitizing card is used in slave mode and has an input power of 12V. It is connected to the power supply at the transceiver and the four input signals are initialized. The four input signals from the transceiver to the digitizing card are:

- Heading Marker: This signal appears once every sweep of the antenna and shows the correct alignment of the antenna.
- Trigger: This signal gives the time for the transceiver to transmit which helps the XIR3000 card to start digitization process of the received signal.
- Bearing: This gives the direction of the target.
- Video: This is the data received by the digitizing card. Figure 3-5 shows the heading marker and the bearing in radar data.
If these signals are not received correctly then ‘missing’ appears next to the signal in the radar sample software [53].

The connections of the digitizing card in the slave mode are shown in Figure 3-6. The input of the digitizing card is connected to the transceiver and the output is an USB connection to the laptop or PC with radar sample application.

The snapshot of radar sample application is given in Figure 3-7 and the parameters are shown in Figure 3-8 [52].
Figure 3-7. Radar data in radar sample application
Figure 3-8. Snapshot of parameters in radar sample application
3.4 radR:

radR is an open source software used for studying biological targets. The software was developed by Taylor et al. It is easily downloadable and John Brzustowski of radR development team happily provides answers to questions. radR is used for real time target recognition and tracking developed using R language works with both Windows and Linux operating systems [27]. This was especially designed for radar data processing. R language is an extension of S language developed by Bell Laboratories [27]. The programs are written in R and C language and interface is coded using tcl/tk by which each scan of the radar data is processed and possible targets (blips) are obtained by noise removal. radR has many plugins of which the ones used in the project are plugin for reading the .REC files, antenna orientation, noise removal, blip processing, saving blips and target tracking.

radR is not only used for post processing and saving files but also for real time tracking. The radar data is processed in radR by setting the antenna parameters and selecting the XIR3000 plugin for reading the data from the digitizing card. The data can be saved by radR as a blip movie which contains only blips information. The XIR3000 plugin was developed for the XIR3000 digitizing card by Russell Technologies SDK and similar plugins for acquiring data using other digitizing cards can also be developed [27].

radR is very useful for saving data in blip movie format which contains only the required target information and the unwanted noise is removed thus saving memory in the case of large data files. The steps for processing data in radR are shown in Figure 3-9 which shows the key steps involved in processing files that is parameter initialization, plugin selections and saving output files.
3.4.1 Radar Data Processing:

Initial steps involve running the batch file, antenna selection and XIR3000 reader plugin. The data is read by the .REC file reader plugin called XIR3000. Initially the radR application is launched and then the XIR3000 plugin is enabled which helps in batch processing of the files. The antenna is selected based on the type of antenna, the beam widths, angle above the ground and axis of rotation. Select the blip processing window and set parameters for noise removal such as selecting the number of learning scans, noise threshold, hot score and cold score, cell size and filtering based on a logical expressing, minimum blip area, maximum blip area, minimum number of samples and maximum number of samples, this process is called blip extraction. Adjacent hot cells are called patches and those that satisfy the criteria of blips are saved as blips.
Learning scans are the number of scan used in background subtraction technique. Hot score and cold score are thresholds which differentiate bird targets from the background noise. The blip area such as minimum and maximum area in meters square is defined. Old stats weighing is a parameter that defines weights between past and present frame’s background mean and mean deviances. Samples are the echo strength digitized sequence. It is given in terms of d bits per sample giving a sample range from 0 to $2^{d-1}$. Samples per cell are the number of rows in one stats cell and pulses per cell are the number of columns of one stats cell. The angular span is the number of columns along the angular area and radial span is the number of rows along the range. Blips maximum and minimum area in terms of square meters can be defined to remove large targets. Further filtering of blips can be specified by a filtering criteria based on formulas defining the targets. Cells that satisfy all the filtering criteria specified forms a blip. Figure 3-10 shows blip processing given by radR.
Figure 3-10. Blip Processing

Table 3.4 gives parameters and recommended values set for data processing.
### Table 3.4: Blip processing parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Min. Range</th>
<th>Max. Range</th>
<th>Recommended Values</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise cutoff</td>
<td>0</td>
<td>65535</td>
<td>0</td>
<td>Min. value to retain small targets.</td>
</tr>
<tr>
<td>Learning scans</td>
<td>0</td>
<td>100</td>
<td>50</td>
<td>Average value for background subtraction technique.</td>
</tr>
<tr>
<td>Old stats weighting</td>
<td>0</td>
<td>1</td>
<td>0.95</td>
<td>Standard value and changes does not affect data significantly.</td>
</tr>
<tr>
<td>Hot score</td>
<td>0</td>
<td>10</td>
<td>3.7</td>
<td>The hot score and cold score value should be the same. Value selected to preserve valid data based on range.</td>
</tr>
<tr>
<td>Cold score</td>
<td>0</td>
<td>10</td>
<td>2.5</td>
<td></td>
</tr>
<tr>
<td>Samples per cell</td>
<td>1</td>
<td>128</td>
<td>5</td>
<td>The size of one stats cell defined based on the size of the smallest target.</td>
</tr>
<tr>
<td>Pulses per cell</td>
<td>1</td>
<td>128</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Min. blip samples</td>
<td>2</td>
<td>50000</td>
<td>40</td>
<td>Min. and max. samples define the patch size. The patches which satisfy the filtering criteria forms blips and are set to maximum ranges.</td>
</tr>
<tr>
<td>Max. blip samples</td>
<td>-1</td>
<td>50000</td>
<td>5000</td>
<td></td>
</tr>
<tr>
<td>Max. blip area</td>
<td>0</td>
<td>50000</td>
<td>5000</td>
<td>Filtering criteria for the patch based on area.</td>
</tr>
<tr>
<td>Min. blip area</td>
<td>0</td>
<td>50000</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Min. angular span</td>
<td>0</td>
<td>1024</td>
<td>2</td>
<td>Defining patch size along the angular region and maximum value is set by selecting value -1.</td>
</tr>
<tr>
<td>Max. angular span</td>
<td>-1</td>
<td>1024</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>Min. radial span</td>
<td>0</td>
<td>1024</td>
<td>2</td>
<td>Defining patch size along the range and maximum value is set by selecting value -1.</td>
</tr>
<tr>
<td>Max. radial span</td>
<td>-1</td>
<td>1024</td>
<td>-1</td>
<td></td>
</tr>
</tbody>
</table>

#### 3.4.2 Blip Processing:

Blip processing deals with noise removal and target identification. The noise cut off is set and is always good to set it to a minimum value to make sure no birds are removed. Learning scans are set by default to 15 scans which can be changed. Learning scans represent the number of scan of radar data used in background subtraction technique. The background subtraction method used is the standard mean and mean deviation of the
learning scans. Further filtering can be performed by selecting appropriate values for hot score, cold score, area of the target and also with logical expressions.

Blips exceeding the user defined z-score value (threshold) are marked as hot. Adjacent hot cells are combined and those that meet the filtering criteria are called blips. Display options can be set to create blip trails of targets and colors for the display.

Blips are obtained by removing noise using the mean and mean deviation calculated by the specified number of learning scans, then a z-score is set up which is a threshold above which the cells are combined to form blips. Figure 3-11 and 3-12 shows the original radar data and the filtered data using radR.

![Figure 3-11. Sample radar data in radR](image-url)
Figure 3-12. Sample radar data filtered using blip processing in radR

A console window is present in radR which helps in running commands while radR is running or to run scripts on radR. The processed data can be saved in a movie format or in a CSV file using save blips plugin. In Table 3.5, a list of radR plugins and their applications are described [54].

Table 3.5: Plugins in radR

<table>
<thead>
<tr>
<th>Plugins</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>antenna</td>
<td>Antenna selection</td>
</tr>
<tr>
<td>seascan</td>
<td>Obtain data from Rutter Inc. Sigma S6 digitizing card</td>
</tr>
<tr>
<td>seascanarch</td>
<td>Reads data saved by Rutter seascan software</td>
</tr>
<tr>
<td>xir3000</td>
<td>Obtain data from Russell Technologies Inc. XIR3000 USB video processor</td>
</tr>
<tr>
<td>xir3000arch</td>
<td>Reads data saved by RTI software</td>
</tr>
<tr>
<td>video</td>
<td>Reads video data</td>
</tr>
<tr>
<td>declutter</td>
<td>Noise removal</td>
</tr>
<tr>
<td>tracker</td>
<td>Creating tracks of targets</td>
</tr>
<tr>
<td>zone</td>
<td>Excludes data within the defined region</td>
</tr>
<tr>
<td>genblips</td>
<td>Generation of artificial blips</td>
</tr>
<tr>
<td>saveblips</td>
<td>Save blip information in .blips file</td>
</tr>
<tr>
<td>blipmovie</td>
<td>Save data in blip movie format</td>
</tr>
</tbody>
</table>
Figure 3-13 shows a snapshot of enabling a plugin in radR.

**Figure 3-13.** Enabling plugin in radR

### 3.5 Summary:

The marine radar and the digitizing card XIR3000B used in the project are discussed. The radar data is processed using radR software. Various plugins in radR and the importance of blip processing parameters for accurate target identification is given in detail. The following chapter discusses tracking and the current tracker models in radR.
Chapter 4

Tracker Models in radR

4.1 Introduction

Tracking provides the current and estimated flight path of any target of interest. Tracking targets using radars may be very challenging due to the randomness of the data and presence of large amount of noise commonly known as clutter. Tracking can also be considered as filtering operation for removal of unwanted targets. Due to the randomness in the radar data, tracking algorithms play a vital role in target detection and clutter removal. This work focusses on analysis of the behavior of birds toward man-made structures. Tracking will be used to determine bird migration patterns. Figure 4-1 shows tracking of birds using radar.
Figure 4-1. Tracking flight paths of birds using radar

Figure 4-2 gives examples of bird tracks and it can be seen that there are no regular patterns for tracks.

Figure 4-2. Types of bird tracks

4.2 Tracking Techniques:

There are many methods used for tracking targets in radar data. Tracking algorithms are based on estimation methods in which a data with random noise is used to predict the
position of the target in the next time interval. Tracking methods use initial position, velocity and acceleration and predict future position of the target. The efficiency of the tracking method depends on the accuracy of initial conditions and the assumptions made during their design. There are many methods that have been developed for different systems, sources and target conditions. Some of the tracking techniques are maximum likelihood, Bayes estimators, maximum a posteriori, minimum variance unbiased estimator, Kalman filter, particle filter and Weiner filter [55].

4.3 Tracker Models in radR:

Tracking radars operate on the principle of locking the target's position and moving along with it. It uses track while scan approach by which multiple targets are tracked by search and scan technique [56]. Marine radars operate in surveillance mode and requires target tracking algorithm for generation of target tracks. Figure 4-3 shows steps involved in target tracking.

![Target Tracking Diagram](image)

**Figure 4-3.** Target Tracking
radR is used for processing radar data which is an open source software available for recording and processing radar data. It has algorithms for recording, detecting, tracking and saving target information. The tracker plugins currently in radR are the nearest neighbor model and the multi-frame model. The nearest neighbor tracking is based on minimum distance and multi-frame tracking is based on maximum gain. Figure 4-4 shows the two current tracker models in radR [54].

![Tracker Models Diagram](image)

**Figure 4-4.** Tracker models in radR

Radar data processing via radR utilizes number of plugins. First of all the tracker plugin is enabled and the appropriate tracker model is selected. The velocity of targets to be tracked and the number of blips required to form a track can be selected in order to remove unwanted targets. Run the data using the player and tracks are created as the files are processed. The data is saved in the tracks.csv files with all the blip information such as time stamp, track number, scan number, blip number, x-coordinates, y-coordinates, z-coordinates, intensity, area, perimeter, radial span, angular span and number samples of each blip.

**4.4 Tracker Model:**

The tracker model reads the previous frames as old data points. Current frame data is stored as blips. The tracker model then uses current blips and performs matching with previously stored blips. Tracks with matched blips are stored in the gain function. Blips
without any match start their own track. The gain function is read by the tracker plugin and tracks are plotted on the radR plot window. Figure 4-5 shows steps involved in radR.

![Diagram of tracking in radR]

**Figure 4-5.** Tracking in radR

The tracker model consists of various functions as shown in Table 4.1. Each function performs a unique task. The update function updates each track by adding blips to the existing track or starting a new track. The tracker algorithm is implemented in the update function where matched blips are selected and are given in the form of gain function which stores the column index of the matched blips. Get menu function obtains the user input for controlling the tracking procedure based on the minimum number of blips required to form a track and maximum speed. Set blip fresh time selects the expiry time of the track and if the controls are changed while running the data then the tracks are refreshed. Other functions are used to define the maximum speed of the track, select, deselect, load and unload a plugin. Figure 4-6 shows the layout of the tracker model file in radR.
Table 4.1: Functions in tracker model in radR

<table>
<thead>
<tr>
<th>Functions</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update</td>
<td>Tracking algorithm identifies best matched blip for each track.</td>
</tr>
<tr>
<td>Get menu</td>
<td>It is used to control parameters to create track dynamically.</td>
</tr>
<tr>
<td>Select</td>
<td>Select function defines the onset of a track.</td>
</tr>
<tr>
<td>Deselect</td>
<td>Deselect function defines the expiry time for a track.</td>
</tr>
<tr>
<td>Load</td>
<td>Load is used to enable the tracker model.</td>
</tr>
<tr>
<td>Unload</td>
<td>Unload is used to disable a tracker model.</td>
</tr>
<tr>
<td>Set blip fresh time</td>
<td>Defines the time for which a blip is retained in the frames.</td>
</tr>
<tr>
<td>Set track stale time</td>
<td>Defines the time for which a track is active.</td>
</tr>
</tbody>
</table>

Figure 4-6. Algorithm for tracker model

4.4.1 Nearest Neighbor Model:

The nearest neighbor plugin creates tracks based on the minimum distance between blips of the current frame to the ones on the previous frame [57]. Filtering is performed based on velocity, turning angle and the blip size. Distance is calculated between blips in the current and previous frames. All possible combination of distances is stored in a
matrix. Maximum distance is computed for each blip stored in a row of the distance
class. Gain is calculated for each blip using the following relation:
\[
\text{Gain} = 100 + \max(\text{distance}) - \text{distance of each blip}
\]  

This gain function stores the best match for each blip for creating tracks. The blips
which are not part of a track are extended to the next scan based on a defined time to
overcome missed detections. The tracker uses the algorithm from Stanford graph base
package of Knuth [58] [59]. Figure 4-7 shows tracks created using the minimum distance
criterion.

![Figure 4-7. Nearest Neighbor Model with track 1 created using the minimum distance](image)

Figures 4-8 and 4-9 are the snapshots of sample data and tracks created by the nearest
neighbor tracker plugin.
Figure 4-8. Sample data in radR

Figure 4-9. Snapshot of tracks created by Nearest Neighbor Plugin in radR

User selectable parameters in the nearest neighbor model in radR are given in Table 4.2.
Table 4.2: Parameters in the nearest neighbor model

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min. Range</th>
<th>Max. Range</th>
<th>Recommended Values</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>How long a blip is retained as a possible track starter</td>
<td>1</td>
<td>3600</td>
<td>4</td>
<td>Time for retaining blips to find matches in seconds.</td>
</tr>
<tr>
<td>How long track stays active after last blip</td>
<td>1</td>
<td>3600</td>
<td>10</td>
<td>Time to preserve a track in seconds.</td>
</tr>
<tr>
<td>Maximum turning rate (degrees per second)</td>
<td>0</td>
<td>180</td>
<td>20</td>
<td>Turning rate is set to remove fast maneuvering targets.</td>
</tr>
<tr>
<td>Maximum rate of change of blip area (percent/s)</td>
<td>0</td>
<td>300</td>
<td>150</td>
<td>Rate of change of blip area that are added to tracks.</td>
</tr>
<tr>
<td>Maximum rate of change of blip intensity (percent/s)</td>
<td>0</td>
<td>300</td>
<td>150</td>
<td>Rate of change of blip intensity that are added to tracks.</td>
</tr>
</tbody>
</table>

4.4.2 Multi Frame Correspondence Model:

This is a non-iterative greedy algorithm which matches targets based on a gain function. The blips with maximum gain are added to the tracks [60]. Assume that the length of the track to be T for m-points. In each track the first point has no backward correspondence and the last point has no forward correspondence. Set of tracks for m-points are given as:

\[ A = \{ T_1, T_2, \ldots, T_m \} \]  

(4.2)

Where A is a set of tracks for m-points in the frame F

\( T_i \) is the track for each point

If \( x_j^k \in T_i \), then it is said \( x_j^k \) is a response of a sensor \( Z_i \) or if \( \forall x_j^k \in T_i \) then \( x_j^k \) is due to sensor noise. \( D = (V, E) \) represents edge weighted graph where V is the vertices and E is the edge of digraph D. In this graph, a directed path P is the number of connected vertices of length k. Vertex disjoint path is denoted by c and is a path of
digraph D such that any path of c contains all the vertices of D and c does not have a common vertex for any two paths. \( W(c) \) represents weights of all edges in c. The path with the maximum weight is selected as a match or gain function. Figure 4-10 shows an example of digraph. In this example there are five vertices \( V = \{a, b, c, d, e\} \) and assume each edge with weight 1. A path for the graph from vertex a to c is given as \( P = \{a, b, c\} \). Consider the vertex c and the different paths to c from four vertices are \( P_a = \{a, b, c\} \), \( P_b = \{b, c\} \), \( P_d = \{d, e, a, b, c\} \) or \( \{d, e, b, c\} \) and \( P_e = \{e, b, c\} \). For each path cover, weights are calculated and the path with maximum weight is selected.

![Figure 4-10. Digraph with five vertices](image)

Matching is performed corresponding to the maximum weight. Gain contains weights of the digraph. Greedy algorithm is used to find the maximum weight path assuming points in frame \( F \) are available up to time \( t_{k-1} \) and gets updated as more frames are available. Thus for every additional frame, tracks are extended to the new frame.

Algorithmic steps for multi-frame correspondence method are:

1. Step 1: Assume two frames, let \( r = 2 \)

2. For each additional frame \( F_i \)

3. Step 2: Create extension digraph D for the frames.

4. Step 3: Compute weights of edges using gain function

5. Step 4: For graph D calculate the maximum path.

6. Step 5: Replace false hypotheses
Step 6: $F_l = F_k$ backtracking is performed

Step 7: Increment $r$

End for loop

False hypothesis can be replaced using two methods namely false hypothesis replacement and non-recursive false hypothesis replacement. False hypotheses replacement deletes the false hypothesis from graph and the graph is checked again for matches between all frames. The non-recursive false hypothesis replacement deletes the false hypothesis and two-frame correspondence is solved. False hypothesis is replaced if the gain is above the threshold value and its value is set to one otherwise it set to zero.

Gain function is given by the distance between actual positions to the predicted position. It helps in obtaining matches close to the predicted value which is similar to the nearest neighbor however the direction of motion is not considered which gives rise to irregular paths [60]. The sample data and snapshot of multi-frame correspondence tracking in radR are shown in Figures 4-11 and 4-12. User selectable parameters in the multi-frame correspondence model in radR are given in Table 4.3.
**Figure 4-11.** Sample Data in radR

**Figure 4-12.** Snapshot of tracks created by multi-frame correspondence plugin in radR
Table 4.3: Parameters in multi-frame correspondence model

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min. Range</th>
<th>Max. Range</th>
<th>Recommended Values</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of scans to backtrack</td>
<td>2</td>
<td>100</td>
<td>2</td>
<td>For each additional scan the tracks can be corrected by backtracking.</td>
</tr>
<tr>
<td>Weight of directional coherence Vs. proximity to prediction</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
<td>Parameters is set to 0.5 as values &lt;0.5 increases false targets and &gt;0.5 has low detection rates.</td>
</tr>
<tr>
<td>Minimum gain of blips to be a part of a track</td>
<td>-150</td>
<td>150</td>
<td>10</td>
<td>A minimum gain value above which the blips are added to tracks.</td>
</tr>
<tr>
<td>Penalty for blips missing in tracks</td>
<td>0</td>
<td>1</td>
<td>0.001</td>
<td>This parameter is not used in the code and does not affect data.</td>
</tr>
</tbody>
</table>

Tracks are stored in tracks.csv file with all the blip information. Parameters stored in this file are given in Table 4.4.

Table 4.4: Parameters in tracks.csv file

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>scan no.</td>
<td>The scan number of the blip.</td>
</tr>
<tr>
<td>track no.</td>
<td>Each track is assigned a number.</td>
</tr>
<tr>
<td>date</td>
<td>Date of the data collected.</td>
</tr>
<tr>
<td>time</td>
<td>Time of occurrence of blip.</td>
</tr>
<tr>
<td>range</td>
<td>Range of target in meters.</td>
</tr>
<tr>
<td>x, y, z coordinate</td>
<td>x, y, z position of target in meters.</td>
</tr>
<tr>
<td>ns</td>
<td>Number of samples for each target.</td>
</tr>
<tr>
<td>area</td>
<td>Area of blip in square meters.</td>
</tr>
<tr>
<td>int</td>
<td>Intensity of blips.</td>
</tr>
<tr>
<td>max</td>
<td>Maximum intensity of the blip.</td>
</tr>
<tr>
<td>aspan</td>
<td>Number of rows of stats cell along angular region</td>
</tr>
<tr>
<td>rspan</td>
<td>Number of columns of stats cell along range.</td>
</tr>
<tr>
<td>perim</td>
<td>Perimeter of the blip.</td>
</tr>
</tbody>
</table>

A sample of tracks.csv file is shown in Table 4.5.
Table 4.5: Sample data in tracks.csv file

<table>
<thead>
<tr>
<th>scan no.</th>
<th>track no.</th>
<th>blip no.</th>
<th>date</th>
<th>time</th>
<th>time stamp</th>
<th>range</th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>ns</th>
<th>area</th>
<th>int</th>
<th>max</th>
<th>aspan</th>
<th>rspan</th>
<th>perim</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>110</td>
<td>107</td>
<td>3/9/2012</td>
<td>3:31:40</td>
<td>1.33E+09</td>
<td>717</td>
<td>481</td>
<td>-499</td>
<td>186</td>
<td>81</td>
<td>664</td>
<td>0.122</td>
<td>0.122</td>
<td>9</td>
<td>9</td>
<td>161</td>
</tr>
<tr>
<td>13</td>
<td>110</td>
<td>136</td>
<td>3/9/2012</td>
<td>3:31:45</td>
<td>1.33E+09</td>
<td>667</td>
<td>493</td>
<td>-415</td>
<td>173</td>
<td>81</td>
<td>548</td>
<td>0.049</td>
<td>0.049</td>
<td>9</td>
<td>9</td>
<td>156</td>
</tr>
<tr>
<td>15</td>
<td>110</td>
<td>163</td>
<td>3/9/2012</td>
<td>3:31:51</td>
<td>1.33E+09</td>
<td>712</td>
<td>495</td>
<td>-478</td>
<td>184</td>
<td>81</td>
<td>652</td>
<td>0.22</td>
<td>0.22</td>
<td>9</td>
<td>9</td>
<td>161</td>
</tr>
<tr>
<td>18</td>
<td>110</td>
<td>184</td>
<td>3/9/2012</td>
<td>3:31:58</td>
<td>1.33E+09</td>
<td>587</td>
<td>371</td>
<td>-480</td>
<td>152</td>
<td>81</td>
<td>662</td>
<td>0.098</td>
<td>0.098</td>
<td>9</td>
<td>9</td>
<td>147</td>
</tr>
<tr>
<td>19</td>
<td>110</td>
<td>196</td>
<td>3/9/2012</td>
<td>3:32:01</td>
<td>1.33E+09</td>
<td>542</td>
<td>430</td>
<td>-475</td>
<td>140</td>
<td>81</td>
<td>558</td>
<td>0.073</td>
<td>0.073</td>
<td>9</td>
<td>9</td>
<td>143</td>
</tr>
</tbody>
</table>
4.5 Summary:

Current tracker models available in radR have been described. The multi-frame correspondence tracker has less false alarm rates compared to the nearest neighbor model. Many tracking algorithms have been implemented on radar data for efficient target tracking and some of the methods are described in the following chapter. These tracking algorithms are also called estimation techniques which use previous state data with noise to predict the position of target in the current state.
Chapter 5

Tracking Algorithms

5.1 Introduction

Radar target tracking is performed by obtaining the position of the target by plan position indicator and estimating the target positions in the consecutive scans. This method is employed for single target tracking in tracking radars. Multi target tracking is performed by estimators such as $\alpha$-$\beta$ filter by predicting the targets position in the current state [61]. Tracking algorithms are implemented using range, azimuth, radial velocities and angular velocities of targets. This data uses gating statistical distance method to create threshold for detection [62]. Angle tracking of targets is done by predicting the error covariance matrix and based on the prediction the radar beam is shifted to the position which has the highest probability of detection [63].

Acceleration influences the target path and is frequently not considered as a reliable parameter for tracking because it is highly nonlinear. However a technique called centripetal acceleration along with radial velocity is used for tracking. Tracking based on acceleration increases the performance of the estimation and provides better results with just one sensor [64]. Noise is a very important factor that affects the radar data. It increases false alarm rates and affects target identification. The effects of noise cannot be
accurately initialized in tracking algorithms as it is a random value for any system. Thermal noise is present due to random motion of electrons in a system. Noise can also come from different modules. A low noise amplifier is used to reduce the noise. These logarithmic amplifiers may also have a constant noise. Parametric amplifiers are also used to reduce noise. However in marine radar the use of these amplifiers does not make a significant difference in noise level.

Bandwidth affects noise power and it is known that as the bandwidth decreases noise power $P_n$ reduces. Target detection rate is less at low bandwidths as shorter pulses in the radar are used for target discrimination and high resolution of data. Noises in the environment occurs due to precipitation, man-made devices, sky noise, solar noise and other services that are in the operational bandwidth of the radar.

Data that are not the targets of interest can be termed as clutter and can be removed by observing consecutive scans. Most of the clutter is non-moving and can be removed by background subtraction techniques. Polarization helps in reducing rain clutter. Near water clutter occurs due to multipath reflection from sea waves and is called sea clutter. This clutter slightly relies on polarization. Short range ringing clutter occurs due to mismatched feeders, they reflect a part of the transmitter pulse and gives rise to high false alarm rates. Feeders of longer lengths have been used to reduce short range ringing clutter however; they mask the actual targets and affect the system’s performance in the long run.

Interference occurs due to various devices operating in the same radar frequency band. The sources may be internal or external. Significant increase in the signal power reduces the effect of the interference noise on the signal. Signal processing is used such
as, moving target indicators, multiple sensor techniques and track before detect are some of the methods used to increase the accuracy of detection and decrease the noise level in data [65].

The efficiency of tracking depends on factors such as target parameters (position, velocity and acceleration), target motion, algorithm selection and initiation of algorithm design. Based on these factors many tracking methods have been developed. Figure 5-1 shows the factors influencing target tracking in a radar data.

![Figure 5-1. Factors influencing tracking](image)

### 5.2 Literature Review of Target Tracking Algorithms

Tracking can also incorporate targets crossing each other using the range time plots of the radar image. Tracking is based on the range of targets with Monte Carlo simulations
combined with radial velocity. The target is tracked for longer times and provides higher
detection rates compared to tracking with only range [66].

Li et al used mixed coordinates and applied a pseudo measurement approach to find
the pseudo linear representation of nonlinear functions, however due to its low
performance it is replaced by difference based linearized models [67]. Bing-Fei Wu and
Jhy-Hong Juang discussed a design for target identification. First the track-maintainer
makes sure only one kind of target is selected at a particular time. Target tracker
identifies the target type based on other attributes such as speed and motion [68]. Particle
and Fuzzy logic particle filter has also been used [69].

Tracking using Interval Based Approach (TIBA) is another tracking method which
can be used in nonlinear tracking as particle filters are computationally expensive. This
method is dependent on the transmitter and receiver positions and by inserting Set
Inverter via Interval Analysis (SIVIA), TIBA performance is increased compared to all
the existing tracking algorithms. However, this method has a disadvantage of position
dependency of the transceivers [70].

Tracking targets that maneuver very fast in the presence of noise is performed using
gating techniques such as centralized gating and model based gating. The gating
techniques are improved as Model Probability Weighted Gating and two stage Model
Probability Weighted Gating are introduced whose performance is higher in case of
maneuvering targets and in the presence of clutter when compared to the existing gating
techniques [94]. Methods like variable structure multiple model particle filter and Elman
nearest neighbor improves accuracy compared to the standard filter designs and has
reliable prediction [95]. The mean square error for the predicted value can be analyzed
using Cramer-Rao Lower Bound (CRLB). A Boolean-convex optimization is combined in the scheduling algorithm to minimize the CRLB. In Multiple Input Multiple Output (MIMO) radar this method of combining sensor scheduling and waveform design algorithm provides best tracking [71].

Target tracking and recognition are considered as two different procedures but can be combined by using the orientation of the target. The orientation and the path of targets are related and particle filter can be used for the joint estimation [72]. In the case of tracking objects in the presence of noise there are higher possibilities of false alarm rates. In such conditions the object is split into several blobs and combined. In the presence of a lot of noise there could be a few missed or unwanted detections present with the valid target. Multiple hypothesis represents the probability density of tracker as a number of modes using piecewise Gaussians of the neighborhood and can be used to expand the detection space. This method effectively tracks targets in the presence of noise [73].

Knowledge based tracking systems are used to reduce false alarms and better tracking in the presence of multiple targets to obtain our targets of interests based on the characteristics. This method has been combined with Extended Kalman Filter (EKF) for nonlinear tracking due to low computation time and Interacting Multiple Model (IMM) for maneuvering targets. Knowledge based algorithm helps reduce the false alarm rate [74].

Multiple hypothesis tracking is also widely used tracking algorithm in which the number of possible solutions is based on the likelihood function which can be modified as a user defined value multiplied by the likelihood function. This modified method works well in the presence of false targets as well as missed values [75]. Residual-Mean
Interacting Multiple Model (RMIMM) is an improved technique of Identity Management algorithm. It uses residual mean of individual Kalman filters [76].

Interacting Multiple Model (IMM) and heuristic pulse interleaving algorithm are proficient methods for multiple target tracking with single sensors. The heuristic pulse interleaving algorithms are used for multiple target tracking on pulse Doppler phased array radar which consists of two parts:

1. Selection of Pulse Repetition Frequency (PRF) which is used to reduce track time
2. Backward interleaving for task scheduling [77].

Interacting Multiple Model (IMM) which uses a number of filters is always one of the best methods for multiple target tracking. This has been improved by increasing the existing time of the models and reduces the complexity of the design. The algorithm developed is called Switch-Time Conditioned IMM (STC-IMM) filter [78].

5.3 Track Initiation:

Efficiency of tracking algorithm depends on the basic initialization step which is done using a regression model. This regression model collects information for a few scans. The information extracted is the velocity of the target and direction of movement. This information is used in the initialization step of a Kalman filter [79].

Another method for track initiation is achieved by using Hough transforms which is a form of a Fourier transform. After target initiation the tracking algorithms such as alpha-beta, alpha-beta-gamma, Kalman Filter, EKF, IMM can be used along with various data association techniques. This method can be used for all tracking algorithms [80]. Further reduction of false alarms is achieved by using track before detect method, which is a process by which the weak targets are eliminated even before the tracking. The valid
targets are selected using particle filter on the raw measurements at the beginning of the scan [81].

5.4 Data Association:

Data Association deals with track to target association and is very important in the presence of multiple targets and noise. Figure 5-2 shows one target which can match with two active tracks and the assignment depends on the technique called data association.

![Figure 5-2. Track 1 & 2 have the same possible target at the next state](image)

In tracking, associating the target to the tracks is very important and among all the data association techniques the Global Nearest Neighbor (GNN) gives better results for radar systems. Joint Probabilistic Data Association (JPDA) develops a set of probabilities for targets matching the tracks and its performance is poor when there are very close maneuvering targets present. Data association for maneuvering targets is performed by converting the data to clusters. This reduces false alarm rates and also reduces the process time. GNN technique along with clustering reduces the time and can be used in real time tracking [82].

Joint Probabilistic Data Association (JPDA) is modified by reducing the complexity resulting in Modified Approximate Joint Probability Density (MAJPDA) algorithm. This
method works very efficiently in the presence of clutter [75]. Munkres algorithm is also used for target to track assignment and has less number of computations [83].

5.5 Tracking by data fusion:

Data fusion techniques which use multiple sensors have higher accuracy compared to single sensor systems. Tracking in case of multiple radars uses centralized and decentralized tracking methods with Bayesian Cramer Rao Bound (BCRB) analysis to test their performance. Centralized technique has been used for accuracy and decentralized to identify the source used in tracking [84]. Figure 5-3 shows tracking by multiple sources.

![Diagram](image)

**Figure 5-3.** Example of target recognition by multiple radars

In most cases fusion algorithms are used when more than one source of data is available such as Infrared Search and Track (IRST) and radar measurements. Fusion algorithm initially augments the measurements from two sources and then calculates the weights for the measurements [85]. Infrared (IR) cameras cannot detect the distance of the object but can give the azimuthal information, the size and shape of the object. Target validation is highly reliable when many sensors are used. A neural network is used with
the input with all possible two dimensional projections of a particular object. The image sensor and radar data are combined together to provide accurate target information [86].

Radar and acoustic data can also be combined to locate the targets location. The targets and their velocities are given as state vectors for the particle filter. The performance of the particle filter depends on the proposal function which determines the weight of particles for estimation. A weighted model was implemented to overcome the delay in acoustic data. Fusion method of acoustic data with radar data without the time delay compensation method for determining state estimates are not accurate [87].

The effective use of sensors is performed by mapping individual sensor to each target location. Partially Observable Markov Decision Process (POMDP) combines particle filter for belief state estimation and sampling based Q-value approximation for identifying the maximum probability of occurrence. Particle filter is used for the conditions non Gaussian and nonlinear estimation [88].

5.6 Target Trackers as Estimators

Estimation is to predict the current value of a parameter of any system based on its previous state input coupled with random noise. Assume population as a known quantity only on a sample value then the sample is a random value subject to change. Since sample is a random value the population is the quantity to be estimated. The conditions that an estimator must satisfy are:

- The average value of an estimator should be as close as possible to the true value and hence unbiased
- The estimator should be designed such that it converges over time
- Minimum Variance
Let $x[n]$ be the observed data with random noise for $N$ observations of the model.

$$x = (x[0], x[1], \ldots, x[N - 1])$$  \hspace{1cm} (5.1)

The pdf of the model is given as $p(x; \theta)$, where $\theta$ is the parameter to be estimated from the data $x$ and the estimate of $\theta$ is given as $\hat{\theta}$ which is represented by the formula:

$$\hat{\theta} = g(x = \{x[0], x[1], \ldots, x[N - 1]\})$$  \hspace{1cm} (5.2)

Where, $g$ is the estimator used.

The design of the estimator depends on the value of $\hat{\theta}$. The final estimate is analyzed to observe its deviation from the true value. Various estimators are designed to minimize this difference value which is termed the mean square error [89].

5.7 Kalman Filter (KF):

In 1960 R.E. Kalman published a recursive estimation technique for linear filtering. It uses a set of noisy data as input and produces state estimates with minimum error rate [90]. Wiener filter shares similarity with Kalman filter which makes use of both auto correlation and cross correlation of received signal to calculate the impulse response of the filter.

Kalman filter is a very popular estimation technique used widely for linear tracking but has certain drawbacks such as noise assumptions which are Gaussian and independent. Alternative approach for linear tracking is a two stage linear filter. This gives optimal results compared to a linear Kalman filter [91]. Gaussian sum filter is an efficient linear tracker which uses mean and covariance of prior states. The disadvantages of the filter are computation time and the memory problem. Non-Gaussian and measurement-dependent functions are given as a weighted sum of Gaussian density
functions to reduce the computation time. The memory problem can be reduced by removing some density functions or combining them or approximating the density into one term. These improvements makes Gaussian sum filter more feasible for tracking applications [92]. In most cases birds don’t follow a linear path and hence the prediction of the target’s location may be inaccurate. In Kalman filters it is assumed that the target velocity is constant. Maneuvering targets however have varying velocity and to overcome this discrepancy in Kalman filter a velocity correction term is introduced. Accurate prediction is obtained by calculating Doppler velocities from multiple sources. This method provides estimates with less error when compared to the nonlinear Kalman filters [93].

Kalman filter is designed by means of state space technique which enables use of Kalman filter as smoother or filter or predictor [94]. Basic assumptions in a Kalman filter are:

- Linearity of a system: Kalman filter is used in systems which work well with the linear assumption and small error rate. The non-linearity concept can be extended from the basic Kalman filter model if the error rate is too high.
- Uncorrelated white noise: White noise has the same power at all frequencies and it makes the model less complicated for implementation. The use of Gaussian noise which gives the amplitude in the form of a bell shape in appropriate for all systems. Any set of random numbers when added are close to a Gaussian model and is the best for first and second order statistics. The Kalman filter has only first and second order statistics which make this assumption of noise perfect [90].
The application of filters is to extract the information from a given signal. Signals are represented as:

\[ y_k = a_k x_k + n_k \]  \hspace{1cm} (5.3)

Where \( y_k \) is the observed signal
\( a_k \) is the gain
\( x_k \) is the information signal
\( n_k \) is noise

Thus the purpose of filters is to estimate \( \hat{x}_k \) and the error is defined as the difference between the estimated value \( \hat{x}_k \) and \( x_k \).

Assume process of a system as:

\[ x_{k+1} = Ax_k + w_k \]  \hspace{1cm} (5.4)

Where \( x_k \) is state vector at time \( k \)
\( A \) is state transition matrix of process from time \( k \) to \( k+1 \)
\( w_k \) is process noise

Observation on \( x_k \) can be given as:

\[ z_k = Hx_k + v_k \]  \hspace{1cm} (5.5)

Where \( z_k \) is actual measurement
\( H \) is a transformation vector between state and measurement
\( v_k \) is measurement noise

It is assumed that the process and measurement noises are white noises and uncorrelated. The filter should be designed to minimize the mean square error. The two noises are assumed to be stationary and the covariance is given as:

\[ Q = E[w_k w_k^T] \]  \hspace{1cm} (5.6)
\[ R = E[v_k v_k^T] \]  \hspace{1cm} (5.7)

Mean square error is:
\[ E[e_k e_k^T] = P_k \]  \hspace{1cm} (5.8)

\( P_k \) is the error covariance given as:
\[ P_k = E[e_k e_k^T] = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] \]  \hspace{1cm} (5.9)

Where \( \hat{x}_k \) is the estimate of \( x_k \)

Let the prior estimate of \( \hat{x}_k \) as \( \hat{x}'_k \). An equation combining measurement with previous estimated data is given as:
\[ x_k = \hat{x}'_k + K_k(z_k - H\hat{x}'_k) \]  \hspace{1cm} (5.10)

Where \( K_k \) is the Kalman gain

\( z_k - H\hat{x}'_k \) is the innovation or measurement module

Substitute Equation (5.5) in Equation (5.10):
\[ x_k = \hat{x}'_k + K_k(Hx_k + v_k - H\hat{x}'_k) \]  \hspace{1cm} (5.11)

Substitute Equation (5.11) in Equation (5.9):
\[ P_k = E[((I - K_kH)(x_k - \hat{x}'_k) \]
\[ - K_kv_k][(I - K_kH)(x_k - \hat{x}'_k) - K_kv_k]^T \]  \hspace{1cm} (5.12)

\( x_k - \hat{x}'_k \) is error of prior estimate and is uncorrelated with measurement noise and

Equation (5.12) is re-written as:
\[ P_k = (I - K_kH)E[(x_k - \hat{x}'_k)(x_k - \hat{x}'_k)^T](I - K_kH) \]
\[ + K_kE[v_k v_k^T]K_k^T \] \hspace{1cm} (5.13)

Substitute Equation (5.7) and Equation (5.9) in Equation (5.12):
\[ P_k = (I - K_kH)P'_k(I - K_kH)^T + K_kR K_k^T \] \hspace{1cm} (5.14)
Where $P_{k}'$ is prior estimate of $P_k$

Equation (5.14) is the covariance update equation of Kalman filter. Expanding Equation (5.14) gives:

$$P_k = P_{k}' - K_k H P_{k}' - P_{k}' k H^T K_k^T + K_k (H P_{k}' k H^T + R) K_k^T \quad (5.15)$$

Trace of matrix is the same as its transpose, hence:

$$T[P_k] = T[P_{k}'] - 2T[K_k H P_{k}'] + T[K_k (H P_{k}' k H^T + R) K_k^T] \quad (5.16)$$

Where $T[P_k]$ is trace of $P_k$

Differentiating Equation (5.16) with respect to $K_k$:

$$\frac{dT[P_k]}{dK_k} = -2(H P_{k}')^T + 2K_k (H P_{k}' k H^T + R) \quad (5.17)$$

Solving Equation (5.17) by equating it to zero gives:

$$(H P_{k}')^T = K_k (H P_{k}' k H^T + R) \quad (5.18)$$

Thus $K_k$ is derived as:

$$K_k = P_{k}' k H^T (H P_{k}' k H^T + R)^{-1} \quad (5.19)$$

Equation (5.19) is the Kalman filter equation. The innovation defined in Equation (5.10) has measurement predicted covariance given as:

$$S_k = H P_{k}' k H^T + R \quad (5.20)$$

Substituting Equation (5.19) in Equation (5.15):

$$P_k = P_{k}' - P_{k}' k H^T (H P_{k}' k H^T + R)^{-1} H P_{k}'$$

$$= P_{k}' - K_k H P_{k}'$$

$$= (I - K_k H) P_{k}' \quad (5.21)$$

Equation (5.21) is the update equation of error covariance matrix in terms of gain. Equations (5.10), (5.19) and (5.21) is used in the estimation of $x_k$. State projection is calculated from:
\[
\hat{x}_{k+1} = A\hat{x}_k
\]  
(5.22)

In recursion process it is important to transfer the error covariance matrix to next time interval given as:

\[
e'_{k+1} = x_{k+1} - \hat{x}_{k+1}
\]
\[
= (Ax_k + w_k) - A\hat{x}_k
\]
\[
= Ae_k + w_k
\]
(5.23)

Extension of Equation (5.9) to time k+1 is:

\[
P'_{k+1} = E[e'_{k+1}e'_{k+1}^T] = E[(Ae_k + w_k)(Ae_k + w_k)^T]
\]
(5.24)

Errors \(e_k\) and \(w_k\) are uncorrelated where noise \(w_k\) occurs from time \(k\) to \(k+1\) and \(e_k\) till time \(k\), hence:

\[
P'_{k+1} = E[e'_{k+1}e'_{k+1}^T]
\]
\[
P'_{k+1} = E[(Ae_k)(Ae_k)^T] + E[w_k(w_k)^T]
\]
\[
P'_{k+1} = AP_kA^T + Q
\]
(5.25)

Figure 5-4 shows the technique of Kalman filter design.

![Figure 5-4. Kalman filter design](image)

Thus the Kalman filter equations are summarized as the prediction and update equations [94]. The prediction of the estimate is given as:
\[ \hat{x}'_k = A\hat{x}_{k-1} \quad (5.26) \]
\[ P'_k = AP_kA^T + Q \quad (5.27) \]

Where, \( \hat{x}'_k \) shows the predicted position of target in the next state

\( A \) is state transformation matrix

\( \hat{x}_k \) is the observed position of target in the current state

\( P'_k \) is the predicted covariance

\( P_k \) is the covariance of the position of the target

\( Q \) is the covariance of process noise

The update equations of Kalman filter are:

\[ K_k = P'_kH^T(HP'_kH^T + R)^{-1} \quad (5.28) \]
\[ x_k = \hat{x}'_k + K_k(z_k - H\hat{x}'_k) \quad (5.29) \]
\[ P_k = (I - K_kH)P'_k(I - K_kH)^T + K_kR K_k^T \quad (5.30) \]

Where, \( K_k \) is the Kalman gain

\( H \) is the transformation matrix

\( R \) is the covariance of measurement noise

\( z_k \) is the actual measurement

\( x_k \) is the update position of target

\( P_k \) is the updated covariance matrix

The \( z_k - Hx_k/(k-1) \) is named as measurement innovation or residual and the Kalman gain reduces the posteriori error of the filter. The prediction stage predicts the position of target in the next state and the measurement update equation updates the position based on \( z_k \), the position of target in the current state. The gain and error estimation covariance
becomes stable with time if noise level is assumed to be constant. Noise assumptions play a very important role in the filter design [95].

The Kalman filter was considered to give the best estimation for linear data. They are popular due to its optimality, easy implementation and yields good results theoretically and practically. Kalman filter is used only for linear estimations and not useful in the case of nonlinear data. Figure 5-5 shows the flow chart of a linear Kalman filter.

**Figure 5-5. Kalman Filter Design**
5.7.1 Implementation of Kalman Filter Tracker Model in radR:

The Kalman filter algorithm is implemented in radR for tracking. The parameters used and the equations are discussed. Figure 5-6 shows the input given to the Kalman filter.

![Kalman Filter Diagram]

**Figure 5-6.** Input and output parameters for Kalman filter

The dimensions and the noise levels used in the filter design are given in Table 5.1. In this Kalman design, the input is the x and y position of targets with positive and negative velocities.

**Table 5.1: Input and output parameters for Kalman filter**

<table>
<thead>
<tr>
<th>Kalman Parameters</th>
<th>radR Parameters</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{k-1}$</td>
<td>$x_{k-1}$</td>
<td>$p$ is position and $v$ is velocity</td>
</tr>
<tr>
<td>$u_k$</td>
<td>Acceleration(t)</td>
<td>acceleration</td>
</tr>
<tr>
<td>$w_k$</td>
<td>0.1*random noise</td>
<td>process noise</td>
</tr>
<tr>
<td>$v_k$</td>
<td>0.1*random noise</td>
<td>measurement noise</td>
</tr>
</tbody>
</table>

Input data obtained from radR for tracking is in the form of data frames. The data frames store all the target information in the previous frame as old.pts and the current frame as blips. An example of the data frame is given in Table 5.2. N is the number of targets in the frame.
Table 5.2: Input Data frame of targets in Kalman filter

<p>| | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
<td>z</td>
<td>t</td>
<td>ns</td>
<td>area</td>
<td>int</td>
<td>max</td>
<td>aspan</td>
<td>rspan</td>
<td>perim</td>
<td>range</td>
<td>scan</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>------</td>
<td>-----</td>
<td>-----</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The input data frame is read into the Kalman filter algorithm for target tracking. The estimated target positions of targets are then matched with targets in the current frame. The Kalman filter as described in the previous section has two stages which are called prediction and update.

The Kalman filter prediction involves the prediction of target position based on transition matrices. The transition matrices are time dependent. In this design, the target position, velocity, acceleration and process noise is used for prediction. The error covariance and the current measurement are calculated based on the predicted value. The implementation of equations in the tracker model is given in Equations (5.31), (5.32) and (5.33). Figure 5-7 shows the two stages in the Kalman filter design.

![Figure 5-7. Prediction stage in Kalman filter](image)

Kalman filter equations are described in the previous section [96] and are:
\[
\hat{x}_k = Ax_{k-1} + Bu_k + w_k \\
P_k = AP_{k-1}A^T + Q \\
z_k = H\hat{x}_k + v_k
\]

Where \( \hat{x}_k \) the prediction target position and \( t \) is the time interval between frames.

\( A \) is a transition matrix and \( A = \begin{pmatrix} 1 & 0 \\ t & 1 \end{pmatrix} \)

\( B \) is a transition matrix and \( B = \begin{pmatrix} t^2 \\ 2t \end{pmatrix} \)

\( H \) is a transition matrix and \( H = \begin{pmatrix} 1 & 0 \end{pmatrix} \)

The noise is assumed as random white noise. radR implementation of Kalman filter prediction is given as:

\[
\hat{x}_k = \begin{pmatrix} 1 & 0 \\ t & 1 \end{pmatrix} x_{k-1} + \begin{pmatrix} t^2 \\ 2t \end{pmatrix} u_k + w_k
\]

\[
P_k = 0.1 \ast random\ noise(t)
\]

\[
z_k = (1 \ 0)x_k + 0.1 \ast random\ noise
\]

The Kalman filter update stage involves computation of Kalman gain and updating of the target position. Figure 5-8 shows the update stage of the Kalman filter.

\textbf{Figure 5-8. Update stage of Kalman filter}
Kalman filter equations [96] are discussed in details in Section 5.7 and they are incorporated in the radR as:

\[ K_k = P_k H^T (HP_k H^T + R)^{-1} \]  
\[ x_k = \hat{x}_k + K_k (z_k - H\hat{x}_k) \]  

The radR implementation of Kalman filter update is given as:

\[ K_k = P_k (1\ 0)^T ((1\ 0)P_k (1\ 0)^T + 0.1 \times \text{random noise})^{-1} \]  
\[ x_k = \hat{x}_k + K_k (z_k - (1\ 0)\hat{x}_k) \]

The design of Kalman filter with the association of parameters within the two stages is shown in Figure 5-9. Implementation of the algorithm is shown in Figure 5-10.

![Figure 5-9. Association of parameters within two stages](image-url)
Figure 5-10. Implementation of Kalman filter algorithm in radR

The algorithm is given in the form of a flow chart in Figure 5-11.
Figure 5-11. Flow chart of the Kaman filter tracker model
Kalman filter estimates values for each target with positive and negative velocities. The nearest neighbor data association is applied to match the estimated values with the targets in the current frame. Figure 5-12 shows the matching of estimated values.

**Figure 5-12. Kalman filter design in radR**

### 5.7.2 Variations of Kalman Filter:

Extended Kalman Filter (EKF) is a nonlinear version of the Kalman filter. EKF is a Kalman filter that linearizes data about the mean and covariance similar to Taylor series [95]. The system nonlinearities are approximated for the final state estimate using appropriate linearization of the nonlinear model. Optimal approximation is obtained by the linearization of the nonlinear model such that it is not affected in any domain [97-99].

In a system with high nonlinearity the EKF gives poor performance and thus Unscented Kalman Filter (UKF) was developed [100-101]. It is an extended formulation of unscented transformation. UKF is based on the idea that it is simpler to find approximation of a probability distribution compared to a nonlinear function.
The UKF gives greater performance during high non linearity as the Gaussian random variable is given as a number of points which obtains the mean and covariance of the system with higher accuracy. The UKF can be used if EKF gives poor performance, for all other cases EKF is a better option because of its low computation cost. The performance of the EKF greatly depends on the accuracy of data linearization. However the performance of UKF was found to be better than particle filters [102]. Kalman-Bucy is the continuous time version of discrete Kalman filter and has been described in Reference [103].

Ensemble Kalman Filter (EnKF) design uses an ensemble which is samples for a state to create an analysis phase for the system. The analysis uses the same Kalman gain value and hence reduces the computational cost. EnKF can be implemented only if the ensemble of a system can be represented statistically. The size assumption of the ensemble is also important as small sizes gives rise to under sampling [104]. The ensemble Kalman filter is very efficient because of the low dimensions of matrices used. These filters are divided into two and are called as perturbed observations and square root filter [105].

5.8 Particle Filter

Tracking for nonlinear and non-Gaussian data can be performed by particle filter as Kalman filter assumes data as Gaussian. In the presence of multiple targets within single radar beam tracking becomes difficult as the discrimination of target is inaccurate. Particle filters are used in this case by calculating the posterior density for different values of targets which is converted to likelihood functions and helps to detect the number of targets. This approach can simultaneously handle processing, data association
and tracking targets [108-110]. Particle filters are also known as sequential Monte Carlo filters and are used when the system is nonlinear and non-Gaussian. Particles filters are a version of point mass filters in which the particles are present in the place of higher probability of occurrence. If for the Bayesian models in the design the integrals cannot be analyzed then the posterior pdf can be represented as a set of randomly defined weights [108-110]. Sequential importance filtering is a simple particle filter estimation method. In this a recursively occurring Bayesian filter technique is used. The posterior density is calculated as weights. The output is in the form of samples and their corresponding weights.

5.8.1 Sequential Importance Sampling (SIS):

In this method the posterior density is defined as a set of samples with weights. Let the initial position of the targets with their associated weights be \((x_{0:k}^i, W_{k}^i)_{i=1}^{N_s}\). \(N_s\) is a user defined parameter which represents the number of particles. The posterior density can be approximated as:

\[
p(x_{0:k}|z_{1:k}) \approx \sum_{i=1}^{N_s} W_{ik} \delta (x_{0:k} - x_{0:k}^i)
\]  

\(x_{0:k}\) is a set of all states up to time \(k\)

\(z_{1:k}\) is the number of measurements available up to time \(k\)

Weights are assigned based on the concept of important sampling. Consider a system with probability density \(p(x)\) which cannot be calculated appropriately. Let this probability density be proportional to a value \(\pi(x)\) which is easy to analyze. The samples for \(N_s\) points are evaluated from importance density represented as \(q\). Weight is calculated as \(w^i \propto \frac{\pi(x^i)}{q(x^i)}\) and the weighted approximation for density is:
If \( x^i_{0:k} \) are derived from importance density then weights are defined as:

\[
W^i_k \propto \frac{p(x^i_{0:k} | z_{1:k})}{q(x^i_{0:k} | z_{1:k})}
\]

(5.43)

The weights are updated with the Equation (5.44):

\[
W^i_k \propto W^i_{k-1} \frac{p(z_k | x^i_k) p(x^i_k | x^i_{k-1})}{q(x^i_k | x^i_{k-1}, z_k)}
\]

(5.44)

The filtered posterior density is given as:

\[
p(x_k | z_{1:k}) \approx \sum_{i=1}^{N_s} W^i_k \delta(x_k - x^i_k)
\]

(5.45)

Figure 5-13 shows the algorithm of SIS particle filters in which number of particles are assumed. The particles are initialized with posterior density and weights. The weights of the particles are updated and they define the probability of occurrence of the particle in the new state [109].
Figure 5.13. Sequential Importance Sampling Particle Filter

5.8.2 Implementation of Particle Filter Tracker Model in radR:

The particle filter algorithm is implemented in radR for tracking. Figure 5.14 shows the structure of the particle filter.

Figure 5.14. Input and output parameters for the particle filter

The input and output parameters used in this design are given in Table 5.3.
Table 5.3: Input and output parameters for particle filter

<table>
<thead>
<tr>
<th>Particle Parameters</th>
<th>Filter Parameters</th>
<th>radR Parameters</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_s$</td>
<td>$N_s = 100$</td>
<td>$N_s$ is the number of particles assumed</td>
<td></td>
</tr>
<tr>
<td>$x_{0:k}^i$</td>
<td>$x_{b:k}^i$</td>
<td>Number of particles around the initial target position</td>
<td></td>
</tr>
<tr>
<td>$W_k^i$</td>
<td>$W_k^i$</td>
<td>Random weights with range (0:1)</td>
<td></td>
</tr>
<tr>
<td>$p(x_k</td>
<td>z_{1:k})$</td>
<td>$\sum_{i=1}^{N_s} W_{lk} \delta(x_k - x_k^i)$</td>
<td>Posterior density is the summation of product of updated weights and predicted values for the particles.</td>
</tr>
</tbody>
</table>

Input data obtained from radR for tracking is in the form of data frames. The data frames store all the target information in the previous frame as old.pts and the current frame as blips. An example of the data frame is given in Table 5.4. $N$ is the number of targets in the frame.

Table 5.4: Input Data frame of targets in particle filter

```
x y z t ns area int max aspan rspan perim range scan
1
2
.
.
.
.
N
```

The input data frame is read and the particle filter algorithm is computed for tracking the targets. The estimated target positions of targets are then matched with targets in the current frame. The particle filter as described in the previous section first initializes a nonlinear system prediction for assumed number of particles and then their weights are
updated. The prediction for the particles is the initial part of the design. Figure 5-15 shows the two stages of the design.

![Particle Filter Diagram](image)

**Figure 5-15.** Prediction for N particles

The initial stage involves prediction of values $p(x_k^i | x_{k-1}^i)$ and $p(z_k | x_k^i)$. Consider a nonlinear system as:

$$x_k = f(x_{k-1}) + w_{k-1}$$  \hspace{1cm} (5.46)

$$z_k = h(x_k) + v_k$$  \hspace{1cm} (5.47)

Where $x_{k-1}$ is the input target position

$w_k$ and $v_k$ are the process and measurement noise

$f$ and $h$ are nonlinear functions of process and observation vector

$z_k$ is the current observed measurement

Equations (5.46) and (5.47) are used to predict the values of the system. Equations for particle filter can be assumed as:

$$x_k = 1.04 \cdot (x_{k-1}) + w_{k-1}$$  \hspace{1cm} (5.48)

$$z_k = 0.96 \cdot (x_k) + v_k$$  \hspace{1cm} (5.49)

Where $w_k = 40 \cdot \exp(-n/10)$

$v_k = \cos(t)$

$f = 1.04$
Equations (5.48) and (5.49) are implemented for $N_s$ number of particles as:

\[
p(x_k^i | x_{k-1}^i) = x_k^i = 1.04 \times (x_{k-1}^i) + w_{k-1}
\]

\[
p(z_k^i | x_k^i) = z_k^i = 0.96 \times (x_k^i) + v_k
\]

The prediction is performed for both positive and negative velocity of targets in the particle filter model. Weights are updated and the posterior density is calculated. Figure 5-16 shows the update part of the particle filter.

\[
W_k^i \propto W_{k-1}^i \frac{p(z_k^i | x_k^i) p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{k-1}^i, z_k)}
\]

\[
p(x_k | z_{1:k}) \approx \sum_{i=1}^{N_s} W_{ik} \delta(x_k - x_k^i)
\]

Implemented weight update and posterior density calculation in radR are:

\[
W_k^i = e^{-\frac{(x_k^i - z_k^i)^2}{2}}
\]

\[
p(x_k | z_{1:k}) \approx \sum_{i=1}^{N_s} W_{ik} \times x_k^i
\]
The posterior density is the estimated value of the target. The association of parameters within the two stages is shown in Figure 5-17. Implementation of the algorithm is shown in Figure 5-18.

**Figure 5-17.** Association of parameters within two stages

![Particle Filter Diagram](image)

**Figure 5-18.** Implementation of Particle filter algorithm in radR

A flow chart of the particle filter algorithm is shown in Figure 5-19.
Figure 5-19. Flow chart of the particle filter tracker model
Particle filter estimates values for each target with positive and negative velocities. The nearest neighbor data association is applied to match the estimated values with the targets in the current frame. Figure 5-20 shows the matching of estimated values.

![Particle filter design in radR](image)

**Figure 5-20.** Particle filter design in radR

### 5.9 Compressive Sampling:

Data occurring in real time are generally sparse but most of the signal processing methods ignores this fact. Compressive Sampling is based on the idea that for certain signals a small number of samples within the signal are enough to represent the entire signal. The signal reconstruction is based on how accurate the signal is approximated from the available samples [111-113].

Consider vectors $\mathbb{C}^N$ which has $l_0$ (quasinorm) defined as:

$$
\|x\|_0 = |\text{supp}(x)| = |\{j : x_j \neq 0\}|
$$

(5.56)

The signal $x$ is sparse if $\|x\|_0 \leq s$, where $s$ defines the sparsity of the signal. Sparsity means non zero values in the signal. Sample is a function of the given signal. Many
samples can be extracted for the signal and is called a sampling matrix \((\Phi)\). Consider for
the given signal \(c^N\), \(m\) samples are extracted. The sampling matrix has \((m, N)\)
dimensions. It is important to identify the measurements required to attain sparse signals.
Two dissimilar sparse signals should not be mapped to the same samples and Restricted
Isometry Property (RIP) is applied to preserve sparsity [111].

Restricted Isometry Property (RIP) helps in characterizing a matrix which contains
sparse vectors. Consider an observed signal represented as:

\[
y = \Phi x
\]

Where \(x \in \mathbb{R}^N\) is the signal to be recovered
\(y \in \mathbb{R}^m\) is observations at current state
\(\Phi\) is a matrix of dimensions \(m \times N\)

In sparse signals if \(m < N\), \(x\) has to be reconstructed with good approximation. To
reconstruct \(x\) accurately it should be a sparse signal and the known matrix \(\Phi\) should
satisfy the RIP. The Restricted Isometry Property (RIP) of \(\Phi\) is given as:

\[
(1 - \delta_r)\|x\|^2_2 \leq \|\Phi x\|^2_2 \leq (1 + \delta_r)\|x\|^2_2
\]

Where \(\Phi\) is the sampling matrix which is a linear function of the data.
\(\delta_r\) is the least number of the sampling matrix which satisfies the RIP property [114].

Signal recovery algorithms are used to reconstruct a signal from a vector which
contains noisy data. A greedy pursuit algorithm can be used for signal reconstruction.

In greedy pursuits, approximations are based on a step by step process until an
optimal condition is reached. In these methods a signal is recovered recursively and
reconstructed using pseudo inverse. Some of the greedy methods are Orthogonal
Matching Pursuit (OMP), stage wise OMP (StOMP) and Regularized OMP (ROMP) [115].

5.9.1 CoSaMP:

Compressive Sampling Matching Pursuit (CoSaMP) is a greedy pursuit method for signal reconstruction [112-113]. Let \( \varphi \) be a sampling matrix of dimensions \( m \times N \) which has a restricted isometry property. The signal is a set of vectors given as \( u = \varphi x + e \), where \( e \) is noise. For precision given by parameter \( \eta \), the CoSaMP gives an approximation \( a \) that satisfies:

\[
\|x - a\|_2 \leq C. \max\left\{ \eta, \frac{1}{\sqrt{s}} \|x - x_{s/2}\|_1 + \|e\|_2 \right\}
\]  

(5.59)

Where \( x_{s/2} \) is the \((s/2)\) – sparse approximation of \( x \)

\( \eta \) is the precision parameter

\( e \) is the noise

\( a \) is the sparse approximation selected to satisfy the condition

Sparse approximation can be selected in two ways, one is by \( s \approx \frac{m}{2} \log N \) where \( m \) is the number of measurements of a signal of length \( N \) and another method is to vary the range of values and select the best fit as the sparse level.

In CoSaMP the most important steps are:

- Identification of largest components of the signal
- Merging the identified values with the current approximation
- Estimation using least squares method
- Retaining only the largest components from the least square
- Update the procedure till the required residue is obtained [111]
5.10 BCoSaMP:

Most signals obtained at real time are sparse. Signals are sampled through compression techniques using linear projection of signal and a projection matrix is formed. The Compressive Sampling Matching Pursuit algorithm (CoSaMP) is a greedy approach and the state error decreases over the number of iterations [116].

BCoSaMP (Block CoSaMP) divides the projection matrix into grids. They are created by the Doppler-d delay plane given as $N_\tau$ (delay factor) and $N_\beta$ (Doppler factor). Target tracking is based on grids. If the number of targets present is less than the Doppler-delay plane then it is a sparse target data. If $P$ is the position of target travelling with a velocity $v$ with the direction of arrival is $\vec{u}$ then the delay and Doppler factor is given as:

$$\tau = \frac{2||P||_2}{c}$$

$$\beta = \frac{2\langle v, \vec{u} \rangle}{c}$$

Where $\tau$ is the delay factor

$P$ is the position of target

$c$ is the speed of light in meters per second

$\beta$ is the Doppler factor

$v$ is the velocity of target

$\vec{u}$ is the direction of arrival of the target

The predicted value of the target is given as:

$$x_k = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^T x_{k-1} + w_k$$

Where $x_k$ is the predicted value

$x_{k-1}$ is the current value
$w_k$ is the noise level

The sparse model is created using the delay $N_t$ and Doppler grid $N_\beta$ which is given as $N_g = N_t N_\beta$. Consider $M$ targets in these grid points then the measurement model at $k$ is given as:

$$y_k = \sum_{i=1}^{N_t} \sum_{j=1}^{N_\beta} \bar{\phi}(\tau_k^i, \beta_k^j) + e_k$$  \hspace{1cm} \text{(5.63)}

Where $\sum_{i=1}^{N_t} \sum_{j=1}^{N_\beta} \bar{\phi}(\tau_k^i, \beta_k^j)$ is a sampling matrix which has the delay-Doppler factors of targets within the grid

$e_k$ is the noise

Equation (5.63) can be modified as:

$$y_k = \bar{\phi}_k + e_k$$  \hspace{1cm} \text{(5.64)}

Where, $\bar{\phi}_k$ is called the sampling matrix and $\bar{\phi}_k = [\bar{\phi}(\tau_k^1, \beta_k^1) \ldots \bar{\phi}(\tau_k^{N_t}, \beta_k^{N_\beta})]$

$e_k$ is the noise

Thus each block gives the sampling matrix of delay Doppler grid of the target. This is the method of creating a block sparse signal for the given data to which the CoSaMP algorithm is applied for tracking.

5.10.1 Algorithm:

**Step 1:** $x_k$ is the predicted value given in Equation (5.62)

**Step 2:** The residue is given as $r = y_k$ which is the current measurement

**Step 3:** Create state proxy $p = r \ast \phi^H$

Here $\phi$ is the sampling matrix with $\phi = [\phi(\tau^1, \beta^1) \ldots \phi(\tau^{N_t}, \beta^{N_\beta})]$

This consists of all possible combinations of delay and Doppler factor.
Step 4: $\Omega = \text{bsupp}(P_{2s})$

Where b supp is called block support which indicates non-zero block values.

The largest blocks of the matrix are identified and if the number of targets is less than the grid size, assume $s = m$ where $m$ is the number of targets.

Choose the highest $2s$ block from the matrix as $\Omega$, where $s$ is the sparsity level.

Step 5: $T = \Omega \cup \text{bsupp}(x_k)$

Merge the block supports with the predicted value.

Step 6: $b|_T = (\varphi)_T y_k$, $b|_{T^c} < -0$ is the state estimation using least squares.

Step 7: Estimated value

$\hat{x} = b_s$

Step 8: Residue

$r = y_k - (\varphi)_k \hat{x}$

Step 9: Continue steps 3 to 8 till the required halting criteria of the residue is reached.

Step 10: Estimated value at the final stage $\hat{x}$ is the target position at current state.

Residue $r$ increases in the beginning and then it decreases at a certain point. This point will be used as a termination criterion and program will be stopped [117]. Figure 5-21 shows the BCoSaMP algorithm.
5.10.2 Implementation of BCoSaMP Tracker Model in radR:

The BCoSaMP algorithm is implemented in radR for tracking. Figure 5-22 shows various input given to the BCoSaMP algorithm.
The input parameters used in the design are given in Table 5.5. The grid size is defined based on many trials and choosing the size with the least error rate.

**Table 5.5: Input and output parameters for BCoSaMP**

<table>
<thead>
<tr>
<th>BCoSaMP Parameters</th>
<th>radR Parameters</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r = y_k )</td>
<td>( y_k = (\tau, \beta) )</td>
<td>Initial target position in terms of Doppler factors</td>
</tr>
<tr>
<td>( N_G )</td>
<td>( N_G = (60*\tau) \times (40*\beta) )</td>
<td>Grid in terms of range and time delay</td>
</tr>
<tr>
<td>( S )</td>
<td>( S = ) number of targets within grid</td>
<td>Number of targets within the grid</td>
</tr>
<tr>
<td>( \varphi_k )</td>
<td>( \varphi = [\varphi(\tau^1, \beta^1), \ldots, \varphi(\tau^{N_T}, \beta^N)] )</td>
<td>Sampling matrix consisting of Doppler factors for the entire grid</td>
</tr>
</tbody>
</table>

Input data obtained from radR for tracking is in the form of data frames. The data frames store all the target information in the previous frame as old.pts and the current frame as blips. An example of the data frame is given in Table 5.6.

**Table 5.6: Input Data frame of targets in BCoSaMP**

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
<th>t</th>
<th>ns</th>
<th>area</th>
<th>int</th>
<th>max</th>
<th>aspan</th>
<th>rs</th>
<th>span</th>
<th>perim</th>
<th>range</th>
<th>scan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>
The input data frame is read and the BCoSaMP algorithm is implemented for the targets. The estimated target positions of targets are then matched with targets in the current frame.

The prediction of position of target is calculated similar to Kalman filter. Figure 5-23 shows the two stages of the design.

![Figure 5-23. Prediction of target position](image)

The prediction of target position is calculated using the formula:

$$x_k = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} x_{k-1} + w_k$$  \hspace{1cm} (5.65)

where $t$ is the sampling interval. $t=0.8$ is used for three nautical mile range for the marine radar.

Implementation of target prediction in radR model:

$$x_k = \begin{bmatrix} 1 & 0.8 \\ 0 & 1 \end{bmatrix} (\tau, \beta)$$  \hspace{1cm} (5.66)

The next stage involves the update of the target position. Figure 5-24 shows the update stage of BCoSaMP.
Update equations in radR model are derived as:

\[ p = (\tau, \beta) \ast [\varphi(\tau^1, \beta^1), \ldots, \varphi(\tau^N, \beta^N)]^H \]  \hspace{1cm} (5.67)

\[ \Omega = \text{Number of targets within each } p \text{ value in the grid} \]  \hspace{1cm} (5.68)

\[ T = \Omega \cup (x_{k_1} \text{ where } x_{k_1} \text{ is not zero}) \]  \hspace{1cm} (5.69)

\[ b|_T = \text{Pseudo inv} (\varphi) \ast (\tau, \beta) \]  \hspace{1cm} (5.70)

\[ \hat{\varphi} = b \]  \hspace{1cm} (5.71)

\[ \text{Residue} = r = y_K - (\varphi_K) \ast \hat{\varphi} \]  \hspace{1cm} (5.72)

This update loop is repeated until the assigned termination criterion is obtained. \( \hat{\varphi} \) is the estimated value of the target. The association of parameters within the two stages is shown in Figure 5-25. Implementation of the algorithm is shown in Figure 5-26.
Figure 5-25. Association of parameters within two stages

Figure 5-26. Implementation of BCoSaMP algorithm in radR

The algorithm is given in the form of a flow chart in Figure 5-27.
Figure 5-27. Flow chart of BCoSaMP model
5.11 Modified BCoSaMP:

The BCoSaMP tracker model is further improved by using the Kalman filter in the estimation instead of least square model. The algorithm is shown in Figure 5-28. The Kalman estimation is used to get the best estimation and the linearity of the Kalman filter does not affect the design as the algorithm is based on grids. The implementation of the model in radR is similar to BCoSaMP with the same initial assumptions.

![Diagram](Figure 5-28. Modified BCoSaMP model for tracking)

5.12 Layout of Tracker Model in radR:

Tracker models in radR consist of the tracking algorithm which specifies criterion for building tracks. Currently there are two tracker models in radR which are the nearest neighbor and multi-frame correspondence. The tracker model has various functions defined for each task and are shown in Figure 5-29. Each function performs a unique task. The update function updates each track by adding blips to the existing track or
starting a new track. Other functions are used to define the maximum speed of the track, select, deselect, load and unload a model. Get Menu function receives the user input for controlling the tracking procedure based on the minimum number of blips required to form a track and its maximum speed. Set blip fresh time selects the expiry time of the track. If the player is stopped or paused the scans are reprocessed. The tracking algorithm is implemented inside the update function. Gain array is created which stores the column index of the matched blips. Details of two current tracker models in radR are discussed in Chapter 4.

![Figure 5-29. Algorithm for tracker model](image)

### 5.12.1 Update Function:

The tracking algorithm is implemented inside the update function. Tracking algorithm identifies appropriate tracks for each blip. Tracks are stored in a file called
which_slots_full. Tail points of each track are saved in old.pts data frame and current blips are available in blips data frame. Blip information is extracted from these data frames using blips$parameter (e.g. blips$x for x-position) or old.pts$parameter. Tracking algorithm builds tracks by matching each blip in the current frame to the active tracks. Blips which have no matches start new tracks. Gain array is defined in this function which stores the index of the matched blips. The blips which have no matches are retained for three consecutive scans and are then terminated. Gain array obtained from the tracker model file is used by tracker plugin to create tracks on the GUI. Figure 5-30 shows the layout of tracker model in radR.

![Diagram of tracker model in radR](image)

**Figure 5-30.** Layout of tracker model in radR

### 5.13 Development of Tracker Models:

Four models were implemented in radR for tracking and are discussed in detail in the following sections. Four tracking algorithms implemented in this work are Kalman filter, particle filter, Block Compressive Sampling Matching Pursuit (BCoSaMP) and modified
BCoSaMP. Figure 5-31 shows four tracker models developed in radR for different data types.

**Figure 5-31. Developed tracker models in radR**

The nearest neighbor model is used as a template. The tracking algorithm is implemented in the update function. Figure 6-4 shows the layout of the tracker model and the existing tracking algorithm (Nearest Neighbor) is replaced by one of the newer tracking algorithms as shown in Figure 5-32. Tracker models can then be created in radR.

**Figure 5-32. Implementation of new algorithms in radR**
5.13.1 Kalman Filter Tracker Model:

The Kalman filter was discussed in Chapter 5. A linear discrete Kalman filter model is used for tracking. Important steps in Kalman filter implementation are input target location, initial assumptions, prediction stage, update stage and data association. Many trials are made to make sure that the assumptions are appropriate for the environmental conditions and the amount of noise levels in the data.

Steps for Kalman filter design includes the creation of file kalman.model.R in the tracker plugin folder. The Kalman filter tracking algorithm is implemented inside the update function. Input is taken as the data from the previous frame and is defined as old.pts in radR. Gaussian noise has been assumed. The velocity of the bird target is set to 8 m/sec. The velocity can be varied based on wind conditions. The prediction and update operations of Kalman filter provide the final estimation for each target. In multiple target environment, there is a possibility of appearance of noise along with accurate bird tracks. Estimated values are matched to the blips in the current frame based on the minimum distance. Since the nearest neighbor technique is used for data association, every point has a track and this inefficiency can be removed by setting a threshold value for the distance. A gain function is created similar to the nearest neighbor tracker model in radR and hence the tracks are updated according to the Kalman filter algorithm.

In this tracker model design, target with positive and negative velocities are assumed to find appropriate tracking in all directions irrespective of the target’s motion. Various steps of the Kalman filter design are shown in Figure 5-33. Outlines of nearest neighbor, multi-frame and Kalman filter are also shown in Figure 5-33.
Figure 5-33. Comparison of current tracker models with Kalman filter

- Initial target position
  - Positive and negative velocity assumption
  - Noise assumptions
  - Initialization of transition matrices
- Target prediction
- Covariance prediction, Kalman gain
- Update target position
- Nearest neighbor data association
- Save max. path cover
- Calculate gain for additional points in new frame
- Track to point gain calculation
- Point to point gain calculation
- Save correction and extension edges between frames
- Set gain above threshold to one
- Set gain below threshold to zero
- Set max. distance, max. speed, min. gain
- Gain calculation
- Velocity filtering
- Velocity filtering
- Change in rate of intensity filtering
- Change in rate of area filtering
- Change in rate of turning angle filtering
- Gain calculation
- Velocity filtering
- Change in rate of intensity filtering
- Distance between blips in two frames
- Change in rate of area filtering
- Change in rate of turning angle filtering
- Bipartite matching to form tracks

- Nearest Neighbor
- Multi-frame
- Kalman filter
- Gain array with matched blips
- Tracker plugin
5.13.2 Particle Filter Tracker Model:

The particle filter is implemented inside the update function. The implementation of the model is similar to the Kalman filter tracker model. Initial parameters for particle filter are the number of particles, weight assignment for each particle, the target initial position, the nonlinearity of data and the assigned probability of detection based on weight levels.

Steps for particle filter design include the creation of particle.model.R file in the tracker plugin folder. The basic model of Sequential Importance Sampling (SIS) particle filter is considered in this work with 100 particles. Initializations include input data which is the position of targets, number of particles, pdf of each particle in the form of weights.

Second stage includes the filter design for N number of particles. Weights are updated for each particle. The pdf is calculated to update the target position. Thus estimates for the targets are obtained. The data association is the final stage of the design and the nearest neighbor technique is used. A gain function is created with the matched blips for tracking. To overcome the limitation of missed target detection, blips in the frame are retained for at least three frames in the future scans. Comparison of current tracker models with particle filter is given Figure 5-34.
Figure 5-34. Comparison of current tracker models with particle filter.
5.13.3 BCoSaMP Tracker Model:

The block compressive sampling matching pursuit algorithm is used for tracking based on the sparse techniques. The algorithm is modified for the marine radar data as the radar is assumed to be stationary in this work. The model is implemented inside the update function in filename.model.R file. The data is assumed to be sparse. In case of real time data, it is important to assign an appropriate residue function to avoid an infinite loop.

Steps for BCoSaMP filter design include the creation of bcosamp.model.R file in the tracker plugin folder. Initial steps required for this algorithm are conversion of the target position, direction of arrival or bearing and velocity to Doppler models. The sampling interval is selected based on the data range for prediction of target position. The grid size is defined around the target based on the predicted value. The sparse level is defined based on the number of targets present within the assigned grid. A sampling matrix containing the Doppler factors for the defined grid is created. The second part of the filter design includes formation of state proxy using the sampling matrix, identifying the largest sparse blocks and merging with the predicted value.

The update stage is the least square estimation of the blocks. The residue is calculated for the value obtained and the loop is stopped when the required termination criteria is reached. The value of position is updated when the loop terminates giving the estimated target position. Final estimated values are then converted back to Cartesian form and the nearest neighbor data association is performed. A threshold is applied for each estimated value based on the value of residue. Comparison of current tracker models with BCoSaMP Algorithm is given in Figure 5-35.
Figure 5-35. Comparison of current tracker models with BCoSaMP Algorithm
5.13.4 Modified BCoSaMP Tracker Model:

The modified BCoSaMP model is created similar to the BCoSaMP model with the same assumptions. The least square estimation is replaced with Kalman filter estimation for the model.

Figure 5-36 shows various models developed in radR environment and results are discussed in the following sections.

![Figure 5-36. New tracker models in radR](image)

5.14 Summary:

Four tracker models are discussed and they have been implemented in radR. Four tracker algorithms are Kalman filter, particle filter, BCoSaMP and modified BoCoSaMP. Tracker algorithms incorporate various parameters and various assumptions are described. Compressive Sampling Matching Pursuit (CoSaMP) is used for reconstruction
of sparse signals. It is proved to be robust and efficient method for signal recovery. This technique is used for efficiently tracking multiple targets and its efficiency is proved to be better than or equal to particle filter with less computation time [117]. The study area and results with various types of data are discussed in the following chapter.
Chapter 6

Simulation and Results

6.1 Introduction:

Ottawa National Wildlife Refuge, OH was recommended as the study site by the project team’s wildlife biologists. This wildlife refuge was established in 1961 to serve as a habitat for migrating birds. The habitat at Ottawa Wildlife Refuge is protected to offer nesting and stopover opportunities during migration period and habitat for birds [118]. The radar was set up at the location shown in Figure 6-1 along the coastline of western Lake Erie basin to study behavior of birds. The marine radar was run all through the night and the data was analyzed to study behavior of nocturnal migrating birds during the fall and spring migration period.
Four tracking algorithms have been considered in this work and have been implemented within radR. Performance of these implemented tracking algorithms is evaluated and compared with currently available Multi-frame tracker model in radR and is discussed in the following sections.

6.2 Radar Data:

The output of the tracking algorithms are analyzed with three types of data set. Single simulated track at constant velocity of 8 m/sec for 25 scans. Figure 6-2 shows the simulated track. Four simulated tracks occurring simultaneously for 25 scans in radR. The four tracks are simulated in the radR and the graphical representation of the tracks is shown in Figure 6-3.

Figure 6-1. Study area- Ottawa National Wildlife Refugee

Courtesy: Google Earth
Figure 6-2. Single simulated track at constant velocity

Figure 6-3. Simulated multi-target tracks
Figure 6-4 shows the tracks simulated with the enabling of target trails plugin.

Figure 6-4. Simulated multi-target tracks in radR environment

Marine radar data collected at the Ottawa National Wildlife Refuge on April 20, 2012 using T-bar antenna in the vertical mode from 4:13 AM to 6:45 AM was also used. Various tracking algorithms are tested with different data sets and their performance is analyzed. Snapshot of marine radar data is shown in Figure 6-5.
6.3 Simulations of Tracker Models:

Kalman filter tracking algorithm was tested with previously described simulated tracks. Figure 6-6 shows the output of the Kalman filter tracking with the actual target track. The error rate of the Kalman filter is shown in Figure 6-7. Kalman filter in case of a single track estimates the target position with very less error rate. It can also be seen that the error is decreasing with the increase in time.
Figure 6-6. Kalman filter single target tracking

Figure 6-7. Error rate of Kalman filter
Figure 6-8 shows the output of the Kalman filter tracking with four target tracks. The error rate of the Kalman filter is shown in Figure 6-9. It can also be seen that the error is decreasing with the increase in time.

**Figure 6-8.** Multi target tracking using Kalman filter

**Figure 6-9.** Error rates for the four tracks
The tracks are simulated in radR environment and the tracker model is tested by adding background noise to the data. Each scan consists of 20 randomly occurring blips. Figure 6-10 shows the Kalman filter tracker model in the case of four targets without any background noise.

![Figure 6-10. Kalman filter tracker model output in radR](image)

Figure 6-11 shows the tracks created in the presence of background noise. The tracks created are not continuous due to the effect of noise.
Figure 6-11. Kalman filter tracker model output with noise in radR

Figure 6-12 shows the output of the particle filter tracking with the actual target track. Figure 6-13 shows the error rate of particle filter estimated target track. Particle filter in case of a single track estimates the target position more accurately than Kalman filter model.
Figure 6-12. Particle filter single target tracking

Figure 6-13. Error rate of particle filter
Figure 6-14 shows the Particle filter estimated track for multi targets. Figure 6-15 gives the error rate of the four tracks.

**Figure 6-14.** Multi target tracking using particle filter

**Figure 6-15.** Error rates for the four tracks
The tracks are simulated in radR environment and the tracker model is tested by adding background noise to the data. Each scan consists of 20 randomly occurring blips. Figure 6-16 shows the particle filter tracker model output in the case of four targets without any background noise.

Figure 6-16. Particle filter tracker model output in radR

Figure 6-17 shows the tracks created in the presence of background noise. The tracks created are not continuous due to the effect of noise.
The BCoSaMP estimation of the target is shown in Figure 6-18. Figure 6-19 shows the error rate of the estimated track.
Figure 6-18. BCoSaMP single target tracking

Figure 6-19. Error rate of BCoSaMP tracking
Figure 6-20 shows the BCoSaMP estimated track for multi targets. Figure 6-21 gives the error rate of the four tracks.

**Figure 6-20.** Multi target tracking using BCoSaMP

**Figure 6-21.** Error rates for the four tracks
The tracks are simulated in radR environment and the tracker model is tested by adding background noise to the data. Each scan consists of 20 randomly occurring blips. Figure 6-22 shows the BCoSaMP tracker model output in the case of four targets without any background noise.

![BCoSaMP tracker model output in radR](image)

**Figure 6-22.** BCoSaMP tracker model output in radR

Figure 6-23 shows the tracks created in the presence of background noise. The tracks formed are not continuous and some error is also induced in the tracks.
Figure 6-23. BCoSaMP tracker model output with noise in radR

The modified BCoSaMP estimation of the target is shown in Figure 6-24. Figure 6-25 shows the error rate of the estimated track.
Figure 6-24. Modified BCoSaMP single target tracking

Figure 6-25. Error rate of modified BCoSaMP tracking
Figure 6-26 shows the BCoSaMP estimated track for multi targets. Figure 6-27 gives the error rate of the four tracks.

**Figure 6-26.** Multi target tracking using modified BCoSaMP

**Figure 6-27.** Error rates for the four tracks
The tracks are simulated in radR environment and the tracker model is tested by adding background noise to the data. Each scan consists of 20 randomly occurring blips. Figure 6-28 shows the modified BCoSaMP tracker model output in the case of four targets without any background noise.

Figure 6-28. Modified BCoSaMP tracker model output in radR

Figure 6-29 shows the tracks created in the presence of background noise.
Figure 6.29. Modified BCosAMP tracker model output with noise in radR

The performance of four tracker models developed is analyzed with the error rates of estimated tracks. Figure 6-30 shows the error rates in the case of single track. It can be seen that the modified BCosAMP has the least error compared to the other models.
Figure 6-30. Error rates of four tracker models for a single track

Figures 6-31 to 6-34 shows the error rates of each track in the simulated multi target environment for the four tracker models. It can be seen that the particle filter has the lowest error rate compared to other models with the increase in number of scans.
Figure 6-31. Error rates of track 1 for four tracker models in multi target environment

Figure 6-32. Error rates of track 2 for four tracker models in multi target environment
**Figure 6-33.** Error rates of track 3 for four tracker models in multi target environment

**Figure 6-34.** Error rates of track 4 for four tracker models in multi target environment
6.4 Simulation of Four Tracker Models with Marine Radar Data:

A balloon with a GPS is used to calibrate the radar as its path was tracked. The path of the balloon was observed to be random on the radar screen with very low velocity. The track of the balloon seen in the radar display is shown in Figure 6-35.

![Track of target moving away from radar](image)

**Figure 6-35.** Track of the balloon moving away from the radar

The tracker models are used to identify if the balloon is being tracked and outputs are shown in Figures 6-36 to 6-40. Due to low velocity and random motion of the balloon the tracks formed are not continuous and a lot of noise also gets included when the tracks are formed. The particle filter is the only model that estimates the target path closely with noise included along with its path. The other models just track the balloon at certain points along the track. The optimized values are given in Table 6.1.
### Table 6.1: Optimized blip processing parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Optimized Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise cutoff</td>
<td>0</td>
</tr>
<tr>
<td>Learning scans</td>
<td>15</td>
</tr>
<tr>
<td>Old stats weighting</td>
<td>0.95</td>
</tr>
<tr>
<td>Hot score</td>
<td>2.8</td>
</tr>
<tr>
<td>Cold score</td>
<td>2.8</td>
</tr>
<tr>
<td>Samples per cell</td>
<td>4</td>
</tr>
<tr>
<td>Pulses per cell</td>
<td>4</td>
</tr>
<tr>
<td>Min. blip samples</td>
<td>30</td>
</tr>
<tr>
<td>Max. blip samples</td>
<td>50000</td>
</tr>
<tr>
<td>Max. blip area</td>
<td>50000</td>
</tr>
<tr>
<td>Min. blip area</td>
<td>300</td>
</tr>
<tr>
<td>Min. angular span</td>
<td>2</td>
</tr>
<tr>
<td>Max. angular span</td>
<td>-1</td>
</tr>
<tr>
<td>Min. radial span</td>
<td>2</td>
</tr>
<tr>
<td>Max. radial span</td>
<td>-1</td>
</tr>
</tbody>
</table>

Tracker models with recommended blip processing parameters given in Chapter 3 and with a set of optimized values for blip processing are analyzed. Variation in blip processing parameters does not affect tracking significantly. BCoSaMP model does not track the target when the balloon is moving away from the radar for optimized blip processing parameters.
**Figure 6-36.** Kalman filter tracking of the balloon

**Figure 6-37.** Particle filter tracking of the balloon
Figure 6-38. BCoSaMP tracking of the balloon

Figure 6-39. Modified BCoSaMP tracking of the balloon
Nearest neighbor model is not discussed because the tracks were very sensitive to noise that the track of the balloon was overlapped with many false alarms making the track of the balloon undetectable.

Figure 6-41 shows the track of the balloon moving towards the radar. Figures 6-42 to 6-46 show the tracks detected by the tracker models.
Figure 6-41. Track of the balloon moving towards the radar

Figure 6-42. Kalman filter tracking of the balloon
**Figure 6-43.** Particle filter tracking of the balloon

**Figure 6-44.** BCoSaMP tracking of the balloon
Figure 6-45. Modified BCoSaMP tracking of the balloon

Figure 6-46. Multi-frame Correspondence tracking of the balloon
Tracker models are evaluated using marine radar data collected with T-bar antenna in the vertical mode. The number of targets identified and the false alarm rates of the tracker models are compared. Figure 6-47 shows the number of targets detected.

![Target Detection with Marine Radar Data](image)

**Figure 6-47.** Number of targets detected

The computation time of various algorithms are given in Table 6.2:

<table>
<thead>
<tr>
<th>Tracker Model</th>
<th>Computation Time (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-Frame</td>
<td>24</td>
</tr>
<tr>
<td>Kalman filter</td>
<td>75</td>
</tr>
<tr>
<td>Particle Filter</td>
<td>208</td>
</tr>
<tr>
<td>BCoSaMP</td>
<td>120</td>
</tr>
<tr>
<td>Modified BCoSaMP</td>
<td>180</td>
</tr>
</tbody>
</table>

The computation time is calculated for marine radar data that was collected for 2 hours 32 minutes.
6.5 Summary:

The performance of various tracker models is discussed by simulating tracker models with various data. Nearest neighbor tracker model is not discussed as it contains a lot of noise and true tracks are lost when there is high amount of noise. The particle filter has high computation time due to the use of large number of particles however in marine radar data it was the only method to track the target accurately. Particle filter has the highest detection rates but is affected by noise. The BCoSaMP also gives reliable output as the false alarms are less and has detection rates close to particle filter. BCoSaMP can be used when less computation time is required. These models help in increasing the data processing reliability and accuracy compared to the current tracking models in radR.
Chapter 7

Conclusion and Future Work

7.1 Conclusion:

Radar tracking is used in the study of bird flight patterns. The work will help in understanding the behavior of bird towards man-made structures. Tracking birds is challenging in radar data due to noise, clutter and wide coverage area of the radar. Four tracker models were developed considering a linear system, nonlinear system and sparse data conditions to identify the most apt tracker model for radar data. The tracker models performance is analyzed based on three types of data sets which are:

- **One simulated track**: BCoSaMP and modified BCoSaMP have the least error rate. The newly developed model has 1% error and gives the most ideal output.

- **Four simulated tracks with noise**: In simulated data with four tracks show various error rates for all models. The least error rate varies for each track. It is also shown that addition of noise results in induced error in the created tracks.

- **Calibration of Marine radar data via balloon movement**: The balloon was detected well by the particle filter compared to other models. The number of targets identified is high for particle filter but it is very sensitive to noise. The BCoSaMP model also has reliable output as the identification rates are closer to particle filter and has significantly less false alarms.
The analysis with various data shows that particle filter has the most optimum tracking and high identification rates compared to the multi-frame model in radR. BCoSaMP also gives reliable results based on the fact that there is significant decrease in false alarm rates. The tracker models should be selected based on the data type and presence of noise. These plugins are developed to aid in data processing ensuring higher reliability and accuracy.

7.2 Future Work:

Tracker models can be further improved to reduce the false alarm rates. False alarms can be reduced by defining a dynamic system to identify the noise levels. However implementation for radar data can be challenging owing to the fact that noise levels are time variant. In these models, only nearest neighbor data association is used this can also be improved by using better data association techniques and additional gating techniques for thresholding. The computation time can be reduced by parallelizing the tracker algorithms.
References


[8] Radio Telemetry and bird movements from:

http://old.nationalwind.org/publications/wildlife/avian95/avian95-08.htm

[10] James P. Ferry and J. Oren Bumgarner, Community Detection and Tracking on Networks from a Data Fusion Perspective, Jan 6, 2012, from:

http://library.fws.gov/Circulars/Mig_of_Birds_16_98.pdf


[16] Sonobat from: http://sonobat.com


Strike 2005, Vancouver, Canada, August 2005, from:
http://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1006&context=birdstrike2005


[34] Ziv Yaniv, Median Filtering, from:

[35] Chin-Ya Huang and Mon-Ju Wu, Image Segmentation, 2006, from:
http://homepages.cae.wisc.edu/~ece533/project/f06/huang_wu_rpt.pdf


[41] Study Components Available from: http://www.accionana.com/getattachment/837f72a6-dc5c-4eaa-995a-20794aa247bd/


[53] RadarSample Usage Notes Available from: Russell Technologies Inc.

[54] radR Available from: http://www.radr-project.org/


[76] Inseok Hwang, Hamsa Balakrishnan, Kaushik Roy and Claire Tomlin, Multiple-Target Tracking and Identity Management in Clutter, with Application to Aircraft Tracking, American Control Conference, Vol. 4: p. 3422-3428.


[107] Richard Karlsson and Fredrik Gustafsson, Monte Carlo Data Association for Multiple Target Tracking, from: http://www.control.isy.liu.se/~fredrik/reports/01ieee.pdf


