NTNU – Trondheim
Norwegian University of
Science and Technology

Radar Detection, Tracking and Warning of Avian Targets at Airports

Reporting of potentially hazardous bird presence at airports using low cost magnetron Moving Target Detector (MTD) radar

John-Olav Brattebø

Master of Science in Electronics
Submission date: December 2014
Supervisor: Egil Eide, IET
Co-supervisor: Erik Løvli, Radian AS
Yngve Steinheim, FLO-IKT
Tore Smestad, FFI

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

To my children,
Elida and Viljar

Problem description

Radian AS in Kongsberg is currently working on new methods for improving the detection capability of the standard ship radar. The company has developed a PC-based platform in cooperation with SINTEF ICT for demonstration and testing of new applications. The methods are well suited for detecting moving objects over land and sea.

A specific and important application is the observation of birds near airports. In Norway there has been recorded 400-500 "Birdstrikes" per year, most of which are harmless. Some cases involve delays, costs and repairs. On January 2nd 2014, the Norwegian flight DY0165 hit a swan after departing from Avinor Airport Værnes. The collision resulted in a significant hole in the wing near the port engine. Avinor contacted in retrospect Radian AS, for information about using radar for the observation of birds. A test radar was established at Værnes Airport in April 2014.

Avinor's staff at Værnes need a tool to aid the discovery and observation of birds, where early warning is an important aspect. A radar is unaffected by visibility, light and most weather conditions and can report instances of birds around the clock. Radar images can be stored consecutively on the hard drive, which enables analysis of events and studies of bird movement patterns and activities.

Central to the development are automated methods of detection based on the MTD (Moving Target Detector). A PC-based demonstrator is under development. Initial experiences show that the demonstrator detects birds over land and sea.

In the master's thesis, the goal is to automate warning of birds by developing a bird trajectory detector and alarm. The bird trajectory detector shall identify bird tracks by recognizing birds' typical motion patterns. The radar antenna has a rotation period of approximately 2.5 s. By observing the birds' movement during several antenna rotations, it is possible to separate the birds from both false detections and other moving objects (cars, aircraft). The master's thesis should contain a theoretical analysis of the bird trajectory detector's performance and it should be tested and optimized using real radar signal recordings.

The alarm / early warning function builds on the bird trajectory detector and shall give warning in accordance with certain criteria such as when birds are crossing specified limits in distance within given angular ranges. The probability of alarms / false alarms must be acceptably high / low for the users.

Supervisor	:	Adjunct Associate Professor Egil Eide NTNU-IET egil.eide@iet.ntnu.no, +47 73 59 14 13
External supervisors	:	Erik Løvli, Radian AS erik@radian.no, +47 32722340 / +47 90971793
		Chief Engineer Yngve Steinheim, SINTEF-ICT, FLO-ICT yngstein@gmail.no, +47 99 04 32 97
		Researcher Tore Smestad, Norwegian Defense Research Establishment (FFI) tore.smestad@ffi.no tlf. +47 93803618

Oppgavetekst

Radian AS på Kongsberg arbeider for tiden med nye metoder for bedring av deteksjonsevne i standard skipsradar, og har utviklet en PC-basert plattform i samarbeid med SINTEF-IKT for demonstrasjon og testing av nye anvendelser. Metodene er godt egnet til å detektere objekter i bevegelse over land og sjø.

En konkret og viktig anvendelse er observasjon av fugl nær flyplasser. I Norge registreres 400-500 "Birdstrikes" i året, der de fleste er ufarlige. Noen tilfeller medfører forsinkelser, kostnader og reparasjoner. Ved Avinor Lufthavn Værnes møtte Norwegians DY0165 en svane etter avgang 2 januar 2014. Kollisjonen medførte et betydelig hull i vingen ved babord motor. Avinor tok i ettertid kontakt med Radian AS, for å få informasjon om bruk av radar til observasjon av fugl. En forsøksradar ble etablert på Værnes i april 2014.

Avinors fagfolk på Værnes ønsker et hjelpemiddel som kan lette arbeidet med å oppdage, og observere fugl, der tidlig varsling er et viktig aspekt. En radar er upåvirket at sikt, lys og de fleste værforhold og kan rapportere forekomster av fugl døgnet rundt. Radarbildene kan lagres fortløpende på harddisk, som muliggjør analyser av hendelser og studier av fuglers bevegelsesmønstre og aktiviteter.

Sentralt i utviklingen står automatiserte metoder for deteksjon basert på MTD (Moving Target Detector). En PC basert demonstrator er under utvikling på Værnes. Innledende erfaringer viser at demonstratoren detekterer fugl over land og sjø.

I masteroppgaven er målet å automatisere varsling av fugl ved å utvikle en fuglebanedetektor og alarm. Fuglebanedetektoren skal identifisere fuglebaner ved å gjenkjenne fuglenes typiske bevegelsesmønstre. Radaren har en antennerotasjonstid på ca 2,5sek. Ved å observere fuglenes forflytning over noen antennerotasjoner er det mulig å skille fugl fra både falske deteksjoner og andre bevegelige objekter (biler, fly). Det skal foretas en teoretisk analyse av fuglebanedetektorens ytelse og den skal testes og optimaliseres ved hjelp av reelle signalopptak.

Alarmfunksjonen bygger på fuglebanedetektoren og skal gi varsel etter visse kriterier som at fugl passerer spesifiserte grenser i avstand innenfor gitte vinkelområder. Sannsynligheten for alarm / falsk alarm må være akseptabelt høy / lav for brukerne.

Faglærer	:	Førsteamanuensis II Egil Eide NTNU-IET egil.eide@iet.ntnu.no tlf.73 59 14 13
Veiledere	:	Erik Løvli, Radian AS erik@radian.no tlf. 32 72 23 40 / 90 97 17 93
		Oing. Yngve Steinheim, SINTEF-IKT og FLO-IKT yngstein@gmail.no tlf. 99 04 32 97
		Forsker Tore Smestad, Forsvarets Forskningsinstitutt (FFI) tore.smestad@ffi.no tlf. 93 80 36 18

Summary

During the last years, Radian AS has in close cooperation with SINTEF-ICT been working on new methods for improving the detection capability of the standard civil marine radar. A coherent-on-receive radar demonstrator with a PC-based platform has been developed and the methods are found well suited for detecting small moving targets over land and sea. An operational birdstrike (aircraft-bird collision) avoidance radar system based on this technology has recently been deployed at Værnes Airport, Norway. This system uses a standard civil marine radar, a computer and methods of detection based on a Moving Target Detector (MTD) processor. The resulting radar video is broadcast to both the airport's security central and Air Traffic Control (ATC) Tower to allow initiation of precautionary measures. Since the current system demands manual interpretation and constant monitoring of the MTD radar video, there is need for an Automatic Detection and Tracking (ADT) system and a warning system that draws attention to specific situations.

In this Master's Thesis, methods for radar detection, tracking and Early Warning (EW) of avian targets at airports are investigated. The work is based on theoretical analysis, testing with real radar measurements and simulation that incorporates real measurements. The methods of detection are improved by modification of the MTD processor. A specialized, batch-processing tracker called a *Bird Flight Path Detector (BFPD) Tracker* is developed and implemented to automatically identify and track birds in the airport vicinity. An EW functionality is also developed and implemented to monitor the resulting tracking data and give warning of potentially hazardous situations in advance. Furthermore, the performance of the proposed tracker and the resulting total system is optimized, analyzed and evaluated.

The detection capability of the radar is found sufficient for use in a birdstrike avoidance application. According to performed theoretical calculations, the existing radar system is able to detect a single goose at about 4 km with a probability of detection of $P_d = 0.7$ and a probability of false alarm of $P_{fa} = 10^{-3}$. Testing shows that in practice, multiple flocks (of varying numbers) of geese are detected consistently enough to allow continuous tracking by the BFPD Tracker up to about 4 km in range over both land and sea. It is also shown that the BFPD Tracker is be able to identify and follow all of the important bird presence while simultaneously exhibiting a probability of true (caused by birds) confirmed track establishment around 70% and a probability of true batch association around 96-100%. The latter is hence a good indicator of true bird presence. Simulation experiments show that the total system is able to detect an avian target roughly the size of a single goose at ranges of about 3 km with $P_d \approx 0.875$ and $P_{fa} \approx 10^{-3}$. Simulation also shows that the BFPD Tracker is able to track this target *continuously* up to 4 km over sea with an RMS error of 2.4 m in range, 0.08° in azimuth and 1.6 m/s in velocity.

The EW functionality is found capable of identifying and giving warning of almost all manually identified potentially hazardous situations while showing a very low probability

of false warning ($\ll 1\%$). Long-term testing and corresponding knowledge of the true bird activity is needed to accurately estimate the probability of false warning, but this work indicates that the BFPD Tracker and EW function may be suited for tracking and EW application in an ATC Tower. Near real-time processing is deemed feasible with standard computing hardware and if the system is developed further it may help mitigate overall birdstrike risk and contribute to improved safety in aviation.

Sammendrag

Radian AS har de siste årene, i tett samarbeid med SINTEF-IKT, arbeidet med nye metoder for bedring av deteksjonsevne i standard skipsradar. En coherent-on-receive demonstrator radar med en PC-basert plattform har blitt utviklet, og metodene er velegnet for å detektere små bevegelige mål over land og sjø. Et operativt radarsystem for å unngå birdstrikes (fly-fugl kollisjoner) som er basert på denne teknologien har nylig blitt utplassert på Værnes Lufthavn. Dette systemet benytter en standard skipsradar, en PC og deteksjonsmetoder som er basert på en Moving Target Detector (MTD) prosessor. Den resulterende radar-videoen blir kringkastet til bde flyplassens vaktsentral og kontrolltårn for å muliggjøre iverksettelse av preventive tiltak. Det nåværende systemet krever manuell tolkning og konstant overvåkning av MTD radar-videoen, som skaper et behov for et Automatisk Deteksjon og Tracking/sporing (ADT) system samt et varslingssystem for å trekke oppmerksomhet mot bestemte situasjoner.

I denne masteroppgaven undersøkes metoder for radar deteksjon, tracking (sporing) og varsling av fugl på flyplasser. Arbeidet er basert på teoretisk analyse, testing med reelle radarmålinger og simulering som inkorporerer reelle målinger. Deteksjonsmetodene er forbedret ved å modifisere MTD prosessoren. En spesialisert, batch-prosesserende tracker kalt en *fuglebanedetektor* (Eng: *Bird Flight Path Detector (BFPD) Tracker*) er utviklet og implementert til å automatisk identifisere og spore fugl i flyplassens nærområde. En varslingsfunksjon/alarm er også utviklet og implementert til å overvåke den resulterende sporingsdataen og varsle om potensielt farlige situasjoner før de oppstår. Videre er ytelsen til den foreslåtte trackeren og det resulterende totale systemet optimalisert, analysert og evaluert.

Radarens deteksjonsevne er funnet tilstrekkelig for anvendelse i en applikasjon for å unngå birdstrikes. I følge utførte teoretiske beregninger kan det nåværende radarsystemet detektere en enkelt gås på ca. 4 km med en sannsynlighet for deteksjon på $P_d = 0.7$ og en sannsynlighet for falsk alarm på $P_{fa} = 10^{-3}$. Undersøkelsene viser at i praksis er flere flokker (av ukjent antall) med gjess detektert ofte nok til å tillate kontinuerlig tracking med fuglebanedetektoren opp til avstander rundt 4 km både over land og sjø. Videre vises det at fuglebanedetektoren klarer å identifisere og spore all den viktige tilstedeværelsen av fugl mens den utviser en sannsynlighet for etablering av sann (forårsaket av fugl) bekreftet track rundt 70%, og en sannsynlighet for sann batch-assosiasjon rundt 96-100%. Sistnevnte er da en god indikator på sann tilstedeværelse av fugl. Simuleringene viser at det totale systemet kan detektere et fuglemål i størrelsesorden en enkelt gås på avstander rundt 3 km med $P_d \approx 0.875$ og $P_{fa} = 10^{-3}$. Simuleringene viser også at fuglebanedetektoren klarer å spore dette målet *kontinuerlig* ut til 4 km over sj med en RMS feil på 2.4 m i avstand, 0.08° i asimut og 1.6 m/s i hastighet.

Varslingsfunksjonen vises i stand til å identifisere og gi varsel om nesten alle de

manuelt identifiserte situasjonene som er potensielt farlige med en veldig lav sannsynlighet for falskt varsel ($\ll 1\%$). Testing over lengre tid med tilhørende kunnskap om den sanne fugleaktiviteten kreves for å nøyaktig kunne estimere sannsynligheten for falskt varsel, men dette arbeidet indikerer at fuglebanedetektoren og varslingsfunksjonen kan vre egnet for sporings- og varslings-applikasjon i et kontrolltårn. Det anses gjennomførbart å oppnå nær sanntidsprosessering med standard data-hardware og hvis systemet blir videreutviklet kan det hjelpe med å forbedre den generelle risikoen for birdstrike og bidra til økt sikkerhet i luftfart.

Preface

This thesis is submitted in fulfillment of the requirements for the degree of master of science (MSc) at the Department of Electronics and Telecommunications, Norwegian University of Science and Technology (NTNU). The work was carried out in the period June 2014 to December 2014, under the supervision of Adjunct Associate Professor Egil Eide (NTNU-IET).

During the course of this work, and during project work the preceding year at NTNU, the author has been involved in a radar project by Radian AS that seeks to apply their developed radar technology in a birdstrike avoidance radar at Norwegian airports. In 2013, the first experimental coherent-on-receive birdstrike avoidance radar (based on a civil marine radar) was deployed at Alta Airport, Norway. Its operational function was as a weather radar, but it also provided the opportunity to test the technology in a birdstrike avoidance application, which resulted in the author's project report of 2013.[13]

In 2014, the author has also participated in the process of deploying a second radar system at the larger Værnes Airport through multiple meetings and presentations with Avinor (The governmental company in charge of Norwegian airports), the Norwegian Aircraft/Bird-Committee (Norsk Fly/Fugl-utvalg (NFF)), Værnes Airport Air Traffic Control (ATC, TRD-TWR) and Værnes Airport Fire and Rescue (TRD-PBR). This radar was deployed and started operational testing in birdstrike avoidance on April 8th, 2014, and has supplied the measurements that are used in the work of this thesis.

Acknowledgment

First, I would like to express my sincere gratitude to my excellent advisers/supervisors Erik Løvli (Radian AS), Yngve Steinheim (FLO-IKT), Tore Smestad (FFI) and Egil Eide (NTNU) for their guidance, challenging and constructive mentoring, informative technical discussions and for including me in the project. I have learnt very much from you!

I would also like to thank Avinor and the crew at Værnes Airport Air Traffic Control (TRD-TWR) and Fire and Rescue (TRD-PBR) for their interest and commitment to the project, for facilitating and incorporating the birdstrike avoidance radar into their operations. It has brought extra meaning and motivation to my work to have such an important application available close to present, to possibly have an impact on real-life operations and contribute to aviation safety at a major Norwegian airport.

Last, but not least, I thank my wife Marthe Brattebø for her unshakable support, my beautiful daughter Elida Marie Brattebø and my lovely baby boy Viljar Brattebø who was born in the middle of my work on this thesis. You are the light of my life and I love you!

J.O.B.

Table of Contents

List of Tables

List of Figures

Abbreviations

ADT	=	Automatic Detection and Tracking
BFP	=	Bird Flight Path
BFPD	=	Bird Flight Path Detector
CA	=	Normalized Clutter Attenuation
CA-CFAR	=	Cell-Averaging CFAR
CACM-CFAR	=	Cell-Averaging Clutter Map CFAR
CCL	=	Connected Component Labeling
CFAR	=	Constant False Alarm Rate
CM	=	Clutter Map
CoM	=	Center of Mass
COHO	=	Coherent Oscillator
CUT	=	Cell Under Test
CPI	=	Coherent Processing Interval
DMHC	=	Delete Measurements in High Clutter areas
DMUA	=	Delete Measurements in Uninteresting Areas
EW	=	Early Warning
FFT	=	Fast Fourier Transform
IF	=	Intermediate Frequency
IID	=	Independent and Identically Distributed
JPDA	=	Joint PDA
KB	=	Knowledge-Base
MH	=	Multiple Hypotheses
MHT	=	Multiple Hypotheses Tracker
MTI	=	Moving Target Indicator
MTT	=	Multiple Target Tracking
MTD	=	Moving Target Detector
OS-CFAR	=	Ordered Statistic CFAR
OSCM-CFAR	=	Ordered Statistic Clutter Map CFAR
pdf	=	probability density function
PDA	=	Probabilistic Data Association
PPI	=	Plan Position Indicator
PRI	=	Pulse Repetition Interval
PRF	=	Pulse Repetition Frequency
PS	=	Prolate Spheroid

Radar	=	Radio detection and ranging
RCS	=	Radar Cross Section
SCR	=	Signal to Clutter power Ratio
SCV	=	Subclutter Visibility
SF	=	Score Function
SNR	=	Signal to Noise power Ratio
SNCR	=	Signal to Noise and Clutter power Ratio
STALO	=	Stable Local Oscillator
SWEM	=	Sphere of Water with Equal Mass
TO	=	Time of Observation

Nomenclature

General Radar

R_{max}	=	Maximum radar range [m]
R	=	Envelope of received signal
P_{av}	=	Average transmitter power [W]
G	=	Antenna gain
A	=	Antenna area [m^2]
ρ_a	=	Antenna aperature efficiency
σ_t	=	Radar Cross Section (RCS) of the target
n	=	Number of pulses integrated
$E_i(n)$	=	Integration efficiency
F	=	Propagation factor
α	=	Attenuation coefficient [nepers per unit distance]
k_B	=	Boltzmann's constant = 1.38×10^{-23} J/Kelvin
T_0	=	Standard temperature = 290 K
F_n	=	Receiver noise figure
B	=	Receiver bandwidth [Hz]
τ	=	Pulse width [s]
f_p	=	Pulse Repetition Frequency (PRF) [Hz]
$(S/N)_1$	=	Required minimum SNR as if detection were based on a single pulse
L_f	=	Fluctuation loss (for a Swerling target model)
L_s	=	System loss
σ_t	=	Radar Cross Section (RCS) of the target [m^2]
$(S/C)_{min}$	=	Minimum discernible Signal to Clutter Ratio (SCR)
σ^0	=	Clutter RCS per unit area
θ_B	=	Two-way antenna azimuth beamwidth [rad]
c	=	Speed of propagation [m/s]
ψ	=	Grazing angle [rad]
Ψ_0	=	Mean noise power [W]
$exp(x)$	=	e^x

CFAR

α	=	CM update coefficient
k	=	OS-CFAR rank
K_t	=	Threshold multiplier
N	=	Reference window size
P_d	=	Probability of detection
P_{fa}	=	Probability of false alarm
V_T	=	Threshold
x_i	=	Cell number i in the reference window
Z	=	Background estimate

Tracking

N	=	Batch length
M	=	Required minimum number of observations
TO	=	Time of Observation
T_{HC}	=	Threshold for High Clutter (HC) classification of a cell
v_{min}	=	Minimum allowed absolute speed
v_{max}	=	Maximum allowed absolute speed
v_{tol}	=	Maximum allowed change in speed
a_{tol}	=	Maximum allowed change in course
n_m	=	Maximum allowed number of misses
θ_1	=	Initial difference in course
θ_2	=	Overall difference in track course

Chapter 1

Introduction

1.1 Background

During the last decade, a small radar company called Radian AS [47], and the Norwegian research organization SINTEF [55] have developed a method to perform Doppler processing of the signal from the abundant, but inherently non-coherent civil marine radars. These radars use a high power oscillator called a magnetron to generate the pulsed carrier waveform of the radar. This technology is durable, reliable, cheap and easy to maintain, but does traditionally not allow extraction of Doppler information due the magnetron having a random start phase of each pulse. Doppler processing of the signal from such a radar is made possible by preprocessing and phase-shifting of each pulse to make them coherent, constituting what is known as a pseudocoherent, or coherent-on-receive, radar system. Other parties (e.g. The Norwegian Defense Research Establishment (FFI), The ElectroScience Laboratory / Ohio State University) have also confirmed the feasibility and practicality of making a high power coherent-on-receive radar from a civil marine radar at low cost. Documentation of the process and methods may be found in [59], [8], [36] and [35].

Through Doppler processing and multi-channel filtering, the developed pseudocoherent radar system achieves a positive Clutter Attenuation (CA), which allows the radar to obtain Sub-Clutter Visibility (SCV). This means that the radar can detect a moving target within a spatial resolution cell containing stationary terrain features whose echo is far more powerful than that of the target. This pseudocoherent expansion opens up a new realm of possible applications for civil marine radars. One of these applications is a birdstrike avoidance radar at airports.

A birdstrike is the name of the event when an aircraft crashes with a bird. Though a bird might seem as a small threat for a commercial jet airliner at first glance, the potential threat is very serious. Globally, wildlife strikes have resulted in at least 255 fatalities and the destruction of over 243 aircraft since 1988, of which about 97 % is due to birdstrikes. [21] One of the most well-known events that illustrate the dangers of birdstrikes is the US Airways Flight 1549 incident in 2009 when a commercial jet hit a flock of geese during a

climb from La Guardia Airport, New York City. The flock struck the plane at 3000 ft and took out both engines, forcing the aircraft down over Manhattan. The pilots miraculously managed to perform a safe landing on the Hudson river, bringing quite a lot of public attention to the incident. [62] The birdstrike incident that has caused the greatest number of fatalities was on October 4, 1960, when a Lockheed L-188 Electra flying as Eastern Air Lines Flight number 375 flew through a flock of starlings during take-off from Logan International Airport in Boston, damaging all four engines. The aircraft subsequently crashed into the Boston Harbor, and 62 of the 72 passengers died. After this incident, the United States Federal Aviation Administration (FAA) started to introduce minimum bird ingestion requirement standards for aircraft engines.[64][66]

Figure 1.1: A Thomsonfly Boeing 757 right after suffering a double birdstrike (supposedly two herons) to the right engine causing engine failure during take-off from Manchester Airport in April 2007. The engine started to shoot large jets of flames, but was shut off. The aircraft turned and landed safely with one engine.[64][7]

Birdstrikes are fairly common events in Norwegian civil aviation, counting over 500 incidents during the past year, most of which are collisions with small birds. The standard procedure after a birdstrike, whether the aircraft was seemingly damaged or not, is to make a precautionary landing at the nearest airport and ground the aircraft until it has been repaired and cleared for flight again. Flocks of large birds such as geese obviously form a greater threat than a single sparrow, but all birdstrikes cost a lot of money in the aftermath of an incident. A 2014 report by the US FAA [21] states the projected total monetary loss due to birdstrikes with civil aircraft in the US amounts to $900 million inflation-adjusted U.S. dollars annually.

Since almost all incidents happen during take-off, approach or landing, and often during low-visibility conditions at dusk or dawn, a birdstrike avoidance radar would possibly be able to significantly mitigate the risk of birdstrikes by identifying the most hazardous situations in advance.[39]

January 2nd, 2014, a Norwegian jet airliner struck a large swan shortly after take-off from Værnes Airport. The plane was hit near the right engine, causing large damage to the hull. The plane immediately went around and landed safely at Værnes. After this event, Avinor [3] (the government-owned firm in charge of Norwegian airports) sought new measures to mitigate the risk of birdstrikes. By the initiative of Avinor, an experimental, but operational birdstrike avoidance radar has recently been deployed by Radian AS at Værnes Airport, Norway. This radar is a pseudocoherent (coherent-on-receive) radar based on a

standard civil marine radar. The radar is shown in Figure 1.2.

Figure 1.2: The birdstrike avoidance radar at Værnes Airport (deployed April 8th, 2014) used for measurements in this thesis. Here shown with an aircraft on the tarmac in the background.

Since the summer of 2013, the author has cooperated (as a student) with Radian and SINTEF in the development of a birdstrike avoidance radar.[13] In the fall of 2013, the first radar was placed at Alta Airport, Norway. The measurements from this radar were used for optimizing the detection of birds. As the radar system now is intended to function as an operational asset for the Air Traffic Control (ATC) Tower at Værnes, the need for an Automatic Detection and Tracking (ADT) system arises.

Therefore, the goal of the work in this thesis is to develop a radar tracker that automatically identifies and locates potentially hazardous birdstrike situations and thus may bring them to the operator's attention. This application of the tracker does not emphasize the high-precision tracking of individual targets. It rather prioritizes the ability to identify with high reliability (probability of trueness) the presence of Bird Flight Paths (BFPs) among the radar detections and false alarms. Therefore, the tracker may be accurately called a Bird Flight Path Detector (BFPD). However, since the term *tracker* is well established in the field, and not misleading in this context, it will be used to describe the developed system throughout this thesis by referring to it as a *BFPD Tracker*.

1.2 The challenge of low-cost ADT of birds

The risk of birdstrikes is a serious issue, a challenge and a growing problem in current civil and military aviation. Since there initially has been no effective method to mitigate this risk, the first approach of many official aviation authorities around the world has been to enforce strike reporting. The initiation, incorporation and funding of wildlife hazard mitigation research by governmental agencies has also been a common response. This includes research, development and evaluation of avian radars for use as birdstrike avoidance radars at airports.

In 2001, the US FAA began working with the US Air Force to develop an avian ADT radar system, but refocused their research in 2006 to evaluate the capability of commercially available, low-cost avian radars. [21, p.80] In 2010, they published an Advisory

Circular that states requirements and guidelines on how American airports can select, procure, deploy and manage an avian radar system for mitigation of birdstrike risk. This advisory, which may be found in [45], also firmly establishes the need for a tracking capability in such an application.

For such a system to be justified financially, it has to save more money in avoided birdstrike costs than the costs of the system itself. On the other hand, there is the consideration of potential for loss of human life and injury, which may justify a net financial cost. However, birds are very challenging radar targets because of their small size, low altitude and dynamic nature. It is firmly established that radars may detect and track birds, but the nature of birds as radar targets may cause poor performance unless the radar system is excellent. Several radar systems with sufficient performance for long range ADT of birds exist, such as Doppler weather-radars or military-grade Pulsed Doppler radars, but these are very expensive. The overall conclusion is that there is a need for research and development of a low-cost ADT avian radar for this application, which may be observed in research from the US to the Netherlands and China.[43] [18] [31]

Because of this, several solutions and companies delivering low-cost ADT avian radars have emerged. Some utilize civil marine radars in different ways such as the experimental system proposed in [43] and the avian radar system of the company Robin Radar [29]. Examples of other companies are DeTect Inc. [28], who utilizes a coherent solid-state system in their avian radar called MERLIN, and Accipiter Radar who have supplied avian radars to the FAA research [46]. [50]

Figure 1.3: A British Airways aircraft flying through a flock of birds during approach and landing.

The radar system utilized in the work of this thesis stands out from the rest because it is both coherent, low-cost and utilizes a high transmitter power of 25 kW. Because the radar is coherent, it can perform Doppler processing, achieve a positive Sub-Clutter-Visibility (SCV) and detect small target echoes embedded in heavy clutter. At the same time, the high transmitter power allows for high pulse energy even with a short pulse duration, which is needed to maintain a high detection performance at close ranges.

In general, for any radar waveform, detection performance is dependent on the total pulse energy. The pulse energy is given by

$$E = P_t \tau \tag{1.1}$$

where

E = Pulse Energy [J]
P_t = Transmitter Power [W]
τ = Pulse duration [s]

Equation 1.1 states that with a lower transmitter power one needs a longer pulse duration to obtain the same pulse energy. This radar, when utilizing a transmitter power $P_{t1} = 25$ kW and a pulse of duration $\tau_1 = 150$ ns, achieves a pulse energy of $E_1 = 3.75$ mJ. But a radar is only capable of detecting targets with this full pulse energy at a minimum range of $c\tau/2$. At target ranges closer than this, the echos from targets will reach the radar before transmission of the pulse is done. Then, because the receiver is off when the transmitter is on, a portion of the pulse energy is lost. Targets may still be detected, but utilizing the full available pulse energy is preferable. This loss of pulse energy happens to a smaller extent for radars with high P_t, and is therefore an advantage for high power systems such as the civil marine radars the radar in this thesis is based on.

1.3 Brief introduction to target tracking

Target tracking is the process of keeping an updated record of the position, speed and other information about the targets of interest to a radar system based on measurements gathered by one or more sensors. Radar target tracking is at the same time an expression of the belief that one or more targets actually are present and detected by the radar system. There is usually generated and maintained a track-file with one track for every hypothesized unique target. Such a track in the track-file may consist of a range of different measurements (e.g. position, Doppler, size) and modeled parameters (e.g. velocity, acceleration, maneuvering) that describe the target state. Target tracking is a diverse multidisciplinary field that can be divided into many branches for which the applied techniques and technology may vary greatly.

First, tracking systems may perform single- or multi-target tracking depending on how many targets are observed and tracked simultaneously. Track-While-Scan (TWS) systems perform the tracking and scanning for targets simultaneously, whereas non-TWS systems perform scanning and tracking subsequently. Radar target tracking is also classified in terms of whether the targets are cooperative and use transponders (short for transmitter-responder) such as in civilian Secondary Surveillance Radars (SSR), or whether the targets are non-cooperative, in which case the targets do not transmit any beacon-signal and do not respond to a radar transmission. [68]

Single-scan tracking systems consist of systems that use the measurements from a single scan of the geographical search area to update the tracks, while multi-scan systems scan the search area multiple times before performing tracking on all acquired measurements. Multi-scan systems will thus be able to obtain multiple observations of the same target and may also be called batch processing systems.

Track Before Detect (TBD) systems will start tracking of measurements before declaring detections based on a detection threshold. On the other hand, Detect Before Track (DBT) systems will apply a detection threshold to extract measurements called plots which then are input to the tracker. DBT is the classical approach, while the TBD technique was

developed especially for situations of low target Signal to Noise power Ratio (SNR).[2][41] Further system classification consists of soft-decision systems, that allow several target observations from the same scan to be associated to the same track record, and hard-decision, that only update a track with one target observation (measurement). In the latter, the algorithm typically chooses the observation that satisfies some maximum likelihood requirement. [4]

Knowledge-based tracking (KBT) is another subset of target tracking, where the system utilizes a Knowledge Base (KB) consisting of information such as terrain profiles, environmental maps and characteristics of the target. In this type of system, a fusion of data from the KB and the radar sensors forms the basis for detection and tracking. [9]

The proposed BFPD Tracker developed in this thesis may be called a TWS, non-cooperative, multi-target, multi-scan, DBT, hard-decision, KBT system.

1.4 Objective and Research questions

The problem description was given at the very beginning of this document. Based on this description, the objective of the master's thesis is as follows:

Primary	Automate the warning of birds by developing a bird trajectory detector and alarm that identifies bird tracks by recognizing typical avian motion patterns with an acceptable probability of false alarm/warning.
Secondary	Perform a theoretical analysis of the bird trajectory detector's performance, test and optimize detection and tracking using real radar recordings.

In order to accomplish these objectives, the author has chosen to develop a batch processing, specialized system called a BFPD (Bird Flight Path Detector) Tracker. This system identifies avian movement in the radar detection video and warns of potentially hazardous birdstrike situations. During development, testing and evaluation, there are other issues that emerge as interesting research questions. These research questions will serve as a framework supporting the main objective, and are listed below.

Research Questions

- Is batch processing and the concept of a ”bird trajectory detector” suited for the intended application?
- How does the adjustment of different parameters affect performance and what are the system's critical parameters?
- What are ”potentially hazardous birdstrike situations” and how sinister or how trivial situations can be reliably identified?

Chapter 2

Background Theory

This chapter will give an overview of the most relevant background theory at the basis of this thesis. The first section of this chapter gives some basic understanding of some fundamental theory of radars and clutter. The next section describes in more detail the theoretical basis of radar receivers and target detection. Thereafter follows a section on Doppler processing theory before a section is devoted a short study of birds as radar targets. Then, Section 2.5 will explain some theory on Multiple Target Tracking (MTT), and lastly, Section 2.6 will provide the theoretical basis of Markov Chains with an example of application and calculation.

2.1 Radar and clutter

A radar uses a transmitter to illuminate its surroundings with electromagnetic radiation and analyzes the echoes returning to the radar receiver to obtain information about the position and movement of targets within the search area. Using an antenna with a narrow beam, the radar examines a small sector at a time for increased angular resolution. The antenna is typically rotated or the beam steered in some manner to cover the whole search area. Hence, information about a target's range is derived from the target echo's time of arrival and angular information is derived from the antennas position at the time of reception. Velocity and movement information is derived from a target's Doppler shift and/or from the geometry and time of different observations of the same target. Information about a target's size or shape is also possible to derive from measurements if the radar resolution is sufficient compared to the target's size.

As the antenna is rotated and sweeps over a target, the time the antenna's main beam spends on a target is called the dwell time or the Coherent Processing Interval (CPI). During this time, there are usually fired many pulses of radiation, resulting in n returning echoes/pulses from a single target in one sweep. This number is often called hits per scan. The echoes from these n pulses are integrated/summed to increase the received signal's Signal to Noise power Ratio (SNR).

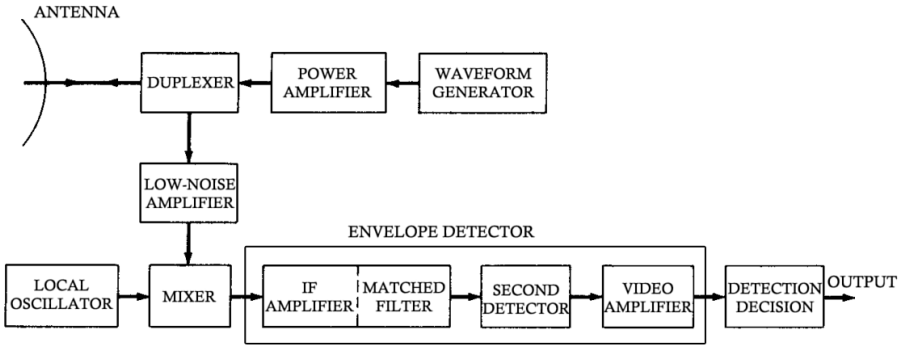


Figure 2.1: Simple block diagram of a radar with a superheterodyne receiver.[57, 1.2]

The returning energy backscattered from the surroundings consists of target echoes, clutter and noise. Clutter is the sum of echoes from all reflectors and scatterers that are not of interest by a particular radar system, and is thus by definition always an unwanted part of the received signal.

Figure 2.1 shows an elementary block diagram of the subsystems usually found in a radar. This particular system utilizes a power amplifier in the transmitter, but in marine radars like the one used in this thesis this is replaced by a magnetron, which is a power oscillator.

The duplexer is a device that allows the transmitter and the receiver to use the same antenna. The duplexer directs the transmitted energy to the antenna without entering and burning out the sensitive receiver and directs the received energy to the receiver and not the transmitter. The duplexer is also often called a circulator.

The lower blocks of Figure 2.1, starting with the Low-Noise Amplifier (LNA), constitute a superheterodyne receiver. The receiver mixes the signal to an Intermediate Frequency (IF) and amplifies it to efficiently perform the process of target detection. The intermediate frequency (IF) amplifier, the second detector and the video amplifier in cascade work as an envelope detector. To detect the presence of a target, the envelope of the received signal needs to cross a certain level of threshold. An envelope detector is either linear-law, which outputs the envelope as the signal's amplitude, or square-law, which outputs the square of the signal. A square-law detector is often assumed for its analytical simplicity. Linear-law detectors are however more practical, and provide nearly the same detection performance.[48] In radars that detect a target's Doppler shift, the envelope detector is replaced by a phase detector that detects the change in phase of the target's echo over the pulses returned from the target during the dwell time (see Section 2.3). [57, p.1.1-1.5][56, p.7-11]

The well known Radar Equation summarizes many of the relationships in radar design. The Radar Equation is given in Equation 2.1.[56, p.88]

$$R_{max}^4 = \frac{P_{av} G A \rho_a \sigma_t n E_i(n) F^4 e^{-2\alpha R_{max}}}{(4\pi)^2 k T_0 F_n (B\tau) f_p (S/N)_1 L_f L_s} \tag{2.1}$$

where

R_{max}	=	Maximum radar range [m]
P_{av}	=	Average transmitter power [W]
G	=	Antenna gain
A	=	Antenna area [m^2]
ρ_a	=	Antenna aperture efficiency
σ_t	=	Radar Cross Section (RCS) of the target
n	=	Number of pulses integrated
$E_i(n)$	=	Integration efficiency
F	=	Propagation factor
α	=	Attenuation coefficient [nepers per unit distance]
k_B	=	Boltzmann's constant = 1.38×10^{-23} J/Kelvin
T_0	=	Standard temperature = 290 K
F_n	=	Receiver noise figure
B	=	Receiver bandwidth [Hz]
τ	=	Pulse duration [s]
f_p	=	Pulse Repetition Frequency (PRF) [Hz]
$(S/N)_1$	=	Required minimum SNR as if detection were based on a single pulse
L_f	=	Fluctuation loss (for a Swerling target model)
L_s	=	System loss

The Radar Equation is valid for a situation where the detection sensitivity is limited by receiver noise. However, clutter is often much larger than the noise in practical radar systems. How clutter affects the detection of targets depends on the clutters characteristics in terms of geographical location, size, reflectivity, radial velocity and internal velocity variation. The most fundamental parameter of clutter characteristics is the clutter cross-section per unit area σ^0 [m2/m2], which is a unitless measure of the clutters reflectivity. The effect of clutter impeding on detection is also dependent on the radars transmitted frequency (clutter strength $\sigma^0 F^4$ from the same patch of land terrain can vary 30 dB over UHF to X-band frequencies). The impact of clutter is often measured as clutter strength $\sigma^0 F^4$ since it often is impractical to separate what is caused by propagation effects (F) and what is caused by the nature of the clutter (σ^0) at the receiver. [56][10][5, p.123-129]

In a scenario where the clutter is much larger than the system noise and the dominating clutter source is surface clutter (land or sea) at low grazing angles, the Radar Equation takes the form of Equation 2.2.

$$R_{max} = \frac{\sigma_t}{\mathrm{SCR}_{min}\sigma^0\theta_B(c\tau/2)\sec(\psi)} \tag{2.2}$$

where

R_{max}	=	Maximum radar range [m]
σ_t	=	Radar Cross Section (RCS) of the target [m^2]
SCR_{min}	=	Minimum discernible Signal to Clutter power Ratio (SCR)
σ^0	=	Clutter RCS per unit area
θ_B	=	Two-way antenna azimuth beamwidth [rad]
c	=	Speed of propagation [m/s]
ψ	=	Grazing angle [rad]

There are several important differences between the situations where the detection

sensitivity is limited by receiver noise and where it is limited by clutter. For instance, in the Radar Equation for surface clutter, the maximum detection range is much more sensitive to variations in target RCS. In this case, the maximum detectable range is also independent of the average transmitted power P_{av} because a net increase in P_{av} increases the clutter as well as the target echoes. Additionally, in a clutter-dominated situation, integration of the radar pulses is generally much less effective than in a system limited by noise. This is because the effectiveness of pulse integration relies on the independence of the noise samples from pulse to pulse.[1] Statistically independent noise tends to not build up as fast in the integration as the target signal, which is often correlated from pulse to pulse. Hence a gain in Signal to Noise power Ratio (SNR) is achieved. However, clutter from stationary features are also persistent and correlated from pulse to pulse, so integration of pulses containing strong clutter provide little benefit. [56, p.404-408]

2.2 Radar receiver and detection

In a superheterodyne receiver as shown in Figure 2.1, the envelope detector gives a signal whose voltage varies with the amplitude of the total signal backscattered to the radar at a given time. This dimension of time (time since the last transmitted pulse) is equivalent to range in the radar's surroundings, with certain reservations. Some propagation effects may distort the time-range relationship, and may cause artifacts or other phenomena in the radar video. This may be caused by effects such as multipath, ducting, refraction and the signal being reflected between multiple scatterers before returning to the radar.

To make a detection decision and detect the presence of a target, the amplitude of the received signal at the output of the envelope detector needs to be compared to a threshold. A detection is thus declared if the amplitude exceeds the threshold. This may result in the detection of a true target when the range-azimuth resolution cell actually contains a target whose presence causes the received amplitude to exceed the cell's threshold level. It may also result in a false alarm, which happens when the signal amplitude in a resolution cell is larger than the threshold for any other reason (noise or clutter). Thus if the threshold is too high, detection of true targets is less probable, and if the threshold is too low, the radar is flooded with false alarms. Due to the probabilistic nature of noise and clutter, the threshold is set to achieve a certain probability of detection (P_d) and probability of false alarm (P_{fa}). The specification of P_d and P_{fa} is of great importance in any radar system and is established by the radar designers requirements. Selection of the proper threshold level is a compromise between missing targets and obtaining false alarms, and depends on how important it is to avoid either of these.[56, p.17]

The three following subsections will cover the detection process in more detail, starting with general detection in 2.2.1 before covering the automatic and locally adaptive techniques of Constant False Alarm Rate (CFAR) in 2.2.2 and Clutter Map (CM) in 2.2.3.

2.2.1 Detection

This section will assume that noise is the limiting factor of detection and that only one pulse is used for detection. Assuming Gaussian system noise, the envelope R of the noise

[1] Receiver noise samples are generally independent in a time $1/B$, where B is the receiver bandwidth.

at the IF amplifier output is Rayleigh distributed.[51] The Rayleigh distribution, and thus the probability density function (pdf) of the noise, is given in Equation 2.3.

$$p(R) = \frac{R}{\Psi_0} exp\Big(- \frac{R^2}{2\Psi_0}\Big) \tag{2.3}$$

where

R = Envelope of received signal
Ψ_0 = Mean noise power
$exp(x)$ = e^x

As shown in Figure 2.2, the envelope of this noise may or may not cross the threshold voltage level (V_T). Figure 2.2 illustrates the relationship between the pdfs of the envelope from a cell with and without a target, the probability of false alarm P_{fa}, probability of detection P_d and probability of miss $1-P_d = P_m$. P_{fa} is the probability that the envelope from a cell that does not contain a target exceeds the threshold, and is given by

$$P_{fa} = P(V_T < R < \infty) = \int_{V_T}^{\infty} \frac{R}{\Psi_0} exp\left(-\frac{R^2}{2\Psi_0}\right) dR \tag{2.4}$$

which when simplified becomes

$$P_{fa} = exp\left(-\frac{V_T^2}{2\Psi_0}\right) \tag{2.5}$$

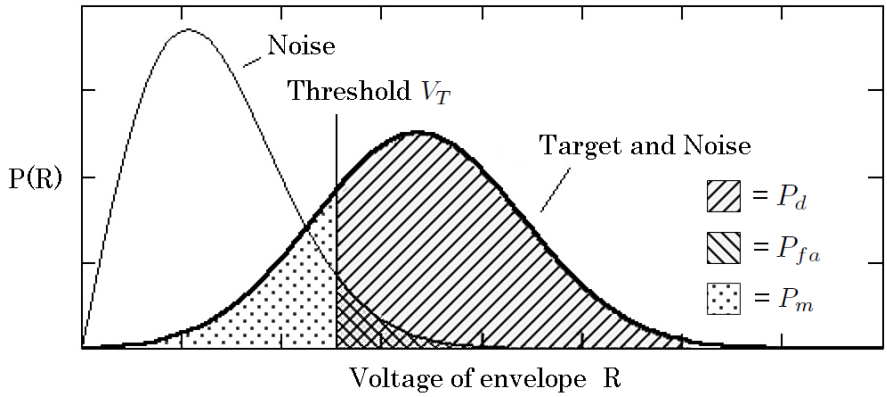


Figure 2.2: Probability density functions of the voltage envelope R at the video output from a resolution cell with and without a target present.

When the radar resolution cell contains a target, the received signal will contain an echo of the pulse reflected by the target as well as noise. Assuming the target echo is a sine wave of amplitude A, the pdf of the received signal envelope R takes the form of a Rician distribution, which is given by

$$p_s(R) = \frac{R}{\Psi_0} exp\left(-\frac{R^2 + A^2}{2\Psi_0}\right) I_0\left(\frac{RA}{\Psi_0}\right) \tag{2.6}$$

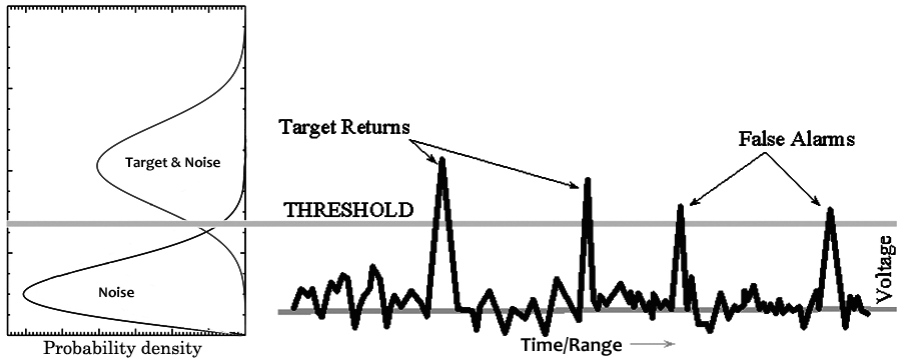


Figure 2.3: The process of detection with a fixed threshold performed on the envelope of a received signal resulting in both detection of true targets and false alarms.

where I_0 is the Bessel function of zero order.[51] The probability of detection P_d is the probability that this envelope exceeds the threshold. Therefore P_d is that of Equation 2.7.

$$P_d = \int_{V_T}^{\infty} p_s(R)dR \tag{2.7}$$

Equation 2.7 has no closed form solution and cannot be evaluated by simple means. Series approximation, numerical and empirical methods have typically been used for calculation of P_d. Equation 2.5 and 2.7 implicate that in a system where detection is limited by noise, V_T, P_d and P_{fa} are inherently connected, and specification of P_d and P_{fa} directly gives the minimum needed SNR for detection. Likewise, specification of P_{fa} and a certain target's SNR gives that target's P_d.[56, p.39-45]

$$P_d = \exp\left[\frac{\ln P_{fa}}{1+\text{SNR}}\right] = P_{fa}^{\frac{1}{1+\text{SNR}}} \tag{2.8}$$

For a Swerling Case 1 fluctuating target[2], the single-sample relationship of P_d, P_{fa} and SNR is given by Equation 2.8.[5, p.80]

Figure 2.3 gives a conceptual description of the detection process. This figure is valid for a short period of time when the antenna can be assumed stationary. On the left, the instantaneous pdfs of the envelope received from a given range-cell with and without a target are shown. These are tilted on the side to match the y-axis (voltage) of the right part of the figure.

The P_d and P_{fa} presented so far have been single-scan probabilities. However, targets often need to be detected several times before the decision is made that a target is present. A common criterion for announcing the presence of a target is to demand M detections of

[2]Swerling Case 1 describes a target with an RCS that varies according to a Chi-squared pdf with two degrees of freedom and is independent from scan to scan. This results in a Rayleigh distributed, scan to scan fluctuating amplitude in the receiver.

the target on N scans, where $1 < M \leq N$. Given a single-scan probability of detection $P = P_d$, the probability of announcing a target present with the M-out-of-N criterion is

$$\text{prob[M out of N]} = \sum_{k=M}^{N} \frac{N!}{k!(N-k)!} P^k (1-P)^{N-k} \tag{2.9}$$

Equation 2.9 is valid for both $P = P_d$ and $P = P_{fa}$. Thus with a 2-out-of-3 criterion, the probability of correctly announcing a target and the probability of a false report are both $3P^2 - 2P^3$ where $P = P_d$ for the former and $P = P_{fa}$ for the latter. For this criterion, the total detection probability becomes larger than that of a single scan if $P_d > 0.5$. With typical values of P_{fa} and a 2-out-of-3 criterion, the probability of a false report becomes much smaller than P_{fa}.[56, p.90]

2.2.2 CFAR

In the previous section, detection was limited by noise and detection decisions were made by comparing the signal to a fixed threshold. When clutter becomes the limiting factor, a fixed threshold will result in excessive false alarms in the areas of high clutter if the threshold is set to give a reasonable overall false alarm rate. At the same time, the threshold will be unnecessarily high in the areas of low clutter, resulting in target misses. Thus, to cope with an inhomogeneous, changing clutter environment throughout the search area, there is need for a dynamic and adaptive threshold. Constant False Alarm Rate (CFAR) is a property of threshold or gain control devices that maintain an approximately constant rate of false target detections when the noise, and/or clutter levels, and/or electronic countermeasures into the detector vary.[5, p.91]

The CFAR makes the threshold adaptive to the local clutter environment. To do this, most CFAR implementations use a sliding window technique that analyzes the neighboring spatial resolution cells of every Cell Under Test (CUT). Only neighboring cells in range are typically used, forming a one-dimensional window. This window is centered on the CUT and includes a total of N reference cells x_i, $i \in [1, 2, ...N]$ that are preceding and succeeding the CUT in range. From these cells a clutter background estimate Z is established for every CUT. The threshold V_T for the CUT is in turn calculated from the given basis of Z by multiplying it with a threshold multiplier K_t, which is constant.

For convenience, the theory in this section assumes a square-law detector ($I^2 + Q^2$, see section 2.3) since it is far more simple analytically than the linear-law detector. The latter has far more complex analytical theory, but is more practical because it uses less dynamic range and yields nearly the same detection performance. This section also makes the usual assumption of Independent and Identically Distributed (IID) clutter and noise samples.[48]

CA-CFAR

One of the most common and simplest implementations of a CFAR is the Cell Averaging CFAR (CA-CFAR). This method uses a background estimate Z_{CA} equal to the arithmetic mean of the reference cells. Due to that a target often is detected in several consecutive range-cells, a number of the cells closest to the CUT are omitted as guard cells in order to

avoid target self-cancellation. Target self-cancellation may occur whenever the presence of a target within a cell raises the detection threshold for that cell. This is also why the CUT itself never is included in the reference window. The threshold V_T for CA-CFAR is given by Equation 2.10. [53][57, p.7.11-7.18]

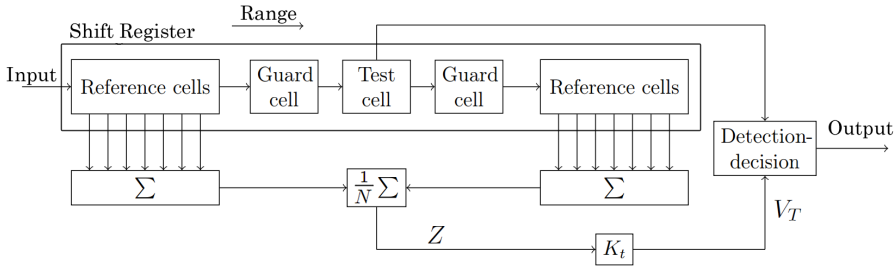


Figure 2.4: Block diagram of the Cell Averaging CFAR (CA-CFAR) implementation.

$$V_{T,CA} = K_t Z_{CA} = \frac{K_t}{N} \sum_{i=1}^{N} x_i \tag{2.10}$$

P_d and P_{fa} for CA-CFAR and a Swerling Case 1 fluctuating target in Rayleigh distributed clutter (far more powerful than the noise) with no interfering targets are given in Equations 2.11 and 2.12. [48, p.664][33, 90-94]

Please note, the author has discovered an error in the expression for P_d in the IEEE T-AES article [48] (Equation 21b in [48]). The correct form is stated in Equation 2.12 in this thesis. This has been verified with the Editor-in-Chief of IEEE T-AES.

$$P_{fa,CA} = \left[1 + \frac{K_t}{N}\right]^{-N} \tag{2.11}$$

$$P_{d,CA} = \left[1 + \frac{K_t}{(1 + \text{SCR})\, N}\right]^{-N} \tag{2.12}$$

Equation 2.12 and the yet to be stated Equation 2.15 give P_d as a function of Signal to Clutter power Ratio (SCR) among others. In these equations, SCR may trivially be replaced by either Signal to Noise power Ratio (SNR) or Signal to Noise and Clutter power Ratio (SNCR) as long as the assumption that both the clutter and noise samples are Independent and Identically Distributed (IID) is valid.

OS-CFAR

Another popular CFAR implementation is the Ordered Statistic CFAR (OS-CFAR). In OS-CFAR, the background estimate Z_{OS} is calculated for every CUT by picking the reference cell with the kth smallest envelope of the N cells in the reference window. The sorted set of reference cells x_i, $i \in [1, 2, ...N]$, where $x_1 \leq x_2 \leq x_3 \ldots \leq x_N$, is called an Ordered

Statistic. The envelope value in cell k is used directly as the background estimate, giving $Z_{OS} = x_k$, where k is the *rank*. The threshold of the CUT for OS-CFAR thus becomes

$$V_{T,OS} = K_t Z_{OS} = K_t x_k \tag{2.13}$$

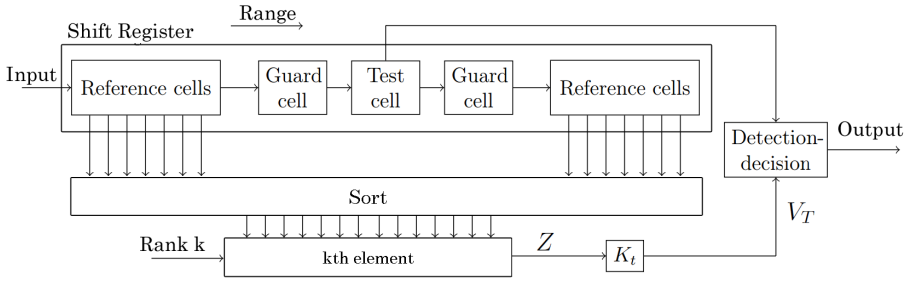


Figure 2.5: Block diagram of the Ordered Statistic CFAR (OS-CFAR) implementation.

If $k = N/2$, the selection of the estimate Z_{OS} is a median filtering process, which gives a low-pass filtering effect of the changes in the clutter background. The probability of detection P_d is a function of the rank k and has its global maximum at $k = 7N/8$. A practical and often used rank is found by Rohling [53] to be $k = 3N/4$. This is chosen as a compromise between larger expansion of clutter areas into the threshold for larger k and a CFAR loss caused by the smaller P_d for ranks further from $k = 7N/8$. Choosing a rank of $k = 3N/4$ gives only a small CFAR loss compared to that of $k = 7N/8$ due to the relatively broad maximum of $P_d(k)$. [53] [12]

P_d and P_{fa} for OS-CFAR and a Swerling 1 fluctuating target in Rayleigh distributed clutter (far more powerful than the noise) with no interfering targets are given in Equation 2.14 and 2.15. Detailed analysis of OS-CFAR can be found in [12], [53] and [54].

$$P_{fa,OS} = \prod_{i=1}^{k} \left(1 + \frac{K_t}{N+1-i}\right)^{-1} \tag{2.14}$$

$$P_{d,OS} = \prod_{i=1}^{k} \left(1 + \frac{K_t}{(1+\mathrm{SCR})\,(N+1-i)}\right)^{-1} \tag{2.15}$$

Generic comparison of CA- and OS-CFAR

CA-CFAR, using the arithmetic mean of the reference cells as the background estimate, has excellent estimation performance in homogeneous clutter and noise situations. If the noise plus clutter is homogeneous within the reference window, CA-CFAR gives an unbiased estimator, meaning $E[Z_{CA}] = E[x_i]$. The estimator also shows a minimum estimation variance $\mathrm{Var}[Z_{CA}] = \mathrm{Var}[x_i]$. The estimator performance increases for larger window sizes N as long as the assumption of homogeneous clutter and noise within the window is valid. Figure 2.6 shows the performance of CA-CFAR in different environments, where a detection is made whenever a cell's amplitude (blue) exceeds the cell's threshold (red).

CA-CFAR performs somewhat poorly in inhomogeneous backgrounds or with interfering targets. Figure 2.6c and 2.6d shows the masking effect a target has in CA-CFAR, raising the threshold in its neighboring cells. As in 2.6d, this may result in target misses.

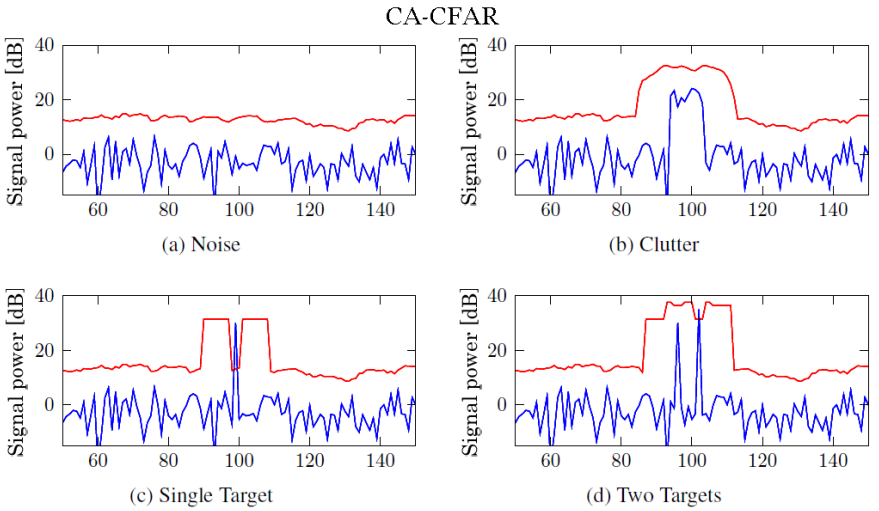


Figure 2.6: Threshold (red) and cell amplitude (blue) in CA-CFAR using $N = 16$ and $P_{fa} = 10^{-6}$.[52]

OS-CFAR is designed for situations where the background is inhomogeneous or when there are several targets within the reference window. It has no need for guard cells and can even include the CUT itself in the reference window without causing target self-cancellation. Targets also do not mask each other. As can be seen in Figure 2.7, the threshold is unaffected by targets, which is very different from CA-CFAR. OS-CFAR is robust and well performing in both inhomogeneous clutter and multiple interfering target situations. The major disadvantage of OS-CFAR is the high processing power required for performing the sorting algorithm, which is much more computationally expensive than the operations of the CA-CFAR. [52][53][12][5, p.88][57, p.7.11-7.18]

2.2.3 Clutter Map

The CFAR methods described in Section 2.2.2 use information from neighboring cells in the range dimension to establish the threshold for the CUT. Another approach to implement a CFAR is by using the temporal (scan to scan) information in the same cell. Such a temporal CFAR, called a Clutter Map (CM) CFAR, stores the received signal power in a cell from each scan and can use this information to determine the cell threshold in a number of ways.

In a pulsed Doppler radar, tangentially moving targets can easily be masked by land clutter since they have the same radial velocity as the clutter and often have echoes far weaker than the clutter. However, a CM-CFAR may detect such a target by comparing

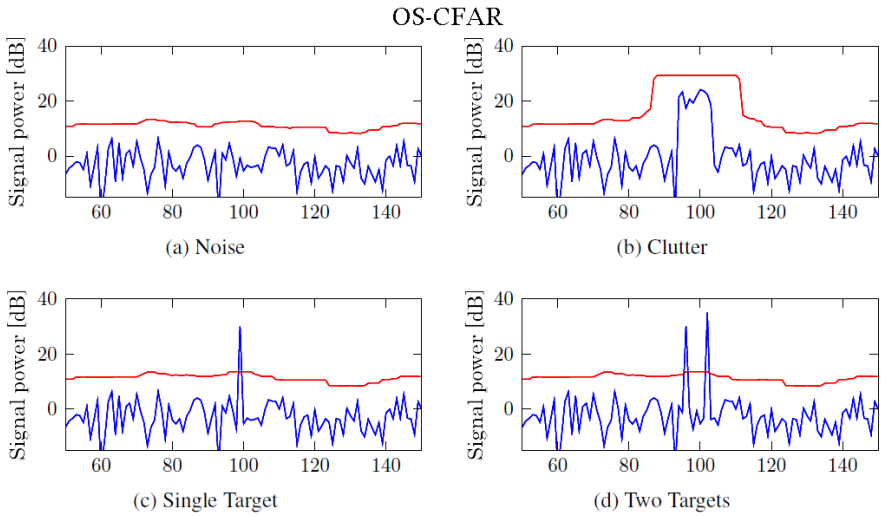


Figure 2.7: Threshold (red) and cell amplitude (blue) in OS-CFAR using $N = 32$, $k = 24$ and $P_{fa} = 10^{-6}$.[52]

the total power received from a cell from scan to scan. This is because a target moving through a cell will contribute to a temporarily increased total received power from the cell.

The use of a CM may yield very good performance in an inhomogeneous clutter environment. In cells where clutter consistently causes false alarms in a sliding window CFAR technique, the CM-CFAR will take this temporal correlation into account. By storing the radar video from previous scans in the CM, the cells with consistently high received power may be located and their thresholds increased according to each cell's history. Consistent false alarms from these cells will hence be suppressed.

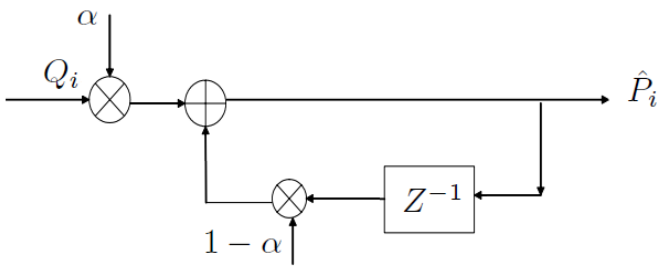


Figure 2.8: Recursive implementation of the CM-CFAR.

A common implementation of a CM is to store one value for every cell, and to update the CM in a successive manner for every new scan. Figure 2.8 shows this implementation, where the CM input Q_i is the image from the current scan i, and $\alpha \in \langle 0, 1]$ is the updating coefficient that determines how quickly the stored cells in the CM will adapt to the current

scan. The output image $\hat{P}_i$ is stored as the updated CM. Equation 2.16, where the cells n of the images are also included, gives the mathematical description of the filter in Figure 2.8.

$$\hat{P}_i(n) = \alpha Q_i(n) + [1 - \alpha]Q_{i-1}(n) \tag{2.16}$$

The CM image $\hat{P}_i(n)$ resulting from Equation 2.16 consists of the current scan i weighted α, and every previous scan $i - T$, $T \in [1, 2, 3...]$ weighted $(1 - \alpha)^T$. The background estimate Z_{CM} for detection in a CUT in the current scan is set by the CUT's corresponding value in the previous scan's updated CM $\hat{P}_{i-1}$. The previous scan's CM $\hat{P}_{i-1}$ is used (instead of $\hat{P}_i$) to avoid target self-cancellation. The threshold $V_{T,CM}$ for a CUT with CM-index n is given in Equation 2.17.[16]

$$V_{T,CM} = K_t Z_{CM} = K_t \hat{P}_{i-1}(n) \tag{2.17}$$

The CM of Figure 2.8 will take some time to adapt to changes in the clutter situation, such as moving clutter. Since the map is built successively, the map will converge on the present situation if this situation persists. The settling time is a measure of the CMs responsiveness to changing clutter and is controlled by α. During the settling time this typically results in excessive false alarms when the CM underestimates the clutter and suppression of detections when the CM overestimates the clutter.

By combining the spatial CA- or OS-CFAR with a temporal CM-CFAR, one may achieve better performance than with either alone. In this thesis, this type of spatial-temporal hybrid CFAR that calculates the thresholds V_t from the spatial and temporal branch separately before merging is called CACM-CFAR or OSCM-CFAR. This type of merging must not be confused with the insertion of spatial techniques (CA and OS) into the temporal CM-CFAR by updating the CM with the output made by a CA- or OS-technique. The spatial-temporal hybrid type of CFAR merging gains the advantages of both spatial and temporal CFAR simultaneously and results in three clear advantages. The sub-clutter visibility of the CM-CFAR remains, target self-cancellation problems are mitigated and periodic response is eliminated due to the sliding window of the spatial branch.[16]

2.3 Doppler Processing

2.3.1 MTI

A Moving Target Indicator (MTI) is a low-PRF Pulsed Doppler Radar that applies a band-stop filter to the Doppler frequency band near zero velocity. This band stop filter is called an MTI filter. The intention of this is to suppresses clutter and stationary targets, emphasizing moving targets. Tangentially moving targets will however have zero (or close to zero) Doppler shift, and will thus also be suppressed by the MTI filter. Clutter with spectral energy sufficient for detection outside the stop-band in the MTI filter will cause false alarms. This may be land clutter if the MTI filters stop-band is not appropriately designed to fit the land clutters spectrum. Rain will often not be blocked by the MTI-filter since it often has a velocity component radial to the radar.

The MTI achieves a significant increase in performance relative non-coherent or non-Doppler radar systems. It does this because when the clutter is filtered out prior to the detection process, thresholding and detection can be performed as if the target were in a clutter-free environment. This is a huge advantage. However, clutter sources (especially rain and land) can have spectra with much energy outside of the MTI filters stop band, in which case the clutter greatly degrades the MTI performance.

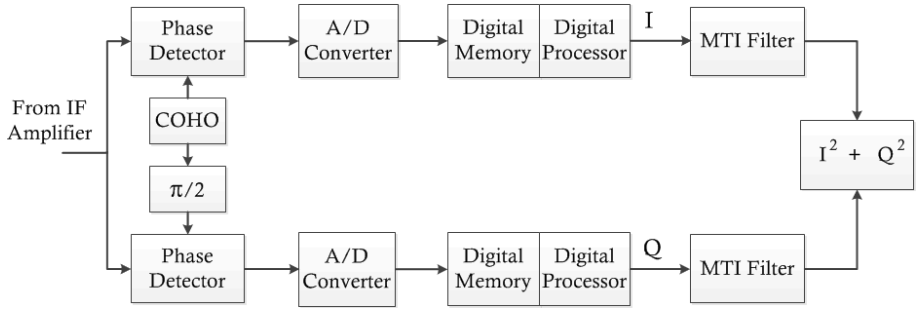


Figure 2.9: Block diagram of a digital MTI Doppler quadrature channel signal processor with a square-law detector.[56, p.138][42, p.393]

Figure 2.9 shows a block diagram of a digital MTI Doppler signal processor that employs a quadrature channel technique. The signal from the IF amplifier (shown in Figure 2.1) is split into an in-phase channel I and a quadrature channel Q. Phase detectors outputs the Doppler signal as $A_d \sin(2\pi f_d t + \phi_0)$ in the I channel, and $A_d \cos(2\pi f_d t + \phi_0)$ in the Q channel. To perform this extraction of the Doppler frequency, the phase detectors need to be input the signal from a Coherent Oscillator (COHO), which is phase-coherent to the transmitted pulse (see section 2.3.3.

The quadrature signals are then sampled and filtered before detection. The quadrature channel processing gives several advantages in terms of application and practicality, but most importantly this technique obviates problems due to blind phases. Blind phases is the phenomena that occurs when the phase between the Doppler signal and the sampling at the PRF results in a loss and may seriously degrade the MTI performance.[56, p.137][42, p.393]

Due to the pulsed nature of Pulsed Doppler Radars, the system is prone to aliasing in the frequency/velocity domain as in a digital filter with a sampling frequency equal to the PRF f_p. This results in what is called blind speeds. The response of an MTI filter is zero at zero frequency, and thus also zero at frequencies nf_p, $n \in \mathbb{Z}$. In terms of radial velocity, this translates to the blind speeds v_n that are given in Equation 2.18.[57, ch.2]

$$v_n = \frac{n\lambda f_p}{2}, n \in \mathbb{Z} \tag{2.18}$$

2.3.2 MTD

The Moving Target Detector (MTD) further exploits the benefits and possibilities of digital Doppler processing by filtering the quadrature Doppler signal into different velocity channels. This way, thresholding and detection may be performed separately in each velocity channel. That is, detection of targets in a velocity channel that does not contain clutter (from land, rain, sea, etc.) is performed as if the targets were in a clutter-free environment even though the targets and clutter may be the same location. However, targets that have the same radial velocity as for instance the rain on a windy day will fall in the same velocity channel as the rain, and can therefore not be separated from the rain clutter. A block diagram of an MTD signal processor is given in Figure 2.10. The MTD processor employs an N-filter filter bank and a Clutter Map (CM).

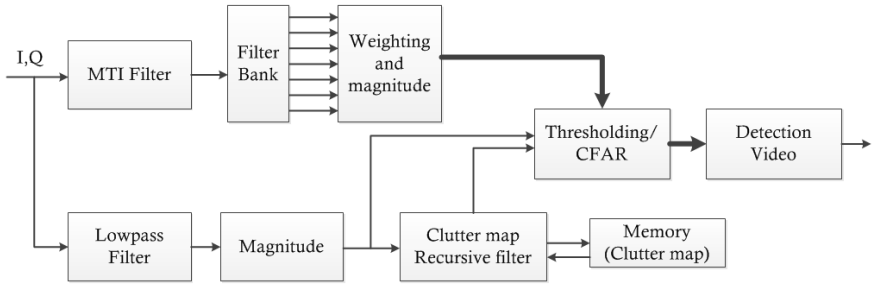


Figure 2.10: Block diagram of a Moving Target Detector (MTD) signal processor.

As practical filter banks have finite stop-band attenuation, the channels leak into each other so that target detection is performed in a clutter-attenuated rather than a clutter-free environment. One important quality measure for an MTD system is the Normalized Clutter Attenuation (CA), which is defined by the ratio of the input $(\text{C/N})_i$ to the output $(\text{C/N})_o$. [5, p.244]

$$\text{CA} = \frac{(\text{C/N})_i}{(\text{C/N})_o} \tag{2.19}$$

Another important quality measure for an MTD system is the Sub-Clutter Visibility (SCV), which measures a radar system's ability to detect moving-target signals superimposed on clutter signals. A SCV of 20 dB means that the radar can detect a target in a spatial resolution cell whose clutter signal is 100 times more powerful than that of the target signal. [57, p.2.22]

One problem with only stating CA is that it does not take into account the attenuation of the wanted signals. Therefore, the perhaps most usual performance measure of the MTI/MTD process is the *MTI improvement factor* I_f which is defined as the ratio of SCR at the output to the SCR at the input, averaged over all Doppler frequencies. However, if the targets are distributed uniformly over all Doppler frequencies, then $I_f = \text{CA}$ and

$$\text{SCV} = \frac{\text{CA}}{(\text{S/C})_o} \tag{2.20}$$

where $(\text{S/C})_o$ is the output SCR required for detection.[5, p.244]

2.3.3 Pseudocoherent radar

Pseudocoherent radar, or coherent-on-receive radar, is a method to obtain coherency and enable the possibility of Doppler processing in an initially non-coherent system.

As can be seen in Figure 2.9, the phase detectors for extracting the Doppler frequency, and thus establish a target's Doppler shift, require the signal from a Coherent Oscillator (COHO) as input. This is a crucial part of the MTI/MTD system. That the oscillator is coherent means that it has a stable phase-relationship from pulse to pulse because it is not pulsed, but oscillates continuously at a stable frequency. The signal from the COHO is modulated by a pulse modulator and amplified to generate the transmitted pulses. MTI/MTD radars commonly use the rather expensive high-power signal amplifiers Traveling Wave Tubes (TWTs) and klystrons to amplify the signal from the low-power COHO. When n pulses are backscattered from a moving target during the dwell time, the phase of the received signal from each of these pulses will vary. This is due to a change in distance from pulse to pulse since the target is moving. This change in phase is detected by the phase detectors, and is used to extract the Doppler frequency.

Radar systems like the civil marine radar used in the work of this thesis do not have a COHO from which the transmitted pulses are generated. These systems use a magnetron, which is not an amplifier, but rather a high-power oscillator. A magnetron is not suited to act as a COHO, but generates a high power carrier waveform when it is excited by a high voltage pulse. The starting phase of this pulse is random, and thus not coherent from pulse to pulse.

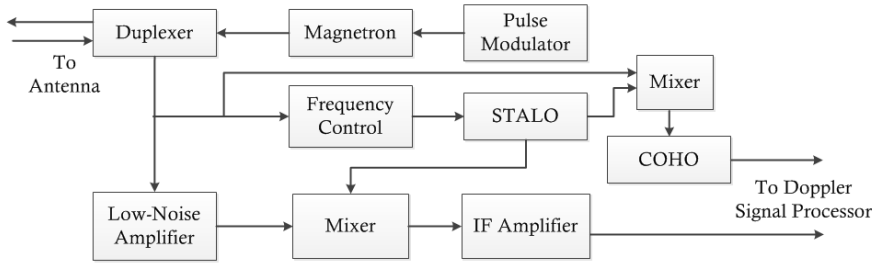


Figure 2.11: Block diagram of a pseudocoherent radar system.

To apply MTI processing in radar systems with magnetron oscillators, one can modify the system and build what is known as a pseudocoherent radar. This is done by estimating a COHO for each pulse based on the transmitted signal. Figure 2.11 shows a block diagram of of a pseudocoherent radar. Leakage of the transmitted pulse through the duplexer and into the receiver is used to lock a stable low-power oscillator to the phase of the transmitted pulse. In a superheterodyne receiver, because the extraction of the Doppler frequency is performed on the IF signal, the leaked part of the transmitted pulse is used to set a Stable Local Oscillator (STALO) which controls the frequency mixing to and from IF. The leakage of the transmitted pulse through the duplexer is hence mixed to IF before it is used to estimate and lock the phase of the COHO. A COHO estimated in this manner may therefore be used to extract a target's Doppler as long as every received target echo originates from the same pulse from which the COHO was estimated.

The phase locking of the COHO in a pseudocoherent radar system is not as accurate as in a fully coherent system, resulting in a lower CA and SCV. Also, because the estimated COHO is only valid for echoes originating from the last transmitted pulse, echoes from previously transmitted pulses that are received after a new pulse has been transmitted will not be canceled (or extracted) by any Doppler domain filtering. Therefore, both strong clutter and moving targets (with large RCS) outside the radar's unambiguous range will degrade the system performance.[56, p.189][70]

2.4 Birds as radar targets

This section will describe the characteristics of birds as targets for detection by radar. There are many challenges associated with detecting birds due to their small size, variable speed, low altitude, high agility and changing shape as they flex their muscles and flap their wings. A bird in flight is a highly complex target and produces a highly fluctuating received radar signal. The Radar Cross Section (RCS) of birds should therefore be modeled and expressed by its statistical properties. Existing data on birds as radar targets mainly consist of ornithological studies, many of which are summarized by Vaughn in [63], but current models of a bird's RCS do not adequately represent real targets.

With respect to detection by avian radar, bird targets are traditionally modeled as a sphere of water with a mass equal to the mass of the bird, henceforth called Sphere of Water with Equal Mass (SWEM). However, studies have shown that a Prolate Spheroid (PS) of the same volume with a length-to-width ratio between 3 and 10 is a better approximation. But the SWEM method will work fine for simple calculations that do not account for target aspect angle. The small size and surprisingly light mass of birds cause them to have a small mean RCS $\overline{\sigma}$. Much of their apparent size is due to their bulky plumage, which contributes little to both weight and RCS. For most single birds, $\overline{\sigma}$ is between 1 to 100 cm^2, though very large birds (geese or eagles) may have a mean RCS around $\overline{\sigma} = 200cm^2$. Migrant flocks, such as a large flock of geese may have a mean RCS exceeding 10^4 cm^2. [63, p.223]

Normal changes in a bird's aspect angle and shape may cause rapid and large fluctuation of the instantaneous RCS $\sigma(t)$. If multipath and propagation effects is not accounted for separately (propagation factor F), which often is the case, these may also be measured as fluctuations in RCS (fading). If several birds are located within the same range-azimuth resolution cell, their individual echoes will also interfere with each other, resulting in further fluctuation of the flock target's RCS. By the SWEM model, most bird RCS at X-band fall within the Mie resonance region ($\lambda/2\pi < r < 5\lambda/\pi$). This results in an additional ± 5.5 dB variation in RCS depending on the bird's exact mass and shape. Figure 2.12 shows an example of RCS fluctuations for a pigeon. [20] In the Radar Equation, such as the variants given in Equation 2.1 and 2.2, RCS is described by a parameter called *Target RCS* σ_t. In these equations, the time dependence of $\sigma(t)$ is omitted for simplicity and $\sigma_t = \overline{\sigma}$, constituting an average-case estimate of the maximum detection range.

At X-band (9.41 GHz), assuming a SWEM RCS model, birds with masses between 0.5g and 0.5kg fall within the Mie region. Heavier birds' RCS will reside in the Optical

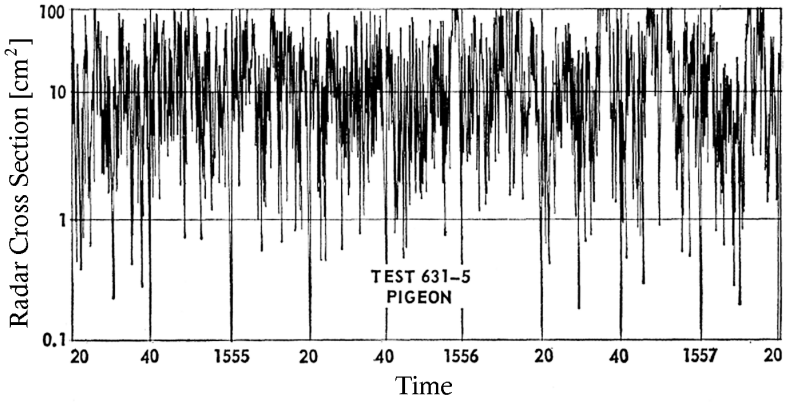


Figure 2.12: A typical three minute time history of Radar Cross Section (RCS) $\sigma(t)$ for a single pigeon at S-band, found by Konrad et. al. in [20].

region, which removes the Mie resonance fluctuation with RCS approaching the sphere's physical area.

For a SWEM model at X-band, the mean RCS of single birds with mass m may be approximated by the following equation.[14, p.272]

$$\text{Single bird mean RCS} = \overline{\sigma}_1 \approx 0.01 m^{2/3} \quad [m^2] \tag{2.21}$$

Equivalently, the mean RCS of a flock of N birds, each with individual mass m, may be approximated by

$$\text{Flock mean RCS} = \overline{\sigma}_N \approx 0.01 N m^{2/3} \quad [m^2] \tag{2.22}$$

The weight of different bird species can be found in Table 2.1. Here, both the SWEM model (Equations 2.21 and 2.22) and the prolate spheroid model used by Vaughn [63] is used to calculate the single bird mean RCS for species of different weight. The Prolate Spheroid (PS) model uses a length-to-width ratio of 5:1 viewed from the side, constituting a best-case scenario, and viewed from the front, constituting a worst-case scenario.

Table 2.1: Bird mean RCS $\overline{\sigma}$ by species/mass at X-band, approximated by the Sphere of Water with Equal Mass (SWEM) model and the 5:1 Prolate Spheroid (PS) model used by Vaughn in [63].

Species / $\overline{\sigma}$ model	Sparrow	Seagull	Goose	Swan
Mass	30 g	900 g	3.3 kg	10 kg
SWEM	19 cm^2	93 cm^2	221 cm^2	464 cm^2
5:1 PS (side-view)	20 cm^2	100 cm^2	500 cm^2	1000 cm^2
5:1 PS (end-view)	0.03 cm^2	0.3 cm^2	0.9 cm^2	1.6 cm^2

As may be seen in Table 2.1, when using the 5:1 PS model and viewing the Prolate Spheroid from the end, the mean RCS is incredibly small - even for large birds. For side-view, the mean RCS is somewhat above that of a sphere (SWEM model), as can be expected.

The instantaneous bird RCS $\sigma(t)$, and the received power therefrom, is reported to follow a Log-normal distribution. The mean of the pdf of the RCS is equal to the temporal mean RCS $\overline{\sigma}$. Overall fluctuations for bird RCS are very large, ranging several magnitudes, which makes it hard to estimate accurately with current models. The characterizing parameter for the amount of fluctuation in the received signal from a bird target is the mean-to-median radio of the RCS. This ratio is in turn dependent on the exact size and shape of the bird relative to the radar wavelength. [20]

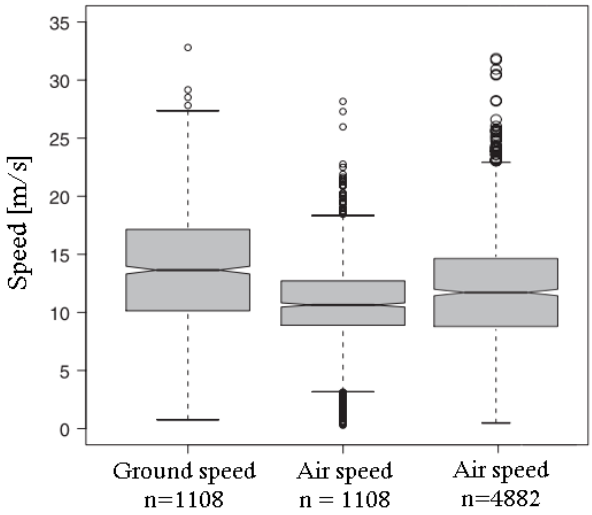


Figure 2.13: Box plots of Ground and Air speed found in two different samples of empirical data consisting of $n = 1108$ (left and middle) and $n = 4882$ (right) birds, by Schmaljohann et.al. in [19].

Birds are very agile and maneuverable at varying speeds depending on species. The pattern of movement is highly relevant with respect to radar tracking of bird targets. Some of the ornithological studies performed by radar, have resulted in data on the ground speed (speed relative the ground) and air speed (speed relative to the air) of birds. Figure 2.13 shows an excerpt of empirical data on the speeds of two samples of 1108 and 4882 birds, presented in [19]. The top and bottom of the boxes are the first and third quartile, thus containing 50 % of all the measurements, while the line inside the box is the second quartile (median). From this figure, one can see that air speed is somewhat smaller than ground speed in most cases, presumably because the birds may achieve a higher ground speed when flying downwind. By comparing the ground and air speed of the first sample (the box-plot to the left and in the middle of Figure 2.13), and observing that the air speeds of the largest sample (the box-plot to the right), it seems probable that a realistic mean

ground speed for the largest sample may be around 15 m/s.

2.5 Multiple-target tracking

This section will give an outline of some of the principles of radar Multi-Target Tracking (MTT) that are most relevant to this thesis.

The use of an MTT system in radar application gives several great advantages, of which the most obvious and perhaps most important in some applications is the automation of target reporting. Without Automatic Detection and Tracking (ADT), a radar operator is required to constantly interpret the radar video. ADT systems do not obviate the purpose of the radar operator. Interpretation by a well trained radar operator is favorable also with ADT systems. But in an ADT system, a human operator is not required for recording of or reacting to the system video output. Even with an operator interpreting the video, in a multiple-target tracking scenario with excessive false alarms, an ADT system might help bring attention to possible targets that otherwise would have been ignored as false alarms. This means one might allow more false alarms into the tracker than what would have been acceptable without it. In other words, an ADT system might allow a lower threshold, resulting in a higher P_d.

An ADT system will also be able to automatically locate and count the detected target activity and for instance store this data in logs. This would also makes it easy to generate summarizing and scheduled activity reports or to manually review previous activity. With the automatic tracker, the output of the tracker might also be input for a surveillance or early-warning type of function that examines the results from the tracker and reacts if certain requirements are met, which is what is done in the system of this thesis.

The basic elements of a simple recursive MTT system is shown in Figure 2.14. The figure represents the cyclic structure of the processing in recursive MTT systems. First, the measurements are processed and prepared to fit the tracker implementation. As a single target often results in multiple detections, the detections are usually extracted and represented as single measurement called a plot. After extraction, a plot typically contains information such as position, Time of Observation (TO), size/shape, Doppler shift and target type. In this thesis, the words plot, measurement and observation are synonymous.

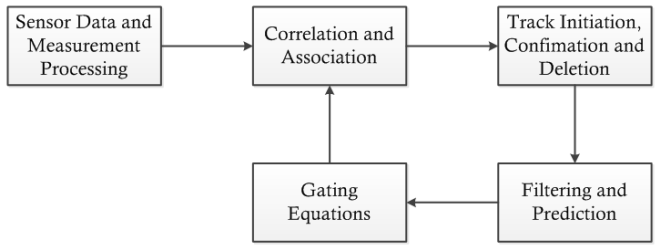


Figure 2.14: Basic elements of a simple recursive MTT system.[11, p.5]

The MTT processing cycle starts with the finding the correlation of the currently received data to the previously established tracks. Plots are then associated to existing tracks

according to the rules established by the system design.

In the next step, plots that have not been associated to any existing tracks are used to initiate new tracks. The quality of every track also needs to be evaluated. After evaluation, all tracks that do not meet the established quality requirements are deleted. The tracks that survive this test are kept as confirmed tracks. This is often done by utilizing a Score Function (SF) that produces a track score based on available information about a track before deleting all tracks with a score that is lower than a certain limit. Other demands might also be given for the confirmed tracks, such as a maximum time without the association of new measurements. The process of identifying the tracks that meet the quality requirements is called Track Confirmation. [11, p.151-173]

Filtering and prediction are then performed. This results in tracks with updated and smoothed target states, covariances and measurements of uncertainty as well as an updated predicted position of a track's target at the time of the next data-set. The predicted position will be calculated by the previous observations of the target and the tracker system's target dynamics model. [37] Based on this predicted position, Gating Equations establish the acceptance gate around the target's predicted position when returning to the cycle's first step.

There are many methods of implementing an MTT system. An iterative single-scan filtering implementation like the fixed-coefficient α-β and the α-β-γ Tracker [34] or the dynamic-coefficient Kalman Filter Tracker [49] have typically been the most common approach. [11, p.19-44] [57, p.7.22-7.34] [15]

In modern MTT systems, more precise non-linear filtering methods such as the Extended Kalman Filter (EKF) are often employed. However, batch/multi-scan methods are the focus of this thesis. Batch methods may be somewhat easier implemented than recursive approaches as the popular EKF. A batch method known as Batch-Least-Squares has also been found to converge both faster and more accurately than EKF trackers, but specific performance will of course vary with both different implementations and input data.[61][11, p.159-168]

With batch tracking methods, the design will vary depending on how many scans of the search area that are included in one batch. One must have measurements over a time sufficient to observe the targets several times to establish a correlation in the measurements, but many scans must thus be recorded before processing can begin, resulting in a delay. The delay in interpreting and hence reacting to the results output from the tracker might render an MTT system useless for several applications. Therefore, the number of scans in one batch N must be balanced to achieve an acceptable delay while still getting enough observations of every target to establish their existence.

2.5.1 Gating

Gating is the first part of the correlation algorithm used to decide which observations are candidates for track updates. For every previously established track, there is determined a predicted position R where the target would be at the time of the next measurement if the target behavior follows the tracker's target dynamics model. A gate of acceptance is established around every such point R to limit the number of plots that are candidates for updating the track in question. This gate is specified in every spatial dimension measured (range, azimuth and elevation) and the exact distance from R to a decision line will be

dependent on the system requirements and model for target dynamics as well as the type of gate that is employed.

Gates might be rectangular, spherical, elliptical, an annular sector, a torus sector or defined by a combination of several rules and restrictions. For example, a system might establish one gate for regular identification of candidates, and one for identification of candidates and the detection of target maneuvering which is activated when no plots are observed within the first gate. For some types of gate shapes there are expressions for an optimal gate G_0 described in the literature. The gates from several tracks might also overlap, so that a plot might be contained by the gates of several tracks. [11, p.83-92]

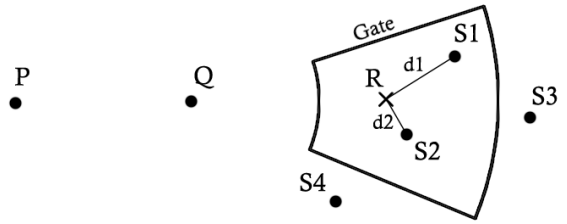


Figure 2.15: The principle of Gating around the predicted position R in the range and azimuth dimensions. Here shown with an annular sector gate and Nearest Neighbor (NN) assignment logic.

Figure 2.15 shows the principle of gating, illustrated by an annular sector gate around the target's predicted position R. P and Q are the two last observations of the target, while $S1$ and $S2$ are plots from the current scan. All the plots whose positions fall within the gate for a certain track, will immediately be accepted as candidates for updating the track. Determining which of the candidate plots to update the track with is called the data association problem, or track association.

2.5.2 Data Association

The most common methods of logic for track association are called Nearest Neighbor (NN), Probabilistic Data Association (PDA), Joint PDA (JPDA) and Multiple Hypotheses (MH) algorithms.

The NN method is a hard-decision logic that associates to a track the single plot that is nearest (and least uncertain) to the predicted position of the target represented by the track.

PDA and JPDA are soft-decision logic methods that associates to a track all the plots that fall within the track's gate. Each plot is weighted with the perceived probability of being the correct association and the system relies on erroneous associations averaging out. [4]

MH, which also is a soft-decision logic, hypothesizes that every plot within the gate is the correct association and establishes a new track for every hypothesis. This results in a quickly expanding hypothesis-tree where the further association of newer data eventually will lead to deletion of all the hypothesized tracks that do not reliably observe a target, leaving the tracks that do.[11, p.5][57, 7.39]

Using the gating technique and measurement situation shown in Figure 2.15, and assuming the NN track association logic, one can construct a matrix called The Assignment Matrix to obtain a solution to the association problem by the classical approach. First, a distance function defines the normalized distance d from the target's predicted position to every plot S in the current scan. The normalized distance d is a function of the position of R and plot S, and might include other parameters such as the Doppler of S and measures of uncertainty. In the situation of Figure 2.15, an NN logic will associate the plot S2 to the track in question because d2 $<$ d1. Assuming the track containing plots P and Q is the only track, the assignment matrix becomes as given in Table 2.2.[11, p.94]

Table 2.2: The Assignment Matrix in the case of Figure 2.15 with Nearest Neighbor (NN) assignment logic. Plot S2 is associated to Track 1 because d2 $<$ d1. An x denotes that the plot is outside the gate.[11, p.94]

	S1	S2	S3	S4
Track 1	d1	d2	x	x

When other tracks are present, and the number of plots S in the current scan grows larger, the complexity of the data association problem increases. One might allow a plot S to be assigned to several tracks, but the classic approach is to make a plot that is assigned to one track unavailable for assignment to other tracks. In the former case, assignment is done independently for every track. In the latter case, after establishing the assignment matrix, one has to find the overall best solution. This is the assignment solution that minimizes the total of the normalized distances d_{ij} of all the plots S_i assigned to tracks with predicted target positions R_j.

2.6 Markov Chains

A Markov Chain is a practical tool for statistical analysis of the properties of a complex system with many states. In this thesis it is used in a simple analysis of the tracker system's statistics of track confirmation and retention given an M-of-N criterion.

A Markov Chain is described by a series of system states $S = \{s_1, s_2, \ldots, s_r\}$ and a transition matrix Φ which contains the probabilities of system state transition. The process starts in one of the system states, and moves from one state to another in steps. In Φ, element $p_{i,j}$ describes the probability of the system transitioning to state s_j given that the system is currently in state s_i. It is also given that the system follows the *Markov property*, which states that the probability of state transition depends solely on the current system state and not on the state history. In a Markov Chain describing a track confirmation process with an M-of-N criterion, the transition matrix Φ depends on the probability of detection P_d. [67]

The process of track confirmation through the system steps k may be represented by the Markov Chain relationship given in Equation 2.23. [25, p.405-413][11, p.193-203]

$$P_{k+1} = \Phi_{k,k+1} P_k \ , \quad k = \{0, 1, 2, \ldots\} \tag{2.23}$$

Here, $P = [p_1, p_2, \ldots, p_r]$ is a vector containing the probabilities of the system being in each system state at step k. P_0 thus gives the initial conditions describing the probability of the system starting in each of the system states. Then after k steps, the cumulative probability of the system being in every possible system state is given by Equation 2.24

$$P_k = \Phi^k P_0 \tag{2.24}$$

Consider a 2-of-3 criterion, demanding 2 detections of a target within 3 consecutive scans for track confirmation. The state diagram of this process is given in Figure 2.16.

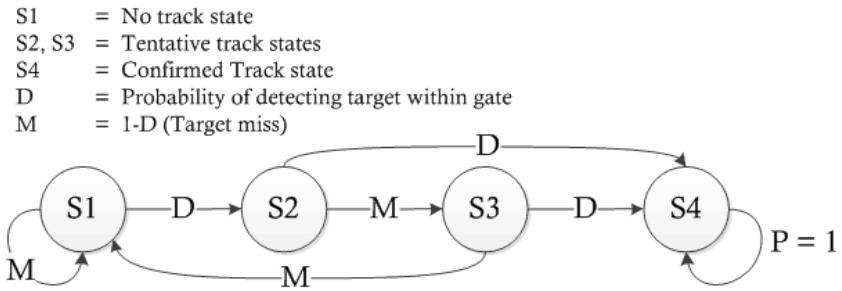


Figure 2.16: State diagram of a 2-of-3 track confirmation criterion.[11, p.194]

Here, state "S4" is the state at which there is established a confirmed track. This state is a *trapping state* since the only state transition is back to itself with probability $P = 1$. This in turn, is because when a confirmed track is established for instance from detections in two consecutive scans, the 2-of-3 criterion is met, regardless the outcome of scan three. For such a process, the Markov Chain relationship of Equation 2.24 becomes

$$\begin{bmatrix} P_k(1) \\ P_k(2) \\ P_k(3) \\ P_k(4) \end{bmatrix} = \begin{bmatrix} M & 0 & M & 0 \\ D & 0 & 0 & 0 \\ 0 & M & 0 & 0 \\ 0 & D & D & 1 \end{bmatrix}^k \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.25}$$

For targets whose range changes little during the total number of scans k, the probability of detection P_d may be assumed to be constant. D, the probability of detecting the target within the set gate is a function of the gating technique and of P_d.

For simplicity, a sufficiently large gate, high SNR and no interfering targets is assumed. Then, $D = 0.7$ gives a probability of confirming a track after $k = 3$ scans $P_3(4) = 0.784$.

Chapter 3

Framework and Methodology

This chapter will describe the total radar and tracker system framework, the methods used in development of the tracker and methods for analyzing and evaluating the performance of the tracker. Detailed description of the tracker system implementation follows in Chapter 4.

Measurements of radar data were performed at Værnes Airport, Norway, during the summer of 2014 to serve as input for the development of the tracker. Additional measurements of geese-flocks were included as the migratory geese arrived during the fall of 2014. The radar system was installed at Værnes as part of Avinor's follow-up after a birdstrike incident at the airport in January, when a Boeing 737 hit a swan during take-off.[1] Radian AS designed and delivered the system based on a Furuno civil marine radar and is responsible for system maintenance and further development. Since the use of a magnetron-based marine radar as an MTD birdstrike avoidance radar is a pioneer project, development is ongoing and the work of this thesis contributes to the development of a tracker.

The established radar site where the measurements were performed is currently the most relevant location for the birdstrike avoidance radar at the airport with the current single-sensor setup. These measurements were recorded directly from the radar and consist of coherent and unprocessed radar signals, which were then used as input data in the testing and development of the tracker. The development environment was set on a PC, where the tracker was developed in the MathWorks® MATLAB® programming language.

3.1 System

This section will describe the total radar system that was used to perform the measurements at Værnes Airport. This system constitutes the framework that the developed tracker is tested on and built into. An overview of the system is given first, followed by the system specifications and special considerations and priorities concerning the overall system.

3.1.1 System Overview

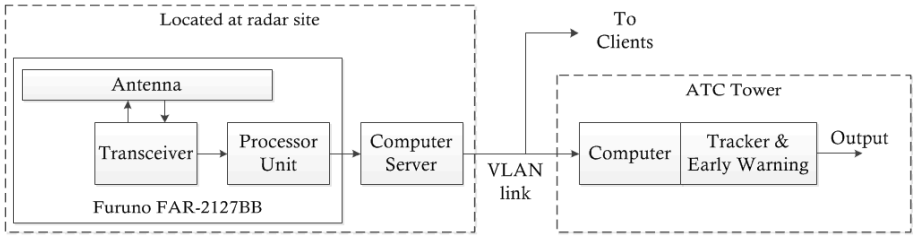


Figure 3.1: Hardware framework and overview of the radar system at Værnes Airport.

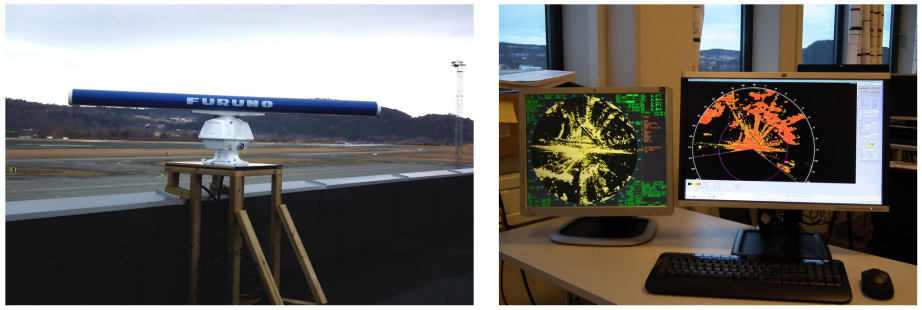


Figure 3.2: The Furuno FAR-2127BB Antenna and Transceiver unit (left), and the Interface of the Processor Unit and Computer Server (right) deployed at the radar site at Værnes Airport.

Figure 3.1 gives an overview of the radar system and the hardware framework of the developed tracker. The total radar system consists of a radar segment located at the radar site at Værnes Airport, and one or several mobile computer client segments. One of these client segments is placed in the Air Traffic Control (ATC) Tower at Værnes Airport for usage in mitigation of the risk of birdstrikes. See Appendix A for radar site and ATC Tower location.

The radar segment shown to the left in Figure 3.1 is stationary at Værnes Airport and includes a Furuno FAR-2127 BlackBox (BB) marine radar and a computer server. The Furuno FAR-2127BB itself can be seen to the left in Figure 3.2. The computer server is located within the building and is fed the raw and unprocessed signal received by the radar. The interface to both the FAR-2127BB and the Computer Server is shown to the right in Figure 3.2. By the principle of operation described by section 2.3.3, this computer server transforms the received pulses to become coherent. This is not a part of an ordinary marine radar system, and is what enables the Doppler processing in the further steps. The server also runs an MTD processor that performs automatic detection of targets and stores the results locally, making them available for access and distribution on a Virtual Local Area Network (VLAN) on the system intranet.

Each client computer with the required access may then view real-time MTD radar

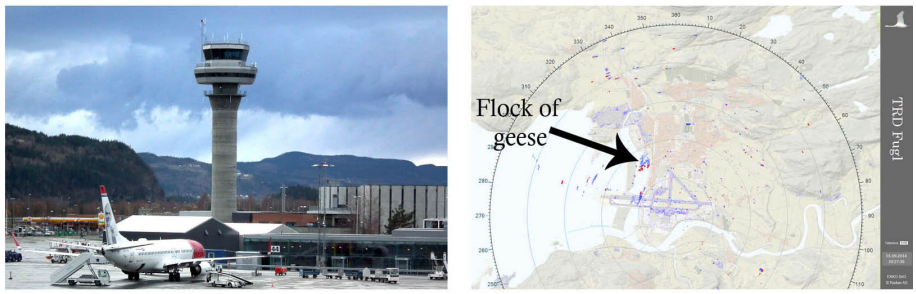


Figure 3.3: The Air Traffic Control (ATC) Tower at Værnes Airport seen from the radar site (left) and the current interface of the MTD output that is available for the ATC operators inside it (right).

video and review previous recordings. Thus, the clients may also perform tracking with an Early Warning (EW) functionality on real-time or near-past video. This is of particular interest to the client computer located at the ATC Tower.

Figure 3.4 illustrates the processing chain from the IF part of the radar receiver to the output consisting of radar video with tracking information and early warning decisions. As shown in Figure 2.1, the superheterodyne radar receiver down-mixes the received signal to IF (60 MHz) before the IF amplifier. In this system, the cascade of the radar receiver and the computer server of Figure 3.1 is effectively a pseudocoherent radar receiver like the one shown in Figure 2.11. The IF amplifier of Figure 3.4 is the same as the one in Figure 2.11. Thereafter, a Doppler Signal Processor extracts the Doppler signal from the coherent pulses into a quadrature channel pair I and Q. The Doppler Signal Processor block is very much like the MTI processor described in Figure 2.9, only it outputs the I and Q channel directly without the MTI filtering and video detection.

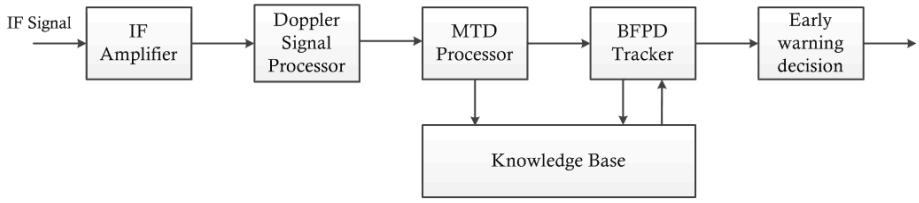


Figure 3.4: Block diagram of the processing chain.

The next step of the processing chain is the MTD processor block, which is given in Figure 2.10. The MTD processor also feeds the Knowledge Base (KB) with amplitude information to define a map of High Clutter (HC) areas which is utilized by the tracker to prevent false tracks. The tracker takes input from both the MTD processor and the KB and outputs the radar video with tracking information. Its internal structure (shown in Figure 4.1) is thoroughly described in Chapter 4. Finally, an early warning function monitors the tracking information and decides whether or not to give a warning signal by comparing the tracker information to some warning situation requirements.

During development of the tracker on the PC setup with MATLAB, the described pro-

cessing chain was interrupted. Right after the Doppler Signal Processor in figure 3.4, physically located within the computer server of Figure 3.1, the I and Q samples were stored as complex matrix files with a meta-data header. These files were transferred to a personal laptop where the rest of the processing chain was performed by the author in a MATLAB environment.

3.1.2 Radar Specifications

Table 3.1: Furuno FAR-2127BB radar specifications. [38]

Radar Model	Furuno FAR-2127BB
Transceiver	RTR-079A
Magnetron	MG5436
Antenna Gain	Slotted waveguide 8 feet, XN24AF 31.5 dBi
Duplexer	Ferrite circulator w/ diode limiter
Beam width (H) θ_B	0.95°
Beam width (V)	20°
Polarization	Horizontal
Rotation ω_r	25.7 rpm
Frequency f_c	9410 MHz $\pm$30 MHz
Output Power P_t	25 kW
PRF mean f_p	3002 Hz
Pulse duration τ	150 ns
First blind speed v_b	47.8 m/s
CPI n	18.5 pulses
Range resolution	22.5 m
Doppler resolution	2.58 m/s
Sampling freq. F_s	40 MHz (range)
IF f_{IF}	60 MHz

Table 3.1 shows the radar system specifications. The rotation rate and mean PRF are measured from the radar output, while most other values and data of Table 3.1 are either stated in the Furuno Operator's Manual [38] or derived from other values. The derivation of the latter may be found in Appendix B.

Table 3.2 gives a specification of the MTD processing parameters. The MTD Filter Bank specifications describe the Filter Bank implementation in MATLAB, while CFAR method specifications describe the fixed parameters of each method.

The choices of many of these parameters are based on the author's previous work on the MTD processor in [13]. The choice of window size of $N = 16$, two guard cells in CA-CFAR and none in OS-CFAR, and the OS-CFAR rank of $3N/4 = 12$ is also readily found described and documented in the literature (for instance in [53]) as generally well-

Table 3.2: Detection and processing system parameter specifications

MTD	
MTD filters	FFT Filter Bank
Filters	16
FFT length	16
Gaussian window α	1.1
CA-CFAR	
Window size N	16
Guard cells	1 on each side of the CUT
OS-CFAR	
Window size N	16
Guard cells	0
Rank	12
CM-CFAR	
Update coefficient α	0.2
Input channels	#1 (zero velocity)
Working channels	#1, 2, 16
Amplitude mult.	Ch.1: 1, Ch.2&16: 0.8

working parameters. The Clutter map working channels and amplitude multiplier in each of these channels were experimented with and established from an analysis of the land-clutter main-lobe in the Doppler domain.

3.1.3 Considerations and Priorities

In the process of designing and implementing the tracker system, several underlying considerations and priorities influenced the general decision-making. Some of these issues occasionally rise to the level where they define whole parts of the system in ways that might seem strange when seen from a traditional tracking perspective. The most important such issues, considerations and priorities are listed below.

- The main intent is to detect the presence and location of birds, and identify potential hazardous situations with high reliability. The intent is not to track individual birds with high accuracy, nor accurately estimate the number of birds by the means of a tracker.
- The risk of missing a true target is much more acceptable than the risk of giving a false warning. This is due to that if the ATC operators get the impression that the tracker and warning system cries wolf frequently, the system will not become an effective tool. However, if a potential situation does not produce a warning, it is most likely not a high threat situation, and the operators still have access to the MTD radar video for manual identification as well as their current non-radar methods.
- The most hazardous birdstrike situations arise from the presence of multiple large birds. These are easier to detect, and most of the time, they fly faster and maneuver

less than small birds. Examples of decisions that are influenced by this is the establishment of the tracker's target dynamics model and the employment of a CM built from the zero-velocity channel into the other non-zero-velocity channels.

- By designing the tracker as the proposed BFPD Tracker, having an emphasis on the detection of potential birdstrike situations, and retaining a batch processing structure, several common challenges and problems in multi-target tracking are circumvented. Some examples of common issues in classical iterative tracking methods that the proposed tracker circumvents are filter-linearity issues, measurement error estimation, covariance matrices and track-merging.

3.2 Simulation

When evaluating a radar tracker system, many quantities of interest are often related to the *true* properties and movement of the observed targets. The error of the tracked target measured position and velocity is one example of such quantities. Since the true target characteristics most often is not known, it is difficult to perform a thorough analysis solely from real radar measurements. Simulation is one of the most common techniques for evaluating the performance of a tracker. A common approach is to design a Monte Carlo simulation, which repeats simulation many times in order to analyze the probabilistic properties of a system. In such simulations, the data input is purely artificially constructed probabilistic data that is not obtained from measurements, but merely designed to test the system.

The simulation method in this thesis uses a hybrid of synthetic data and real measurements, and is meant to give highly realistic data input while sparing much of the time and effort that is needed to design a Monte Carlo simulation. The approach is to define a synthetic bird target and a target flight model before injecting this target into a realistic noise and clutter environment by adding the target to the real measurements. Since the tracker is designed for use in a birdstrike avoidance radar system where flocks of large birds such as geese are of a particular interest, a Swerling Case 1 target model is used in the simulation.

The Central Limit Theorem states that the total voltage signal amplitude returned from a dense flock (within one radar resolution cell) of m birds (of approximately equal size) is Gaussian distributed as m approaches infinity regardless of the individual birds' statistical properties. The total amplitude may however be approximated as Gaussian while $m > 4$. With a Gaussian signal amplitude in both the I and the Q channel, the detected magnitude $\sqrt{I^2 + Q^2}$ is Rayleigh distributed. In the case of a Swerling 1 target model, the target RCS is Chi-squared distributed with two degrees of freedom (equivalent to the Exponential distribution) and is decorrelated (independent) from scan to scan. The resulting signal power received from such a target (derived from Equation 2.1) also follows the Exponential distribution. The amplitude of the voltage (square root of power) signal is Rayleigh distributed with a mean corresponding to the voltage produced by the target's mean RCS σ_t. The Swerling 1 model - giving a Rayleigh distributed amplitude, is thus appropriate when the target consists of many ($m > 4$) individual scatterers of approximately equal size, which is the case for a flock of geese. [6, p.120-124]

First, a synthetic bird target is defined by constructing a two dimensional Gaussian distribution with a spatial extension given by the antenna beam-width in azimuth, and the

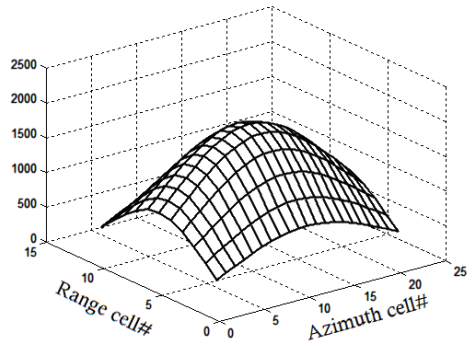


Figure 3.5: The 2D Gaussian amplitude of a synthetic target used for simulation of bird flight.

radar range resolution in range. This gives the target's amplitude distribution that is shown in Figure 3.5.

To make the synthetic target a Swerling 1 fluctuating target, the target peak amplitude is drawn from a Rayleigh distribution for each target observation/scan. [40] The mean of this distribution, and thus the mean of the peak amplitude, is set to a level (2000) that is frequently observed with real targets. This mean peak amplitude is also kept constant, independent of target range, because the Furuno radar employs an Automatic Gain Control (AGC) to adjust the gain for different target ranges. By this method, the mean Signal to Noise and Clutter power Ratio (SNCR) of the simulated target was measured at 15.84 dB.

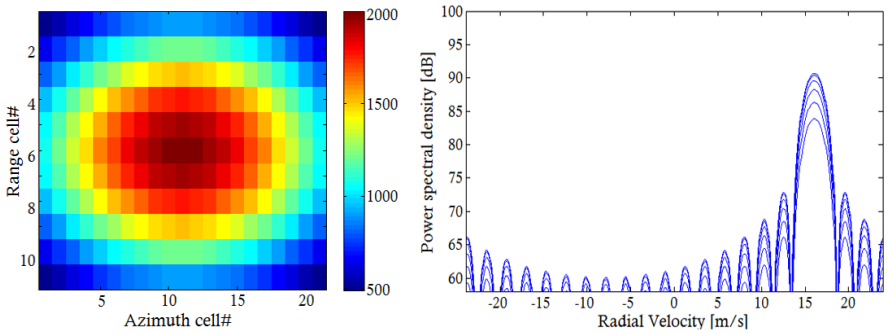


Figure 3.6: The amplitude (left) and Doppler spectrum (right) of the synthetic target shown in Figure 3.5 used for simulation of bird flight. Here shown with modulation corresponding to a target radial velocity of 16 m/s.

During generation of the simulation data, the Doppler shift of the synthetic target is found from the target flight model which describes the exact velocity and geometry between the target and the radar. For each antenna scan, the target's position, speed and course is calculated from the target flight model. The synthetic target is then modulated to the corresponding Doppler frequency and added to the real measurements at the appropriate measurement time and place. Figure 3.6 shows the synthetic target's resulting Doppler

spectrum for a radial velocity of 16 m/s. Figure 3.7 on the other hand, shows an example of what the target looks like in amplitude and Doppler when added to a background of real sea clutter and noise.

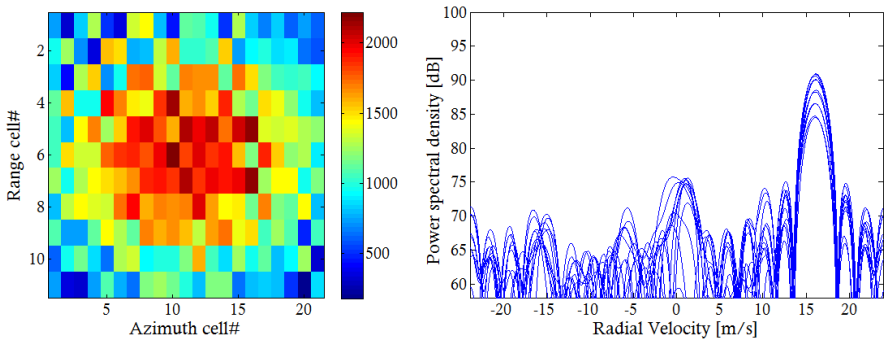


Figure 3.7: The amplitude (left) and Doppler spectrum (right) of a simulated bird target (16 m/s radial velocity) when added to a real sea clutter and noise background.

The target flight model is a model that describes the synthetic target's motion. The target is set to fly in a straight line with velocity components v_x and v_y, starting in the position (x_0,y_0). The target position n scans later p_n is then given as a function of the observation time difference nT. When locating the range-azimuth cells corresponding to the position $p_n = (x_n, y_n)$, the algorithms find the cells that correspond best to p_n based on the azimuth angle-information given in the header of each pulse.

3.3 Datasets

About 77 GB/h (50 MB per scan) of radar data is continuously being recorded by the radar at Værnes Airport. Due to the need for small, relevant sets of radar data for testing and development, three sets of data have been extracted from the daily flow of radar data. These are as follows.

Case 1: This case contains 177 scans worth of radar data in which there is a flock of unknown birds (probably seagulls) incoming towards the airport runway from the West, over the sea in the fjord Trondheimsfjorden. When approaching land, the flock splits into two flocks, one of which continues in a South-East course, flies straight over the airport and disappears to the South-East on the far side of the river.

Case 2: This case contains 283 scans worth of radar data in which there is a flock of unknown birds taking off from a farming field North-East of the airport. The flock is large and inbound to the far East end of the runway.

Case 3: This case contains 182 scans worth of radar data in which there is a large flock of geese incoming from the North. Simultaneously there are seagulls incoming from North-East. The flock of geese flies South along the coast-line to intersect the runway, where it splits into two flocks. One of these turns to a South-West course at the runway intersection, while the other geese maneuvers in a sharp turn and heads North-West out over sea along with some other unknown birds. The last third of this case also contains an aircraft. This aircraft is taxing West towards the part of the main runway where the flock of geese intersect, and takes off to the East right after the geese have passed. This was however not a potential birdstrike situation since the birds flew over the aircraft (within a couple hundred meters) while it was still taxing.

Additionally, all cases contain an unknown number of birds that are not mentioned in their description above.

3.4 Initial calculations

3.4.1 P_d-calculations

As an initial analysis of the detection capabilities of the radar system, some calculations of probability of detection P_d versus range for different bird species and RCS models were performed. As described in Section 2.2.1 P_d cannot be evaluated by simple means, but is often calculated using numerical computer estimation or empirical methods. Here, P_d for different birds were calculated using two methods. The first is by the computer tool Computer Aided Radar Performance Evaluation Tool 2 (CARPET 2) [22], and the second is to calculate SNR "in-the-clear" by means of a Microsoft Office Excel spreadsheet, importing to MathWorks' MATLAB, transforming SNR to P_d and plotting the results. The Excel spreadsheet was made and provided by supervisor Yngve Steinheim, and is shown in Appendix D.2. All calculations in this section assumes a Swerling Case 1 target and a probability of false alarm $P_{fa} = 10^{-3}$.

Calculations with CARPET allows for calculating with different sea- and/or land-clutter models and is taking into account the radiation diagram of the antenna. The Excel spreadsheet assumes that the target always is in the middle of the antenna's main-beam and uses a free-space model with the target being the only scatterer present. There are also several other differences in the way these two methods calculate P_d.

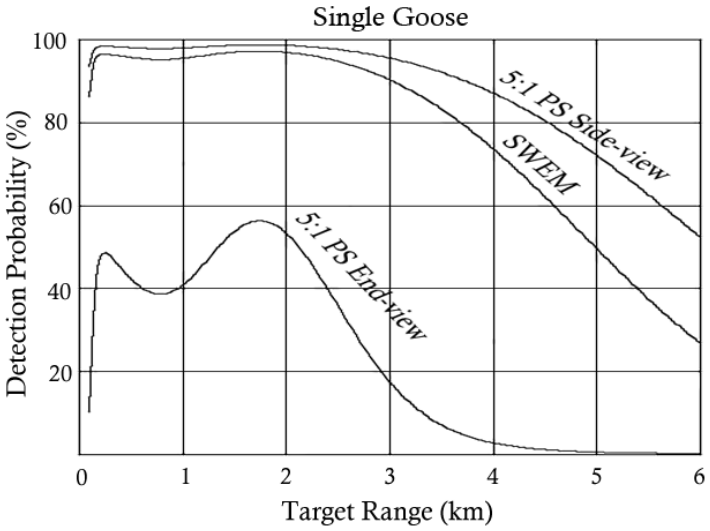


Figure 3.8: Probability of detection P_d as a function of target range for a single goose by the 5:1 PS and the SWEM RCS-model. Calculations assume a Swerling Case 1 type target and include sea state 2 sea-clutter, which is responsible for the dip in the curves around 1 km.

The curves given in Figure 3.8 were calculated by CARPET 2. These use a Swerling Case 1 target model, the 5:1 Prolate Spheroid (PS) model (by Vaughn [63]) and the Sphere of Water with Equal Mass (SWEM) model for a single goose, accounted for in Section 2.4, and use a sea state 2 sea-clutter model.

Figure 3.8 shows that the PS side-view and end-view models constitute a best-case and worst-case respectively. The SWEM model gives more of an average-case. The figure also shows that by demanding $P_d > 0.7$, a single goose by the SWEM model may be detected out to approximately 4.2 km.

As may be concluded from Section 2.4 on birds as radar targets, as well as from Figure 3.8, there is great uncertainty associated with the calculations of P_d due to the uncertainty and inaccuracy of the models of bird RCS. However, these calculations may provide useful for estimating detection probability, design and dimensioning of the tracker or other components related to the total system.

Figure 3.9 shows the probability of detection for a single bird as a function of target range for different species. These calculations are based on mean RCS values from the SWEM model as stated in Table 2.1.

By Figure 3.9, a single sparrow, seagull, goose and swan may be detected with a probability of $P_d = 0.7$ at ranges of 1.9, 2.8, 3.5 and 4.2 km respectively. Compared to that given in Figure 3.8, the range for a single goose at $P_d = 0.7$ has dropped from 4.2 to 3.5 km. This difference arises because there are many factors at play in the calculations of P_d, and the two methods include different parameters and sometimes use slightly dif-

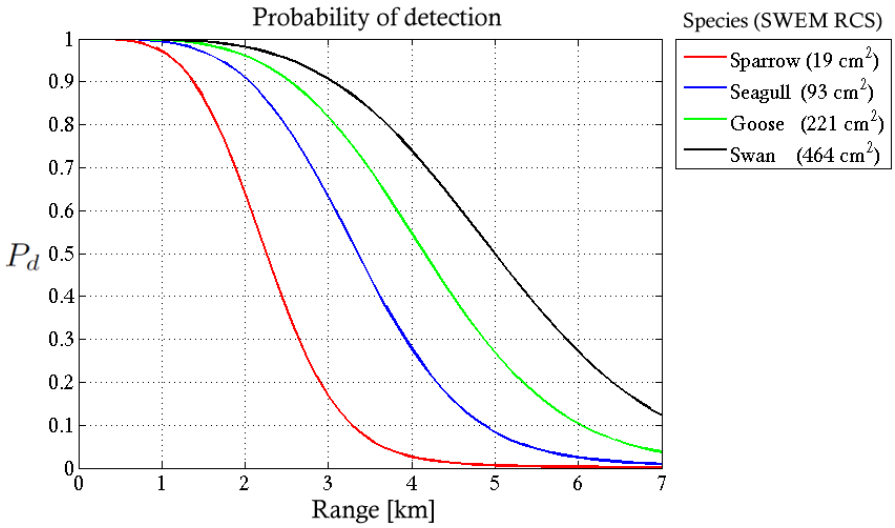


Figure 3.9: Probability of detection P_d for a single bird as a function of target range for different species using the SWEM RCS model and a free-space propagation model.

ferent values for the same parameters. For instance, the method for Figure 3.8 includes sea-clutter in its calculations, while the method for Figure3.9 does not. The two methods of calculation are fully accounted for in Appendix D.

These curves also indicate that if for instance seagulls are reliably detected and tracked at a range of 4 km, there are probably multiple birds present within the resolution cell (which is about 22.5×66 m at 4 km). Multiple birds in flock increase the flock RCS proportionally to the number of birds in the flock (see Equation 2.22). The presence of multiple birds within the resolution cell will therefore significantly improve on the detection capability presented here.

3.4.2 Clutter Attenuation (CA)

Figure 3.10 shows the Doppler spectrum of an approximated point target (a single runway landing-light lantern) at Værnes Airport. This figure clearly shows that the clutter energy is attenuated for frequencies and velocities that are not close to zero. Read from the figure, the power spectral density of the clutter is attenuated roughly 30 dB for the MTD channels #3 through #15. This is a result of pulse-to-pulse signal coherency and the FFT filtering that is performed within the MTD processor.

Normalized Clutter Attenuation (CA), defined by Equation 2.19, is calculated and displayed in Figure 3.11. Here, CA is calculated as a function of 3x decimated PRIs that translate to an azimuth sector of 220°. A narrow annular sector of 220° by three range-cells (at about 200m in range) at the output of MTD filter number 4 (ca.$10 - 14$ m/s) is used in the calculation of CA.

By taking the mean of the CA presented in Figure 3.11, the mean CA is found to be $\text{CA} = 26.0$ dB.

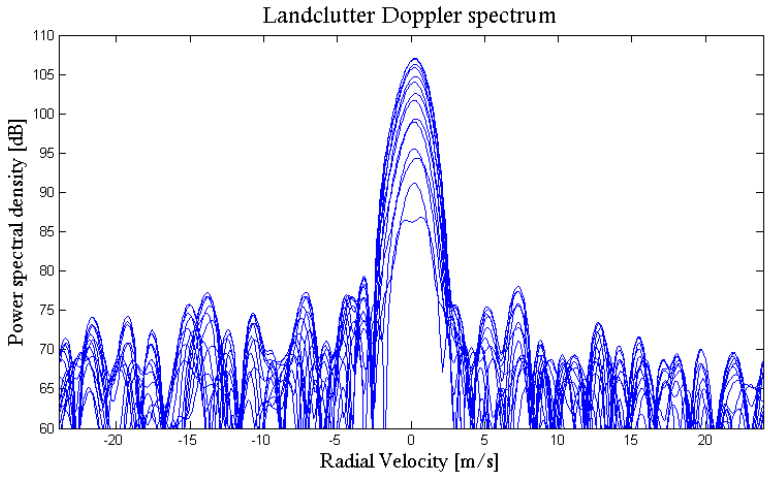


Figure 3.10: The Doppler spectrum of stationary, approximated point clutter (single runway landing-light) at Værnes Airport.

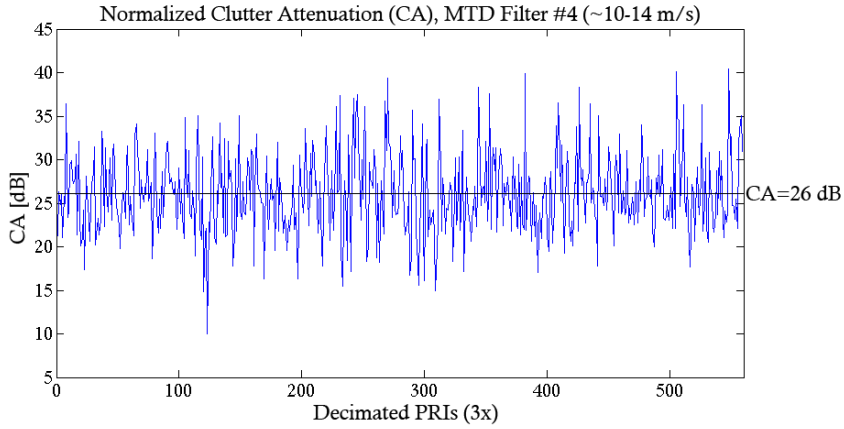


Figure 3.11: The Normalized Clutter Attenuation (CA) in MTD filter #4 (ca.$10 - 14$m/s), plotted against 3x decimated PRIs (azimuth) for an annular sector 220° wide and 3 range-cells long, mostly consisting of land clutter. The mean value of CA $= 26.0$ dB is also shown.

Chapter 4

Tracker Design and Analysis

This chapter will outline the architecture of the developed tracker, provide an analysis of the tracker and explain some of the choices and priorities that were made.

Figure 4.1 shows a block diagram of the internal structure of the tracker and Figure 3.4 shows where the BFPD Tracker fits into the rest of the processing chain.

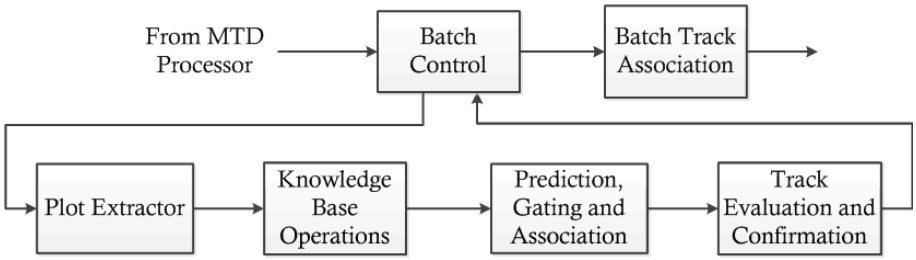


Figure 4.1: Block diagram of the BFPD Tracker.

Please note that several terms (e.g. *track*, *BFP* and *batch association*) used throughout this thesis is specifically chosen to describe concepts that may differ somewhat from both the reader's intuitive and the common usage of this term. The precision of these terms is especially important in this and the following chapters of the thesis. Therefore, the reader is referred to the Glossary at the far back of the thesis for the definition of terms that are commonly used in the thesis and have a very specific meaning that may not be intuitively understood.

4.1 Input

The MTD processor outputs data in the form of one NxM matrix per scan of the antenna, where N is the number of range-samples plus the number of header-elements and M is the number of pulses recorded from the scan. This matrix is called the Detection Matrix

and consists of detections on a background of zeros. The numerical value of a single detection represents the Doppler-velocity channel in which the detection was established, and thus also its radial velocity. Since the range-azimuth grid of the Detection Matrix is finer than the radar resolution (determined by the pulse length τ and the CPI), detections will typically lump into ”blobs” of connected detections called *detectionheaps*. This is also caused by decimation and interpolation of the Detection Matrix during the MTD filtering and detection process. The detectionheaps will have different shapes, but those that originate from high SNR targets with small spatial extension will most often have a rectangular shape in the Detection Matrix.

4.2 Plot Extraction

Since the tracking of targets that consist of multiple positions and measurements is complicated, tracking is performed on point targets called plots. The system therefore needs a plot extraction algorithm to define the plots from the input Detection Matrix. The plot extraction process is traditionally not considered part of the tracker itself, but rather an outside process that feeds the tracker with plots. However, in this system the plot extractor is something the tracker cannot function without, and is therefore simply defined as part of the BFPD Tracker. [69]

Adjoining detections showing 4-connectivity (see Figure 4.2) in the Detection Matrix are in this thesis called a detectionheap. Small detectionheaps are often assumed to originate from single targets. The first part of the plot extraction problem hence becomes a Connected Component Labeling (CCL) problem with the detectionheaps being the connected components. Large detectionheaps may originate from multiple closely separated targets and require a splitting function to divide the large detectionheap into an appropriate number of smaller ones. This issue will be addressed later in this section.

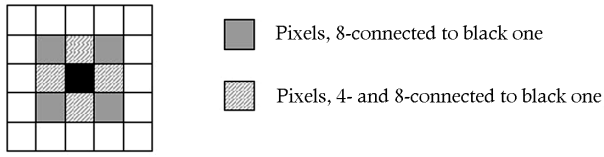


Figure 4.2: Pixel connectivity patterns.[32]

CCL (also called ”blob-extraction”) is an algorithmic application of graph theory and is commonly used in tracking and computer vision applications.[65] There exists a large number of algorithmic implementations that solve a CCL problem. [27][60][26] Additionally to labeling every detectionheap, the plot extractor needs to establish the target plot’s position as the coordinates of the connected component’s Center of Mass (CoM) and establish the plot’s Time of Observation (TO), scan, Doppler and so on. Finally, the extracted plots are stored in a list of plots called the Plot Matrix. The Plot Matrix of this tracker system is established with the structure according to Table 4.1.

As shown in Table 4.1, each plot in this tracker system consists of 8 elements. The first two elements are the Cartesian coordinates of the plot’s position. The coordinates x

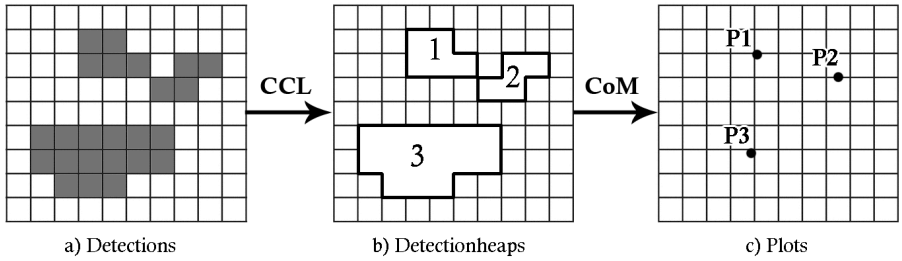


Figure 4.3: The process of plot extraction with Connected Component Labeling (CCL) employing 4-connectivity and Center of Mass (CoM) plot positioning. This figure only shows the principle of operation, and is not an example of real data. Note that if 8-connectivity were used, detectionheaps number 1 and 2 would become one detectionheap and result in a single plot.

Table 4.1: Structure of the tracker's Plot Matrix.

x_1	y_1	TO_1	$Doppler_1$	$Size_1$	R-cell#$_1$	Az-cell#$_1$	Scan#$_1$
x_2	y_2	TO_2	$Doppler_2$	$Size_2$	R-cell#$_2$	Az-cell#$_2$	Scan#$_2$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

and y are that of a East North Up (ENU) coordinate system where the y-axis points along true (geodetic) north direction and the origin is the radar. The coordinate system is also stretched in such a manner that the coordinates is given in meters. For example, the point (100,100) is located 100 m North and 100 m East of the radar. The dimensions of x and y is also called *Easting* and *Northing* (relative to the radar site). For reference, the radar is located at 63.45429° N, 10.92787° E.

The next element is the Time of Observation, which is the exact time at which the plot measurement was observed. Then follows the plot's Doppler shift and size. The Doppler shift reflects the doppler components of the detections, while the size is the number of detections in the detectionheap from which the plot was established. The next two elements are the range cell number and the azimuth cell number of the cell in which the detectionheap's calculated CoM is located. The last element of a plot, is the scan number from which the measurement originated.

One of the plot extraction algorithms implemented in the tracker is a Two-Pass CCL algorithm [71][65][26][60] that iterates the Detection Matrix in two passes. This algorithm can be found in the digital appendix Appendix E under the name *GeneratePlotmatrix_TwoPass.m*. In the first pass, the algorithm labels all the detections and records which labels are equivalent. In the second pass it resolves all label equivalencies making every detectionheap uniquely labeled. This situation is shown in the transition from a to b in Figure 4.3. The detailed principle of operation of the implemented Two-Pass CCL algorithm that assumes 4-connectivity is as follows. For a graphical and easily understood presentation of the clockwork of a generic Two-Pass CCL algorithm, the reader is referred to [65].

First pass

1. Define a Label Matrix L equal in size to the Detection Matrix D, but that consists of elements $L(i, j)$ that are all equal to zero.
2. Define a label counter $c = 1$, a label-equivalence list E and a detectionheap list H.
3. Iterate left to right through all cells $D(i, j)$ in the top row in the Detection Matrix and repeat for all rows down to the bottom.
4. If the cell $D(i, j)$ is not equal to the background (i.e. has a value larger than zero), check whether North-neighboring cell $L(i-1, j)$ and West-neighboring cell $L(i, j-1)$ in the Label Matrix have already been assigned a label number. Then perform the following logic:
   - If neither the North- nor West-neighboring cell have been assigned a label, $D(i, j)$ is considered member of a new detectionheap and $L(i, j)$ is assigned a label number from the label counter. Then increment the label counter.
   - If the North- and West-neighboring cells have been assigned the same label, or if only one has been assigned a label, assign this label to $L(i, j)$.
   - If the North and West-neighboring cells have been assigned different labels, record into E that these labels are equivalent, and assign the lowest of the two to $L(i, j)$.
5. Tidy and restructure the label-equivalence list E such that it consists of unique and nonoverlapping sets of equivalent labels.

Second Pass

7. Iterate the elements $L(i, j)$ of the Label Matrix in the manner as described in step 4.
8. If the cell $L(i, j)$ has been assigned a label (i.e. has a value larger than zero), find the set in E that contains this label and assign to $L(i, j)$ the lowest of the labels in this set.
9. Record in the detectionheap list H that the element at range grid i and azimuth grid i is a member of detectionheap number $L(i, j)$ and has a Doppler value of $D(i, j)$.

Very large detectionheaps will still only result in one single plot. When a low threshold is set to obtain a high P_d, multiple closely separated targets may melt together to one detectionheap. With the Two-Pass CCL algorithm, such a detectionheap would then result in a single plot, losing information about the multiple targets. Therefore, a splitting functionality is introduced to the Two-Pass CCL plot extraction algorithm. This splitting function divides the large detectionheaps into several smaller heaps of a predetermined maximum size. The splitting procedure is as follows.

Splitting

1. Iterate through the detectionheap list H, identify detections that belong to the same detectionheap and repeat the following steps for all detectionheaps.

2. Start at the first detection of the detectionheap and calculate the grid distance to all other detections within the detectionheap. The distance D_{de} between detections d and e in range-azimuth cells (i_d, j_d) and (i_e, j_e) is calculated by

$$D = \sqrt{(i_d - i_e)^2 + (j_d - j_e)^2} \tag{4.1}$$

3. Isolate all detections that are within a maximum distance of $D_{max} = 20$ and extract these detections as a new detectionheap. After this, if any detections remain within the detectionheap in question, these are next in line for performing steps 2-3 on.

The maximum distance used in the splitting is set to $D_{max} = 20$ to achieve splitting into detectionheaps roughly the size of the radar resolution, which would result from a point target. For simplicity, this requirement does not take into account the differences of the range and azimuth dimension. A grid-distance of 20 azimuth-cells in the Detection Matrix approximately equals one degree in azimuth, which is about the same as the 3-dB-width of the antenna's main lobe in azimuth. D_{max} would in principle allow splitting into detectionheaps that are up to 40 grid elements wide, but is set to 20 and not half of that since the cell one starts measuring distances from is always located on the edge of the detectionheap. Therefore, a $D_{max} = 20$ results in splitting into detectionheaps roughly 20 grid elements in diameter or smaller. This is not a very precise method of splitting and it increases the computational load requirement of the tracking process because the number of plots is significantly increased. However, the splitting is practical and effective, greatly improving the tracking of large flocks or closely separated birds.

The other implemented plot extraction algorithm makes the simplification of assuming that all detectionheaps are rectangular in the Detection Matrix. In this system, this assumption is reasonable for most detectionheaps that originate from true targets when the false alarm rate is moderate. When searching for and labeling detectionheaps, this algorithm iterates the matrix left to right, top to bottom, and initiates a new search whenever it finds an unlabeled detection. This new search operates within the discovered detectionheap, and locates its assumed CoM by simply searching to find the heap's edges straight right (increasing azimuth) and straight down (out in range) from the cell of discovery. This algorithm may be found in the digital appendix Appendix E under the name *GeneratePlotmatrix.m*.

After performing the CCL part of the plot extraction algorithm, plots need to be defined into the Plot Matrix with all the information shown in Table 4.1. The position of each plot is established from each detectionheap's CoM, resulting in the plots shown in Figure 4.3c. The plot position is initially given by which range-azimuth-cell contains the detectionheap's CoM. From this range- and azimuth-cell number, the position coordinates x and y are derived through a modified polar to Cartesian coordinate transform. It follows that the precision of x and y is poorer than what is indicated in Figure 4.3 since the position of a plot P always will be in the center of a cell in the Detection Matrix. However, the detectionheaps of this system usually consists of many more detections than those shown in Figure 4.3, mitigating this inaccuracy.

Figure 4.4 shows an example of the resulting plots (black crosses) from the three algorithms Two-Pass CCL, Two-Pass CCL with Splitting and the Simplified Plot Extraction

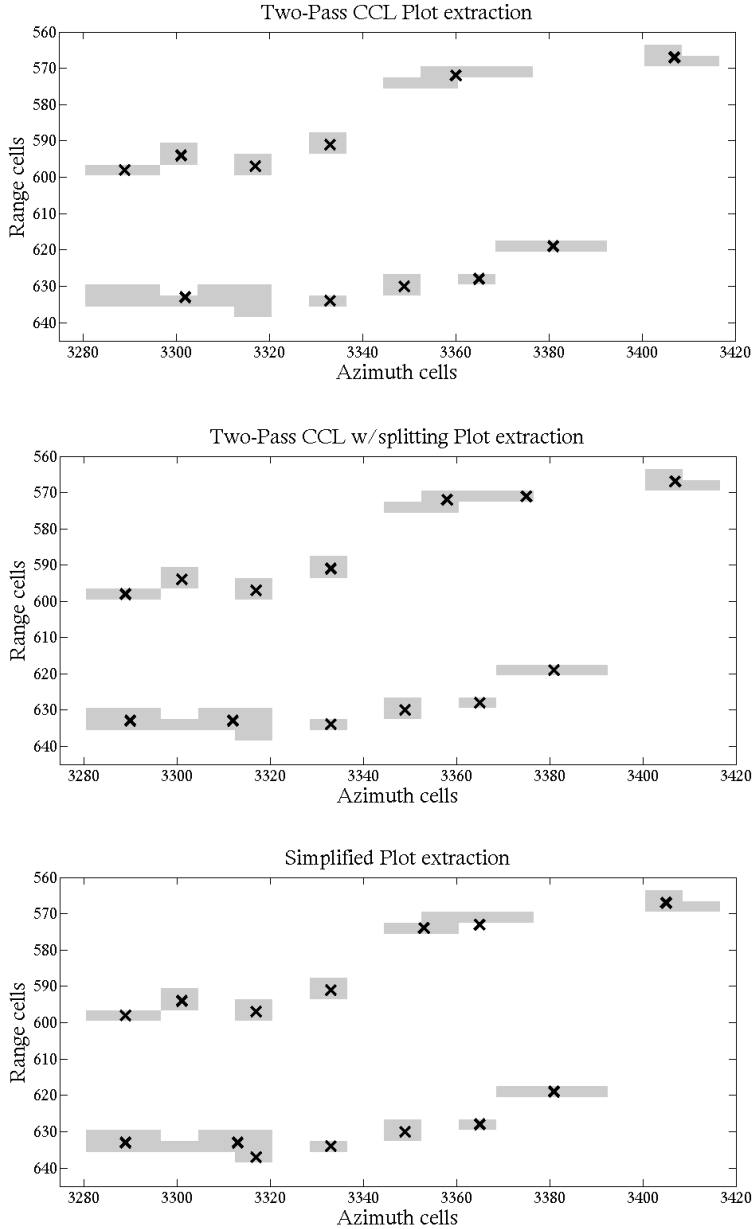


Figure 4.4: An example of how the three different plot extraction algorithms extract plots (black crosses) from different detectionheaps (gray) in a Detection Matrix made from the performed radar measurements. *Top:* Two-Pass CCL Plot Extractor algorithm. *Middle:* Two-Pass CCL with Splitting of large heaps into several plots. *Bottom:* Simplified Plot Extractor algorithm.

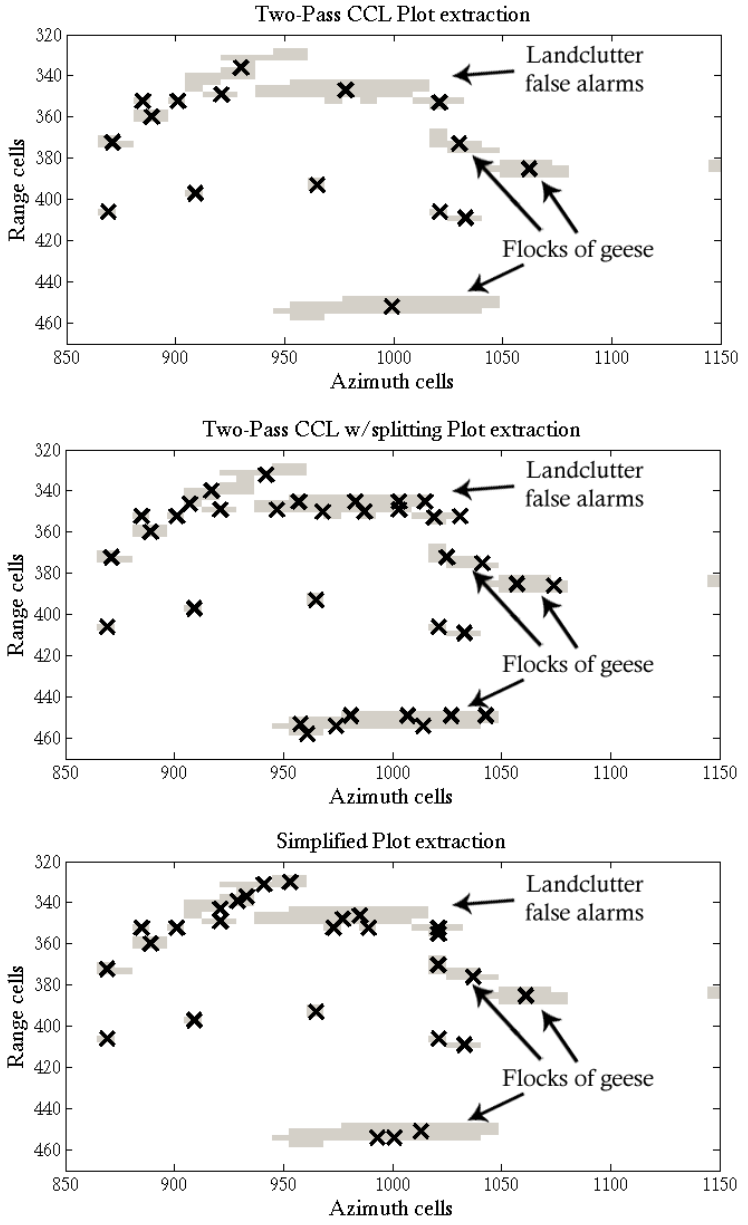


Figure 4.5: An example of how the three different plot extraction algorithms extract plots (black crosses) from different detectionheaps (gray) in a Detection Matrix made from the radar measurements of a large flock of geese. Black arrow indicators show the three detectionheaps resulting from the geese. *Top:* Two-Pass CCL Plot Extractor algorithm. *Middle:* Two-Pass CCL with Splitting of large heaps into several plots. *Bottom:* Simplified Plot Extractor algorithm.

algorithms when operating on a Detection Matrix output by the MTD processor. With detectionheaps of non-rectangular shape and an appreciable size, it becomes clear that the simplified plot extractor naturally splits large detectionheaps into more than one plot. The Two-Pass CCL w/Splitting achieves the same, but with plots in different numbers and at different locations due to the differences in procedure. At the same time, precision of the CoM estimate is better in the Two-Pass CCL than in the Simplified Plot Extraction algorithm. This may be observed in the two algorithms' placement of the plot that is established from the far upper right detectionheap, where the Simplified Plot Extractor algorithm gets a CoM that is slightly biased to the left.

In the example of Figure 4.4, the difference between these three algorithms does not seem immense. However, when extracting plots from several large flocks of geese when many parts of the flock are successfully detected in the MTD processor, the difference is much larger. This is shown in Figure 4.5. In such a case, which is of prime interest in the birdstrike-avoidance application, the Simplified Plot Extraction algorithm (Figure 4.5 bottom) does not produce enough plots to allow reliable tracking of the flock. While the Two-Pass CCL algorithm (Figure 4.5 top) accurately estimates the flock's CoM, it produces even fewer plots – only one per 4-connected detectionheap. The Two-Pass CCL w/ Splitting algorithm (Figure 4.5 middle) is designed to have a functionality that splits large detectionheaps into an appropriate number of plots, and is thus the most suited algorithm for the job. The Two-Pass CCL with Splitting algorithm is overall far superior to the other two, in accuracy of both plot positioning and number of plots, and is thus the only plot extraction algorithm that is used.

4.3 Knowledge-Base Operations

The Knowledge-Base (KB) is a collection of information such as information of the environment, terrain, target characteristics, classification parameters, measurements by other systems and the like. A Knowledge-Based Tracker (KBT) exploits the information stored in the KB to achieve better performance in detection and tracking. Advanced KB systems may be implemented in cognitive radar and artificial intelligence systems. The BFPD Tracker utilizes a simple KB that consists two maps to perform the following KBT techniques.[24] [17] [9]

- *Delete Measurements in High Clutter areas (DMHC)*
  The DMHC technique uses the radar measurements to classify radar resolution cells (or areas of arbitrary size) as High Clutter (HC) or Low Clutter (LC). Plots output by the plot extractor with positions within these areas are deleted. [24, p.176-183] [9, p.1106]

- *Delete Measurements in Uninteresting Areas (DMUA)*
  The DMUA technique uses a predefined map that merges information of areas that for some reason should be excluded in the tracking process (i.e. roads, towns or areas of especially high false alarm rate). This map defines the Uninteresting Areas (UA). All plots whose positions are within the UA are deleted.

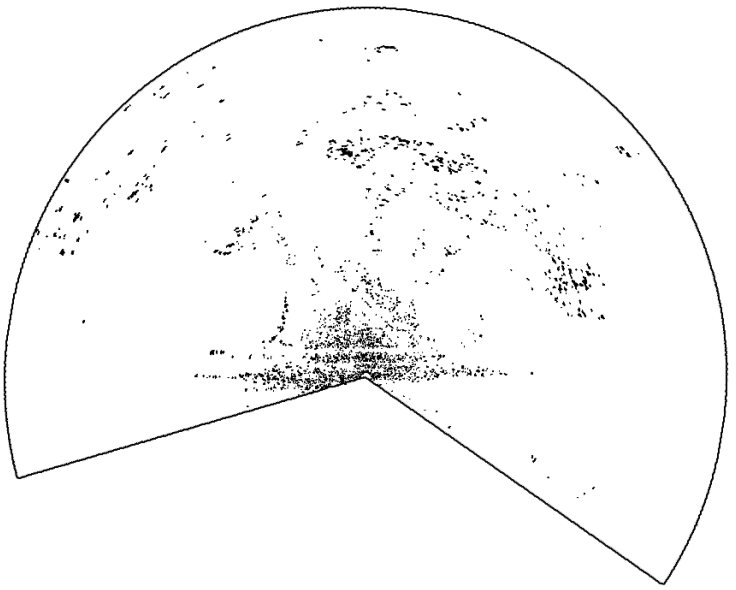


Figure 4.6: Example of a High Clutter (HC) areas map, generated by thresholding and filtering of the radar measurements performed at Værnes Airport.

The HC areas map is generated automatically from the radar measurements. There are many ways to implement this, but in this system it is implemented in the following four-step process.

1. For all cells, find the time-average of all scans in the current batch.
2. Threshold detect on the resulting image, setting all cells with an amplitude level below the threshold T_{HC} equal to zero. The threshold T_{HC} is set to 250, and the maximum possible amplitude (defined in the MTD processor) is 255.
3. Filter the resulting image through a two dimensional 2-cell spatial-averaging filter (the MATBAL function *fspecial*). This is equivalent to low-pass filtering the threshold-detected image in both dimensions with a FIR-filter with coefficients [0.25 0.25; 0.25 0.25].
4. Define all cells in the resulting image with an amplitude level > 0 as HC area cells.

Figure 4.6 shows the resulting image from such a process with 7 scans in a batch.

Contrary to the HC areas map, the generation of the UA map in this system is straightforward. The UA map is defined manually by analyzing the radar measurements and consulting with road and terrain maps. This results in the fairly simple looking UA map shown in Figure 4.7, where the main goal is to exclude observations of cars and especially tricky clutter areas.

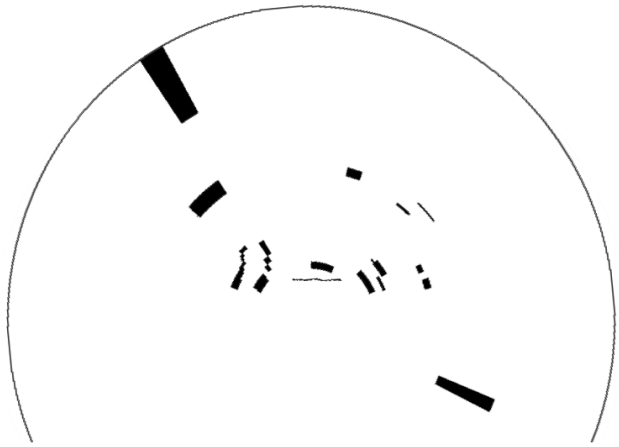


Figure 4.7: Example of a Uninteresting Areas (UA) map, used for blanking of specific areas.

The DMHC mitigates false tracks resulting from powerful land clutter. But clutter areas containing moving objects will give sporadic false plots (resulting from false alarms) also in the higher velocity channels. To avoid tracking on these false plots, such areas are included in the UA map.

4.4 Prediction, Gating and Association

In this part of the tracker, the tracks for the current batch are established. The tracks that are established in this block may be called tentative tracks, since they include all possible tracks found by the tracker (within the current batch).

To limit the number of tentative tracks (and required processing power) while maintaining reliable tracking ability of the targets of interest, the tracker establishes tentative tracks based on a target dynamics model. This model describes the patterns of movement that can be expected from the targets, and is in this case designed to allow tracks of large birds. The target dynamics model should also be as specific as possible to prevent the tracking of targets whose motion is outside what can be expected from the targets of interest.

The requirements of the target dynamics model used by the tracker is given in Table 4.2.

The establishment of tracks is performed in accordance with Figure 2.15, 4.8 and 4.9, and the description of Section 2.5 by the following procedure.

First, all potential starting points of a track within the batch are located by steps 1 through 5:

1. Start in every plot P measured in scan $i = [1, 2]$ of the batch. Starting in $i = 2$ assumes a target miss in the first scan.

Table 4.2: The BFPD Tracker's target dynamics model requirements, descibing the expected movement pattern of the large birds of interest.

Absolute	**Requirement**
Minimum speed	10 m/s
Maximum speed	23 m/s
During two observation intervals	
Maximum change in speed	8 m/s
Maximum change in course	20°

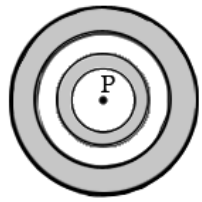


Figure 4.8: Principal illustration of the two annuli resulting from the velocity requirements of the target dynamics model and a time of one (inner annulus) and two (outer annulus) scans.

2. Establish an annulus around P by deriving a minimum and maximum distance from the minimum and maximum target speed given by the target dynamics model and a time difference of one scan period.

3. Locate every plot Q in scan $i + 1$ that falls within this annulus and initiate tracks from these pairs of plots (P,Q).

4. If no plots were found within this annulus, assume miss in scan $i + 1$ and establish a similar annulus based on a time difference of two scan periods.

5. Locate every plot Q in scan $i + 2$ that falls within this annulus and initiate tracks from these pairs of plots (P,Q).

For all established tentative tracks thus far – now consisting of plot pairs (P,Q) where P is in scan i and Q in scan j, hypothesize that these plots originated from the same target. Then perform the following steps (6-9) for all tracks. For each track, the following steps should be repeated such that all scans in the batch are included. P and Q should always be kept as the second latest (P) and the latest (Q) target observation within the current track. Likewise, i and j should always be kept equal to the scan numbers of P and Q respectively.

6. Predict the future position R of the target in scan $j + 1$ and establish an annular sector gate around R in the manner shown in Figure 4.9. This gate is established by the restraints that describe a target's maximum change in speed and course from scan to scan, which is given by the target dynamics model.

7. Locate all plots S in scan $j + 1$ that fall within this gate. If any are found, assign exactly one plot to the track according to a Nearest Neighbor (NN) logic (see Section

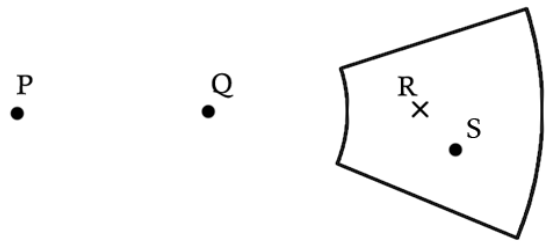


Figure 4.9: The prediction and gating process of tracking within a batch (steps 6-9). Note that in this figure, P and Q must be located in two consecutive scans and therefore do not have a miss in-between them.

2.5.2) where the distance d_n in question is the geometric Euclidean distance from R to S_n.

8. If no plot S is found, assume a miss in scan $j + 1$ and predict the future position R2 of the target in scan $j + 2$ and establish a similar gate around this point from the restraints in the target dynamics model. Since the time difference of Q and R

9. Locate all plots S in scan $j + 2$ that fall within this gate. If any are found, assign exactly one plot to the track according to a Nearest Neighbor (NN) logic.

A few especially noteworthy properties emerge from this algorithmic procedure.

- If a target is missed in the first two scans of a batch, it will not be tracked. This constitutes an built in initial 1-out-of-2 criterion within each batch.
- If a target anytime is missed in two consecutive scans, the tracker will stop tracking of this target.
- There is no restriction of the number of tracks that can include the same plot, so a single specific plot may often be included in multiple tracks. This gives rise to tracks with multiple common plots, but also increases the chance of keeping the track of a true target in a high false plot rate background.
- Many of the tracks initialized by this algorithm only consist of two plots. These are plots that happened to be within a distance of each other in time and space that corresponds to what is defined as a realistic bird velocity by the target dynamics model.
- This procedure by itself only operates within the realm of one batch. Therefore, all tracking information from the previous is omitted during the batch tracking process. This makes the tracker fail to exploit previous batch measurements when tracking within the current batch, but allows it to start with a clean slate for each batch.
- Whenever a target miss occurs, the allowed variability of the target course and speed causes the gate around R2 (in scan $j + 2$) to be twice as large as it would have been

around R (in scan $j + 1$). In the case of a miss in the second scan of a batch, the annulus around P will be twice as thick (and further out from P) to compensate for the uncertainty of the target velocity over two scan periods as opposed to one.

As a sidenote, consider the following. The BFPD Tracker allows the association of a single plot to several tracks, and in this way kind of resembles a Multiple Hypotheses Tracker (MHT) since the tracks will represents multiple hypothetical bird flight paths (BFPs) from the same measurements. It does not however employ many of the typical structural components of a MHT regarding hypothesis-tree expansion, hypothesis evaluation and deletion, which are at the core of MHT operation.

4.5 Track Evaluation and Confirmation

Before the track evaluation and confirmation, the tentative tracks found by the tracker include tracks of varying quality. The tracks' lengths also vary from two plots to the number of scans per batch. In the process of track evaluation and confirmation, different parameters and measures of track quality will be established. Every track whose quality does not meet the minimum quality requirements is then deleted. The tracks that survive this test constitute the confirmed tracks.

As described in Section 2.5, different measures of track quality are often accumulated into one number called the *track score*, which is calculated by a weighting function called a Score Function (SF). This tracker system implements the confirmation process in a more direct way, keeping all measures of quality separate and setting a minimum quality requirement independently for each measure. Despite this, for simple reference to the track confirmation procedure, it will be referred to as an SF.

The measures of quality utilized for track confirmation in the tracker and their corresponding requirements of quality is given in Table 4.3. These requirements are used unless otherwise stated.

Velocity measurements are either deduced from the geometry or the Doppler-shift of the observations. In the fourth requirement of Table 4.3, the velocity measurement from Doppler of each plot is compared to the velocity measurement derived from the change in position and time from the previous observation. The track quality measures of Table 4.3 that are based on the variance of a quantity uses the *var*-function of MATLAB to give an unbiased estimator of the population variance of the quantity within one batch. Since the tentative tracks within one batch have different lengths (number of plots), the variances are

Table 4.3: The quality measures that make up the Score Function (SF) of the track confirmation process and their requirements of quality.

Track Quality measure	**Requirement**
Plots per batch (M-out-of-N criterion)	4 out of 7
Normalized variance of geometrically deduced velocity	max 40 m^2/s^2
Normalized variance of Doppler-deduced radial velocity	max 50 m^2/s^2
Difference in radial velocity deduced from geometry and Doppler	max 2 w/ diff > 5 m/s
Normalized variance of size	max 10^4 $detections^2$

normalized by $N-1$ where N is the number of measurements in the track. Size is a value measured for each plot, consisting of the number of detections the plot was extracted from (See Chapter 4.2).

4.6 Batch Track Association

When the processing of a batch is finished, the tracker searches for correlation between the tracks of the current batch and tracks from the previous batch. This allows for track association between adjacent batches and supplies the basis for an early warning decision.

Track correlation between adjacent batches is implemented such that all tracks in the previous are attributed a track speed and course based on the last two measurements within each track. In accordance with implementation of track association within one batch (Section 4.4), for each track T in the previous batch, a future position R is predicted based on the track speed and course. R is based on a time of one scan period. A gate that is extended outwards in range is established around R to include possible future positions outward in range. This allows a specified number of misses between the last plot of track T, and the first plot of any track that is to be associated to T. The current batch is then searched for tracks that have their first target observation within this gate. If any such tracks are found, all these tracks are associated to track T.

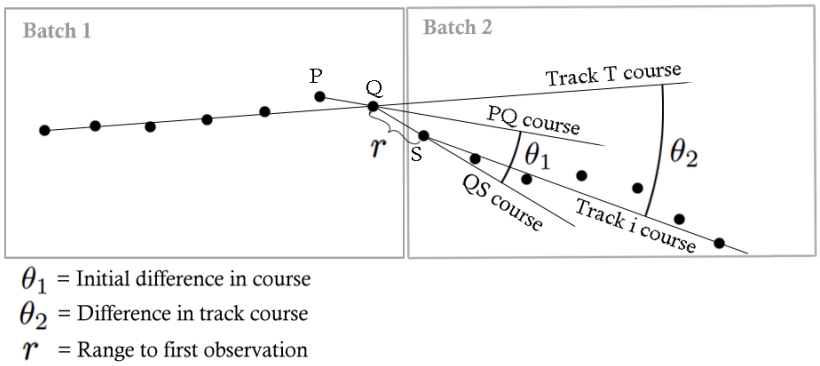


Figure 4.10: Defining the parameters Range r, Initial course difference θ_1 and Track course difference θ_2 in the batch track association procedure.

Figure 4.10 illustrates the definition of the range and course parameters θ_1 and θ_2 that is used in the batch track association procedure. The black dots represents target observations of a target in the Euclidean plane, seen from above, flying Eastwards and turning South-East. The first 7 observations belong to batch number 1, whereas the next 7 belong to batch 2. Course lines are defined from the first and two last observations in every batch in the manner shown in Figure 4.10. In turn, θ_1 and θ_2 are defined from these lines.

Figure 4.11 shows the gate establishment in the batch track association procedure. This gate is based on design requirements for r and θ_1 that needs to be fulfilled to make an association. The minimum and maximum requirements for the range r is derived from

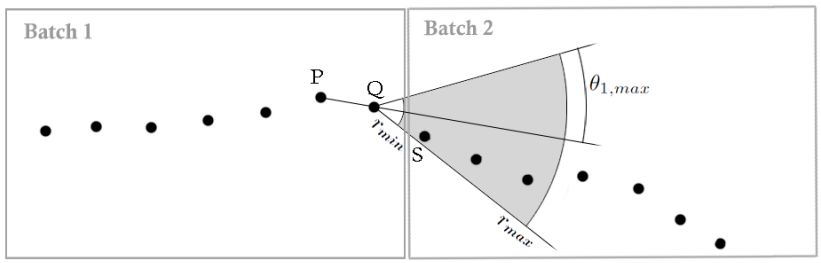


Figure 4.11: Gating in the batch track association procedure.

Table 4.4: The designed requirements for batch track association.

Between two consecutive batches	**Parameter**	**Requirement**
Maximum change in speed	v_{tol}	8 m/s
Maximum allowed number of misses	n_m	2
Maximum initial difference in course	$\theta_{1,max}$	30°
Maximum overall difference in track course	$\theta_{2,max}$	30°

the fixed requirements for change in speed between batches v_{tol} and the allowed number of misses between two consecutive batches n_m. In addition to this, the tracker operates with a maximum requirement for the difference in overall track course θ_2. This is to avoid making associations where two tracks in consecutive batches pass the gating test because they are closely separated, but are going in very different directions. The requirements for the batch track association procedure is given in Table 4.4

4.7 Early Warning

Though the Early Warning An Early Warning (EW) functionality is designed to give a signal that draws attention to potential hazardous birdstrike situations of a certain magnitude. For such a system to work properly, one first needs define what should classify as a situation that is worthy of activating such a signal. In the birdstrike-avoidance radar application, especially dangerous situations occur when a flock of heavy birds flies into the path of an aircraft. Additionally, an EW system ought to give warning sufficiently early so that precautionary measures might be taken. Hence, it is the author's view that an EW for this application should focus on the situations where large radar targets that are consistently tracked are headed on a course that will intersect a line defined by the airport runway. We call this line the *runway line*, which is shown in Figure 4.12. Note that this line extends East and West indefinitely.

The EW functionality in this system makes use of the batch associations that are established by the procedure described in the previous section. The EW identifies birds of sufficient size and number by looking for two batch associations that connect three confirmed tracks that are located in three consecutive batches. These confirmed tracks must also be likely to originate from the same target(s) according to the established tracking

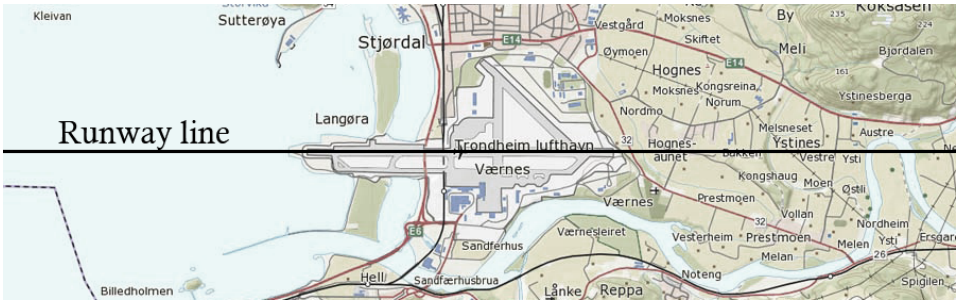


Figure 4.12: Definition of the ”Runway line” used in implementation of the Early Warning (EW) system. The runway line runs through the center of the airport runway and is located 370 meters North of the radar.

requirements and target dynamics model. Further, the system analyzes the course (direction of flight) of these tracks at the points where the batch associations were made (”QS course” in Figure 4.10) and determines whether the target(s) will intersect the runway line. The requirements for activation of an EW signal is summarized in Table 4.5.

Table 4.5: Requirements for activation of an Early Warning (EW) signal.

Requirement for EW activation	**Value**
Number of consecutive and logically coherent confirmed tracks	3
Number of consecutive batch associations	2
Course deviation at consecutive batch associations	Max. 30°
Track course at last batch association bound to intersect runway line	True

When the requirements of Table 4.5 are met, the EW signal can be used to highlight the track(s) of interest, sound a warning, or activate something else. From this implementation, it also is easy to make further restrictions of when the EW should be triggered for instance by synchronizing with a flight schedule and only giving a warning signal when there is an aircraft either inbound for landing or about to take off.

For this EW system, it is paramount that false warnings are extremely rare. If a warning signal should be meaningful enough to initiate precautionary measures, it cannot be subject to probable falseness (or even *perceived* probable falseness). However, while a false warning signal means that an EW is activated when there is no incoming bird(s), there is also the issue of doubt concerning the real magnitude of the situation’s seriousness. The radar and tracker cannot reliably differentiate the situations where a birdstrike is truly imminent because it can neither accurately determine the number and species of birds that are detected nor predict the future and the chaotic behavior of birds. Therefore an EW will merely direct attention to the situations where an imminent birdstrike may be possible and probable.

4.8 Markov Chain analysis

This section will give a brief analysis of how the probability of establishing a confirmed track is influenced by different criteria for target detection consistency. For simplicity, this section will assume that the track confirmation process only consists of requirements for length and detection consistency, and not any of the other measures of quality that is described in Section 4.5.

Given a target with a certain P_d, a larger probability of establishing a confirmed track induces a larger probability of also giving an early warning signal for that target. Likewise, for a certain P_{fa}, a larger probability of establishing confirmed tracks also induces a larger probability of establishing both *false* confirmed tracks and false warning signals.

The M-out-of-N-criterion gives the consistency criterion that a target has to be detected in at least M out of N scans for it to result in a confirmed track. The tracker, employing a 4-out-of-7 criterion and disallowing two consecutive misses, results in the state diagram given in Figure 4.13. The additional requirement of the target not being missed in any two consecutive scans, will make it less probable to establish confirmed tracks from false alarms (false confirmed tracks). However, it also decreases the probability of establishing a confirmed track from a true target with a certain P_d.

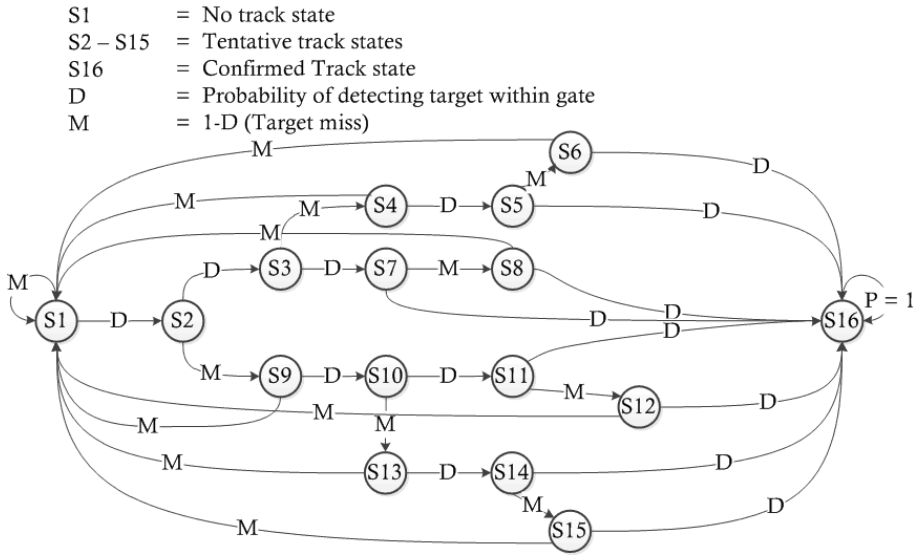


Figure 4.13: The state diagram of the tracker with a 4-out-of-7 criterion and disallowing two consecutive misses.

The state diagram of Figure 4.13 translates to the following Markov chain calculation setup (see Section 2.6). The corresponding Transition Matrix, describing the probabilities of state transition from scan k to $k + 1$, becomes

$$\Phi_{k,k+1} = \begin{bmatrix} M & 0 & 0 & M & 0 & M & 0 & M & M & 0 & 0 & M & M & 0 & M & 0 \\ D & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & D & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & M & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & D & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & M & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & M & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & M & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & M & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & M & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & M & 0 & 0 \\ 0 & 0 & 0 & 0 & D & D & D & D & 0 & 0 & D & D & 0 & D & D & 1 \end{bmatrix} \tag{4.2}$$

where D is the probability of detecting the target within the established gate at a given time of measurement and $M = 1 - D$.

Before the first scan within a batch, there are no recorded observations to track, so the initial state probability vector P_0 is

$$P_0 = [1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0]^T \tag{4.3}$$

Assuming a sufficiently large gate, high SNR and no interfering targets, P can be set equal to the probability of detection P_d. Therefore, by Equation 2.24, the cumulative probability of the system being in state S given a target with $P = P_d = 0.8$ after $k = 7$ scans is given by $P_7(S)$, where P_7 is given by Equation 4.4

$$P_7 = \Phi_{k,k+1}^7 P_0 = \begin{bmatrix} 0.01051 \\ 0.01889 \\ 0.02560 \\ 0.00512 \\ 0.00410 \\ 0.00082 \\ 0.02048 \\ 0.00410 \\ 0.00640 \\ 0.00512 \\ 0.00410 \\ 0.00082 \\ 0.00102 \\ 0.00082 \\ 0.00082 \\ 0.89129 \end{bmatrix} \tag{4.4}$$

Hence, the probability of establishing a confirmed track (system state #16) from that target after 7 scans is $P_7(16) \approx 0.891$. For comparison, Equation 2.9 gives the analogous probability for a pure 4-out-of-7-criterion (allowing any number of misses in a row) $P[4 \text{ out of } 7] = 0.967$. This probability is expectedly higher, because it is based on less restrictive demands of detection consistency.

To find the most relevant values of P_d and P_{fa}, Equations 2.11, 2.12, 2.14 and 2.15 were used. Table 4.6 shows the resulting P_d and P_{fa} for CA- and OS-CFAR with window

Table 4.6: P_d and P_{fa} for CA-CFAR and OS-CFAR with different threshold multipliers K_t given $N = 16$, $k = 12$ and SCR $= 15$ dB.

K_t		4	6	8	10
CA-CFAR	P_d	0.885	0.833	0.784	0.738
	P_{fa}	0.028	0.0061	0.0015	0.00042
OS-CFAR	P_d	0.854	0.790	0.731	0.677
	P_{fa}	0.015	0.0028	0.00067	0.00019

size $N = 16$, OS-CFAR rank $k = 12$ and an SCR of 15 dB. The probabilities are given for different threshold multipliers K_t and a Swerling 1 fluctuating target in Rayleigh clutter.

By extending the comparison to different criteria, different P_d and performing the equivalent analysis for the probability of establishing a false tracks from false alarms, the data of Table 4.7 is obtained. The corresponding probabilities for a 2-out-of-3 and a 3-out-of-5-criterion is also readily found by Equation 2.9 and included for comparison. The values for the Tracker implementation (labeled "Tracker" in Table 4.7) is calculated by the described Markov Chain technique.

Table 4.7: Probabilities of establishing a confirmed track from a target with P_d or consecutive false alarms with P_{fa} for different criteria of track establishment.

	Probability of establishing					
	confirmed track from detections (P_d)			false confirmed track from false alarms (P_{fa})		
P_d / P_{fa}	0.7	0.8	0.9	10^{-4}	10^{-3}	10^{-2}
BFPD Tracker	0.747	0.891	0.976	2×10^{-15}	2×10^{-11}	2×10^{-7}
4 of 7	0.874	0.967	0.997	4×10^{-15}	4×10^{-11}	3×10^{-7}
3 of 5	0.837	0.942	0.991	1×10^{-11}	1×10^{-8}	1×10^{-5}
2 of 3	0.784	0.896	0.972	3×10^{-8}	3×10^{-6}	3×10^{-4}

Table 4.6 shows some realistic values for P_d and P_{fa} given CA- and OS-CFAR detection. However, this table is based on the assumption of a target SCR of 15 dB in Rayleigh clutter. If the SCR decreases, or the clutter characteristics differs from Rayleigh clutter, both of which probably happens most of the time in the real application, the indicated probabilities of Table 4.7 may worsen significantly. If no clutter is present, and the SCR is replaced by an SNR of 15 dB, this SNR translates to approximately a 100 cm^2 target (a seagull by the SWEM model) at 2.5 km.

With this in mind, Table 4.7 gives an indication of the probability of establishing a confirmed track from a present target and a false confirmed track from false alarms when there is no target. As is shown, the tracker with its 4-out-of-7-criterion and disallowing 2 misses in a row performs comparable to that of a 2-out-of-3-criterion when there is a target present. In this case, the combined criteria of the tracker yield a slightly higher probability of establishing a confirmed track than the target's P_d. Figure 4.14 illustrates the general curves from which the data in Table 4.7 is sampled. As can be seen here, the probability of establishing a confirmed track is larger than P_d for the implemented when $P_d > 0.64$. In the contrary case, when there is no target present, the combined criteria

of the tracker yields a lower probability of establishing a confirmed track than that of a 4-out-of-7-criterion.

To confirm the correctness of the method of these Markov Chain calculations, probabilities for the pure 4-out-of-7-criterion was calculated by the same method. This resulted in a 36x36 Transition Matrix, for which the calculations should exactly duplicate the answers found by Equation 2.9, which they do. The Transition Matrix and Markov Chain calculations for the 4-out-of-7-criterion may be found in Appendix C

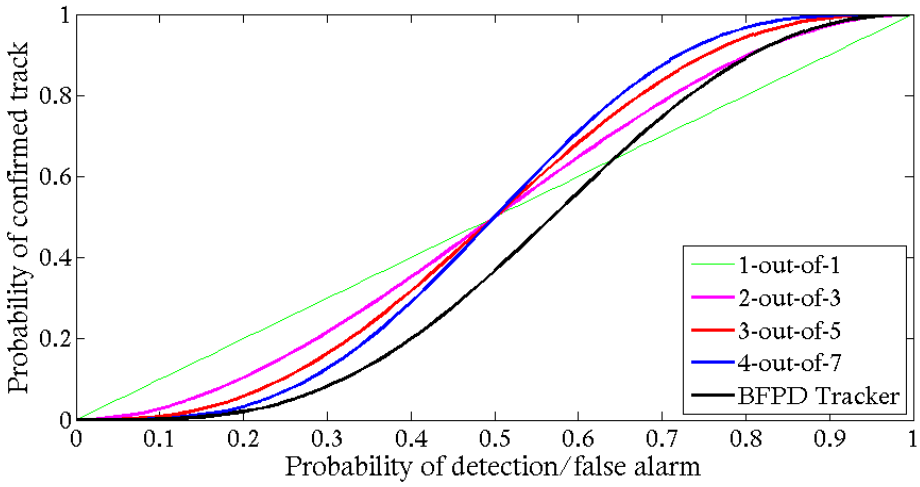


Figure 4.14: Theoretical probability of establishing a confirmed track as a function of the probability of detection P_d and/or the probability of false alarm P_{fa}.

Chapter 5

Results

This chapter will present the most important results from the developed BFPD Tracker. While an overall discussion will be given in the following chapter, immediate discussion regarding details of the results are given alongside the results themselves for easier readability.

First, the output of the BFPD Tracker (confirmed tracks and batch associations) will be presented. Then, the performance of the BFPD Tracker will be evaluated when seen in a cascade with the MTD processor. Thereafter follows results showing the effect of the Knowledge Base operations, and results showing the output of the developed Early Warning (EW) functionality. Then, the results from the simulation are presented before variations of batch length and track confirmation criteria are investigated.

It is important to keep in mind that the goal of the thesis and the developed BFPD Tracker is to generate output that is mostly visual in its nature. The final system output is mostly constituted of radar video with overlying tracking data. To present this resulting video output in paper form, the time dimension of the output is ignored and the output from each set of data is shown as a single image. Thus, when interpreting the results, there is a substantial advantage gained by watching the video data compared to the images that ignore the time dimension. Video files for all the results presented in this chapter are included in the digital appendix that follows this thesis. For better interpretability of the results, the author would like to strongly urge the reader to watch the videos and material that is found in the digital appendix as an addition to the figures and tables of this chapter.

While most of the figures in this chapter show a time history of a whole case (about 180 scans, or 7 minutes), it is important to understand that all decisions made by the BFPD Tracker and EW function is only made on the basis of the information that is given in the current and previous batches. In other words, no future information beyond the current batch is available for decision-making in the system at any time, though all history is shown a single picture. While this is intuitively shown in the videos, the figures might be misleading as they ignore the time dimension of the video.

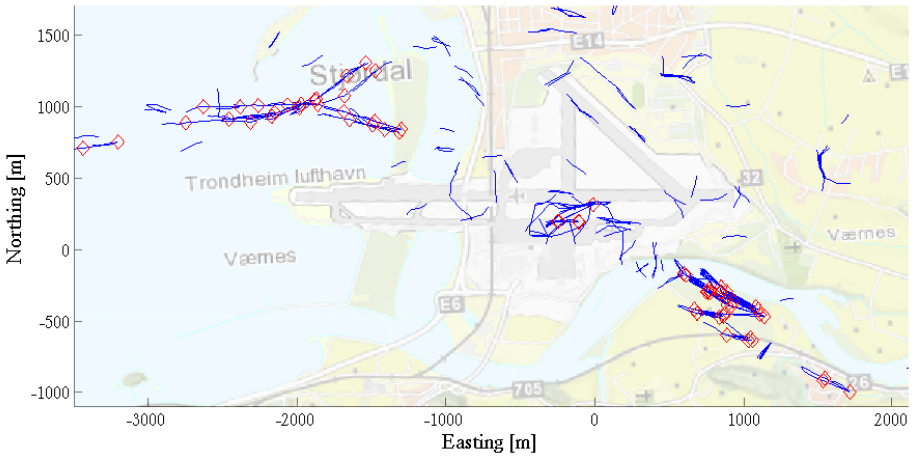


Figure 5.1: The confirmed tracks (blue) and batch associations (red) from incoming birds in Case 1. A flock of birds is inbound from the West over sea, splits into two flocks, disappears over airport and reemerges in the South-East flying away from the airport. CACM-CFAR, $K_t = 6$.

5.1 The BFPD Tracker output

This section will show the output of the BFPD Tracker in the form of confirmed tracks and batch associations.

As may be remembered from Chapter 4, a confirmed track consists of a set of plots, which all consist of information of the plot's position (x,y), Time of Observation (TO), Doppler shift, size and its originating range-azimuth-cell (r,az) and scan i (see Table 4.1). A batch association consists of two uniquely identified confirmed tracks (T_1,T_2) originating from two consecutive batches (B_1,B_2).

All this data is what a subsequent early warning function will base its assessments on when deciding which situations should trigger a warning. However, for convenient presentation here, only the spatial information of both tracks and batch associations are shown.

Figure 5.1 shows all confirmed tracks (blue lines) and batch associations (red diamonds) from the incoming birds (presumably seagulls) in Case 1 (24 batches, 168 scans total), with the underlying MTD processor running CACM-CFAR with $K_t = 6$. The birds are tracked rather consistently when flying over sea, emerging with consistent tracks around 3 km over sea. This is fairly consistent with the calculations of Section 3.4.1, where it is calculated that a Swerling 1 case target with the RCS of a single seagull has a P_d around 0.7-0.8 at about 2.5-3 km in free-space for a $P_{fa} = 10^{-3}$. While the true P_{fa} in Figure 5.1 is unknown, Equation 2.11 gives P_{fa} for a CA- and not a CACM-CFAR, which is assumably the closest available theoretical value for P_{fa}. For the settings of Figure 5.1, this equation yields $P_{fa} = 6.1 \times 10^{-3}$.

The tracks are lost when they reach the shore. Fairly consistent detection of at least one bird is maintained inland by the MTD processor. Due to the BFPD Tracker's designed strictness for tracks confirmation, there are many instances where tracks are deleted even

though the MTD has fairly reliable detection of a target. This is easily observable when comparing the MTD output to the tracker output, and while most common over land, it also happens over sea. However, for many scans while the birds are over the airport, the MTD also fails to detect the birds with the current settings.

The birds are shortly detected and tracked over the airport, disappear again, and then reappear with consistent detection and tracking over the river to the South-East.

There is a distinct observable difference in both detection and tracking results over sea compared to that over land, with sea clutter being the easiest background. This is probably due to land clutter being much more powerful than sea clutter (at low grazing angles). At the same time, many propagation effects such as shadowing, mirroring and multipath propagation are probably at play in addition to the clutter. These effects may very well degrade detection and tracking performance over land. Also, after the larger flock splits up into two smaller, the smaller flocks are harder to detect and track, which also might explain why the tracks are lost shortly after the splitting.

A number of short confirmed tracks that lacks the red diamond markings of the batch associations are also present. Most of which do not follow any obvious direction or pattern. Although many of these probably originate from birds, and are thus *true* confirmed tracks, some are undoubtedly false. There are many phenomena that may cause false confirmed tracks (i.e. cars, a line of consistent false alarms) and it is extremely hard to eliminate all. Neither is it preferable to do so (at least in this application), because if all false confirmed tracks are eliminated, there are probably also more true confirmed tracks that are missed.

The results from Case 2, consisting of a large number of unknown, small birds over land, are conclude to give little insight that cannot be obtained from Cases 1 and 3, and is therefore omitted.

Figure 5.2 shows an image equivalent to that of Figure 5.1, but for the large flock of geese that is inbound from the North in Case 3. Comparing to Figure 5.1, it is immediately evident that these targets are more reliably tracked (and therefore also detected). From this, it may be concluded that the targets most likely consist of birds with larger RCS or in greater number than those in Case 1. This was also confirmed visually by the operators on duty in the Værnes ATC Tower during the recording of the Case 3 data.

The flock of geese are reliably detected and tracked from about 4 km both over land and sea. According to the calculations of Section 3.4.1, a single goose has a P_d around 0.7-0.8 at 3-3.5 km in free-space with a $P_{fa} = 10^{-3}$. It is not unreasonable that the larger RCS resulting from a flock allows for a considerable increase in range compared to this, even with the presence of clutter and propagation effects. A $P_{fa} > 10^{-3}$, which likely is the case (referring to Equation 2.11), means that the true P_d is higher than what the calculations of Section 3.4.1 would suggest. The expected maximum range for consistent detection and tracking of geese is hence also longer.

The flock splits up right North of the runway, West of the sub-passage of the highway E6, into one flock heading North-West and one heading South-West. For both of the new flocks, there are a couple of batches towards the end of the Case 3 data where they do not form any confirmed tracks. The flock heading North-West seems to be somewhat dispersing, which may cause less consistent tracking, and the flock heading South-West

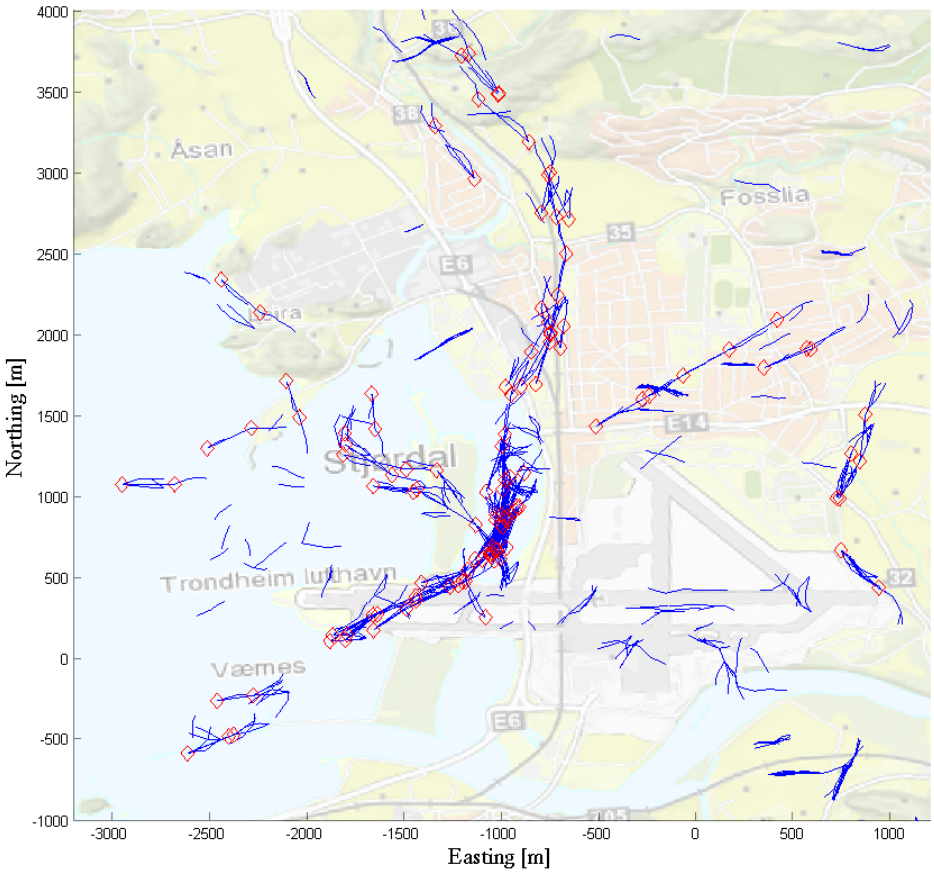


Figure 5.2: The confirmed tracks (blue) and batch associations (red) from an incoming flock of geese in Case 3. A large flock of geese are inbound from the North over land, turns slightly and flies along the coastline/E6 heading towards the runway. The flock splits into two flocks, one sharply turning North-West somewhat dispersing, while the other turns South-West and crosses the runway. Other birds are also present. CACM-CFAR, $K_t = 6$.

seems to be victim of a shadowing effect caused by the airport terminal and hotel buildings.

Simultaneously, to the North and North-East of the airport, over the town Stjørdal, there are definitely other birds present. These are unidentified birds flying with a South-West course, but are probably seagulls or birds of similar size judged by their detection and tracking consistency. Maybe also other geese.

Figure 5.3 shows a close-up of where the flock splits up, during the time-lapse of six consecutive batches (of 7 scans each). This figure is primarily meant to give the reader a sense of the time-dimension that is ignored in the other images of the radar video. Secondly, it illustrates the BFPD Tracker's ability to track closely separated targets and follow them as they intersect and diverge. Due to the design of the tracker and the visual nature of the output data, the merging and splitting of tracks is not of particular concern. At the same time, this figure shows how well the tracker naturally performs such actions for every batch when the number of plots is limited.

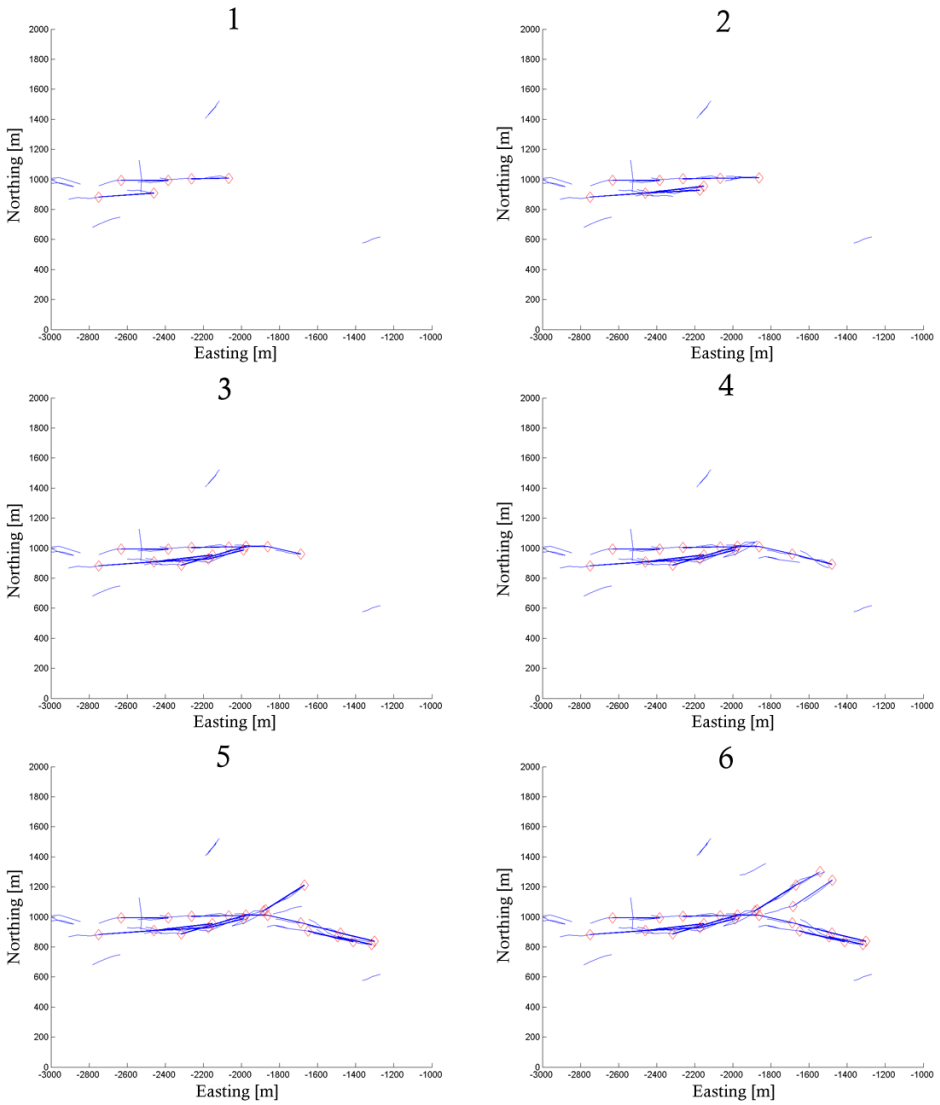


Figure 5.3: The confirmed tracks (blue) and batch associations (red) from the incoming birds in Case 1 for six consecutive batches (of 7 scans each) around the time when the flock splits up into two. The birds are incoming from the West, and these illustrations numbered 1 (top left) through 6 (bottom right) show how the status quo is updated for each batch as the birds fly East and split up. Here shown running the tracker on CACM-CFAR processed data with $K_t = 6$.

5.2 Tracker performance with CFAR variations

This section will account for the tracker's performance when input MTD-data that is processed with different CFAR types and parameters. For the entire system, the MTD processor and the tracker cannot be evaluated completely separated. The total working cascade is important for the end result, and the following results may give some insight to how the establishment of tracks is influenced by the settings of the MTD processor.

Table 5.1 shows the number of extracted plots, tentative and confirmed tracks established by the BFPD Tracker with DMHC and DMUA enabled for different CFAR types and settings in the MTD processor. CACM- and OSCM-CFAR is found to greatly outperform the classic CA- and OS-CFAR, so while CA with $K_t = 6$ is included to illustrate the effect of the Clutter Map (CM), the OS-variations are omitted. For details on how the Knowledge Base (KB) operations DMHC and DMUA affect the outcome, see Section 5.3.

When reading Table 5.1, it is important to keep in mind that fewer extracted plots and resulting tracks is not necessarily better (since true tracks may be lost), though it generally means shorter processing times and that a higher percentage of the confirmed tracks are true (originating from birds).

Table 5.1: Total number of extracted plots, tentative and confirmed tracks established by the BFPD Tracker (with DMHC and DMUA enabled) for different CFAR types and settings (on Case 1 data).

CFAR	K_t	Extracted Plots	Tentative Tracks	Confirmed Tracks
CA	6	113447	1668	909
CACM	5	140324	4666	1292
	6	69021	745	368
	8	24517	190	153
OSCM	4	84036	1160	443
	6	19724	185	155

Table 5.1 shows that for CACM-CFAR, threshold multipliers K_t of 8 and 6 give a reasonable 153 and 368 confirmed tracks from the 24 batches of Case 1 in total. This number counts every unique track in every separate batch as one track, so the number is quite large even though there may not be several hundred birds present. However, when using $K_t = 5$, the tracker is flooded with tracking possibilities, establishing 4666 tentative tracks, which is more than 6 times as many tentative tracks to follow and evaluate than for $K_t = 6$. Therefore, $K_t = 5$ also results in greatly increased processing times compared to $K_t = 6$. The increase in processing time from $K_t = 6$ to 5 is far greater than the increase from $K_t = 8$ to 6.

From all this it is indicated that with a CACM-CFAR in the MTD processor, the K_t that does not flood the tracker with false tracks lies somewhere around 5.5-6.5 depending on how many false confirmed tracks, and how long processing times are acceptable. For OSCM-CFAR in the MTD processor, the tracker seems to be flooded with tracks at values of K_t smaller than $4 - 4.5$. It is observed that, regardless of the CFAR type, a higher K_t may be employed to ensure a higher reliability (probability of trueness) of both reported tracks and warnings. This also decreases the probability that a given bird will form a con-

firmed track and/or trigger an Early Warning (EW).

Due to the different nature of OSCM- and CACM-CFAR, the two will result in an approximately equal number of plots/tracks for different values of K_t when using an equal window size $N = 16$. For instance, OSCM with $K_t = 6$ yields 185 tentative and 155 confirmed tracks, which is very comparable to the 190 tentative and 153 confirmed tracks when CACM with $K_t = 8$ is used.

Equation 2.11 states that CA-CFAR (N=16) with $K_t = 6$ and 8 yields a $P_{fa} \approx 6.1 \times 10^{-3}$ and 1.5×10^{-3} respectively. Comparatively, Equation 2.14 states that OS-CFAR (N=16, k=12) with $K_t = 4$ and 6 yields a $P_{fa} \approx 14.4 \times 10^{-3}$ and 2.8×10^{-3}. These values for P_{fa} do not directly apply to the CACM- and OSCM-CFAR, but their relationship seems to be relatively unchanged when employing the hybrid CFARs expanded with a Clutter Map (CM).

In this section, a distinction is made between unmerged and merged tracks, as well as between tentative and confirmed tracks. As explained in Chapter 4, the tracker is designed to pursue tracking on different combinations of the same plots by allowing the association of one plot to multiple tracks. This often results in many separate tentative and confirmed tracks that lie very closely separated, or even mostly on top of each other, since they may share many plots. The number of tracks given in Table 5.1 is describing such unmerged tracks.

Hence, these closely separated tracks need to be merged for the number of tracks to accurately represent the number of targets present. This is currently done by plotting all tracks, manually analyzing the resulting radar video and counting all tracks. When coupled with interpretation of the MTD output, this is also the best way to interpret whether a confirmed track is true (originating from present birds) or false (originating from something other than birds).

To evaluate the performance of the tracker as a function of CFAR parameters, the portion of true and false merged confirmed tracks and merged batch associations needs to be quantified. This will give critical information of the reliability of the tracker output for different CFAR settings. Tables 5.2 and 5.3 show such results from a manual counting. Deciding whether a track or batch association is caused by the real presence of birds (and thus is true) is an inherently uncertain process since the ground-true of all bird activity is unknown. These decisions are however made highly reliable by close manual inspection of the MTD output (by the author) and by calling all cases of reasonable doubt as false incidents.

Table 5.2 gives an overview of the merged confirmed tracks for every batch of the Case 1 data when varying the CFAR parameters. This table shows that for CACM with $K_t = 6$ and OSCM with $K_t = 4$, roughly half of the merged confirmed tracks are true. With a raised MTD detection threshold, for CACM with $K_t = 8$ and OSCM with $K_t = 6$, the percentage of true merged confirmed tracks is raised to between $70 - 80$ %. While the higher detection threshold makes the merged confirmed tracks about 25 percent points more reliable, it also reduces the total number of merged confirmed tracks from 210 to 70

Table 5.2: True and false merged confirmed tracks (manually counted) for varying CFAR settings on Case 1 data.

Merged Confirmed Tracks (Case 1)

CFAR	CACM				OSCM			
K_t	6		8		4		6	
Batch	True	Tot	True	Tot	True	Tot	True	Tot
1	4	11	1	2	5	15	2	3
2	6	11	2	2	4	8	1	1
3	2	6	1	2	3	7	0	1
4	0	4	2	3	4	7	1	1
5	6	11	3	4	4	13	2	3
6	5	13	2	2	9	11	3	3
7	5	10	3	3	5	9	3	3
8	3	9	0	0	5	9	1	1
9	5	11	4	5	4	8	3	3
10	6	11	3	4	4	8	4	4
11	4	6	2	4	4	8	1	3
12	5	8	4	6	8	11	4	6
13	1	10	0	1	5	10	0	1
14	4	9	2	2	4	9	1	2
15	2	5	1	1	5	7	2	3
16	3	5	2	2	7	14	1	1
17	3	11	1	3	6	11	1	2
18	4	8	4	6	9	15	4	5
19	4	8	3	3	6	12	2	2
20	3	8	0	3	4	13	0	0
21	4	9	1	1	4	11	1	2
22	7	9	3	4	7	11	3	3
23	7	9	4	5	6	11	5	5
24	6	8	2	2	5	9	3	3
SUM	99	210	50	70	127	247	48	61
% True	47.1%		71.4%		51.4%		78.7%	
% False	52.9%		28.6%		48.6%		21.3%	

for CACM, and from 247 to 61 for OSCM.

Table 5.3 gives the equivalent of Table 5.2 for batch associations in the Case 1 data. This table illustrates that the batch associations have a very high reliability. In the total of 24 batches for all four CFAR modes that were investigated, all but one batch association were concluded to be true. This single false batch association was a case of doubt, and may in reality also be caused by birds, but was classified as false since the evidence of true bird presence was not strongly convincing. Because almost all of the batch associations were true for all four CFAR modes, a batch association is a strong and highly reliable indication of true bird presence in an established Bird Flight Path (BFP).

Table 5.3: True and false merged batch associations (manually counted) for varying CFAR settings on Case 1 data.

Merged Batch Associations (Case 1)

CFAR	CACM				OSCM			
K_t	6		8		4		6	
Batch	True	Tot	True	Tot	True	Tot	True	Tot
1	0	0	0	0	0	0	0	0
2	1	1	1	1	2	2	1	1
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	1	1	0	0	0	0
6	3	3	0	0	1	1	0	0
7	2	2	1	1	3	3	1	1
8	3	3	0	0	2	2	1	1
9	1	1	0	0	1	1	1	1
10	3	3	2	2	3	3	2	2
11	2	2	0	0	1	1	1	1
12	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0
16	0	0	1	1	2	2	1	1
17	0	0	0	0	0	0	0	0
18	1	2	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0
20	1	1	0	0	1	1	0	0
21	0	0	0	0	0	0	0	0
22	1	1	0	0	1	1	0	0
23	3	3	2	2	3	3	3	3
24	3	3	3	3	3	3	3	3
SUM	24	25	11	11	23	23	14	14
% True	96%		100%		100%		100%	
% False	4%		0%		0%		0%	

Table 5.4 and 5.5 show the total number of true and false confirmed tracks and batch associations on Case 3 data. These are equivalent to that of Table 5.2 and 5.3 for Case 1 and are comparable in principle. Making this comparison, it is evident that higher values of K_t are preferable when a large flock of large birds like geese is present. CACM with $K_t = 6$ and OSCM with $K_t = 4$ perform worse in Case 3 than in Case 1. CACM with $K_t = 8$ and OSCM with $K_t = 6$ maintain a true merged confirmed track rate of 70 to 80 % and a true merged batch association rate of 100 %. The main reason for the difference in results is probably the different nature of the two cases. Case 1 contains data on a medium

Table 5.4: True and false merged confirmed tracks (manually counted) for varying CFAR settings on Case 3 data.

Merged Confirmed Tracks (Case 3)

CFAR	CACM				OSCM			
K_t	6		8		4		6	
	True	Tot	True	Tot	True	Tot	True	Tot
SUM	157	420	118	167	152	509	109	139
% True	37.4%		70.7%		29.9%		78.4%	
% False	62.6%		29.3%		70.1%		21.6%	

Table 5.5: True and false merged batch associations (manually counted) for varying CFAR settings on Case 3 data.

Merged Batch Associations (Case 3)

CFAR	CACM				OSCM			
K_t	6		8		4		6	
	True	Tot	True	Tot	True	Tot	True	Tot
SUM	59	60	55	55	50	51	45	45
% True	98.3%		100%		98.0%		100%	
% False	1.7%		0%		2.0%		0%	

sized flock of unknown species (presumably seagulls), while case 3 contains data on large flock of geese in addition to other birds. Tuning of the total system to make reliabe detection and tracking of BFPs from geese (or other birds of large size or numbers) should therefore employ CFAR multipliers on the order of $K_t = 8$ for CACM, and $K_t = 6$ for OSCM.

Figure 5.4 shows the number of unmerged tentative and confirmed tracks for every batch in Case 1 when MTD processing with CACM $K_t = 5$, 6 and 8. It has already been shown that a higher threshold multiplier in the MTD processor yields fewer tracks, with a greater percentage of confirmed tracks relative the total, which is expected. This figure gives another view of this tendency, that also is evident in Table 5.1 where the total number of tracks is shown. Note that all confirmed tracks start out as tentative tracks.

Figure 5.4 along with Table 5.2 show that when the CFAR threshold multiplier is increased, the number of total tracks and the portion of tentative tracks that are deleted drop drastically while the reliability (probability of trueness) of the remaining confirmed tracks increases.

One example of the more interesting details revealed by Figure 5.4, emerges from comparing batch number 18 and 23 through the different CFAR multipliers. For the lowest $K_t = 5$ (top), very many tentative tracks are established through all batches. Batch number 18 and 23 hold the top two places in both the number of tentative and confirmed tracks. Here, batch number 18 has the most confirmed tracks, which suggests that many true

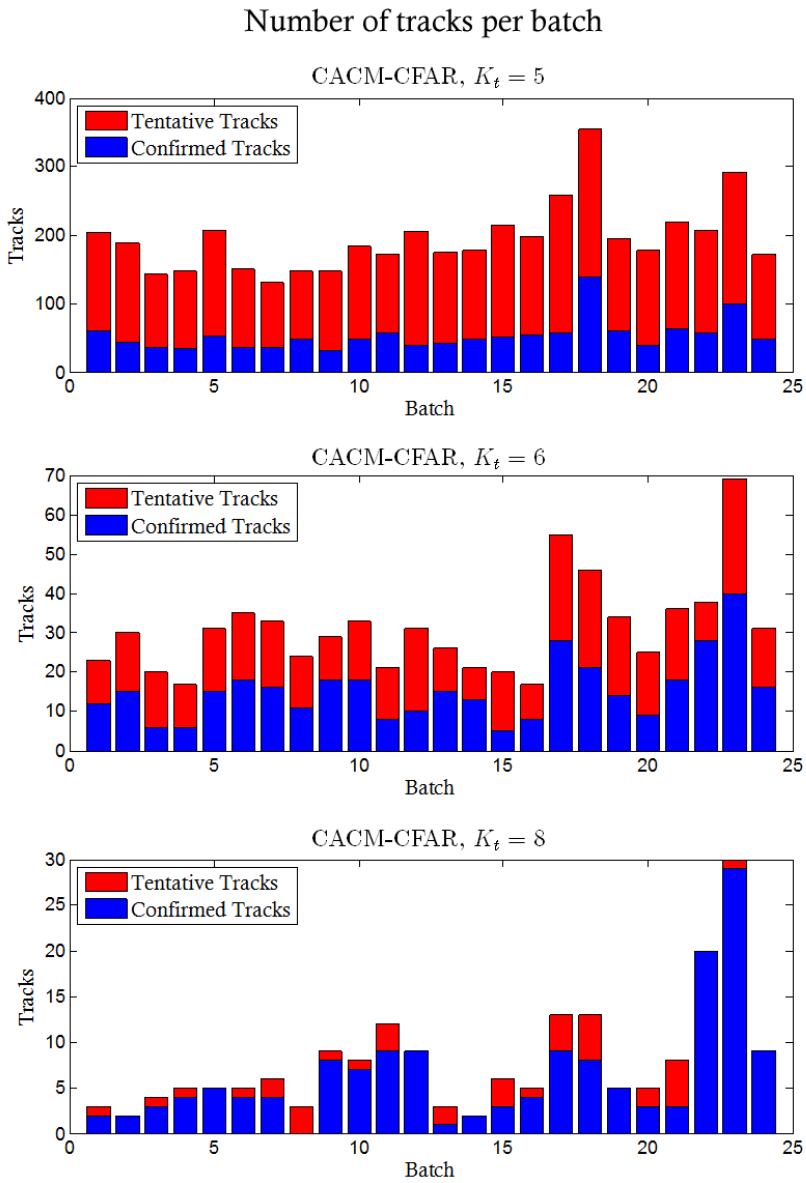


Figure 5.4: Number of unmerged tentative and confirmed tracks per batch (on the Case 1 data) found by the tracker when input MTD-processed data using CACM-CFAR with threshold multipliers $K_t = 5$ (top), 6 (middle), and 8 (bottom).

targets were visible and tracked in this batch. However, this picture changes when the multiplier K_t is increased. At the highest $K_t = 8$ (bottom), when the tracker is only fed with the most reliable plots, batch number 23 towers with the far highest number of

confirmed tracks while batch number 18 is hardly distinguished. Since the tracks are more reliable when based on only the most reliable plots, this suggests that there really were not remarkably many true targets visible and tracked in batch 18. Rather, batch number 23 seems to be the best in this regard, which is also seen in Figure 5.1 as batch 23 is when the birds have emerged at the far side of the river South-East of the Airport.

5.3 Knowledge Base operations

This section will quantify the effect imposed on the number of plots and established tracks by the Knowledge Base (KB) operations. The KB operations consist of the two methods called Delete Measurements in High Clutter areas (DMHC) and Delete Measurements in Uninteresting Areas (DMUA), whose implementations were defined in Chapter 4.3.

Table 5.6 gives the number of extracted plots and established tentative and confirmed tracks for the BFPD Tracker without DMHC and DMUA, with only DMUA enabled, and with both enabled (default). The numbers are stated as totals over 168 scans on Case 1 data. This is similar to that of Table 5.1, where the numbers for the BFPD Tracker (with DMHC/DMUA enabled and CACM-CFAR,$K_t = 6$) may be recognized under "CACM $K_t = 6$".

Table 5.6: 168 scans total number of extracted plots, tentative tracks and confirmed tracks established by the BFPD Tracker (Case 1).

24 batches total values CACM-CFAR, $K_t = 6$	Extracted Plots		Tentative Tracks		Confirmed Tracks	
Operation mode	Abs.	Rel.	Abs.	Rel.	Abs.	Rel.
No KB Operations	83136	100%	1759	100%	700	100%
DMHC	80441	96.8%	1602	91.1%	647	92.4%
DMUA	71434	85.9%	819	46.6%	392	56%
DMHC&DMUA	69021	83%	745	42.4%	368	52.6%

Table 5.6 shows that the enabling of both DMHC and DMUA reduces the number of plots by 17 %, tentative tracks by 57.6 % and confirmed tracks by 47.4 %. This is a significant reduction of the number of plots. The result is that the number of tracks is even far more reduced than the number of plots, which makes it effective and saves very much unnecessary processing. It is also evident that most of this reduction is due to the DMUA, which alone reduced the number of plots by 14.1 %, tentative tracks by 53.4 % and confirmed tracks by 44 %.

One can also see from the fact that the number of tentative tracks is reduced more than the confirmed tracks that the majority of the plots removed by the KB operations would have resulted in deleted tracks. In fact, even though there may be trackable targets (e.g. birds, cars) located within the uninteresting areas defined by the DMUA, these are by definition not of any interest and are thus correctly dismissed. Contrary to the DMUA, the DMHC may actually result in the loss of true tracks that are of interest, which is why it is designed to only slightly reduce the number of plots. The DMHC only removes the plots that originate from the cells that have the most extreme high clutter power, since these cells are likely to cause false plots.

All in all, the KB operations make a very effective reduction of measurements that most likely were false. A moderate reduction in the number of plots (17 %) cause a great reduction of the tentative tracks (57.6 %), which greatly reduces the computational requirements of the BFPD Tracker.

5.4 Early Warning

This section will present the output of the Early Warning (EW) processing block of the BFPD Tracker (see ”Early warning decision” in Figure 3.4.

Figure 5.5 shows the EW activations in Case 1 as red squares. The rest of the figure is equal to Figure 5.1, showing BFPs with confirmed tracks and batch associations. When the birds are incoming Eastwards over the fjord, they are not headed towards the runway, and therefore do not cause an EW. They do however, as presented in Section 5.1, cause consistent confirmed tracks and batch associations. This means that not only is it very likely that the tracks and batch associations are true, but the birds are also of significant size and/or numbers. Therefore, when one of the tracks indicate that part of the flock turns South and heads towards the runway, an EW signal is triggered.

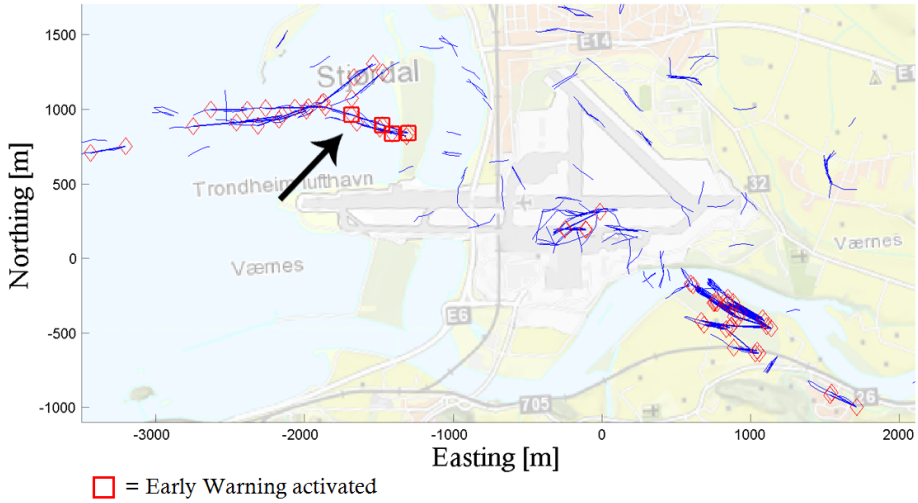


Figure 5.5: BFPD Tracker output on Case 1 data showing confirmed tracks (blue lines), batch associations (red diamonds) and Early Warning (EW) activations (red squares). The black arrow indicates the first activation of the EW, where it is activated because the birds turn towards the runway. CACM-CFAR $K_t = 6$.

Figure 5.6 shows the EW activations in Case 3 in the same manner as Case 1 is shown in Figure 5.5. This figure also exhibits three black arrows that indicate the first activation of the EW by the incoming flock of geese (arrow no. 1) and other birds (arrows no. 2 and 3).

The flock of geese first trigger the EW at a range of about 3.5 km. The geese fly at approximately 1 km per minute (16.7 m/s), which means that this EW gives a warning about 3.5 minutes in advance of the potential birdstrike when the geese reach the aircraft flight corridor at the runway. As a part of the flock of geese breaks off in a sharp turn North-West right North of the runway, these no longer pose an immediate threat as they are flying away from the runway and therefore do no longer activate an EW.

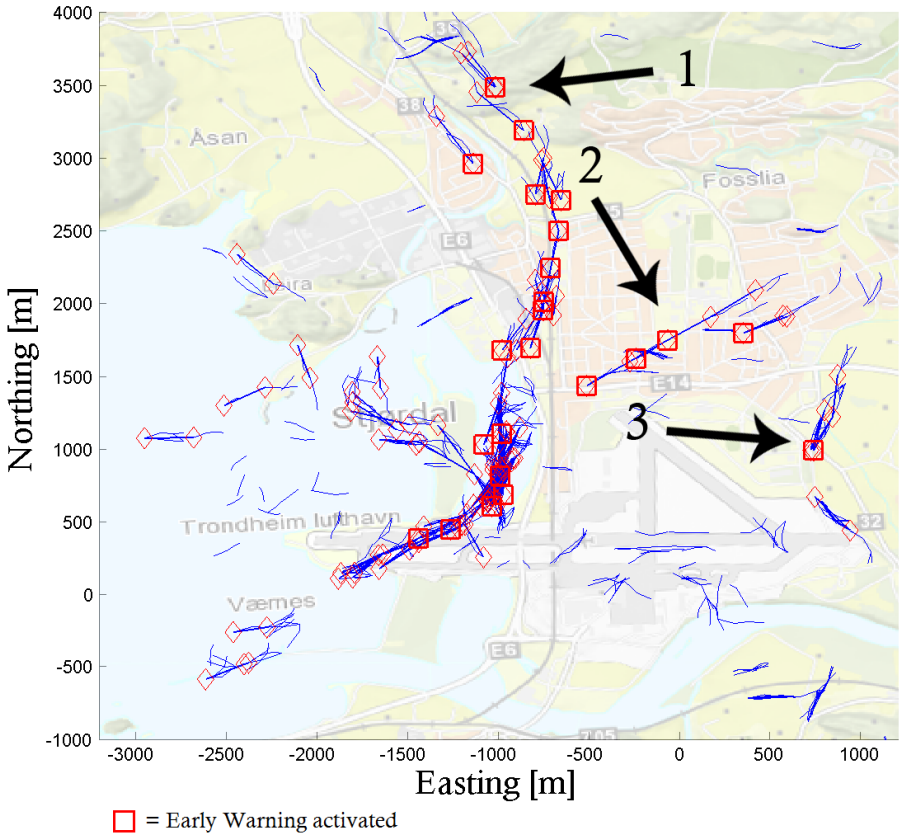


Figure 5.6: BFPD Tracker output on Case 3 data showing confirmed tracks (blue lines), batch associations (red diamnonds) and Early Warning (EW) activations (red squares). Black arrow number 1 indicates the first activation of the EW by the incoming flock of geese. Arrow number 2 and 3 indicate the first EW activations by other birds. CACM-CFAR $K_t = 6$.

Considering the high reliability (> 96 %) of the batch associations that was found in Section 5.2, the reliability of these EW signals should be very high. All but one of the batch associations that were established during the generation of the results were found to be true. Hence, when an EW demands the requirements of Table 4.5, false warnings should become extremely rare. All of the activations of an EW signal that was established by the Case 1 and 3 data are classified true (by manual inspection by the author). However, the data sample size is too small to say much about the false warning rate. The data that the EW was tested on consists of 343 scans (i.e. about 13 minutes) with positively confirmed presence of many different birds. Trial of the BFPD Tracker system on data over longer periods of time must be made to accurately determine the rate of false warning occurence.

5.5 Simulation

This section will give the results from the simulation of a Swerling case 1 type target over real sea clutter and noise. The simulated target was measured to have a mean Signal to Noise and Clutter power Ratio (SNCR) of 15.84 dB in the simulations of this section. The method of simulation is described in Section 3.2.

Figure 5.7 shows the tracked target flight path along with the target's true flight path. Figure 5.8 shows the observations and clearly states in which scans the target was missed. In this simulation run, the target was missed in 7 out of 56 scans. Monte Carlo simulation is required to accurately estimate a probability of detection for the target in sea clutter. However, based on this specific simulation run of 56 scans and assuming a constant probability of detection P_d, the maximum likelihood estimator of P_d for the simulated target is $P_d = \frac{49}{56} = 0.875$. By Equation 2.8, this means that the single-sample probability of false alarm in this case is $P_{fa} \approx 5.2 \times 10^{-3}$ if one ignores that the sea clutter and noise have different pdfs and correlation times (or assumes that detection is limited by noise). Comparatively, Equation 2.11, states that $P_{fa} = 6.1 \times 10^{-3}$ for a CA-CFAR with $N = 16$ and $K_t = 6$. While this is not valid for the CACM-CFAR that is run in the simulation, it may still serve as a reference since it is the closest scenario for which we can easily calculate P_{fa}.

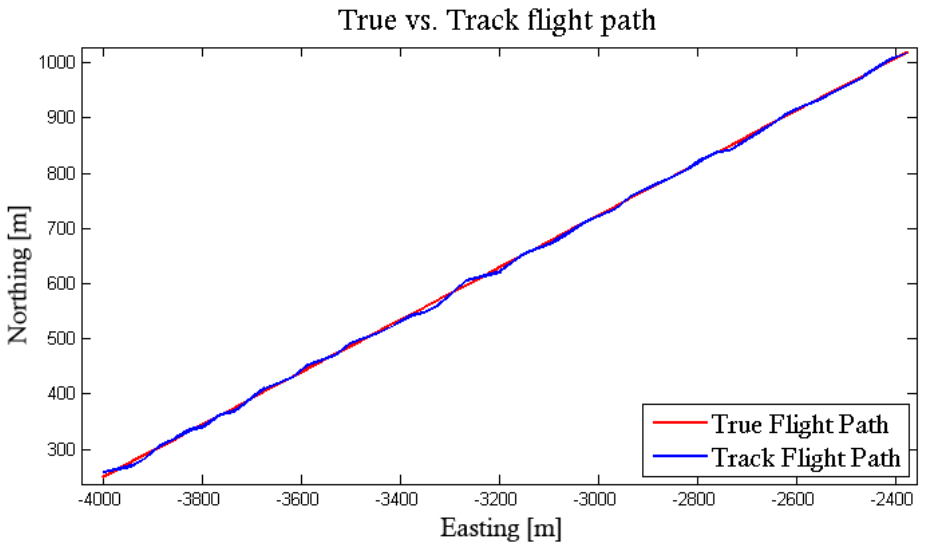


Figure 5.7: The true and the tracked flight path seen from above in the euclidean plane.

Figures 5.9 and 5.10 show the error of the tracked position as a function of observation number (time), divided into the range and azimuth dimension respectively. As might be expected for a radar of this type and a target at 2.5 - 4 km, the position error is greater in the azimuth dimension than in range. In range, the error is within ± 6 meters, while the error in azimuth is within $\pm 0.23°$.

Figure 5.11 shows the error of the geometrically induced absolute velocity (speed) of

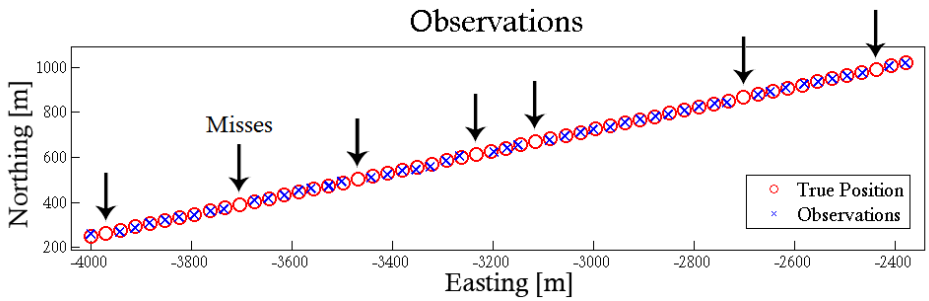


Figure 5.8: The target observations (blue crosses) plotted along with the true target positions at the time of measurement (red circles) in the euclidean plane. Black arrows indicate misses, i.e. where the target was not detected.

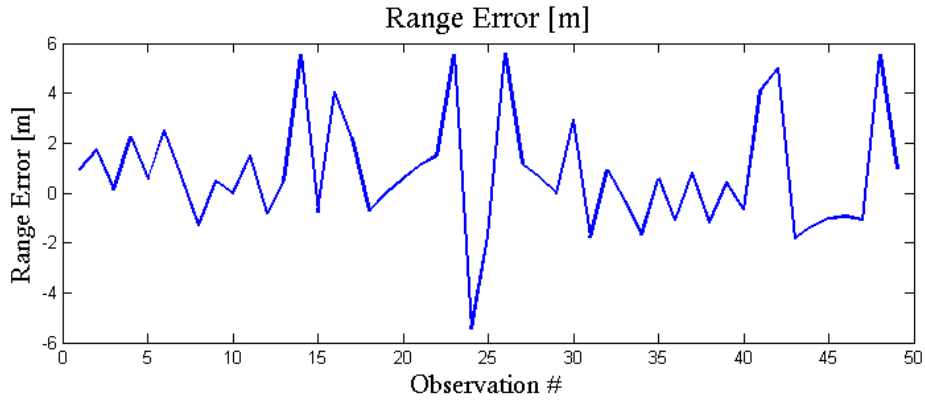


Figure 5.9: The range error of the target observations.

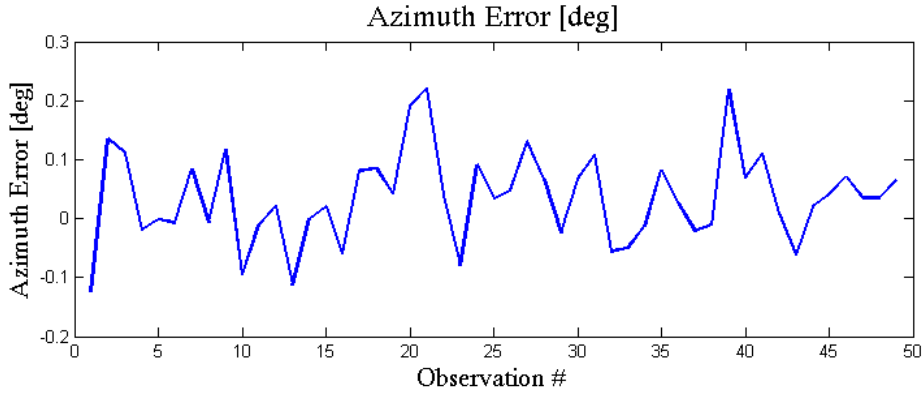


Figure 5.10: The azimuth error of the target observations.

the simulated target as measured by the tracker. This error in geometrically measured velocity, which is the velocity that the BFPD Tracker uses to determine which plots fit the target dynamics model, is found to fluctuate within 3 m/s. If this error becomes too large, it will cause the loss of tracks.

Figure 5.12 gives the geometrically and Doppler-induced radial velocity along with the target's true radial velocity. The difference between the radial velocity measured from geometry and Doppler is shown in Figure 5.13 and is used by the BFPD Tracker to determine the quality of tracks. Therfore, the results of Figure 5.12 and 5.13 will influence the track confirmation process.

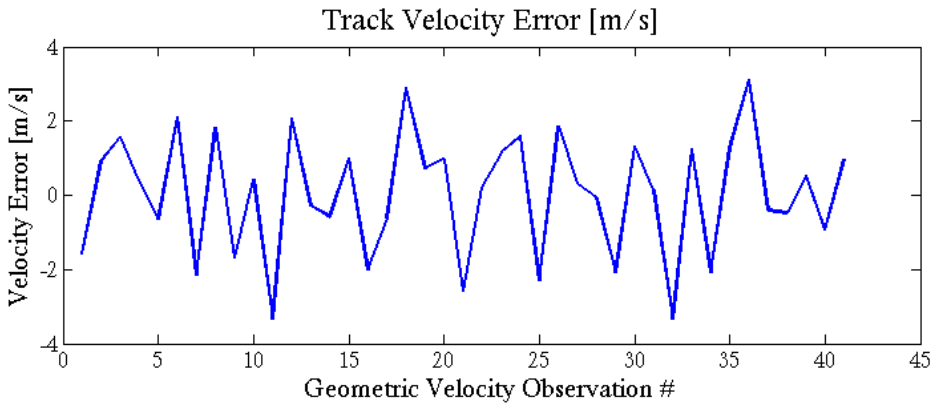


Figure 5.11: The error of the observed target absolute velocity, measured from the geometry and timing of the observations.

Table 5.7: RMS and peak errors in tracking of a simulated target over sea.

Position	**RMS Error**	**Peak**
Range	2.37 m	6 m
Azimuth	0.084° ≈ 5.86 m @ 4km	0.23°
X (Easting)	2.56 m	6 m
Y (Northing)	4.82 m	12 m
Velocity		
Geometric	1.64 m/s	3 m/s
Geometric radial	1.28 m/s	4 m/s
Doppler radial	1.56 m/s	2.5 m/s

Table 5.7 gives a summary of all the error signals for position and velocity by stating their Root Mean Square (RMS) and peak values.

The RMS position error is 2.37 meters in range and 0.084° in azimuth, which is fairly good considering that the radar resolution cell is $22.5\text{m} \times 0.95°$. This error in position translates to an RMS error in the measured velocity of 1.64 m/s (labeled "Geometric" in

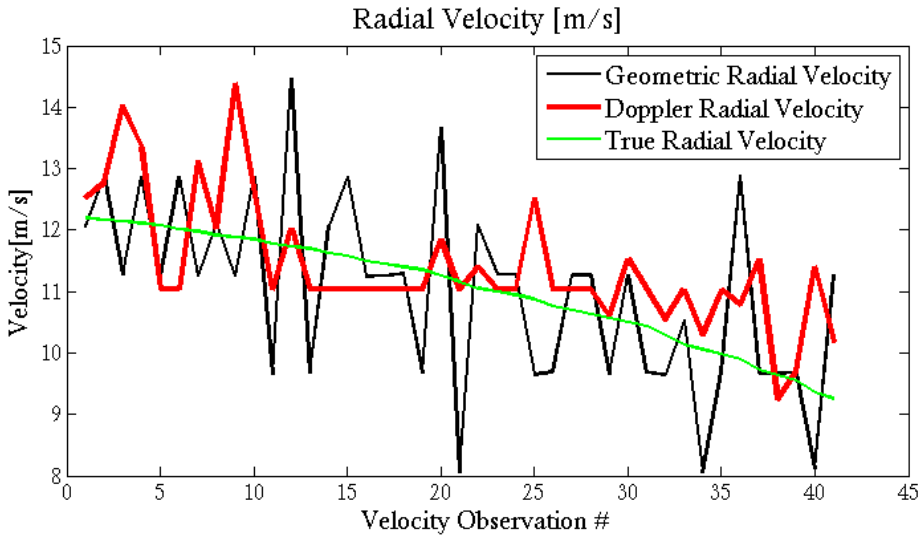


Figure 5.12: Observed target radial velocity measured from geometry (black) and Doppler (red) along with the target's true radial velocity (green).

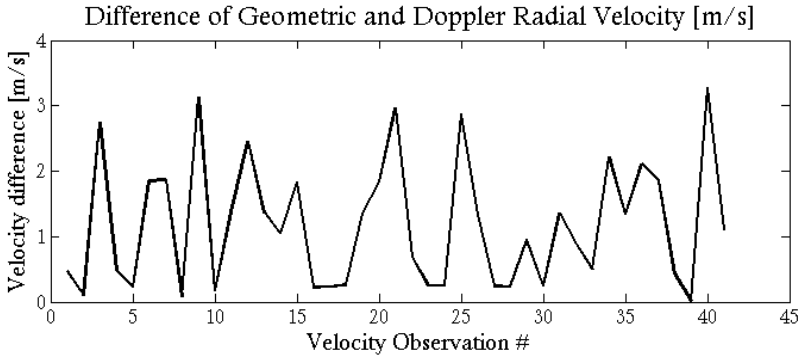


Figure 5.13: The absolute difference of the simulated target's radial velocity as measured from geometry and from Doppler.

Table 5.7), while peak variations of this velocity is within 3 m/s. Compared to the tracker's tolerances of variations in speed – which are 8 m/s from scan to scan, an absolute roof of 23 m/s and floor of 10 m/s, this error is significant, but manageable.

If a target were to accelerate 2.1 m/s^2 = 0.22 G, and thus increase its speed by 5 m/s between two scans at the same time when a measuring error of 3 m/s occurs, the error might cause loss of track. Similarly, during tracking of targets at speeds over 20 or less than 13 m/s, a velocity measurement error of 3 m/s will likely cause loss of track. However, such an occurence might also just cause what looks like a single miss as the position of a target plot may erroneously be interpreted as outside the tracking gate. Problems in tracking due to this measurement error should be insignificant as long as it keeps at the

level of current observations. Though, if such problems should arise, they may be mitigated by adjusting the target dynamics model to allow for larger fluctuations of target speeds.

The radial velocity measured from the geometry of the measurements (black plot in Figure 5.12) and from Doppler information (red plot in Figure 5.12) is used in the track confirmation process. If these two signals are in sufficiently disagreement for a given track, the track will be deleted. Figure 5.13 shows a plot of the difference between these two velocity measurement signals as a function of observation number for the simulated target. Their difference is less than 3 m/s most of the time. The BFPD Tracker is set to delete all tracks that have such a difference larger than 5 m/s in at least 3 of the 4 to 7 measurements of each track within a batch, as can be seen in Table 4.3.

5.6 Batch length variations

So far, the batch length of the BFPD Tracker has been set at $N = 7$ scans per batch. Also until now, the M-out-of-N criterion utilized in the track confirmation process has been 4-out-of-7. This section will present some results when varying these parameters, N and M, while leaving all other aspects and parameters of the BFPD Tracker unchanged. This also means that the BFPD Tracker's property of always denying two consecutive misses in a track is maintained.

One motivation for doing so, is to decrease the track establishment time. If $N = 7$, the absolute earliest time from when a target becomes consistently detected to when a track for that target can be established is $(N-1)T = 6 \times 2.3 = 13.8$ seconds. This is the time it takes to just gather the measurements necessary for making the batch calculations. If an EW demands three confirmed tracks in three consecutive batches, the earliest time from beginning of consistent target detection to EW activation is $3(N-1)T = 41.4$ s. While these numbers work well for an incoming flock of geese, which is consistently detected at around 4 km (approximately 4 minutes fly-time from runway), it is a too long delay for quickly emerging threats at closer ranges. Therefore, primarily shorter alternatives for batch length and track confirmation criteria are tested and presented in this section.

Please note that as the batch length is changed, the previously presented results on the reliability of confirmed tracks, batch associations and EWs are no longer valid.

The following Figures 5.14-5.17 show superimposed confirmed tracks (blue lines) and batch associations (red diamonds) from ca. 170 scans (varying number of batches), resulting from shorter batch length and different track confirmation criteria. For the first two, $K_t = 6$ is used and EWs are shown as red squares. In the last two, the track confirmation criteria are less restrictive and result in excessive tracks, batch associations and EWs. Therefore, the CFAR multiplier is increased to $K_t = 8$ and the EWs are omitted altogether in these pictures to limit the amount of plotted information.

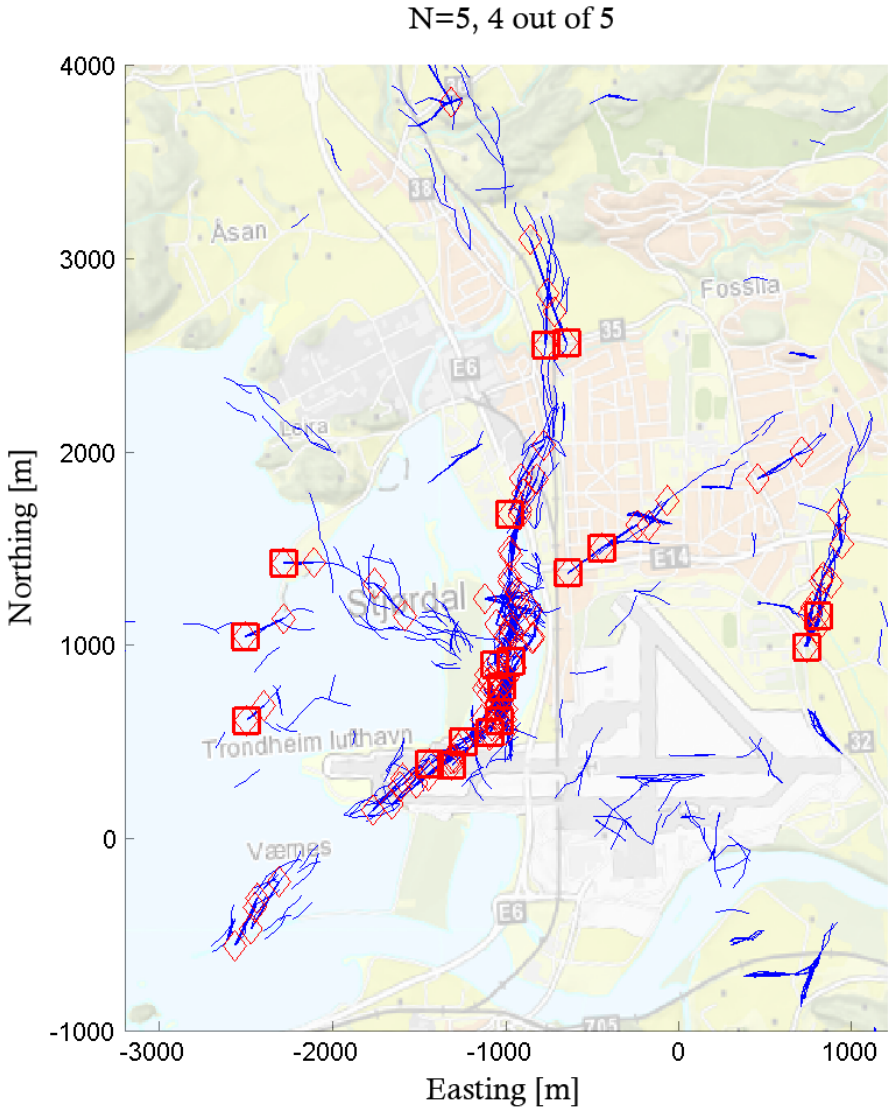


Figure 5.14: Confirmed tracks (blue lines), batch associations (red diamonds) and Early Warnings (red squares) for a batch length of $N = 5$ scans with a track confirmation criterion of 4-out-of-5. CACM-CFAR, $K_t = 6$

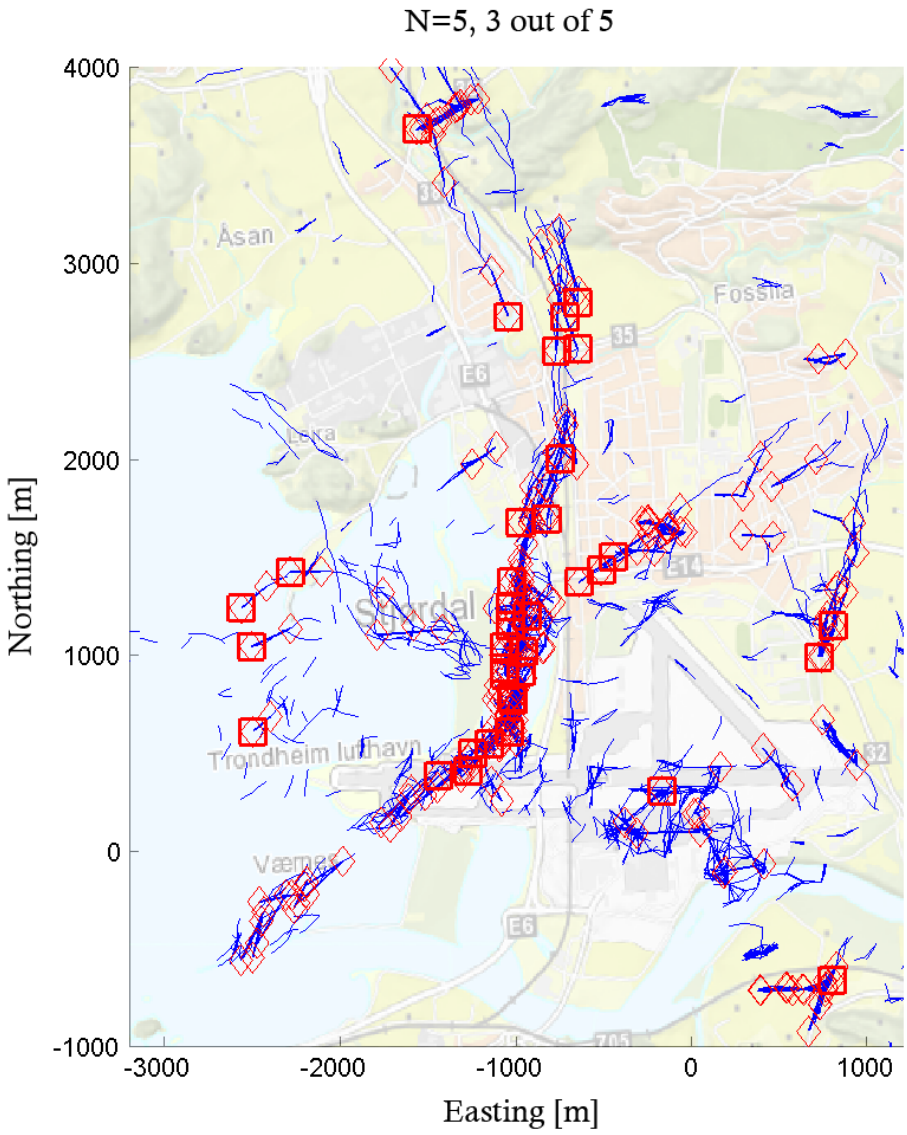


Figure 5.15: Confirmed tracks (blue lines), batch associations (red diamonds) and Early Warnings (red squares) for a batch length of $N = 5$ scans with a track confirmation criterion of 3-out-of-5. CACM-CFAR, $K_t = 6$

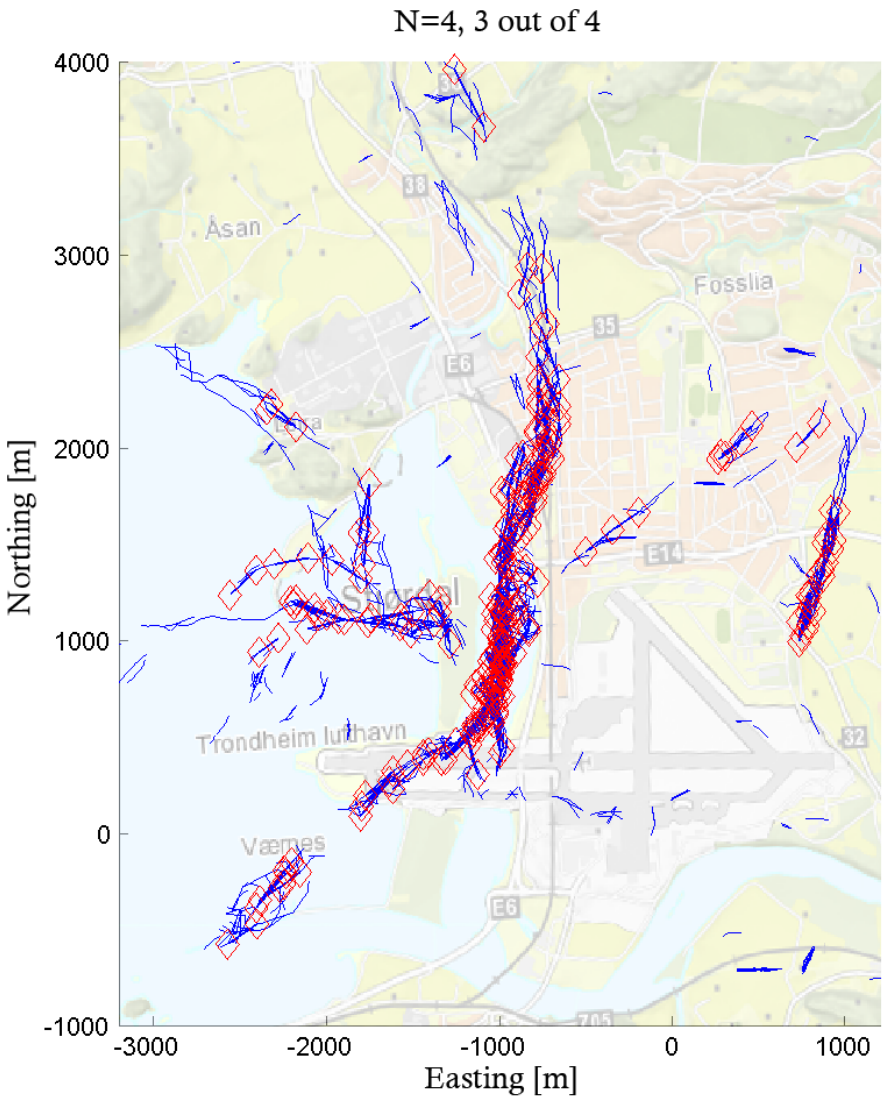


Figure 5.16: Confirmed tracks (blue lines) and batch associations (red diamonds) for a batch length of $N = 4$ scans with a track confirmation criterion of 3-out-of-4. CACM-CFAR, $K_t = 8$

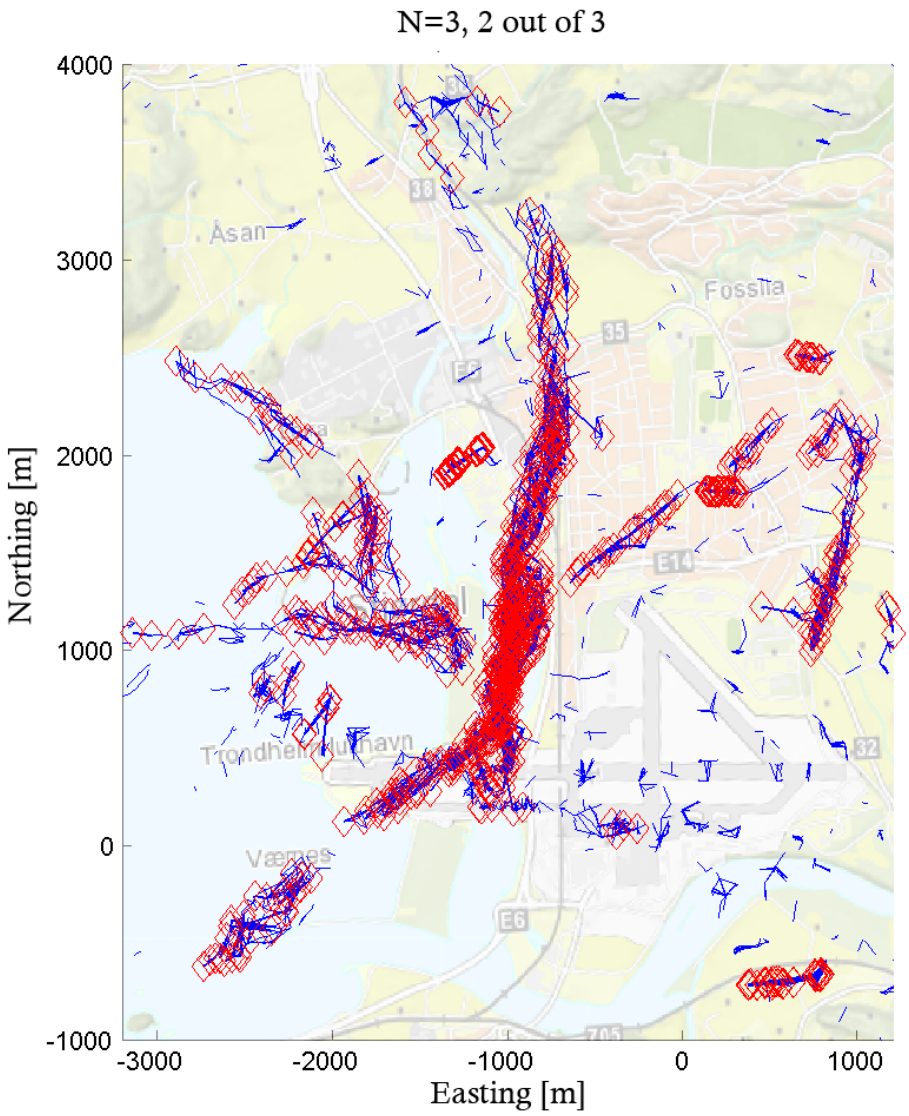


Figure 5.17: Confirmed tracks (blue lines) and batch associations (red diamonds) for a batch length of $N = 4$ scans with a track confirmation criterion of 2-out-of-3. CACM-CFAR, $K_t = 8$

Notice that the increase of K_t from 6 to 8 in Figures 5.16 and 5.17 significantly reduces the amount of inland false tracks. Also, it is obvious that less restrictive demands for track confirmation (i.e. 2-of-3 vs. 3-of-4 and 3-of-5 vs. 4-of-5) significantly increases the number of total tracks. This in turn also increases the number of batch associations and EWs if all other parameters are left unchanged. For $N = 3$, 2-out-of-3 and $N = 4$, 3-out-of-4, the EW activations (omitted from figures) become unreliable. While it is evident that these settings (short batches) give very good tracking capability, even of maneuvering targets, the EW activation criteria must be made more restrictive to maintain a high level of reliability of the EW.

When decreasing the batch length N, it seems from the results of Figures 5.14-5.17 that one gets more reliable tracking of birds. This effect is well visible to the left of these figures, where the flock of geese bound North-West disperses. Here, the BFPD Tracker loses the targets for a batch length of $N = 7$ (See Figure 5.2). With a batch length of $N = 5$ (Figures 5.14 and 5.15), the details, consistency and coherence of the tracks for these geese are noticably increased. This is further improved with $N = 4$ and $N = 3$ as shown by the South-West bound birds flying over the town Stjørdal.

The shortest batch length of $N = 3$ does however introduce a considerable amount of false confirmed tracks, so a batch length of $N = 4$ with a preceding CACM-CFAR with $K_t = 8$ in the MTD processor might seem to perform the best of the four. These settings result in remarkably low numbers of false tracks (especially inland) while giving a significant improvement in tracking capability compared to a batch length of $N = 7$. To make the batch associations and EWs as reliable as was shown for a 4-out-of-7 criterion and $N = 7$, the batch establishment and EW activation requirements with a 3-out-of-4 criterion must be made more restrictive.

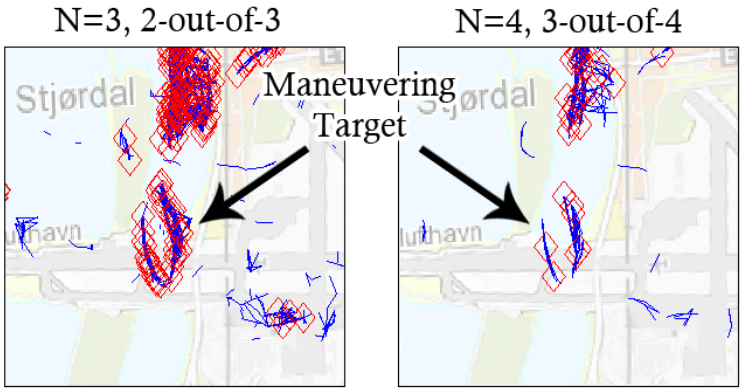


Figure 5.18: Comparison of tracking of a highly maneuvering target for batch length $N = 3$ with a 2-out-of-3 track confirmation criteria (left) versus $N = 4$ with 3-out-of-4 (right). CACM-CFAR, $K_t = 8$.

Figure 5.18 shows the tracking of a highly maneuvering target for batch length $N = 3$ with a 2-out-of-3 track confirmation criteria compared to that of $N = 4$ with 3-out-of-4.

This target is a bird, or a flock of birds, that flies South-bound over the runway and makes a sharp counter-clockwise turn before flying back up North. The shorter batch length of $N = 3$ with a 2-out-of-3-criterion gives a considerable increase in the ability to track maneuvering targets. For larger batch lengths and more restrictive track confirmation criteria, this target is not tracked during its turn over the runway area. Note that the target dynamics model is unchanged, and that this difference in performance is solely due to the difference of batch lengths and M-out-of-N-criteria.

Table 5.8 shows the resulting total number of tentative tracks, confirmed tracks and batch associations for the four processing-cases given in Figures 5.14-5.17. Again, notice that the CFAR threshold multiplier is not kept constant through all four settings, but is increased from 6 to 8 for the shorter batch lengths of $N = 3$ and 4. Even so, the shortest batch length of $N = 3$ still yields a massive increase in the number of tracks. Thus it is illustrated that the need for a track merging function arises and gets increasingly important for shorter batch lengths. For the original setting of $N = 7$ and a 4-out-of-7-criterion, the number of tentative tracks were kept managably low (less than about 1200 tentative tracks) and the resulting batch associations were found very reliable.

Table 5.8: Total number of tentative tracks, confirmed tracks and batch associations for tracking on Case 3 data with different settings of batch length N and M-out-of-N-criteria.

Batch length	M-of-N	K_t	Tentative Tracks	Confirmed Tracks	Batch Associations
5	4-of-5	6	2467	998	149
5	3-of-5	6	6106	3433	262
4	3-of-4	8	1983	1565	419
3	2-of-3	8	10258	9800	1438

For shorter batch lengths there are several advantages such as higher tracking capability of maneuvering targets and shorter delay times, but a track merging function seems necessary. This function should merge tentative tracks that lie closer to each other than some minimum track distance. Such a function should not be difficult to implement, but could however increase the computational demands of the tracker.

Figure 5.19 shows an interesting case where first, a highly maneuvering bird target of unknown species and number flies in over the runway and turns back North. Then, the large flock of geese inbound from the North intersect the runway, right before an aircraft (commercial airliner) comes taxing Westwards, turns and takes off to the East on the main runway. All this happened within a 200m diameter area within a time-frame of about 2.5 minutes. The figure was generated by CACM-CFAR MTD processing with $K_t = 8$, a batch length of $N = 4$ and a 3-out-of-4 criterion.

It is clear that the aircraft (arrows 1 and 2) is not tracked very well, which is a good thing, though it does result in some confirmed tracks while it is on-ground taxing. This is presumably because the aircraft does not fit the target dynamics model very well, but is pretty close during taxing. It is also observed in Figure 5.19 that the plot extractor splits

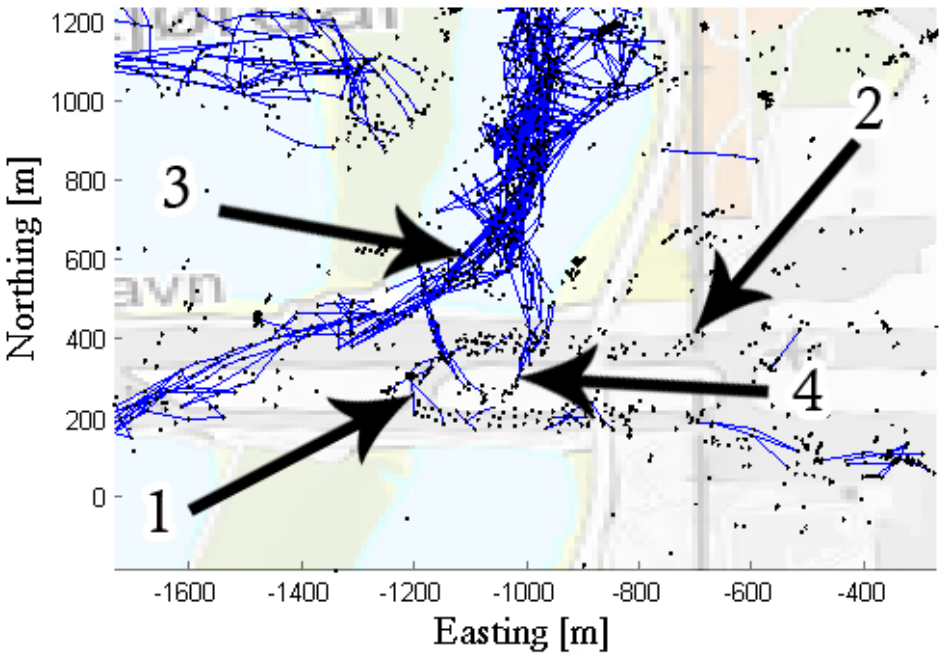


Figure 5.19: A close-up image of time overlay radar video from Case 3 showing plots (black dots), confirmed tracks (blue lines) and black arrows that point out an aircraft during taxing (1) and take-off (2), the passing flock of geese and one or more highly maneuvering birds (4). The aircraft and birds all passed within a 200 m diameter area within about 2.5 minutes. CACM-CFAR $K_t = 8$, batch lengh $N = 4$ and a 3-out-of-4 track confirmation criterion.

the large aircraft into many plots because it assumes all detections are from birds (which are small). This image also shows there is little chance of aircraft being misinterpreted as birds by the tracker at other times than during taxing.

Both the geese and the maneuvering unknown birds are tracked with fair reliability as previously shown. This close-up image does however reveal smaller details and allows closer examination of the tracks since the batch association indicators (red diamonds) that scatter the previously shown tracks are not shown here.

Chapter 6

Discussion

This chapter will give an overall discussion of the BFPD Tracker, the results and findings from the work in this thesis. Discussion and commentary regarding details of the results were given alongside the results in Chapter 5 and will not be repeated here.

6.1 Achievements of this work

First, some discussion of findings regarding the structure of the tracker is given. Thereafter follows a discussion of the performance and tuning of the BFPD Tracker. And lastly, a discussion of the simulation results follows before the main findings are summarized and the research questions from Section 1.4 are attempted answered.

6.1.1 Structure of the tracker

The structure of the Bird Flight Path Detector (BFPD) Tracker was outlined in Chapter 4. This section will discuss some general issues regarding this structure.

The BFPD Tracker does currently not employ a track merging function. The purpose of such a function would be to merge multiple confirmed tracks that are closer than some limit into one single track. This was intentional, to simplify programming and keep all tracks precisely as measured at the same time. If the Early Warning (EW) function is simple and demands little computational power, the lack of a merging function presumably also keeps the overall computational requirement low. On the other hand, the extra calculations of a merging function may be more than compensated for due to the smaller resulting set of tracks and batch associations to run the EW function (and other further processing) on.

A merging function also may make the number of tracks accurately indicate the number of areas where Bird Flight Paths (BFPs) are found and tracked. Employment of a track merging function should definitely be investigated in further development and application of the proposed BFPD tracker into a prototype system.

Generally speaking, the specific implementation of the proposed tracker in MATLAB is ineffective. Many small and large implementation issues, and the fact that MATLAB is an interpreted programming language contribute to large processing times. One of the more significant examples of such is that due to FFT-processing with azimuth decimation in the Moving Target Detector (MTD), the plot extractor (which represents 30-50 % of the current processing time) works on 8 times the amount of data it needs at the current radar resolution. Processing times are not the focus of this thesis, but the overall impression from the work is that it is fully realistic and practical to achieve real-time processing with the proposed BFPD Tracker by increasing the efficiency of the implementation, utilizing a compiled language such as C/C#/C++ and/or by using powerful hardware. This is important for the actual application at airports.

The Knowledge-Base (KB) operations Delete Measurements in High Clutter areas (DMHC) and Delete Measurements in Uninteresting Areas (DMUA) were found to be very effective. These techniques are simple for stationary radars since the geographic terrain surrounding the radar is located in the same relative position at all times. DMUA is very decisive, and the map that defines the uninteresting areas should be carefully designed to include problematic areas while not including areas where tracking of an actual bird is of interest. These consideration may very well be conflicting in certain areas. Likewise, the High Clutter areas map of the DMHC should be carefully monitored to not include too large areas of land clutter. Since land clutter often has very high amplitudes, the DMHC might delete measurements over land altogether if not tuned correctly. Because of this last point, it might be advantageous to replace the DMHC by (or combine it with) a KB device that only deletes measurements from spatial cells that too frequently give rise to detections/plots. Even with the DMHC as it is tuned and described in Section 4.3, it may remove many of the true plots (resulting from birds) over the closest land areas of the airport, as can be seen from Figure 4.6. Also, as is shown in the video-file of Figure 5.19 (see digital appendix), many small areas still produce consistent plots that are not deleted by the KB operations. These are the areas where the DMHC and DMUA fail, arguing for the use of another KB device to eliminate frequent and lingering plots from small areas. This may also be improved by carefully designing the map of the DMUA to include these small and specific areas. The Clutter Map (CM) component of the Constant False Alarm Rate (CFAR) unit in the MTD does also not intercept the false alarms from these areas even though they are fairly static over a small area. This is because the CM operates on small cells and the amplitude in each channel is not large enough over time to build up the detection threshold in these cells.

6.1.2 Performance and tuning

Chapter 5 started with the presentation of the BFPD output for a batch length of $N = 7$ scans, with the tracker being fed data that was MTD-processed with a Cell Averaging Clutter Map (CACM) CFAR and $K_t = 6$. These settings, and the other parameters stated in Chapter 4, were found to work fairly well and constitute a basis for a BFPD Tracker that other parameter settings were tested against. As a batch length of $N = 7$ scans was found to give reasonable performance during the early testing in the development process, this setting was kept static during the whole development of the BFPD Tracker. All re-

quirements stated in Chapter 4 were also designed to perform well with this batch length. Therefore, as confirmed in section 5.2, the resulting batch associations and EW activations were naturally very reliable indicators of bird presence and potentially hazardous birdstrike situations, respectively.

Variations of CACM-CFAR and Ordered Statistic Clutter Map (OSCM) CFAR were also tested. The figures of the tracker output presented in this thesis were based on a CACM-CFAR, but indicative results from utilizing an OSCM-CFAR were shown in tables. There was not observed a great difference in the performance of the two CFARs. But by interpretation of the results, OSCM is perhaps performing slightly better on average than CACM. However, this comparison is not perfectly fair. To make a rightful comparison of the CFARs, they should ideally be compared at settings of K_t that result in equal P_{fa} (or P_d) for the two (see Equations 2.11, 2.12, 2.14 and 2.15). Simultaneously, OSCM is more computationally expensive and is perhaps performing better, but the accuracy of the experiments performed is good enough to show that there is not a great difference in average performance.

First when the proposed BFPD Tracker was fully developed and functioning, other batch lengths than $N = 7$ were tested. During these experiments, as presented in Section 5.6, interesting tracking results from other batch lengths emerged. A batch length of $N = 4$ (along with a CFAR multiplier increased to $K_t = 8$, see Figure 5.16) was found to give a particularly well performing balance between the consistent detection and tracking of BFPs and the suppression of false tracks. This raises the need for adjusting the values that determine the establishment of batch associations and activation of EWs such that the latter will meet the user's requirements for EW reliability.

Anyway – for any batch length, long-term testing of the BFPD Tracker in a prototype system at the airport is probably needed to determine whether the proposed tracker settings will accomplish satisfactory tracking and EW reliability. With this said, the work in this thesis indicates that it is possible to accomplish a workable Early Warning (EW) with a false EW rate that is sufficiently low so that precautionary anti-birdstrike measures might be initiated on the basis of the output of the developed tracker.

The whole system cascade consisting of an MTD processor, the BFPD Tracker and an EW function is evaluated. It is shown that for the right tuning of system parameters, batch associations and EW activations are made highly reliable indicators of bird presence and potential birdstrike situations respectively. The BFPD tracker undoubtedly supplies the radar operator with greatly beneficial information about bird presence, and may act as an automated surveillance system that identifies and quantifies bird activity. Also, since the BFPD tracker effectively sorts out and identifies bird presence in a large set of plots, the radar MTD processor may be operated at a lower detection threshold than what would be practical with merely manual interpretation of the MTD output. This in turn, will increase the probability of detection for a given target.

The tracker and EW output will serve the ATC operators best when the information is presented in a suitable manner. The presentation of the tracker output in this thesis is chosen merely for use in this context, and could (and should) be changed to a more suited format in the actual application. It would for instance be preferable to show in the radar video both the output of the MTD processor and the output of the tracker at the same time,

overlaid on a map. For a trained operator, the MTD output will show important information that is not present in the tracker output. The MTD detections could be shown with a fitting time history that fades into the background, while the most consistent confirmed tracks from the last three batches are shown on top of this, as an example. EW activations for the present batch, with no shown history, could then be continuously shown at the head of each BFP that is classified as a threat by the system.

The target dynamics model with target speed requirements of Table 4.2 worked very well. These requirements seem to allow tracking of birds that are of birdstrike concern, while excluding most other movement. There are however times when cars are interpreted as birds and may disturb the tracker output. For an example of this, see Figures 5.14 and 5.15 at coordinates (400,1900) where a short track with a sharp bend over the town of Stjørdal closely follows the road shown on the map. Though some artifact tracks such as this one results from other objects than birds, it seems that the important cases of bird activity are included by the model, and are thus allowed to be tracked. This is also in accordance with the theory given in Section 2.4.

6.1.3 Simulation

The simulation results of Section 5.5 collectively show the precision of tracking. The section gives important quantified results and states that the tracking position RMS error is 2.37 m in range, and 0.084° in azimuth, which in turn yields a RMS error in measured velocity of 1.64 m/s. The findings confirm that the tracking precision is sufficient for tracking of birds in a birdstrike avoidance application. The simulation results are however based on a Swerling 1 type target with a measured mean SNCR of 15.84 dB over sea clutter, which is not always the case.

For this radar system, an SNCR of 15.84 dB corresponds approximately to a radar target of 220 cm^2 at 3 km. By the SWEM model, this is equivalent to a bird of about 3.3 kg, which is the average weight of a goose. As an interesting parallel, the United States' Federal Aviation Administration (FAA) has defined what they call a Standard Avian Target (SAT), which is used in the FAA's specifications of requirements for acquisition of avian radars at American airports.[45] The SAT is defined to have an RCS of -16 dBsm $\approx$ 251 cm^2. The SAT is meant to approximate an average crow of 0.5 kg, but differs from the calculations of bird RCS in this thesis due to the use of a different model and assumptions regarding birds' RCS. As seen by the radar, the 0.5 kg SAT crow is therefore very similar to a single goose as described in Section 2.4, and also comparable to the simulated target.

While the FAA's specifications require that a system is able to detect an SAT at 2 km ($P_d = 0.9$) [45, p.21], the simulations show that this system is able to detect and track the simulated target (which is similar, but not identical to an SAT) at ranges of 2-4 km ($P_d = 0.875$) over sea. Note that the simulated target is most similar to an SAT around 3 km in range.

For the simulated target, the potential birdstrike hazard would depend on the model used for linking RCS to bird weight. For a SWEM model, the target corresponds to a 3.3 kg bird. For a 5-to-1 Prolate Spheroid side-view model, it roughly corresponds to a 2 kg

bird, and for the model used by the FAA in defining the SAT, it corresponds to a 0.5 kg crow. Comparatively, the Norwegian Civil Aviation Authority (Luftfartstilsynet) has classified all birds over 1 kg to be large in reference to a birdstrike. Therefore, the simulated target may be considered significant and a hazard if it were in proximity of the airport flight zones.

Situations with higher SNCR will be caused by the presence of a target with larger mean RCS or multiple targets with a larger common mean RCS. Such situations obviously also pose a more significant threat to aviation, and are thus of greater hazard. Similarly, a smaller mean RCS is of smaller hazard and yields a smaller SNCR. Therefore, for the situations of larger hazard than a single 0.5 kg crow SAT (or a 3.3 kg goose by the SWEM model), detection and tracking should be as good or better as the simulation on average.[1] For less significant situations on the other hand, the performance will be poorer on average than the simulations, as might be expected. How much worse is hard to say, but during testing and simulation the author experienced that for a somewhat weaker target than the simulated target with SNCR of 15.84 dB, the tracking was commonly disjointed since two consecutive misses occurred within a single batch or lost altogether because all batches contained at least two consecutive misses.

6.1.4 Main Findings in relation to objectives and research questions

Objectives

The objective of the master's thesis was given in Section 1.4. The primary objective was to automate the warning of birds by developing a system that identifies birds' movement by analyzing radar detections. This is achieved by the proposed BFPD Tracker. The probability of false warning is adjustable and seems realistically placeable within the requirements that may be expected for use in a birdstrike avoidance application (e.g. 1 false warning per 1000 warnings). However, longer periods of testing need to be performed to accurately estimate this probability. On the other hand, the probability of false alarm (single false report of target from a single scan) and of false confirmed track are documented in this thesis. The former might be specified arbitrarily by the detection threshold multiplier K_t in the MTD processor, but lower probability of false alarm comes at the cost of lowered probability of detection of true targets. The latter – the probability of false confirmed track, is found highly susceptible to parameter tuning in the tracker and can be pushed down to the order of 1 % without interfering with the detection and tracking of the most significant BFPs.

The secondary objective was to perform a theoretical analysis of the BFPD Tracker, test and optimize detection and tracking using real radar recordings. When optimizing detection and tracking, the whole processing cascade needs to be taken into account. This principle also applies to the theoretical analysis of the tracker, but because it is very hard

[1] This assumes that the number of misses in the simulation run accurately represents the average case scenario. While this almost certainly is not the case, it does not seem an unreasonable approximation based on the different simulation runs that were performed.

to qualitatively analyze the system as a whole, the theoretical analysis focuses on analyzing some system key components separately. First, some initial calculations on the radar's probability of detecting different bird species at different ranges (and different clutter/noise environments) was performed along with calculations of the normalized Clutter Attenuation (CA) in Section 3.4. These calculations confirm that the system should be able to detect a single bird up to about 4 km with a P_d of about 0.7 depending on species and environment. It also quantifies the radar's detection performance on four relevant bird species and finds the CA over land to be 26 dB on average. These results seem reasonable and are in agreement with the observed real performance of the system when detecting birds in the terrain in and around Værnes Airport.

As part of the theoretical analysis, the probability of establishing a confirmed track from measurements was also investigated in Section 4.8 by making use of Markov Chains. Here it was found that the BFPD Tracker's increased strictness for track confirmation compared to a classic M-out-of-N criterion induced a significant increase in performance for a batch length of $N = 7$ and a minimum of 4 observations per batch. The probability of establishing confirmed tracks from true detections ($0.7 < P_d < 1$) was found similar to that of a 2-out-of-3 criterion while at the same time, the probability of establishing confirmed tracks from false alarms ($10^{-4} < P_{fa} < 10^{2}$) was found about four orders of magnitude lower than that of a 2-out-of-3 criterion (and half of that of the much stricter 4-out-of-7 criterion).

Testing and optimization of the Automatic Detection and Tracking (ADT) were carried out by making use of real radar signals that were recorded at Værnes Airport. These signals included the recording of many birds that were visually identified and confirmed by the airport ATC personnel at the time of recording. Testing consisted of very many smaller branches of development and troubleshooting in the search for overall improvement. Optimization was performed by manually varying the many parameters of the MTD and tracker.

In addition to the above, testing with a simulated target that was added to complex samples of real sea clutter and noise was performed to further quantify the tracker and system performance. The simulation found that a 15.84 SNCR target is reliably detected and tracked during a flight varying from about 4-2 km in range over sea. It also found that the RMS error of the measured position is 2.37 m in range and 0.084° in azimuth, and the RMS error of the velocity induced by position measurements is 1.64 m/s. This geometrically induced velocity was also found to be in agreement with the velocity measurements derived from Doppler measurements to an accuracy of ± 3 m/s, which is sufficient for this application.

Research questions

Section 1.4 also states some research questions.

The first of these questions was: Is batch processing and the concept of a "bird trajectory detector" suited for the intended application?

In hindsight, it seems that the principle of implementation chosen in this work is a nice method for circumventing several classic tracking estimation problems and still be able to track birds with relatively good performance. Batch processing in particular obviates the

need for several housekeeping tasks, such as keeping a limit on the number of simultaneous and redundant tracks that are caused by the association of a single plot to several tracks. This is automatically limited since the simultaneous tracks only exist within a single batch. Another example is that the batch processing structure makes all information about all measurements that are to be tracked (within the relatively short batch) available right at the beginning of processing. This makes for instance the estimation of a resolution cell's mean clutter level (over the batch length) in the DMHC easy. On the other hand, it creates the need for batch association calculations to create linkage between the difference batches. Another possible approach to link the batches, that may perform as well as this one, is to specifically search for tracks in the extension of the tracks already found in previous batches. All in all, it seems the batch processing structure is not a bad choice, but it is not known whether a similarly designed tracker with iterative processing structure would outperform the current implementation. Batch processing is not abundantly used in tracker research and development, but it is perhaps an underestimated processing scheme in regards to tracking. Studies such as [61] have shown that batch processing (in this case, a Batch-Least-Squares-scheme) may actually outperform even the well-established iterative Extended Kalman Filtering (EKF) algorithm. Overall, the results of this thesis reveals that batch processing and the BFPD Tracker's principle of operation is well suited for application in an avian tracker. Furthermore, it also seems functional to extend this tracker with an EW functionality for use in a birdstrike avoidance radar system at airports.

The second research question was: How does the adjustment of different parameters affect performance and what are the system's critical parameters?

The system's critical parameters are a selected few parameters whose precise tuning will make or break the overall system performance. In the MTD, the most critical parameter is the threshold multiplier $K_t \in [0, \infty\rangle$. In the tracker itself, the most critical parameters are the batch length $N \in [2, 3, 4 \ldots]$ and the minimum number of observations within a batch M (integer between 1 and N). But also worth mentioning is the collective setting of the parameters that make up the Score Function (SF) in the BFPD Tracker. These parameters are listed in Table 4.3, and define the requirements of track quality for establishing and keeping a confirmed track.

When the threshold multiplier K_t is decreased, it causes more false alarms, a higher probability of detection for all targets, more observations and may cause the amalgamation of several false alarms or detections into larger detectionheaps. When K_t is increased, the opposite effect is induced.

A longer batch length N is observed to exclude tracking of maneuvering targets to a larger extent, causes a larger static time delay from real-time, and requires to a larger degree that the target is observed consistently for a longer period of time. A shorter batch length includes tracking of more maneuvering targets and causes less time delay, but gives less opportunities for the system to detect the target within each batch and is more prone to accept tracking of false alarms from specific zones of high land clutter. A higher ratio M/N causes fewer, but more reliable confirmed tracks, while a lower ratio M/N causes more, but less reliable confirmed tracks. Likewise, stricter requirements in the SF specification (the parameters of Table 4.3) also induces fewer, more reliable confirmed tracks, and vice versa.

The third research question was: What are potentially hazardous birdstrike situations and how sinister or how trivial situations can be reliably identified?

The Norwegian Civil Aviation Authority (Luftfartstilsynet) has, as mentioned, classified all birds weighing over 1 kg as large in a birdstrike context. Commercial airliners are required to withstand birdstrikes of a certain magnitude. For instance, aircraft engines are required to withstand the ingestion of a single bird of maximum weight between 1.8 and 3.65 kg (depending on engine inlet area) without catching fire, suffering uncontained failure, becoming impossible to shut down or losing more than 50 % of maximum thrust for the initial 14 minutes after ingestion. Likewise, the general aircraft airframe is required to continue safe flight and subsequent normal landing after the impact of a 1.8 kg bird at cruising speed at mean sea level, and a 3.6 kg bird for the empennage in particular.[58]

The comprehensive 2013-report on Wildlife strikes to Civil Aircraft by the FAA (found in [21]) states the total number of birdstrikes in the US from 1990 to 2013 by species. It also includes information of the number of these that caused aircraft damage, negative Effect On Flight (EOF), the number that consisted of strikes by multiple birds, total aircraft downtime and total costs. For Geese, Gulls and Sparrows these results are summarized in Table 6.1, which gives empirical evidence for the birdstrike hazard of each species group. Reading from this table, one can see that birdstrikes involving geese have a 53.3% probability of resulting in aircraft damage and a 28.2% probability of having a negative EOF. Likewise, birdstrikes involving gulls have a 14.2% probability for aircfaft damage and 11.7% for negative EOF.

Table 6.1: Total number of birdstrikes, and birdstrikes that caused aircraft damage, negative Effect On Flight (EOF), multiple birds and resulting aircraft downtime in hours and total costs in USD by species group. 24-year totals (1990-2013) obtained from the FAA report found in [21].

Species group	Total birdstrikes	Aircraft damage	Neg. EOF	Multiple birds	Aircraft downtime	Total costs
Geese	1934	53.3%	28.2%	41.7%	135829 h	$ 152.3 M
Gulls	9656	14.2%	11.7%	21.5%	59078 h	$ 53.5 M
Sparrows	3725	1.6%	3.1%	20.6%	1102 h	$ 0.86 M

The simulation work in this thesis, as said in Section 6.1.3, indicates that this system can detect and track a target roughly the size of a gull (221 cm^2) reliably ($P_d = 0.875$ and continuous tracking) up to around 3 km in range over sea with an acceptable false alarm rate. The testing with real radar measurements confirms this result, but also shows that performance is varying. Consistent detection and tracking beyond 3 km are also observed in the experiments with real measurements, but these results are harder to firmly quantify due to the total lack of knowledge about these targets.

Testing with the Case 3 data (containing multiple flocks of geese) indicates that a medium-sized flock of geese is consistently detected and continuously tracked at ranges up to about 4 km over both sea (see Figure 5.16) and land (see Figure 5.2). The geese are also detected at ranges above 4 km, but not consistently so. For all results, it may be expected that the overall performance (consistency and/or maximum range) of detection, tracking and warning will be improved if a radar target were to have a larger individual

target Radar Cross Section (RCS) or if it were to consist of a higher number of birds. Likewise, a smaller bird RCS than what is quantified in the context of all results, will likely cause loss of consistency and maximum range.

These results, along with the hazard-data of Geese and Gulls, give a fair description of how the system is able to identify birdstrike situations of different hazardousness or consequence. How probable it is that a birdstrike actually occurs is a different matter, which is not investigated in this thesis. When judging a potential impact by a birdstrike avoidance radar system on the overall birdstrike risk, the probability of a birdstrike happening in different situations also needs to be accounted for. Overall birdstrike risk is found by the expression Risk = Probability $\times$ Hazard, in which Hazard is synonymous to the consequence or aftermath of a birdstrike.

6.1.5 Main contributions

Tracking is a well established field, as is radar. Detection of birds by radar is also relatively widespread, but the utilization of a specialized tracker for automatic tracking and warning, and a pseudocoherent (coherent-on-receive) radar based on a standard civil marine radar for birdstrike prevention is novel research, as far as the author knows. The work of this thesis is an extension of almost 10 years of research and development by Radian AS and SINTEF research organization. The author was introduced to the main part and findings of this work, and was provided a working MTD processor that employed Fast Fourier Transform (FFT) Doppler processing of the coherent radar signal and employed a Cell Averaging- or Ordered Statistic-CFAR. The author was also provided access to an operational radar system that was deployed at Værnes Airport in 2014, a windows-based radar display tool called PPI for detailed viewing of the MTD output (developed by Radian AS), and some computational framework for P_d-calculations.

The following points outline the main contributions made by the author in the work of this thesis.

- Development and implementation of a Bird Flight Path Detector (BFPD) Tracker. This is a complete workable tracking system that is made from scratch and includes the following units/functions.
  - Two different Plot Extraction algorithms
  - Knowledge Base (KB) operations
  - Tracking of targets within one batch
  - Evaluation of track quality
  - Inter-batch association of tracks that originate from the same target(s)
  - An Early Warning (EW) functionality that makes the system operator aware of situations of a certain magnitude.
- Implementation of a Cell Averaging Clutter Map Constant False Alarm Rate (CACM-CFAR) into the existing MTD processor.

- Implementation of an Ordered Statistic Clutter Map Constant False Alarm Rate (OSCM-CFAR) into the existing MTD processor.
- Analysis and optimization of the BFPD Tracker.
- Evaluation of the total system with the MTD processor operating in cascade with the BFPD Tracker.

6.2 Methodical discussion

Due to the many parameters of the CFAR, Clutter Map, plot extractor, target dynamics model, gating/prediction, data association process, track confirmation and batch association process, there is a multitude of different parameter configurations that may all yield mediocre to excellent performance. Since all combinations could not be tested during the optimization process many parameters were simply estimated with a rough calculation deemed reasonable by the author. These were then kept constant through the development and testing, and were only changed if the setting of a parameter was discovered to specifically impede the overall performance. In other words – the performed optimization process does not guarantee optimal system performance, but merely locates some points in parameter space that yield the best performance currently achieved.

The plot extractor is a part of the system that greatly influences the overall tracker performance. The importance of this component should not be underestimated. The accurate extraction of plots from a detection matrix is also a complex problem that can be solved in many ways. For the most part, the Two-Pass CCL algorithm with plot splitting was superior to the other plot extraction algorithms that were developed in the work of this thesis. Therefore, this algorithm was deemed the best fit, and results from the use of other algorithms were omitted to reduce the amount of results and focus on the tracking itself. To achieve detailed insight to the plot extraction algorithms, beyond the description of Section 4.2, the reader is referred to the digital appendix, which contains all code of the BFPD Tracker system.

The BFPD Tracker is a specialized tracker that was specifically designed for the bird-strike avoidance radar application. Furthermore, it was designed to work well in the specific processing cascade described in this thesis, which it does. However, it would not be surprising to find that the BFPD Tracker may perform worse (or better) than the level of performance found here when it is utilized on other types of data.

The input also is dependent on the radar's location and surroundings. For instance, shadowing, mirroring and other propagation effects is commonly observed in the input radar data that is used in this thesis. Therefore, tracking is sometimes lost merely because there are no detections of the target. This is solely due to external factors, and are outside the scope of this thesis. Relocation of the radar, moving it up or down in height or changing any part of its surroundings will possibly effect the performance total system substantially, either positive or negative.

Likewise, changing any part of the radar, will influence the tracking results. However, the performance of the civil marine radar and computer that constitute the pseudocoherent

radar itself exhibits impressive performance for initially being non-coherent, but is neither especially superior nor poor compared to other systems that are capable of detecting birds (not comparing monetary costs). Therefore, there is not good reason to believe that the performance of the BFPD Tracker will change drastically if the input is being generated by a different radar system as long as this system has a detection-performance comparable to that of the radar used here.

A key strength in the method used in this thesis is the three-way combination of theoretical analysis, simulatory and real experiments. The theoretical analysis provides a framework and foundation for the experiments. The simulation provides key knowledge about the tracking performance given a specific, known target. This method is further strengthened by using real clutter and noise recordings, which eliminates any simulatory mismatch in the statistical properties of these entities. And last, the experiments with real radar recordings provide an evaluation of the performance in real life situations. The combination of these methods makes the results more robust when the results from the three is in agreement, and increases the accuracy and generality of both the results and their implications.

6.3 Implications

6.3.1 Application

Most Multiple Target Tracking (MTT) applications (at least by the author's impression) emphasize the precise tracking of multiple individual targets. Many modern trackers also do not allow tracking of all targets when for instance a single target splits into several new targets, or are unsuited when a multitude of targets fly with variable, but small separation and are detected separately. For instance, tracking systems whose targets are nautical vessels or civil aircraft seldom see these situations. Rather, the birdstrike-prevention application prioritizes the high-reliability detection of the presence of multiple targets that fly closely separated, land, take off and maneuver quickly and chaotically. With proper tuning, the system might exclude tracking of targets with particularly high or low maneuvering. These properties make the system suitable for tracking application in a birdstrike avoidance radar system, for which there currently does not exist a widespread and abundant system at airports. Successful deployment of such a system at airports will possibly enhance the safety of all aviation – civil and military.

Globally, as birdstrike poses an increasingly troubling risk at airports around the world, this is currently an area of development and focus of several parties. To mention some, Robin Radar [29] has during the last couple of years worked on employing such a system at the International Airport of Amsterdam Schiphol, amongst others.[30] The US FAA is investigating the capabilities of commercially available systems. The US Air Force (USAF) has developed the US Avian Hazard Advisory System (AHAS) which uses Geographic Information System (GIS) to make predictive models of bird habitat, migration and breeding characteristics for use in birdstrike mitigation.[23] The US Department of Defense (DoD) also runs a prevention program called the Bird Aircraft Strike Hazard (BASH).[44]

The properties of the tracker may also open for other tracking applications that emphasize similar properties to the BFP detection problem. One such application might be in a ground surveillance system that looks to automatically identify the presence and movement trends of pedestrians. An especially provident application might be detection of the presence of drones or micro-UAVs over airports, military bases or governmental facilities. In the future, drones are also likely to fly in flocks or swarms, and might be fitted with surveillance or weaponry features. A scenario where the encounter of such drones is likely would create an imminent need for a flight path detecting tracker that looks for single or closely separated, highly maneuverable airborne targets that have especially small Radar Cross Sections (RCS).

6.3.2 Future work

For the system and the tracker to actually be workable and applicable in a way that will improve aviation safety at airports (or any other application), further work is needed. The list below comprises some points that will be of importance if the tracker is to contribute to aviation safety in any significant way.

- *Implementing a track merging function*
  A track merging function is a function built into the tracker that merges multiple confirmed tracks that are closer than some distance limit into a single confirmed track. The need for such a function is especially induced by the tracker associating a single plot to multiple tracks. The author would recommend a distance limit of about 20 m in either direction perpendicular to the target course. The precision of the position of the merged track may be somewhat reduced (to about the size the limiting distance in the worst case), but the number of confirmed tracks will be reduced heavily. Thus, the computational load of any operations on the confirmed tracks (such as an EW) will also decrease, and the number of confirmed tracks will accurately represent the number of areas where BFPs are detected and tracked. Therefore the readability of the tracking output is also improved. Better performance is not guaranteed with such a function implemented, but it is definitely worth investigating.

- *Revision and translation to compiled programming language to achieve near real-time processing*
  By revision of the code and implementation as made by the author, there will quickly be discovered many parts of the code that may be changed to improve efficiency. Code efficiency has not been of very high priority during the work of this thesis. Additionally, the code needs to be translated into a compiled programming language such as C/C#/C++ to achieve near real-time processing. Parts of the programming will also have to be changed. By near real-time, it is here meant that a single batch of N scans is recorded, processed and output by the tracker before the subsequent batch is done being recorded. If the processing of one batch takes a little longer than the recording of the next, delays will build up and whole batches of data will have to be dropped to catch up with the near real-time processing schedule.

- *Integration of the tracker into a complete birdstrike avoidance radar system*
  Parts of the tracker may be more or less directly used in a near real-time prototype system, since the tracker is still batch processing. However, interfacing code structures also need to be developed for the tracker to function properly within this framework. Therefore, the integration of the tracker into a complete birdstrike avoidance radar system will demand some amount of research and development depending on the data flow of this system.

- *Deployment and live testing of a complete, near real-time birdstrike avoidance radar prototype with tracking at a major Norwegian airport*
  To experience the tracker's true capability in this application, its impact on aviation safety and potential as an operational asset, a prototype employing a further developed version of the BFPD Tracker needs to be placed at a major airport. Much of the framework needed for such a deployment is already in place at Værnes Airport, and integration of the tracker into a near real-time prototype system may mostly consist of software development and updating. Live testing will provide practical experience and make possible some empirical evaluation of the system's impact on the birdstrike frequency and magnitude at that airport.

Chapter 7

Conclusion

A proposed BFPD Tracker that is intended to aid the avoidance of birdstrikes at airports has been developed, implemented, evaluated and tested on real radar data recorded at Værnes Airport. The tracker utilizes batch processing of different batch lengths to identify and track avian activity in the airport vicinity out to 5.3 km by analyzing the output of a Moving Target Detector (MTD) processor.

A Early Warning (EW) function that analyzes the tracker output has also been developed. The goal of this EW is to automate the identification of potential hazardous birdstrike situations and draw the ATC operator's attention to these situations as they are discovered. A well-functioning tracker and EW function could relax the need for continuous monitoring of the radar video by a human operator.

The idea of a tracker that uses batch processing and detects Bird Flight Paths (BFPs) within each batch was found suited for application in a birdstrike avoidance radar system. The performance of the tracker, and the detection- and tracking-capability of the total system was analyzed, evaluated and attempted optimized through theoretical analysis, simulation and testing with real radar recordings from Værnes Airport, Norway.

In short, the results showed that the tracker system is capable of detecting and tracking bird targets with sufficient performance to have potential as an operational asset in mitigating birdstrike risk. The system was found to be able to consistently detect, and continuously track individual birds roughly the size of a seagull at several kilometers in range. Also, an incoming flock of geese was consistently detected and continuously tracked at 4 km in range. An Early Warning (EW) system, tuned to yield a very low false warning rate, was also activated by this flock of geese at a range of 3.5 km, giving warning about 3.5 minutes in advance of a potential birdstrike incident. If the system is developed further it may provide an increasingly needed and requested capacity at airports and contribute to increased aviation safety.

Bibliography

[1] Adressavisen. Fly kolliderte med svane på Værnes, August 2014. http://www.adressa.no/nyheter/innenriks/article8892767.ece.

[2] H. Askar, Xiaofeng Li, and Zaiming Li. Performance analysis of dim moving point target detection algorithms. In *Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on*, volume 2, pages 977–981 vol.2, June 2002.

[3] Avinor. Avinor, August 2014. http://www.avinor.no.

[4] Y. Bar-Shalom, F. Daum, and J. Huang. The probabilistic data association filter. *Control Systems, IEEE*, 29(6):82–100, Dec 2009.

[5] D.K. Barton. *Modern radar system analysis*. Artech House Radar Library. Artech House, 1988.

[6] D.K. Barton. *Radar Equations for Modern Radar*. Artech House radar library. Artech House, 2013.

[7] British Broadcasting Corporation (BBC). Pilot lands jet after bird strike. http://news.bbc.co.uk/2/hi/uk_news/england/manchester/6606375.stm, April 2007. Accessed November 2014.

[8] E. Ben-Yaacov, D. Quartler, and N. Levanon. Inter-pulse coding and coherent-on-receive modifications of magnetron-based marine radar #x2013; experimental results. In *Microwaves, Communications, Antennas and Electronics Systems (COMCAS), 2011 IEEE International Conference on*, pages –, Nov 2011.

[9] A. Benavoli, L. Chisci, A. Farina, L. Timmoneri, and G. Zappa. Knowledge-based system for multi-target tracking in a littoral environment. *Aerospace and Electronic Systems, IEEE Transactions on*, 42(3):1100–1119, July 2006.

[10] J. Barrie Billingsley. *Low Angle Radar Land Clutter - Measurements and Empirical Models*. William Andrew Publishing Inc., SciTech Publishing Inc. and IEE, 1st edition, 2002.

[11] S.S. Blackman. *Multiple-target Tracking with Radar Applications*. Radar Library. Artech House, 1986.

[12] Stephen Blake. Os-cfar theory for multiple targets and nonuniform clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 24(6):785–790, November 1988. ieeexplore.ieee.org.

[13] John-Olav Brattebø. Detection of birds with a Moving Target Detector (MTD) magnetron radar. Project report, Norwegian University of Science and Technology (NTNU), 2013.

[14] J.N. Briggs and Institution of Electrical Engineers. *Target Detection by Marine Radar*. IEE Radar Series. Institution of Engineering and Technology, 2004.

[15] J. Demetry and H. Titus. Adaptive tracking of maneuvering targets. *Automatic Control, IEEE Transactions on*, 13(6):749–750, Dec 1968.

[16] Z. Ebrahimian. Complex spatial/temporal cfar. In *Wireless Communications and Applied Computational Electromagnetics, 2005. IEEE/ACES International Conference on*, pages 1045–1049, 2005.

[17] Benavoli et. al. Knowledge-based radar tracking. *Knowledge-Based Radar Detection, Tracking and Classification*, May 2008.

[18] Chen Weishi et. al. Flight path detection of bird targets in radar images. *Chinese Journal of Electronics*, 18(1):192–194, 2009.

[19] H. Schmaljohann et. al. Quantification of bird migration by radar a detection probability problem. *Ibis*, 150(2):342–355, 2008.

[20] T.G.Konrad et. al. Radar characteristics of birds in flight. *Science*, 159(3812):274–280, January 1968.

[21] Federal Aviation Administration (FAA). Serial report number 20 - wildlife strikes to civil aircraft in the united states 1990-2013. http://www.faa.gov/airports/airport_safety/wildlife/media/Wildlife-Strike-Report-1990-2013-USDA-FAA.pdf, July 2014. Accessed September 2014.

[22] Netherlands Organisation for Applied Scientific Research TNO. Computer aided radar performance evaluation tool. https://www.tno.nl/downloads/carpet.pdf, 2014. Accessed October 2014.

[23] US Air Force. United states, avian hazard advisory system. http://www.usahas.com/home/. Accessed November 2014.

[24] F. Gini and M. Rangaswamy. *Knowledge Based Radar Detection, Tracking and Classification*. John Wiley & Sons Inc., 1st edition, 2008.

[25] Charles M. Grinstead and James L. Snell. *Introduction to Probability*. American Mathematical Society, 2nd edition, 2006.

[26] Siddharth Gupta, Diana Palsetia, Md.Mostofa Ali Patwary, Ankit Agrawal, and Alok Choudhary. A new parallel algorithm for two-pass connected component labeling. In *Parallel Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International*, pages 1355–1362, May 2014.

[27] Lifeng He, Yuyan Chao, Kenji Suzuki, and Kesheng Wu. Fast connected-component labeling. *Pattern Recognition*, 42(9):1977 – 1987, 2009.

[28] DeTect Inc. www.detect-inc.com, July 2014. Accessed September 2014.

[29] Robin Inc. www.robinradar.com. Accessed September 2014.

[30] Robin Inc. Press release december 18, 2012 - robin flies high: Bird strike prevention radar system takes off. http://www.robinradar.com/wp-content/uploads/2014/08/Robin-Radar-Systems-Press-Release-Bird-strike-prevention-radar-takes-off_19.12.123.pdf. Accessed November 2014.

[31] Norsk institutt for naturforskning (NINA) / Norwegian institute for nature research. Nina rapport 532 - konfliktar mellom vilt og luftfart: Samling av eksisterande kunnskap i endnote web litteraturbase. http://www.nina.no/archive/nina/pppbasepdf/rapport/2009/532.pdf, December 2009.

[32] Intel. Intel Software Developer Zone - Reference Manual for Intel Integrated Performance Primitives 8.2 - Flood Fill Functions, August 2014. https://software.intel.com/en-us/node/504475.

[33] Ebi Jose. *Robotic Navigation and Mapping with Radar*. Artech House Radar Library. Artech House, 2012.

[34] Paul R. Kalata. #x003b1;- #x003b2; target tracking systems: a survey. In *American Control Conference, 1992*, pages 832–836, June 1992.

[35] Emil B. Kalveland. Using digital signal processing to discern moving targets with a navigation radar. Bachelor's thesis, Trondheim and Sør-Trøndelag University College, 2012.

[36] Stein Kristoffersen, Karina V. Hoel, Øyvind Thingsrud, and Emil B. Kalveland. Digital coherent processing to enhance moving targets detection in a navigation radar. In *International Radar Conference 2014*, October 2014.

[37] X.R. Li and V.P. Jilkov. Survey of maneuvering target tracking. part i. dynamic models. *Aerospace and Electronic Systems, IEEE Transactions on*, 39(4):1333–1364, Oct 2003.

[38] Furuno Electric Company LTD. Furuno operator's manual. www.furunousa.com/ProductDocuments/FAR21x7-%2028x7%20Operator's%20Manual%20P%20%204-8-11.pdf, April 2011. Accessed August 2014.

[39] Norwegian Civil Aviation Authority Luftfartstilsynet. Flight Safety Statistics - 17. Birdstrike, August 2014. http://www.luftfartstilsynet.no/flysikkerhetsstatestikk/C_Birdstrike1.htm.

[40] Mark A. Richards. Generating Swerling Random Sequences, December 2008. http://users.ece.gatech.edu/mrichard/Generating

[41] M. Morelande and B. Ristic. Signal-to-noise ratio threshold effect in track before detect. *Radar, Sonar Navigation, IET*, 3(6):601–608, December 2009.

[42] Fred E. Nathanson. *Radar Design Principles - Signal Processing and the Environment*. McGraw-Hill Inc., SciTech Publishing Inc., 2nd edition, 1991.

[43] M. Xia N.Huansheng, C. Weishi and L. Jing. Bird-aircraft strike avoidance radar. *Aerospace and Electronic Systems Magazine, IEEE*, 25(1):19–28, Jan 2010.

[44] US Department of Defense. Bird/wildlife aircraft strike hazard (bash). http://www.dodpif.org/groups/bash.php. Accessed November 2014.

[45] U.S. Department of Transportation Federal Aviation Administration (FAA). Advisory circular number 150/5220-25. http://www.faa.gov/documentLibrary/media/Advisory_Circular/150_5220_25.pdf, November 2010. Accessed September 2014.

[46] Accipiter Radar. www.accipiterradar.com. Accessed September 2014.

[47] Radian AS. Radian, August 2014. http://www.radian.no.

[48] R. S. Raghavan. Analysis of ca-cfar processors for linear-law detection. *Aerospace and Electronic Systems, IEEE Transactions on*, 28(3):661–665, 1992.

[49] D.B. Reid. An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, 24(6):843–854, Dec 1979.

[50] Jane's Airport Review. Radars offer bird-strike warning. http://www.accipiterradar.com/media/pdf/Janesairportreview_Jun12009.pdf, June 2009. Accessed September 2014.

[51] S.O. Rice. Mathematical analysis of random noise. *Bell System Technical Journal*, 23(3):282–332, July 1944.

[52] H. Rohling. Ordered statistic cfar technique - an overview. In *Radar Symposium (IRS), 2011 Proceedings International*, pages 631–638, Sept 2011.

[53] Hermann Rohling. Radar cfar thresholding in clutter and multiple target situations. *IEEE Transactions on Aerospace and Electronic Systems*, AES-19(4):608–621, July 1983. ieeexplore.ieee.org.

[54] M. Shor and N. Levanon. Performances of order statistics cfar. *Aerospace and Electronic Systems, IEEE Transactions on*, 27(2):214–224, Mar 1991.

[55] SINTEF. SINTEF Research Organization, August 2014. http://www.sintef.no.

[56] Merrill Skolnik. *Introduction to Radar Systems*. McGraw-Hill, 3rd edition, 2002.

[57] Merrill Skolnik. *Radar Handbook*. McGraw-Hill, 3rd edition, 2008.

[58] SKYbrary. Aircraft certification for bird strike risk. http://www.skybrary.aero/index.php/Aircraft_Certification_for_Bird_Strike_Risk, March 2014. Accessed November 2014.

[59] G.E. Smith, N. Majurec, A. O'Brien, J. Pozderac, C.J. Baker, J.T. Johnson, D.R. Lyzenga, O. Nwogu, D.B. Trizna, D. Rudolf, and G. Schueller. High power coherent-on-receive radar for marine surveillance. In *Radar (Radar), 2013 International Conference on*, pages 434–439, Sept 2013.

[60] Swati and G. Dixit. Improved algorithm for blob detection in document images. In *Confluence The Next Generation Information Technology Summit (Confluence), 2014 5th International Conference* -, pages 703–708, Sept 2014.

[61] H. Satz T. H. Kerr. Comparison of batch and kalman filtering for radar tracking. In *The Annual AIAA/BMDO Technology Conference [10th] Held in Williamsburg, Virginia on July 23-26, 2001. Volume 1. Unclassified Proceedings*, pages 9.1–9.7, 2001.

[62] The New York Times. 1549 to tower: 'we're gonna end up in the hudson'. http://www.nytimes.com/2009/01/18/nyregion/18plane.html, January 2009. Accessed August 2014.

[63] C.R. Vaughn. Birds and insects as radar targets: A review. *Proceedings of the IEEE*, 73(2):205–227, Feb 1985.

[64] Wikipedia. Bird Strike, November 2014. http://en.wikipedia.org/wiki/Bird_strike.

[65] Wikipedia. Connected Component Labeling, December 2014. http://en.wikipedia.org/wiki/Connected-component_labeling.

[66] Wikipedia. Eastern Air Lines Flight 375, November 2014. http://en.wikipedia.org/wiki/Eastern_Air_Lines_Flight_375.

[67] Wikipedia. Markov Chain, November 2014. http://en.wikipedia.org/wiki/Markov_chain.

[68] Wikipedia. Secondary Surveillance Radar, November 2014. http://en.wikipedia.org/wiki/Secondary_surveillance_radar.

[69] Christian Wolff. Radar Tutorial - Plot Extractor, December 2014. http://www.radartutorial.eu/10.processing/sp08.en.html.

[70] Christian Wolff. Radar Tutorial - Pseudo-coherent Radar, August 2014. http://www.radartutorial.eu/08.transmitters/Pseudo-coherent%20Radar.en.html.

[71] Suzuki K. Wu K., Otoo E. Optimizing two-pass connected component labeling algorithms. *Pattern Analysis and Applications*, 12(2):117–135, June 2009.

Appendix

A Radar unit and location

Figure 7.1: The view of the sea as seen from the radar site. Also showing some of the terminal building and the runway at Værnes Airport.

Figure 7.2: The radar site, the terminal building, the ATC Tower and the runway at Værnes Airport.

Figure 7.3: A map of Værnes Airport and surroundings.

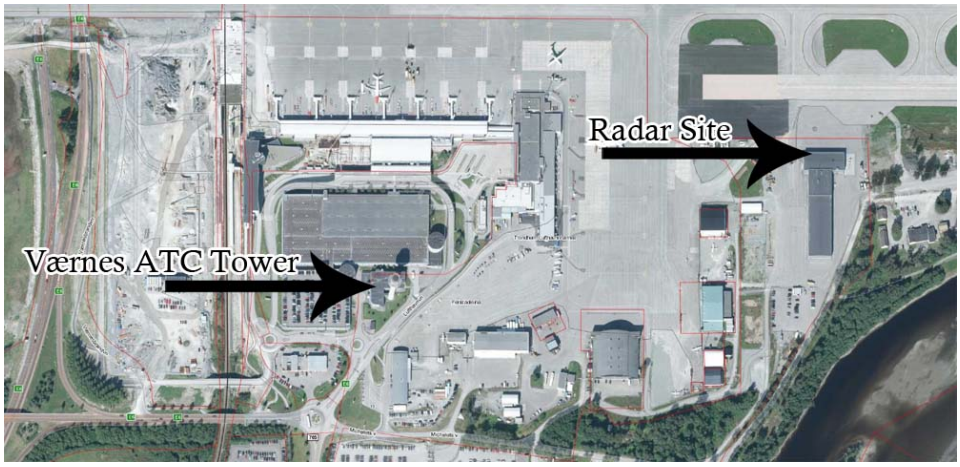


Figure 7.4: A close-up hybrid map and aerial photo of the airport. Black arrows indicate the locations of the radar site and the ATC Tower.

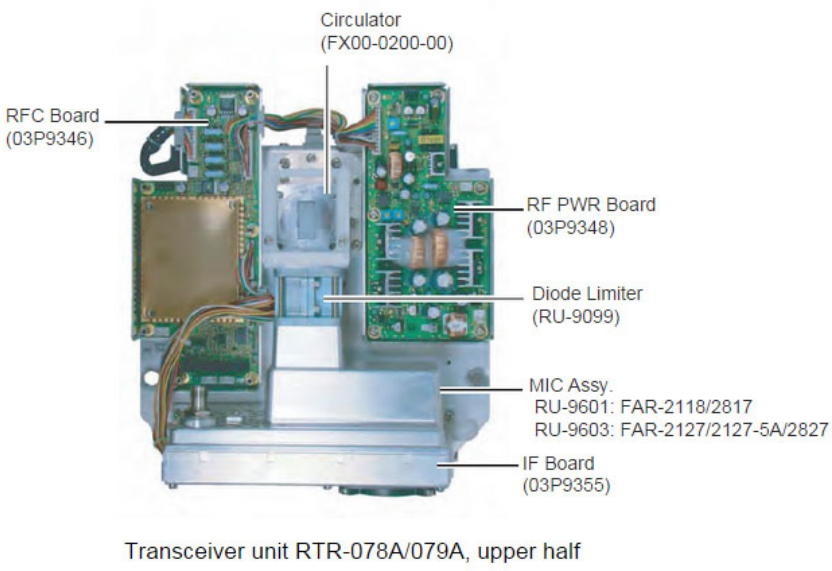


Figure 7.5: The upper half of RTR-079A transceiver unit and its internal components.

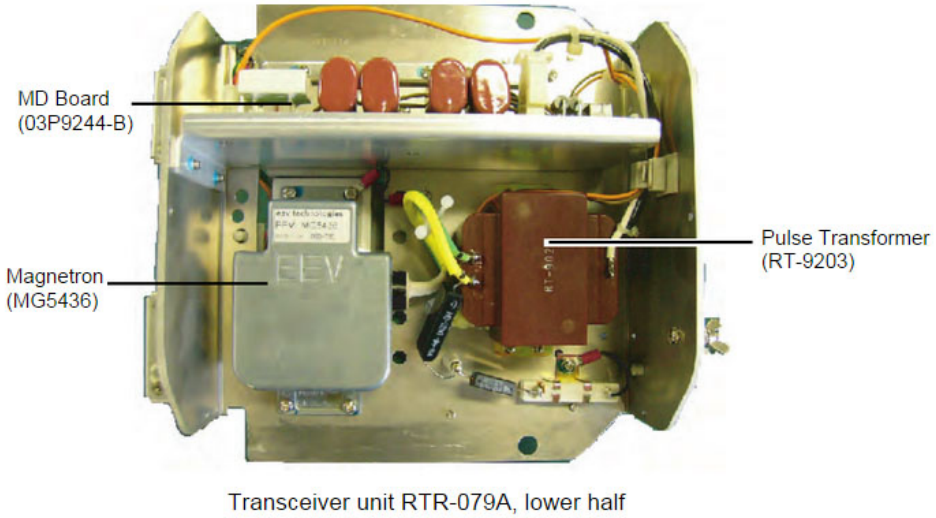


Figure 7.6: The lower half of RTR-079A transceiver unit and its internal components.

B Formulas for derivation of specifications

The radar range resolution is given by

$$\Delta R = c\tau/2 \tag{B.1}$$

where ΔR is the range resolution, c is the speed of light and τ is the pulse duration.

The doppler shift from a moving target at a stationary radar is

$$f_d = \frac{2 f_t v_r}{c} \tag{B.2}$$

where, f_d is the doppler shift frequency, f_t is the carrier frequency, v_r is the radial velocity of the target and c is the speed of light.[56, p.105]

The first blind speed of a radar is

$$v_b = \frac{\lambda f_p}{2} \tag{B.3}$$

where λ is the wavelength of the carrier and f_p is the PRF.[56, p.114]

The CPI in number of pulses per scan is

$$n = \frac{\theta_B f_p}{6\omega_r} \tag{B.4}$$

where n is the number of pulses returned per scan from a point target (also called hits per scan), θ_B is the antenna beamwidth in degrees, f_p is the PRF and ω_r is the antennas rotation rate in rpm.[56, p.45]

The Doppler resolution Δv of the radar (in m/s) is given by the following relationship.

$$\Delta v = v_b/n \tag{B.5}$$

C MATLAB code

This section will show some of the MATLAB code used for calculations and analysis during the work that is presented in this thesis. The MATLAB code of the Bird flight Path Detector (BFPD) Tracker itself can be found in the digital appendix (see Appendix E).

C.1 Markov Chain calculations

```
format long
% Probability of detecting the target within its predicted gate
D = 0.8;
% Probability of NOT detecting the target within its predicted gate
M = 1-D;
```

```
disp('Markov Chain Calculations:')

%% 4 out of 7 and not 2 consecutive misses Criterion
% Transition Matrix
T = [
M D 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 D 0 0 0 0 0 M 0 0 0 0 0 0 0;
0 0 0 M 0 0 D 0 0 0 0 0 0 0 0 0;
M 0 0 0 D 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 M 0 0 0 0 0 0 0 0 D;
M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
0 0 0 0 0 0 0 M 0 0 0 0 0 0 D;
M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
M 0 0 0 0 0 0 0 0 D 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 D 0 M 0 0 0;
0 0 0 0 0 0 0 0 0 0 M 0 0 0 D;
M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
M 0 0 0 0 0 0 0 0 0 0 0 0 D 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 0 M D;
M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1;]';
% Initial state probability vector
P = [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]';
% Calculation
k = 7;
Pk = (T^k)*P;
P_SystemConfTrack = Pk(16)

%% 4 out of 7 Criterion
T2 = [          %10                  20                  30
M D 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 D 0 0 0 0 0 0 0 0 0 0 0 0 M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 M 0 0 0 0 0 0 0 D 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 M 0 0 0 0 D 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 M 0 D 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
M 0 0 0 0 0 D 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
0 0 0 0 0 0 0 0 M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
0 0 0 0 0 0 0 0 0 M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
0 0 0 0 0 0 0 0 0 0 M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
0 0 0 0 0 0 0 0 0 0 0 0 M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
0 0 0 0 0 0 0 0 0 0 0 0 0 M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D 0 0 0 0 0 0 0 0 M 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 M 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D 0 M 0 0 0 0 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 M 0 0 0 0 0 0 0 0 0 D;
M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 M 0 0 0 0 0 0 0 0 D;
M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D 0 0 0 0 0 0 0 0;
M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D 0 0 0 0 M 0 0 0;
```

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D 0 M 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 M 0 0 0 0 0 D;
M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D 0 0 0 0;
M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D 0 0;
M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D 0;
M 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 D;
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   1;]';
% Initial state probability vector
P2 = [1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   0 0 0]';
% Calculation
k = 7;
Pk = (T2^k)*P2;
P_4of7 = Pk(36)

%% 2 out of 3 Criterion
% Transition Matrix
T3 = [
M D 0 0;
0 0 M D;
M 0 0 D;
0 0 0 1]';
% Initial state probability vector
P3 = [1 0 0 0]';
% Calculation
k = 3;
Pk = (T3^k)*P3;
P_2of3 = Pk(4)

%% Formula calculations
disp('Formula Calculations:')
M = 4;
N = 7;
temp = zeros(N-M+1,1);
for k = M:N
temp(k) = ( factorial(N) / (factorial(k)*factorial(N-k)) )* D^k * (1-D)
   ^(N-k);
end
P_4of7 = sum(temp)

M = 2;
N = 3;
temp = zeros(N-M+1,1);
for k = M:N
temp(k) = ( factorial(N) / (factorial(k)*factorial(N-k)) )* D^k * (1-D)
   ^(N-k);
end
P_2of3 = sum(temp)
```

C.2 SNR to P_d

Here follows the MATLAB code for transforming SNR from Excel (see Appendix D.2) to P_d and plotting.

```
close all

%% From Blake Chart (Excel)
% Range: 0.1-5.1 km
SNR_500cm2 = [77.91, 65.86, 58.82, 53.82, 49.94, 46.77, 44.09, 41.77,
   39.72, 37.89, 36.23, 34.71, 33.32, 32.03, 30.83, 29.71, 28.65, 27.66,
   26.71, 25.82, 24.97, 24.16, 23.39, 22.64, 21.93, 21.25, 20.59, 19.96,
   19.34, 18.75, 18.18, 17.63, 17.09, 16.57, 16.06, 15.57, 15.09, 14.63,
   14.17, 13.73, 13.30, 12.88, 12.47, 12.07, 11.67, 11.29, 10.91, 10.55,
   10.18, 9.83, 9.49];
SNR_100cm2 = [70.92, 58.87, 51.83, 46.83, 42.95, 39.78, 37.10, 34.78,
   32.73, 30.90, 29.24, 27.72, 26.33, 25.04, 23.84, 22.72, 21.66, 20.67,
   19.72, 18.83, 17.98, 17.17, 16.40, 15.65, 14.94, 14.26, 13.60, 12.97,
   12.35, 11.76, 11.19, 10.64, 10.10, 9.58, 9.07, 8.58, 8.10, 7.64, 7.18,
    6.74, 6.31, 5.89, 5.48, 5.08, 4.68, 4.30, 3.92, 3.56, 3.20, 2.84,
   2.50];
SNR_1cm2 = [50.92, 38.87, 31.83, 26.83, 22.95, 19.78, 17.10, 14.78,
   12.73, 10.90, 9.24, 7.72, 6.33, 5.04, 3.84, 2.72, 1.66, 0.67, -0.28,
   -1.17, -2.02, -2.83, -3.60, -4.35, -5.06, -5.74, -6.40, -7.03, -7.65,
   -8.24, -8.81, -9.36, -9.90, -10.42, -10.93, -11.42, -11.90, -12.36,
   -12.82, -13.26, -13.69, -14.11, -14.52, -14.92, -15.32, -15.70,
   -16.08, -16.44, -16.80, -17.16, -17.50];
snr = [SNR_500cm2; SNR_100cm2; SNR_1cm2];

% Range: 0.1-6.1 km
% SNR_500cm2 = [77.91, 64.21, 56.64, 51.39, 47.36, 44.09, 41.34, 38.96,
   36.87, 35.01, 33.32, 31.78, 30.37, 29.07, 27.85, 26.71, 25.65, 24.64,
   23.69, 22.79, 21.93, 21.11, 20.33, 19.59, 18.87, 18.18, 17.52, 16.88,
   16.26, 15.6, 15.09, 14.54, 14.00, 13.47, 12.96, 12.47, 11.99, 11.52,
   11.06, 10.62, 10.18, 9.76, 9.35, 8.94, 8.55, 8.16, 7.79, 7.42, 7.05,
   6.70, 6.35];
% snr = [SNR_500cm2];

%% Input
start_range = 0.1;
stop_range = 5.1;

pfa = 1e-3;

calc = [3 4];
%% Pd Calculation
if ismember(1,calc)
    beta = sqrt(-2*(log(pfa))) ;
    range = linspace(start_range, stop_range, 51);
    for ii = 1 : size(snr,1)
        for jj = 1 : length(range)
            % In the Marcum Q function, the SNR is defined as a voltage
  ratio, but
            % the rre SNR is a power ratio. To take the square root,
  divide by 20 instead of 10.
            alpha = 10.^((snr(ii, jj)) / 20);
            fun = @(v) v .* exp(-(v.^2 + alpha.^2)/2).* besseli(0, alpha
```

```
    .*v);
                % Add 3dB to SNR to match Skolnik, p2.20
                % Don't add 3dB to SNR to match McDonough, p 261.
                pd(ii, jj) = 1 - integral(fun,0,beta);
            end
        end

        figure, plot(range, pd(1,:),'Color','k','LineWidth',2)
        hold on, plot(range, pd(2,:),'Color','b','LineWidth',2)
        hold on, plot(range, pd(3,:),'Color','r','LineWidth',2)

        whitebg('white')                    % Bakgrunnsfarge inni figur
        set(gcf,'Color','w')                % Bakgrunnsfarge i ramme
        set(gca,'XColor','k')               % X-akse farge
        set(gca,'YColor','k')               % Y-akse farge
        set(get(gca,'Title'),'Color','k')   % Farge p  figur-tittel
        set(gcf,'InvertHardcopy','off')     % Lagre figur i farger slik
    vises (WYSIWYG)
        grid on

        leg = legend('500 cm^2','100 cm^2', '1 cm^2', 'Location', '
    NorthEastOutside');
        htitle = get(leg,'Title');
        set(htitle,'String','RCS','FontName','Calisto MT', 'FontSize', 16)
        title('Probability of Detection (Single Goose)','FontName','Calisto
    MT', 'FontSize', 16)
        xlabel('Range [km]','FontName','Calisto MT', 'FontSize', 18)
        ylabel('$P_D$','FontName','Calisto MT', 'FontSize', 20, 'Interpreter
    ', 'latex')
        set(gca,'FontName','Calisto MT','FontSize',14)
        set(gcf,'Position',[50 200 1000 600])
        axis([0 5.1 0 1])
    end
    %% Pd Calculation 2
    if ismember(2,calc)
        snr_dB=snr;
        snr = 10.^(snr./10);
        for i = 1:size(snr,1)
            pd2(i,:) = pfa.^(1./(snr(i,:)+1));
        end
        range = linspace(start_range, stop_range, 51);

        figure, plot(range, pd2(1,:),'Color','k','LineWidth',2)
        hold on, plot(range, pd2(2,:),'Color','b','LineWidth',2)
        hold on, plot(range, pd2(3,:),'Color','r','LineWidth',2)
        whitebg('white')
        set(gcf,'Color','w')                % Bakgrunnsfarge i ramme
        grid on

        leg = legend('500 cm^2','100 cm^2', '1 cm^2', 'Location', '
    NorthEastOutside');
        htitle = get(leg,'Title');
        set(htitle,'String','RCS','FontName','Calisto MT', 'FontSize', 16)
        title('Probability of Detection (Single Goose)','FontName','Calisto
    MT', 'FontSize', 16)
        xlabel('Range [km]','FontName','Calisto MT', 'FontSize', 18)
        ylabel('$P_D$','FontName','Calisto MT', 'FontSize', 20, 'Interpreter
```

```
    ', 'latex')
      set(gca,'FontName','Calisto MT','FontSize',14)
      set(gcf,'Position',[900 200 1000 600])
      axis([0 5.1 0 1])
end

%% Pd Calculation 3
if ismember(3,calc)
    snr_sparrow_19cm2 = [63.70, 51.66, 44.61, 39.61, 35.74, 32.57,
   29.89, 27.56, 25.52, 23.68, 22.02, 20.51, 19.12, 17.83, 16.63, 15.50,
   14.45, 13.45, 12.51, 11.62, 10.77, 9.96, 9.18, 8.44, 7.73, 7.05, 6.39,
    5.75, 5.14, 4.55, 3.98, 3.42, 2.89, 2.37, 1.86, 1.37, 0.89, 0.43,
   -0.03, -0.47, -0.90, -1.32, -1.73, -2.14, -2.53, -2.91, -3.29, -3.66,
   -4.02, -4.37, -4.72];
    snr_seagull_93cm2 = [70.60, 58.56, 51.51, 46.51, 42.63, 39.46,
   36.78, 34.46, 32.41, 30.58, 28.92, 27.41, 26.02, 24.73, 23.52, 22.40,
   21.35, 20.35, 19.41, 18.52, 17.67, 16.85, 16.08, 15.34, 14.63, 13.94,
   13.29, 12.65, 12.04, 11.45, 10.88, 10.32, 9.79, 9.26, 8.76, 8.27,
   7.79, 7.32, 6.87, 6.43, 6.00, 5.57, 5.16, 4.76, 4.37, 3.98, 3.61,
   3.24, 2.88, 2.53, 2.18];
    snr_goose_221cm2 = [74.36, 62.32, 55.27, 50.27, 46.39, 43.22, 40.54,
    38.22, 36.17, 34.34, 32.68, 31.17, 29.77, 28.48, 27.28, 26.16, 25.10,
    24.11, 23.17, 22.27, 21.42, 20.61, 19.84, 19.10, 18.39, 17.70, 17.04,
    16.41, 15.80, 15.21, 14.64, 14.08, 13.54, 13.02, 12.52, 12.03, 11.55,
    11.08, 10.63, 10.19, 9.75, 9.33, 8.92, 8.52, 8.13, 7.74, 7.37, 7.00,
   6.64, 6.29, 5.94];
    snr_swan_464cm2 = [77.58, 65.54, 58.49, 53.49, 49.61, 46.44, 43.76,
   41.44, 39.39, 37.56, 35.90, 34.39, 33.00, 31.71, 30.51, 29.38, 28.33,
   27.33, 26.39, 25.50, 24.65, 23.84, 23.06, 22.32, 21.61, 20.92, 20.27,
   19.63, 19.02, 18.43, 17.86, 17.30, 16.77, 16.24, 15.74, 15.25, 14.77,
   14.30, 13.85, 13.41, 12.98, 12.55, 12.14, 11.74, 11.35, 10.96, 10.59,
   10.22, 9.86, 9.51, 9.16];
    snr = [snr_sparrow_19cm2; snr_seagull_93cm2; snr_goose_221cm2;
   snr_swan_464cm2];

    snr_dB=snr;
    snr = 10.^(snr./10);
    for i = 1:size(snr,1)
        pd3(i,:) = pfa.^(1./(snr(i,:)+1));
    end
    range = linspace(start_range, stop_range, 51);

    figure, plot(range, pd3(1,:),'Color','r','LineWidth',2)
    hold on, plot(range, pd3(2,:),'Color','b','LineWidth',2)
    hold on, plot(range, pd3(3,:),'Color','g','LineWidth',2)
    hold on, plot(range, pd3(4,:),'Color','k','LineWidth',2)
    whitebg('white')
    set(gcf,'Color','w')                % Bakgrunnsfarge i ramme
    grid on

    leg = legend('Sparrow (19 cm^2)','Seagull  (93 cm^2)', 'Goose   (221
   cm^2)', 'Swan    (464 cm^2)', 'Location', 'NorthEastOutside');
    htitle = get(leg,'Title');
    set(htitle,'String','Species (SWEM RCS)','FontName','Calisto MT', '
   FontSize', 16)
    title('Probability of Detection','FontName','Calisto MT', 'FontSize'
   , 16)
```

```
    xlabel('Range [km]','FontName','Calisto MT', 'FontSize', 18)
    ylabel('$P_D$','FontName','Calisto MT', 'FontSize', 20, 'Interpreter
', 'latex')
    set(gca,'FontName','Calisto MT','FontSize',14)
    set(gcf,'Position',[900 200 1000 500])
    axis([0 5.1 0 1])

end

%% Plot Pd from Excel
if ismember(4,calc)
    %range er 51 sampler fra 0,1 til 7 km
    pd_sparrow_19cm2 = [1.0000, 0.9999, 0.9994, 0.9979, 0.9947, 0.9886,
  0.9784, 0.9627, 0.9402, 0.9095, 0.8699, 0.8207, 0.7624, 0.6960,
  0.6234, 0.5474, 0.4708, 0.3967, 0.3277, 0.2658, 0.2121, 0.1670,
  0.1300, 0.1006, 0.0775, 0.0596, 0.0460, 0.0357, 0.0279, 0.0220,
  0.0176, 0.0142, 0.0116, 0.0096, 0.0081, 0.0069, 0.0059, 0.0051,
  0.0045, 0.0040, 0.0036, 0.0033, 0.0030, 0.0027, 0.0025, 0.0024,
  0.0022, 0.0021, 0.0020, 0.0019, 0.0018];
    pd_seagull_93cm2 = [1.0000, 1.0000, 0.9999, 0.9996, 0.9989, 0.9977,
  0.9955, 0.9922, 0.9874, 0.9806, 0.9715, 0.9595, 0.9444, 0.9256,
  0.9029, 0.8761, 0.8449, 0.8095, 0.7699, 0.7266, 0.6800, 0.6309,
  0.5800, 0.5283, 0.4767, 0.4261, 0.3774, 0.3313, 0.2884, 0.2490,
  0.2136, 0.1820, 0.1543, 0.1302, 0.1095, 0.0919, 0.0770, 0.0645,
  0.0541, 0.0455, 0.0383, 0.0323, 0.0274, 0.0234, 0.0200, 0.0172,
  0.0149, 0.0130, 0.0113, 0.0100, 0.0088];
    pd_goose_221cm2 = [1.0000, 1.0000, 0.9999, 0.9998, 0.9995, 0.9990,
  0.9981, 0.9967, 0.9947, 0.9918, 0.9879, 0.9827, 0.9761, 0.9678,
  0.9576, 0.9453, 0.9306, 0.9134, 0.8936, 0.8710, 0.8456, 0.8174,
  0.7865, 0.7530, 0.7172, 0.6793, 0.6398, 0.5991, 0.5575, 0.5157,
  0.4741, 0.4332, 0.3935, 0.3553, 0.3190, 0.2848, 0.2530, 0.2237,
  0.1970, 0.1728, 0.1511, 0.1317, 0.1146, 0.0995, 0.0863, 0.0748,
  0.0649, 0.0563, 0.0488, 0.0425, 0.0370];
    pd_swan_464cm2 = [1.0000, 1.0000, 1.0000, 0.9999, 0.9998, 0.9995,
  0.9991, 0.9984, 0.9975, 0.9961, 0.9942, 0.9917, 0.9885, 0.9845,
  0.9795, 0.9734, 0.9661, 0.9575, 0.9474, 0.9357, 0.9223, 0.9071,
  0.8900, 0.8710, 0.8501, 0.8272, 0.8024, 0.7758, 0.7474, 0.7174,
  0.6861, 0.6535, 0.6200, 0.5857, 0.5511, 0.5163, 0.4816, 0.4473,
  0.4138, 0.3811, 0.3497, 0.3195, 0.2909, 0.2639, 0.2386, 0.2150,
  0.1932, 0.1732, 0.1549, 0.1382, 0.1232];

    pd4 =[pd_sparrow_19cm2; pd_seagull_93cm2; pd_goose_221cm2;
  pd_swan_464cm2;];
    range = linspace(0.1,7,51);

    figure,  plot(range, pd4(1,:),'Color','r','LineWidth',2)
    hold on, plot(range, pd4(2,:),'Color','b','LineWidth',2)
    hold on, plot(range, pd4(3,:),'Color','g','LineWidth',2)
    hold on, plot(range, pd4(4,:),'Color','k','LineWidth',2)
    whitebg('white')
    set(gcf,'Color','w')                % Bakgrunnsfarge i ramme
    grid on

    leg = legend('Sparrow (19 cm^2)','Seagull  (93 cm^2)', 'Goose   (221
 cm^2)', 'Swan    (464 cm^2)', 'Location', 'NorthEastOutside');
    htitle = get(leg,'Title');
    set(htitle,'String','Species (SWEM RCS)','FontName','Calisto MT', '
```

```
    FontSize', 16)
      title('Probability of Detection','FontName','Calisto MT', 'FontSize'
    , 16)
144   xlabel('Range [km]','FontName','Calisto MT', 'FontSize', 18)
      ylabel('$P_D$','FontName','Calisto MT', 'FontSize', 20, 'Interpreter
    ', 'latex')
146   set(gca,'FontName','Calisto MT','FontSize',14)
      set(gcf,'Position',[900 200 1000 500])
148   axis([0 7 0 1])

150 end
    clearvars
152
```

D P_d calculations

D.1 CARPET 2 settings

Troposcatter	off
Attenuation	off
Free space	on
Surface based Duct	off
Evaporation Duct	off
Cosmic Noise	off
Sea Clutter	on
Land Clutter	off
Constant Gamma	off
Rain	off
Chaff	off
Barrage Jamming	off
Responsive Jamming	off
Phase Noise	off
Doppler Processing	on
Timing Jitter	off
Rotating	off
Tracking	none
Atmosph. Pressure	1020 hPa
Humidity	70 %
Air Temperature	15 C
Water Temperature	10 C
Wind Force	3 Bfrt
Wind Direction	0 deg
K-Factor	1.33
Refractivity	328 Nunit
Sea State	2
Salinity	35 prom
Evap. Duct Height	10 m
Surf. Duct Height	100 m
Soil	average
Water Content Soil	60 %
Std Surface Height	0.1 m
Galactic Noise	average

Land Reflectivity	-38 dB
Rainfall Rate	4.0 mm/hr
Chaff Density	30.0 g/km^3
Min Range Rain	5 km
Max Range Rain	15 km
Max Altitude Rain	3 km
Jammer Power	10 kW
Antenna Gain Jam.	12.0 dBi
Bandw Barrage Mode	600 MHz
Bandw Respons Mode	10 MHz
Jammer Range	200 km
Jammer Altitude	3 m
Carrier Frequency	9410 MHz
Peak Power	25 kW
Pulse Length	0.15 μs
Inst. Bandwidth	6.7 MHz
PRF	3 kHz
Pulse Bursts	1
Transmitted Pulses	19
Transmitter Losses	2.0 dB
White Noise Level	-120 dB/Hz
Colored Noise Lvl	-40 dBc
Cut-off Frequency	1 Hz
Timing Jitter TX	0.1 ns
Antenna Type	rect.
Vertical Illum.	parab.
Azimuth Beamwidth	1.0 °
Elevation Beamw.	20 °
Transmit Gain	31.5 dBi
Polarization	H
Tracking	none
Receive Gain	31.5 dBi
Beamshape Loss	1 dB
Antenna Tilt	0 °
Radar Height	20 m
Azimuth Sidelobes	-25 dB
Frame Time	2 s
MTI	on
Doppler Bank	off
Taper DFB	Blackm.
Noise Figure	3.0 dB
Receiver Losses	1 dB
False Alarm Prob.	1e-003
Fill Pulses	0

Timing Jitter RX	0.1 ns
Target RCS	0.010 m^2
Target Velocity	15 m/s
Target Range	4 km
Target Altitude	: 50 m
Swerling Case	1
Minimum Plot range	0.1 km
Maximum Plot range	6 km
Minimum Plot vty	0 m/s
Maximum Plot vty	25 m/s
Minimum Plot hght	1 m
Maximum Plot hght	1000 m
Range Axis	linear
Performance Prmtr	range

D.2 Manual P_d calculations

RADAR RANGE

		Num value:	Unit:		Decibels:	
Probability of detection:	Pd:	0,7				
Probability of false alarm	Pfa:	1,00E-03				
Target (Sw 1 / Sw 0)						
Target cross section	σ:	0,0221	sqm		-16,56	dB
Radar Range Equation:						
Pulse power	Pt:	2,50E+04	Watt		43,98	dB
Pulse width	τ:	1,50E-07	sec		-68,24	dB
Transmit gain	Gt:	1412,54			31,50	dB
Receive gain	Gr:	1412,54			31,50	dB
Target cross section	σ	0,0221	sqm		-16,56	dB
Frequency	Fs:	9,41E+09	Hz		99,74	dB
System noise temperature	Ts:	1417,99	K		31,52	dB
Basic detectability factor	D(n,ne):	2,52			4,02	dB
Beamshape loss	Lp:	1,45		(Default)	1,60	dB
Transmit line loss:	Lt:	1,58			2,00	dB
Matching factor	M:	1,58			2,00	dB
Miscellaneous processing losses	Lx:	2,00			3,00	dB
Free space range	Ro:	3,92	km			
	(Ro:	2,12	nmi)			
Pattern Propagation Factor	F:	1		(Default)	0,00	dB
Range including F		3,92	km			
Atmospheric attenuation:	Lα:	1,02			0,09	dB
Range including F and Lα:	**R:**	**3,90**	**km**			
	(R:	**2,11**	**nmi)**			

Figure 7.7: Excel "Radar Range" spreadsheet.

RADAR PARAMETERS

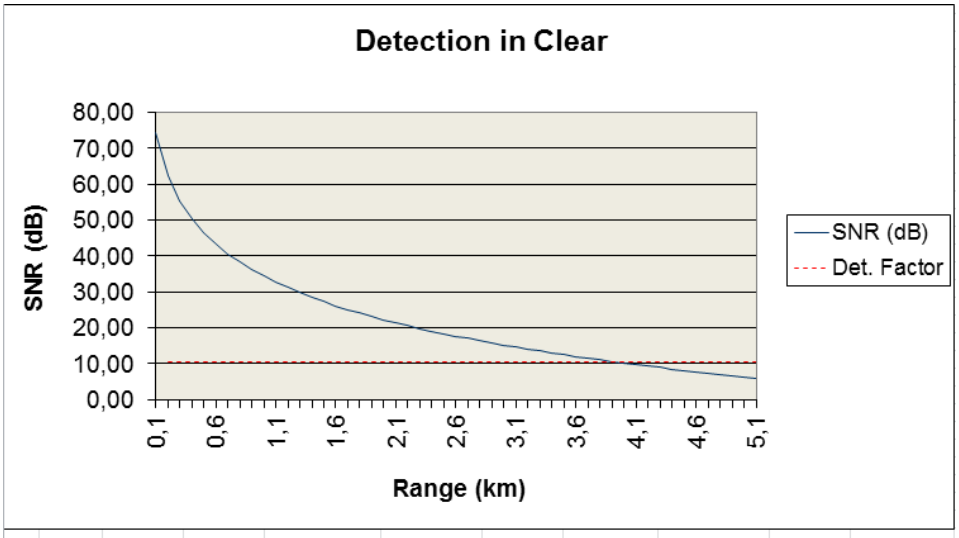
		Num value:	Unit:			Decibels:	
Transmitter and waveform:							
Pulse Power	Pt:	2,50E+04	Watt			43,98	dB
Pulse width	τ:	1,50E-07	sek			-68,24	dB
Frequency	Fs:	9,41E+09	Hz			99,74	dB
Number of pulses integrated	n:	18					
Frequency diversity:		1					
Transmission line loss:							
1st loss factor:					2,0	dB	
2nd loss factor:						dB	
Total Tx line loss	Lt:	1,58				2,00	dB
Antenna:							
Receive gain	Gr:	1412,54				31,5	dB
Transmit gain	Gt	1412,54				31,5	dB
Pysical temperature	Tta:	290	K	(Default)		24,62	dB
Ohmic loss	La:	1,12				0,50	dB
Antenna height	hr:	18	m				
Beam elevation	θt	0	deg				
Receiver:							
Transmission line temp	Tt:	290	K	(Default)			
Transmission line loss:							
1st loss factor					2,25		
2nd loss factor					0,0		
Total RX line loss	Lr:	1,68				2,3	dB
Low Noise Amplifier:							
Gain		223,87				23,5	dB
Noise Factor	NF:	3,16				5,0	dB

Figure 7.8: Excel ”Radar Parameters” spreadsheet.

SYSTEM NOISE TEMPERATURE

Antenna temperature:				
Attenuation koefficient (sea level)	kα:	0,0229	dB/km	
Athmospheric loss (500 km)	LA:	2,65	dB	
Athmospheric noise temp	TA:	132,44	K	
Cosmic brightness noise temp	Tc/LA:	1,49	K	
Solar noise temp	αsTs/LA:	0,27	K	
Total sky noise temp(lossless ant)	Ta':	134,20	K	
Ground noise temp	Tg:	36,00	K	(Default)
Total antenna noise temp	Ta:	168,43	K	
Transmission line noise temp	Tr:	196,9	K	
Receiver noise temp	LrTe:	1052,71	K	
System Noise Temperature	**Ts:**	**1417,99**	K	

Figure 7.9: Excel ”Noise Temperature” spreadsheet.



Start range:	0,1	km	Target size:	0,0221	sqm	No of pulses:	18
Stop range:	5,1	km	Target elevation	0	deg	Diversity:	1
			Antenna height:	18	m	D(n,ne)MLpLx	10,62 dB
						Detection range:	3,90 km

Intermediate calculations:		Num value:		Decibels:	
	Pulsepower:	2,50E+04	W	43,98	dBW
	Pulsewidth:	1,50E-07	s	-68,24	dBs
	Wavelength:	3,19E-02	m	-14,96	dBm
	Transmit gain:	1412,54		31,5	dB
	Receive gain:	1412,54		31,5	dB
	Target RCS	0,0221	sqm	-16,55607726	dBsqm
	No (kTs)	1,95683E-20	W/Hz	-197,08	dBW/Hz
	Transm. Loss (Lt):	1,58		2	dB
	Pattern prop factor	1		0	dB

Figure 7.10: Excel ”Detection in the clear” spreadsheet.

Range (km)		Lα		SNR	Range (km)	SNR (dB)	Det. Factor
0	0,1	0,00	1,001	27293970	0,1	74,36	10,62
1	0,2	0,00	1,001	1704940,8	0,2	62,32	10,62
2	0,3	0,01	1,002	336594,38	0,3	55,27	10,62
3	0,4	0,01	1,002	106442,36	0,4	50,27	10,62
4	0,5	0,01	1,003	43574,963	0,5	46,39	10,62
5	0,6	0,01	1,003	21002,676	0,6	43,22	10,62
6	0,7	0,02	1,004	11330,526	0,7	40,54	10,62
7	0,8	0,02	1,004	6638,1163	0,8	38,22	10,62
8	0,9	0,02	1,005	4141,8785	0,9	36,17	10,62
9	1	0,02	1,005	2716,0013	1	34,34	10,62
10	1,1	0,03	1,006	1854,0516	1,1	32,68	10,62
11	1,2	0,03	1,007	1308,3688	1,2	31,17	10,62
12	1,3	0,03	1,007	949,38929	1,3	29,77	10,62
13	1,4	0,03	1,008	705,45316	1,4	28,48	10,62
14	1,5	0,04	1,008	535,02969	1,5	27,28	10,62
15	1,6	0,04	1,009	413,07183	1,6	26,16	10,62
16	1,7	0,04	1,009	323,94585	1,7	25,10	10,62
17	1,8	0,04	1,010	257,59693	1,8	24,11	10,62
18	1,9	0,05	1,010	207,38574	1,9	23,17	10,62
19	2	0,05	1,011	168,82467	2	22,27	10,62
20	2,1	0,05	1,012	138,81657	2,1	21,42	10,62
21	2,2	0,05	1,012	115,18352	2,2	20,61	10,62
22	2,3	0,05	1,013	96,367889	2,3	19,84	10,62
23	2,4	0,06	1,013	81,238393	2,4	19,10	10,62
24	2,5	0,06	1,014	68,961844	2,5	18,39	10,62
25	2,6	0,06	1,014	58,916659	2,6	17,70	10,62
26	2,7	0,06	1,015	50,633629	2,7	17,04	10,62
27	2,8	0,07	1,015	43,754689	2,8	16,41	10,62
28	2,9	0,07	1,016	38,003828	2,9	15,80	10,62
29	3	0,07	1,017	33,166293	3	15,21	10,62
30	3,1	0,07	1,017	29,073532	3,1	14,63	10,62
31	3,2	0,08	1,018	25,59218	3,2	14,08	10,62
32	3,3	0,08	1,018	22,615909	3,3	13,54	10,62
33	3,4	0,08	1,019	20,059346	3,4	13,02	10,62
34	3,5	0,08	1,019	17,85348	3,5	12,52	10,62
35	3,6	0,09	1,020	15,942179	3,6	12,03	10,62
36	3,7	0,09	1,020	14,279514	3,7	11,55	10,62
37	3,8	0,09	1,021	12,827693	3,8	11,08	10,62
38	3,9	0,09	1,022	11,555458	3,9	10,63	10,62
39	4	0,09	1,022	10,436822	4	10,19	10,62
40	4,1	0,10	1,023	9,4500793	4,1	9,75	10,62
41	4,2	0,10	1,023	8,5770189	4,2	9,33	10,62
42	4,3	0,10	1,024	7,8022948	4,3	8,92	10,62
43	4,4	0,10	1,024	7,1129233	4,4	8,52	10,62
44	4,5	0,11	1,025	6,4978762	4,5	8,13	10,62
45	4,6	0,11	1,025	5,9477516	4,6	7,74	10,62
46	4,7	0,11	1,026	5,4545054	4,7	7,37	10,62
47	4,8	0,11	1,027	5,0112322	4,8	7,00	10,62
48	4,9	0,12	1,027	4,6119844	4,9	6,64	10,62
49	5	0,12	1,028	4,2516235	5	6,29	10,62
50	5,1	0,12	1,028	3,9256972	5,1	5,94	10,62

Figure 7.11: Excel ”Signal to Noise power Ratio” spreadsheet.

E Digital Appendix

This section lists the contents of the digital appendix that follows this thesis as a .zip-file.

- Videos
  - Radar video of Figure 5.1
  - Radar video of Figure 5.2
  - Radar video of Figure 5.5
  - Radar video of Figure 5.6
  - Radar video of Figure 5.14
  - Radar video of Figure 5.15
  - Radar video of Figure 5.16
  - Radar video of Figure 5.17
  - Radar video of Figure 5.19
- MATLAB code of the BFPD Tracker
  - *Batch_Tracking_Processor.m*
    The BFPD Tracker processor that controls and runs multi-batch tracking.
  - *BirdFlightPathDetector_v1_1_JOB.m*
    The function that performs all tracking operations within a single batch.
  - *CA_CFAR_v1_2.m*
    The provided function that performs Cell Averaging CFAR in the MTD processor.
  - *CACM_CFAR_v1.m*
    The function that performs Cell Averaging Clutter Map CFAR in the MTD processor.
  - *CNav_v10_coh_PPI_JOB.m*
    The provided MTD processor, modified to deal with Clutter Map and other implementations made by the author.
  - *EarlyWarning.m*
    The function that analyzes the output of the BFPD Tracker (Batch_Tracking_Processor.m).
  - *File_Processor2.m*
    The processor that performs MTD processing of large batches files (scans).
  - *Find_birdtracks.m*
    The function that performs the specific tracking of BFPs within a single batch.
  - *GeneratePlotmatrix.m*
    The function that performs the simplified plot extraction.
  - *GeneratePlotmatrix_TwoPass.m*
    The function that performs the Two-Pass CCL plot extraction.

- *GenerateSimData.m*
  The script that models the movement of a simulated target, synthesizes it and generates coherent unprocessed radar signal files that contain the simulated target.
- *OS_CFAR_v1_1.m*
  The provided function that performs Ordered Statistic-CFAR in the MTD processor.
- *OSCM_CFAR_v1.m*
  The function that performs Ordered Statistic Clutter Map-CFAR in the MTD processor.
- *ResultsAnalysis.m*
  The script that performs a range of analytical tasks on the results as saved by the BFPD Tracker.
- *SimulationAnalysis.m*
  The script that performs a range of analytical tasks on the results from simulations as saved by the BFPD Tracker.

Note: Detailed description and documentation can be found within the files themselves. However, the code commentary is sometimes written in Norwegian.

Glossary

Batch	=	A set of N consecutive scans.
Batch association	=	An identification of two confirmed tracks, located in two consecutive batches, that fit the criterias for assuming that the two tracks originate from the same target.
Bird Flight Path (BFP)	=	A set of tracks in consecutive batches that are connected by batch associations.
Confirmed tracks	=	Tracks within a single batch that satisfy the established track length and quality requirements.
Detection	=	A radar spatial resolution cell with a level of amplitude larger than the cell's threshold in one of the cell's velocity channels.
Detectionheap	=	A collection of adjoining detections that show 4-connectivity in the Detection Matrix.
False alarm	=	A detection that does not originate from the presence of a bird.
False plots/tracks	=	Measurements, plots or tracks that according to all available information and by best assessment ability possible seem to originate from anything other than birds.
False warning	=	A warning signal that is initiated by false plots/tracks.
Merged confirmed track	=	A confirmed track that is made by combining multiple closlely separated confirmed tracks into a single track where there is only observed a single target in the MTD output.
Merged batch associations	=	An identification of two merged confirmed tracks, located in two consecutive batches, that fit the criterias for assuming that the two tracks originate from the same target.
Plot	=	A measurement with 2D coordinates, extracted from a whole, or part of a detectionheap. Includes information such as Doppler and Time of Observation.
Reliability	=	Probability of trueness, or the ability to be depended on.
Target dynamics model	=	A set of rules, that describe the movement and restrictions in maneuvering that is expected of the targets of interest (birds that can cause potential hazardous birdstrike situations).
Tentative tracks	=	All tracks of varying score/quality that are found within a single batch.

The glossary is continued on the next page

Track	=	A set of plots within a batch that is found to fit a spatial and temporal pattern that matches that of the target dynamics model.
True plots/tracks	=	Measurements, plots or tracks that according to all available information and by the best assessment ability possible seem to originate from birds.
True warning	=	A warning signal that is initiated by true plots/tracks.
Warning	=	A report, or activated signal, that indicates the identification of a potential hazardous birdsstrike situation resulting from a BFP.