Problem description

Background

Multitarget tracking is a key ingredient in collision avidance system for autonomous vehicles. Multi-frame tracking methods are commonly acknowledged as gold standards for multi-target tracking. The purpose of this master thesis is to develop a complete multi-frame system for autonomous ships, based on sensor inputs from radar and the Automatic Identification System (AIS).

Proposed tasks

The following task are proposed for this thesis:

- Extend an integer-linear-programming (ILP) based tracking method with suitable algorithms for track initiation and track management
- Develop a framework for fusion between radar tracks and AIS tracks
- Develop alternatives to N-scan pruning in order to enhance the computational efficiency of the tracking method
- Implement the tracking system in Python and/or C++
- Test the tracking system on simulated data

Autosea

This thesis is associated with the AUTOSEA project, which is collaborative research project between NTNU, DNV GL, Kongsberg Maritime and Maritime Robitics, focused on achieving world-leading competence and knowledge in the design and verification of methods and systems for sensor fusion and collision avoid- ance for ASVs. The project has access to supervision and physical test platforms through our industry partners.

Preface

This Master thesis is written at Norwegian University of Science and Technology (NTNU) as final work in the Engineering Cybernetics study program with specialisation in Robotics and Vessel control and was carried out during the spring of 2017.

I would like to thank the people that have made this thesis a reality. First of all, my supervisor associate professor Edmund F. Brekke for giving me freedom and trust to seek out my own solution and guidance with constructive feedback. Next, a great thanks to my co-supervisors Ph.D. students Erik F. Wilthil and Andreas L. Flåten for helpful discussion and giving me access to our computing server 'Syn'. I would also thank Paal Kristian for a beeing a great companion at the office.

Erik Liland
Trondheim, 5. June 2017

Abstract

To enable autonomous sea vessel's safe voyage, a real time situational awareness system is required. A tracking system incorporating both radar and AIS sensor data is preferred in maritime situations since radar and AIS have different strong and weak properties. A multi-sensor multi-frame multitarget tracking system based on radar and AIS measurements from on vessel sensors is developed. The system consist of a two main parts, a logic based initialization algorithm and a Track Oriented Multi Hypothesis Tracker (TOMHT). This tracking system is demonstrated on simulated multitarget data with different tuning settings, external environment and AIS configurations.

The track loss was improved from X% to X% when comparing pure radar measurements with full class A AIS coverage and to X% with class B AIS.

Sammendrag

Løysinga er 42.

Contents

List of Figures

List of Tables

Glossary

VHF The frequency range between 30 and 300 MHz.

Algorithm Step by step operations to be performed.

Clutter Noise in the form of false measurements where the amount is assumed Poisson distributed.

Coalescence Come together to form one whole.

COLREGS Convention on the International Regulations for Preventing Collisions at Sea.

Gate An area in which a track expects and approves new measurement to associate with itself.

Gross tonnage A measurement of a ship's overall internal volume.

Measurement A point in the measurement space where something is detected.

Measurement noise Also called observation noise, which is noise that affects the accuracy of the measurement not the existence. White Gaussian with zero mean and covariance R.

Nautical Mile Length used in maritime navigation. Equals 1 minute of latitude (1852 meters).

NMEA Communication protocol used between electronic maritime equipment, based on RS-422.

Python An open-source programming language.

Radar Acronym for Radio Detection And Ranging. A device that uses radio waves to measure distance and bearing to other objects.

RS-232 Serial single ended communication standard.

RS-422 Serial differential communication standard.

Scan A procedure which measures the entire area of coverage of the system.

Score A measure of the goodness of a measurement-to-track association.

SOLAS The International Convention for the Safety of Life at Sea.

Solver A program that solves optimization problems.

System noise Also called process noise, which is noise in the model behaviour. This noise compensates for the uncertainty and non-modelled dynamics of the true system.

Target An actual object which the system is trying to track.

Track forest A forest of track hypothesis trees.

Track hypothesis A leaf node with its predecessors.

Track hypothesis tree An acyclic graph spanning from the root node of each target where each node is a track hypothesis.

Tracking The process of initiating, maintaining and terminating tracks from measurements.

Acronyms

P_D Probability of detection

AIS Automatic Identification System

ASV Autonomous Surface Vessel

BFS Breath First Search

CAS Collision Avoidance System

CDF Cumulative Distribution Function

CPA Closest point of Approach

CPHD Cardinalized Probability Hypothesis Density

CSTDMA Carrier Sense Time Division Multiple Access

DAG Directed Acyclic Graph

DFS Depth First Search

DNV GL Det Norske Veritas Germanischer Lloyd

FISST Finite Set Statistic

HOMHT Hypothesis Oriented Multi Hypothesis Tracker

ILP Integer Linear Programming

IMO International Maritime Organization

JPDAF Joint Probabilistic Data Association Filter

MHT Multi Hypothesis Tracking

MILP Mixed Integer Linear Programming

MMSI Mobile Maritime Safety Identity

MUNIN Maritime Unmanned Navigation through Intelligence in Networks

NIS Normalized Innovation Squared

NM Nautical Mile

NNF Nearest Neighbour Filter

NTNU Norwegian University of Science and Technology

PDAF Probabilistic Data Association Filter

PDF Probability Density Function

PHD Probability Hypothesis Density

RADAR RAdio Detection And Ranging

RAM Random Access Memory

RFS Random Finite Set

RPM Rotations Per Minute

SAR Synthetic Aperture Radar

SOTDMA Self Organized Time Division Multiple Access

SSD Solid State Storage

TDMA Time Division Multiple Access

TOMHT Track Oriented Multi Hypothesis Tracker

UTC Coordinated Universal Time

UTM Universal Transverse Mercator coordinate system

VHF Very High Frequency

VTS Vessel Traffic Service

Nomenclature

$\bar{\mathbf{x}}$	Predicted state
$\hat{\mathbf{x}}$	Filtered state
λ_{ν}	Poisson spatial density of the number of new measurements
λ_{ϕ}	Poisson spatial density of the number of false measurements
λ_{AIS}	Possion spatial density of the number of 'extraneous' AIS measurements
λ_{ex}	Total spatial density of the number of "extraneous" measurements
cNLLR	Cumulative Negative Log Likelihood Ratio
NLLR	Negative Log Likelihood Ratio
μ_a	Average true converted measurement bias
$\bar{\boldsymbol{P}}$	Predicted state covariance
$\hat{\boldsymbol{P}}$	Filtered state covariance
$\boldsymbol{\Phi}$	State transition matrix
$\boldsymbol{H}$	State observation matrix
$\boldsymbol{K}$	Kalman gain
$\boldsymbol{Q}$	Process noise covariance matrix

$\boldsymbol{R}$	Measurement covariance matrix
$\boldsymbol{S}$	Residual covariance
σ_r	Range measurement standard deviation
σ_θ	bearing measurement standard deviation
θ_m	Measured angle
$\hat{\mathbf{z}}$	Predicted measurement
$\boldsymbol{\tau}$	Binary vector where the selected track hypotheses are 1
$\tilde{\mathbf{x}}$	Measurement innovation for measurement
$\mathbf{v}$	Observation noise
$\mathbf{w}$	Process noise
$\mathbf{x}$	State vector
$\mathbf{z}$	Measurement
i	Measurement index
j	Target index
k	Time index
l	Hypothesis index
m_k	Number of measurements in scan k
ms	Millisecond
N	Number of scans to keep in track tree
r_m	Measured range
t	Time
t_k	Time at time index

Chapter 1

Introduction

1.1 Motivation

Automation- and control technology have throughout the history been a crucial part of reliving humans from for instance dangerous, exhaustive, repetitive or boring work. Examples of this is automation and robotics in production facilities, remotely operated vehicles for working and exploring the deep sea, disarming explosives and explore space. The level of self control varies from remotely controlled to self sensing and planning without human interaction.

The early motivation for automation was probably, and in many situations still are, to improve speed, quality and consistency, which all tends to lead to better economics. With a still decreasing threshold for automating processes, more focus is applied on easing the burden on people, either by combining robotics and humans in the same operation, or by fully automate the task. These jobs are typically repetitive, dangerous or both.

Although humans are capable of both self improving and easily adapting to new tasks, they will always have good and bad days, performing the same task slightly different or be bored and unfocused. These are all aspects that leads to inconsistency and errors, which may not be a problem in a production environment with quality inspections, though inconvenient, but can be fatal in critical applications.

There also exists several places where humans and automated system work together to exploit both strengths, for instance in aviation where the pilots are always present in the cockpit, but the autopilot are flying the plane most of the time. This gives the pilots freedom from a very static and repetitive task where a human error could have fatal consequences. This symbiosis is somewhat similar to the workload on the bridge of commercial vessels, where the autopilot steering the ship most of the time, while the crew is setting the course.

For vessels that do very repetitive routes and jobs, like ferries and short domestic cargo transport, the mental fatigue on the crew can be an issue. Because of the need for crew in emergency situations, customer service and ship maintenance, larger ferries would still need crew if their navigation were to be automated. The vessels could be controlled by an automated route planning- and Collision Avoidance System (CAS). The control system would never be tired, bored, intoxicated or distracted in the same ways as humans can. This is some aspects that make Autonomous Surface Vessels (ASVs) applicable for certain use cases.

The sensor and control system needed for safe automation of any vessel is large and complex, and requires several layers of fault barriers to prevent system errors for spreading, and the ability to self monitor its own performance. The control system would know its own position and desired position, it would have access to maps to make a route, a CAS to deviate from its planned route to act in accordance with the rules at sea (COLREGS) based on real-time situation information from the sensors on the vessel.

For ASVs to be a viable alternative to human guided ships, the potential savings must be more than marginal, and the control system must be at least as safe as a human operated vessel. The state-of-the-art is not at this point yet, but recent initiatives by large corporations in development in ASVs and the regulation of a dedicated test area for ASVs in Trondheimsfjorden in Norway are just two examples on the direction this technology is headed.

The worlds first autonomous ferry might be between Ravnkloa and Vestre kanalkai in Trondheim or between Breivik and Larvik in Porsgrunn. The first project is a collaboration between Norwegian University of Science and Technology (NTNU) and Det Norske Veritas Germanischer Lloyd (DNV GL), with the aim to develop a small autonomous battery powered passenger and bike-cycle ferry as an alternative to a bride over a canal. The second project is a partnership between YARA and Kongsberg Maritime, with the goal of having the vessel fully autonomous from 2020.

Another indicator of the momentum autonomous surface vessels have is the Maritime Unmanned Navigation through Intelligence in Networks (MUNIN) project, which is a collaborate project between several European companies and research institutes, partially funded by the European Commission. The project aims at developing and verifying concept of autonomous vessels with remote control from onshore control stations.

This work is focused on the sensor fusion which generates a real time data stream into the control system, enabling situational awareness and the foundation for predictive CAS like [1].

1.2 Previous work

This work is based on a pre-master project executed autumn 2016 [2, url=true]. In this project, it was shown that several off-the-shelf Integer Linear Programming (ILP) solvers was capable of solving the data association optimization problem in a single sensor Track Oriented Multi Hypothesis Tracker (TOMHT). It also showed that under good to moderate conditions, the performance return when increasing multi-scan window more than a relative low threshold, was very low.

1.3 Outline of the thesis

Chapter 2 provides an introduction to the sensor systems used in this work, as well as some of the different flavours of tracking methods that exist. Chapter 3 gives a brief introduction to radar and Automatic Identification System (AIS) as systems, with focus on their pre-processing requirements prior to the MHT module. Chapter 4 presents an overview of the complete measurement-to-track system and an in-depth explanation of the fused radar and AIS TOMHT tracking system. Chapter 5 presents the different scenarios that are used in performance evaluation of the tracker and the results of the simulated scenarios. A discussion of the results and evaluation of the performance with respect to safety at sea is presented in Chapter 6. Suggestion for future work is presented in Chapter 7, followed by a conclusion in Chapter 8.

Chapter 2

Theoretical Background

2.1 Radar

2.1.1 Overview

RAdio Detection And Ranging (RADAR) is a detection technology that uses radio waves to observe stationary and moving objects. A transmitter sends out radio waves and a receiver is waiting for reflected echo's from objects, the time the echo is delayed determines the distance to the object. The transmitter and receiver will in many situations be in the same location, can be both stationary and mobile and fixed or rotating orientation. Depending on frequency, a radar can observe solid objects like air-crafts, ships, terrain, road vehicles and less solid objects like people and weather formations.

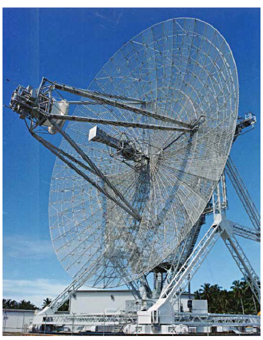

Figure 2.1: Fixed radar antenna

Figure 2.2: Rotating radar antenna

Figure 2.3: Maritime radar antenna

2.1.2 History

The fist implementation of an instrument that were able to detect the presence of distance metallic objects by radio waves was done by Christian Hülsmeyer in 1904. His invention did not measure the distance to objects, but whether there was an object in the direction of the instrument. The radar as we know it today was introduced in the mid to late 1930's, with world war two triggering a research to improve the still immature technology to be used in military applications. After the war, the technology matured and where put in use in several civil applications, where air traffic control, maritime safety and weather monitoring is the most common.

2.1.3 Principles

The electromagnetic waves that a radar emits travels at the speed of light in air and vacuum. It reflects back when there is a change in the density of the medium it is travelling through, which is what happens when radio waves hit targets. Electrically conductive materials tends to be good reflectors, since they have a very different atomic density than air. On the other hand, materials with poor conductivity, and also some magnetic materials, then to absorb radio waves good. Like light, there is many ways an incoming radio wave can be reflected, primarily dependent on the geometry of the target. A corner with angles less than 180° will reflect the incoming radio waves directly back to the sender, and is a good thing on targets that want to be visible on a radar. This principle are the basis for radar reflectors commonly used to boos the radar signature on smaller vessels, see Figure 2.4. The opposite is used on targets that try to minimize their radar signature, and is the reason why stealth vessels and aircraft are tiled by flat areas.

Figure 2.5: USS Zumwalt

Figure 2.6: KNM Gnist

Figure 2.7: F177 Nighthawk

2.2 AIS

The Automatic Identification System (AIS) is a maritime safety and information system primarily designed for collision avoidance. AIS works by broadcasting messages on the Very High Frequency (VHF) band at irregular intervals with information on the vessels. AIS transceivers are required on international voyaging vessels over 300 gross tonnage, and on all passenger vessels. AIS signals are received at bother vessels and shore stations for use in Vessel Traffic Service (VTS) stations, open tracking databases like www.marinetraffic.com, fleet-monitoring and search and rescue. Since the AIS messages contains position, course and speed, AIS tracks can be overlaid on a map in a chart plotter or on top of a radar image, giving the operator two sensors to verify each other.

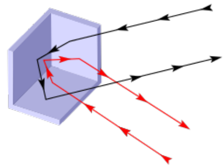

Figure 2.4: Corner reflector

2.2.1 History

AIS was designed and developed by technical comities in the International Maritime Organization (IMO). Its objective was to enhance vessels safety and efficiency by increasing their ability to see and identify other vessels. The main motivation for adopting AIS was its independence of humans in operation, since it automatically identifies other vessels and displays the information on the navigational system on the bridge. It also enables automatic calculation of Closest point of Approach (CPA) and time until CPA, in which the navigation system could alarm the bridge on incoming traffic on dangerous course. This gives the navigator on the bridge more and better information for making decisions, but with the caveat that not all vessels have AIS. In the 2002 IMO SOLAS Agreement, it is required that vessels over 300 gross tonnage and all passenger vessels must be equipped with class A AIS transceivers. A simpler and cheaper AIS version named class B aimed at smaller vessels and yachts was published in 2006, followed by a large increase in the amount of non-commercial vessels equipped with AIS.

2.2.2 Messages

AIS broadcasts both static, dynamic and voyage information with varying intervals based on the vessels speed, status and on request from shore stations. Static, dynamic and voyage messages are listed in Table tables 2.1 to 2.3. When the AIS standard was developed, the peak traffic situations in the two most densely trafficked waterways, Singapore and

Dover Straits, where used to calculate the update frequency for the AIS system. Based on these two locations and a desire to keep the number of reports per minute below 2000, the dynamic information report intervals for class A and B was set as in Table 2.4 and 2.5 respectively [3]. Static information is transmitted every 6 minutes, and on request from VTS stations. AIS transceivers are utilizing two reserved VHF channels; AIS 1 — 87B (161.975MHz) and AIS 2 — 88B (162.025MHz) to improve robustness against interference. An important note is that AIS transceivers are alternating which channel they are transmitting on, which means that if a receiver is only listening on one channel, the effective update rate halves.

2.2.3 Class A

Class A AIS transceivers are designed for Self Organized Time Division Multiple Access (SOTDMA) transmission, which is a way of reserving transmission time slot for the next broadcast. SOTDMA is based on Time Division Multiple Access (TDMA), with an extension allowing for self organizing of time slots compared to TDMAs dedicated timing manager. This effectively gives class A AIS transmissions priority over Class B equipment which may not have SOTDMA. Class A transceivers are also required to have build-in display, minimum transmission power of 12.5W, ability to filter targets and communication interfaces like RS-232 and NMEA.

2.2.4 Class B

Class B AIS transceivers are designed to be simpler and cheaper than Class A transceivers, which is accomplished through less strict requirements for hardware and operation. Class B AIS transmits at lower power, usually 2W and transmits at larger time intervals than Class A, see Table 2.5. It is not required to have a build-in display and can use both Carrier Sense Time Division Multiple Access (CSTDMA) and SOTDMA for transmission. CSTDMA is a simpler approach to time division than SOTDMA since it only listen for a single time slot to be unused before it transmits.

Static AIS information	
MMSI	Maritime Mobile Service Identity
Call sign	Maritime radio (VHF) call sign
Name	Name of vessel
IMO Number	Vessel IMO number
Length and beam	
Location of positioning fixing antenna	
Height over keel	

Table 2.1: Static AIS information

Dynamic AIS information	
Position	In WGS84 frame
Position accuracy	Better or worse that 10 meter
Position time stamp	UTC in whole seconds
Course over ground (COG)	
Speed over ground (SOG)	
Heading	
Navigational status	
Rate of turn (ROT)	

Table 2.2: Dynamic AIS information

Voyage AIS information	
Draught	Depth in water
Hazardous cargo	Type
Destination	Name of place
Estimated time of arrival (ETA)	
Route plan / waypoints	
Number of persons on board	

Table 2.3: Voyage AIS information

Vessels status	Reporting Interval
Ship at anchor or moored and not moving faster than 3 knots	3 minutes
Ship at anchor or moored and moving faster than 3 knots	10 seconds
Ship 0–14 knots	10 seconds
Ship 0–14 knots and changing course	3.3 seconds
Ship 14–23 knots	6 seconds
Ship 14–23 knots and changing course	2 seconds
Ship > 23 knots	2 seconds
Ship > 23 knots and changing course	2 seconds

Table 2.4: Class A Reporting Intervals

Vessels status	Reporting Interval
Ship < 2 knots	3 minutes
Ship 2–14 knots	30 seconds
Ship 14–23 knots	15 seconds
Ship > 23 knots	5 seconds
Search and Rescue aircraft	10 seconds
Aids to navigation	3 minutes
AIS base station	10 seconds

Table 2.5: Class B Reporting Intervals

2.3 Tracking

2.3.1 Overview

Tracking, in this context, is the process of estimating the state of stationary and moving targets that are observed by a system without included association data. The challenge is to know which measurements belong together over time, often refereed to as the data association problem. An observation system can be a radar, sonar or any other sensor that, passively or actively, detects objects within an area or volume. Any observation system will be prone to noise, both in form of internal- and external noise from the environment. This noise will cause false measurements that the tracking system must take care of. These false measurements are often referred to as clutter.

In this work 'tracking algorithm' will be used to describe the main logic in a tracking method or approach, while 'tracking system' will be used on complete systems with everything around the main algorithm included. A tracking system can be defined as: *A system that process consecutive measurements from one or more observation system and associates measurement from the same target into tracks with initialization of new tracks and termination of dead tracks.* A track is a sequence of states associated with a subset of all measurements from the observation systems.

Tracking is a relative new filed of study, driven by the military and aerospace industry and enabled by the development of microprocessors and computers from the 1960's. The applications ranges from sonar tracking on both submarines and navy vessels, to air control and missile guidance. This historical background is likely the reason for most published papers using these types of applications as background for testing. In recent years, tracking people and vehicles from visual- and Synthetic Aperture Radar (SAR) imagery have also become a topic in the research community [4]–[6]. New applications areas like oceanography, autonomous vehicles and biomedical research have also found use of tracking [7]–[9].

There are several factors contributing to the challenge of good tracking; clutter, lower than unity Probability of detection (P_D), multiple detections of the same vessel and wakes. Clutter is a term for unwanted measurements or noise, which is inherent in every observation system. For a maritime radar, this can be caused by waves, rain, snow, birds or shore echo. A common assumption on clutter is to assume the amount being Poisson distributed, and spatially uniformly distributed. P_D is a measure of how persistent the target is in the measurements, and will vary much between different types of targets, primarily dependent on their size, construction material and shape. Multiple detections of each target can occur when the target is large and have several distinct areas which reflects radar waves better than the rest of the vessel, for instance when the hull of a boat is made from fibreglass and the metal objects inside is reflecting. Wakes are reflection caused by the turbulent water behind a moving object, which can be a problem for both sonar and maritime- and air-radar.

2.3.2 Single-target tracking

The simplest approach to tracking is single-target tracking, where it is assumed that there are only one target in the measurement area, and any other measurement is regarded as either extra measurements of the target or clutter.

Nearest Neighbour Filter (NNF)

The simplest single-target tracking algorithm is the NNF, where the closest measurement is always selected [10]. This approach is very vulnerable to clutter and dense multitarget scenarios, which is probably the reason its usage is very sparse.

Probabilistic Data Association Filter (PDAF)

The arguably best single-target tracking algorithm is PDAF, which calculates probabilities of all target to measurement association for all measurement inside its gate. A gate is an ellipse (2D) or ellipsoid (3D) which outlines the confidence area / volume for a predicted state and covariance for a given confidence value. The state update is then based on a weighted sum of measurement innovations where the weightings are the probabilities for their respective innovations. One of the main assumptions in PDAF is that all the measurements inside its gate contains information about the targets true state. This assumption performs well in single-target scenarios since targets often create more than measurements, and thus using all measurements in state estimation effectively rejects much of the clutter in each scan.

2.3.3 Multitarget tracking

A more generalized approach assuming that there can be any number of targets is called multitarget tracking. The problem expands to jointly estimate both the number of targets and their trajectories since targets can enter and leave the surveillance area.

While a large number of tracking techniques have been developed, the three most used are Joint Probabilistic Data Association Filter (JPDAF), Multi Hypothesis Tracking

(MHT) and Random Finite Set (RFS) [9]. Compared to MHT and JPDAF, RFS is a relatively new approach to tracking, and is not matured the same way MHT and JPDAF are. MHT and JPDAF also differs from RFS in that they both do data association and filtering, whereas RFS directly seeks both optimal and suboptimal estimates of the multitarget state [9].

JPDAF

JPDAF is a multitarget expansion of PDAF which is a single-target tracking technique. PDAF calculates joint posteriori association probabilities for every target in every scan. Each targets probability is a weighted sum of probabilities, where the weights are the key difference between PDAF and JPDAF. Like MHT, JPDAF suffers from high computational cost, and an efficient implementation approach exist and is patented by QinetiQ [11].

MHT

MHT is a decision logic which generates and maintains alternative hypotheses when new measurements are received and within the gate. By making several possible hypotheses, the decision on which measurement to choose can be propagated into the future when more information is available. MHT is multi frame method, meaning it has the ability to utilize multiple scans to make better decisions. Each hypothesis is given a score based on a likelihood ratio as a reflection of how well the measurement fits the model, which are accumulated to evaluate the combinations of consecutive measurements.

In contrast to the PDAF and JPDAF methods who suffers from track coalescence, MHT methods split when in doubt. The idea of using multiple hypotheses was first introduced by [12], but the first complete algorithm was presented in [13], where a Hypothesis Oriented Multi Hypothesis Tracker (HOMHT) was developed. Following this, a TOMHT was proposed in [14] and the score function for MHT was later deduced and discussed in [15] since no explicit track-score function were given in [14]. MHT is, in the same way as PDAF/JPDAF, developed under two main assumptions; that at most one measurement can originate from each target in each scan, and that a target does not necessarily show on every scan, or in other words, P_D less than 1. The MHT approach to tracking and data association was for a long time dismissed by many because of its computationally large cost. The dramatic increase in computational capability from the 1980's to the late 2010's have, however, lead to a new spring for MHT, with an increasing interest for use in tracking system. In 2004 Blackman stated that "Multiple hypothesis tracking is generally accepted as the preferred method for solving the data association problem in modern multiple target tracking system" [16]. Already in 2001 did Blackman publish a demonstration that MHT is capable of real-time demands [17].

MHT comes in two variations, HOMHT and TOMHT. They differ in their approach to arrange the measurements into hypotheses in that HOMHT builds hypotheses that are different ways of organizing the measurements into tracks, while TOMHT maintains already existing tracks and a hypothesis is only a possible track for a single target and not all targets. This means that in TOMHT, to select the

RFS

RFS is a family of Bayesian methods and filters that is based on representing a multitarget state as a finite set of single target states. By not having the multitarget states represented as a vector, the estimation error is more meaningful and mathematically consistent. This is because the vector representation is ordered, and thus the only meaningful way of representing the estimation error would be to minimize the estimation error over all permutations of states. This can be represented as distances between finite sets, and is a much more well defined and understood concept.

The RFS approach to multi sensor multitarget tracking was done by Mahler in 1994, which lay the foundation for the development of Finite Set Statistic (FISST). One popular filter based on RFS is the Probability Hypothesis Density (PHD) filter. The PHD filter is an approximation of the multitarget Bayes filter derived by Mahler using FISST [9]. The PHD filter estimates the number of targets and then selects the same numbers highest probable tracks. One large drawback with the PHD filter is its assumption that the predicted multitarget RFS is Poisson distributed. This assumption is relaxed in the Cardinalized Probability Hypothesis Density (CPHD) filter where the prior and predicted multitarget densities are independently and identically distributed clusters.

Chapter 3

Radar and AIS preprocessing

The process from raw radar and AIS data to target tracks is made up from several processing steps. The aim of this chapter is to give the reader a basic understanding of these steps and their challenges.

3.1 Radar preprocessing

Rotating maritime radars (Figure 2.3) are wide and short, giving them a tall and narrow beam. A ping transmit and receive sequence is carried out for each antenna rotation angle in the radars scan resolution. This gives reflections as signal level in spokes described by polar coordinates, rotation angle and distance. Each spoke has a width determined by the design of the antenna, primarily the width of the antenna, and a number of cells dictated by the discretization and sampling interval of each spoke. The spokes is then run through a detection algorithm, which is filtering adjusting the received signal according to detection setting. The detection algorithm is often built in to the radar system, with both fixed and user adjustable detection parameters.

When displayed on a screen in a vessels, the output from the detection step is viewed and interpreted by the operators. In an automated scenario with autonomous vessels, the next step would be to transform the detections from polar vessel body frame to for instance a Cartesian world fixed local frame. Which frame to convert two is a design choice, and can be dependent on use-case, interconnected systems and performance requirements. This transformation is strongly dependent on knowing the position and attitude of the vessel at each spoke sampling time, which is fed from the vessels navigation system.

With all the spoke resolution cells converted to a world-fixed Cartesian coordinate system, it is desirable to remove land reflections if any. This step is dependent on highly detailed digital maps of the area in question, and is commercially available for most of the world. Since maps have both offsets and inaccuracies to some extent, a cleaner land masking can be accomplished by dilating the coastline. This is in may situations acceptable since the vessels will never be that close to shore, and any targets masked away is in a region out of interest.

The last step in the radar processing chain is to convert a point cloud into measurements, as one target will in most cases fill multiple resolution cells and therefore it does not yield good result to send all cells with detection forward as measurements. This clustering of the detections also need to take into consideration the assumption that each target maximum generates one measurements. This leads to clustering algorithms that assumes that detections closely spaced are originating from the same target, and thus should be one measurement. There is many clustering algorithms available to solve this problem, some builds graphs with vertices between neighbouring detections given a neighbour criterion, some estimates the number of clusters and optimizing the detections into this number of clusters [18], [19]. When a set of detections are clustered, their respective measurement is calculated as the centroid of the detections, which would be weighted by their signal strength if available. These measurements are sent to the tracking module.

3.1.1 Frame conversion

The radar measurements is by nature in polar frame, and the target motion model is best described in a Cartesian frame. The most usual solution is to convert the radar measurements to a Cartesian frame, and to avoid biased and optimistic covariances of the converted measurements, a procedure which compensates for these errors, (3.1) and (3.2), should be used in stead of the standard conversion (3.3) and (3.4) [20].

$$\begin{aligned} \begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} r_m \cos\theta_m \\ r_m \sin\theta_m \end{bmatrix} - \mu_a \\ \mu_a \triangleq \begin{bmatrix} E[\tilde{x}|r_m, \theta_m] \\ E[\tilde{y}|r_m, \theta_m] \end{bmatrix} &= \begin{bmatrix} r_m \cos\theta_m (e^{-\sigma_\theta^2}) - e^{-\sigma_\theta^2/2} \\ r_m \sin\theta_m (e^{-\sigma_\theta^2}) - e^{-\sigma_\theta^2/2} \end{bmatrix} \end{aligned} \tag{3.1}$$

$$
\begin{aligned}
R_a^{11} &\triangleq r_m^2 e^{-2\sigma_\theta^2}[\cos^2\theta_m(\cosh 2\sigma_\theta^2 - \cosh\sigma_\theta^2) + \sin^2\theta_m(\sinh 2\sigma_\theta^2 - \sinh\sigma_\theta^2)] \\
&\quad + \sigma_r^2 e^{-2\sigma_\theta^2}[\cos^2\theta_m(2\cosh 2\sigma_\theta^2 - \cosh\sigma_\theta^2) + \sin^2\theta_m(2\sinh 2\sigma_\theta^2 - \sinh\sigma_\theta^2)] \\
R_a^{22} &\triangleq r_m^2 e^{-2\sigma_\theta^2}[\sin^2\theta_m(\cosh 2\sigma_\theta^2 - \cosh\sigma_\theta^2) + \cos^2\theta_m(\sinh 2\sigma_\theta^2 - \sinh\sigma_\theta^2)] \\
&\quad + \sigma_r^2 e^{-2\sigma_\theta^2}[\sin^2\theta_m(2\cosh 2\sigma_\theta^2 - \cosh\sigma_\theta^2) + \cos^2\theta_m(2\sinh 2\sigma_\theta^2 - \sinh\sigma_\theta^2)] \\
R_a^{12} &\triangleq \sin\theta_m\cos\theta_m e^{-4\sigma_\theta^2}[\sigma_r^2 + (r_m^2 + \sigma_r^2)(1 - e^{\sigma_\theta^2})]
\end{aligned}
\tag{3.2}
$$

$$
x = r_m\cos\theta_m \qquad y = r_m\sin\theta_m \tag{3.3}
$$

$$
\begin{aligned}
R_L^{11} &\triangleq r_m^2\sigma_\theta^2\sin^2\theta_m + \sigma_r^2\cos^2\theta_m \\
R_L^{22} &\triangleq r_m^2\sigma_\theta^2\cos^2\theta_m + \sigma_r^2\sin^2\theta_m \\
R_L^{12} &\triangleq (\sigma_r^2 - r_m^2\sigma_\theta^2)\sin\theta_m\cos\theta_m
\end{aligned}
\tag{3.4}
$$

$$
\begin{aligned}
r_m &= \text{measured range} \\
\theta_m &= \text{measured bearing} \\
\sigma_r &= \text{range measurement standard deviation} \\
\sigma_\theta &= \text{bearing measurement standard deviation} \\
\boldsymbol{R_a} &= \text{average true converted measurement covariance} \\
\boldsymbol{R_L} &= \text{linearised converted measurement covariance}
\end{aligned}
$$

3.2 AIS preprocessing

AIS does not suffer from the association uncertainty, clutter and low accuracy like radar measurement. It does however have some issues caused by suboptimal or erroneous transmitter implementation, transmission collision caused by TDMA leading to ID (Mobile Maritime Safety Identity (MMSI)) swaps, and delayed messages leading to out of order reception. In order to remove most of these errors, it is desirable to filter the incoming AIS messages before sending them to the tracking module.

3.2.1 Out-of-order filtering

All AIS messages are stamped with the Coordinated Universal Time (UTC) of transmission, and 'frequently arrive out-of-order' [21], illustrated in Figure 3.1 stolen from [21].

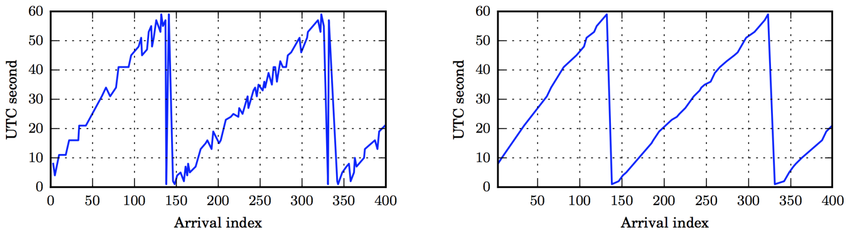

Figure 3.1: Unfiltered and filtered AIS arrival time

One of the simplest ways of remedying this issue is to discard all messages with older timestamps that the current newest for each MMSI. This will lead to a loss of data, which will lead to a slower AIS update period for the tracking module.

3.2.2 ID swap filtering

According to [22], 2% of the received AIS messages in a data-mining study contained erroneous ID or MMSI. One of the errors where that many vessels transmitted messages with the same MMSI (11930446). This is the default MMSI on equipment from a specific manufacturer. Another example is two vessels which swapped IDs for a moment when they were passing, with a recovery after about 15 minutes. The latest example could be caused by simultaneous transmission or reflections, but the cause is not examined in the paper. Although much rare that the out-of-order reception

To remedy this issue, a simple test logic can be incorporated to check for obvious faults like sudden large position change and known default IDs. When a message and MMSI is categorized as bad, it would be held back from the tracking module.

3.2.3 Synchronization

Since AIS messages arrive asynchronously and the tracking module is only accepting AIS updates along with radar updates, we must synchronize the incoming AIS messages. In this work, all AIS measurements are buffered from when they are received until the next radar scan.

Chapter 4

MHT Module

To create a complete tracking *system*, rather than a tracking *algorithm*, it is often necessary to complement the main algorithm with support modules. The system, or module if it is a part of a bigger system, presented here is an extension of the pre master project [2]. The aim of this chapter is to provide a complete walkthrough of the the track oriented MHT system developed in this thesis. The motion model which is used throughout the entire tracking system when predicting and filtering target behaviour is presented first. Next follows an overview of the algorithm used to initiate new tracks into the MHT algorithm, followed by the entire MHT tracking algorithm with all its sub-routines.

It will be assumed throughout this thesis that radar data is processed, as outlined in Section 3.1, into a set of points. These points is referred to as radar measurements.

4.1 Motion Model

4.1.1 Reference frame

A local Cartesian NED-frame, like Universal Transverse Mercator coordinate system (UTM) will be used throughout this thesis, with the assumption than all input sensors are transformed to this frame (see Section 3.1.1). This local projection from a geodetic coordinate system to a Cartesian coordinate system is acceptable as long as the the area the system is working on is within one grid. A global geodetic frame, like WGS84 would be preferable in situations where the system tracks object over world-scale lengths but would yield non-linear equations of motion.

4.1.2 Constant velocity model

The state (4.1) of the targets are modelled with in four dimensions in a Cartesian frame where the positive x-axis is pointing east and the positive y-axis is pointing north. The two latest states are the velocities in their respective direction.

$$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}^T \tag{4.1}$$

Since modelling the behaviour of any ship under unknown command is next to impossible, a common assumption in tracking theory is that every target will continue on as usual, more precisely that their velocity is constant. Although simple, this model captures the essence of most vessels at sea, and when looking at maritime training [23] and regulation [24], they both dictates that vessels should hold steady course and change course in clear decisive turns. This model is also very common in tracking applications and is used in [6], [9], [13], [21], [25]–[27] among others. To give room in our model for manoeuvring, the process noise covariance is set according to the assumed manoeuvring capabilities of the vessels. This could be set as a fixed value for all targets, as done in this work, or estimated based on the history of the track or AIS information. This behaviour can be modelled as a linear time invariant system with time evolution (4.2), measurement model (4.3), transition and observation matrices (4.4) and system and measurement noise matrices (4.5).

$$\mathbf{x}_{k+1} = \mathbf{\Phi}\mathbf{x}_k + \mathbf{w}_k \qquad \mathbf{w} \sim \mathcal{N}(0; \boldsymbol{Q}) \tag{4.2}$$

$$\mathbf{z}_{k+1} = \boldsymbol{H}\mathbf{x}_k + \mathbf{v}_k \qquad \mathbf{v} \sim \mathcal{N}(0; \boldsymbol{R}) \tag{4.3}$$

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{4.4}$$

$$\boldsymbol{Q} = \sigma_v^2 \begin{bmatrix} \frac{T^3}{3} & 0 & \frac{T^2}{2} & 0 \\ 0 & \frac{T^3}{3} & 0 & \frac{T^2}{2} \\ \frac{T^2}{2} & 0 & T & 0 \\ 0 & \frac{T^2}{2} & 0 & T \end{bmatrix} \qquad \boldsymbol{R} = \sigma_m^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.5}$$

$$
\begin{aligned}
\mathbf{\Phi} &= \text{state transition matrix} \\
\boldsymbol{H} &= \text{state observation matrix} \\
\boldsymbol{Q} &= \text{system covariance matrix} \\
\mathbf{w} &= \text{system noise} \\
\mathbf{v} &= \text{measurement noise} \\
\mathbf{z} &= \text{measurement vector} \\
k &= \text{time index} \\
T &= \text{time step}
\end{aligned}
$$

4.2 Track Initiation

In comparison with HOMHT who treats every measurement as a potential new track since its hypotheses are essentially different ways of organizing its measurements into tracks, which are repeated for every iteration, TOMHT does not have any built-in initialization of tracks since it only maintains an already existing track with track splitting and measurement-to-track association for every iteration. To remedy this lack, we need an algorithm that can find consistent and predictable patterns in an assumed uniformly distributed measurement space of clutter.

In this work, new tracks are initiated with 2/2 & m/n logic [9] on the unused measurements after each MHT iteration. As the name of the method indicates, this is a two step verification, where the first act as a rough filter and the second as a fine filter. As one of the main assumptions in most tracking systems, the measured radar clutter is assumed uniformly distributed in the measurement area. This assumption is quite rough in the radial measurement space, and even worse approximation in Cartesian measurement space, as illustrated in Figures 4.1 and 4.2. Although far from perfect, tests shows that we still get satisfactory performance from the 2/2&m/n method.

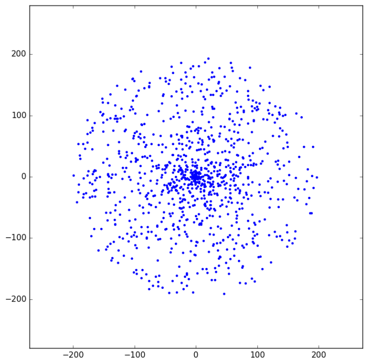

Figure 4.1: Uniform radial clutter

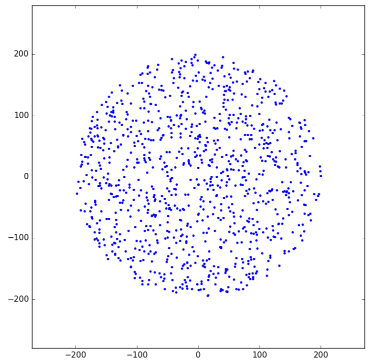

Figure 4.2: Uniform Cartesian clutter

The flow of the method is illustrated in Figure 4.3, and for better clarity, the algorithm is explained from the last step to the first step, since this is the sequence a newly started initiation algorithm will perform its operations.

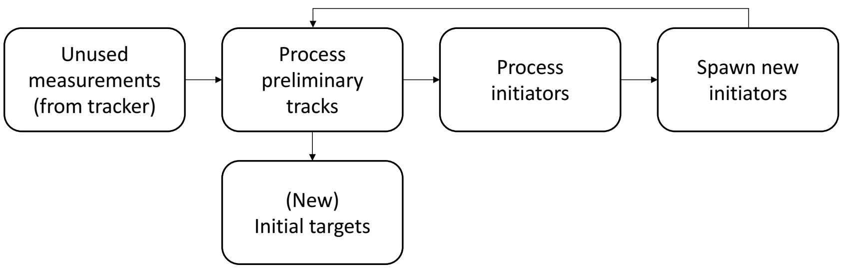

Figure 4.3: 2/2&m/n flowchart

4.2.1 Spawn new initiators

All measurement unused by the 'Process preliminary tracks' and 'Process initiators' steps will be the basis for new *initiators*. An initiator is a measurement that that awaits its match in the next scan. The idea is that uniformly distributed clutter will not (often) reappear at approximately the same location two times in a row, effectively filtering out most of the clutter.

4.2.2 Process initiators

When the next scan arrives, all the unused measurements from the 'Process preliminary tracks' step will be used as candidates in this step. Since an initiator is only a position and not a full state with velocity, all directions are equally likely, and the only design parameter in this step is maximum speed of targets to be tracked. This parameter sets an outer limit on the circle acting as a gate for the second and confirming measurement. When matching initiators with a second measurement, we want to select the closest measurement, making the assumption that the two consecutive measurements are the most likely to belong together. In a single target scenario, where this would be to calculate the distance to all the alternative measurements and select the lowest, the association is already done. While in a multitarget scenario, we *could* select the closest measurement to any initiator, but we would have to do this one initiator at a time. This would lead to different results depending on the arrangement of the initiators in the programming of this method. A different approach would be to calculate all the different distances for any possible combination of initiators and measurements, sort the list, and assign the distances from the shortest to the longest possible distances. This approach would not be influenced by randomness like the arrangement of the initiators in a programming language, but would not necessarily give the global optimal association regarding the how many initiators that are assigned measurements and their respective distances.

Since we can have situations like exemplified in Figure 4.4, where two initiators have the same measurement inside their gates, and one of them have a second measurement inside its gate, we need to take the global consequence of any assignment into consideration. If using method 1; to sequentially select the best, we have two possible outcomes. When starting with initiator 1, this initiator would be associated with measurement 2, and initiator 2 would not be associated with any measurements. On the other hand, starting with initiator 2 would lead to this initiator being associated with measurement 2, and initiator 1 would be associated with measurement 1. This randomness in outcome based on which initiator the algorithm starts with is clearly not a desired property. If using using method 2; to sequentially select the globally shortest distance, we would first associate initiator 1 with measurement 2, and there would not be any measurements left for initiator 2, leaving this empty.

A third option is to formulate the problem as a global combinatorial problem, and use an 'off-the-shelf' solution to solve the problem. We have essentially a matrix with initiators along one axis and measurements along the second axis and the distance between

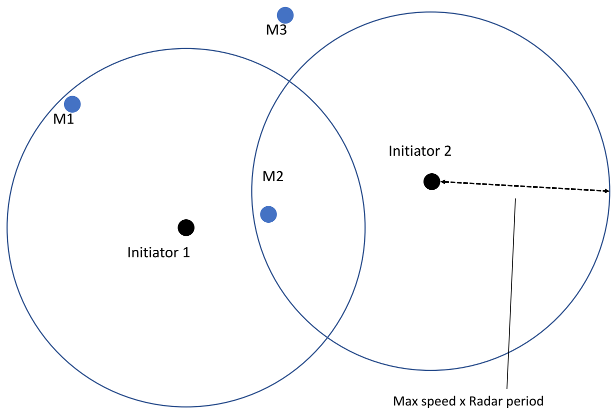

Figure 4.4: Initiator gating example

them in their intersections, as in (4.6) for our example.

$$\begin{array}{c c} & \begin{matrix} M_1 & M_2 & M_3 \end{matrix} \\ \begin{matrix} I_1 \\ I_2 \end{matrix} & \begin{bmatrix} 3 & 1 & 5 \\ 7 & 2 & 6 \end{bmatrix} \end{array} \tag{4.6}$$

The values above the threshold set by the maximum speed multiplied with the time period between the radar scans can be set to infinity to symbolise that this combination is not possible, see (4.7) where the gate threshold is 4.

$$\begin{array}{c c} & \begin{matrix} M_1 & M_2 & M_3 \end{matrix} \\ \begin{matrix} I_1 \\ I_2 \end{matrix} & \begin{bmatrix} 3 & 1 & \infty \\ \infty & 2 & \infty \end{bmatrix} \end{array} \tag{4.7}$$

If we remove the columns with only infinity, we are removing measurements that cannot be associated under any circumstances, thus reducing the size of the problem, see (4.8). With this pre processing, we want to assign each row to a column so that the sum of the selected intersections are minimal.

$$\begin{array}{c c} & \begin{matrix} M_1 & M_2 \end{matrix} \\ \begin{matrix} I_1 \\ I_2 \end{matrix} & \begin{bmatrix} 3 & 1 \\ \infty & 2 \end{bmatrix} \end{array} \tag{4.8}$$

We now have formulated our problem in a way that it can be solved by the ‘Hungarian’[1] algorithm [28], which will give us the association $I_1 \rightarrow M_1$ and $I_2 \rightarrow M_2$. From the associations, a full state is calculated and a new preliminary track is created. A preliminary track contains a state, covariance and counters of number of checks and passed checks.

4.2.3 Process preliminary tracks

When a new set of unused measurements arrive from the tracker, all the preliminary tracks are predicted to the time of the measurements. We now have the same association challenge between the predicted states and the measurements as with the initiators and measurements. Since we now have a full state and covariance for every preliminary track, we calculate the Normalized Innovation Squared (NIS) for every combination of preliminary tracks and measurements, and selects the best combination. The preliminary tracks that is associated with a new measurement, their passed counter is incremented with one, while all preliminary tracks’ checks counter is incremented with one.

For preliminary tracks that have enough passed measurements a new initial target is sent to the tracker. All preliminary tracks with check counter above the threshold is categorized as dead and deleted.

4.3 MHT Overview

The aim of this section is to outline the major steps in the MHT module and the flow of data and decisions. Figure 4.6 shows the main steps that the module perform at each iteration / radar scan. The MHT algorithm is working on a set of Directed Acyclic Graphs (DAGs) or tree structures, often called a forest. The forest contains as many trees as targets the algorithm is tracking. Each tree consist of a root node and a set of child nodes, where each row represent a scan or discrete time. The leaf nodes are referred to as track hypotheses since leaf nodes represents itself and its parents.

When new AIS and radar measurements are received, all leaf nodes are predicted forward to the time of the radar measurements. The radar measurements are then gated for each leaf node, and new hypotheses are generated for measurements within the gate. The AIS measurements are gated at their time, and then predicted to the time of the radar measurements. The radar measurements are then gated gated based on each of the filtered AIS measurements, where the gated radar measurements give rise for fused

[1] also known as the Munkres or Kuhn-Munkres algorithm

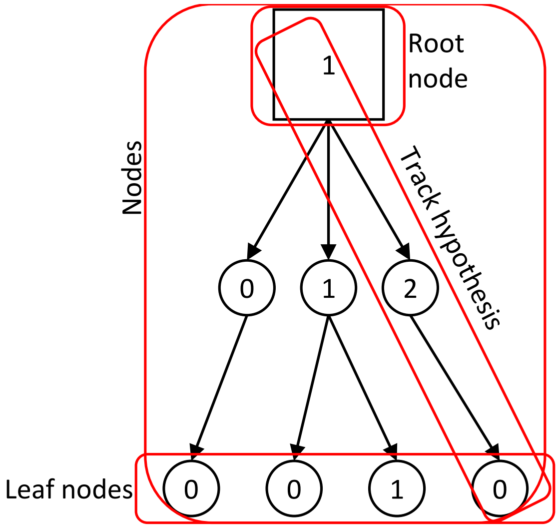

Figure 4.5: Hypothesis tree

hypotheses. Each new hypothesis is then given a score, which is the cumulative score of the parent node score and the new node's score. The target trees are then clustered according to which trees that shares measurements, whereon clusters with only one tree has the option of removing / merging similar hypotheses to reduce the size of the tree. For each cluster, the cluster-wise globally best association combination is selected using ILP. Then, for each selected hypothesis the parent N steps above becomes the new root of that tree, and the unused children to the previous root node are removed. Next, targets who's best hypothesis have a score below the threshold is terminated, followed by the initialization of new targets from the initiator module.

The algorithm described in the three following sections are repeated for every leaf node in the forest. To avoid an extensive use of node- and measurement indexing the procedure is explained for one leaf node, with this node referred to as 0–node, and is repeated for all leaf nodes in the forest.

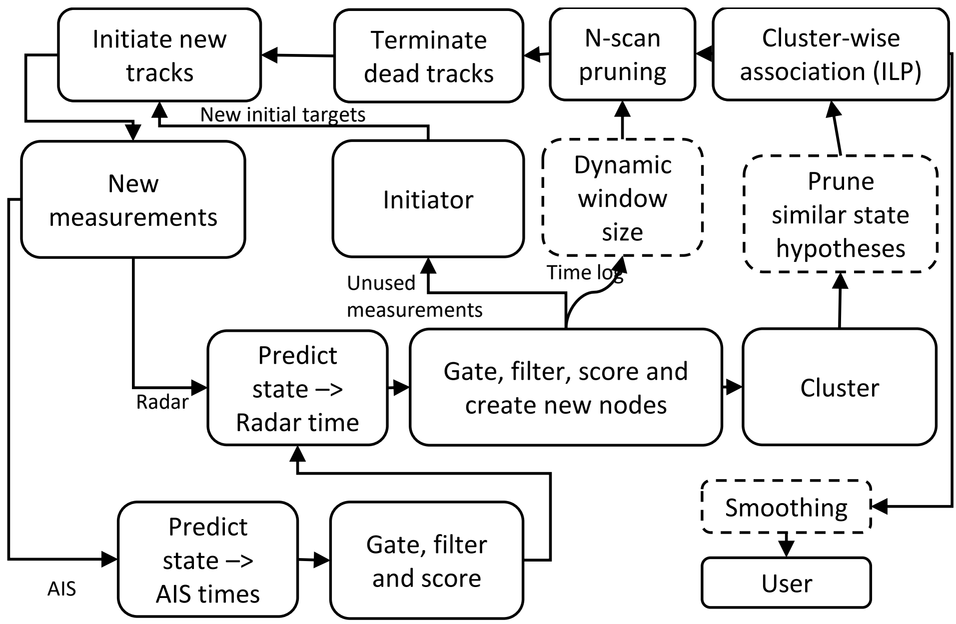

Figure 4.6: Algorithm flowchart

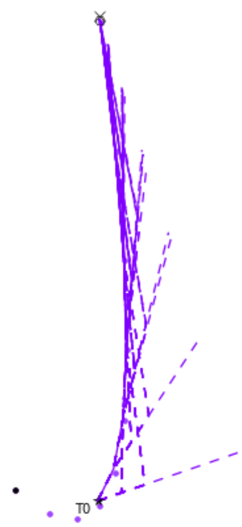

Figure 4.7: Hypotheses when turning

4.4 Process radar measurements

4.4.1 Predict to radar time

To compare new radar measurements with existing hypotheses, we should predict their states to the same time as the radar measurements, which can be done with the Kalman filter 'time update' equation (4.9) and the motion model from Section 4.1. The residual covariance (4.10), which is a part of the 'measurement update' sequence of a Kalman filter are also calculated as the residual covariance is needed in the gating and these matrices are not dependent on the measurement residual.

$$\begin{aligned} \bar{\mathbf{x}}_1 &= \boldsymbol{\Phi}(\Delta T_R)\mathbf{x}_0 \\ \bar{\boldsymbol{P}}_1 &= \boldsymbol{\Phi}(\Delta T_R)\boldsymbol{P}_0\boldsymbol{\Phi}^T(\Delta T_R) + \boldsymbol{Q}(\Delta T_R) \end{aligned} \tag{4.9}$$

$$\boldsymbol{S}_1 = \boldsymbol{H}\bar{\boldsymbol{P}}_1\boldsymbol{H}^T + \boldsymbol{R}_{Radar} \tag{4.10}$$

$$\begin{aligned} \bar{\mathbf{x}} &= \text{predicted state} \\ \mathbf{x}_0 &= \text{origin node state} \\ \bar{\boldsymbol{P}} &= \text{predicted state covariance} \\ \boldsymbol{P}_0 &= \text{origin node state covariance} \\ \boldsymbol{Q} &= \text{system noise covariance} \\ \Delta T_R &= \text{Radar time period} \end{aligned}$$

4.4.2 Gate

To limit the number of hypotheses the leaf node have to create, the measurements are gated based on the leaf node's predicted covariance and a set confidence value. The size of the gate (Figure 4.8) will reflect how insecure the prediction is, which is a function of how many detections and missed detections the leaf node and its parents have had. The gate is defined as NIS less that a threshold set by the χ^2–distribution Cumulative Distribution Function (CDF) with as many degrees of freedom as the measurement, two degrees of freedom for a maritime radar, and a set confidence value. A set of confidence levels and belonging χ^2 CDF values are listed in Table 4.1. The measurement residual and NIS is calculated for each measurement, and the measurements that does not pass the test are discarded.

$$\begin{aligned} \tilde{\mathbf{z}} &= \mathbf{z} - \boldsymbol{H}\bar{\mathbf{x}}_1 \\ NIS &= \tilde{\mathbf{z}}^T\boldsymbol{S}^{-1}\tilde{\mathbf{z}} \leq \eta^2 \end{aligned} \tag{4.11}$$

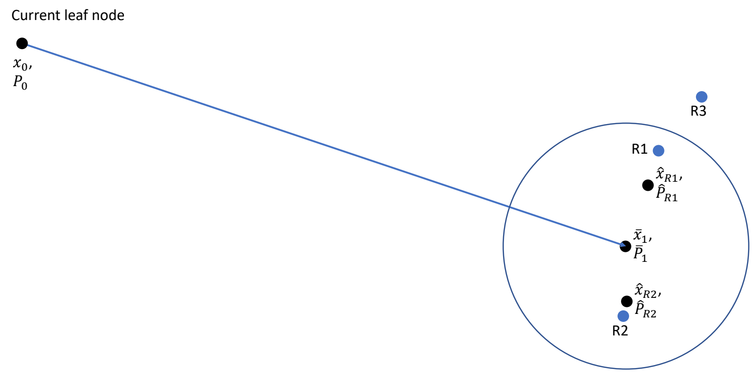

Figure 4.8: Radar gating

$$\tilde{\mathbf{z}} = \text{Measurement residual}$$
$$\eta^2 = \text{Inverse } \chi^2 \text{ CDF}$$

4.4.3 Filter, score and create new nodes

The scoring used in this tracking system is based on a dimensionless score function by Bar-Shalom [15]. His paper discusses the issue of scoring measurement-to-track associations and comparing scores based on different numbers of measurement and measurement dimensions. He proposes a dimensionless *likelihood ratio*, which is the Probability Density Function (PDF) of a measurement having originating from the track, to the PDF of it not originating from the track.

Each track hypothesis is scored according to (4.12). Where the cumulative score being the sum through time since we are using the logarithm of the likelihood ration. For the fused hypotheses, their score is the cumulative score of both the measurements, giving them a better score reflecting that they are more likely hypotheses that pure radar or AIS

Confidence	70%	80%	90%	95%	97.5%	99%	99.5%
η^2	2.41	3.22	4.61	5.99	7.38	9.21	10.60

Table 4.1: Inverse χ^2 CDF for two degrees of freedom

hypotheses.

$$\text{NLLR} = \frac{1}{2}NIS + \ln\frac{\lambda_{ex}|2\pi S|^{1/2}}{P_D} \tag{4.12}$$

$$\text{cNLLR}_k^j \triangleq \sum_{l=0}^{k} NLLR_{i,j}(l) \tag{4.13}$$

Zero hypothesis

To account for the possibility that the target is not present in this scan, a *zero* hypothesis, or *dummy* hypothesis as it is sometimes called, is generated with the predicted state and covariance $\bar{\mathbf{x}}_1, \bar{\boldsymbol{P}}_1$. This node is numbered 0 in Figure 4.5 and 4.10.

Pure radar hypotheses

For every radar measurement inside the gate in (4.11), a new track hypothesis is generated with filtered state and covariance according to the regular Kalman 'measurement update' equation (4.14). These hypotheses are numbered incrementally from 1 and upwards.

$$\begin{aligned} \boldsymbol{K}_1 &= \bar{\boldsymbol{P}}_1 \boldsymbol{H}^T \boldsymbol{S}_1^{-1} \\ \hat{\mathbf{x}}_1 &= \bar{\mathbf{x}}_1 + \boldsymbol{K}\tilde{\mathbf{z}} \\ \hat{\boldsymbol{P}}_1 &= (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\,\bar{\boldsymbol{P}}_1 \end{aligned} \tag{4.14}$$

$$\begin{aligned} \boldsymbol{K} &= \text{Kalman gain} \\ \boldsymbol{S} &= \text{Covariance innovation} \\ \hat{\boldsymbol{P}}_1 &= \text{filtered state covariance} \end{aligned}$$

4.5 Process AIS measurements

As elaborated in Section 3.2 all AIS measurements are preprocessed to remove out-of-order messages and ID-swap errors. And for each radar scan, only the latest AIS update from each target (MMSI number) are passed through to the MHT tracking loop.

The integration of AIS measurements into the MHT framework is not obvious and multiple approaches is possible. Since the AIS and radar measurement originates from different times, the fusion process is done as a sequential update [20]. The first step in any approach would be to decide which AIS measurements the leaf node shall consider. This leads to two alternatives; compare and gate at the AIS message time or the radar measurements time.

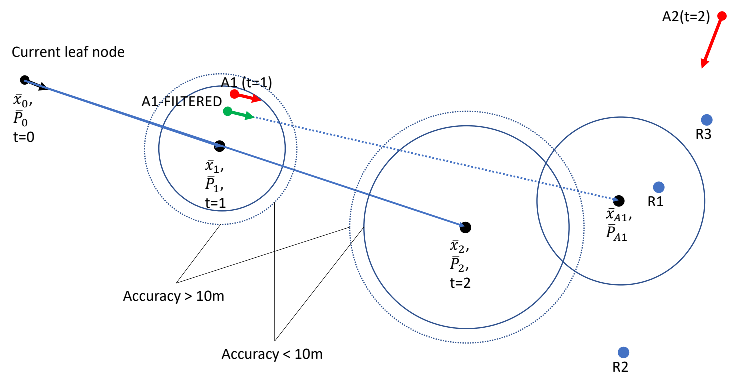

Figure 4.9: AIS gating

If gating at AIS time, the node would have to be predicted to the time of the all the different AIS measurements received since last scan. Since most maritime radars operate between 24 and 48 Rotations Per Minute (RPM) and the AIS time is always integer, a maximum of 2 different AIS times will exist in between two radar scans, making this approach feasible. Since each AIS measurement can have different accuracy and measurement covariance, the gating would have to be carried out one AIS measurement at a time with its own covariance residual. For all AIS measurement inside their gates, a filtered state and covariance is calculated and predicted forward to the radar measurements time. This prediction is then used to gate radar measurements and create a fused node for each radar measurement inside this gate.

Another approach would be to use the predicted AIS measurements at the radar time and gate them based on the predicted origin node state $\hat{\mathbf{x}}_1$ and AIS measurement covariance. This approach could lead to unwanted AIS measurements inside the gate since the AIS measurements from other vessels can be predicted into the gate, thus leading to a more challenging association. This is examplified in Figure 4.9 A2, where AIS measurement 2 is would be inside the radar measurement gate but not in the AIS time gates.

In this work, the first approach is used for its assumed better performance and utilization of original data rather than predicted data.

4.5.1 Predict to AIS times

To gate the AIS measurements, we first have to predict the target state and covariance (4.16) for each of the AIS time stamps (4.15) in the received AIS measurements. Since

AIS only transmits a single integrity status, better or worse than 10 meter, maximum two possible $\boldsymbol{R}_{AIS}$ must be considered. With maximum two different time stamps and two different accuracies, a maximum of four different gates must be considered.

$$\begin{aligned}\Delta T_{AIS_1} &= t_{AIS_1} - t_0 \\ \Delta T_{AIS_2} &= t_{AIS_2} - t_0\end{aligned} \tag{4.15}$$

$$\begin{aligned}
\bar{\mathbf{x}}_1 &= \boldsymbol{\Phi}(\Delta T_{AIS_1})\mathbf{x}_0 \\
\bar{\boldsymbol{P}}_1 &= \boldsymbol{\Phi}(\Delta T_{AIS_1})\boldsymbol{P}_0\boldsymbol{\Phi}^T(\Delta T_{AIS_1}) + \boldsymbol{Q}(\Delta T_{AIS_1}) \\
\boldsymbol{S}_{1_{High}} &= \boldsymbol{H}\bar{\boldsymbol{P}}_1\boldsymbol{H}^T + \boldsymbol{R}_{AIS,High} \\
\boldsymbol{S}_{1_{Low}} &= \boldsymbol{H}\bar{\boldsymbol{P}}_1\boldsymbol{H}^T + \boldsymbol{R}_{AIS,Low} \\
\bar{\mathbf{x}}_2 &= \boldsymbol{\Phi}(\Delta T_{AIS_2})\mathbf{x}_0 \\
\bar{\boldsymbol{P}}_2 &= \boldsymbol{\Phi}(\Delta T_{AIS_2})\boldsymbol{P}_0\boldsymbol{\Phi}^T(\Delta T_{AIS_2}) + \boldsymbol{Q}(\Delta T_{AIS_2}) \\
\boldsymbol{S}_{2_{High}} &= \boldsymbol{H}\bar{\boldsymbol{P}}_2\boldsymbol{H}^T + \boldsymbol{R}_{AIS,High} \\
\boldsymbol{S}_{2_{Low}} &= \boldsymbol{H}\bar{\boldsymbol{P}}_2\boldsymbol{H}^T + \boldsymbol{R}_{AIS,Low}
\end{aligned} \tag{4.16}$$

4.5.2 Gate, filter and score

For each leaf node all AIS measurements are gated with the gate matching their time and accuracy. The measurements that pass the gating is then filtered with the predicted state and covariance matching its time, giving rise to an intermittent node (4.17).

$$\hat{\mathbf{x}}_1 = \bar{\mathbf{x}}_1 + \boldsymbol{K}\tilde{\mathbf{z}} \tag{4.17}$$

Since the score for radar measurements is a dimensionless likelihood ratio, we must use the same dimensionless likelihood on the AIS measurements. Depending on how we want the AIS to affect our tracking, three different scoring strategies are explored. The first approach is to view the AIS as a pure *aiding*, where its only purpose is to improve the gating and uncertainty for the radar measurements. In this case, all AIS measurements are gated with a logarithmic score of zero, leading to neither improvement or worsening of the accumulative score of the track.

A second approach is to modify the score function for radar measurements. The maybe simplest way of doing this is to score according to (4.18), which is a modified version of the radar measurement score function (4.12). The first difference being that the probability of detection is changed with the AIS probability of receive, which is approximated to unity. Since the AIS measurements does not contain Poisson distributed clutter and the only way to 'remove' the clutter aspect of the scoring is to set λ_{ex} to 1, which is to say that there is an enormous amount of clutter and would consequently yield very bad scores for AIS measurements.

$$\text{NLLR} = \frac{1}{2}NIS + \ln|2\pi S|^{1/2} \tag{4.18}$$

Since the AIS measurements are inherently labelled, then for a single target all AIS measurements except maximum one (depending on whether it have AIS transceiver or not) can be considered as clutter. If we then make the same assumptions as for radar measurements with respect to uniform spatial distribution and Poisson density distribution, we can estimate the expected number of AIS 'clutter' measurements λ_{AIS} based on the amount of targets with AIS transmitters and the observation area (4.19), where r_{radar} is the radar range. This estimate could be calculated for a single frame, or averaged over a sliding window to reflect the AIS message flow over time since AIS transmission is not synchronised with the radar period. The resulting score function would be (4.20).

$$\lambda_{AIS} = \frac{n_{AIS}}{\pi r_{radar}^2} \tag{4.19}$$

$$\text{NLLR} = \frac{1}{2}NIS + \ln\lambda_{AIS}|2\pi S|^{1/2} \tag{4.20}$$

Of the three scoring methods explored, only the first and last is usable and mathematically sane. The second approach would yield scores so bad that the hypotheses containing an AIS measurement would never be selected, hence working against its purpose. Testing has shown that both the first and last method gives an improvement over pure radar tracking, and since the last method gives a little more improvement since it scores the AIS measurements with a reasonable values compared to the radar measurements, hence not giving the AIS measurements an enormous advantage or disadvantage. The third method is used in all the simulations in Chapter 5.

4.5.3 Predict to radar time

All gated and filtered states from Section 4.5.2 is then predicted forward to the time of the radar measurements. This time delta will differ based on the time of the AIS measurement. Radar measurements are then gated for each predicted state and covariance, whereon a fused hypothesis are created for each radar measurement inside the gate. If there is no radar measurements inside the gate, a pure AIS hypothesis are created.

Fused hypotheses

The predicted state and covariance is then filtered with the gated radar measurement according to (4.14). The radar measurement is scored according to (4.12), and the hypothesis score is the sum of the AIS and radar score.

Pure AIS hypotheses

If no radar measurements are present in the gate, pure AIS hypotheses are created. This can be the situation when a target is broadcasting an AIS message, but is either in radar shadow or is not detected by the radar for any reason. These hypotheses are not created when one or more radar measurements are available, based on the assumption that if a radar measurement is present, the difference between a fused hypothesis and a pure AIS hypothesis is quite small since the AIS measurement covariance typically will be much smaller than the radar measurement covariance, leading to a fused state very close to the AIS measurement. The pure AIS hypothesis will use its predicted state, covariance and score, somewhat similar to radar zero hypotheses.

4.6 Clustering

The problem of finding the globally optimal set of track hypotheses increases exponentially with the number of hypotheses in the problem. To reduce the size of the problem, it is desirable to split it into smaller independent problems. Both because it enables parallel computation and it reduces the total cost of solving the problem. Track trees that have common measurements must be solved together, since they can have mutual exclusive leaf nodes. The clustering can be done efficiently through Breath First Search (BFS) or Depth First Search (DFS) on a graph made from the track hypothesis tree.

By constructing a 0–1 adjacency matrix describing the connection between all the nodes in the track forest, the clustering problem is equivalent to the *connected components* problem in graph theory [29].

4.7 Optimal data association

The aim of this section is to elaborate the use of ILP to solve the data association problem in MHT that arises when there are multiple, possibly mutual exclusive, possibilities of measurement arrangements within the existing set of tracks.

When the targets are divided into independent clusters, each of them can be treated as a global problem where we want to minimize the cost or maximize the score of the selected track hypotheses (leaf nodes). The selected track hypotheses must also fulfil the constraints, that each measurement can only be a part of one track, and that minimum and maximum one track hypothesis can be selected from each target. Since only binary values, selected or not selected, is possible for selection of hypotheses, the problem becomes an ILP. In the case where a cluster is only containing one target tree, the best hypothesis can be selected by running a search among the leaf nodes after the highest score, since none of the leaf nodes are excluding other leaf nodes in other target trees. This will often be the case for targets that are largely spaced out, and their gates are not and have not overlapped in a while. For any other case, where there are two or more targets in a cluster, the procedure in Section 4.7.1 must be carried out.

4.7.1 Integer Linear programming

The essence of any optimization problem is a cost function and a set of constraints. In our problem, we want to select the combination of hypotheses (leaf nodes) that gives the highest score / lowest cost, while not selecting any measurement more than one time and ensure that we select minimum and maximum one hypothesis from each target.

Our score function is the sum of the selected node scores, and since we are using negative logarithmic scores the goal is to minimize the overall score, making the problem a minimum cost problem. Our cost vector $\mathbf{c}$ is made up from the scores of all the leaf nodes in the forest build by the trees clustered together, arranged in the order they are visited by a DFS. The accompanying selection vector $\boldsymbol{\tau}$ is of the same dimension with boolean values, where the selected hypotheses are value 1 and all other is value 0. These two together form the objective function (4.21).

$$\min_{\boldsymbol{\tau}} \quad \mathbf{c}^T \boldsymbol{\tau} \tag{4.21}$$

To ensure that we are selecting the same measurement maximum one time, a binary matrix $\boldsymbol{A_1}$ describing the association between nodes and measurements are created.$\boldsymbol{A_1}$

has as many rows are there are real measurement and as as many columns as there are hypotheses. All radar- and AIS measurement are real measurement, whereas zero nodes does not contain any real measurements. $\boldsymbol{A_1}$ has value 1 in the intersections of hypotheses and measurements that are associated, 0 elsewhere. The order of the columns in $\boldsymbol{A_1}$ is the same as the rows in $\mathbf{c}$ and $\boldsymbol{\tau}$. Since we are only limiting a maximum of one usage of each measurement and no minimum, the constraint becomes an inequality constraint (4.22) where $\mathbf{1}$ is a vector of ones with the same dimension as $\boldsymbol{\tau}$.

$$\boldsymbol{A_1}\boldsymbol{\tau} \leq \mathbf{1} \tag{4.22}$$

The second constraint we need to impose is that we need to select minimum and maximum one hypothesis from each tree. This can be done by creating a boolean matrix $\boldsymbol{A_2}$ which describes the relationship between hypotheses and trees / targets. $\boldsymbol{A_2}$ will have as many rows as there are targets in the cluster, and columns as $\boldsymbol{A_1}$. The intersections between hypotheses and targets that belong together is value 1, all other is value 0. Since this constraint has a fixed requirement, it is formulated as an equality constraint (4.23) with $\mathbf{1}$ as in (4.22).

$$\boldsymbol{A_2}\boldsymbol{\tau} = \mathbf{1} \tag{4.23}$$

The complete ILP formulation becomes (4.24), where $\boldsymbol{\tau}$ is a binary vector with dimension equal the number of leaf nodes in the track forest.

$$\begin{aligned} \max_{\boldsymbol{\tau}} \quad & \mathbf{c}^T\boldsymbol{\tau} \\ \text{s.t.} \quad & \boldsymbol{A_1}\boldsymbol{\tau} \leq \mathbf{b_1} \\ & \boldsymbol{A_2}\boldsymbol{\tau} = \mathbf{b_2} \\ & \boldsymbol{\tau} \in \{0,1\}^M \end{aligned} \tag{4.24}$$

An example based on Figure 4.10 at time step 2, where the $\boldsymbol{A}$ matrices and $\mathbf{C}$ vector would be (4.25).

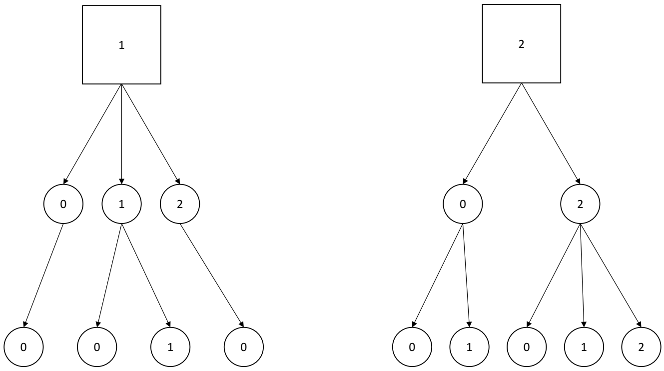

Figure 4.10: Track hypotheses forest

$$
\begin{aligned}
\boldsymbol{A_1} &= \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{b_1} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \\
\boldsymbol{A_2} &= \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \mathbf{b_2} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\
\mathbf{c} &= \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 & \lambda_6 & \lambda_7 & \lambda_8 & \lambda_9 \end{bmatrix}^T
\end{aligned} \tag{4.25}
$$

4.7.2 Solvers

The problem (4.24) is formulated on standard form, which enables the use of existing off-the-shelf ILP solvers. There are a lot of off-the-shelf ILP and Mixed Integer Linear Programming (MILP) solvers on the marked, both free open source and commercial. The performance difference of some solvers where tested in [2], where the difference where found marginal, most likely because each optimization problem is relatively small and the initialization and preprocessing of the solver and problem played a significant part of the runtime compared to the actual solving. In this work, Google Optimization Tools is used as interface between the programming language and the solver. The default solver CBC was used exclusively in this work as its performance was on par with the others tested in [2].

4.8 Dynamic window

For any MHT to be realistic over time it need to have a sliding window removing the unused hypotheses N steps back in time. The sliding window size (N) could be a static design parameter or a function of the runtime of that tree, which reflects the overall size of the tree. This enables the system to adapt its core parameters to guarantee its runtime demands.

4.9 N-Scan pruning

To keep the computational cost within reasonable limits, it is necessary to limit the amount of time steps backwards in time that the algorithm computes. This is done by removing all branches but the active track hypothesis at the current root node, and assign the one remaining node as new root node. This procedure is graphically explained in Figure 4.11, where a solid square frame indicates the current root node, and a dotted square frame indicates the new root node. The bold arrows in the figure represents the active track.

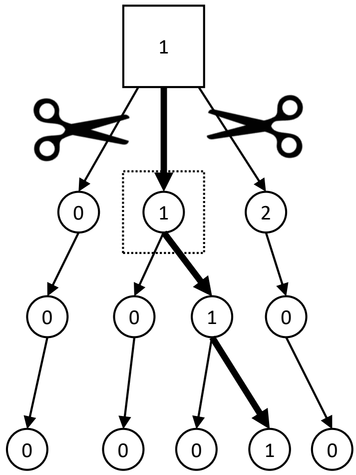

Figure 4.11: N-scan pruning

4.10 Track termination

Since targets can disappear from the observation region both by leaving the radar range and by driving behind objects that puts them in a radar shadow, it is necessary to terminate these tracks. It is also desirable to terminate falsely initiated tracks as soon as possible, since a guidance system would steer clear of any objects reported to it. The first scenario, where the target is leaving the radar range can easily be detected and the track can be terminated quickly based on the predicted position of the target relative to our own position. The second scenario, can be approached in different ways depending on the available data and computational power. The simplest solution is to terminate all tracks where the selected node after each iteration have a score higher that a threshold. This could terminate tracks with consecutive miss detections, which is desirable for false tracks, but can lead to premature termination of true targets with temporarily low P_D. This means that the termination threshold becomes a trade-of between killing false tracks and keeping targets with lower P_D and shadowed targets.

A more advanced approach could be to utilize map data to estimate whether or not a target is in a radar shadow of land objects, and then make a decision on whether this target should be given a lower P_D temporary or terminated based. This could also be done between targets if target extent is estimated, where targets behind other targets are given a lower P_D to punish miss detections less.

In this work, only range and score termination is implemented.

4.11 Track smoothing

After each iteration of the MHT algorithm a track list based on the selected hypotheses from Section 4.7 is passed forward to the operator and guidance system. Both human operators and guidance systems will try to predict the targets behaviour based on their historical track. To improve the visualisation for this purpose it is possible to smooth the tracks based on the real measurements in the track and masking the zero measurements in a Kalman smoother [30] with the same model as used in the predictions. As illustrated in Figure 4.12, this will lead to a smother and in most cases a more accurate representation of the true track since it avoids the straight lines caused by dead reckoning. This smoothing will however not affect the tracking performance since it is done after miss detections are corrected and only on the track list sent out of the tracking module.

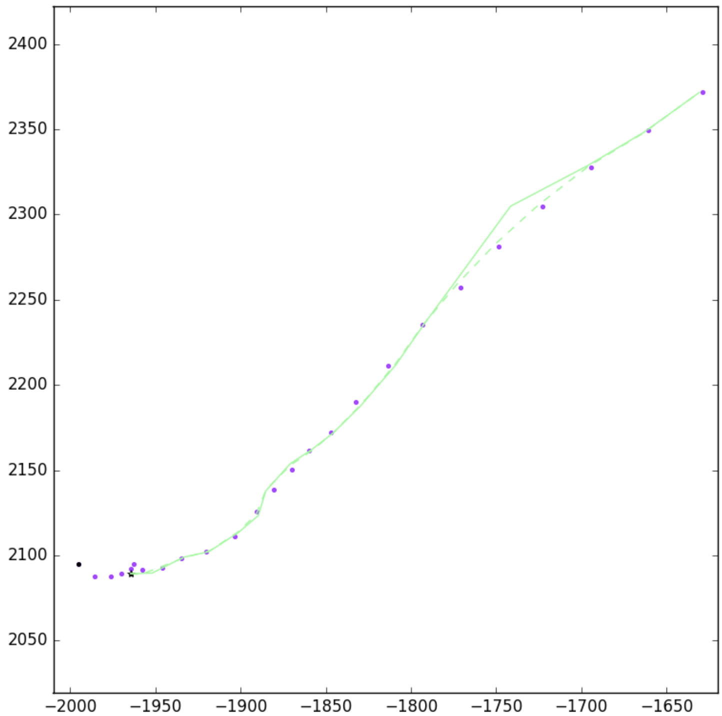

Figure 4.12: Track smoothing

Chapter 5

Results

The performance evaluation of any tracking system difficult since the degrees of freedom are very large and there are no single of few obvious performance metrics. There are however two distinct found in the literature, pure Monte Carlo testing and situation / scenario testing. The first being that parameters like number of tracks, start position, velocity, manoeuvring, missed detections and clutter all are randomly selected and repeated many times. From a pure numerical point of view this approach seems reasonable, but it does not necessary create realistic tracking scenarios for observation dimensions higher that one. The second approach to testing is to create one or more scenarios which then is simulated with random variables like detection and noise. This approach is vulnerable to the created scenarios, since the design can heavily impact the measured performance. This method allows also for construction of very specific situations where it is desirable to test multiple tracking system on the same custom created situation for comparison purposes.

5.1 Testing scheme

The performance evaluation of any MHT system is tedious in that it is necessary to test very many different situations to get a good understanding of how the system is performing. The two largest factors contributing to the difficulty is the random nature of the clutter and lost detections. It is also desirable to evaluate the initialization and tracking performance under both varying environmental (external) conditions and tuning (internal) setting. We want good tracking of targets with low probability of detection in cluttered environment, and secondly it must be able to do this within the time frame of the radar rotation period. The initialization module must be able to detect targets with probability of detection lower than unity without initializing too many false tracks into the MHT algorithm. The testing is separated into two parts; initialization and tracking.

The performance metrics for the initialization module is how long time it takes to initialize the correct tracks, which is tested under a range of internal and external conditions, see (5.1). All combinations of these parameters were simulated on all scenarios (Table 5.2), which are the same routes but with different AIS configurations. From these simulations, the time to initiate true targets and amount of erroneous targets are calculated. A track is categorized as correct initialized if the state difference between the true track and the initial track is less that a threshold. All initial tracks that does not correspond to a true track is categorized as erroneous. To analyse the impact of the erroneous tracks, the lifespan of falsely initiated tracks is plotted to see whether they die out at the same rate as they are initiated, or if they accumulate.

$$\begin{aligned}
\mathbf{P_D} &= \begin{bmatrix} .9 & 0.8 & 0.6 \end{bmatrix} \\
(m/n) &= \begin{bmatrix} (1/1) & (1/2) & (1/3) & (1/4) \\ (2/2) & (2/3) & (2/4) & (2/5) \\ (3/3) & (3/4) & (3/5) & (3/6) \end{bmatrix} \\
\boldsymbol{\lambda}_{\phi} &= \begin{bmatrix} 0 & 5 \cdot 10^{-6} & 1 \cdot 10^{-5} \end{bmatrix}
\end{aligned} \tag{5.1}$$

When testing the tracking performance, it is desirable to remove the variable of initialization to better see difference in *tacking* rather than *initialization*. All simulations testing tracking performance are carried out with all targets correctly initialized at initial time, and with the initiator set to $(2/4)$ such that the unused measurements from the tracking algorithm would be treated as normal. This would also give lost targets a change to get re-initialized, which is an important property for any safety critical system.

$$\begin{aligned}
\mathbf{P_D} &= \begin{bmatrix} .9 & 0.8 & 0.6 \end{bmatrix} \\
\mathbf{N} &= \begin{bmatrix} 1 & 3 & 6 & 9 \end{bmatrix} \\
\boldsymbol{\lambda}_{\phi} &= \begin{bmatrix} 0 & 5 \cdot 10^{-6} & 1 \cdot 10^{-5} \end{bmatrix}
\end{aligned} \tag{5.2}$$

Since the targets are initialized perfectly in every situation, we are interested in how good our system is able to *keep* on the tracks. We measure this by means of the Euclidean distance between the estimated and true track (5.3).

$$\Delta P = \|\mathbf{p}_{track} - \mathbf{p}_{target}\|_2 \tag{5.3}$$

The first track performance metric is the track loss percentage, where a track is considered correct if $\Delta P \leq \varepsilon_p$ for all t after initial convergence. If a track is deviating more than the threshold and never return within the threshold again, it is considered lost at the time-step when it exceeded the threshold. If the track should converge after exceeding the threshold, it is considered restored at the time-step it is returning within the limit.

The second and closely related metric is the tracking percentage, where the total time a target is correctly tracked is summed up and compared with the existence time of the target. This station is illustrated in Figure 5.1 where a track is lost at (1000,1250), and re-initialized at (750,1800). The track is considered lost at (1000,1250), while the tracking percentage is also accounting the last track from (750,1800).

The third metric is how close the estimated track is to the true track on average. This is measured as the square root mean square deviation of the entire track (5.4).

$$RMSD = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\Delta P_t)^2} \tag{5.4}$$

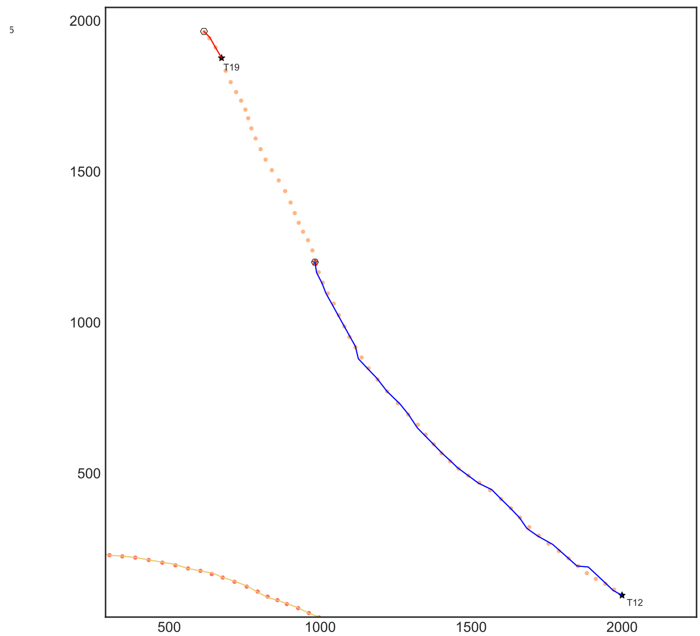

Figure 5.1: Tracking percentage illustration

5.2 Scenario

All simulations in this work is based on a generated scenario, as shown in Figure 5.2, with black dots marking the initial time and position. The radar range is 5500 meter (3 Nautical Miles (NMs)), which gives an area of surveillance of approximately 95 square km. The scenario contains 16 targets, which all starts inside the observable area of the radar. The scenario contains a mixture of fast and slow moving vessels, some with sharp turns and some almost at stand still. Table 5.1 shows the initial states of all targets, and the true path is generated once from these initial values.

From this base scenario, five scenarios where generated with different AIS configuration on the vessels, see Table 5.2. The first scenario represent the baseline with only radar information available, whereas the rest have some level of AIS information. Scenario 1 adds three class B AIS transmitters, and is representing a situation where all the targets are smaller vessels with some voluntarily installed AIS transceivers. In scenario 3, all vessels have AIS class B installed. This scenario represents a best case situation regarding yacht and leisure vessels from an autonomous anti collision perspective and is only realistic if AIS class B where to be mandatory for these vessel classes. Scenario 2 is the same as scenario 1, with the difference that the vessels have class A transmitters in stead of class B. This gives them higher and smarter rate of transmission, which in theory should improve tracking under challenging conditions. This scenario can be viewed as a few commercial vessels travelling in between a large group of yachts. The last scenario, where all targets are equipped with class A transmitters is the ultimate situation for any fusion tracking system. This case would be realistic in a crowded professional working area, for instance harbours, fishing areas and off-shore installations.

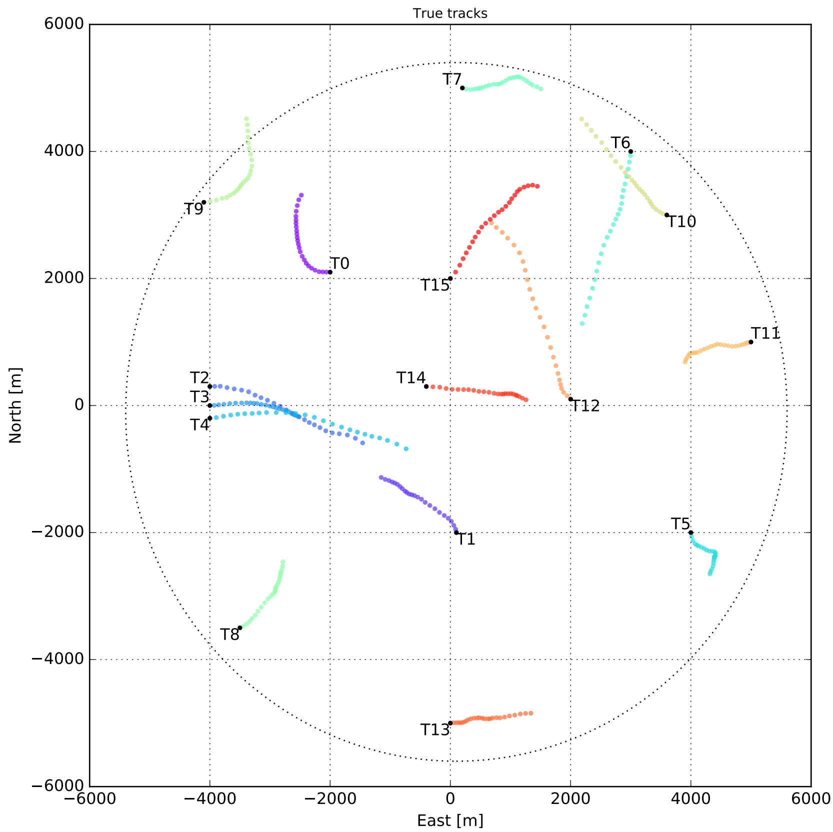

Figure 5.2: True tracks

Target	North	East	North speed	East speed
0	2100.0	-2000.0	10.0	270.0
1	-2000.0	100.0	8.0	-2.0
2	300.0	-4000.0	-1.0	12.0
3	0.0	-4000.0	12.0	90.0
4	-200.0	-4000.0	1.0	17.0
5	-2000.0	4000.0	-8.0	1.0
6	4000.0	3000.0	-8.0	2.0
7	5000.0	200.0	-1.0	10.0
8	-3500.0	-3500.0	5.0	10.0
9	3200.0	-4100.0	2.0	17.0
10	3000.0	3600.0	3.0	-10.0
11	1000.0	5000.0	-2.0	-7.0
12	100.0	2000.0	8.0	-10.0
13	-5000.0	0.0	2.0	10.0
14	300.0	-400.0	0.0	17.0
15	2000.0	0.0	15.0	15.0

Table 5.1: Initial states

T	S0	S1	S2	S3	S4
0	-	B	A	B	A
1	-	-	-	B	A
2	-	B	A	B	A
3	-	-	-	B	A
4	-	B	A	B	A
5	-	-	-	B	A
6	-	B	A	B	A
7	-	-	-	B	A
8	-	B	A	B	A
9	-	-	-	B	A
10	-	B	A	B	A
11	-	-	-	B	A
12	-	B	A	B	A
13	-	-	-	B	A
14	-	B	A	B	A
15	-	-	-	B	A

Table 5.2: AIS scenario configuration

5.3 Simulation

Both initialization- and tracking performance is averaged over a set of 50 Monte Carlo simulations with differently seeded clutter- and missed detections points. All simulations were done with a sampling interval at 2.5 seconds (24 RPM), which is the most common rotation speed on a coastal maritime radar. Since early simulations showed that the variation between the different scenarios when analysing the initialization time and termination of false track were negligible. Based on this, only scenario 0 were used for initialization analysis to avoid excessive plot 'duplicates'. Each of the 144 initialization variations and 240 track performance simulations where run 50 times with differently seeded clutter and misdetections on a dual Intel i7−6700 server running Linux Ubuntu with Solid State Storage (SSD) storage and 64 GB Random Access Memory (RAM).

5.4 Initialization results

5.5 Tracking results

Chapter 6

Discussion

Discussion here.

6.1 Alternative design / open questions

What to do when a track previously associated with an MMSI no longer has new AIS measurements from that MMSI inside its gate? Make an extra hypothesis with a jump? An how to score this?

What to do with unused AIS measurements, i.e. ones that are not previously associated with any tracks and are not within any gate? Use them as measurements in the initialization procedure? Initiate them as new tracks immediately?

In what ways can AIS meta-data improve the radar tracking? From AIS length information, it might be possible to estimate the maximum turning rate for a vessel.

Chapter 7

Future work

Chapter 8

Conclusion

Conclusion here.

Appendix A

Initialization time plot

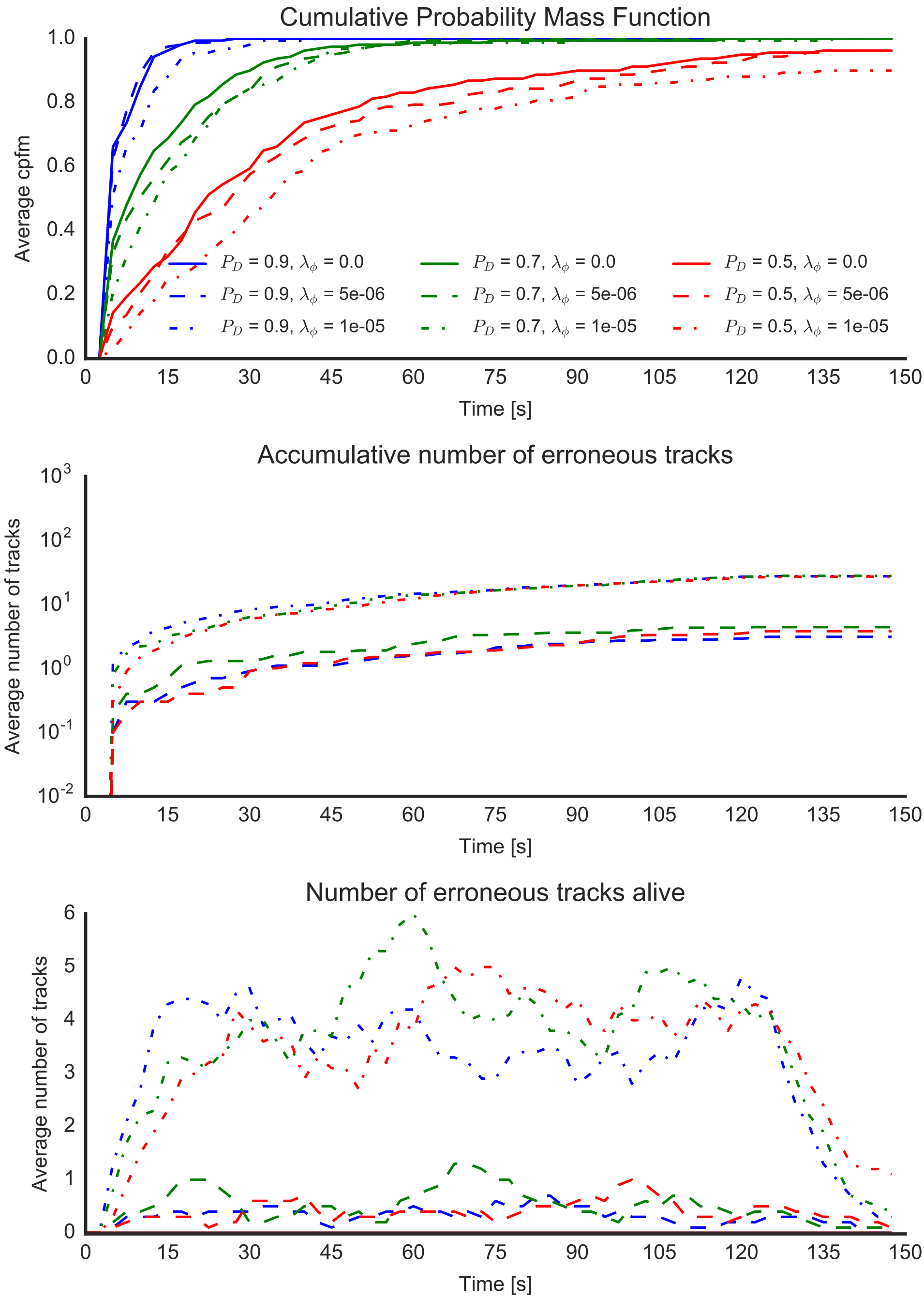

Figure A.1: Initialization time (1/1)

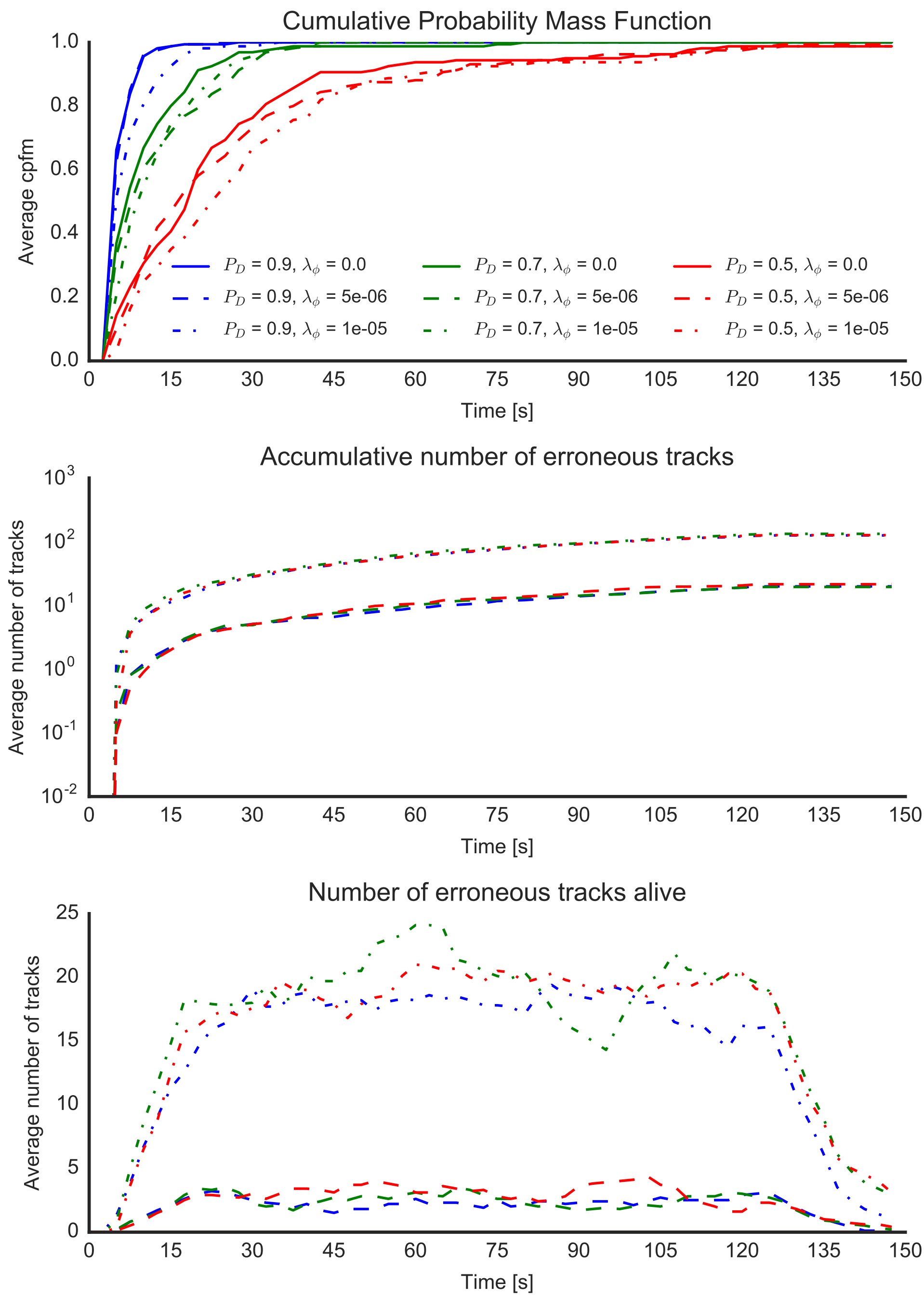

Figure A.2: Initialization time (1/2)

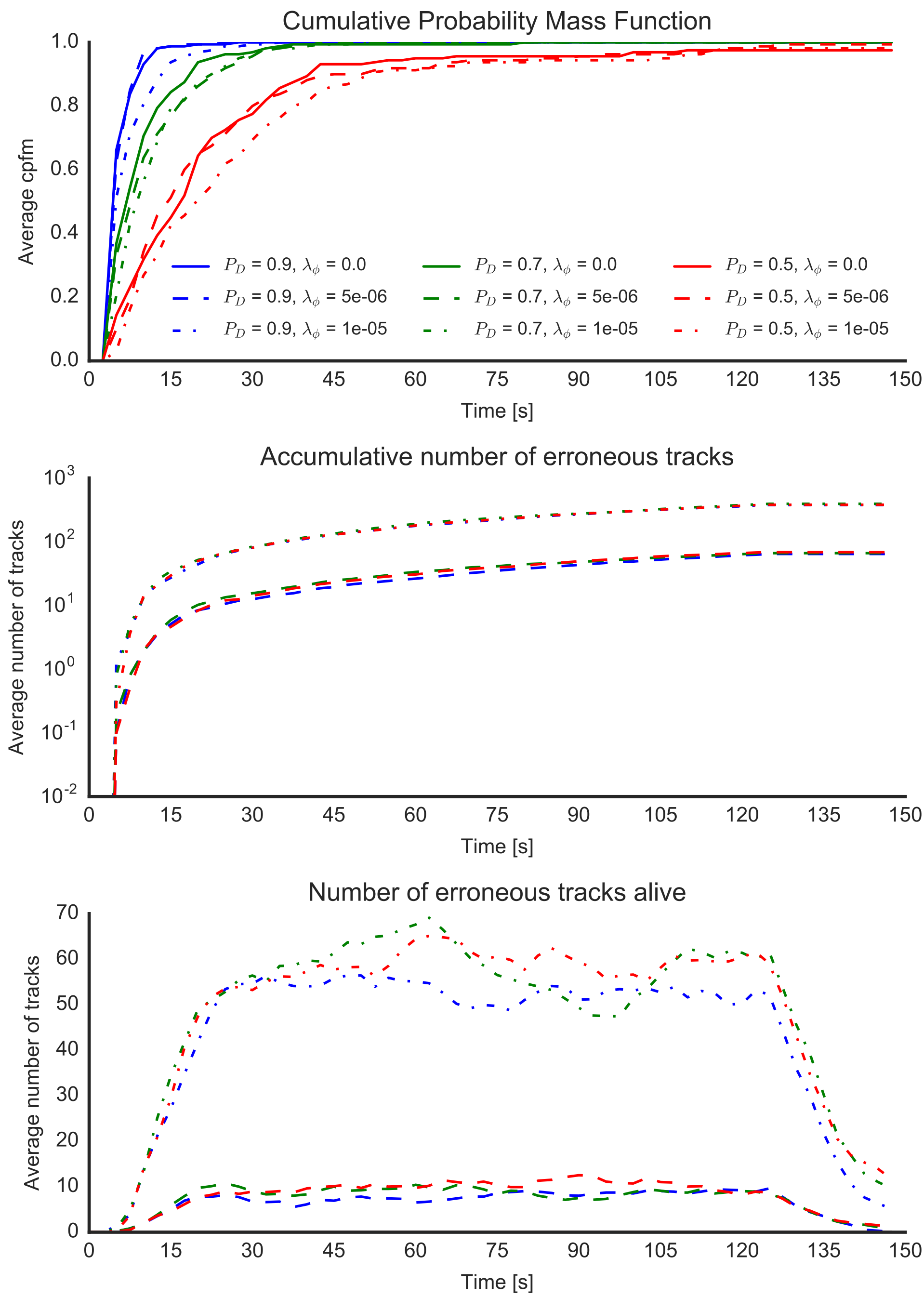

Figure A.3: Initialization time (1/3)

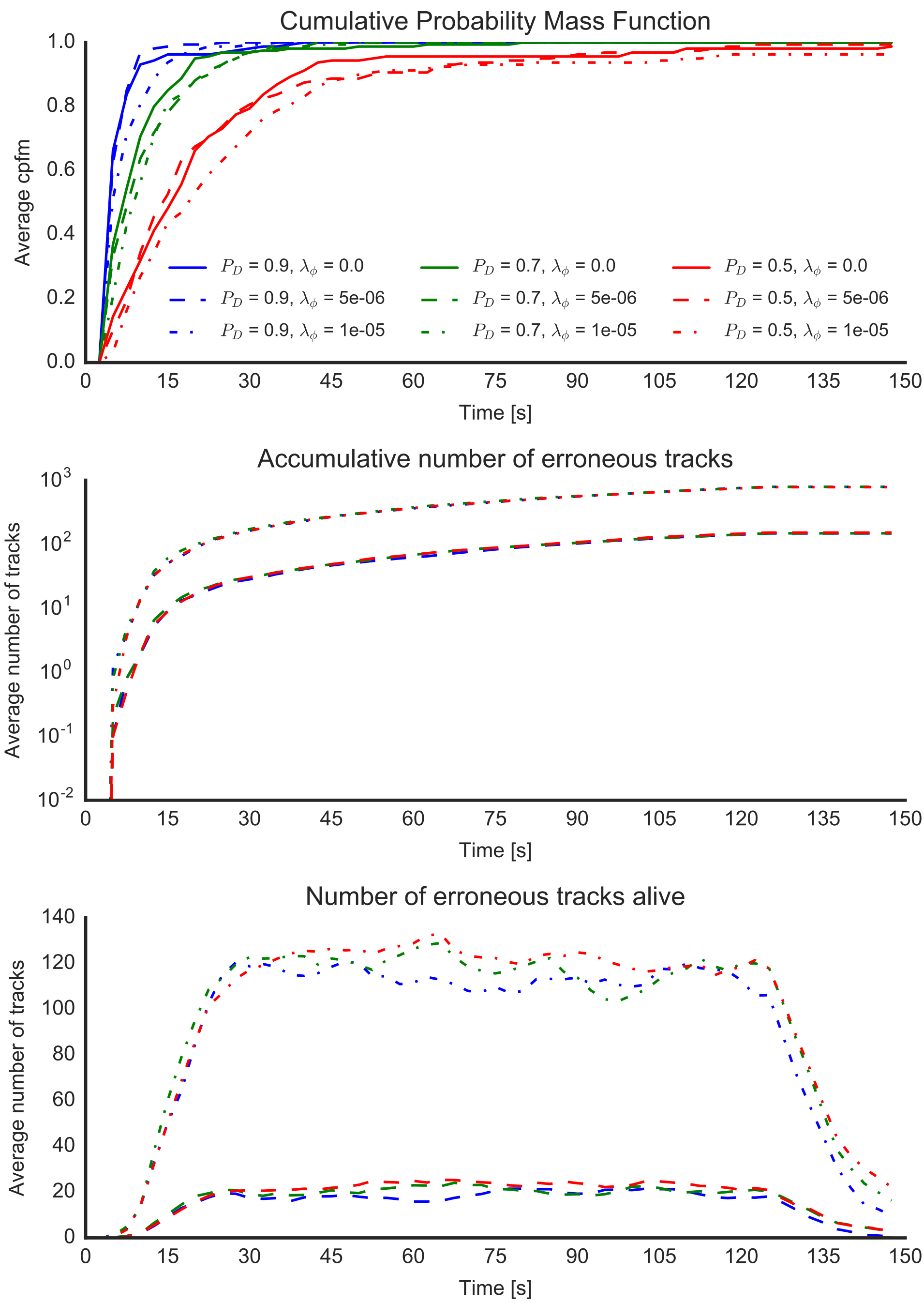

Figure A.4: Initialization time (1/4)

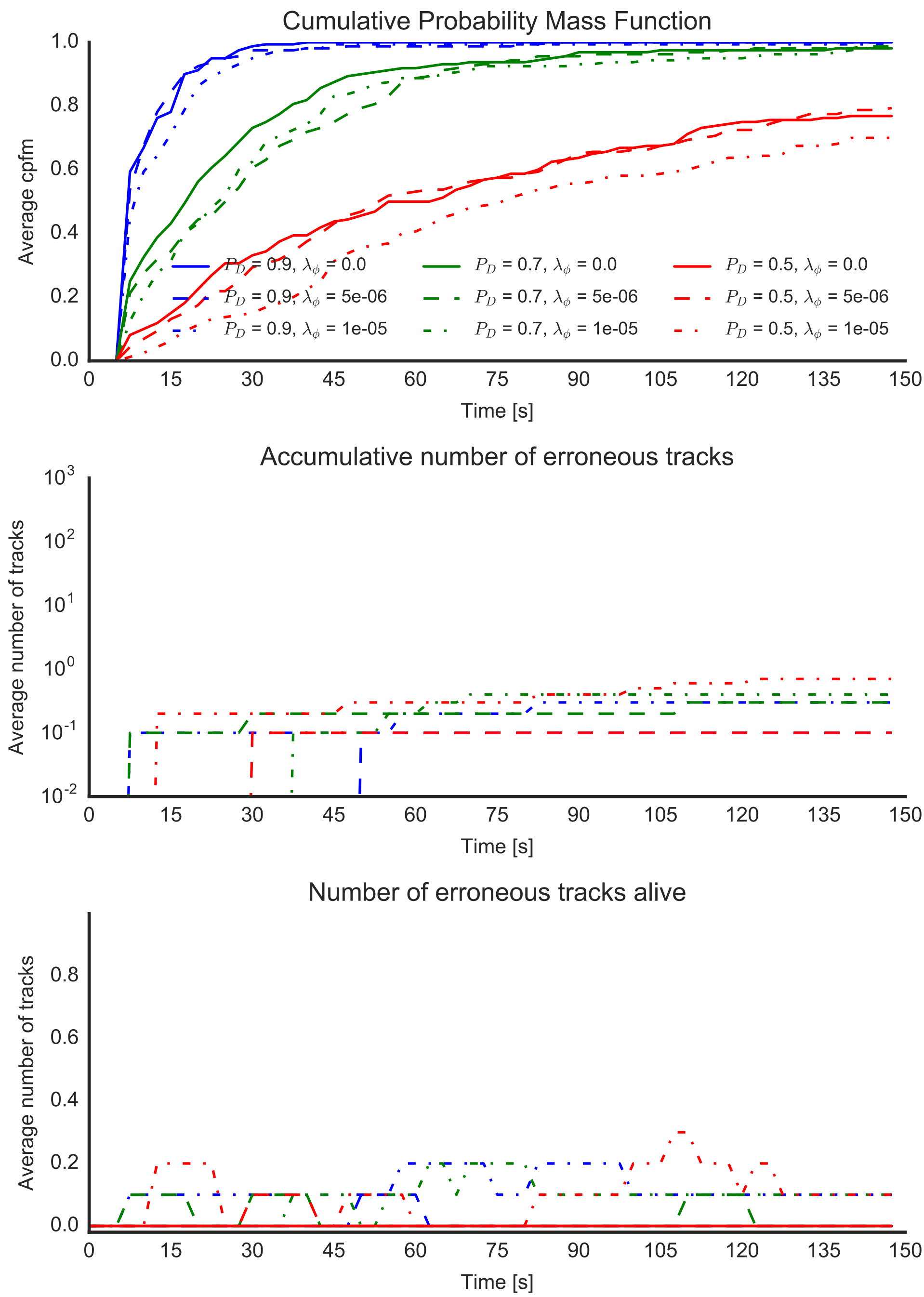

Figure A.5: Initialization time (2/2)

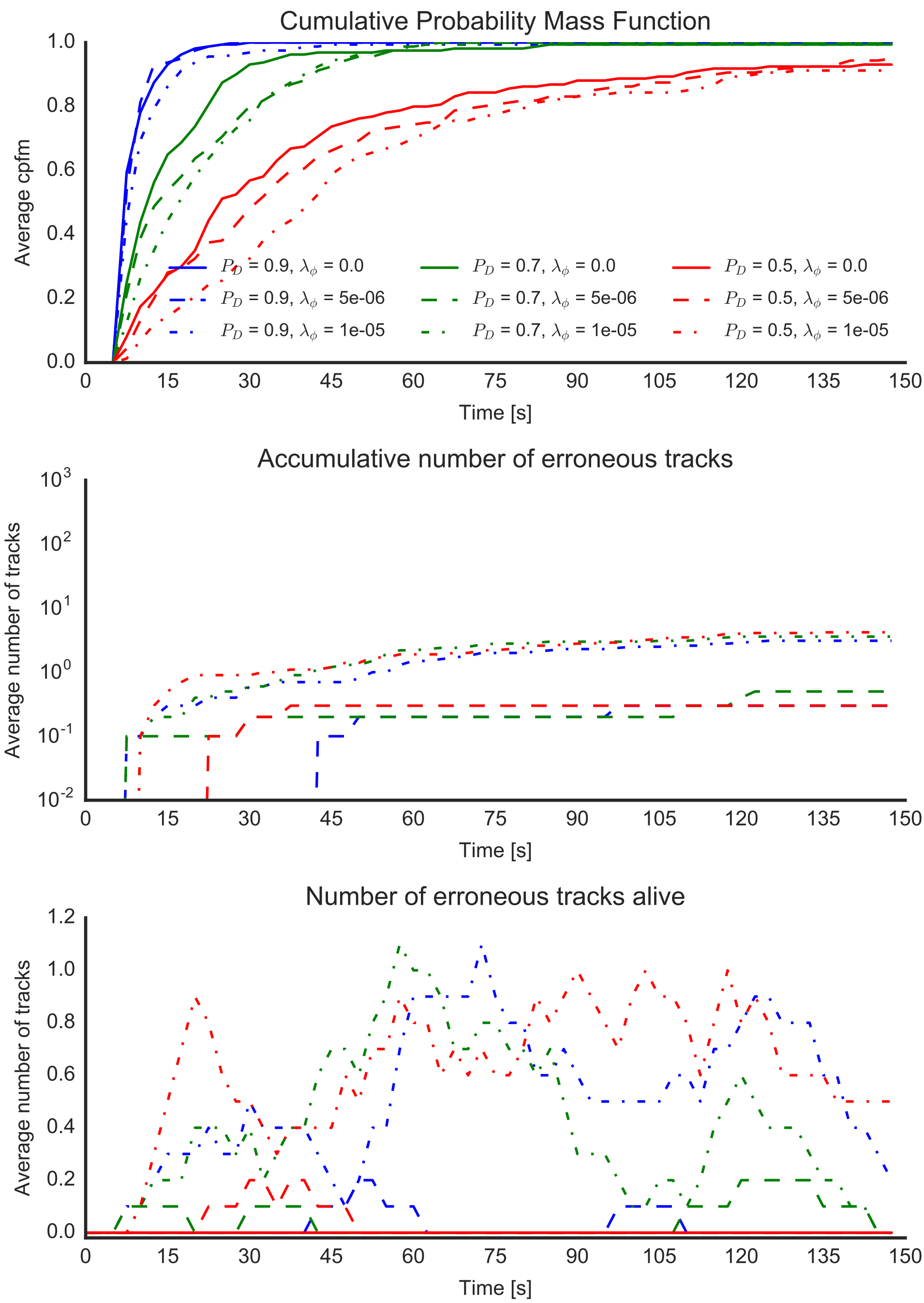

Figure A.6: Initialization time (2/3)

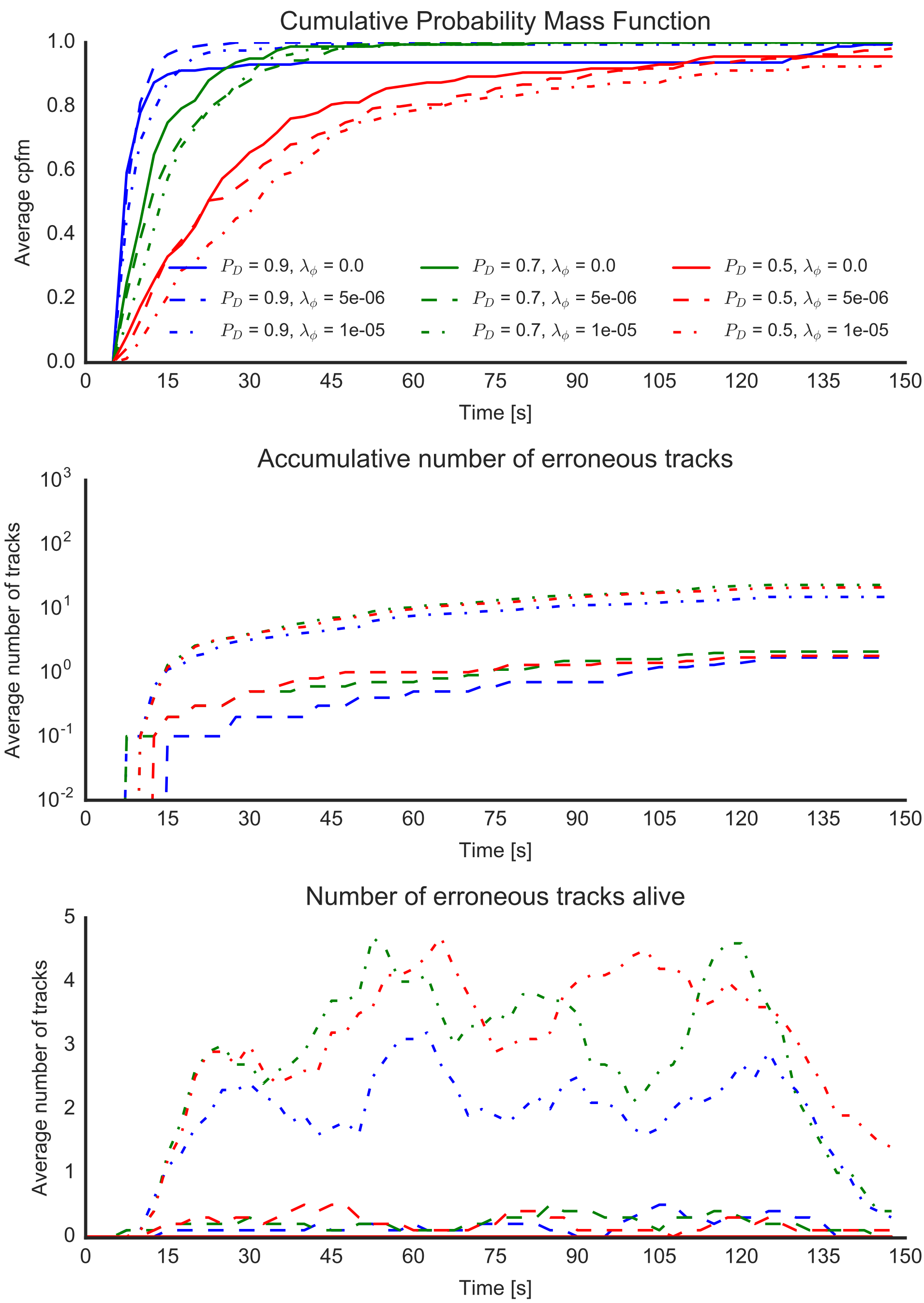

Figure A.7: Initialization time (2/4)

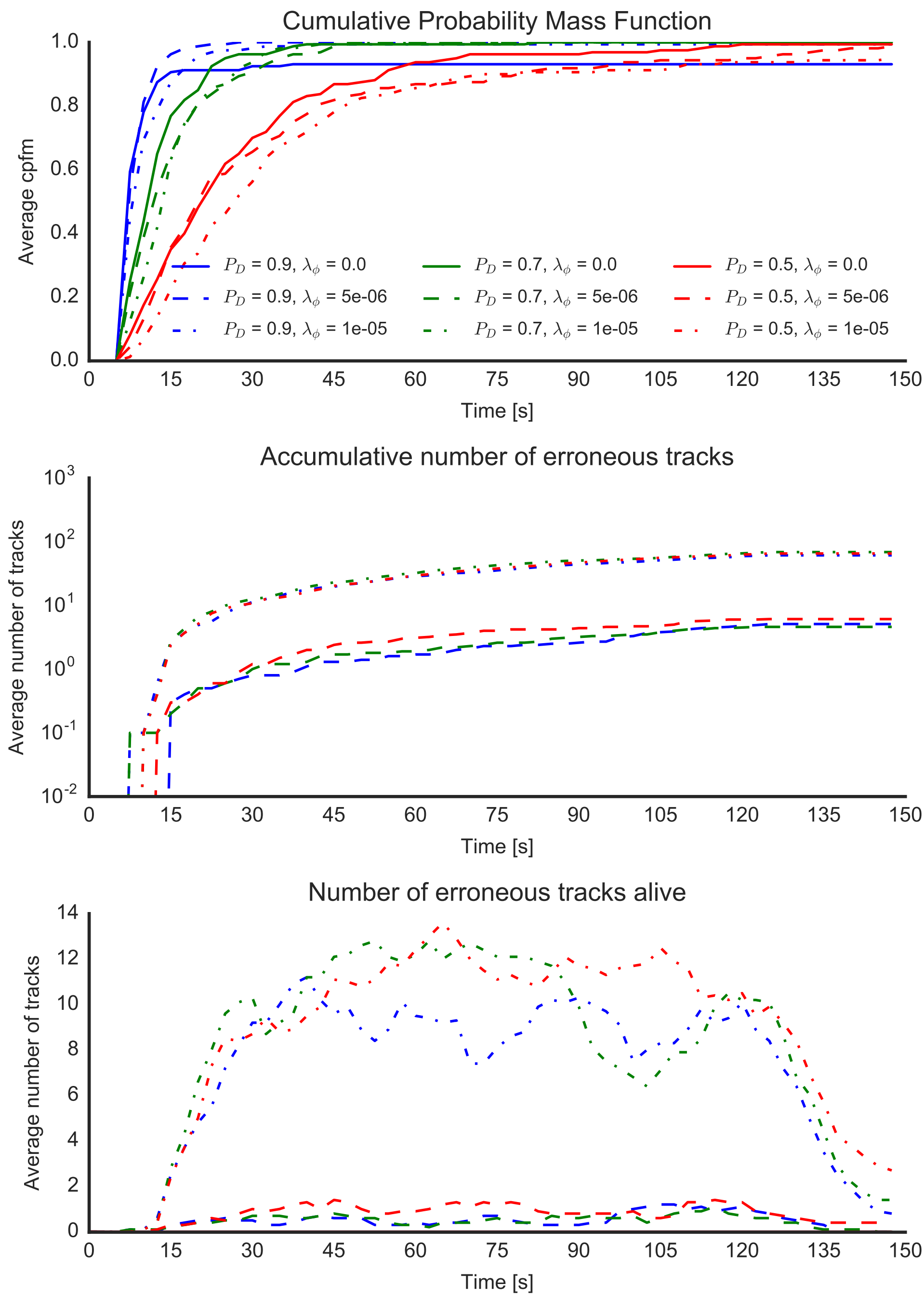

Figure A.8: Initialization time (2/5)

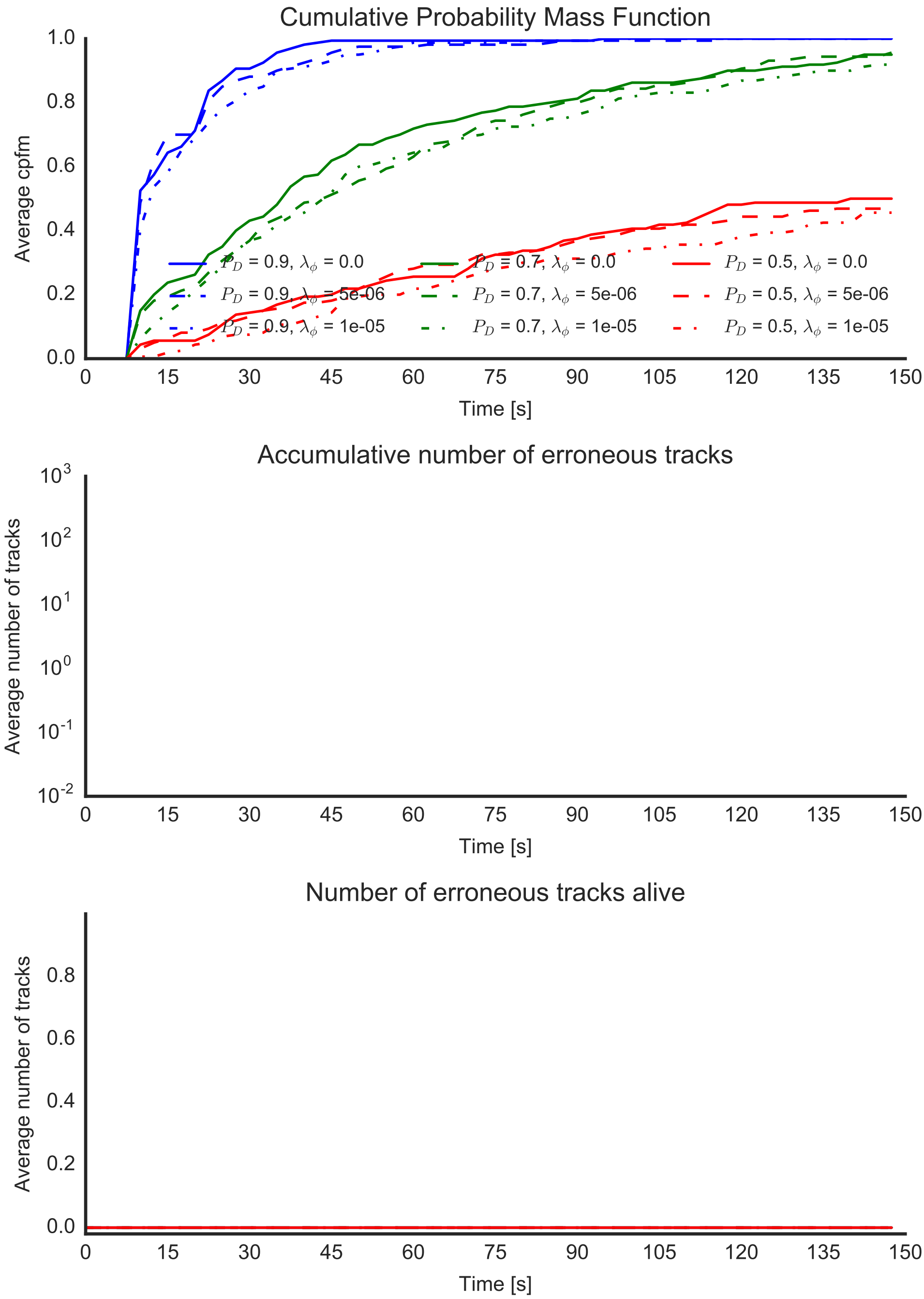

Figure A.9: Initialization time (3/3)

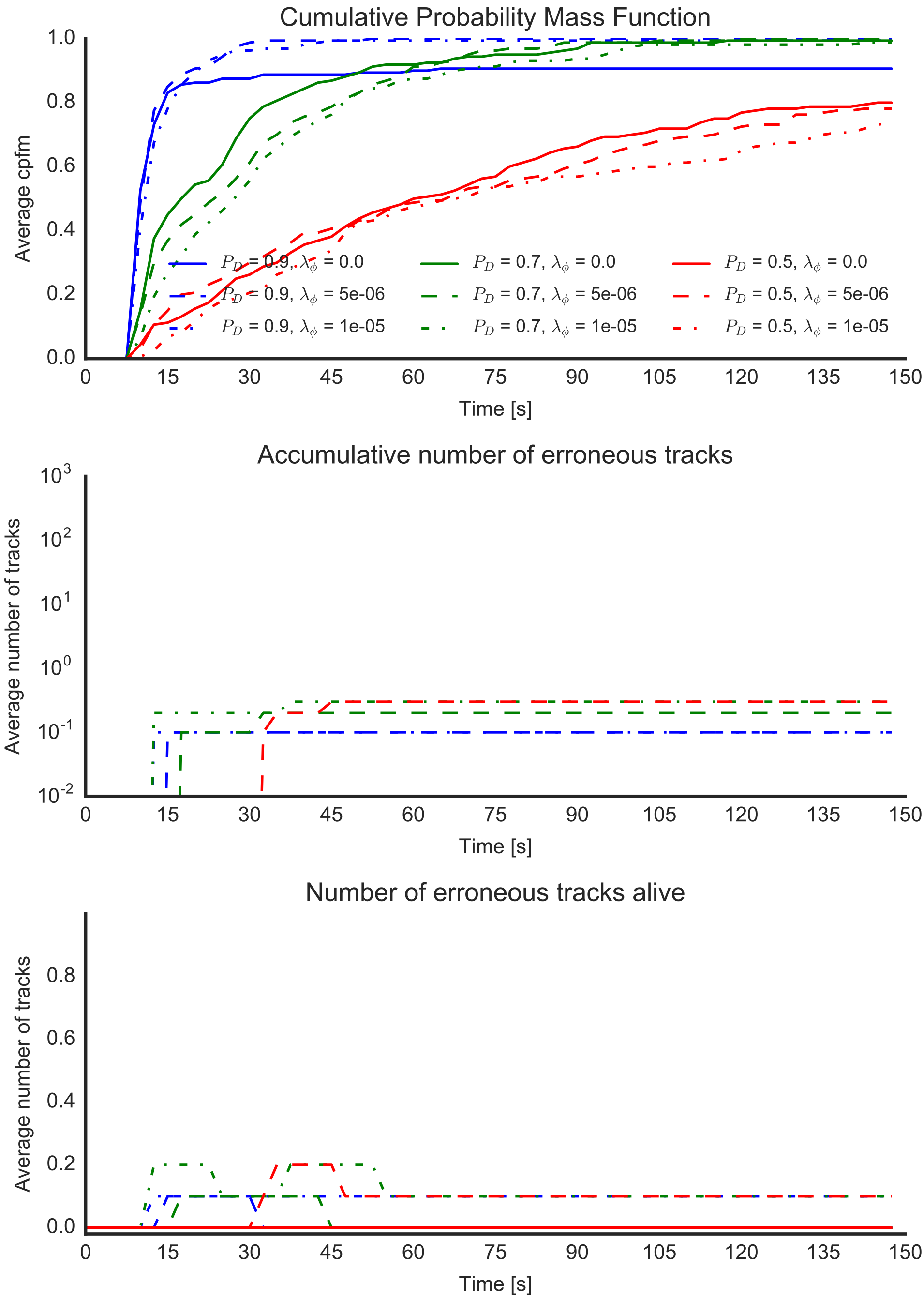

Figure A.10: Initialization time (3/4)

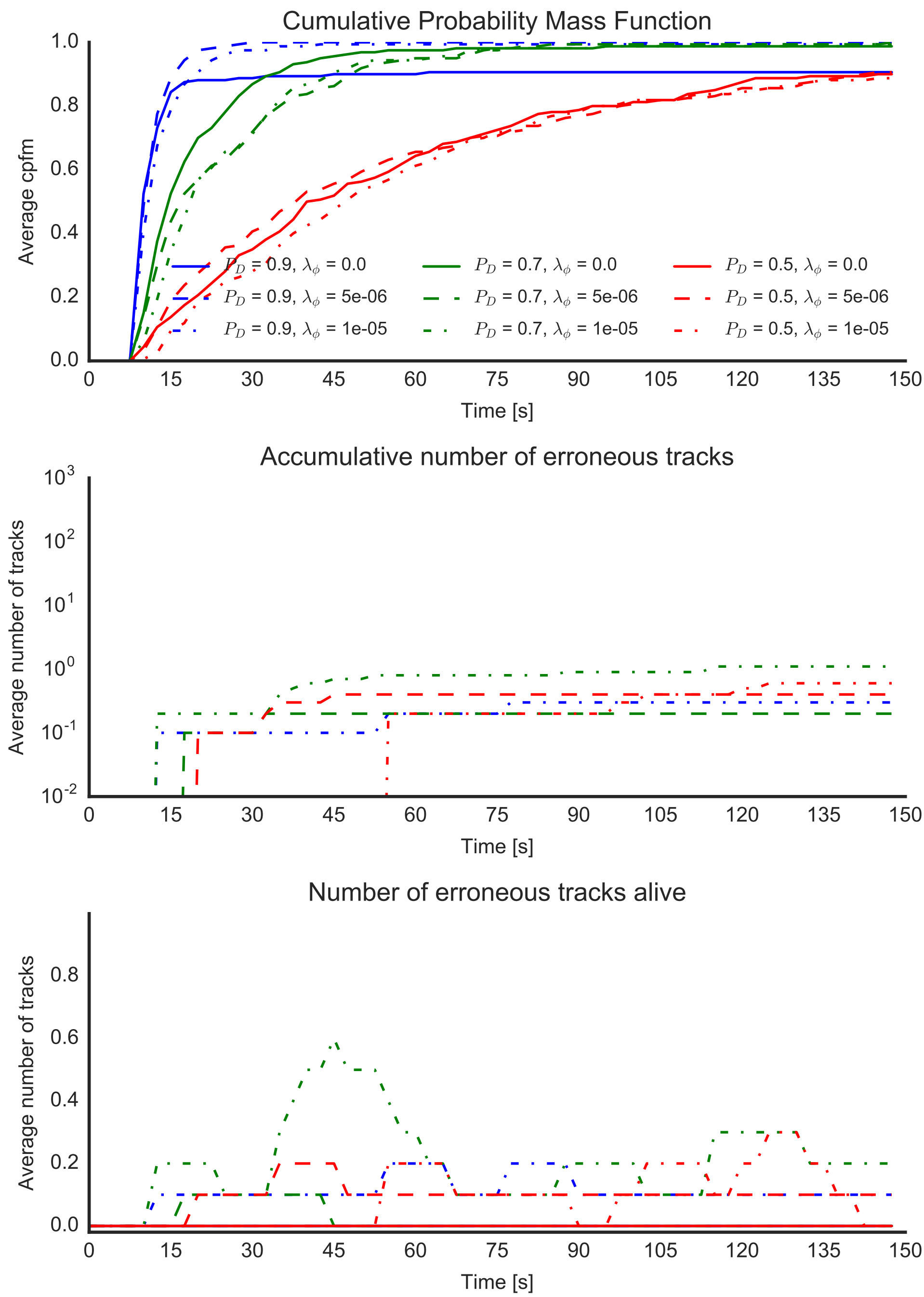

Figure A.11: Initialization time (3/5)

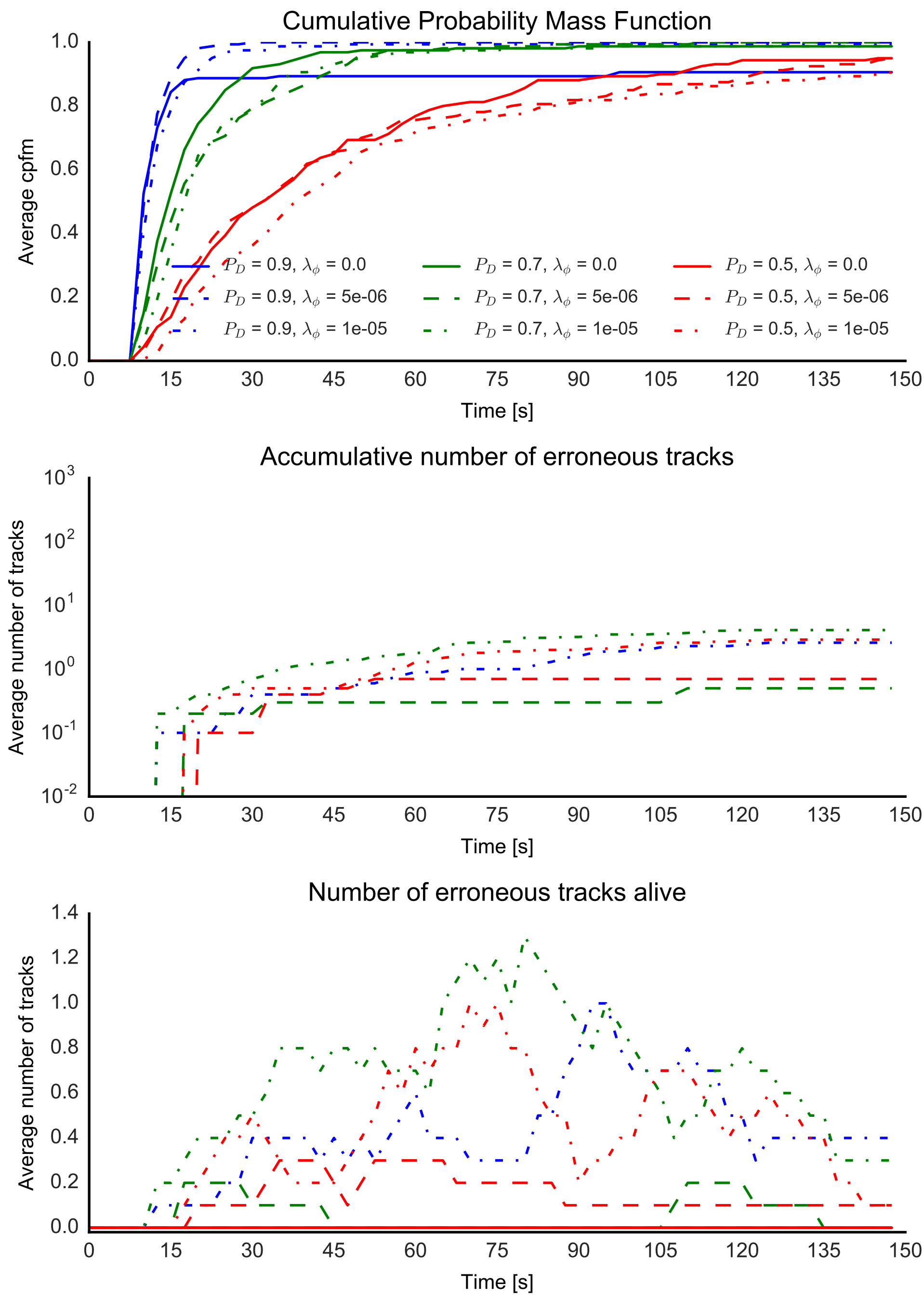

Figure A.12: Initialization time (3/6)

Appendix B

Track loss plot

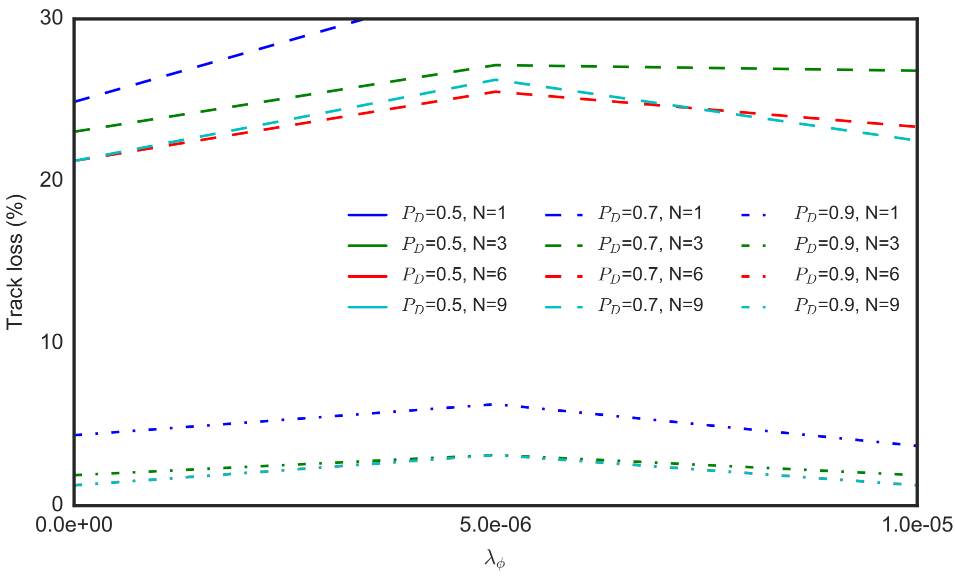

Figure B.1: Scenario 0 — Track loss

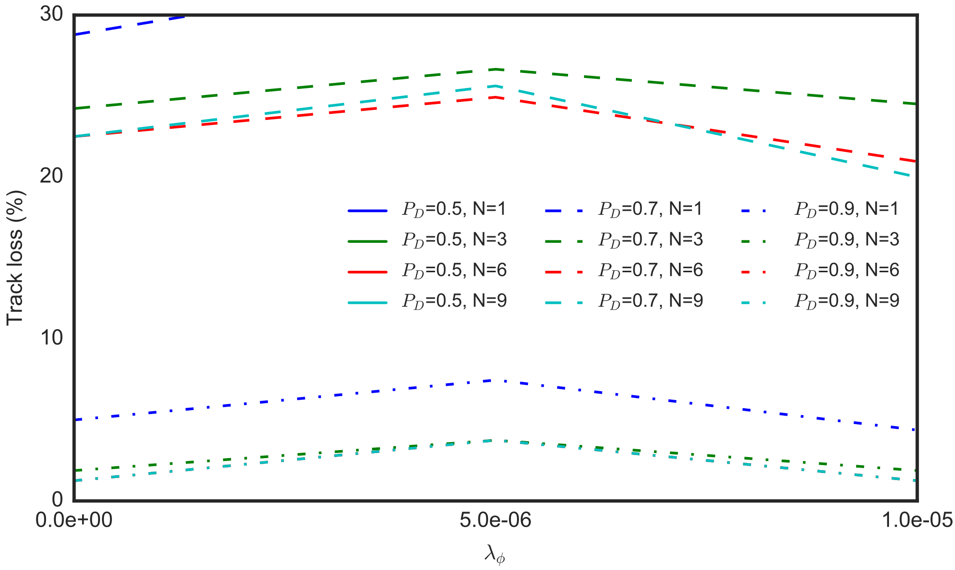

Figure B.2: Scenario 1 — Track loss

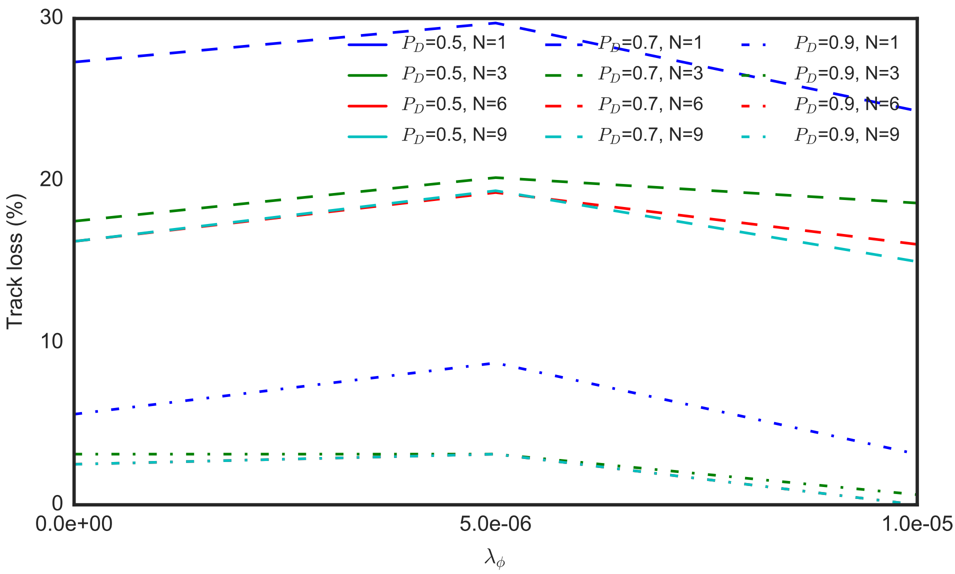

Figure B.3: Scenario 2 — Track loss

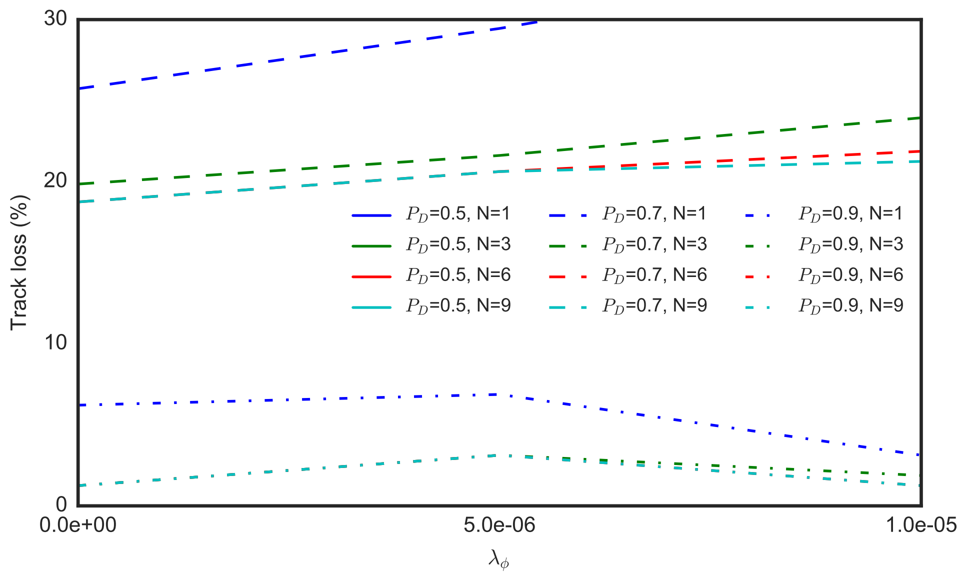

Figure B.4: Scenario 3 — Track loss

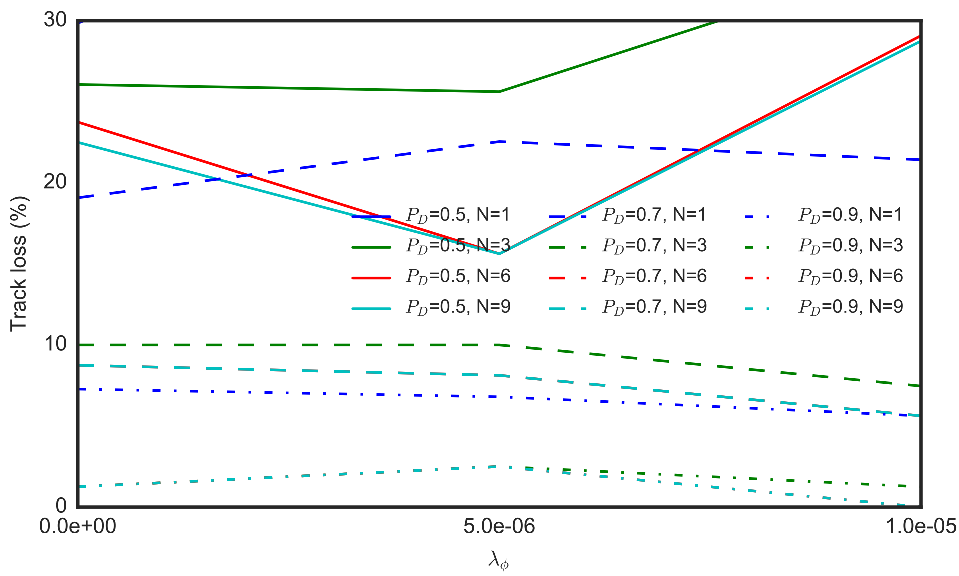

Figure B.5: Scenario 4 — Track loss

Appendix C

Tracking percentage plot

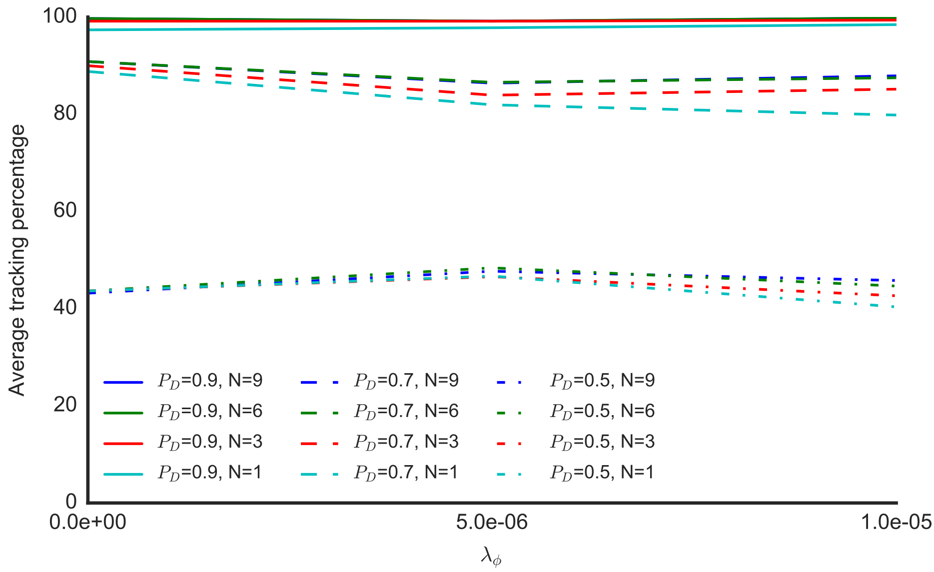

Figure C.1: Scenario 0 — Tracking percentage

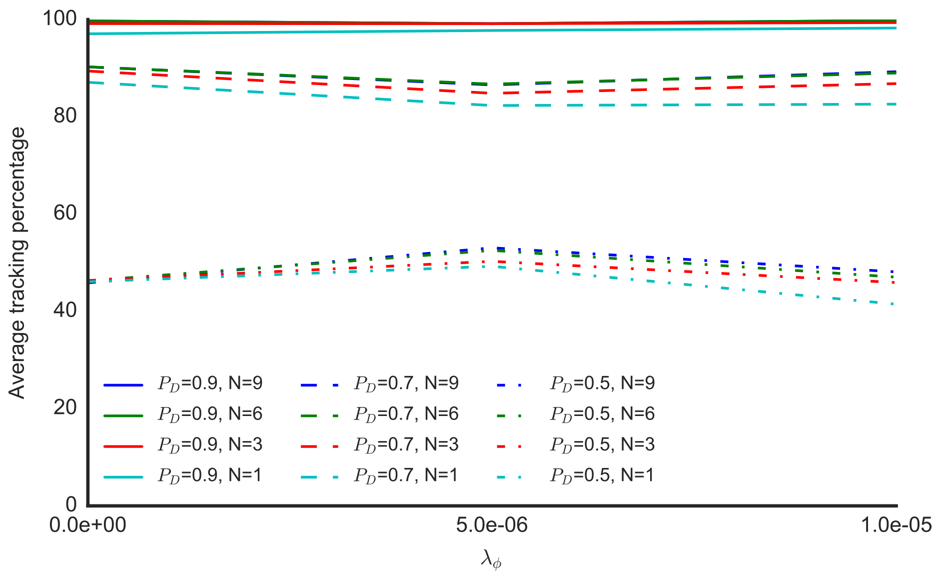

Figure C.2: Scenario 1 — Tracking percentage

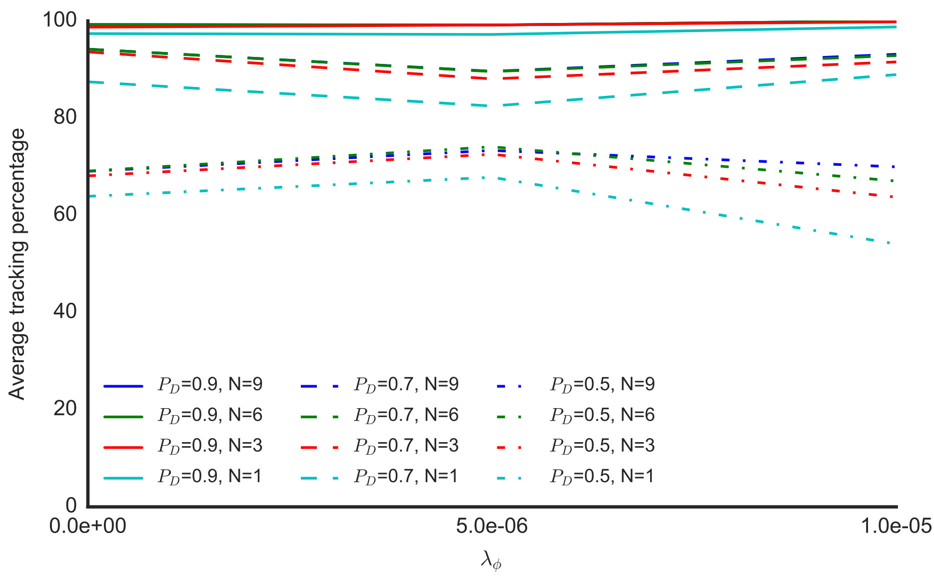

Figure C.3: Scenario 2 — Tracking percentage

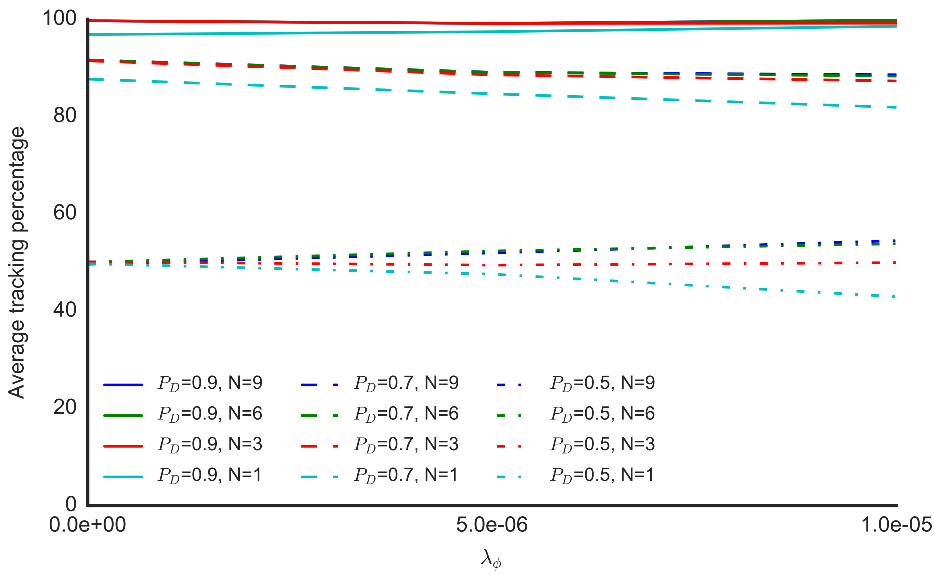

Figure C.4: Scenario 3 — Tracking percentage

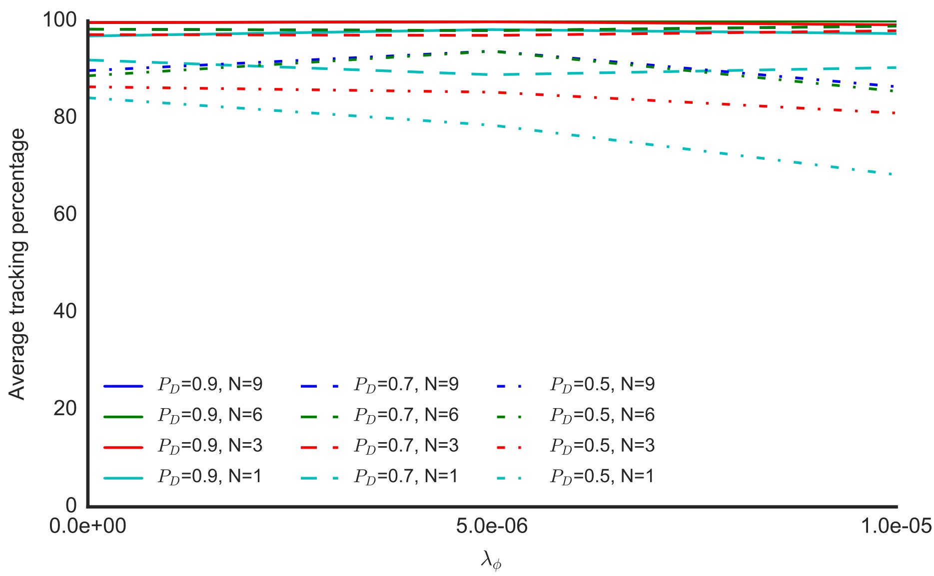

Figure C.5: Scenario 4 — Tracking percentage

Appendix D

Tracking runtime plot

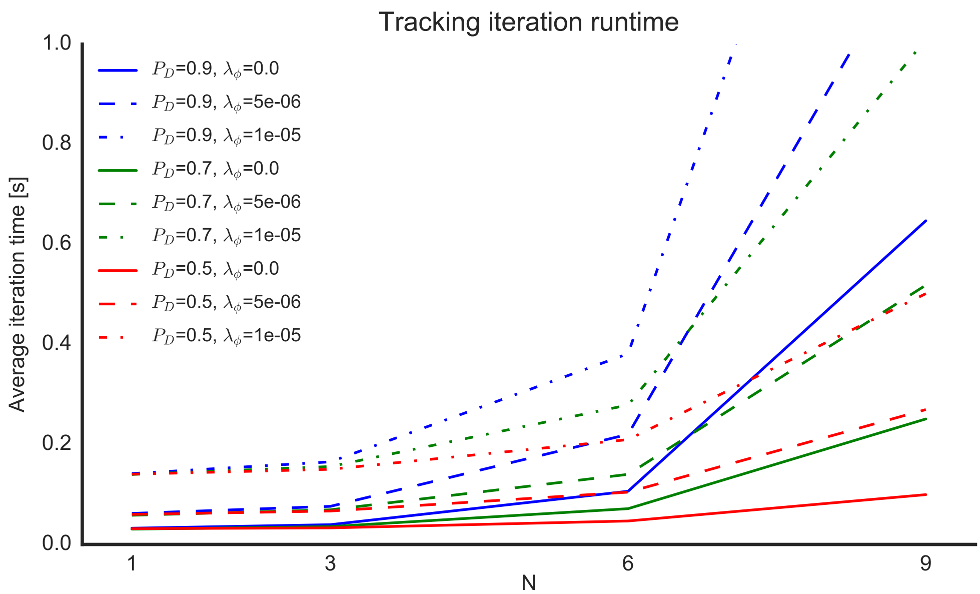

Figure D.1: Scenario 0 — Tracking runtime

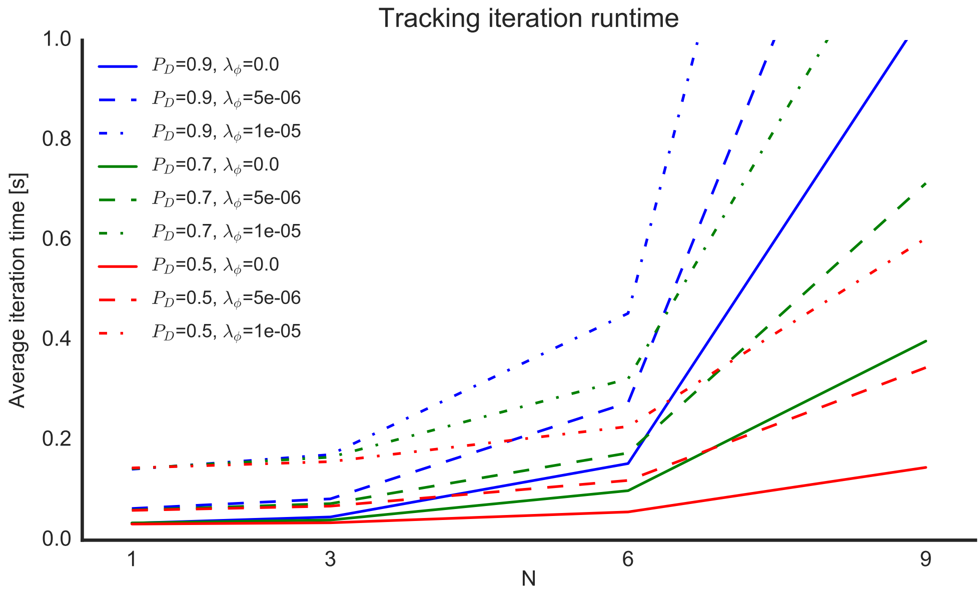

Figure D.2: Scenario 1 — Tracking runtime

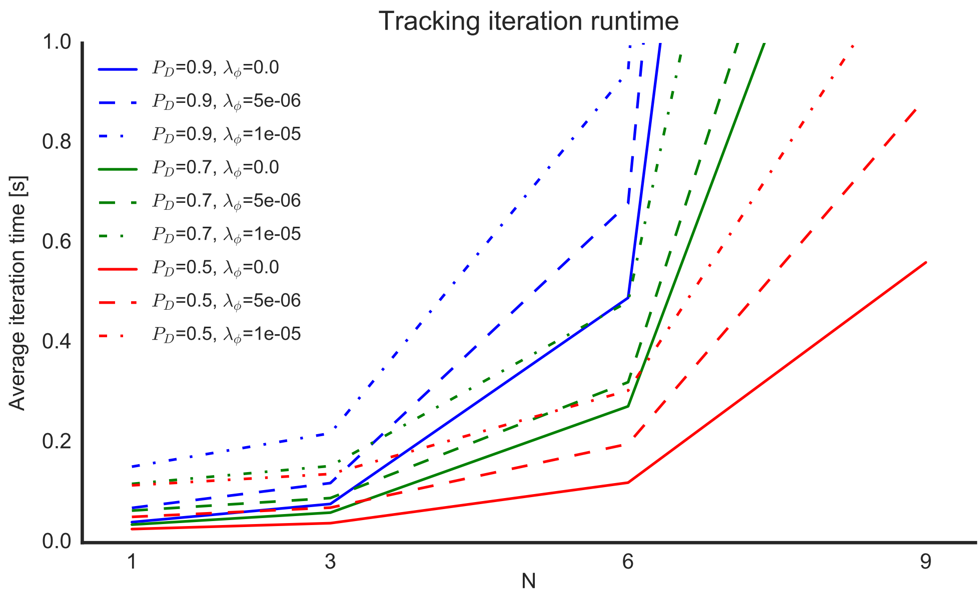

Figure D.3: Scenario 2 — Tracking runtime

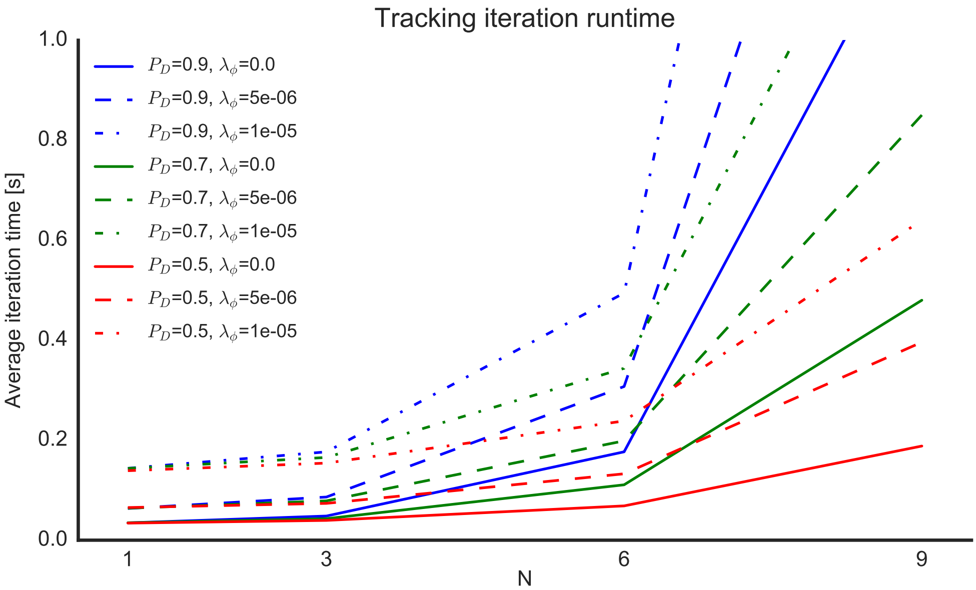

Figure D.4: Scenario 3 — Tracking runtime

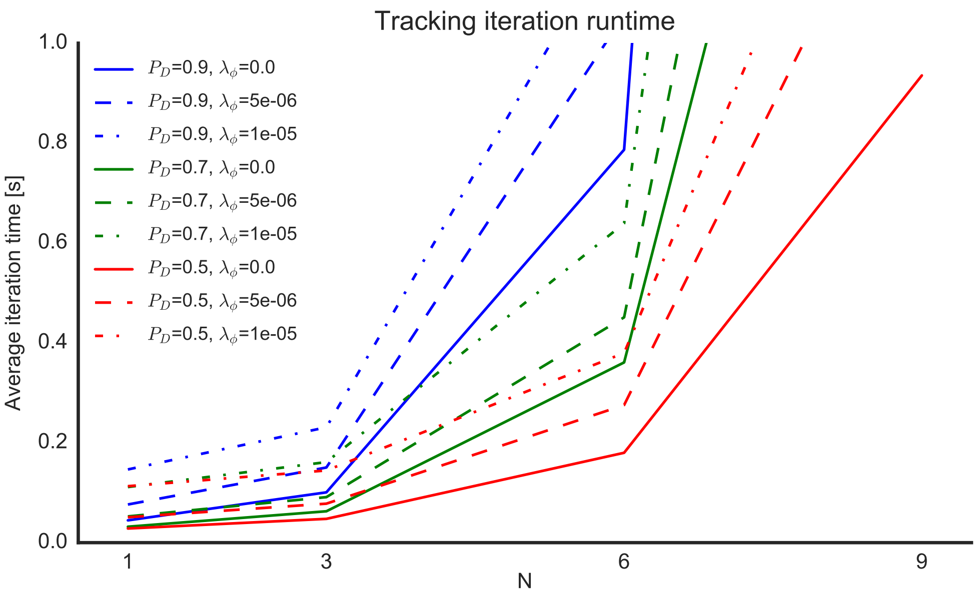

Figure D.5: Scenario 4 — Tracking runtime

Appendix E

Source code

pyMHT, the Python3 implementation of the MHT module developed in this thesis can be found at GitHub: https://github.com/erikliland/pyMHT.

Except for packages available via PyPI the only requirements is the Google Optimization Tools (OR-Tools) which can be obtained from https://developers.google.com/optimization/, and Jacob Frelinger's Cython / C++ implementation of the Munkres algorithm https://github.com/jfrelinger/cython-munkres-wrapper. The later is installed automatically with the pyMHT package.

Bibliography

[1] I. B. Hagen, "Collision Avoidance for ASVs Using Model Predictive Control", no., 2017.

[2] E. Liland, *An ILP approach to Multi Hypothesis Tracking*, Trondheim, 2017. [Online]. Available: osf.io/jp4pp.

[3] IALA, "IALA Guideline No. 1028 On The Automatic Identification", Tech. Rep. 1028, 2004, pp. 1–131.

[4] C. Carthel, S. Coraluppi, R. Grasso, and P. Grignan, "Fusion of AIS, RADAR, and SAR data for maritime surveillance", *Proc. SPIE*, 6748, no., 67480Y–67480Y–8, 2007.

[5] C. Carthel, S. Coraluppi, and P. Grignan, "Multisensor tracking and fusion for maritime surveillance", *FUSION 2007 - 2007 10th International Conference on Information Fusion*, no., 2007.

[6] S. Coraluppi, C. Carthel, M. Luettgen, and S. Lynch, "All-Source Track and Identity Fusion", in *DTIC Technical Reports (U.S. Defence Technical Information Center)*, 2000.

[7] M. T. Wolf, C. Assad, Y. Kuwata, A. Howard, H. Aghazarian, D. Zhu, T. Lu, A. Trebi-Ollennu, and T. Huntsberger, "360-degree visual detection and target tracking on an autonomous surface vehicle", *Journal of Field Robotics*, 27, no., pp. 819–833, 2010.

[8] P. Švec, A. Thakur, E. Raboin, B. C. Shah, and S. K. Gupta, "Target following with motion prediction for unmanned surface vehicle operating in cluttered environments", *Autonomous Robots*, 36, no., pp. 383–405, 2014.

[9] B.-N. Vo, M. Mallick, Y. Bar-Shalom, S. Coraluppi, R. Osborne, R. Mahler, and B.-T. Vo, "Multitarget tracking", *Wiley Encyclopedia of Electrical and Electronics Engineering*, no., 2015.

[10] Y. Bar-Shalom and T. E. Fortmann, “Algorithms for Tracking A Single Target In Clutter”, in *Tracking and Data Association*, New York: Academic, 1998, pp. 119–185.

[11] P. Horridge and S. Maskell, “Real-time tracking of hundreds of targets with efficient exact JPDAF implementation”, in *2006 9th International Conference on Information Fusion, FUSION*, 2006.

[12] R. Singer, R. Sea, and K. Housewright, “Derivation and evaluation of improved tracking filter for use in dense multitarget environments”, *IEEE Transactions on Information Theory*, 20, no., pp. 423–432, 1974. [Online]. Available: http://ieeexplore.ieee . org / xpl / freeabs % 7B % 5C _ %7Dall . jsp ? arnumber = 1055256 % 7B % 5C % %7D5Cnhttp : / / ieeexplore . ieee . org / lpdocs / epic03 / wrapper . htm ? arnumber = 1055256.

[13] D. Reid, “An algorithm for tracking multiple targets”, *IEEE Transactions on Automatic Control*, 24, no., pp. 843–854, 1979.

[14] T. Kurien, “Issues in the design of practical multitarget tracking algorithms”, in *Multitarget-Multisensor Tracking: Application and Advances*, First, Artech House, 1990, ch. 3, pp. 219–245.

[15] Y. Bar-Shalom, S. S. Blackman, and R. J. Fitzgerald, “Dimensionless score function for multiple hypothesis tracking”, *IEEE Transactions on Aerospace and Electronic Systems*, 43, no., pp. 392–400, 2007.

[16] S. Blackman, “Multiple hypothesis tracking for multiple target tracking”, *IEEE Aerospace and Electronic Systems Magazine*, 19, no., pp. 5–18, 2004. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1263228.

[17] S. S. Blackman, R. J. Dempster, and R. W. Reed, “Demonstration of Multiple Hypothesis Tracking (MHT) Practical Real-Time Implementation Feasibility”, *Proceedings of SPIE Vol. 4473*, 4473, no., pp. 470–475, 2001.

[18] M. Mahmuddin and Y. Yusof, “Automatic estimation total number of cluster using a hybrid test-and-generate and K-means algorithm”, *2010 International Conference on Computer Applications and Industrial Electronics*, no., pp. 593–596, 2010.

[19] D. Pelleg and A. Moore, “X-means: Extending k-means with efficient estimation of the number of cluster”, *Seventeeth International Conference on Machine Learning*, no., 2000.

[20] Y. Bar-Shalom and X.-R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques.* YBS Publishing, 1995.

[21] E. F. Wilthil, A. L. Flåten, and E. F. Brekke, "A Target Tracking System for ASV Collision Avoidance Based on the PDAF", no., pp. 1–20,

[22] A. Harati-Mokhtari, A. Wall, P. Brooks, and J. Wang, "Automatic identification system (AIS): Data Reliability and Human Error Implications", *Journal of Navigation*, 60, no., pp. 373–389, 2007.

[23] C. H. Allen, *Farwells Rules Of The Nautical Road*, Eight Edit. Annapolis, MD: Naval Institute Press, 2005.

[24] IMO, "COLREGS - International Regulations for Preventing Collisions at Sea", *Convention on the International Regulations for Preventing Collisions at Sea, 1972*, no., pp. 1–74, 1972.

[25] E. Brekke, O. Hallingstad, and J. Glattetre, "Improved target tracking in the presence of wakes", *IEEE Transactions on Aerospace and Electronic Systems*, 48, no., pp. 1005–1017, 2012.

[26] H. Chen, T. Kirubarajan, and Y. Bar-Shalom, "Performance limits of track-to-track fusion versus centralized estimation: Theory and application", *IEEE Transactions on Aerospace and Electronic Systems*, 39, no., pp. 386–400, 2003.

[27] B. Habtemariam, R. Tharmarasa, M. McDonald, and T. Kirubarajan, "Measurement level AIS/radar fusion", *Signal Processing*, 106, no., pp. 348–357, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.sigpro.2014.07.029.

[28] J. Munkres, "Algorithms for the Assignment and Transportation Problems", *Journal of the Society for Industrial and Applied Mathematics*, 5, no., pp. 32–38, 1957.

[29] S. Chen, L. Liang, and Y. Tian, "The number of connected components in a graph associated with a rectangular (0,1)-matrix", *Linear Algebra and its Applications*, 487, no., pp. 74–85, 2015.

[30] R. G. Brown and P. Y. Hwang, *Introduction to random signals and applied Kalman filtering : with MATLAB exercises.* 2012, vol. 4th.