CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Transportation Sciences

Bc. Petr Lukeš

Design and Development of a Tracker for Requirements of the ATM Laboratory

Master's thesis

2019

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta dopravní
děkan
Konviktská 20, 110 00 Praha 1

ČVUT FD

K621 .... Ústav letecké dopravy

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Bc. Petr Lukeš

Kód studijního programu a studijní obor studenta:

N 3710 – PL – Provoz a řízení letecké dopravy

Název tématu (česky): **Návrh a realizace trackeru pro potřeby ATM Laboratoře**

Název tématu (anglicky): Design and Development of a Tracker for Requirements of the ATM Laboratory

Zásady pro vypracování

Při zpracování diplomové práce se řiďte osnovou uvedenou v následujících bodech:

- Návrh IMM (Interacting Multiple Model) estimátoru pro suboptimální trackování dynamických vzdušných cílů
- Zhodnocení výkonnosti použité metody asociace dat
- Analýza adaptace estimátoru na manévry vzdušných cílů
- Vliv aplikované metody kombinace výsledných odhadů na včasný přechod mezi modely pohybu
- Výkonnostní analýza navrženého trackovacího systému při aplikaci na simulovaných datech a datech z reálného prostředí

ČVUT
FD

Rozsah grafických prací: dle pokynů vedoucího diplomové práce

Rozsah průvodní zprávy: minimálně 55 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)

Seznam odborné literatury: BAR-SHALOM, Yaakov., Xiao-Rong. LI a Thiagalingam. KIRUBARAJAN. Estimation with applications to tracking and navigation.

BLACKMAN, Samuel S. a Robert. POPOLI. Design and analysis of modern tracking systems.

Vedoucí diplomové práce: **Ing. Stanislav Pleninger, Ph.D.**

Datum zadání diplomové práce: **27. července 2018**
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání diplomové práce: **28. května 2019**
a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta dopravní
-3-

L. S.

Kraus
....
doc. Ing. Jakub Kraus, Ph.D.
vedoucí
Ústavu letecké dopravy

....
doc. Ing. Pavel Hrubeš, Ph.D.
děkan fakulty

Potvrzuji převzetí zadání diplomové práce.

....
Bc. Petr Lukeš
jméno a podpis studenta

V Praze dne..27. července 2018

Poděkování

Na tomto místě bych rád poděkoval všem, kteří mě podporovali při tvorbě této práce. Poděkování patří zejména panu Ing. Stanislavu Pleningerovi, Ph. D. za odborné konzultace, ochotu a vedení diplomové práce. Poděkování patří také kolegům z ERA a.s. a to zejména Ing. Petru Litošovi Ph.D. za ochotu při kombinaci studia a profesního života a Ing. Milanovi Říhovi za odborné vedení a předávaní cenných zkušeností v oblasti vícesenzorového trackování. V neposlední řadě patří velké poděkování mým rodičům, kteří mě podporovali v průběhu celého studia, tvorby této práce a umožnili mi se rozvíjet v oblastech mého zájmu.

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci, zpracovanou na závěr studia na ČVUT v Praze Fakultě dopravní.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Nemám závažný důvod proti užívání tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 28. května 2019

. . . .
Podpis

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta dopravní

Návrh a realizace trackeru pro potřeby ATM Laboratoře

diplomová práce
květen 2019
Bc. Petr Lukeš

ABSTRAKT

Cílem této diplomové práce je návrh a realizace trackeru s využitím moderních metod pro trackování cílů. Tracker je navržen pro zpracovávání primárních měření z nízkonákladové pasivní 3D multilaterace široké oblasti. IMM (interacting multiple model) estimátor je vybrán jako vhodný nástroj pro filtraci a predikci a je navržen vzhledem k očekávaným vzdušným cílům. V práci je provedeno vyhodnocení modifikací estimátoru pro značně aperiodické zdroje. Pro vyřešení zdvojených tracků a měření byl vytvořen algoritmus v rámci datové asociace. Návrh trackeru následuje jeho realizace v Matlabu a C++ včetně vyhodnocení jeho výkonnosti na reálných datech ze střední Evropy.

KLÍČOVÁ SLOVA

aperiodický, asociace dat, filtrace, IMM, Kalmanův filtr, manévrující cíle, Markovův řetězec, multilaterace, mód S, trackování

CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Transportation Sciences

Design and Development of a Tracker for Requirements of the ATM laboratory

Master's thesis
May 2019
Bc. Petr Lukeš

ABSTRACT

The aim of the Master's thesis is to design and develop a tracker using modern target tracking techniques. The tracker is designed for processing primary measurements from a low-cost passive 3D Wide Area Multilateration system. The IMM (interacting multiple model) estimator is selected as a suitable tool for filtering and prediction and is designed for the expected airborne targets. Modifications of the estimator for highly aperiodic sources are also evaluated. A data association algorithm used to resolve measurement and track duality is developed. The design of the tracker is followed by its development in Matlab and C++ and its performance is evaluated using data from real traffic in central Europe.

KEY WORDS

aperiodic, data association, filtering, IMM, Kalman filter, manoeuvring targets, Markov chain, multilateration, mode S, tracking

Contents

Abbreviations

API	Application Programming Interface
ARTAS	ATM Surveillance Tracker And Server
A-SMGCS	Advanced-Surface Movement Guidance and Control System
ASTERIX	All Purpose Structured EUROCONTROL Surveillance Information Exchange
ATC	Air Traffic Control
ATM	Air Traffic Management
CNS	Communication Navigation Surveillance
CPU	Central Processing Unit
DRMS	Distance Root Mean Square
DWNA	Discrete White Noise Acceleration
DWPA	Discrete Wiener Process Acceleration
ECAC	European Civil Aviation Conference
ENU	East North Up
GNN	Global Nearest Neighbour
GNSS	Global Navigation Satellite System
GPB	Global Pseudo-Bayesian
GPS	Global Positioning System
GUI	Graphical User Interface
ICAO	International Civil Aviation Organization
IFR	Instrument Flight Rules
IMM	Interacting Multiple Model
JPDA	Joint Probability Data Association
MLT	Multilateration
MM	Multiple Model
NCT	Nearly Coordinated Turn / Nearly Constant Turn Rate
NN	Nearest Neighbour
PDA	Probability Data Association
PSR	Primary Surveillance Radar
RMS	Root Mean Square
SSR	Secondary Surveillance Radar
TCP	Transmission Control Protocol
TDOA	Time Difference Of Arrival
VRMS	Vertical Root Mean Square
VS-IMM	Variable Structure Interacting Multiple Model
WAM	Wide Area Multilateration
WGS84	World Geodetic System 1984

Mathematical notation

$\{\dots\}$	Set
$\triangleq$	By definition
$\square$	End of proof
A'	Matrix transpose
A^{-1}	Matrix inversion
$\lvert A\rvert$	Matrix determinant
$P\{B\}$	Probability of an event
$p(x)$	Probability density function
$\mathcal{N}(\hat{x}, P)$	Gaussian distribution
$\mathcal{N}(x; \hat{x}, P)$	Probability density function of a Gaussian distribution
$E[\cdot]$	Expected value
$\operatorname{var}[\cdot]$	Variance
$\operatorname{cov}[\cdot]$	Covariance
$\operatorname{corr}[\cdot]$	Correlation
$\hat{x}$	Estimated value
$\tilde{x}$	Error

1 Introduction

Air transport plays a crucial role in medium to long distance transport of goods and passengers. Ever increasing demand after its services and high economical impact it has on local and global economy places the air transport into an indispensable position. The current forecasts for air traffic in ECAC predicts yearly growth of 2-3 % IFR movements for the next seven years. This fact coupled with already high density of air traffic in Europe forces the appropriate authorities and the air transport industry to look for innovations in technologies used to manage the use of the airspace.[1][2]

Surveillance, as one of the main pillars of Communication, Navigation and Surveillance (CNS) systems, occupies the function of determining the position of aircraft in the air and on the ground. It is a key component to fulfil the high requirements of safety for which the air transport is known. The provision of aircraft position is not the only and final output of the surveillance function but more a foundation on which another tools are built. An example of this could be seen in the Advanced-Surface Movement Guidance and Control System (A-SMGCS) where Surveillance Service is the core and only required service in the system. Airport Safety Support Service, Routing Service and Guidance Service are services built on top of the Surveillance Service to provide conflict alerting, trajectory generation and automatic control of infrastructure.[3]

As it often is the better the foundations the greater the result. Given the inaccuracies of raw measurements from surveillance sensors there has always been demand to decrease the physical and technological limitations of sensors using mathematical tools. These tools can be separated into two groups. The first contains filters, which are algorithms to filter out the (hopefully) random noise from the raw sensor measurements and to find out the best state estimate. The second group contains data association and data fusion algorithms to associate the measurement with an appropriate target in space. Both of these groups are in practice implemented into tracker. Main function of this system is to find the best state estimate of a targets in space using remote measurements from single or multiple sensors.[4]

Trackers differ greatly based on their practical application. There are simple trackers used in embedded systems as in ACAS which uses adaptive $\alpha - \beta - \gamma$ filter. Example of a complex tracker can be ATM Surveillance Tracker and Server (ARTAS) that is used for aircraft tracking in European multinational surveillance networks and uses some of the most advanced tracking techniques currently in use.[5][6]

The purpose of this thesis is to design and develop an appropriate tracker for the requirements of ATM Laboratory at the Department of Air Transport, Faculty of Transportation Sciences, Czech Technical University in Prague. The ATM Laboratory is equipped with four low-cost 1090 MHz receivers. These receivers are used for wide variety of scientific purposes. One of these is low-cost passive Wide Area Multilateration (WAM) system whose primary measurements (plots) will be used as an input to the designed tracker. As was described earlier the use of a tracker is not only for state estimation but also as a foundation for other applications. The tracker at the ATM Laboratory will in future serve not only for WAM system but also for other research and scientific experiments which include working with position (and possibly other) measurements.

This thesis is divided in a total of ten sections. Excluding this introductory section and the conclusions at the end there is eight sections describing and dealing with the designed tracker.

To fully understand what the designed tracker must provide and what it should be designed for the ATM Laboratory is briefly introduced. In that section the focus is on a hardware and software part of the low-cost passive WAM system which will be providing measurements to the tracker. The system is described in necessary depth for specifying the requirements of the laboratory. Performance analysis of the system is also done as it is required for a correct design and implementation.

Next a brief introduction to target tracking is presented. The introduction focuses on main topics in target tracking such as measurement to track association, track maintenance, filtering and prediction and supporting functions. This chapter is written to introduce what are the main components of the tracker and what these components are trying to solve.

After introducing the basic concepts in target tracking the next chapter focuses on filtering and prediction. First, the cornerstone of tracking and state estimation of dynamic systems, the Kalman filter is introduced. A focus is also on the initialization of the Kalman filter to minimize the number of necessary prior constants required in the tracking process. After the Kalman filter its extended form for non-linear systems is introduced. Next introduction of some insufficiencies brought by the Kalman filter is presented. To combat these insufficiencies the multiple model approach to the target tracking is thoroughly described and derived. Finally the comparison of the Kalman filter and the selected multiple model estimator is presented to reason why the latter is better.

After thorough investigation into filtering and prediction and what method should be used for the designed tracker the interacting multiple model estimator is designed. Modifications of the estimator necessary for the source characteristics of the WAM system are tested.

The next focus is on track maintenance and measurement to track association function of the designed tracker. In both cases the information provided by the sensor is considered and best method fulfilling the necessary requirements is presented.

After designing the tracker and its components its software implementation is presented. The description is oriented as a high level overview since the full source code is provided with the thesis.

The final section before conclusions presents a performance evaluation of the designed tracker on real data from the WAM system. The focus is on testing the measurement to track association function and also the filtering provided by the designed interacting multiple model estimator.

2 ATM Laboratory

2.1 Description

The ATM Laboratory is located at the Department of Air Transport, Faculty of Transportation Sciences, Czech Technical University in Prague. Its purpose is to provide facilities and equipment to students starting their bachelors studies up to doctorate students and researchers employed by the department.

The focus of the laboratory is on wide variety of topics which belongs to Communication, Navigation, Surveillance and Air Traffic Management sphere. Currently and in the recent years the most notable research done includes the topics of Global Navigation Satellite Systems (GNSS) jamming and spoofing, 1090 MHz radio frequency band pollution or the low-cost passive multilateration systems.

Most of the current research done is based on the analysis of passively received data from aircraft transponders on the 1090 MHz radio frequency band. The laboratory uses low-cost receivers for its purposes which allows for investigation of application of low-cost solutions in the CNS/ATM sphere. The number of receivers was gradually growing and about two years ago hit the minimal required count of four receivers for a 3D passive multilateration system. This allowed the laboratory to expand its research into applications of position estimation using low-cost solutions.

Although the receivers can process all messages received at the 1090 MHz radio frequency band the laboratory right now uses only Mode S messages as those bring more scientific potential due to data-link capabilities.

A hardware segment of the passive multilateration system at the ATM Laboratory was for the most part realized by Ing. Martin Zach during his studies at the department. Thorough description of the hardware can be found in his Master's thesis [7]. Brief description of the current situation with the relevant information for the design of the tracker is presented below.

There were two attempts to create a software segment of the passive multilateration system at the laboratory. First attempt done by Ing. Lukáš Umlauf was for 2D passive multilateratiton system due to the fact that the laboratory had only three receivers at that time. The second was done by Ing. Eliška Turková. Both solutions, although appropriate for their use in Master's theses, were not sufficient for real-time application. The problem with both solutions is that it was created in Matlab programming language. Although Matlab is one of the best tools for researchers it has an inherent performance hit for real-time applications due to its interpreted language nature. The real-time passive 3D multilateration software with abilities necessary for subsequent tracker application was developed by me. This software solution is not a subject to evaluation in this thesis but is briefly presented below for its importance as a primary input to the developed tracker.[8][9]

2.2 Multilateration system

2.2.1 Hardware

The hardware segment for 3D low-cost multilateration is made by four identical units distributed around Prague. Location of each unit can be found in table 1.

Table 1: 1090 MHz receivers locations

name	latitude (deg)	longitude (deg)	elliptical heigh (m)
Strahov	50.0805	14.3957	334.80
Pankrac	50.0504	14.4362	379.577
LKPR	50.1063	14.2734	387.978
LKLT	50.1292	14.5258	284.91

Each unit is composed of multiple devices of which the most relevant for this thesis are:

- 1090 MHz aerial
- GPS aerial
- RADARCAPE receiver

An approximate area of coverage of each aerial for different heights at each location can be seen in pictures 1 through 4. White colour corresponds to height between ground level and 10 000 ft. Green colour corresponds to height between 10 000 ft and 20 000 ft. Purple colour corresponds to height from 20 000 ft up to 30 000 ft and the red corresponds to heights above 30 000 ft. For better relative distance estimate Prague was marked by a yellow dot and Bratislava by a blue dot.

It can be seen that the most limiting aerial is Pankrac. This is because of a steel construction at the top of the building where the aerial is located. The steel construction limits the aerial field of view to 180 degrees to the west. This has a significant impact on the results of the 3D multilateration given that successful message reception on all four receivers is required to compute the 3D position. The possibility of an aircraft tracking is then limited to Prague centre and to west of Prague.

The GPS aerials are same across all units. The utmost importance of these aerials and even more of a GPS module in each RADARCAPE receiver is the fact that it provides a distributed time synchronization. A centralized time synchronization for WAM system would never be a low-cost solution. The GPS module in the receivers provides each received message with a time stamp. The technical documentation of the receiver states that the resolution of time stamping is better than 50 ns. Recently research was done at the laboratory to find a method for determining accuracy of time stamping with distributed time synchronization. Sadly this research did not bring a generally applicable solution which could be used for the whole system. For the software segment of the multilateration system and the following figures the value of 50 ns is used.[7][11]

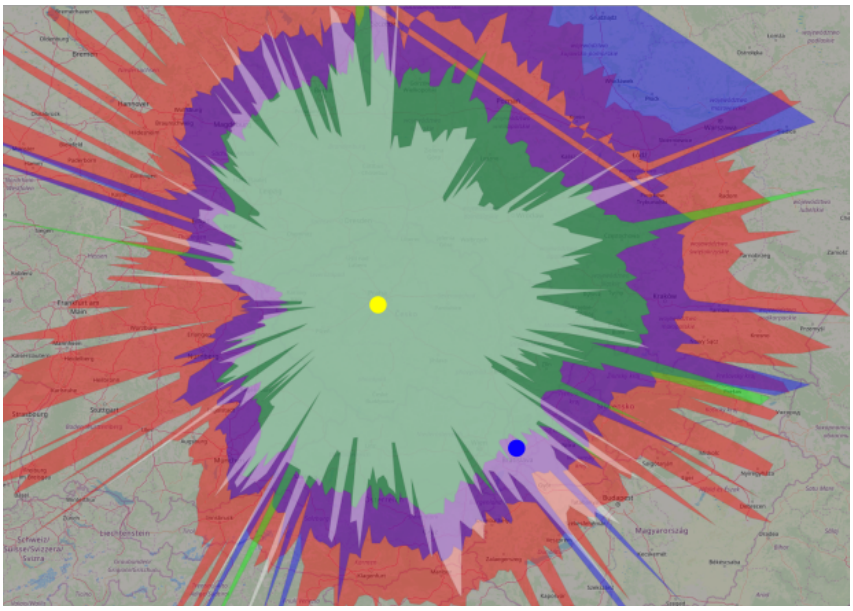

Figure 1: Strahov aerial area of coverage [10]

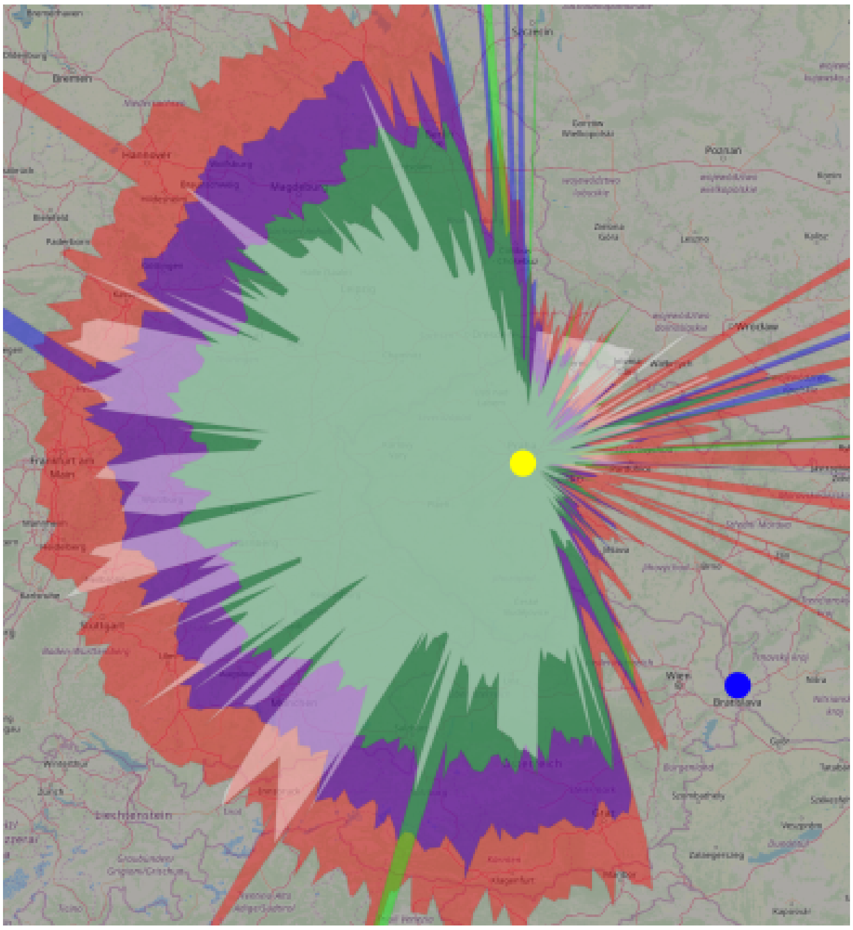

Figure 2: Pankrac aerial area of coverage [10]

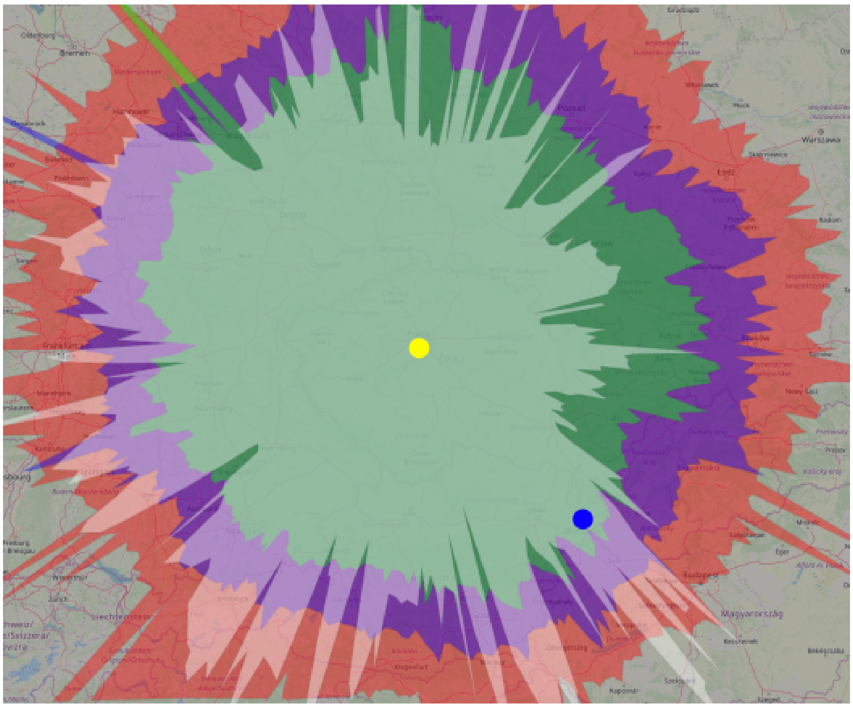

Figure 3: LKPR aerial area of coverage [10]

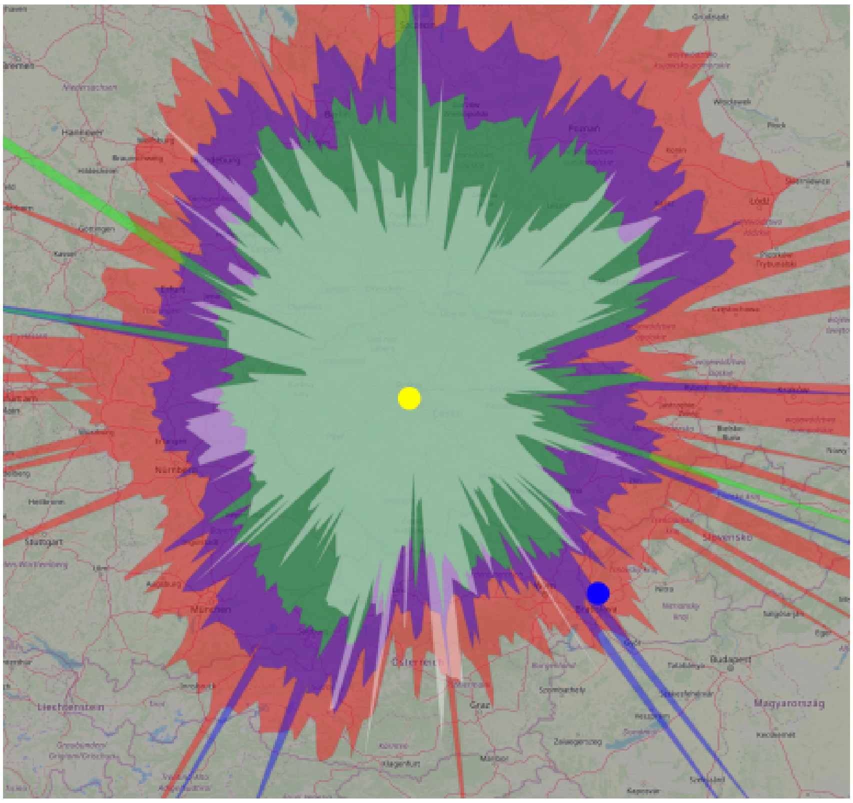

Figure 4: LKLT aerial area of coverage [10]

The impact of position dilution of precision for our geometrical aerial configuration at a height of 5000 m can be seen in a figure 5.

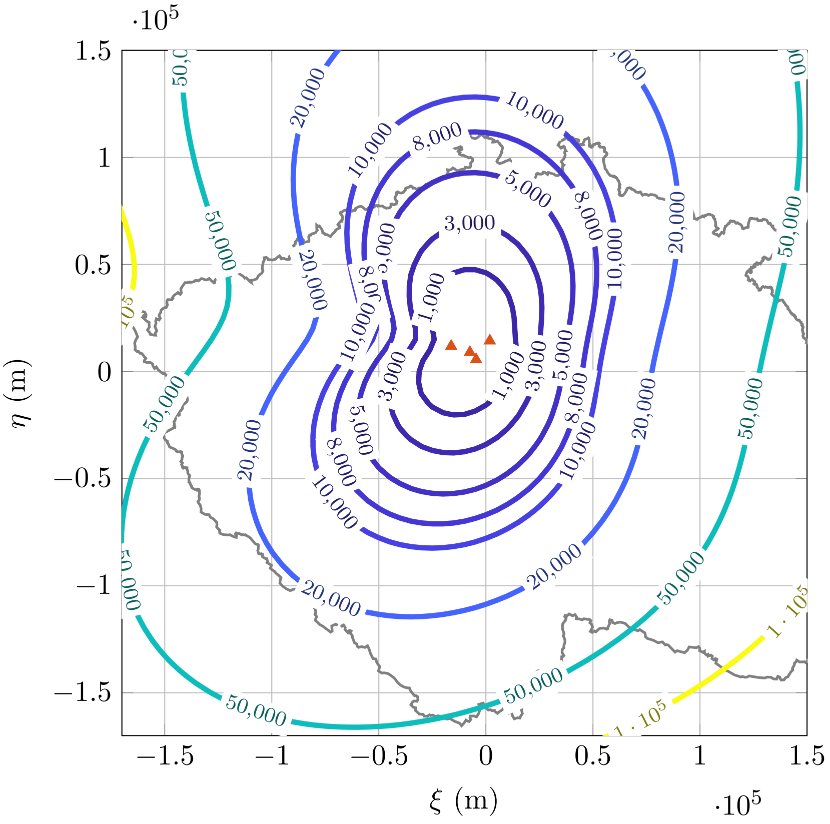

Figure 5: 2DRMS (m) of receivers geometrical configuration at 5000 m

The curves represent isolines of the same twice the distance root mean square (2DRMS) value. 2DRMS is defined as

$$2\text{DRMS} = 2\sqrt{\sigma_\xi^2 + \sigma_\eta^2} \tag{2.1}$$

and represents the confidence interval of 95.4 % to 98.2 %. The orange triangles indicate the receivers positions.[12]

In the figure we can see that the dilution of precision greatly affects the obtained accuracy even near Prague. Usage of plots generated by the multilateration system is thus not suitable for any application with this accuracy. The use of tracker is required. Given the large inaccuracy a modern tracking method should be used to obtain optimal performance for tracking of civil aircraft.

A figure 6 shows isolines for twice vertical root mean square value at a height of 5000 m defined as

$$2\text{VRMS} = 2\sigma_\zeta \tag{2.2}$$

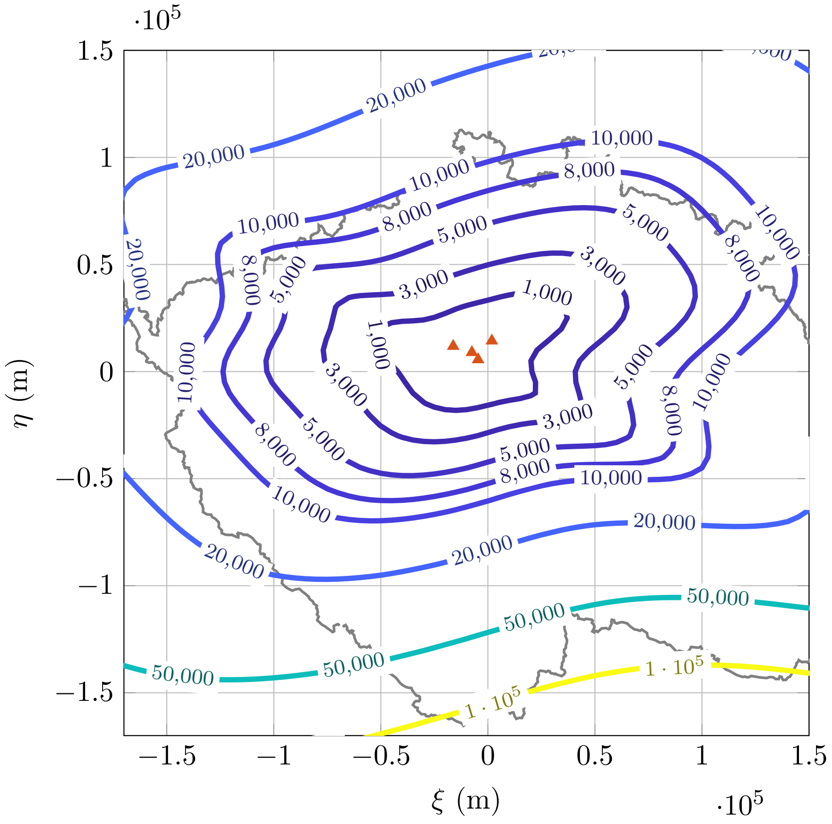

Figure 6: 2VRMS (m) of receivers geometrical configuration at 5000 m

2.2.2 Software

The software segment of passive 3D WAM using Mode S messages was recently created to verify a possibility of using low-cost 1090 MHz receivers as a part of surveillance infrastructure. The architecture of the software was carefully designed to meet high demands of real-time processing. The C++ language was picked as an appropriate tool for this task. Emphasis was put on concurrent programming techniques.

A high level design of the software segment is presented by a sequence diagram in the picture 7.

To manage the high performance demands each element in the sequence diagram is represented by a single thread of execution. This allows for parallel processing and distributes the required CPU time across multiple CPU cores.

A brief description of each element except the designed tracker is presented below.

TCP receiver element represents an abstraction of a real receiver in the software design. This element is responsible for connecting to the RADARCAPE receiver, reception of binary messages using Transmission Control Protocol (TCP) and their partial decoding. For the WAM purposes only decoding required is of a time stamp from the RADARCAPE GPS receiver module. For the tracker purposes an ICAO 24 bit address is also required as only Mode S messages are being processed and the ICAO 24 bit address will be used later in measurement to track association

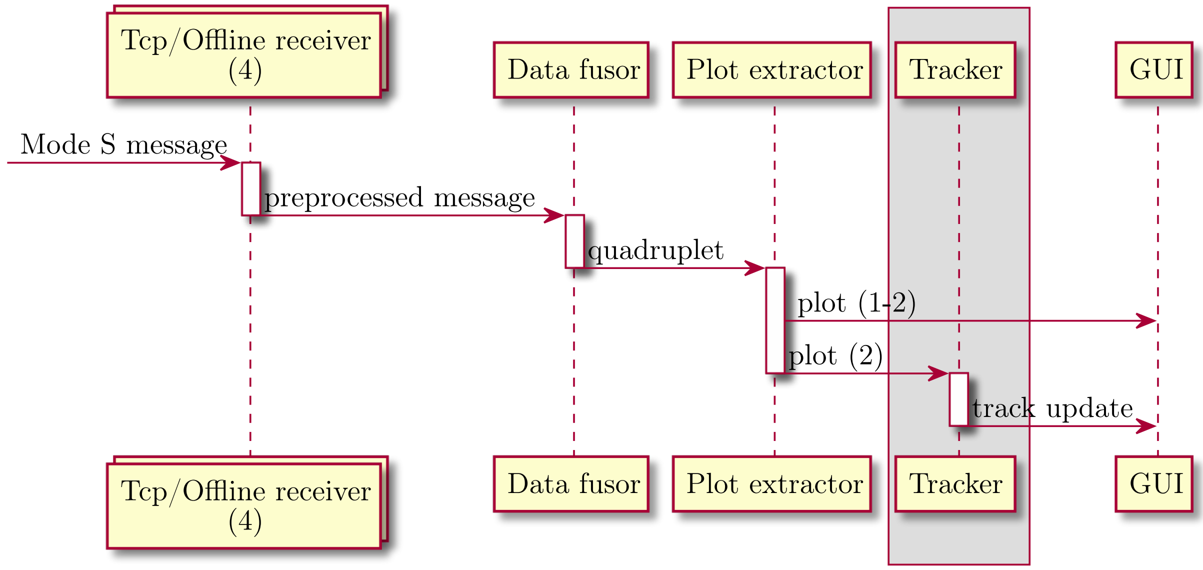

Figure 7: Diagram of WAM software segment

function. The output of the TCP receiver then consists of time of a day the message was received at the receiver, raw 56 or 112 bit data and decoded segments of the message.

Offline receiver is an alternative to the TCP receiver presented above. The only difference is that it does not obtain Mode S messages by connecting directly using the TCP socket but uses as an input datafile from a server at the laboratory. The input data file consists of recorded Mode S messages. The Offline receiver must shift the time stamps of each message according to the time the program started. This allows for real-time replay of the data even from different parts of a day. The output of the Offline receiver is the same as of the TCP receiver.

Data fusor is an element that takes as an input preprocessed messages from all four TCP or Offline receivers. It then proceeds to find quadruplets of messages which contains the same 56 or 112 bit data and their time difference of arrivals (TDOA) are below or equal a maximum value the geometrical configuration allows. The quadruplets that satisfy these criteria are then passed to the next element. Other messages are discarded.

Plot extractor takes as an input quadruplets of preprocessed messages from the Data fusor. It then computes a target position from TDOA combination obtained from each quadruplet. There are several ways to compute the final target position. Each has its advantages and disadvantages and some can even improve accuracy of the final measurement. This is due to the fact that the solution of intersection of three hyperboloids, which is the principle behind 3D multilateration, is a non-linear problem. The method used in this software solution is an algorithm by Bucher and Misra. Its derivation can be found in [13]. The use of this method was inspired by [8] and its main advantage is that it is an exact algorithm. There is no need for iteration and the computation of the target position is straightforward. The problem is that this method leads to a quadratic equation thus two solutions are found. There were already published some basic principles on how to resolve plot duality but none were general enough that it would be always successful. The plot duality resolution will be one of the requirements that the designed tracker should try to solve.[14][15]

Another information the Plot extractor must provide to the tracker is accuracy of the measurement. This accuracy is generally represented by an appropriate covariance matrix. Despite the accuracy dependence on the method used for the target position computation a common approach using least squares method and first order Tylor series approximation is used. This method was selected for its generic applicability and because we only estimate the receiver time stamping accuracy thus a slight difference between used algorithms for covariance matrix computation is negligible for our application.

There are two output branches from the Plot extractor. First goes to the developed tracker. This output consist of measurement vector $[\xi \ \eta]'$, its covariance matrix, elliptical height and its variance for both solutions of the Bucher-Misra's algorithm. Given that the position is already known, the time of measurement can be recalculated to the time of transmission from the aircraft using

$$t_t = t_r - \frac{\Delta s}{c} \tag{2.3}$$

where t_r is time of reception at the chosen receiver, Δs is distance between chosen receiver and the calculated aircraft position and c is speed of light. This time of transmission is calculated for both solutions of the algorithm and passed to the tracker. The output of the tracker is then passed to the Graphical User Interface (GUI).

The second branch goes straight to the GUI. First the measurements with negative elliptical height are filtered out. This filtration provides the simplest form of plot duality resolution and allows for analysis how successful this approach is. After the filtration the unique or dual plots are handed over to the GUI.

It is to be noted that exact solution to the Bucher-Misra's algorithm is computed in East North Up (ENU) coordinate system. The most correct way would be to transform the solutions from ENU coordinate system using stereographic projection. This approach although the correct one is unnecessary for current application in this thesis. Simpler solution was selected that the measurement vector $[\xi \ \eta]'$ is equal to $[e \ n]'$ and the ENU coordinate system si transformed to WGS 84 coordinate system to obtain elliptical height. The transformation for obtaining correct height was done to minimize the impact of ENU coordinate system on the plot duality resolution using plots with positive height only.

GUI for the purposes of this thesis has only a function of displaying the output from 3D WAM and from the designed tracker. The air surveillance scene with plots can be seen in the picture 8.

The blue triangles represent positions of the RADARCAPE receivers. The yellow crosses represent unique plots i.e. plots where the second solution from Buchar-Mira's algorithm did not pass the positive height test. The red crosses represent plots where both solutions have non-negative elliptical heights.

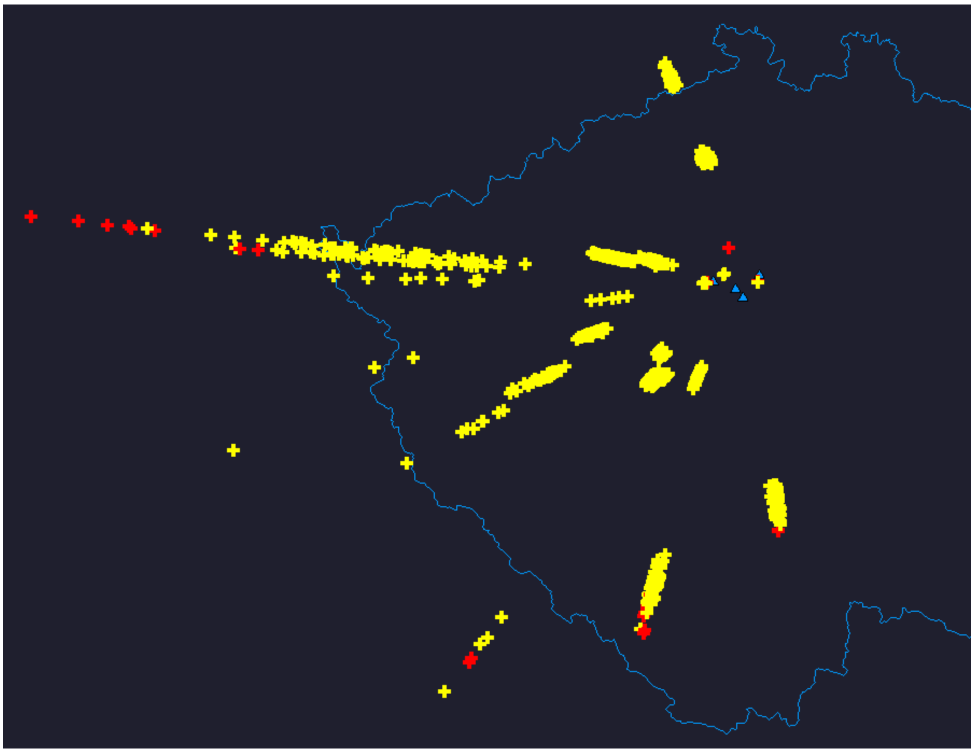

Figure 8: 3D WAM GUI

2.2.3 Performance

To properly determine all the requirements of the designed tracker a performance analysis of the passive 3D WAM at the ATM Laboratory must be done. For the tracker purposes the main points of interest in this regard are:

1. Number of generated plots per second

2. Distribution of inter-arrival times between plots for individual aircraft

The first point is important for the overall performance requirements of the tracker. If the number of plots per second would be excessive a design decision to split the tracker into multiple threads of execution may be required.

The second point is very important for the mathematical side of the tracker design. Given that a lot of the literature on target tracking and position estimation uses a periodic sources due to easier algorithm validation, highly aperiodic surveillance source as a 3D WAM might need some modifications. To correctly evaluate the design decisions a knowledge of an approximate distribution of inter-arrival times between plots is a must.

To analyse the performance a time interval on 4th February 2019 between 12:20 and 13:00 UTC was chosen. Results of the analysis for the first point can be seen in the figure 9.

Maximum number of plots per second in the examined interval reached a value of $190\,\mathrm{s}^{-1}$. Mean value is $84.9729\,\mathrm{s}^{-1}$. Given that modern professional tracking systems are typically capable of maintaining up to 2000 targets at the same time a performance requirement of processing around 200 plots per seconds should be attainable on moderate hardware.

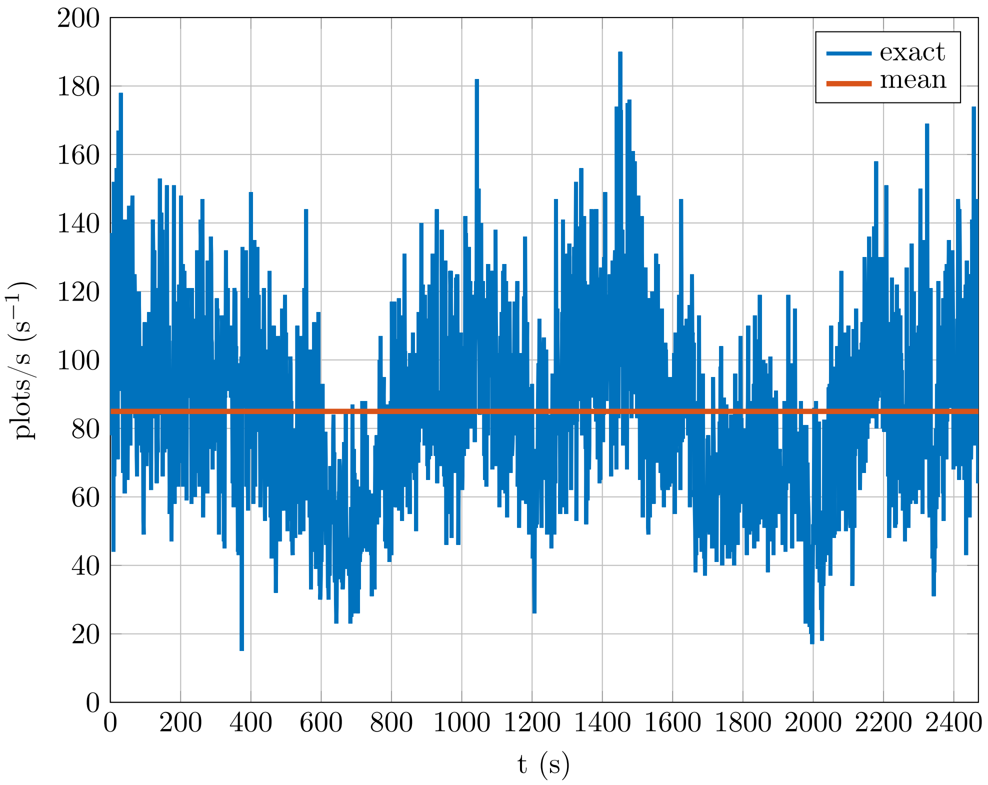

Figure 9: 3D WAM generated plots per second

For the second point a single aircraft with most generated plots inside our examined interval was selected. The number of plots generated for this aircraft was 6400 in a time span of 13.87 minutes. This corresponds to 7.69 plots per second. A histogram of plot inter-arrival times normalized to probability density function is presented in the figure 10. In the figure is also displayed a probability density function of exponential distribution with mean value corresponding to mean inter-arrival time obtained from the data.

It can be seen that an approximation of inter-arrival times to exponential distribution is suitable.

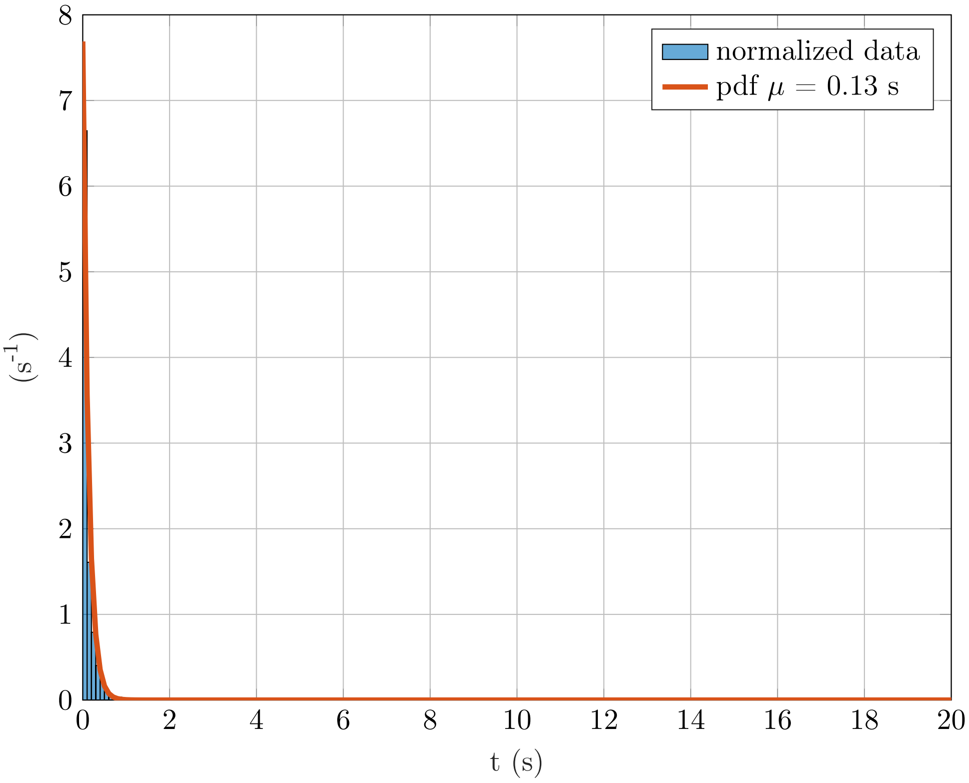

Figure 10: 3D WAM plots interarrival times distribution for single aircraft

2.3 Tracker requirements

To summarize, the requirements of the ATM Laboratory on the designed tracker are:

- Tracker shall use modern tracking algorithms.
- Tracker shall be suitable for civil ATC.
- Tracker shall be suitable for highly aperiodic measurements.
- Tracker may use ICAO 24 bit address.
- Tracker shall be capable of adequate plot duality resolution.
- Tracker shall be capable of processing input from 3D WAM at the ATM Laboratory in real time.

3 Target tracking overview

3.1 System description

The main function and a purpose of a tracker is target tracking which represents processing measurements from single or multiple sensors in different representations and compounding these measurements to tracks to obtain the best state estimate of a target.

Generally speaking a target can be anything for which a measurement is produced i.e. a pedestrian in an airport terminal, car on a street or in case of this thesis aircraft. State of a target then represents its position, velocity or any other different combination of kinematic quantities.

Sensors whose measurements are an input to a tracker can be of numerous different types. Typical examples can be Primary Surveillance Radar (PSR), Secondary Surveillance Radar (SSR), Multilateration (MLT), video camera used for pedestrian tracking using image recognition and much more. Measurements from these sensors can have variety of different forms from the simplest being a position in Cartesian coordinate system to more complex ones as range rate obtained from Doppler radar.

Target tracking can be divided into two groups according to a number of sources:

- Single-sensor target tracking
- Multiple-sensor target tracking

In multiple-sensor target tracking a numerous problems with complicated solutions arise from the necessity of fusing data from different sensors. The most notable of these problems are processing delayed data and an approach to solve multiple-source data fusion where each source have different measurement characteristics. For the purpose of this thesis only the single-sensor target tracking will be considered because only measurements from single sensor represented by 3D WAM will be processed.[16]

Another division of target tracking can be made according to number of tracked targets:

- Single-target tracking
- Multiple-target tracking

Multiple-target tracking is a technique which processes data from a sensor that can observe multiple targets in real time. A prime example for this is 3D WAM at the laboratory. Single-target tracking is based on closed-loop principle where a surveillance sensor which observes a target gets an input from a tracker to its driver to modify its position to keep pointing on the tracked target. In this regard single aircraft in an airspace covered by WAM does not represent a single-target tracking but a multiple-target tracking where only one aircraft is presented. This thesis will focus solely on multiple-target tracking.[17]

Target tracking in general has two approaches to how the data can be processes inside a tracker:[17]

- Batch processing

- Recursive (sequential) processing

Batch processing is a method which uses all data from all scans throughout a history at each new scan to create tracks and find target's best state estimate. This method, although an optimal solution, quickly exceeds available computational resources for real time processing and is not suitable for real time multiple-target tracking.[17]

Recursive processing is a method which sequentially creates tracks and update already created tracks with new measurements as they come in real time. This method may not be optimal, because for example sometimes a track can be updated by a measurement which belongs to different target, but is the only applicable solution due to the computational complexity of batch processing.[17]

Difference between batch processing and recursive processing is similar to difference between least squares estimator and maximum a posteriori estimator.

A simplified diagram of recursive single-sensor multiple-target tracking can be seen in the figure 11.

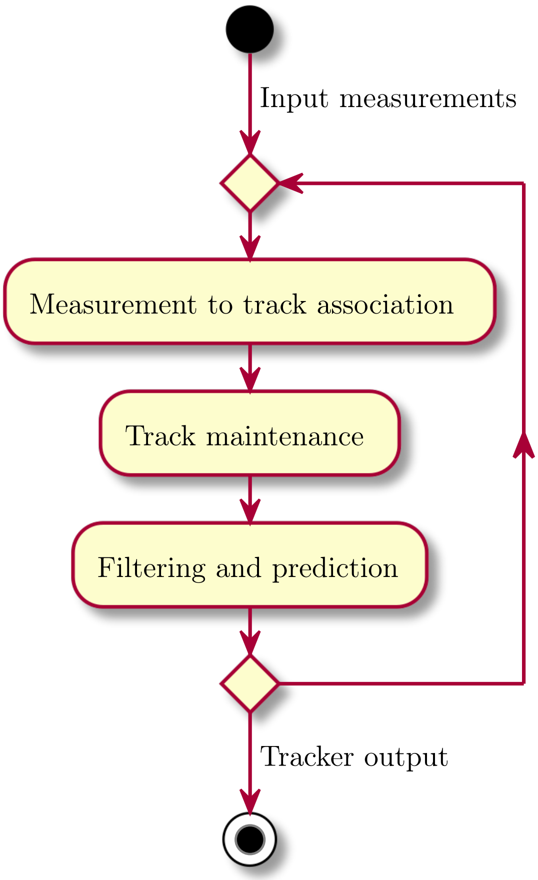

Figure 11: Single-sensor multiple-target tracking

Although functions of the individual blocks in the diagram overlap this representation is an useful simplification for a decomposition of target tracking.[17]

In the next few subsections a brief description of each individual block will be presented.

Couple of next sections of this thesis will then focus deeply on the individual blocks to design an appropriate tracker for the ATM Laboratory.

3.2 Measurement to track association

Measurement to track association is part of a target tracking process focusing on assignment of measurements to tracks. In a picture 12 we can see an example situation which measurement to track association function of the target tracking tries to solve.

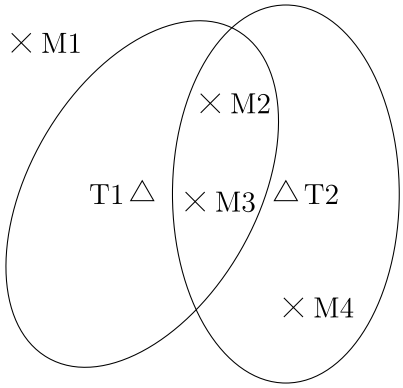

Figure 12: Measurement to track association example

The triangles denote target tracks and the crosses represent measurements from a source. A problem then arises how to assign received measurements to tracks for subsequent filtering, which measurements represents new tracks or which are clutter.

Measurement to track association is typically decomposed into two subsequent parts:

1. Gating
2. Data association

3.2.1 Gating

Purpose of gating in the measurement to track association is to filter out clearly unsatisfying measurements whose inclusion in data association would only increase computational complexity or would even decrease the validity of the final result. In theory measurement to track association would work without gating but for tracking targets in a nations sized area it is a necessity.

Most straightforward approach is the use of Mahalanobis distance. This statistical distance, which can be intuitively interpreted as an multi-dimensional Z-score, represents for two dimensional position measurement an ellipse. These ellipses can be seen in the picture 12.

The size of the ellipses corresponds to a probability level that was selected as an validation threshold for the gating. Usually a probability level around 99 % is selected. This means that 99 % of measurements of a track fall inside its gate.

In the picture 12 it can be seen that measurement M1 falls outside the gate of both tracks. The picture shows only one measurement outside gates for both tracks but given that the length of one side of the picture can be couple hundred or even dozen meters most of the measurements from a source fall outside the gate.

The use of Mahalanobis distance is only one of the approaches to the gating. There are simpler or even more complex solutions. A current trend is to use multi-stage gating where complexity of validation with each stage increases. This causes that Mahalanobis distance where a matrix inversion computation is required is computed for only minimum number of measurement-track pairs.[18][19][20]

3.2.2 Data association

Data association focuses on assigning the measurements that were successfully validated during gating to tracks. There are numerous techniques to solve this problem and a research is still being done in this area. The most common methods currently used can be separated into two groups:

- Single measurement assignment
- Multiple measurement assignment

Single measurement assignment is group of algorithms which assign a maximum of single measurement to a single track. This group contains optimization problems as Nearest Neighbour (NN) or Global Nearest Neighbour (GNN). These problems consist of finding the nearest measurement to a track according to a certain criterion. There are numerous algorithms to solve this problem. On one hand one can have greedy algorithms which sequentially distribute nearest measurements. On the other hand there are algorithms to solve an assignment problem where a global criterion is minimized.[16][17]

Multiple measurement assignment is group of algorithms that take all the measurement that fall inside validation gate and compute a weighted average. This average is then used for track update. This method is mostly used when dealing with sensors with high clutter density. Typical examples of these algorithms are Probability Data Association (PDA) and Joint Probability Data Association (JPDA).[17][21]

Neither single measurement assignment nor multiple measurement assignment can be seen as the ideal way to do the data association. Each method has its own advantages and disadvantages. An exact form of data association inside a tracker is highly dependent on sources used and on computational requirements.

3.3 Track maintenance

Track maintenance is a part of the target tracking process which is mainly responsible for three tasks:

- Track creation
- Track confirmation
- Track deletion

Track creation as its name suggests is a decision logic for when a track should be created. This newly created track typically is not immediately presented to a user because in a high clutter environment a number of spurious tracks can be created. For this reason another decision logic called track confirmation must be implemented.

Track confirmation is highly dependent on the source used. The simplest method for track confirmation is M-N logic. This logic confirms track if there are M correlations of measurements to track out of N scans. More advanced method of track confirmation is based on track score function which indicates a belief that the associated measurements comprise a valid track. If a track score function exceeds a certain value a track is confirmed and presented to a user.[17]

Track deletion is similar to track confirmation. If M out of N measurements is not correlated or a track score function sinks below a certain threshold a track is deleted.[17]

Track maintenance and also data association is much simplified when using SSR or MLT with 24 bit ICAO address.

3.4 Filtering and prediction

Filtering and prediction is a part of target tracking where filtering of noise from associated measurements and prediction of a track state is being done.

Purpose of filtering can be seen in the figure 13.

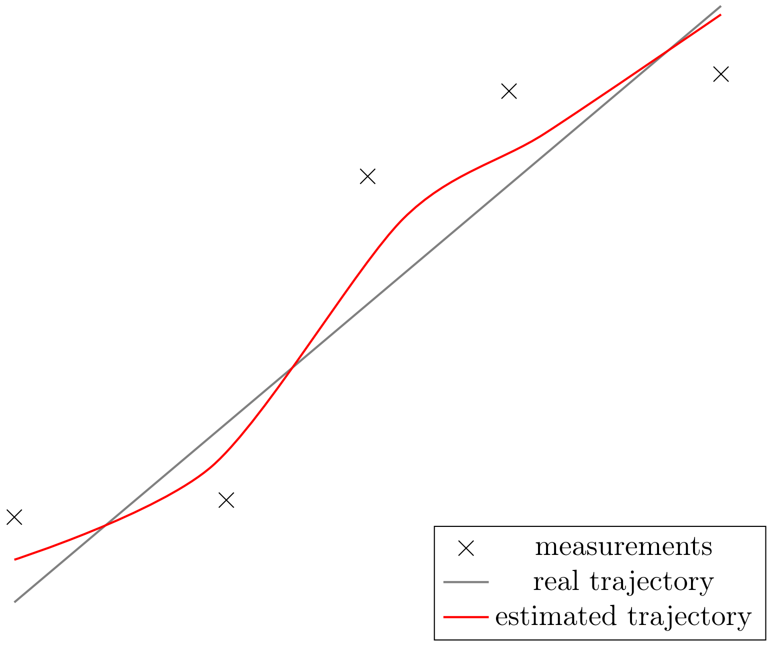

Figure 13: Purpose of filtering

Each received measurement contains certain degree of noise due to sensor accuracy. Filtering function then tries to find the best state estimate of an target by filtering out the sensor noise. There are numerous algorithms that can be used for filtering. Simpler being constant gain $\alpha - \beta$ or $\alpha - \beta - \gamma$ filters which were used in aircraft tracking mostly because of their lower computational requirements then its successors. The currently most widely used algorithm for filtering is Kalman filter.[22]

Development of electronics and the ever increasing computational performance modern processors achieve allowed for a use of more modern methods in target tracking. After successfully achieving enough computational performance for use of Kalman filters a focus shifter on efficient tracking during manoeuvres. Although modern approaches are still mostly based on the Kalman filter its adjustments can cause a significant improvement in final state estimate accuracy.[23]

The prediction is closely connected to the filtering and the measurement to track association. As it will be seen later a prediction is also a part of Kalman filter. Accuracy of the prediction is completely defined by filtered state estimate. For the measurement to track association a prediction is necessary for predicting a target state to a time of measurement. Only after the prediction is done a valid gating and data association metric can be computed.

Thorough description of filtering and prediction necessary for 3D WAM tracking at the laboratory will be presented in section 4.

3.5 Supporting functions

To process input measurements and to provide necessary output a tracker has many other functions that assist the core function of target tracking. Among these functions is input data decoding from specified formats as ASTERIX, transformations between different measurement formats to common coordinate system, coding to output format, data filtration, correction calculation and many others.

These functions although necessary for the function of professional trackers can in some cases be very simplified for the use at the ATM Laboratory. A decision was made that these functions in their appropriate form will be part of the 3D WAM software. The designed tracker and thus this thesis will only focus on the target tracking.

4 Filtering and prediction

4.1 Problem description

The problem of filtering and prediction in target tracking is to estimate a state of a system in an optimal way and to predict the state in time if required. Both filtering and prediction are interconnected as part of an estimation process to obtain a state estimate of dynamic systems.

The difference equations guarding evolution of discrete-time dynamic systems are[24]

$$x(k+1) = f(x(k), u(k), v(k)) \tag{4.1}$$

$$z(k) = h(x(k), w(k)) \tag{4.2}$$

where

- x is n_x dimensional state vector of the system
- u is n_u dimensional known input vector
- v is process noise
- $f(\dots)$ represents the system's dynamics
- z is n_z dimensional measurement (observation) vector
- w is measurement noise
- $h(\dots)$ represent the measurement function
- k represents time step

The equation (4.1) is known as the state dynamics equation and the equation (4.2) as the measurement equation.

The purpose of the estimation is to obtain the best state estimate of $x(k)$ from the measurements provided by the measurement equation. The word best represents a minimization of state estimation errors based on some criterion.[24]

In the view of a Bayesian estimation the state dynamics equation can be represented by a conditional probability density function[25]

$$p(x(k+1)|x(k), u(k)) \tag{4.3}$$

and the measurement equation by

$$p(z(k)|x(k)) \tag{4.4}$$

the final state estimate is then represented by a probability density function of a state conditioned on previous data

$$p(x(k)|Z^k) \tag{4.5}$$

where

$$Z^k \triangleq \{z(0), z(1), \dots, z(k)\} \tag{4.6}$$

which is a sequence of all the measurements available up to and including time k.

The state estimation of dynamic systems is comprised of two steps.

1. Prediction
2. Filtering

The prediction step represents predicting the current state of a dynamic system by its state dynamics to a future time without information obtained by new measurement. That is

$$p(x(k)|Z^k) \rightarrow p(x(k+1)|Z^k) \tag{4.7}$$

This is in a Bayesian sense represented by a marginalization of a joint probability density function[25]

$$\begin{aligned} p(x(k+1)|Z^k) &= \int_{-\infty}^{\infty} p(x(k+1), x(k)|Z^k, u(k))dx(k) \quad (4.8)\\ &= \int_{-\infty}^{\infty} p(x(k+1)|x(k), u(k))p(x(k)|Z^k)dx(k) \quad (4.9) \end{aligned}$$

The filtering step is a transition from the predicted state estimate to the updated best state estimate using the new measurement. That is

$$p(x(k+1)|Z^k) \rightarrow p(x(k+1)|Z^{k+1}) \tag{4.10}$$

This a posteriori conditional probability density function is calculated using Bayes' theorem as follows[25]

$$p(x(k+1)|Z^{k+1}) = \frac{p(z(k+1)|x(k+1))p(x(k+1)|Z^k)}{p(z(k+1)|Z^k)} \tag{4.11}$$

The conditional probability density function on the left hand side of the numerator

$$p(z(k+1)|x(k+1)) \tag{4.12}$$

is measurement equation (4.4). The probability density

$$p(x(k+1)|Z^k) \tag{4.13}$$

is the a priori conditional density of the state prediction (4.8). The denominator is a normalization constant independent of the state.

The equations (4.9) and (4.11) represent a general recursive estimation algorithm. These equations although conceptually applicable and optimal cannot be applied for real world application in the presented form. This is because equations (4.9) and (4.11) calculate with raw probability density functions with whose an recursive computation can get quite complex.[25]

To confront this an approximation of all the used conditional probability density functions in the previous recursion to Gaussian distribution allows for recursive recalculation of solely the first raw and the second central moment of the distribution. An algorithm using this approximation for linear systems is known as the Kalman filter.[24]

4.2 Kalman filter

4.2.1 Introduction

The Kalman filter named after its author R.E. Kalman is in its basic form an optimal recursive Bayesian estimator for linear dynamic systems. From its publication in 1960 the Kalman filter gained a widespread use in multitude of applications and is de facto a current standard in estimation of linear dynamic systems. Its optimality, possibility of relative ease of implementation and the following popularity catalysed further development in estimation which even allowed for relaxations of some restrictions the Kalman filter has.[4][26][27]

The Kalman filter is extensively examined in literature with multiple approaches to its derivation. In the next few paragraphs the discrete time Kalman filter will be introduced as it is a cornerstone of estimation and target tracking.[4][22][24]

4.2.2 Algorithm overview

The discrete time Kalman filter is an optimal estimator of discrete time linear dynamic systems. The vector difference equations of these systems are[4]

$$x(k+1) = F(k)x(k) + G(k)u(k) + v(k) \tag{4.14}$$

$$z(k) = H(k)x(k) + w(k) \tag{4.15}$$

where

- F is $n_x \times n_x$ state transition matrix
- x is n_x dimensional state vector of the system
- G is $n_u \times n_u$ input transition matrix
- u is n_u dimensional known input vector
- v is process noise
- H is $n_z \times n_z$ measurement matrix
- z is n_z dimensional measurement (observation) vector
- w is measurement noise
- k represents time step

and are analogous to equations (4.1) and (4.2) for a linear system.

The process noise $v(k)$ has a covariance matrix

$$E[v(k)v(k)'] = Q(k) \tag{4.16}$$

and the measurement noise has a covariance matrix

$$E[w(k)w(k)'] = R(k) \tag{4.17}$$

The prediction (4.9) segment of Kalman filter algorithm consists of two steps:

1. State prediction

$$\hat{x}(k+1|k) = F(k)\hat{x}(k|k) + G(k)u(k) \tag{4.18}$$

2. State covariance prediction

$$P(k+1|k) = F(k)P(k|k)F(k)' + Q(k) \tag{4.19}$$

The filtering (4.11) segment of Kalman filter is as follows

3. Measurement prediction

$$\hat{z}(k+1|k) = H(k+1)\hat{x}(k+1|k) \tag{4.20}$$

4. Measurement prediction covariance

$$S(k+1) = H(k+1)P(k+1|k)H(k+1)' + R(k+1) \tag{4.21}$$

5. Measurement prediction error, also known as measurement residual or innovation

$$\tilde{z}(k+1|k) = z(k+1) - \hat{z}(k+1|k) \tag{4.22}$$

6. Filter gain

$$W(k+1) \triangleq P(k+1|k)H(k+1)'S(k+1)^{-1} \tag{4.23}$$

7. Updated state estimate

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1)\tilde{z}(k+1|k) \tag{4.24}$$

8. Updated state covariance

$$P(k+1|k+1) = [I - W(k+1)H(k+1)]P(k+1|k) \tag{4.25}$$

The state covariance update equation (4.25) has an alternative called Joseph form

$$\begin{aligned} P(k+1|k+1) =& [I - W(k+1)H(k+1)]P(k+1|k)[I - W(k+1)H(k+1)]' \\ &+ W(k+1)R(k+1)W(k+1)' \end{aligned} \tag{4.26}$$

An advantage of this form is that it holds for an arbitrary gain and is less prone to round-off errors.[4]

During testing of the developed tracker the round off errors were present for an extended Kalman filter shown later. These errors caused a very small negative numbers to appear on the main diagonal of the updated state covariance matrix. To avoid the errors only this method, although more expensive to calculate, will be used.

For the derivation of the Kalman filter there had to be made number of necessary assumptions. These assumptions are:[4]

- The process noise $v(k)$ is zero mean white and Gaussian with known covariance $Q(k)$
- The measurement noise $w(k)$ is zero mean white and Gaussian with known covariance $R(k)$
- The true initial state $x(0)$ is modelled as a random Gaussian variable with known mean and covariance $\hat{x}(0|0)$ and $P(0|0)$
- The process noise, measurement noise and initial state are assumed mutually uncorrelated. That is

$$E[x(0)v(k)'] = 0 \quad \forall k \tag{4.27}$$

$$E[x(0)w(k)'] = 0 \quad \forall k \tag{4.28}$$

$$E[v(k)w(j)'] = 0 \quad \forall k, j \tag{4.29}$$

- Matrices $F(k), G(k), H(k), Q(k), R(k)$ are known

The Gaussian assumptions allow for the Kalman filter to carry out the Bayesian estimation recursion (4.9),(4.11) only on the first two moments of the Gaussian distribution. The first two moments are also sufficient statistics of the Gaussian distribution. With that no information is lost.[4][25]

The Matrices $F(k), G(k), H(k), Q(k), R(k)$ are assumed know but can be time varying. This assumption also applies to the process noise $v(k)$ and the measurement noise $w(k)$ which can be non-stationary. This is an important fact that plays a crucial role in target tracking using sources with non-constant measurement noise. As was shown in the pictures 5 and 6 the accuracy of multilateration is significantly influenced by dilution of precision caused by geometrical configuration of aerials. Because of that the covariance matrix of the measurement noise constantly changes as the aircraft flies in an airspace.[4]

4.2.3 The optimality

It was sad earlier that the Kalman filter is an optimal estimator. The criterion that the Kalman filter optimises is the minimum mean square error.[4]

A minimum mean square error estimator is defined as[4]

$$\hat{x}(k) = \arg\min_{\hat{x}(k)} E[(\hat{x}(k) - x)^2 | Z^k] \tag{4.30}$$

If the Gaussian assumptions are hold for both of the noises and the initial state then the Kalman filter represents optimal (overall best) minimum mean square error estimator. As it is in engineering the real world sometimes does not want to obey ideal conditions. Then if the noises or the initial state is not Gaussian, the Kalman filter represents linear minimum mean square estimator that is the best estimator in the class of linear estimators.[4]

There are other properties then minimizing the expected value of the squared estimation error that the Kalman filter must fulfil. An example can be stability and unbiasnes. Both of these are fulfilled by Kalman filter. More investigation in the derivation of Kalman filter and its properties can be found in [4][22][24][26][28] and other literature.

4.2.4 Initialization of Kalman filter

When someone gets to a practical implementation of the Kalman filter a question arises how to initialize the filter. As it was sad earlier one of the assumptions for the derivation of the Kalman filter is that the true initial state is modelled as a random Gaussian variable with known mean and covariance.

$$x(0) \sim \mathcal{N}\big(\hat{x}(0|0), P(0|0)\big) \tag{4.31}$$

For this assumption to be hold one should initialize the Kalman filter with the initial state generated by a random number generator according to previous equation. Although possible one would need to know the initial state estimate and its covariance matrix.[4]

When doing simulations one could select the initial state estimate $\hat{x}(0|0)$ and some appropriate covariance matrix. But doing this would cause the true initial state $x(0)$ to different each run even for the same simulation scenario. When doing Monte Carlo runs on some scenario one typically wants to preserve the scenario. To combat this one can use the following identity of Gaussian distribution.[4]

$$\begin{aligned}
p\big(x(0)|\hat{x}(0|0)\big) &= \mathcal{N}\big(x(0); \hat{x}(0|0), P(0|0)\big) && (4.32)\\
&= \frac{1}{\sqrt{|2\pi P(0|0)|}} e^{-\dfrac{1}{2}\big(x(0)-\hat{x}(0|0)\big)' P(0|0)^{-1}\big(x(0)-\hat{x}(0|0)\big)} && (4.33)\\
&= \frac{1}{\sqrt{|2\pi P(0|0)|}} e^{-\dfrac{1}{2}\big(\hat{x}(0|0)-x(0)\big)' P(0|0)^{-1}\big(\hat{x}(0|0)-x(0)\big)} && (4.34)\\
&= \mathcal{N}\big(\hat{x}(0|0); x(0), P(0|0)\big) && (4.35)\\
&= p\big(\hat{x}(0|0)|x(0)\big) && (4.36)
\end{aligned}$$

Proof. To show that $p\big(x(0)|\hat{x}(0|0)\big) = p\big(\hat{x}(0|0)|x(0)\big)$ exponents in equations (4.33) and (4.34) must be equal and the covariance matrices of both distributions must be equal. This can be seen below. For better readability the time notation was left out.

$$\begin{aligned}
(x-\hat{x})'P^{-1}(x-\hat{x}) &= x'P^{-1}x - x'P^{-1}\hat{x} - \hat{x}'P^{-1}x + \hat{x}'P^{-1}\hat{x} && (4.37)\\
&= \hat{x}'P^{-1}\hat{x} - \hat{x}'P^{-1}x - x'P^{-1}\hat{x} + x'P^{-1}x && (4.38)\\
&= (\hat{x}-x)'P^{-1}(\hat{x}-x) && (4.39)
\end{aligned}$$

$$\begin{aligned}
P &= E[(x-\hat{x})(x-\hat{x})'] && (4.40)\\
&= E[xx' - x\hat{x}' - \hat{x}x' + \hat{x}\hat{x}'] && (4.41)\\
&= E[\hat{x}\hat{x}' - \hat{x}x' - x\hat{x}' + xx'] && (4.42)\\
&= E[(\hat{x}-x)(\hat{x}-x)'] && (4.43)\\
&= P && (4.44)
\end{aligned}$$

□

In the view of the identity one can select the true initial value $x(0)$ and then generate the initial estimate using

$$\hat{x}(0|0) \sim \mathcal{N}\big(x(0), P(0|0)\big) \tag{4.45}$$

thus the scenario will be preserved for each run and the Kalman filter assumption about the initial state estimate will not be broken.

For practical applications of the Kalman filter in tracker systems the true initial estimate is not known. There are two general approaches to the filter initialization.[4]

- One-point initialization
- Two-point differencing

Let the Kalman filter take as an input position measurements in one coordinate

$$z(k) = \xi(k) + w(k), \qquad w(k) \sim \mathcal{N}(0, R(k)) \tag{4.46}$$

We want to estimate the state vector

$$x = \begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix} \tag{4.47}$$

where ξ is the position and $\dot{\xi}$ is a velocity.

The one-point initialization would then wait for the first measurement, set the initial state estimate to

$$\hat{x}(1|1) = \begin{bmatrix} z(1) \\ 0 \end{bmatrix} \tag{4.48}$$

and set the initial state estimate covariance matrix to

$$P(1|1) = \begin{bmatrix} R(1) & 0 \\ 0 & ? \end{bmatrix} \tag{4.49}$$

The time of the initial estimate was shifted by one to indicate that it is necessary to wait for the first measurement.

The second value on the diagonal of the initial state covariance matrix must be selected a priori. The value depends on the maximum velocity the observed target can have. With a larger maximum velocity a larger value must be set for the missing matrix element. With a larger covariance matrix it will take longer for the Kalman filter to converge to a sufficiently good estimate. Zero was set for the initial velocity estimate as it is assumed that the target can move in both directions equally and zero is the centre of $\mathbb{R}$ domain.[4]

To accelerate the initial convergence of the velocity estimate the two-point differencing method can be used. This method relies on deferring the Kalman filter initialization until the first two measurements are received. After this point one can construct the initial estimate as

$$\hat{x}(2|2) = \begin{bmatrix} z(2) \\ \frac{z(2)-z(1)}{t} \end{bmatrix} \tag{4.50}$$

where t represents the time between the first two measurements. To find the initial state covariance matrix for the two-point differencing method a definition of the covariance matrix of the state is used[29]

$$P(k|k) = E\left[(x - \hat{x}(k|k)) (x - \hat{x}(k|k))' \right] \tag{4.51}$$

The derivation of the initial state covariance matrix for the time $k = 2$ depends on the process noise and thus on the selected motion model represented by the state transition matrix F. This method will be used in the designed tracker as it minimises the required time for the state estimate to get a sufficiently good value and does not depend on an arbitrary preselected value. The derivation for the selected motion models of the designed tracker will be presented later.

If one would have a state vector consisting of

$$x = \begin{bmatrix} \xi \\ \dot{\xi} \\ \ddot{\xi} \end{bmatrix} \tag{4.52}$$

then there conceptually arises the possibility of using three-point differencing. Higher point differencing than two is recommended only when there is significantly higher number of derivatives than two. This is caused because with each higher derivative it takes a lot longer to obtain a relatively good state estimate. The first estimate then might not represent a benefit. The delay of estimating higher derivatives can be seen when analysing the estimated target acceleration after a manoeuvre.[4]

The initialization at the time $k = 2$ will be in the following text shifted back for better consistency to value $k = 0$. The two-point differencing will then use measurements at time $k = 0$ and $k = -1$.

4.3 Extended Kalman filter

4.3.1 Introductiton

The Kalman filter presented above is an optimal minimum mean square estimator when the linear Gaussian assumptions are hold. If the noises entering the system are not Gaussian or the dynamics or measurement equation is not linear a non-linear estimator should be used.[4]

The Bayesian estimation recursion governed by the equations (4.9) and (4.11) is generally applicable even for the non-linear estimation. Similarly to the Kalman filter the necessity arose to create an efficient recursive algorithm that could be generally applicable for the state estimation in non-linear systems. One of these algorithm is known as extended Kalman filter and as its name suggests it is an extension to the Kalman filter.

For the purposes of the designed tracker the noises will be assumed Gaussian for every motion model selected. What will not hold is the linearity requirement. Because of this the extended Kalman filter is briefly presented below.

4.3.2 Algorithm overview

Lets now focus on a non-linear system with additive noise. Its vector difference equations analogous to dynamic equation (4.14) and measurement equation (4.15) are

$$x(k+1) = f\left(k, x(k)\right) + v(k) \tag{4.53}$$

$$z(k) = h\left((k, x(k)\right) + w(k) \tag{4.54}$$

where the non-linear system's dynamics are denoted by the function $f(\dots)$. The $h(\dots)$ is a non-linear measurement function. Both of the noises are assumed white and zero mean with covariance matrices $Q(k)$ and $R(k)$ respectively.

Linearisation in the extended Kalman filter is obtained using vector Taylor series expansion about the latest known estimate. The Taylor series is only an approximation to allow for the linearisation. This causes the extended Kalman filter to lose some of the properties the Kalman filter has. An example is a stability which is not guaranteed or that an estimation error is only approximately zero mean. Some of the deficiencies the extended Kalman filter has can be mitigated by another extension to the extended Kalman filter into iterated extended Kalman filter. This filter tries to eliminate the inaccuracy caused by the Taylor series expansion about the latest estimate and not about the updated estimate which is not known at the moment of the expansion.[4][17]

The Taylor series expansion can be of an arbitrary order. The algorithm presented below will be for the first order Taylor series expansion as its use is mostly found in literature. For the motion model for which the extended Kalman filter will be used in the designed tracker higher order expansion was found by authors of [4] not to provide any improvements.

The first order extended Kalman filter algorithm is very similar to the Kalman filter and is as follows.[4]

1. State prediction

$$\hat{x}(k+1|k) = f(k, \hat{x}(k|k)) \tag{4.55}$$

2. State covariance prediction

$$P(k+1|k) = f_x(k)P(k|k)f_x(k)' + Q(k) \tag{4.56}$$

where

$$f_x(k) \triangleq [\nabla_x f(k,x)']'|_{x=\hat{x}(k|k)} \tag{4.57}$$

$$\triangleq \begin{bmatrix} \frac{\partial f_1(k,x)}{\partial x_1} & \dots & \frac{\partial f_1(k,x)}{\partial x_{n_x}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{n_x}(k,x)}{\partial x_1} & \dots & \frac{\partial f_{n_x}(k,x)}{\partial x_{n_x}} \end{bmatrix}_{x=\hat{x}(k|k)} \tag{4.58}$$

is the Jacobian of the vector function $f(\dots)$ evaluated at the latest state estimate $\hat{x}(k|k)$

3. Measurement prediction

$$\hat{z}(k+1|k) = h(k+1, \hat{x}(k+1|k)) \tag{4.59}$$

4. Measurement prediction covariance

$$S(k+1) = h_x(k+1)P(k+1|k)h_x(k+1)' + R(k+1) \tag{4.60}$$

where

$$h_x(k+1) \triangleq [\nabla_x h(k+1,x)']'|_{x=\hat{x}(k+1|k)} \tag{4.61}$$

$$\triangleq \begin{bmatrix} \dfrac{\partial h_1(k+1,x)}{\partial x_1} & \cdots & \dfrac{\partial h_1(k+1,x)}{\partial x_{n_x}} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial h_{n_z}(k+1,x)}{\partial x_1} & \cdots & \dfrac{\partial h_{n_z}(k+1,x)}{\partial x_{n_x}} \end{bmatrix}_{x=\hat{x}(k+1|k)} \tag{4.62}$$

is the Jacobian of the vector function $h(\dots)$ evaluated at the predicted state estimate $\hat{x}(k+1|k)$

5. Measurement prediction error (innovation)

$$\tilde{z}(k+1|k) = z(k+1) - \hat{z}(k+1|k) \tag{4.63}$$

6. Filter gain

$$W(k+1) \triangleq P(k+1|k)h_x(k+1)'S(k+1)^{-1} \tag{4.64}$$

7. Updated state estimate

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1)\tilde{z}(k+1|k) \tag{4.65}$$

8. Updated state covariance (Joseph form)

$$\begin{aligned} P(k+1|k+1) =& [I - W(k+1)h_x(k+1)]P(k+1|k)[I - W(k+1)h_x(k+1)]' \\ &+ W(k+1)R(k+1)W(k+1)' \end{aligned} \tag{4.66}$$

When comparing the previous equations of the first order extended Kalman filter to the equations of the Kalman filter (4.18) - (4.26) the only differences are in the state and the measurement prediction which are done directly using appropriate transition functions and in the use of the Jacobians.

4.4 Manoeuvring targets

4.4.1 Kalman filter insufficiencies

The Kalman filter is a state of the art estimator. When designing implementation of the Kalman filter for target tracking one needs to select a motion model with an appropriate process noise covariance matrix that best fits the expected target motion. In doing so an approximation of the motion model is necessary because in most cases the true motion the target will perform is not known. The process noise covariance matrix serves as an parameter to allow for deviations of the target motion from the selected motion model. This is the main principle behind the so called filter tuning.

1. Select the best approximation to the expected target motion

2. Select an appropriate process noise covariance matrix to allow for motion deviations

For simpler applications an approximation of the target motion to a single motion model is possible. Slight deviations are then easily covered by a small process noise covariance matrix and the filtering step sufficiently filters out the measurement noise. Problem arises what to do when a target with a wide range of motions is to be tracker. In this case the process noise covariance matrix allows for compromise between

- Efficient filtering of measurement noise when the target is not manoeuvring
- Maximum estimation error during manoeuvres

A Monte Carlo simulation with 100 runs for a single scenario was performed to show an example of the influence the compromise has. The evaluated scenario consist of an aircraft travelling at the speed of $300\,\mathrm{ms}^{-1}$ for 30 seconds. After that a 180 degrees coordinated turn ending at a time mark 77 s with a normal acceleration of 2 g was performed. After the turn the aircraft continued for another 30 seconds at the same speed. Measurement noise was selected to be $\sigma_{w_\xi} = \sigma_{w_\eta} = 100\,\mathrm{m}$. Time between measurements was selected to be of a constant value 1 second. The motion model of the system's dynamics was selected as nearly constant acceleration[1]. Two Kalman filters were used with the same motion model differing only in the size of a process noise standard deviation of acceleration σ_v which is used to calculate the process noise covariance matrix.

The figure 14 depicts root mean square (RMS) position error of both estimated trajectories and also the position standard deviation of measurement.

The RMS position error is defined for a position vector

$$x = \begin{bmatrix} \xi \\ \eta \end{bmatrix} \tag{4.67}$$

as

$$\mathrm{RMS}(\tilde{x}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\tilde{\xi_i}^2 + \tilde{\eta_i}^2)} \tag{4.68}$$

where N is the number of Monte Carlo runs and

$$\tilde{x} = \begin{bmatrix} \tilde{\xi} \\ \tilde{\eta} \end{bmatrix} = \begin{bmatrix} \xi - \hat{\xi} \\ \eta - \hat{\eta} \end{bmatrix} \tag{4.69}$$

where ξ, η represent true aircraft position and $\hat{\xi}, \hat{\eta}$ estimated position.

The position standard deviation of the measurement is defined as

$$\sigma_w = \sqrt{\sigma_{w_\xi}^2 + \sigma_{w_\eta}^2} \tag{4.70}$$

From the simulation results it can be seen that a smaller process noise covariance matrix results in a better filtering during non-manoeuvring motion but significantly lags behind when the manoeuvre occurs. With the smaller process noise covariance matrix even the position standard deviation of measurement was exceeded. The filter with the larger process noise covariance has better results during manoeuvre but worse results during constant motion.

[1] Its form can be seen in the section 5.2.3.

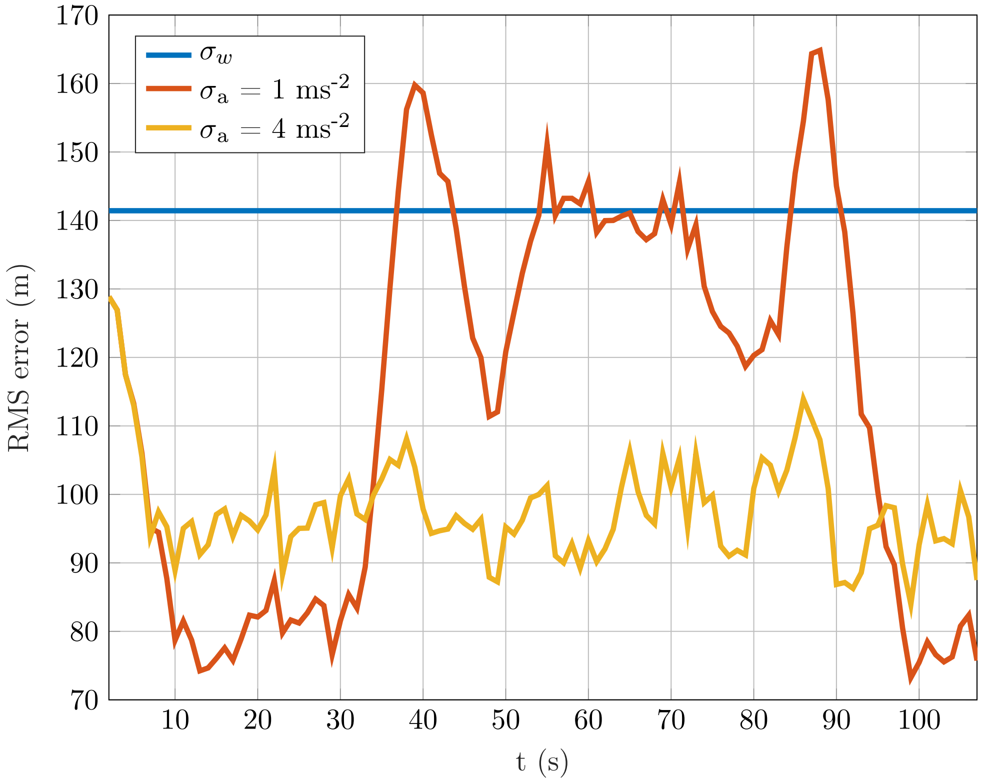

Figure 14: Kalman filter tuning compromise

Given that the aircraft can perform wide variety of manoeuvres the tuning compromise of the Kalman filter between linear non-accelerated motion and manoeuvre would result in a significantly lower accuracy than would be wanted. The accuracy during manoeuvres is critical for the safety of air traffic and for the target tracking in general. To obtain better performance during manoeuvres and to mitigate the performance reduction during non-manoeuvring motion an alternative such as adaptive estimation algorithm should be used.

4.4.2 Adaptive estimation algorithms

The adaptive estimation algorithms are state estimation techniques that can adapt in some way to different kinds of uncertainties. In the previously introduced Kalman filter the only uncertainties were zero mean white measurement and process noise with known covariance matrices. The manoeuvring targets are characterized by equation (4.14)[2] where not only the process noise represents the uncertainty but also the input $u(k)$ representing the manoeuvre command is unknown.[4]

The adaptive algorithms that are used to adapt to the uncertainties can be divided into three groups:[4]

1. The unknown input is modelled as a random process

[2] In the following text only linear models are considered. For the non-linear models the approach is same with the necessity of linearisation.

2. The unknown input is estimated in real time
3. The system is assumed to behave according to one of finite number of models

When the unknown input is modelled as a random process the manoeuvre is assumed to be additional process noise. The random process noise can be assumed white or even coloured. When using coloured noise the process noise cannot be used directly and must be pre-whitened. A famous example of a use of coloured process noise in tracking of manoeuvring targets is the Singer model.[4][29]

If the additional process noise is assumed to be constant the Kalman filter will only have larger process noise covariance matrix. Given that manoeuvres are constantly changing in their duration and magnitude two adaptive techniques were developed.[4]

- **Continuous noise level adjustment**
  This technique starts target tracking with a low process noise. When a manoeuvre occurs it is expected that the innovation, which as was presented earlier by equation (4.22) is a difference between received measurement and a predicted measurement, will significantly increase. For this a statistical test can be done on some probability level. When the threshold is exceeded the process noise is continuously scaled up by a small factor until the statistical test reduces below the selected threshold. The similar approach in the opposite way is done when the manoeuvre ceases.[4]
- **Noise level switching**
  This algorithm is similar to the previous one but the process noise has several preselected discrete levels. Between these levels a switching occurs based on statistical testing and preselected thresholds.[4]

The second group of adaptive algorithms consist of estimating the input in real time. This input estimation is done by assuming that the input (manoeuvre) is constant for a certain period of time. There are two algorithms dealing with this problem.[4]

- **Input estimation and estimate correction**
  This approach is based on estimating input in a system without input using least squares method on a sliding window of number of last measurements. If the input in a form of for example acceleration is estimated then it is tested for a statistical significance. If the selected threshold is exceeded the final estimate is corrected.[4]
- **Variable state dimension**
  In this approach if the manoeuvre is detected then instead of computing the input estimate and correcting the final estimate the state vector $x(k)$ is augmented by another state component. An example can be switching between nearly constant velocity model and nearly constant acceleration model.[4]

$$x = \begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix} \rightarrow x = \begin{bmatrix} \xi \\ \dot{\xi} \\ \ddot{\xi} \end{bmatrix} \tag{4.71}$$

> Switching between lower order model and higher order model is suitable as it allows for good tracking performance in both situations. Using nearly constant acceleration model even for a target who is travelling at constant speed unnecessarily increases estimation error for position and velocity.[4]

The third group of adaptive algorithms, where the system is assumed to behave according to one of finite number of models, known as multiple model (MM) approach will be covered in more detail next. One of those algorithms will be used in the designed tracker.

4.5 Multiple model

4.5.1 Introduction

The multiple model approach describes so called hybrid systems. These systems are identified by having not only continuous but also discrete uncertainties. The continuous uncertainties are represented by the process noise $v(k)$ in the model dynamics and that the state is described by stochastic difference equation. The discrete uncertainties are present by assuming that the targets obeys one of several motion models.[4][30]

4.5.2 Static multiple model estimator

Given a set of r models

$$M \in \{M_j\}_{j=1}^{r} \tag{4.72}$$

If we assume that the system obeys a single fixed model then we talk about static multiple mode estimator. Even though the single model is fixed and the multiple model estimator is static the fixed model still has its own dynamics so in the end it is still a dynamic estimator.[4]

In this and the following sections a term mode and model will be used. The mode refers to real world behaviour of a target. When aircraft is turning it is in turn mode. The model then refers to mathematical representation of the system.[16]

The multiple model approach, same as the Kalman filter, is based on Bayesian estimation. To calculate posterior probability of model M_j being correct we use Bayes rule in the following way.[4]

Given assumed known prior probability of model M_j being correct

$$\mu_j(0) = P\{M_j|Z^0\} \quad j = 1, \ldots, r \tag{4.73}$$

where it is assumed that the correct model is present in the set of r models. That is

$$\sum_{j=1}^{r} \mu_j(0) = 1 \tag{4.74}$$

Although most of the time we will not have the exact motion model present. The multiple model estimator stability causes convergence to unit probability for the nearest model to the true one.[31]

For the posterior probability we then obtain a recursion

$$\mu_j(k) \triangleq P\{M_j|Z^k\} \tag{4.75}$$

$$= P\{M_j|z(k), Z^{k-1}\} \tag{4.76}$$

$$= \frac{p\left(z(k)|Z^{k-1}, M_j\right) P\{M_j|Z^{k-1}\}}{p\left(z(k)|Z^{k-1}\right)} \tag{4.77}$$

$$= \frac{p\left(z(k)|Z^{k-1}, M_j\right) P\{M_j|Z^{k-1}\}}{\sum_{i=1}^{r} p\left(z(k)|Z^{k-1}, M_i\right) P\{M_i|Z^{k_1}\}} \tag{4.78}$$

$$= \frac{p\left(z(k)|Z^{k-1}, M_j\right) \mu_j(k-1)}{\sum_{i=1}^{r} p\left(z(k)|Z^{k-1}, M_i\right) \mu_i(k-1)} \tag{4.79}$$

The left term in the numerator under linear Gaussian assumptions is

$$p\big(z(k)|Z^{k-1}, M_j\big) = \mathcal{N}\left(z(k); \hat{z}(k|k-1), S(k)\right) \tag{4.80}$$

$$= \mathcal{N}\left(z(k) - \hat{z}(k|k-1); 0, S(k)\right) \tag{4.81}$$

$$= \mathcal{N}\left(\tilde{z}(k|k-1); 0, S(k)\right) \tag{4.82}$$

$$= p\left(\tilde{z}(k|k-1)\right) \tag{4.83}$$

$$= \Lambda_j(k) \tag{4.84}$$

which is the likelihood function of mode M_j being in effect at time k. The expression $\tilde{z}(k|k-1)$ is the innovation from (4.22).

A diagram of the static multiple model estimator algorithm can be seen in the figure 15.

Each Kalman filter in the algorithm stands for a single motion model. The filters then provide mode-matched state estimate and mode-matched state estimate covariance matrices as their output. After the initialization each filter takes as an input its previous estimate. The probability of each model being correct at time k, that is the probability of a mode corresponding to the filter at time k being correct, is computed using (4.79) in a step called mode probability update.[4]

The state estimate and covariance combination combines the output from each Kalman filter into weighted state and state covariance estimate for the multiple model output purposes. Under the Gaussian assumptions the probability density function of the combined state is obtained using Gaussian mixture as described below.[4]

$$p\big(x(k)|Z^k\big) = \sum_{j=1}^{r} \mu_j(k) \mathcal{N}\left(x(k); \hat{x}_j(k|k), P_j(k|k)\right) \tag{4.85}$$

Expected value of this probability density function which is the combined state estimate is

$$\hat{x}(k|k) = E[p\big(x(k)|Z^k\big)] \tag{4.86}$$

$$= E\left[\sum_{j=1}^{r} \mu_j(k) \mathcal{N}\left(x(k); \hat{x}_j(k|k), P_j(k|k)\right)\right] \tag{4.87}$$

$$= \sum_{j=1}^{r} \mu_j(k) E\left[\mathcal{N}\left(x(k); \hat{x}_j(k|k), P_j(k|k)\right)\right] \tag{4.88}$$

$$= \sum_{j=1}^{r} \mu_j(k) \hat{x}_j(k|k) \tag{4.89}$$

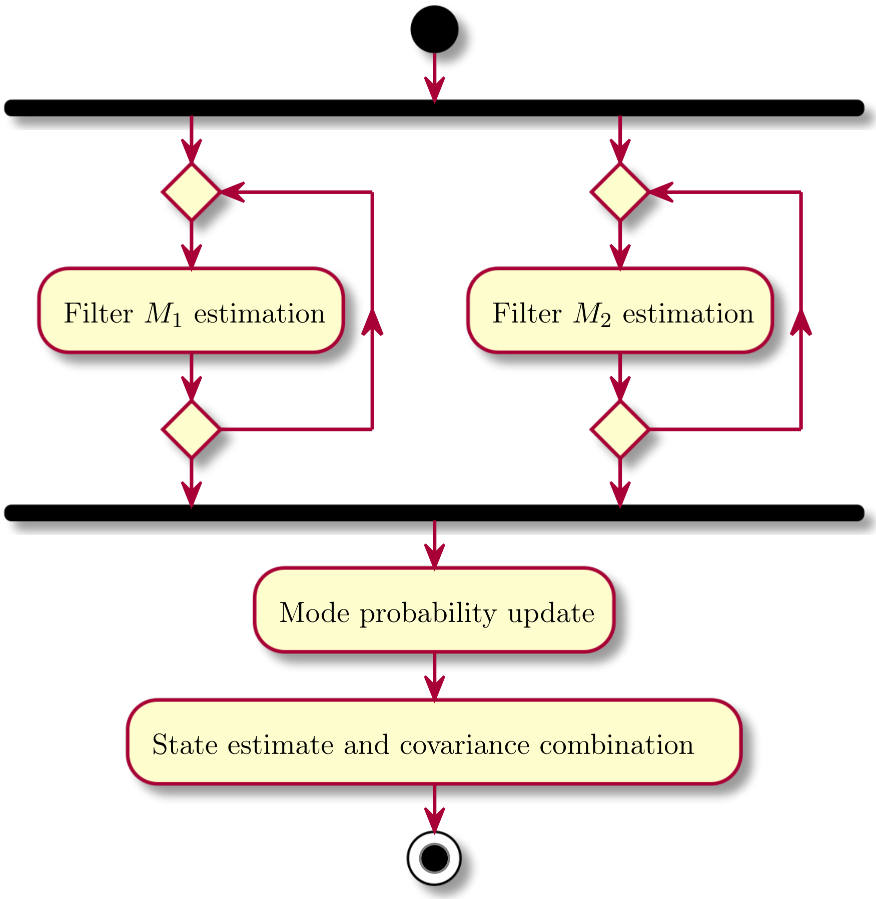

Figure 15: Static multiple model estimator

The covariance of the combined state estimate is then

$$P(k|k) = \sum_{j=1}^{r} \mu_j(k) \left(P_j(k|k) + (\hat{x}_j(k|k) - \hat{x}(k|k)) (\hat{x}_j(k|k) - \hat{x}(k|k))' \right) \tag{4.90}$$

Some modifications can be done to the previously described static multiple model estimator to allow for switching between different models and thus for better manoeuvre adaptation. These modifications are currently unnecessary approximations and the use of dynamic multiple model estimator is a better alternative.[4]

4.5.3 Dynamic multiple model estimator

The dynamic multiple mode estimator describes estimation of hybrid systems which can undergo mode switching in time. That is the most suitable model from the bank of r preselected models can differ each sampling period. These systems can be described by the following model dynamics and measurement equation[30]

$$x(k+1) = F\left(k, M(k+1)\right) x(k) + v\left(k, M(k+1)\right) \tag{4.91}$$

$$z(k) = H\left(k, M(k)\right) x(k) + w\left(k, M(k)\right) \tag{4.92}$$

where $M(k+1)$ is the mode during sampling period ending at time $k+1$.

First let the l^{th} mode history up to and including time k denote as

$$M^{k,l} = \{M_{\iota_{1,l}}, \dots, M_{\iota_{k,l}}\} \tag{4.93}$$

where the index $\iota_{k,l}$ is model index at time k from l^{th} history. Given two possible modes ($r = 2$) and two measurements ($k = 2$) we obtain 4 different possible mode histories. These histories can be seen in the table 2.[4]

Table 2: Mode histories for two measurements at time $k = 2$ [4]

l	$\iota_{1,l}$	$\iota_{2,l}$
1	1	1
2	1	2
3	2	1
4	2	2

It can be easily seen that

$$l = r^k \tag{4.94}$$

that is the number of histories grows exponentially in time.

Next it is assumed that the mode switching occurs according to homogeneous Markov chain. This means that the mode transition probabilities

$$p_{ij} \triangleq P\{M(k) = M_j | M(k-1) = M_i\} \tag{4.95}$$

are assumed time invariant and independent of $x(k)$.

Let

$$M_j(k) \triangleq \{M(k) = M_j\} \tag{4.96}$$

denote event that mode M_j is in effect at time k. The l^{th} modes history up to and including time k can be rewritten from (4.93) as

$$M^{k,l} = \{M^{k-1,s}, M_j(k)\} \tag{4.97}$$

$M^{k-1,s}$ is the sequence up to time k and M_j is the mode at time k.

Under the assumption of Markov property which says that the probability of transition to current state is defined only by the previous state one can write[32]

$$P\{M_j(k)|M^{k-1,s}\} = P\{M_j(k)|M_i(k-1)\} = p_{ij} \tag{4.98}$$

The posterior conditional probability density of the final estimate is

$$\begin{aligned} p\big(x(k)|Z^k\big) &= \sum_{l=1}^{r^k} p\big(x(k), M^{k,l}|Z^k\big) && (4.99) \\ &= \sum_{l=1}^{r^k} p\big(x(k)|M^{k,l}, Z^k\big) P\{M^{k,l}, Z^k\} && (4.100) \end{aligned}$$

which is a Gaussian mixture of all r^k histories.

Given that each mode history has to be associated with single Kalman filter this optimal approach is computationally infeasible as the number of filters would exponentially increase with time. To counter this a suboptimal technique is required.

There are two suboptimal techniques for the dynamic multiple model estimator suitable for practical applications. First technique is called generalized pseudo-Bayesian (GPB). The algorithm slightly differs based on its order. The first order GPB uses r filters in parallel whose output is combined at the end of each cycle to a single estimate. This estimate is then used as an input to the filters in the next iteration. The combination causes merging the histories and with them the hypotheses to ensure computational complexity similar to static multiple model estimator.[4]

The second order GPB estimator defers the decision to allow for consideration of each possible hypothesis in the last two periods. This means that there is a total number of r^2 hypotheses each represented by a single filter which are merged at the end of each cycle into r hypotheses.[4]

The second suboptimal technique is the interacting multiple mode estimator. This estimator will be derived in the following section.

4.5.4 Interacting multiple model estimator

The interacting multiple model (IMM) estimator is the current state of the art algorithm for the estimation of manoeuvring targets. Its main advantage is very good performance to computational complexity ratio. Compared to the previously mentioned generalized pseudo-Bayesian algorithm the interacting multiple mode has computational complexity similar to the first order GPB or the static MM estimator but the performance of the estimator is nearly as good as of the second order GPB.[16][33]

Since the introduction of the algorithm in 1984 by H. Blom the wide popularity and subsequent research even allowed for some extensions. Even though some as the variable structure interacting multiple model (VS-IMM) algorithm can benefit in certain situations the increase in performance is typically followed by an significant increase in the implementation complexity.[34][35][36]

Even though the interacting multiple model estimator is used at the forefront of the target tracking the good performance to computational ratio makes it suitable in our scenario. There is no need to track two thousand targets but moderate hardware presented at the ATM Laboratory should suffice for the tracking of couple hundred even with the use of this very advanced technique.

The derivation of IMM estimator is based on assuming that there is r mode-matched filters where each takes as an input different combination of previously estimated states. That is the posterior conditional probability density function of the final state estimate is an Gaussian mixture of r different filters.[33]

$$p\big(x(k)|Z^k\big) = \sum_{j=1}^{r} p\big(x(k), M_j(k)|Z^k\big) \tag{4.101}$$

$$= \sum_{j=1}^{r} p\big(x(k)|M_j(k), Z^k\big) P\{M_j(k)|Z^k\} \tag{4.102}$$

$$= \sum_{j=1}^{r} p\big(x(k)|M_j(k), z(k), Z^{k-1}\big)\mu_j(k) \tag{4.103}$$

The posterior conditional probability density function of the final state estimate of a single filter with model M_j is obtained using Bayes theorem.

$$p\big(x(k)|M_j(k), z(k), Z^{k-1}\big) = \frac{p\big(z(k)|M_j(k), x(k)\big)p\big(x(k)|M_j(k), Z^k\big)}{p\big(z(k)|M_j(k), Z^{k-1}\big)} \tag{4.104}$$

This equation is an equivalent to the fundamental equation of Bayesian estimation for filtering step (4.11). The left term in the numerator represents the measurement equation (4.92) in the form of probability density function. The term in the denominator is the likelihood function of mode M_j being in effect at time k derived by equations (4.80) through (4.84). The right term in the numerator is the probability density function of a predicted estimate by filter with model M_j and is in the IMM estimator derived in the following way.[4]

$$p\big(x(k)|M_j(k), Z^{k-1}\big) = \sum_{i=1}^{r} p\big(x(k), M_i(k-1)|M_j(k), Z^{k-1}\big) \tag{4.105}$$

$$= \sum_{i=1}^{r} p\big(x(k)|M_j(k), M_i(k-1), Z^{k-1}\big)P\{M_i(k-1)|M_j(k), Z^{k-1}\} \tag{4.106}$$

The rightmost term in the last equation is called mixing probability. This is a probability of mode M_i ,represented by a corresponding model, being in affect at time $k-1$ given mode M_j was in effect at time k conditioned on the previous measurements up to time $k-1$.[4]

$$\mu_{i|j}(k-1|k-1) \triangleq P\{M_i(k-1)|M_j(k), Z^{k-1}\} \tag{4.107}$$

$$= \frac{P\{M_j(k)|M_i(k-1), Z^{k-1}\}P\{M_i(k-1)|Z^{k-1}\}}{P\{M_j(k)|Z^{k-1}\}} \tag{4.108}$$

$$= \frac{P\{M_j(k)|M_i(k-1), Z^{k-1}\}P\{M_i(k-1)|Z^{k-1}\}}{\sum_{i=1}^{r} P\{M_j(k), M_i(k-1)|Z^{k-1}\}} \tag{4.109}$$

$$= \frac{P\{M_j(k)|M_i(k-1), Z^{k-1}\}P\{M_i(k-1)|Z^{k-1}\}}{\sum_{i=1}^{r} P\{M_j(k)|M_i(k-1), Z^{k-1}\}P\{M_i(k-1)|Z^{k-1}\}} \tag{4.110}$$

$$= \frac{p_{ij}\mu_i(k-1)}{\sum_{i=1}^{r} p_{ij}\mu_i(k-1)} \tag{4.111}$$

That is the mixing probability is under the Markov property defined solely by the mode transition probabilities and the probability of mode M_i being at effect at time $k-1$.

Now using the newly defined notation for the mixing probability we can rewrite the equation (4.106) in the following way.

$$p\big(x(k)|M_j(k), Z^{k-1}\big) = \sum_{i=1}^{r} p\big(x(k)|M_j(k), M_i(k-1), Z^{k-1}\big)\mu_{i|j}(k-1|k-1) \tag{4.112}$$

Substituting the previous equation back to (4.104) and this back to the equation (4.103) we obtain

$$
\begin{aligned}
p\big(x(k)|Z^k\big) &= \sum_{j=1}^{r} p\big(x(k)|M_j(k), z(k), Z^{k-1}\big)\mu_j(k) && (4.113)\\
&= \sum_{j=1}^{r} \frac{p\big(z(k)|M_j(k), x(k)\big)p\big(x(k)|M_j(k), Z^k\big)}{p\big(z(k)|M_j(k), Z^{k-1}\big)}\mu_j(k) && (4.114)\\
&= \sum_{j=1}^{r} \frac{p\big(z(k)|M_j(k), x(k)\big)\mu_j(k)}{p\big(z(k)|M_j(k), Z^{k-1}\big)}\\
&\quad \cdot \sum_{i=1}^{r} p\big(x(k)|M_j(k), M_i(k-1), Z^{k-1}\big)\mu_{i|j}(k-1|k-1) && (4.115)
\end{aligned}
$$

in which we can see that the calculation of the final estimate probability density function depends on r^2 terms and results in a requirement of r^2 filters. This is the same as for the second order generalized pseudo-Bayesian estimator. To allow the computational complexity reduction the interacting multiple model estimator approximates the probability density function of a predicted estimate (4.112) using moment matching technique.[4]

$$
\begin{aligned}
p\big(x(k)|M_j(k), Z^{k-1}\big) &= \sum_{i=1}^{r} p\big(x(k)|M_j(k), M_i(k-1), Z^{k-1}\big)\mu_{i|j}(k-1|k-1) && (4.116)\\
&= \sum_{i=1}^{r} \mathcal{N}\Big(x(k); E\left[x(k)|M_j(k), \hat{x}_i(k-1|k-1)\right],\\
&\qquad \operatorname{cov}\left[x(k)|M_j(k), P_i(k-1|k-1)\right]\Big)\mu_{i|j}(k-1|k-1) && (4.117)\\
&\approx \mathcal{N}\Big(x(k); E\left[x(k)|M_j(k), \hat{x}_j^0(k-1|k-1)\right],\\
&\qquad \operatorname{cov}\left[x(k)|M_j(k), P_j^0(k-1|k-1)\right]\Big) && (4.118)
\end{aligned}
$$

The moment matching technique approximates the Gaussian mixture (4.117) by a single Gaussian probability density function with its moment matched to the moments of the Gaussian mixture. The mean $\hat{x}_j^0(k-1|k-1)$ and the covariance $P_j^0(k-1|k-1)$ which are an input to the previously derived state prediction are computed according to (4.89) and (4.90) respectively.

$$
\begin{aligned}
\hat{x}_j^0(k-1|k-1) &= \sum_{i=1}^{r} \hat{x}_i(k-1|k-1)\mu_{i|j}(k-1|k-1) && (4.119)\\
P_j^0(k-1|k-1) &= \sum_{i=1}^{r} \mu_{i|j}(k-1|k-1)\Big(P_i(k-1|k-1)\\
&\quad + \big(\hat{x}_i(k-1|k-1) - \hat{x}_j^0(k-1|k-1)\big)\\
&\quad \cdot \big(\hat{x}_i(k-1|k-1) - \hat{x}_j^0(k-1|k-1)\big)'\Big) && (4.120)
\end{aligned}
$$

The moment matching technique for the probability density function of the predicted state is the fundamental principle o the IMM estimator that allows to maintain only r filters with slight reduction of performance when compared to second order GPB estimator. The mixed state $\hat{x}_j^0(k-1|k-1)$ and covariance $P_j^0(k-1|k-1)$ estimates matched to mode M_j, are obtained through interaction of r different filter. This is why the estimator is called interacting. The interaction is same as if the the different filters estimates were merged and the beginning of each estimation cycle instead of end as for GBP estimator.[4][34]

The algorithm of the interacting multiple model estimator with two different models can be seen in the picture 16 and consist of the following steps:[4]

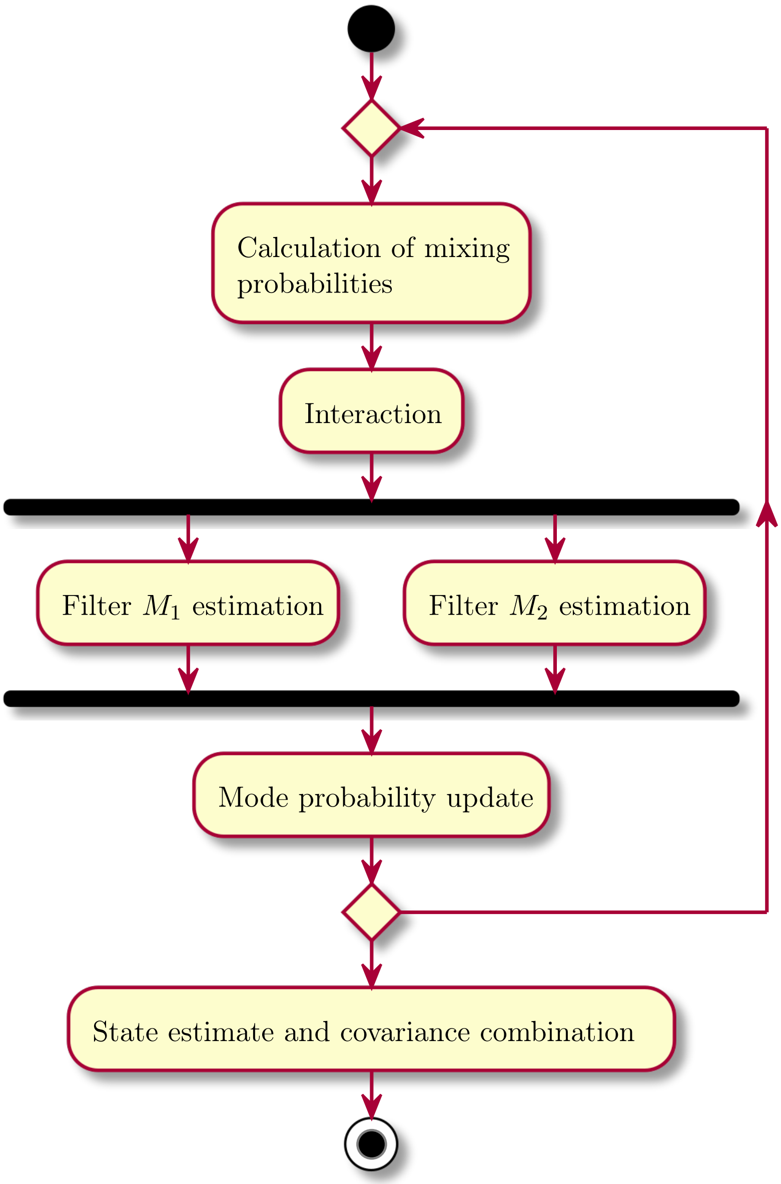

Figure 16: Interacting multiple model estimator

1. Calculation of mixing probabilities

$$\mu_{i|j}(k-1|k-1) = \frac{p_{ij}\mu_i(k-1)}{\sum_{i=1}^{r} p_{ij}\mu_i(k-1)} \tag{4.121}$$

2. Interaction

$$\hat{x}_j^0(k-1|k-1) = \sum_{i=1}^{r} \hat{x}_i(k-1|k-1)\mu_{i|j}(k-1|k-1) \tag{4.122}$$

$$\begin{aligned} P_j^0(k-1|k-1) = \sum_{i=1}^{r} \mu_{i|j}(k-1|k-1)\Big(&P_i(k-1|k-1) \\ &+ \big(\hat{x}_i(k-1|k-1) - \hat{x}_j^0(k-1|k-1)\big) \\ &\cdot \big(\hat{x}_i(k-1|k-1) - \hat{x}_j^0(k-1|k-1)\big)'\Big) \end{aligned} \tag{4.123}$$

3. Mode matched filtering
   The mixed estimates $\hat{x}_j^0(k-1|k-1)$, $P_j^0(k-1|k-1)$ are used as an input to a filter with model M_j. The filter then uses the obtained measurement $z(k)$ to calculate an updated estimate.

4. Mode probability update
   The probability of target being in mode M_j at time k is

$$\mu_j(k) \triangleq P\{M_j(k)|Z^k\} \tag{4.124}$$

$$= P\{M_j(k)|z(k), Z^{k-1}\} \tag{4.125}$$

$$= \frac{p\big(z(k)|M_j(k), Z^{k-1}\big)p\{M_j(k)|Z^{k-1}\}}{p\big(z(k)|Z^{k-1}\big)} \tag{4.126}$$

Now using (4.80)

$$\mu_j(k) = \frac{\Lambda_j(k)p\{M_j(k)|Z^{k-1}\}}{p\big(z(k)|Z^{k-1}\big)} \tag{4.127}$$

$$= \frac{\Lambda_j(k)\sum_{i=1}^r p\{M_j(k), M_i(k-1)|Z^{k-1}\}}{p\big(z(k)|Z^{k-1}\big)} \tag{4.128}$$

$$= \frac{\Lambda_j(k)\sum_{i=1}^r p\{M_j(k)|M_i(k-1), Z^{k-1}\}P\{M_i(k-1)|Z^{k-1}\}}{p\big(z(k)|Z^{k-1}\big)} \tag{4.129}$$

$$= \frac{\Lambda_j(k)\sum_{i=1}^r p_{ij}\mu_i(k-1)}{p\big(z(k)|Z^{k-1}\big)} \tag{4.130}$$

$$= \frac{\Lambda_j(k)\sum_{i=1}^r p_{ij}\mu_i(k-1)}{\sum_{j=1}^r p\big(z(k), M_j|Z^{k-1}\big)} \tag{4.131}$$

$$= \frac{\Lambda_j(k)\sum_{i=1}^r p_{ij}\mu_i(k-1)}{\sum_{j=1}^r p\big(z(k)|M_j, Z^{k-1}\big)P\{M_j(k)|Z^{k-1}\}} \tag{4.132}$$

$$= \frac{\Lambda_j(k)\sum_{i=1}^r p_{ij}\mu_i(k-1)}{\sum_{j=1}^r \Lambda_j(k)\sum_{i=1}^r P\{M_j(k), M_i(k-1)|Z^{k-1}\}} \tag{4.133}$$

$$= \frac{\Lambda_j(k)\sum_{i=1}^r p_{ij}\mu_i(k-1)}{\sum_{j=1}^r \Lambda_j(k)\sum_{i=1}^r P\{M_j(k)|M_i(k-1), Z^{k-1}\}P\{M_i(k-1)|Z^{k-1}\}} \tag{4.134}$$

$$= \frac{\Lambda_j(k)\sum_{i=1}^r p_{ij}\mu_i(k-1)}{\sum_{j=1}^r \Lambda_j(k)\sum_{i=1}^r p_{ij}\mu_i(k-1)} \tag{4.135}$$

These four steps are sufficient to perform the recursion of the interacting multiple model estimator. For output purposes there is also necessity to obtain a single final state estimate.

This is done obtaining the state and covariance as previously defined by the Gaussian mixture (4.103).

5. Estimate and covariance combination

$$\hat{x}(k|k) = \sum_{j=1}^{r} \hat{x}_j(k|k)\mu_j(k) \tag{4.136}$$

$$P(k|k) = \sum_{j=1}^{r} \mu_j(k) \left(P_j(k|k) + (\hat{x}_j(k|k) - \hat{x}(k|k))\,(\hat{x}_j(k|k) - \hat{x}(k|k))'\right) \tag{4.137}$$

4.6 Kalman filter vs interacting multiple model estimator

Some insufficiencies of the Kalman filter were presented in the section 4.4.1. There a simulation was done to show the compromise the Kalman filter tuning produces in the state estimation process. To combat this numerous adaptive techniques were developed. Currently the state of the art technique used in target tracking is the interacting multiple model estimator and its variations.[16]

The IMM estimator enables switching between different models that can vary in its motion model or only in the process noise covariance. When comparing the IMM estimator against single Kalman filter a significant performance increase is noted. A comparison of RMS position error of the Kalman filters from the section 4.4.1 for the same scenario against an IMM estimator which will be further specified later can be seen in the figure 17.[23][37]

From the figure a typical trend when comparing IMM estimator against the Kalman filter can be seen. The performance of the IMM estimator is significantly better in constant motion model against a single Kalman filter that is supposed to be sufficient for the whole scenario. When comparing against the Kalman filter with lower process noise the IMM estimator performs similarly or even better for the constant velocity motion. When a manoeuvre occurs the IMM estimator has a similar or even smaller RMS error when comparing against the manoeuvring Kalman filter and significantly better performance against the lower process noise Kalman filter.[16]

Even though the performance of the manoeuvre Kalman filter is under the position standard deviation of measurement it is not always the case. For significantly manoeuvring targets the RMS position error can easily exceed the measurement standard deviation and cause unacceptable errors. The IMM estimator is shown to keep the position RMS error below the measurement during or equal to the measurement standard deviation.[4][37]

Another big advantage of the IMM estimator against the Kalman filter is that the state estimate covariance matrix reacts to a manoeuvre. This allows effective use of the state covariance matrix in gating, data association, multi-sensor fusion and other situations. The non-reacting covariance matrix of a Kalman filter leads to false belief in a precision of a state estimate during manoeuvre. The reaction of the IMM estimator covariance matrix can be seen in the figure 18.[16]

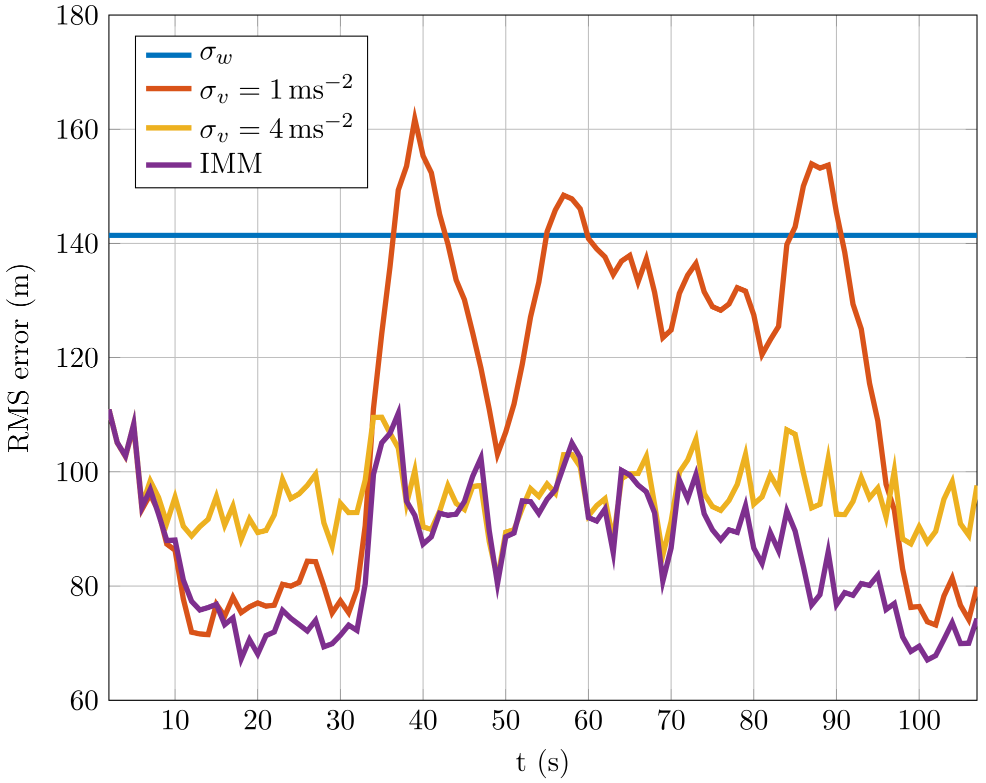

Figure 17: Kalman filter vs IMM

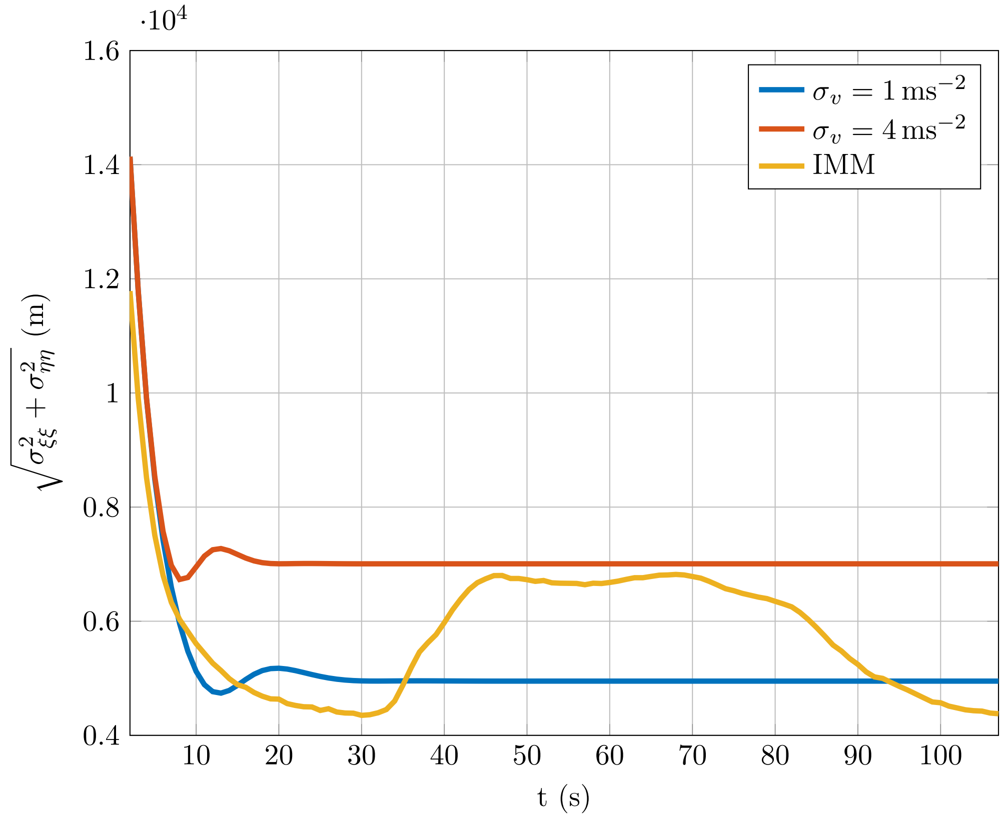

Figure 18: IMM state estimate covariance reaction

5 Interacting multiple model design

5.1 Coupling

At first when designing the interacting multiple model estimator for aircraft tracking a decision needs to be made how to carry out the three dimensional estimation. There are in general two approaches on how to do this.

- Coupled
- Decoupled in horizontal and vertical plane

When doing estimation for fully coupled motion the problems become inherently non-linear and the motion models get significantly complex. The estimation in fully coupled models is an intriguing problem where the advantage lies if the target really does the three dimensional coupled motion. These target are mainly military aircraft. On the other hand the linearisation and the additional computational complexity and the cost of implementation is not outweighed by the performance increase. This was shown for example in [38] where fully coupled IMM was compared against decoupled IMM for an air defence applications. The results showed that the coupled IMM performed better only when the aircraft were doing three dimensional manoeuvres. The authors even concluded that tracking in decoupled coordinates performs well for targets manoeuvring in all three dimensions.

As the developed tracker focuses on tracking civilian aircraft with mode S transponder it is reasonable to assume that the target motions are decoupled in horizontal and vertical plane. The independence between the horizontal and vertical plane can be assumed since most of the manoeuvres of civilian aircraft such as turning are done in horizontal plane only. Of course there are always deviations but these deviations can be easily covered by a slight process noise in a motion model.[17][39]

5.2 Individual models

5.2.1 Design considerations

Simplest approach for aircraft tracking with decoupled horizontal and vertical plane is to decouple even the horizontal coordinates and only use nearly constant velocity and nearly constant acceleration models for each coordinate independently. The constant velocity model performs well during non-manoeuvring motion and the nearly constant acceleration model can cover manoeuvres including turns.

For the tracking of civilian aircraft it is assumed that most of the manoeuvres the aircraft perform in horizontal plane are coordinated horizontal turns. For this a motion model known as nearly coordinated turn or nearly constant turn rate was developed to increase performance during not only the coordinated turn but also a turn in general.[38][39]

For civilian aircraft doing coordinated turn the so called Air Traffic Control tracking was already described in literature as [4] and [37]. In this literature the authors focused on ATC tracking using two motion models for the horizontal plane. They used nearly constant velocity motion model and nearly coordinated turn motion model with state vector consisting of

$$x = \begin{bmatrix} \xi \\ \dot{\xi} \\ \eta \\ \dot{\eta} \end{bmatrix} \tag{5.1}$$

for the horizontal plane.

Interacting multiple model with these motion models and the state vector above showed very good performance in the ATC tracking. For our purposes we will augment the state vector for the horizontal plane by acceleration to allow for linear accelerated motion.[4][37]

$$x_h = \begin{bmatrix} \xi \\ \dot{\xi} \\ \ddot{\xi} \\ \eta \\ \dot{\eta} \\ \ddot{\eta} \end{bmatrix} \tag{5.2}$$

With this the motion models used in the designed tracker will consist of

- Nearly constant velocity model
- Nearly constant acceleration model
- Nearly coordinated turn model

The word nearly in this regard represents the fact that the process noise is present and thus there can be slight deviations from the defined motion model.

For tracking in the vertical plane a similar state vector but only for one coordinate will be used

$$x_v = \begin{bmatrix} \zeta \\ \dot{\zeta} \\ \ddot{\zeta} \end{bmatrix} \tag{5.3}$$

where ζ represents the vertical coordinate.

The vertical coordinate tracking will be guided by only two models due to the decoupled motion nature.

- Nearly constant velocity model
- Nearly constant acceleration model

These two models should be sufficient to cover constant velocity climbing and a constant acceleration in vertical plane.

For the vertical motion and also in the horizontal plane the nearly constant acceleration model can be modified to use a larger process noise covariance to also cover manoeuvres. Then one should talk about continuous Wiener process acceleration model or discrete Wiener process acceleration model as the acceleration does not change only slightly. For our purposes the acceleration model with large process noise covariance matrix might be used to help cover large deviations caused by a large measurement noise.

When deriving the motion model the primary approach is to discretise the continuous kinematic equations. This relatively lengthy derivation can be substituted by using direct discrete approach. In the end the only difference in the motion models presented above is in the process noise covariance matrix. When doing preliminary testing of the discretised continuous motion models against the direct discrete motion models no performance difference was found. Although the discretisation is mathematically more correct approach in both cases the motion models are only approximations. The main advantage of the direct discrete motion models, which will be presented in the next sections, is the representation of tuning constant for the process noise. For the discretised case it is represented by power spectral density. For the direct discrete models it is represented by variance or standard deviation of acceleration which is a more natural quantity to work with.[4]

5.2.2 Discrete white noise acceleration model

Nearly constant velocity model is a motion model describing non-manoeuvring targets. This model is a special form of discrete white noise acceleration (DWNA) model where the changes in the velocity caused by the process noise are small compared to the actual velocity of the target. The DWNA model is derived in the following way.[4]

Assuming dynamic equation for the nearly constant velocity motion model in a single coordinate with a state vector $x(k)$ equivalent to

$$x(k) = \begin{bmatrix} \xi \\ \dot{\xi} \\ \ddot{\xi} \end{bmatrix} \tag{5.4}$$

we get

$$x(k+1) = Fx(k) + \Gamma v(k) \tag{5.5}$$

where Γ is the n_x dimensional noise gain vector and the discrete-time process noise $v(k)$ is a zero-mean white sequence

$$E[v(k)v(j)] = \sigma_v^2 \delta_{ij} \tag{5.6}$$

The assumption of the discrete white noise acceleration model is that the target undergoes a constant acceleration in each sampling interval and these acceleration are uncorrelated between periods. This acceleration is caused by the process noise $v(k)$. Although this assumption is clearly unrealistic it is successful in modelling motion uncertainties.[4]

Given that we assume the process noise is the acceleration during each sampling period the standard deviation of $v(k)$ then has the units of acceleration.

The transition matrix F is derived from the kinematic equations of the constant velocity motion and has the following form.

$$F = \begin{bmatrix} 1 & t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.7}$$

where t is the length of the k$^{\text{th}}$ sampling period. The bottom row is zero as the constant velocity motion assumes the acceleration to be equal to zero.

The velocity increment from the constant acceleration represented by the process noise gain Γ follows from the equations of the linear accelerated motion.

$$\Gamma = \begin{bmatrix} \frac{1}{2}t^2 \\ t \\ 0 \end{bmatrix} \tag{5.8}$$

Again the zero in the bottom row is only to augment the state vector as the acceleration is assumed to be zero.

The process noise covariance matrix is then

$$Q = E[\Gamma v(k) v(k) \Gamma'] \tag{5.9}$$

$$= \Gamma E[v(k) v(k)] \Gamma' \tag{5.10}$$

$$= \Gamma \sigma_v^2 \Gamma' \tag{5.11}$$

$$= \begin{bmatrix} \frac{1}{4}t^4 & \frac{1}{2}t^3 & 0 \\ \frac{1}{2}t^3 & t^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \sigma_v^2 \tag{5.12}$$

For the two-point differencing initialization method we also need to derive the initial state estimate covariance matrix. This is done in the following way.

Given that we initialize the state vector using (4.50) we get for our augmented vector

$$\hat{x}(0|0) = \begin{bmatrix} \hat{x}_1(0|0) \\ \hat{x}_2(0|0) \\ \hat{x}_3(0|0) \end{bmatrix} = \begin{bmatrix} z(0) \\ \frac{z(0) - z(-1)}{t} \\ 0 \end{bmatrix} \tag{5.13}$$

Now from the state estimate covariance definition (4.51) we know the form of each element

$$P_{ij}(0|0) = E[\tilde{x}_i(0|0) \tilde{x}_j(0|0)'] \tag{5.14}$$

The whole covariance matrix for this case is

$$P(0|0) = \begin{bmatrix} E[\tilde{x}_1(0|0)\tilde{x}_1(0|0)'] & E[\tilde{x}_1(0|0)\tilde{x}_2(0|0)'] & 0 \\ E[\tilde{x}_2(0|0)\tilde{x}_1(0|0)'] & E[\tilde{x}_2(0|0)\tilde{x}_2(0|0)'] & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.15}$$

The necessary estimation errors have the following form.

$$\tilde{x}_1(0|0) = x_1(0) - \hat{x}_1(0|0) \tag{5.16}$$
$$= x_1(0) - x_1(0) - w(0) \tag{5.17}$$
$$= -w(0) \tag{5.18}$$

$$\tilde{x}_2(0|0) = x_2(0) - \hat{x}_2(0|0) \tag{5.19}$$
$$= x_2(-1) + v_2(-1) - \frac{z(0) - z(-1)}{t} \tag{5.20}$$
$$= x_2(-1) + v_2(-1) - \frac{1}{t}\big(x_1(-1) + x_2(-1)t + v_1(-1) + w(0) - x_1(-1) - w_1(-1)\big) \tag{5.21}$$
$$= v_2(-1) - \frac{1}{t}\big(v_1(-1) - w(-1) + w(0)\big) \tag{5.22}$$

The index for the measurement noise $w(k)$ was omitted as we assume only position measurements and this derivation is done for single coordinate.

The elements of the initial state covariance matrix are then

$$P_{11}(0|0) = E[\tilde{x}_1(0|0)\tilde{x}_1(0|0)'] \tag{5.23}$$
$$= E\left[(-w(0))(-w(0))\right] \tag{5.24}$$
$$= \sigma_w^2(0) \tag{5.25}$$

$$P_{12}(0|0) = E[\tilde{x}_1(0|0)\tilde{x}_2(0|0)'] \tag{5.26}$$
$$= E\left[-w(0)\left(v_2(-1) - \frac{1}{t}\big(v_1(-1) - w(-1) + w(0)\big)\right)\right] \tag{5.27}$$
$$= \frac{1}{t}\sigma_w^2(0) \tag{5.28}$$
$$\tag{5.29}$$

From the properties of a covariance matrix it follows that

$$P_{21}(0|0) = P_{12}(0|0) \tag{5.30}$$

The last missing element in the initial state estimate covariance matrix, assuming constant variance of the process noise, is

$$P_{22}(0|0) = E[\tilde{x}_2(0|0)\tilde{x}_2(0|0)'] \tag{5.31}$$
$$= E\left[\left(v_2(-1) - \frac{1}{t}\big(v_1(-1) - w(-1) + w(0)\big)\right)^2\right] \tag{5.32}$$
$$= \frac{t^2}{4}\sigma_v^2 + \frac{1}{t^2}\left(\sigma_w^2(0) + \sigma_w^2(-1)\right) \tag{5.33}$$

The previous equations are valid for a single coordinate tracking. It fully describes the DWNA model for the vertical coordinate ζ. For the tracking in a horizontal plane it is assumed for the non-manoeuvring motion that the coordinates are decoupled. In that regard we obtain the state transition matrix, the covariance matrix of the process noise and the initial state estimate covariance matrix by stacking.

The transition matrix has the form

$$\begin{bmatrix} F & 0 \\ 0 & F \end{bmatrix} \tag{5.34}$$

The process noise covariance

$$\begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} \tag{5.35}$$

The initial state estimate covariance matrix is

$$\begin{bmatrix} P(0|0) & 0 \\ 0 & P(0|0) \end{bmatrix} \tag{5.36}$$

5.2.3 Discrete Wiener process acceleration model

The nearly constant acceleration model is a special case of discrete Wiener process acceleration (DWPA) model when the change in acceleration is relatively small compared to actual acceleration. It is assumed that the acceleration increment in a single period is a discrete-time Wiener process. That is the jerk is a zero-mean white noise sequence as it is a derivative of Wiener process.[4]

The transition matrix for a single coordinate is derived from the kinematic equations of linear accelerated motion and has the form.

$$\begin{bmatrix} 1 & t & \frac{1}{2}t^2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} \tag{5.37}$$

Assuming the process noise standard deviation σ_v also represents the acceleration same as for DWNA model the noise gain vector has the form

$$\Gamma = \begin{bmatrix} \frac{1}{2}t^2 \\ t \\ 1 \end{bmatrix} \tag{5.38}$$

The process noise covariance matrix is then

$$Q = \begin{bmatrix} \frac{1}{4}t^4 & \frac{1}{2}t^3 & \frac{1}{2}t^2 \\ \frac{1}{2}t^3 & t^2 & t \\ \frac{1}{2}t^2 & t & 1 \end{bmatrix} \sigma_v^2 \tag{5.39}$$

For the two-point differencing the initial state estimate covariance matrix is derived in the same way as for the DWNA model.

$$P(0|0) = \begin{bmatrix} E[\tilde{x}_1(0|0)\tilde{x}_1(0|0)'] & E[\tilde{x}_1(0|0)\tilde{x}_2(0|0)'] & E[\tilde{x}_1(0|0)\tilde{x}_3(0|0)'] \\ E[\tilde{x}_2(0|0)\tilde{x}_1(0|0)'] & E[\tilde{x}_2(0|0)\tilde{x}_2(0|0)'] & E[\tilde{x}_2(0|0)\tilde{x}_3(0|0)'] \\ E[\tilde{x}_3(0|0)\tilde{x}_1(0|0)'] & E[\tilde{x}_3(0|0)\tilde{x}_2(0|0)'] & E[\tilde{x}_3(0|0)\tilde{x}_3(0|0)'] \end{bmatrix} \tag{5.40}$$

The estimation errors are

$$\begin{align} \tilde{x}_1(0|0) &= x_1(0) - \hat{x}_1(0|0) \tag{5.41} \\ &= x_1(0) - x_1(0) - w(0) \tag{5.42} \\ &= -w(0) \tag{5.43} \end{align}$$

$$\begin{align} \tilde{x}_1(0|0) &= x_2(0) - \hat{x}_2(0|0) \tag{5.44} \\ &= x_2(-1) + x_3(-1)t + v_2(-1) - \frac{z(0) - z(-1)}{t} \tag{5.45} \\ &= x_2(-1) + x_3(-1)t + v_2(-1) - \frac{1}{t}\Big(x_1(-1) + x_2(-1)t \nonumber \\ &\quad + x_3(-1)\frac{t^2}{2} + v_1(-1) \ + w(0) - x_1(-1) - w(-1)\Big) \tag{5.46} \\ &= v_2(-1) + \frac{t}{2}x_3(-1) - \frac{1}{t}\big(v_1(-1) + w(0) - w(1)\big) \tag{5.47} \end{align}$$

$$\begin{align} \tilde{x}_3(0|0) &= x_3(0) - \hat{x}_3(0|0) \tag{5.48} \\ &= x_3(-1) + v_3(-1) \tag{5.49} \end{align}$$

The elements of the initial state covariance matrix are

$$\begin{align} P_{11}(0|0) &= E[\tilde{x}_1(0|0)\tilde{x}_1(0|0)'] \tag{5.50} \\ &= E\left[(-w(0))(-w(0))\right] \tag{5.51} \\ &= \sigma_w^2(0) \tag{5.52} \end{align}$$

$$\begin{align} P_{12}(0|0) &= E[\tilde{x}_1(0|0)\tilde{x}_2(0|0)'] \tag{5.53} \\ &= E\left[-w(0)\left(v_2(-1) + \frac{t}{2}x_3(-1) - \frac{1}{t}\big(v_1(-1) + w(0) - w(1)\big)\right)\right] \tag{5.54} \\ &= \frac{1}{t}\sigma_w^2(0) \tag{5.55} \end{align}$$

$$\begin{align} P_{13}(0|0) &= E[\tilde{x}_1(0|0)\tilde{x}_3(0|0)'] \tag{5.56} \\ &= E\left[-w(0)(x_3(-1) + v_3(-1))\right] \tag{5.57} \\ &= 0 \tag{5.58} \end{align}$$

$$P_{22} = E[\tilde{x}_2(0|0)\tilde{x}_2(0|0)'] \tag{5.59}$$

$$= E\left[\left(v_2(-1) + \frac{t}{2}x_3(-1) - \frac{1}{t}\big(v_1(-1) + w(0) - w(1)\big)\right)^2\right] \tag{5.60}$$

$$= \frac{t^2}{4}\left(\sigma_v^2 + \sigma_{\hat{x}_3(-1)}^2\right) + \frac{1}{t^2}\left(\sigma_w^2(0) + \sigma_w^2(-1)\right) \tag{5.61}$$

where $\sigma_{\hat{x}_3(-1)}^2$ is the variance of the initial acceleration estimate.

$$P_{23}(0|0) = E[\tilde{x}_2(0|0)\tilde{x}_3(0|0)'] \tag{5.62}$$

$$= E\left[\left(v_2(-1) + \frac{t}{2}x_3(-1) - \frac{1}{t}\big(v_1(-1) + w(0) - w(1)\big)\right)\big(x_3(-1) + v_3(-1)\big)\right] \tag{5.63}$$

$$= \frac{t}{2}\left(\sigma_{\hat{x}_3(-1)}^2 + \sigma_v^2\right) \tag{5.64}$$

$$P_{33}(0|0) = E[\tilde{x}_3(0|0)\tilde{x}_3(0|0)'] \tag{5.65}$$

$$= E\left[\big(x_3(-1) + v_3(-1)\big)^2\right] \tag{5.66}$$

$$= \sigma_{\hat{x}_3(-1)}^2 + \sigma_v^2 \tag{5.67}$$

It follows from the properties of a covariance matrix that

$$P_{21}(0|0) = P_{12}(0|0) \tag{5.68}$$

$$P_{31}(0|0) = P_{13}(0|0) \tag{5.69}$$

$$P_{32}(0|0) = P_{23}(0|0) \tag{5.70}$$

The previous derivations were done again only for a single coordinate motion. For the tracking in a horizontal plane it is also assumed that the coordinates are decoupled. This is allowed because for our application this motion model should be the most probable only for linear accelerated motion. Otherwise for turns the nearly coordinated turn model with coupled ξ and η coordinates will be used. With that the transition matrix, the process noise covariance matrix and the initial state estimate covariance are expanded by stacking as was done for the DWNA model.

5.2.4 Nearly constant turn rate model

For tracking manoeuvres in the horizontal plane a nearly constant turn rate also known as nearly coordinated turn seems as a very suitable option. This model was shown to provide more accurate estimation for turn manoeuvres than a DWPA model.[4][37][40]

The motion model assumes constant speed and turn rate (angular velocity) accompanied by a zero mean white Gaussian noise. Using the assumptions derivation of the state transition matrix for a direct discrete case can be done in the following way.[17]

The known motion equations for components of the velocity in Cartesian coordinate system are

$$\dot{\xi}(k) = s\cos\varphi \tag{5.71}$$
$$\dot{\eta}(k) = s\sin\varphi \tag{5.72}$$

where s denotes the constant speed and the φ denotes the „heading“ angle assuming the heading uses mathematical convention i.e. starting from first quadrant counter clockwise.

The predicted velocity components after a time period t are

$$\begin{aligned}
\dot{\xi}(k+1) &= s\cos(\omega(k)t+\varphi) &(5.73)\\
&= s\cos(\omega(k)t)\cos\varphi - s\sin(\omega(k)t)\sin\varphi &(5.74)\\
&= \dot{\xi}(k)\cos(\omega(k)t) - \dot{\eta}(k)\sin(\omega(k)t) &(5.75)
\end{aligned}$$

$$\begin{aligned}
\dot{\eta}(k+1) &= s\sin(\omega(k)t+\varphi) &(5.76)\\
&= s\sin(\omega(k)t)\cos\varphi + s\cos(\omega(k)t)\sin\varphi &(5.77)\\
&= \dot{\xi}(k)\sin(\omega(k)t) + \dot{\eta}(k)\cos(\omega(k)t) &(5.78)
\end{aligned}$$

The predicted position is then derived in the following way.

$$\begin{aligned}
\xi(k+1) &= \xi(k) + \int_0^t s\cos(\omega(k)\tau+\varphi)d\tau &(5.79)\\
&= \xi(k) + s\left(\frac{\sin(\omega(k)t+\varphi)}{\omega(k)} - \frac{\sin\varphi}{\omega(k)}\right) &(5.80)\\
&= \xi(k) + s\left(\frac{\sin(\omega(k)t)\cos\varphi}{\omega(k)} + \frac{\cos(\omega(k)t)\sin\varphi}{\omega(k)} - \frac{\sin\varphi}{\omega(k)}\right) &(5.81)\\
&= \dot{\xi}(k)\frac{\sin(\omega(k)t)}{\omega(k)} - \dot{\eta}(k)\frac{1-\cos(\omega(k)t)}{\omega(k)} &(5.82)
\end{aligned}$$

$$\begin{aligned}
\eta(k+1) &= \eta(k) + \int_0^t s\sin(\omega(k)\tau+\varphi)d\tau &(5.83)\\
&= \eta(k) + s\left(\frac{\cos\varphi}{\omega(k)} - \frac{\cos(\omega(k)t+\varphi)}{\omega(k)}\right) &(5.84)\\
&= \eta(k) + s\left(\frac{\cos\varphi}{\omega(k)} - \frac{\cos(\omega(k)t)\cos\varphi}{\omega(k)} + \frac{\sin(\omega(k)t)\sin\varphi}{\omega(k)}\right) &(5.85)\\
&= \eta(k) + \dot{\xi}(k)\frac{1-\cos(\omega(k)t)}{\omega(k)} + \dot{\eta}(k)\frac{\sin(\omega(k)t)}{\omega(k)} &(5.86)
\end{aligned}$$

For a state vector consisting of

$$x(k) = \begin{bmatrix}\xi\\ \dot{\xi}\\ \eta\\ \dot{\eta}\\ \omega\end{bmatrix} \tag{5.87}$$

using the previous expressions we can write the state transition matrix as

$$F = \begin{bmatrix} 1 & \frac{\sin(\omega(k)t)}{\omega(k)} & 0 & -\frac{1-\cos(\omega(k)t)}{\omega(k)} & 0 \\ 0 & \cos(\omega(k)t) & 0 & -\sin(\omega(k)t) & 0 \\ 0 & \frac{1-\cos(\omega(k)t)}{\omega(k)} & 1 & \frac{\sin(\omega(k)t)}{\omega(k)} & 0 \\ 0 & \sin(\omega(k)t) & 0 & \cos(\omega(k)t) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.88}$$

Assuming the process noise in the form

$$v(k) = \begin{bmatrix} v_a \\ v_a \\ v_{\dot{\omega}} \end{bmatrix} \tag{5.89}$$

where v_a is the noise in the form of acceleration and the $v_{\dot{\omega}}$ is the noise in the form of angular acceleration. The noise gain matrix representing the noise increments then has the form

$$\Gamma = \begin{bmatrix} \frac{1}{2}t^2 & 0 & 0 \\ t & 0 & 0 \\ 0 & \frac{1}{2}t^2 & 0 \\ 0 & t & 0 \\ 0 & 0 & t \end{bmatrix} \tag{5.90}$$

The process noise covariance is then

$$Q = E[\Gamma v(k) v(k) \Gamma'] \tag{5.91}$$

$$= \Gamma E[v(k) v(k)] \Gamma' \tag{5.92}$$

$$= \Gamma \begin{bmatrix} \sigma_{v_a}^2 & 0 & 0 \\ 0 & \sigma_{v_a}^2 & 0 \\ 0 & 0 & \sigma_{v_{\dot{\omega}}}^2 \end{bmatrix} \Gamma' \tag{5.93}$$

Looking at the direct discrete dynamic equation (5.5) it can be seen, assuming the angular velocity not to be known, that the estimation problem of nearly coordinate turn model is inherently non-linear. The dynamic equation (5.5) for non-linear case is

$$x(k+1) = f(k, x(k)) + \Gamma v(k) \tag{5.94}$$

With the non-linearity for the state estimation the use of the extended Kalman filter is required.

The state prediction of the extended Kalman filter (4.55) is done directly using motion equations (5.75),(5.78),(5.82) and (5.86). Looking at the equations for the prediction of position

(5.82),(5.86) it can be seen that the functions domains are not defined for $\omega = 0$. For this the limiting form of the position prediction is necessary.

$$\begin{aligned}
\lim_{\omega \to 0} \xi(k+1) &= \lim_{\omega \to 0} \xi(k) + \dot{\xi}(k) \frac{\sin(\omega(k)t)}{\omega(k)} - \dot{\eta}(k) \frac{1 - \cos(\omega(k)t)}{\omega(k)} \quad (5.95) \\
&= \lim_{\omega \to 0} \xi(k) + \dot{\xi}(k) \cos(\omega(k)t)t - \dot{\eta}(k) \sin(\omega(k)t)t \quad (5.96) \\
&= \xi(k) + \dot{\xi}t \quad (5.97)
\end{aligned}$$

$$\begin{aligned}
\lim_{\omega \to 0} \eta(k+1) &= \lim_{\omega \to 0} \eta(k) + \dot{\xi}(k) \frac{1 - \cos(\omega(k)t)}{\omega(k)} + \dot{\eta}(k) \frac{\sin(\omega(k)t)}{\omega(k)} \quad (5.98) \\
&= \lim_{\omega \to 0} \eta(k) + \dot{\xi}(k) \sin(\omega(k)t)t + \dot{\eta}(k) \cos(\omega(k)t)t \quad (5.99) \\
&= \eta(k) + \dot{\eta}t \quad (5.100)
\end{aligned}$$

Next to calculate the state estimate covariance prediction (4.56) the Jacobian (4.57) must be derived.

For the equation predicting the ξ coordinate (5.82) the Jacobian elements are

$$\frac{\partial \xi(k+1)}{\partial \xi} = 1 \tag{5.101}$$

$$\frac{\partial \xi(k+1)}{\partial \dot{\xi}} = \frac{\sin(\hat{\omega}(k)t)}{\hat{\omega}(k)} \tag{5.102}$$

$$\frac{\partial \xi(k+1)}{\partial \eta} = 0 \tag{5.103}$$

$$\frac{\partial \xi(k+1)}{\partial \dot{\eta}} = -\frac{1 - \cos(\hat{\omega}(k)t)}{\hat{\omega}(k)} \tag{5.104}$$

$$\frac{\partial \xi(k+1)}{\partial \hat{\omega}} = \frac{\hat{\dot{\xi}}(k)t \cos(\hat{\omega}(k)t)}{\hat{\omega}(k)} - \frac{\hat{\dot{\xi}}(k) \sin(\hat{\omega}(k)t)}{\hat{\omega}^2(k)} - \frac{\hat{\dot{\eta}}(k)t \sin(\hat{\omega}(k)t)}{\hat{\omega}(k)} + \frac{\hat{\dot{\eta}}(k)(1 - \cos(\hat{\omega}(k)t)}{\hat{\omega}^2(k)} \tag{5.105}$$

The Jacobian segment belonging to the velocity prediction $\xi(k+1)$ (5.75) is

$$\frac{\partial \dot{\xi}(k+1)}{\partial \xi} = 0 \tag{5.106}$$

$$\frac{\partial \dot{\xi}(k+1)}{\partial \dot{\xi}} = \cos(\hat{\omega}(k)t) \tag{5.107}$$

$$\frac{\partial \dot{\xi}(k+1)}{\partial \eta} = 0 \tag{5.108}$$

$$\frac{\partial \dot{\xi}(k+1)}{\partial \dot{\eta}} = -\sin(\hat{\omega}(k)t) \tag{5.109}$$

$$\frac{\partial \dot{\xi}(k+1)}{\partial \hat{\omega}} = -\hat{\dot{\xi}}t \sin(\hat{\omega}(k)t) - \hat{\dot{\eta}}t \cos(\hat{\omega}(k)t) \tag{5.110}$$

The segment belonging to the η coordinate prediction equation (5.86) is

$$\frac{\partial\eta(k+1)}{\partial\xi} = 0 \tag{5.111}$$

$$\frac{\partial\eta(k+1)}{\partial\dot{\xi}} = \frac{1-\cos(\hat{\omega}(k)t)}{\hat{\omega}(k)} \tag{5.112}$$

$$\frac{\partial\eta(k+1)}{\partial\eta} = 1 \tag{5.113}$$

$$\frac{\partial\eta(k+1)}{\partial\dot{\eta}} = \frac{\sin(\hat{\omega}(k)t)}{\hat{\omega}(k)} \tag{5.114}$$

$$\frac{\partial\eta(k+1)}{\partial\hat{\omega}} = \frac{\hat{\dot{\xi}}(k)t\sin(\hat{\omega}(k)t)}{\hat{\omega}(k)} - \frac{\hat{\dot{\xi}}(k)(1-\cos(\hat{\omega}(k)t)}{\hat{\omega}^2(k)} + \frac{\hat{\dot{\eta}}(k)t\cos(\hat{\omega}(k)t)}{\hat{\omega}(k)} - \frac{\hat{\dot{\eta}}(k)\sin(\hat{\omega}(k)t)}{\hat{\omega}^2(k)} \tag{5.115}$$

The last row of the Jacobian,under the assumption of constant angular velocity, is

$$\begin{bmatrix}0 & 0 & 0 & 0 & 1\end{bmatrix} \tag{5.116}$$

Same as for the state prediction not all elements of the Jacobian are defined for angular velocity equal to zero. For this the limit of the Jacobian has the form

$$\lim_{\hat{\omega}\to 0} f_x(k) = \begin{bmatrix} 1 & t & 0 & 0 & -\hat{\dot{\eta}}(k)\frac{t^2}{2} \\ 0 & 1 & 0 & 0 & -\hat{\dot{\eta}}(k)t \\ 0 & 0 & 1 & t & \hat{\dot{\xi}}(k)\frac{t^2}{2} \\ 0 & 0 & 0 & 1 & \hat{\dot{\xi}}(k)t \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.117}$$

To calculate the initial state estimate covariance matrix for the initialization of the extended Kalman filter the two-point differencing method is used. Same as it is impossible to estimate acceleration from only two position measurements the initial state estimate of the angular velocity is set to zero. Proceeding with this estimate it is reasonable to assume the limiting form of the state transition which using the expressions (5.97), (5.100) and substituting $\omega = 0$ to (5.75) and (5.78) we obtain an alternative form of the state transition matrix (5.88).

$$F = \begin{bmatrix} 1 & t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & t & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.118}$$

From the state transition matrix and from the noise gain (5.90) it can be seen that the state dynamics is same as for the discrete white noise acceleration model (5.7) which describes the nearly constant velocity motion. Because of this the derivation of the initial state estimate covariance matrix is same as for the DWNA model using equations (5.16) through (5.33).

Only difference is the necessity to select a variance of the initial state estimate of the angular velocity.This will be done similarly as for the DWPA model acceleration (5.67).

The initial state estimate covariance matrix of the nearly coordinated turn has the form

$$\begin{bmatrix} \sigma_w^2(0) & \frac{1}{t}\sigma_w^2(0) & 0 & 0 & 0 \\ \frac{1}{t}\sigma_w^2(0) & \frac{t^2}{4}\sigma_{v_a}^2 + \frac{1}{t^2}\left(\sigma_w^2(0) + \sigma_w^2(-1)\right) & 0 & 0 & 0 \\ 0 & 0 & \sigma_w^2(0) & \frac{1}{t}\sigma_w^2(0) & 0 \\ 0 & 0 & \frac{1}{t}\sigma_w^2(0) & \frac{t^2}{4}\sigma_{v_a}^2 + \frac{1}{t^2}\left(\sigma_w^2(0) + \sigma_w^2(-1)\right) & 0 \\ 0 & 0 & 0 & 0 & \sigma_{\hat{x}_5(-1)} + \sigma_{v_{\dot{\omega}}}^2 \end{bmatrix} \tag{5.119}$$

The previous equations are fully sufficient for the extended Kalman filter algorithm. The interaction step of the interacting multiple model algorithm requires all models to have the same state to allow for the computation of the equations (4.122) and (4.123). A transformation between the different state vectors thus needs to be derived.

The transformation from state vector of the nearly coordinated turn model denoted by index T to the state vector of the DWNA and DWPA model denoted by index D is defined by

$$\xi_D(k) = \xi_T(k) \tag{5.120}$$

$$\dot{\xi}_D(k) = \dot{\xi}_T(k) \tag{5.121}$$

$$\ddot{\xi}_D(k) = \frac{\partial s \cos\varphi}{\partial t} = -s\omega(k)\sin\varphi = -\omega(k)\dot{\eta}_T(k) \tag{5.122}$$

$$\eta_D(k) = \eta_T(k) \tag{5.123}$$

$$\dot{\eta}_D(k) = \dot{\eta}_T(k) \tag{5.124}$$

$$\ddot{\eta}_D(k) = \frac{\partial s \sin\varphi}{\partial t} = s\omega(k)\cos\varphi = \omega(k)\dot{\xi}_T(k) \tag{5.125}$$

Now assuming the first order Taylor series expansion the state estimate covariance matrix of the nearly coordinated turn model will be transformed using

$$P_D(k|k) = A_{TD} P_T(k|k) A'_{TD} \tag{5.126}$$

where

$$A_{TD} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\omega(k) & -\dot{\eta}_D(k) \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & \omega(k) & 0 & 0 & \dot{\xi}_D(k) \end{bmatrix} \tag{5.127}$$

is the Jacobian derived from the equations (5.120) through (5.125).

Transformation in the opposite direction is necessary for setting the initial state of the nearly constant turn rate model with a mixed state. The state transformation is done as follows.[17]

$$\xi_T(k) = \xi_D(k) \tag{5.128}$$

$$\dot{\xi}_T(k) = \dot{\xi}_D(k) \tag{5.129}$$

$$\eta_T(k) = \eta_D(k) \tag{5.130}$$

$$\dot{\eta}_T(k) = \dot{\eta}_D(k) \tag{5.131}$$

$$\omega(k) = \frac{\dot{\xi}_D(k)\ddot{\eta}_D(k) - \dot{\eta}_D(k)\ddot{\xi}_D(k)}{s^2} \tag{5.132}$$

The Jacobian for the transformation of the state estimate covariance matrix of the DWNA or DWPA model is

$$A_{DT} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & A_{DT_{52}} & -\frac{\dot{\eta}(k)}{s^2} & 0 & A_{DT_{55}} & \frac{\dot{\xi}(k)}{s^2} \end{bmatrix} \tag{5.133}$$

where

$$A_{DT_{52}} = \frac{\ddot{\eta}(k)\left(\dot{\eta}^2(k) - \dot{\xi}^2(k)\right) + 2\dot{\xi}(k)\dot{\eta}(k)\ddot{\xi}(k)}{s^4} \tag{5.134}$$

$$A_{DT_{55}} = \frac{\ddot{\xi}(k)\left(\dot{\eta}^2(k) - \dot{\xi}^2(k)\right) - 2\dot{\xi}(k)\dot{\eta}(k)\ddot{\eta}(k)}{s^4} \tag{5.135}$$

Looking at the equation (5.132) and the bottom row in the Jacobian (5.133) the expressions are not defined for speed equal to zero. This is because of how the coordinated turn is defined. To make the transformation valid even for the limiting case of zero speed the most straightforward solution is to assume

$$\omega(k) = 0 \tag{5.136}$$

and

$$A_{DT} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.137}$$

After the transformation of the state estimate covariance matrix of the DWNA or DWPA model

$$P_T(k|k) = A_{DT}P_D(k|k)A'_{DT} \tag{5.138}$$

the bottom row and the right most column will be all zeros. Then the element corresponding to the variance of angular velocity should be set to

$$P_{T_{56}}(k|k) = \sigma^2_{v_{\dot{\omega}}} \tag{5.139}$$

This approach will make the transformation from the DWNA or DWPA model to the nearly coordinated turn model cause the filter to reinitialize the state estimate of the angular velocity when the aircraft stops. As there is no possible solution on how to estimate turn rate for a standing aircraft around its own axis from only position measurements this approach seems as the most valid.

The extended Kalman filter can be also used for the linearisation of the measurement equation. Given that the measurements from the 3D WAM at the ATM Laboratory have the measurement matrix H equal to

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{5.140}$$

for the DWNA and DWPA model or

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{5.141}$$

for the nearly coordinated turn model there is no need to calculate the Jacobian (4.61). The linearisation is already done by the multilateration software. The measurements equations for all the models used for the designed tracker are thus linear.

5.3 IMM modifications

5.3.1 Introduction

The interacting multiple model estimator in its raw form as derived by H. Blom is a very effective tool for tracking of manoeuvring targets. In the literature numerous modifications specialized for different situations can be found. For the requirements of the designed tracker there are two possible modifications that come up to mind.

- Time varying Markov chain transition matrix
- Hard decision during state estimate and covariance combination.

The possibility of benefit from these modifications rises from the distribution of the measurement inter-arrival times the 3D WAM provides. This distribution as was shown in the figure 10 is approximately exponential.

During the derivation of the interacting multiple model it is assumed that the prior Markov chain transition is constant. This is reasonable for a source with a constant or a nearly constant refresh period. Given the distribution of the measurement inter-arrival times this assumption is not valid. A comparison of different techniques to find the most appropriate method for computing the Markov chain transition matrix will be done.

The state estimate and covariance combination step of the IMM algorithm computes the final state estimate that is passed as an output from the tracker. The mathematically most correct way following the initial assumption of (4.103) is to calculate a final state estimate as a Gaussian mixture. Another possible method is to use only the most probable mode as an output. This should decrease the influence of the modes the aircraft is clearly not in on the final state estimate. The benefit of the single mode preference should increase with larger refresh period. The overall influence with variable inter-arrival times should be evaluated. A Monte Carlo simulation to compare the hard decision method with the Gaussian mixture will be presented later.[17]

5.3.2 Algorithm comparison technique

To compare different algorithms a Monte Carlo simulations will be used. For the comparison of the results from the Monte Carlo simulations an optimal technique based on significance test of the sample mean of differences will be used.[4]

To evaluate a performance of an estimation algorithm j an expected value of a cost function is tried to be minimized.

$$J^{(j)} = E[C^{(j)}] \tag{5.142}$$

For testing in this thesis the cost function C will be squared position error

$$C = \tilde{\xi}^2 + \tilde{\eta}^2 \tag{5.143}$$

then the expected value of the cost function represents mean squared position error.

To compare two algorithms using Monte Carlo simulation it is beneficial to use the same random variables for each pair of runs. Using the same random variables for each pair of runs the cost functions of both algorithms are positively correlated. That is[4]

$$\text{corr}[C_i^{(1)}, C_i^{(2)}] > 0 \tag{5.144}$$

where i denotes the run i out of N and (1) and (2) denote the first and the second algorithm.

To test which algorithm is better the hypotheses are set to

$$H_0 : \qquad \Delta = J^{(2)} - J^{(1)} \leq 0 \tag{5.145}$$

$$H_1 : \qquad \Delta = J^{(2)} - J^{(1)} > 0 \tag{5.146}$$

The null hypothesis states that the first algorithm is not better that the second algorithm. The alternative hypothesis states that the first algorithm is better that the second.

The type I error probability is defined as

$$P\{H_1|H_0\} = \alpha \tag{5.147}$$

which represents the level of significance of the null hypothesis. Given the positive correlation of the cost functions the test whether the alternative hypothesis should be accepted is based on sample performance differences.[4]

$$\Delta_i = C_i^{(2)} - C_i^{(1)} \tag{5.148}$$

Given the sample mean of differences

$$\bar{\Delta} = \frac{1}{N}\sum_{i=1}^{N}\Delta_i \tag{5.149}$$

and its standard error

$$\sigma_{\bar{\Delta}} = \sqrt{\frac{1}{N^2}\sum_{i=1}^{N}(\Delta_i - \bar{\Delta})^2} \tag{5.150}$$

the alternative hypothesis H_1 is accepted if

$$\frac{\bar{\Delta}}{\sigma_{\bar{\Delta}}} > \kappa \tag{5.151}$$

where κ is the threshold obtained from the Gaussian distribution.

To accept the alternative hypothesis the test is whether the sample mean of differences (5.149) is positive and statistically significant. Because of this the threshold κ is based on the upper tail of the Gaussian probability density function.[4]

The advantage of the correlated cost functions for each pair of runs can be derived in the following way.

The sample mean of differences (5.149) can be rewritten to the form

$$\bar{\Delta} = \frac{1}{N}\sum_{i=1}^{N} \Delta_i \tag{5.152}$$

$$= \frac{1}{N}\sum_{i=1}^{N} C_i^{(2)} - \frac{1}{N}\sum_{i=1}^{N} C_i^{(1)} \tag{5.153}$$

$$= \bar{C}_i^{(2)} - \bar{C}_i^{(1)} \tag{5.154}$$

Variance of the sample mean of differences is then

$$\mathrm{var}[\bar{\Delta}] = \mathrm{var}[\bar{C}_i^{(2)} - \bar{C}_i^{(1)}] \tag{5.155}$$

$$= \mathrm{var}[\bar{C}_i^{(2)}] + \mathrm{var}[\bar{C}_i^{(1)}] - 2\mathrm{cov}[\bar{C}_i^{(2)}, \bar{C}_i^{(1)}] \tag{5.156}$$

$$= \mathrm{var}[\bar{C}_i^{(2)}] + \mathrm{var}[\bar{C}_i^{(1)}] - 2\sigma_{\bar{C}_i^{(2)}}\sigma_{\bar{C}_i^{(1)}}\mathrm{corr}[\bar{C}_i^{(2)}, \bar{C}_i^{(1)}] \tag{5.157}$$

The positive covariance obtained using the same random variables for each pair of runs decreases the final variance of the sample mean of differences which increases the validity of the test.

The sample performance differences in the form (5.148) represents a difference at time k. Thus even the significance test (5.151) is for a single time k. As it is more convenient to test if the algorithm is better for the whole tested trajectory or for some larger interval the sample performance difference will be substituted by[4]

$$\Delta_i(k,l) = \frac{1}{l-k+1}\sum_{m=k}^{l}\left(\bar{C}_i^{(2)}(m) - \bar{C}_i^{(1)}(m)\right) \tag{5.158}$$

5.3.3 Markov chain transition matrix

Typically in literature when designing the interacting multiple model it is assumed that the Markov chain transition probabilities (4.95) are constant. This is valid as the derivation of the dynamic multiple model estimator and the subsequent IMM estimator is done in discrete time. Even for the most common surveillance sensor in the form of a radar the assumption of a constant refresh period can be under a slight approximation considered valid. For application of the IMM estimator on the measurement obtained from the 3D WAM this assumption is not sufficient.

An optimal method for calculating the time dependent Markov chain transition probabilities is derived by assuming that the process describing the model transitions is a continuous-time homogeneous Markov chain.[41]

For a continuous-time Markov chain the Markov property is defined as[32]

$$P\{M(t+s) = j|M(s) = i, M(u) = l\} = P\{M(t+s) = j|M(s) = i\} \tag{5.159}$$

where $0 \leq u < s$. The time homogeneity is defined as an independence of the previous probability on s.

$$P\{M(t+s) = j|M(s) = i\} = P\{M(t-s) = j|M(0) = i\} \tag{5.160}$$

The transition probability function $P_{ij}(t)$ can be derived in the following way.

$$\begin{aligned}
P_{ij}(t) &= P\{M(t+s) = j|M(0) = i\} && (5.161)\\
&= \sum_{k=1}^{\infty} P\{M(t+s) = j, M(t) = k|M(0) = i\} && (5.162)\\
&= \sum_{k=1}^{\infty} P\{M(t+s) = j|M(t) = k, M(0) = i\}P\{M(t) = k|M(0) = i\} && (5.163)\\
&= \sum_{k=1}^{\infty} P\{M(t+s) = j|M(t) = k\}P\{M(t) = k|M(0) = i\} && (5.164)\\
&= \sum_{k=1}^{\infty} P_{kj}(s)P_{ik}(t) && (5.165)
\end{aligned}$$

The last equation above is also known as Chapman-Kolmogorov equation.[32]

To obtain the expression of the transition probability function we must derive Kolgomorov's backward equation as follows.

$$\begin{aligned}
P_{ij}(h+t) - P_{ij}(t) &= \sum_{k=1}^{\infty} P_{kj}(h)P_{ik}(t) - P_{ij}(t) && (5.166)\\
&= \sum_{k\neq i} P_{kj}(h)P_{ik}(t) - (1 - P_{ii}(h))P_{ij}(t) && (5.167)\\
& && (5.168)
\end{aligned}$$

Dividing the previous equation by h and taking a limit

$$\lim_{h\to 0} \frac{P_{ij}(h+t) - P_{ij}(t)}{h} = \lim_{h\to 0} \left(\sum_{k\neq i} \frac{P_{kj}(h)}{h} P_{ik}(t) - \frac{(1 - P_{ii}(h))}{h} P_{ij}(t) \right) \tag{5.169}$$

and using

$$\lim_{h\to 0} \frac{1 - P_{ii}(h)}{h} = v_i \tag{5.170}$$

$$\lim_{h\to 0} \frac{P_{ij}(h)}{h} = q_{ij} \tag{5.171}$$

where v_i is the rate at which the process makes a transition when in state i. q_{ij} are instantaneous transition rates into state j when in state i. Both can be proven using the assumption that the transition is exponentially distributed as is for the Markov chain because of its memoryless property.[32].

The Kolgomorov's backward equation is then

$$\dot{P}_{ij}(t) = \sum_{k\neq i} q_{ik}P_{kj}(t) - v_i P_{ij}(t) \tag{5.172}$$

or in matrix form

$$\dot{P}(t) = GP(t) \tag{5.173}$$

The $P(t)$ is matrix transition probability function with elements $P_{ij}(t)$. G is called infinitesimal generator and its elements are

$$G_{ij} = q_{ij} \quad \forall i \neq j \tag{5.174}$$

$$G_{ii} = -v_i \tag{5.175}$$

Given the boundary conditions

$$P_{ii}(0) = P(M(0) = i|M(0) = i) = 1 \tag{5.176}$$

$$P_{ij}(0) = P(M(0) = i|M(0) = j) = 0 \tag{5.177}$$

then the solution of the Kolmogorov's backward equation (5.173) is [42]

$$P(t) = e^{Gt} = \sum_{n=0}^{\infty} \frac{(Gt)^n}{n!} \tag{5.178}$$

This is the ideal method for computing the Markov transition matrix guiding the transition in the interacting multiple model estimator.

To obtain a numerical solution the infinitesimal generator matrix G is necessary. An practical way to obtain this matrix is to assume the elements on the main diagonal are[17]

$$G_{ii} = -v_i = -\frac{1}{\tau_i} \tag{5.179}$$

where τ_i is a mean sojourn time of mode i.

The mean sojourn times are more natural quantity to estimate and its values usually are around 25 seconds for the discrete white noise acceleration model, 5 seconds for the discrete Wiener process acceleration model and 15 seconds for the nearly coordinated turn model. These are of course tuning parameters of the IMM estimator.[4][17]

To obtain the off-diagonal elements a convenient way is to select forced Markov chain transition matrix Υ where each element on the main diagonal is set to zero i.e. zero probability that the state will transition to itself and the sum of each row equals to one.

The off-diagonal elements of the matrix G are then calculated as

$$G_{ij} = q_{ij} = -\Upsilon_{ij} G_{ii} \tag{5.180}$$

Since the matrix exponential is relatively demanding to compute an approximation to the ideal method (5.178) was proposed in [17] such that

$$P_{ii}(t) = \max\left\{l_i, 1 - \frac{t}{\tau_i}\right\} \tag{5.181}$$

where l_i is a minimum allowable probability. This value can be computed as a limit of the Markov chain transition matrix (5.178) since it converges to steady value as time approaches infinity.

The off-diagonal elements then can be calculated using the forced Markov chain transition matrix Υ. This approach will be later referred to as linear limited approach.

Another approach similar to the previous approximation is to approximate the diagonal elements as

$$P_{ii}(t) = \max\left\{ l_i, e^{-\dfrac{t}{\tau_i}} \right\} \tag{5.182}$$

where the expression is derived assuming the transition is a Poisson process with rate

$$\lambda = \frac{1}{\tau_i} \tag{5.183}$$

for the situation that the transition does not occur. That is

$$P_{ii}\{0\} = \frac{(\lambda t)^0}{0!} e^{-\lambda t} = e^{-\lambda t} = e^{-\dfrac{t}{\tau_i}} \tag{5.184}$$

This approach will be referred to as exponential limited.

To show the difference of the ideal method and the two mentioned approximation a development of the probability of transition from mode one to mode one i.e. element P_{11} of the Markov transition matrix can be seen in the figure 19.

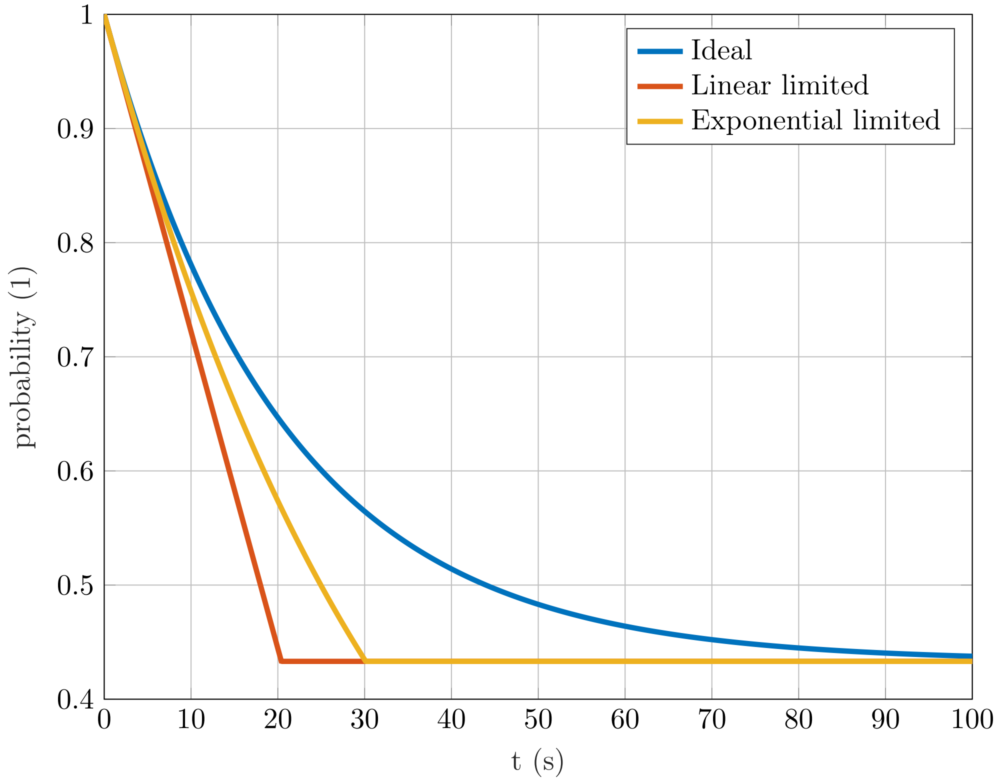

Figure 19: Probability of transition approximations

To compare which of the previously mentioned methods is better or if an approximation of the Markov chain transition matrix with constant values is sufficient enough even for the aperiodic sources as WAM a Monte Carlo simulation is done.

The simulation consist of a single scenario represented by a single aircraft trajectory[3]. The trajectory can be seen in the figure 20.

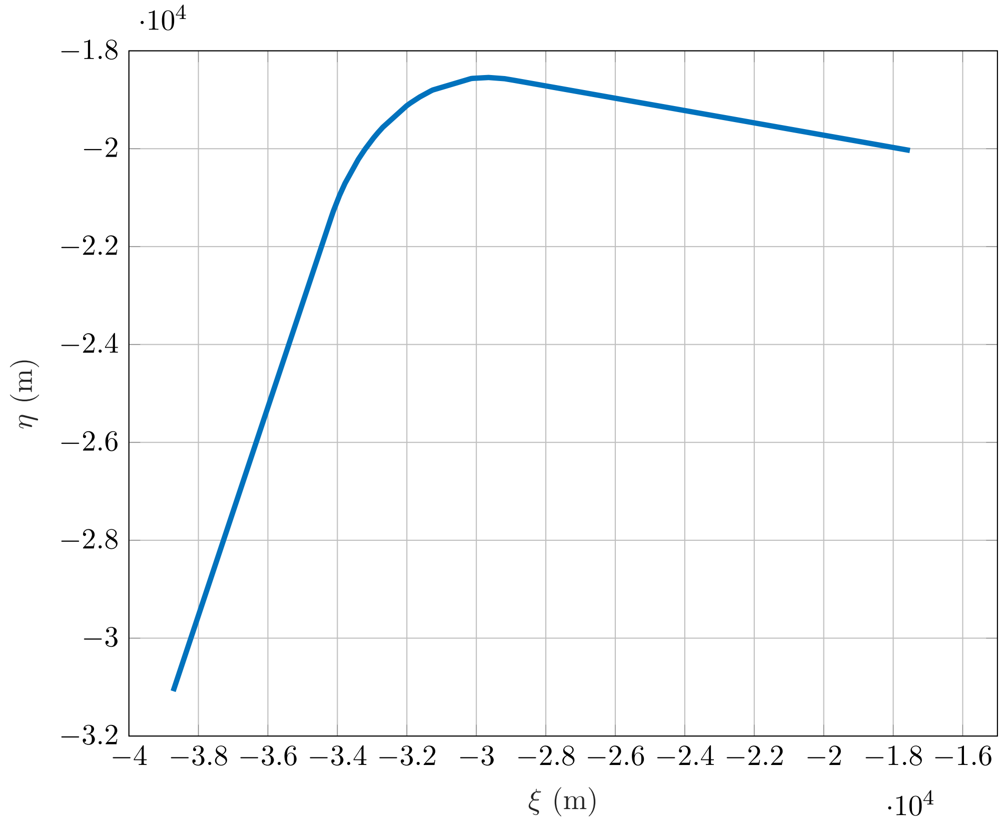

Figure 20: Aircraft trajectory for Monte Carlo simulation

At first the aircraft travels at a constant speed of $260\,\mathrm{ms}^{-1}$ for 42 seconds. After that for 24 seconds a $3\,^{\circ}\mathrm{s}^{-1}$ coordinated turn is performed. This corresponds to a normal acceleration of $13.61\,\mathrm{ms}^{-2}$. Next the aircraft flies with a constant velocity motion for 30 seconds. The final segment consist of 18 second decelerated motion with tangential acceleration of $-4\,\mathrm{ms}^{-2}$. The summary of the segments can be seen in the table 3.

Table 3: Simulated trajectory segments

Number	Duration (s)	Speed (ms^{-1})	a_t (ms^{-2})	a_n (ms^{-2})
1	42	260	0	0
2	24	260	0	-13.61
3	30	260	0	0
4	18	260-192.20	-4	0

To best evaluate the effect the method for the computing the Markov chain transition matrix

[3] The trajectory and its true position timestamps were generated by Track Generator software provided by ERA a.s.

has on a final state estimate an exponential distribution of measurement inter-arrival times was selected. This corresponds to the distribution of measurements obtained from the 3D WAM at the ATM Laboratory. For purposes of the designed tracker most interesting intensity of the exponential distribution λ is $7.69\,\mathrm{s}^{-1}$ as it is the measured value from the section 2.2.3. To properly evaluate which of the tested methods is better other intensities are also tested. All of them with their appropriate reciprocal values which are the mean inter-arrival times can be seen in the table 4.

Table 4: Tested intensities

λ (s^{-1})	μ (s)
0.50	2.00
1.00	1.00
2.00	0.50
4.00	0.25
7.69	0.13
10.00	0.10
20.00	0.05

For each intensity 100 different sets of measurement times were generated. Next typically a measurement is created as a noised true position. For the purposes of these test an additional noise to the measurement would only hide the impact of the different methods. The randomness brought by the distribution of the inter-arrival times is sufficient enough to evaluate the performance of the methods.

To compare the methods four different IMM estimators were created. Each corresponds to a single method of the Markov chain transition matrix computation. The inputs including the tuning constants of the estimators are same for all of them. The constant Markov transition matrix is computed using the ideal method (5.178). The time step for which it is computed is obtained using expected value of the used exponential distribution.

$$t = \mu = \frac{1}{\lambda} \tag{5.185}$$

The alternative hypotheses being tested using the method described in the section 5.3.2 are

- Exponential limited method is better than constant Markov chain transition.
- Linear limited method is better than constant Markov chain transition.
- Ideal method is better than constant Markov chain transition.

Hypotheses were evaluated using the modified equation (5.158) which allows for significance testing on an interval. Five different intervals were selected.

1. $t \in (0; 42\rangle$

2. $t \in (42; 66\rangle$
3. $t \in (66; 96\rangle$
4. $t \in (96; 114\rangle$
5. $t \in (0; 114\rangle$

The first interval corresponds only to a constant velocity motion. This will allow to evaluate the best method for non-manoeuvring targets. The second interval covers the coordinated turn. Focusing only on the turn manoeuvre will allow to evaluate the best approach during a constant manoeuvre but also during the transition from non-manoeuvring motion to a manoeuvre. This transition is a critical part in target tracking as the mode switching occurs. The third interval tests the opposite transition from the coordinated turn to a constant velocity motion. This also holds the same importance as the mode switching occurs. The fourth interval being tested is focusing on the linear decelerated motion. This motion, although also considered as a manoeuvre, differs from the coordinated motion mostly in the size of the acceleration. The total acceleration of the decelerated motion is approximately a third of the acceleration during the coordinated turn. This allows for testing the performance during subtle manoeuvres. The last interval on which the hypotheses are tested is done for the trajectory as a whole. This can be seen as an indicator if the benefits of certain methods in some situations outweighs the drawbacks in different situations.

The results of the simulation can be seen in the figure 21.

The first notable thing from the results is that the linear limited approach is for most of the cases the best choice. This might be indicating that the mathematically more ideal approach might be superseded by a more approximate just because its Markov chain transition matrix has a smaller diagonal. The reason for this can be seen in the figure 19. Given that how relatively short the inter-arrival times are for the tested cases it might be even reasonable to assume that a constant Markov chain transition matrix with a smaller diagonal might be better than a variable one.

Looking at a spread of the results compared to the tested intensities it can be seen that the difference between methods for higher intensities is negligible or is converging to a similar value. This might be caused because for higher intensities the computed Markov chain transition will be nearing identity.

Looking at the lower intensities for the first interval the spread of the different methods stays relatively constant. For the second through fourth interval there is a slightly more divergence between the methods.

When viewing the last segment it can be noted that on average all of the methods have have a very similar results for each of the tested intensities.

Considering the validation of the tested hypothesis the figure also depicts the κ threshold for the type I error probability of $\alpha = 0.05$. This a reasonably moderate criterion and to accept that some of the proposed methods for the Markov chain matrix computation are significantly better than the constant Markov chain transition this threshold should be exceeded.

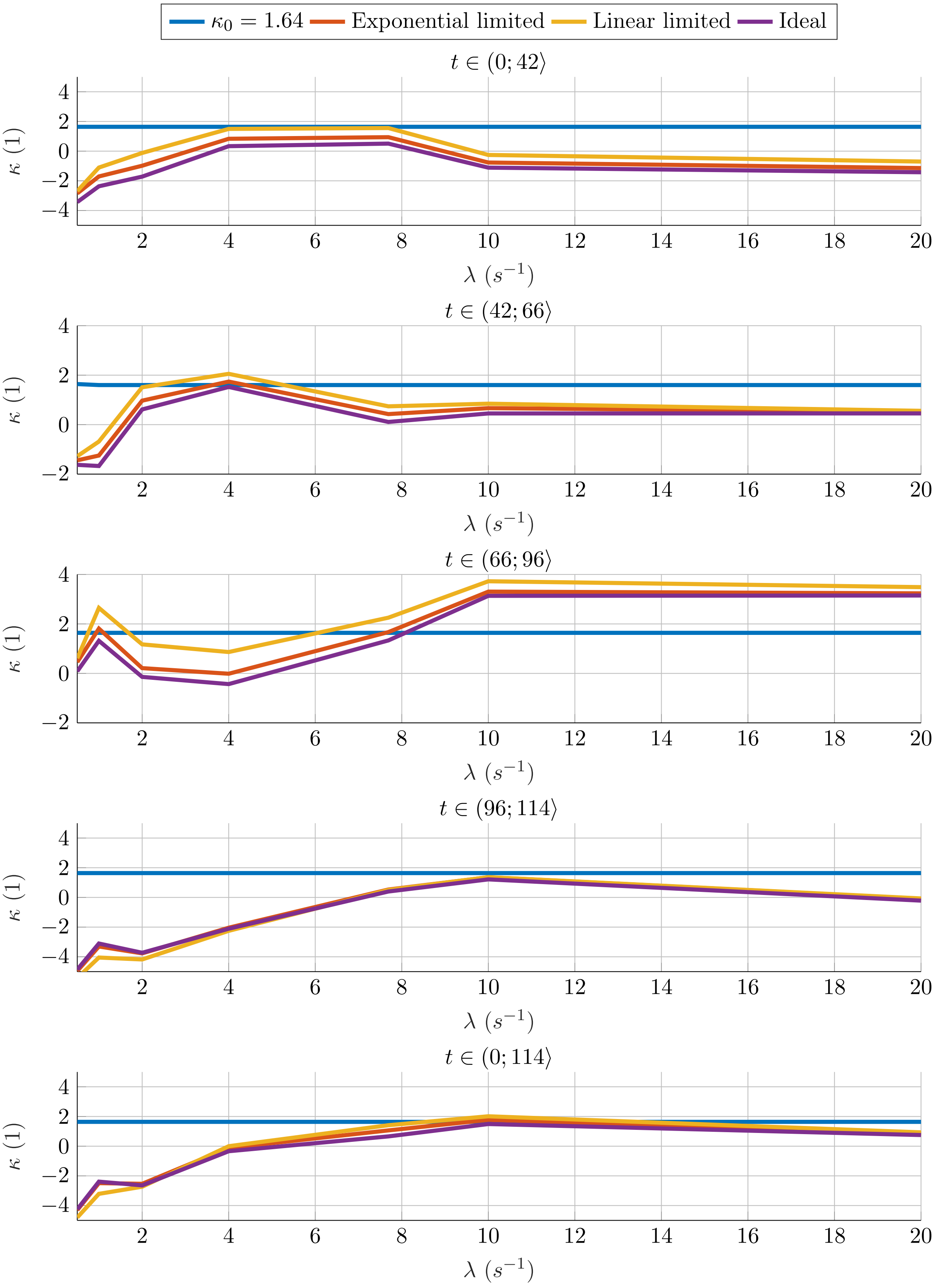

Figure 21: Markov transition matrix simulation results

From the results it can be seen that only for higher intensities during the third interval, i.e. transition from coordinated turn back to a constant velocity motion, the alternative hypotheses are accepted for all methods. Although when viewed separately this might indicate that the adaptive methods for computation of the Markov chain transition matrix might be better during manoeuvre transition this is not supported by the second or the fourth interval.

When looking at the last interval which averages the result across the whole simulated trajectory the only accepted hypotheses are for intensity of 10 plots per second for the linear limited and for the exponential limited method.

Viewing the overall picture some trend that would indicate that some of the proposed methods for the adaptive computation of the Markov chain transition matrix is significantly better is not present.

The significance test shows if the hypothesis should be accepted but does not show the performance increase an alternative method for computing the Markov chain transition matrix can bring. To examine this another Monte Carlo simulation was run. For this only single intensity of 10 messages per second was selected. This intensity was picked as there will be seen the biggest performance increase against the constant Markov chain transition matrix since it is the only one where on average two out of three hypotheses are accepted. For this intensity the same scenario with a single set of measurements was generated. Next a Monte Carlo simulation with 100 runs was performed. In each run the true measurement coordinates were independently augmented by a zero mean white Gaussian noise with standard deviation of 100 meters.

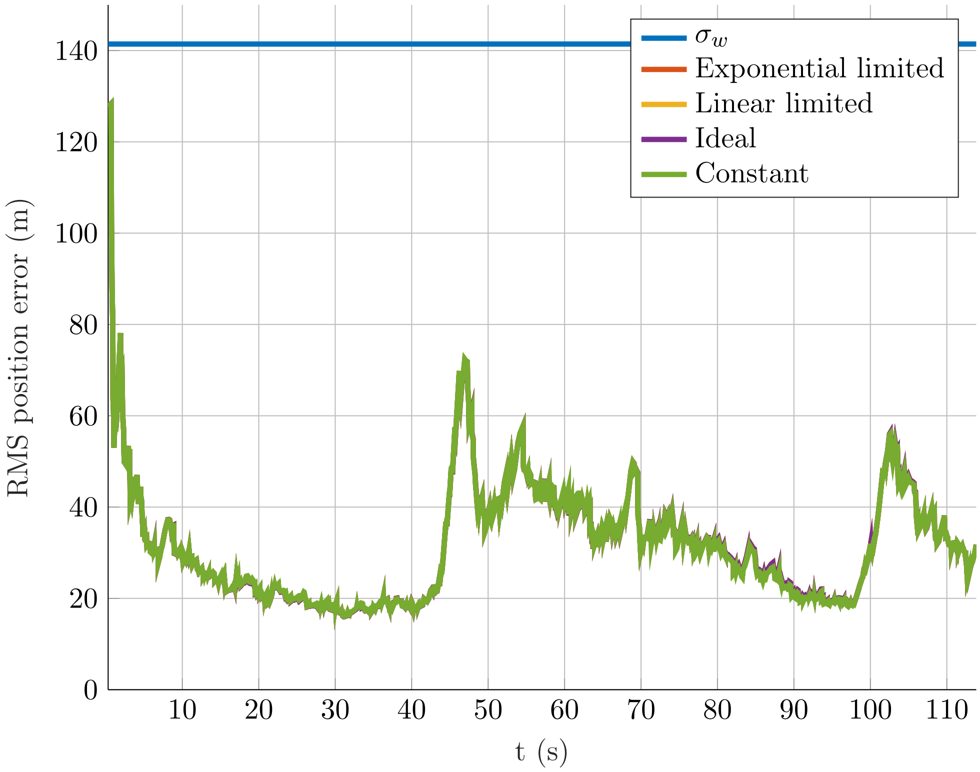

Figure 22: Markov transition matrix simulation results - RMS position error

When looking at the results shown in the figure 22 it can be seen that the performance increase from certain methods is clearly negligible. Given this results a preferred method for the computation of the Markov chain transition matrix should be the least computationally intensive. That is a constant matrix even for highly aperiodic sources such as multilateration. The only plausibility for using a different method which would increase performance by a significant margin might be for data fusion of couple sources with large refresh period such as some long range radars.

5.3.4 State estimate and covariance combination

The interacting multiple model estimator assigns each model with a certain probability that aircraft is in the corresponding mode. For output purposes from the estimator there is a necessity to provide single state estimate. Typically this is done using Gaussian mixture where states from the models are combined as a weighted average using its calculated probabilities (4.136). The final state estimate covariance matrix is also computed according to Gaussian mixture (4.137).

There are advantages and disadvantages to this approach. On one hand the Gaussian mixture weights all of the models thus strictly follows the rules of probability and the initial assumptions of the IMM estimator (4.103). On the other hand if one of the models is significantly preferred, that is it has a high probability, it might be assumed correct and the presence of other models only degrades the final state estimate.

A hard decision algorithm which would minimize the impact of the improbable models can be based on swapping the mode probabilities during the Gaussian mixture computation of state estimate and covariance combination. This can be formulated in the following way.

Assuming probability threshold $\upsilon \geq 0.5$ then $\forall j$

$$\mu_j(k) = \begin{cases} \mu_j(k) & \text{if } \mu_j(k) \leq \upsilon \ \ \forall j \\ \begin{cases} 1 & \text{if } \mu_j(k) > \upsilon \\ 0 & \text{otherwise} \end{cases} \end{cases} \tag{5.186}$$

To evaluate if there is a performance benefit when using the hard decision algorithm on a certain probability level a Monte Carlo simulation with the scenario in the figure 20. A single set of measurements with λ equal to 7.69 plots per seconds was generated. After that a 100 runs were performed where to each run a uncorrelated zero mean white Gaussian noise was added to the true position with standard deviation of 100 meters. The tested probability thresholds υ were selected to be from probability 0.5 up to a probability of 0.9 with a step of 0.1.

The average mode probabilities obtained from the simulation can be seen in the figure 23. These are probabilities before applying the hard decision algorithm. The root mean square position error can be seen in the figure 24. For ease of analysis RMS position error differences were also calculated and are presented in the figure 25. These differences are defined as a RMS position error obtained from the hard decision algorithm for some probability level minus RMS position error from the algorithm using Gaussian mixture. Negative value of these differences says that the hard decision algorithm performs better.

Looking at the results of the RMS position error and its difference two trends can be seen. First when the aircraft is moving with constant velocity motion from time 0 to 42 seconds and from about 78 to 96 seconds the hard decision algorithm performs better. Although the value

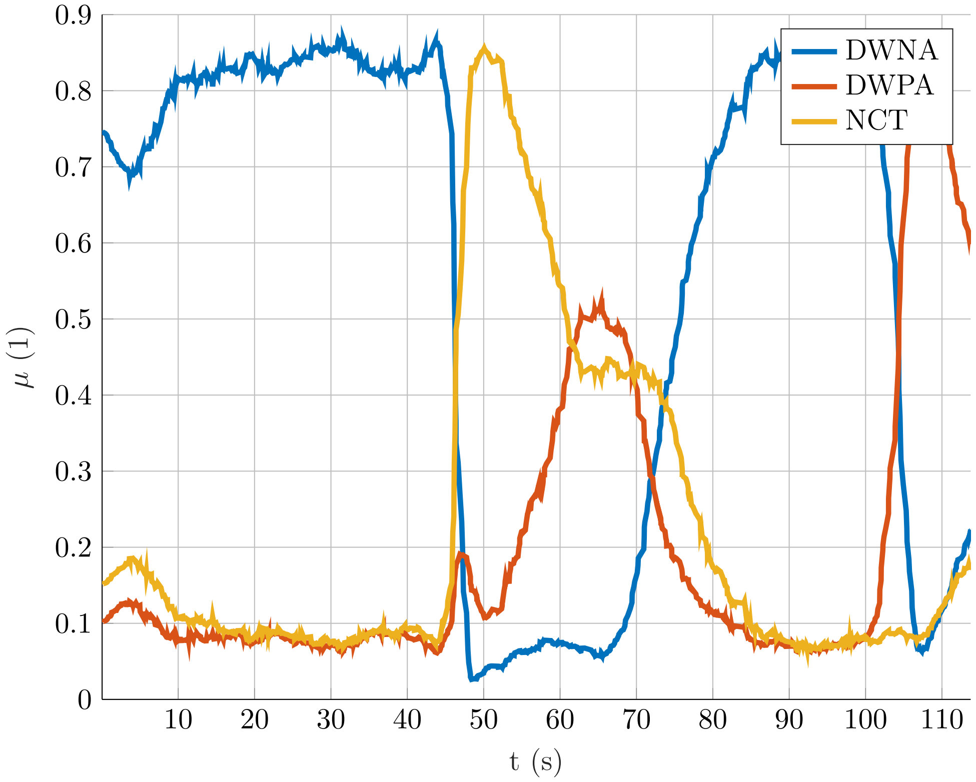

Figure 23: Estimate and covariance combination Monte Carlo - mode probabilities

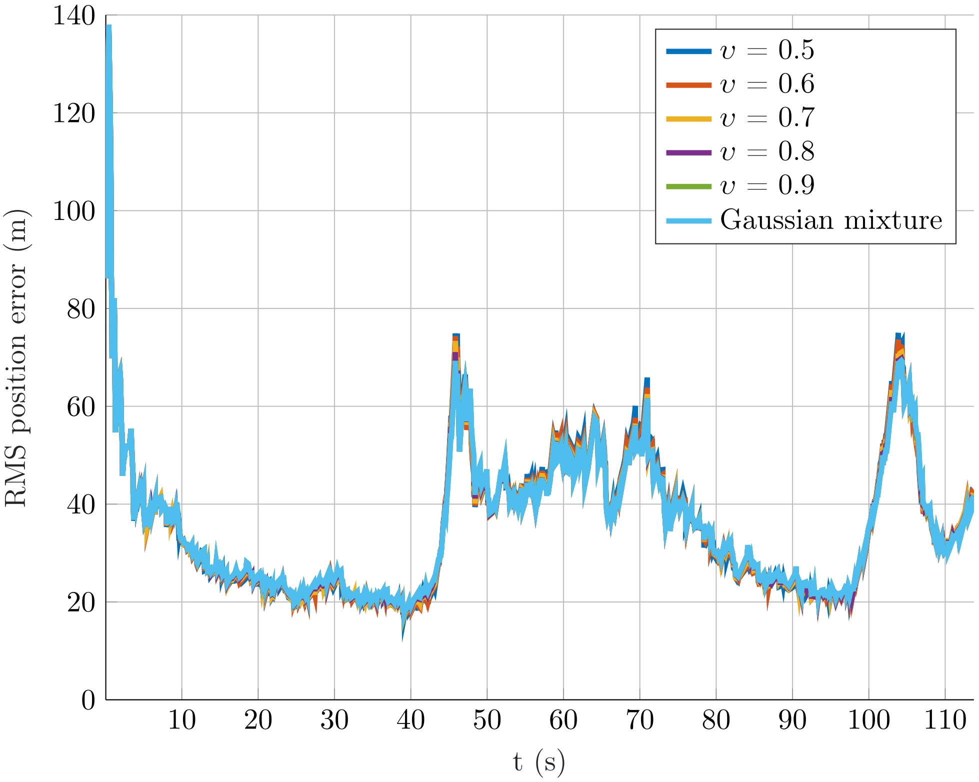

Figure 24: Estimate and covariance combination Monte Carlo - RMS position error

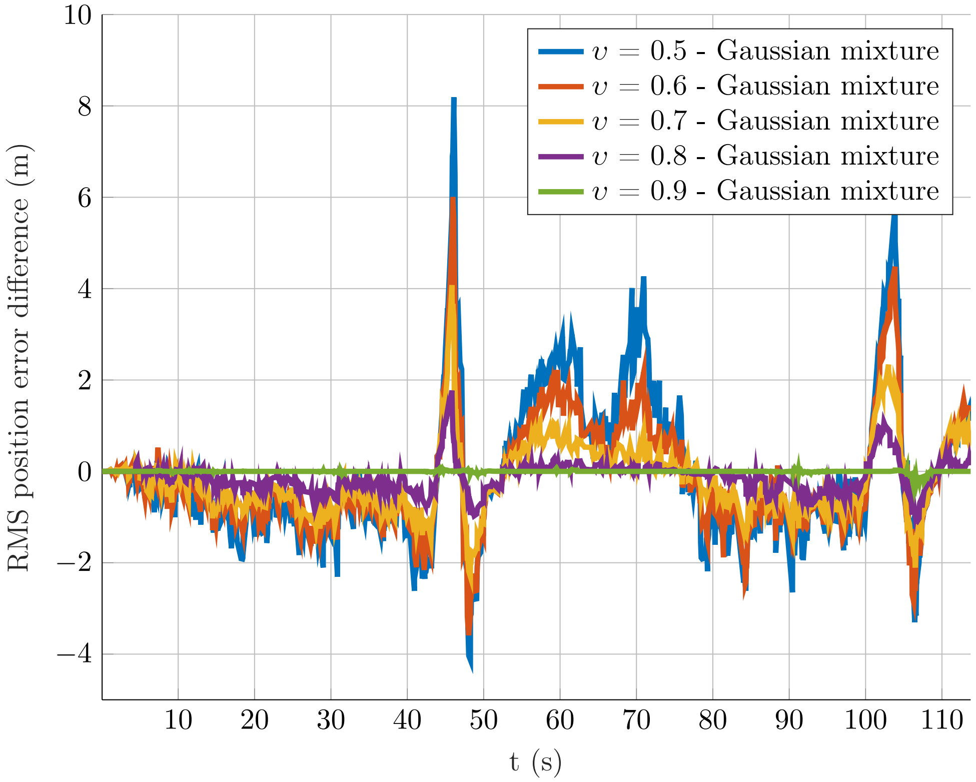

Figure 25: Estimate and covariance combination Monte Carlo - RMS position error difference

of the difference is not a significant when compared to the absolute value of the RMS position error it still increases the performance up to 10 % on these two intervals.

Second trend can be seen when the aircraft is transitioning between modes at time intervals of 42 to 50 seconds and 100 to 110 seconds. At first the Gaussian mixture algorithm performs better. This is because the hard decision algorithm tries to force the previous mode even when its probability starts to descent. On the other hand when the probability of the manoeuvre mode increases above the probability threshold the manoeuvre model obtains preference earlier and the performance of the hard decision algorithm gets above the Gaussian mixture. Sadly the initial peak error of the hard decision algorithm is larger then the subsequent one when the manoeuvre model gets its preference thus the positive contribution of the hard decision algorithm does not outweigh the negative one.

It is to be noted that since estimate and covariance combination happens only for output purposes the modified mode probabilities does not influence the mode transition of the IMM estimator. If they would influence it can be seen from the previously described trend that the mode transition would be delayed thus the performance of the estimator would decrease.

When looking at the intervals of the coordinated turn and of the decelerated motion the RMS position error of the hard decision algorithm is greater than for the Gaussian mixture. Looking at the mode probabilities for the coordinated turn this might be because after a while the influence of the discrete Wiener process accelerated motion model increases. An opposite situation occurs for the decelerated motion interval. It is reasonable to assume that this might be mitigated with further tuning of the IMM estimator and for different trajectories it might not be present. When the manoeuvre model would attain a high stable probability the performance increase should be the same as for the constant velocity motion.

Comparing the different probability threshold the trends are for all of them the same. Only difference is between the compromise of increase in performance for the constant velocity motion and decrease during the manoeuvre and manoeuvre transition.

Viewing the overall picture of the comparison between the hard decision algorithm and Gaussian mixture for the purposes of the designed tracker the slight increase in the performance for constant velocity motion is not outweighed by the decrease during manoeuvre transitions. For this reason the Gaussian mixture algorithm will be used. On the other hand the results show that for areas where there is sparse manoeuvring as oceanic areas the hard decision algorithm might provide convenient performance increase.

6 Track maintenance

6.1 Practical implementation

As was briefly presented in the section 3.3 the track maintenance is part of the tracking process which solves three main tasks:

- Track creation
- Track confirmation
- Track deletion

Given that the developed tracker is currently being designed to only use measurements from the 3D WAM at the ATM Laboratory and that currently only Mode S messages are being processed the use of ICAO 24 bit address is possible.

The ICAO 24 bit address is an unique identifier presented in all Mode S replies transmitted by the aircraft transponder. It is mandatory by the legislation incorporated from the Annex 10 that the ICAO 24 bit address has to be unique world wide. The only exception is for vehicles at airports where the uniqueness has to be maintained within range of 1000 kilometres. Looking at the picture 1-4 the coverage of the WAM at the ATM Laboratory is significantly less than 1000 kilometres. Thus the use ICAO 24 bit address to ease the track maintenance and the subsequent measurement to track association is possible.[43]

The first part of the track maintenance is the track creation. This task is easily solved with the use of ICAO 24 bit address. When a new address which is not in a memory of active tracks is received with a measurement a new track is created. With only a single measurements a valid track can be created without even considering measurement to track association.

The track deletion is typically done using M-N logic where if M measurements out of N possible measurements are not validated the track is deleted. This simple logic is useful for sources as radars where the refresh period is set but for the MLT which is inherently data driven this cannot be done. The most straightforward way how to solve this is to analyse what is the typical mean inter-arrival time and set a time period after which if the track is not updated it is deleted. Given assumed exponential distribution for inter-arrival times a value corresponding to 99.999 percentile. For the $\lambda = 7.69\,\mathrm{s}^{-1}$ which corresponds to mean inter-arrival time of 0.13 seconds the 99.999 percentile is 1.4971 seconds. Given the short time interval a user of the software would be barely capable to notice if some spurious track are being created. Although this value is statistically sufficient assuming the estimated λ is correct for most of the aircraft in the area of coverage for the testing and analysis this value will be set higher to for example 10 seconds.

Track confirmation can also be done using M-N logic where if M measurements are correlated out of expected N measurements the track is confirmed. Using the ICAO 24 bit address the confirmation can be done after a single measurements due to its uniqueness. When doing preliminary testing of the tracker it was seen that sometimes a measurement with invalid aircraft address was received. This is easily recognizable as there will be a track created by only a single measurement at a position which would associate with already existing one and no other measurement will be received for this ICAO 24 bit address. To mitigate this track confirmation will be done when receiving a second measurement.

7 Measurement to track association

7.1 Design considerations

After the track creation in the track maintenance function the measurement to track association function must be carefully designed. When designing this segment of target tracking process the main consideration is about what source will be providing the measurements and what information the measurements hold. The 3D WAM at the ATM Laboratory provides

- Horizontal coordinates ξ, η
- Vertical coordinate in form of elliptical height ζ
- ICAO 24 bit address

When comparing this data against sources such as primary surveillance radar or multilateration using mode A/C replies this is actually very generous. Most of the problematic connected with measurement to track association evolves around finding which measurement should be assigned to which track for the subsequent filtering and prediction. For the designed tracker in this thesis this could be fully solved by the provided unique ICAO 24 bit address.

Given the principle of 3D multilateration which in most cases provides two measurements for a single target the measurement to track association function of the designed tracker must solve four situations.

1. Assigning one measurement to one track
2. Assigning one measurement to two tracks
3. Assigning two measurements to one track
4. Assigning two measurements to two tracks

7.2 Gating

Gating is a part of the measurement to track association whose main purpose is to reduce computational complexity. This is typically done by creating so called gates i.e. regions around a track to which if the measurement falls it is considered during data association. As was mentioned in the section 6.1 for the purposes of the designed tracker it can be assumed that the ICAO 24 bit address is an unique identifier. With this there is no need to consider other methods for gating as the unique address ensures that at most two tracks will be considered for a single measurement.

7.3 Data association

The data association function of the designed tracker will be used to solve the previously mentioned four situations that can occur during the target tracking process.

The first step of data association is to check whether the elliptical height is positive. As was seen during testing this in most cases solves the duality of the measurement and thus significantly reduces the complexity of data association.

7.3.1 One measurement one track

When assigning a single measurement to a single track the easiest and also the most reliable method given the provided informations is to use the unique ICAO 24 bit address. This will reduce all computational complexity to a single hash table search and couple logical operations. Only risk presented with this method is if the ICAO 24 bit address would be invalid due to message reception interference. Although this is a usual situation the 3D WAM requires four identical messages each from a different receiver to calculate the position. It is highly unlikely that an interference would occur at all four receivers and would cause the same data corruption.

7.3.2 One measurement two tracks

To find out which track should be associated with a single received measurement the most straightforward approach is to assign the measurement to the nearest one. The nearest has multiple definitions in data associations. There is geometrical distance, statistical distance, likelihood or even probability obtained from PDA or JPDA algorithms. The complexity of computation significantly increases in the order the definitions were mentioned. For the designed tracker given that we will want to find solution at most for two tracks and two measurements the statistical distance in form of Mahalanobis distance should be sufficient compromise between computational and implementational complexity and the performance.

Given already initialized track and a received measurement Mahalanobis distance is defined as[17]

$$d = \sqrt{\tilde{z}(k+1|k)S(k+1)^{-1}\tilde{z}(k+1|k)'} \tag{7.1}$$

where $\tilde{z}(k+1|k)$ is the innovation (4.22) and $S(k+1)$ is the measurement prediction or innovation covariance matrix (4.21).

The advantage of the Mahalanobis distance over simply using the geometrical distance is the consideration of covariance of the data. Instead of measuring a geometrical circular distance for two dimensions it measures distance inside a confidence ellipse when the data are correlated.

After receiving a single measurement the Mahalanobis distance to both maintained tracks will be calculated and the track with lower one will be assigned the measurement.

Although Mahalanobis distance is a general expression that can be used for n dimensional measurements for the purpose of the designed tracker the distance will be calculated only using the elliptical height. This will reduce the matrix computation to a scalar computation in one dimension and thus allow for a slight reduction of computational complexity.

The only problem with using Mahalanobis distance is the necessity to have an already initialized track. Given that it was decided to use the two-point differencing initialization method when a second measurement is received the measurement prediction covariance $S(k+1)$ is undefined. This is because it is derived using state estimate covariance matrix and its prediction (4.19). To solve this the most straightforward approach is to assume that the time between measurements is negligible and that the track has the covariance of the first measurement. Then the Mahalanobis distance for the association when receiving second measurement is

$$d = \sqrt{\tilde{z}(k+1|k)\left(R(k)+R(k+1)\right)^{-1}\tilde{z}(k+1|k)'} \tag{7.2}$$

where $R(k)$ and $R(k+1)$ is the measurement noise covariance matrix at time k and $k+1$ respectively. The summation of the measurement noise covariances follows from its assumed whiteness.

This approximation is reasonable from multiple point of views.

1. The inter-arrival times are very short with mean value of $\mu = 0.13\,\mathrm{s}$. In this time the target won't move very far.

2. Given that only position measurements are being received only the covariance matrix is being predicted. The covariance matrix increases only by the dynamics of the model which is defined by time. The time difference is the same for both of the track to which a measurement is being associated thus the increase in covariance is same for both. Given that only selection of a smaller Mahalanobis distance is being made and not a comparison against a statistical threshold the fact that both of the distances should be bigger or smaller is insignificant.

7.3.3 Two measurements one track

Assigning two measurements to a single track the same principle is applied as in the previous section. The Mahalanobis distance is computed and the measurement with smaller one is assigned to the track.

7.3.4 Two measurements two tracks

When assigning two measurements to two tracks i.e. which measurement should continue which track if both measurements have positive elliptical height the similar principle as previously is applied.

First the Mahalanobis distance is computed for all four of the combinations. Then by using greedy algorithm the measurement track-pair with smallest distance is selected as the associated one. Then the remaining three combinations reduce to one as one track and one measurement were assigned.

8 Software representation

8.1 Overview

For the purposes of the ATM Laboratory the software representation of the previously derived and designed tracker is divided into two parts.

- Matlab
- C++

8.2 Matlab

The Matlab representation of the designed tracker was developed for two reasons. First as Matlab is a scripting language it is less time consuming to work with when adjusting for specific needs of different tasks. For this reason the Matlab was selected as a suitable language for testing the hypotheses, doing Monte Carlo simulations and for obtaining preliminary results.

The second reason is to allow the laboratory to work with a simpler language for certain tasks where real time performance is not necessary. This will allow to speed up research and preliminary testing without the need to maintain relatively complex C++ code.

The Matlab code used for the design of the tracker is created by four components seen in the figure 26.

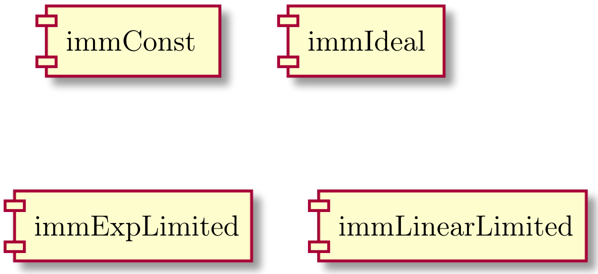

Figure 26: Matlab software components

Each component represents a single interacting multiple model estimator for horizontal motion model with a different method for computing the Markov Chain transition matrix. Although it was presented earlier that the constant Markov Chain transition matrix seems sufficient for real world application, for research purposes all of the versions were kept. In the future it will allow the laboratory to further investigate the influence a time dependent Markov Chain transition matrix cloud have.

The components are represented by functions whose input consists mainly of a matrix of measurements, its standard deviation and tuning parameters of the estimators such as standard deviations of the process noises and other required constants.

The output from the components is same for each IMM estimator. It has the form of all of the states after each measurement

$$\begin{bmatrix} \hat{x}_1(0) & \cdots & \hat{x}_6(0) \\ \vdots & & \vdots \\ \hat{x}_1(k) & \cdots & \hat{x}_6(k) \end{bmatrix} \tag{8.1}$$

its covariance matrices

$$\begin{bmatrix} P_{11}(0) & \cdots & P_{16}(k) & P_{21}(k) & \cdots & P_{66}(0) \\ \vdots & & \vdots & \vdots & & \vdots \\ P_{11}(k) & \cdots & P_{16}(k) & P_{21}(k) & \cdots & P_{66}(k) \end{bmatrix} \tag{8.2}$$

and the computed mode probabilities

$$\begin{bmatrix} \mu_1(0) & \cdots & \mu_3(0) \\ \vdots & & \vdots \\ \mu_1(k) & \cdots & \mu_3(k) \end{bmatrix} \tag{8.3}$$

Although the IMM estimator for the vertical motion which was designed in the previous sections is not explicitly specified in the provided code the estimators for the horizontal motion have greater research usability. If necessary the vertical one is easily created from the horizontal ones by disregarding the nearly coordinated turn model.

8.3 C++

The software representation for the real time processing was written in the C++ programming language. This language was picked for its minimal overhead, objective-oriented nature and large support in open-source community.

First necessary decision was if the designed tracker should have its own thread of execution or if is should be run from the outside. The first option is more suitable in situations where there is little to zero modification of the code required and the tracker will be used as a single component. The second option allows for easier modifications of the designed tracker and does not restrict the user of the code to follow set principles. Since the laboratory will use the tracker mainly for research purposes the second solution was selected.

The high level structure of the designed tracker can be seen in the figure 27.

The Tracker class represents the full designed tracker. It provides the main API for the user. The API consists of three public functions

- `addMeasurement` (x2)
- `checkTrackDeletion`

and two callback functions

- `trackInnovationCb`
- `trackDeletionCb`

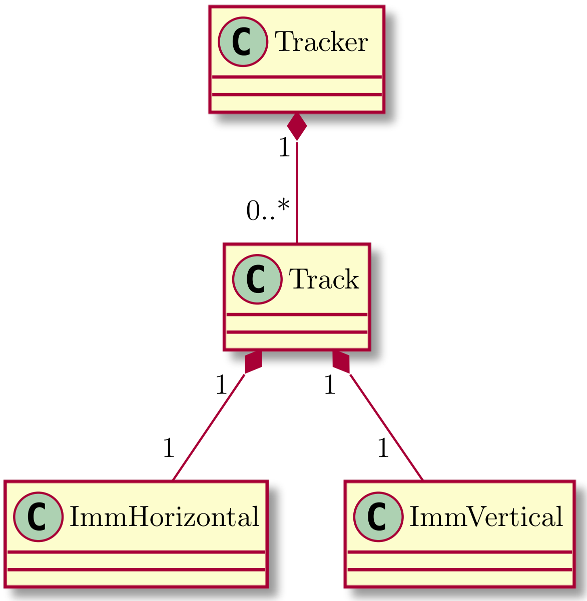

Figure 27: Tracker C++ software structure

which are set through the constructor.

The `addMeasurement` function allow to pass the measurement to the tracker. This function is overloaded due to the MLT providing up to two positions for a single measurement. The `checkTrackDeletion` function is an event trigger which performs a check across all currently tracked tracks if some should be deleted.

The callback function `trackInnovationCb` passes the updated states after innovating from received measurements to the responsible entity. The callback `trackDeletionCb` is called when a track is deleted. It passes a track number which is an identifier unique across all currently maintained tracks.

The Tracker class then contains memory of all the tracks represented by the Track class. Each track contains the horizontal and vertical interacting multiple model estimator which are used to calculate the state estimate of the tracked aircraft.

Current implementation of the tracker in the 3D WAM software at the ATM Laboratory is in a form of dynamically linked library. This is not a necessity and the code can be either statically linked or even included in the project directly. The only requirement for the project compilation is the Eigen[4] library.

[4] Eigen is an open-source library for linear algebra. It is available from eigen.tuxfamily.org

9 Practical application performance evaluation

9.1 Scenario summary

To evaluate the performance of the designed tracker real data obtained from the 3D WAM at the Laboratory were used. A time interval on 4th February 2019 between 12:20 and 13:00 UTC was selected. This is the same interval as was previously analysed in the section 2.2.3. This represents midday interval where a relatively moderate to high traffic is expected.

Due to the high impact the dilution of precision caused by the configuration of the aerials has on the measurement precision the area of coverage was limited to area depicted in the figure 28.

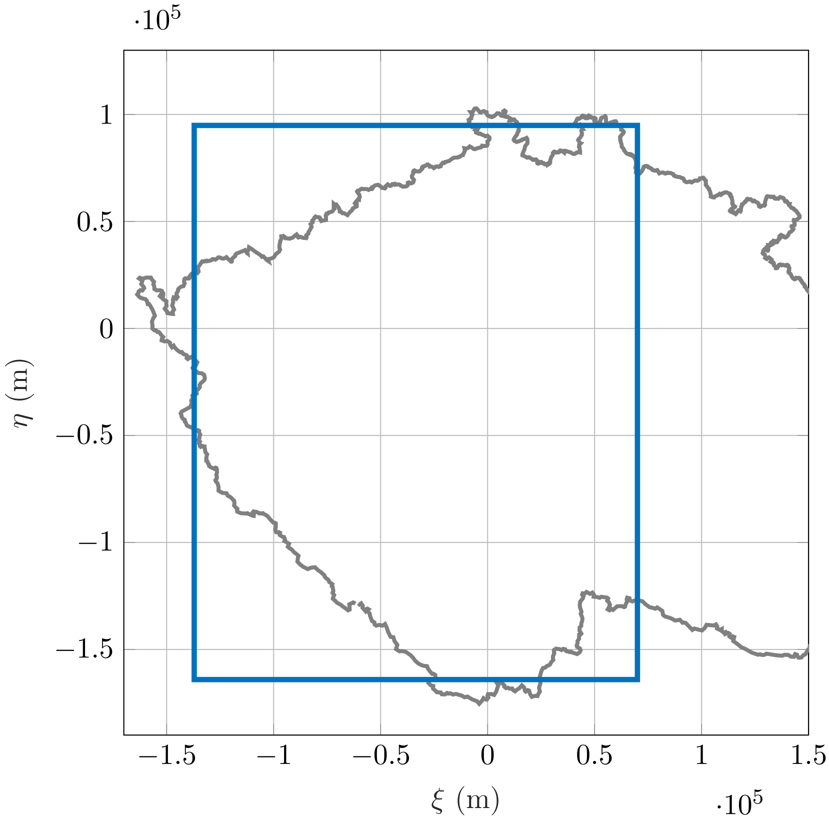

Figure 28: Surveillance area limits

Although this area is still very broad and the measurement errors can reach up to couple dozens kilometres it reduces the impact the extreme cases at the very edge of coverage make.

After implementing the C++ tracker into the 3D WAM software segment the previously mentioned recording was run. During the recording total of 972 unique ICAO 24 bit addresses were present and the corresponding aircraft tracked. The peak number of tracked aircraft at the same time reached 41.

The tracking was tested on moderate hardware at the laboratory and on couple other different computers. Each time CPU load was below 20 % during the whole recording. Comparison of CPU load during the run without the tracker implemented and after implementing can be seen in the figure 29. This comparison was done on a laptop with Intel Core i7-4702MQ processor with base frequency of 2.20 GHz.

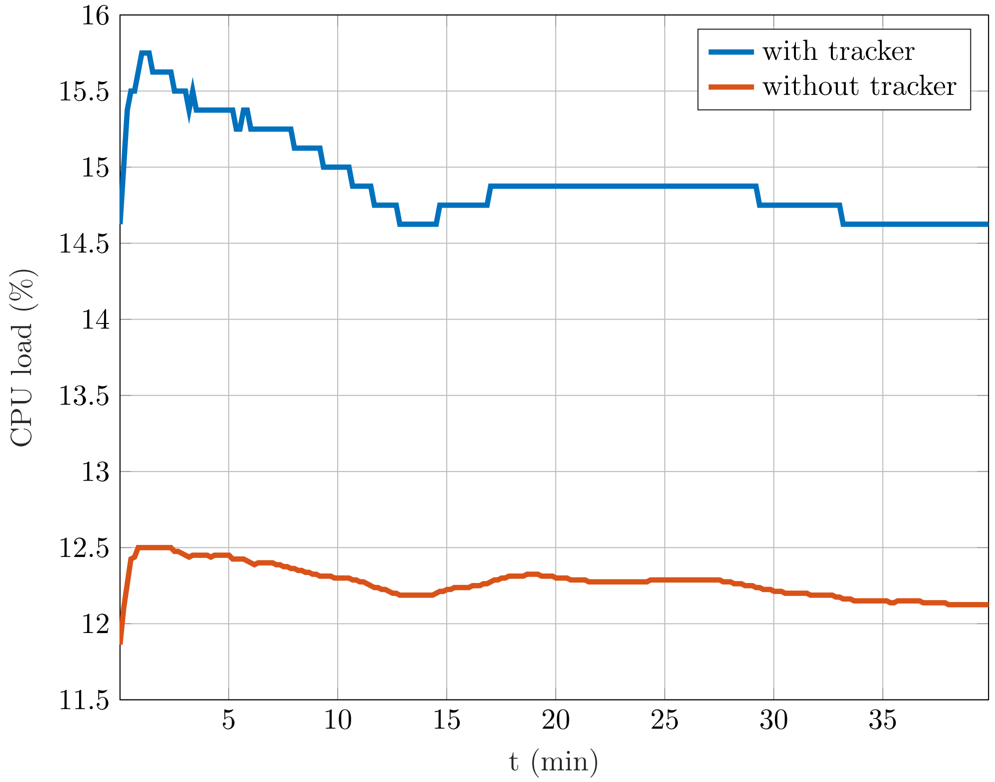

Figure 29: CPU load increase

It can be seen that use of the tracker has about 3 % impact on the performance. For the ATM Laboratory it provides sufficient computational reserve for further improvements and extensions.

Performance analyses focused on track duality resolution and filtering are presented next.

9.2 Track duality resolution

The main problem the measurement to track association function of the designed tracker should solve is track duality. It is caused by multiple subsequent dual measurements provided by the multilateration and produces two independent tracks of the same target with the same ICAO 24 bit address. To solve this an approach was proposed in section 7.3 and implemented in the tracker software.

First step of the resolution is to compare if the elliptical height of a measurement is positive. During the whole recording a total of 204 350 measurements were passed from the MLT to the tracker. Out of these only 1966 measurements were not able to be resolved using the height comparison. This stands for only 0.96 %. It might seem that given the small percentage the problem would be almost solved. This is not the case because the distribution of the unresolved dual measurements is position dependent. Most of the unresolved dual measurements were found to be located between the aerials.

To further evaluate how the subsequent data association algorithm performs in track duality resolution the time until the track duality was resolved was recorded and also a number of measurements it took.

Throughout the recording only 28 tracks were created as dual tracks. For these tracks a histogram of the time it took the association algorithm to resolve which of these tracks is correct can be seen in the figure 30. A histogram of a number of measurements which were required until the track duality was resolved can be seen in the figure 31.

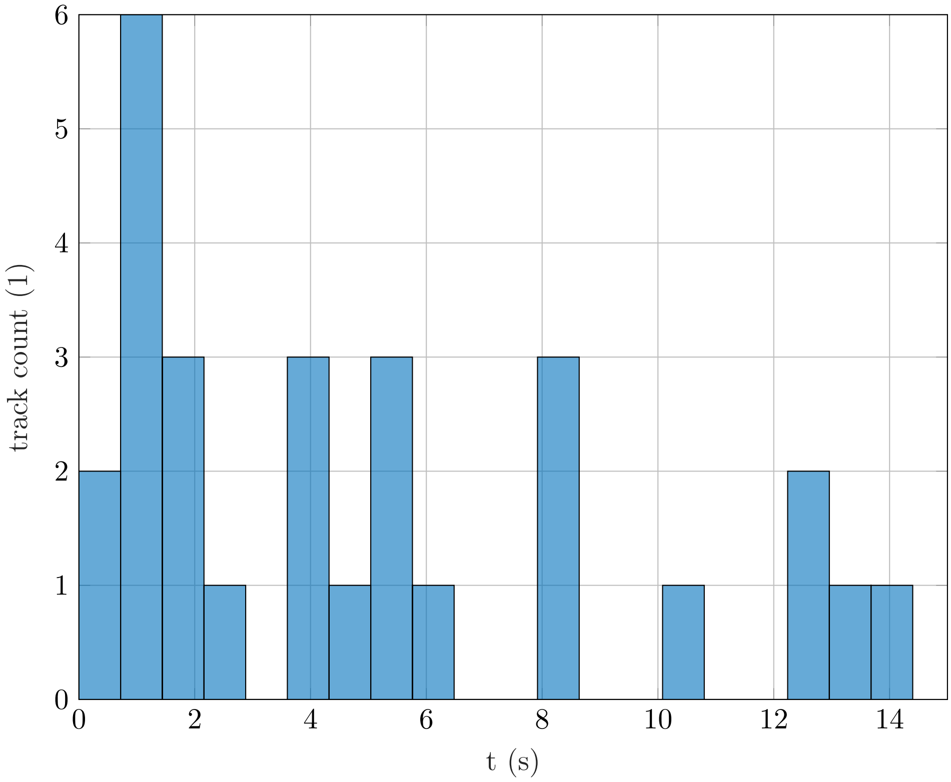

Figure 30: Time until track duality resolution

Looking at the histogram of the time before plot duality resolution there is a peak at about one second but otherwise the it is relatively uniformly distributed with a maximum time of about fourteen seconds. Given the durations this would be unacceptable. Looking at the histogram of number of messages it can be clearly seen that most of the track duality resolution occurred after receiving a second or a third measurement. This means that even though the duration before the resolution can get unacceptable long most of the time the duality would be resolved before presenting the track to the user.

The table 5 presents all of the times and the corresponding number of measurements for all of the 28 tracks. In it a few critical combinations are highlighted. For professional deployment this is unacceptable but for the application at the ATM Laboratory to resolve the duality every time is not required. These results might even significantly change if the geometrical configuration would be modified and it might not occur at all. In the future the configuration will have to be changed as currently for the WAM purposes its accuracy is not within a reasonable range.

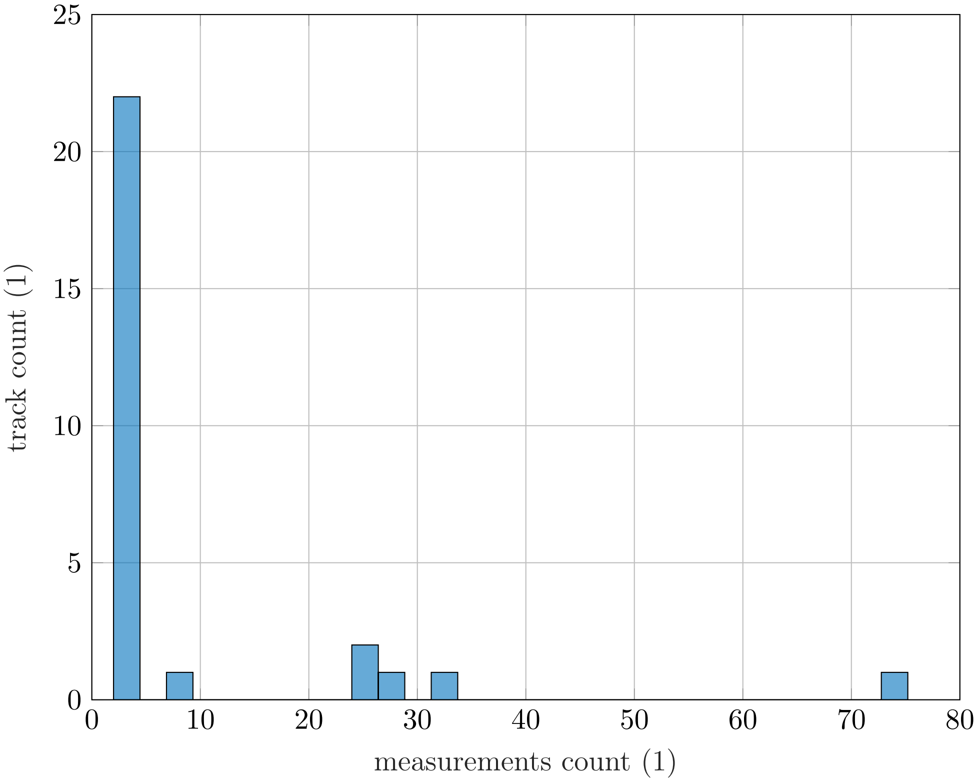

Figure 31: Number of measurements until track duality resolution

Table 5: Time and measurement count until track duality resolution

t (s)	count (1)	t (s)	count (1)
0.39	3	0.85	2
2.15	3	1.86	2
4.22	3	7.93	25
2.87	2	0.62	3
4.87	2	1.03	3
0.91	2	12.00	75
10.24	28	8.11	2
0.87	2	1.02	2
7.97	2	1.13	2
12.44	32	5.45	2
12.25	4	14.21	26
1.45	2	5.46	2
5.87	7	5.47	2
3.77	4	3.82	2

9.3 Filtering performance

To evaluate the filtering performance for each track the output from each track innovation was recorded. After going through the results two tracks were selected and are presented here.

The trajectory of the first selected track relative to the aerials position can be seen in the figure 32. The estimated trajectory corresponds to a straight line motion. The estimated speed can be seen in the figure 33. The graph on the top shows the estimated speed throughout the whole trajectory. It can be seen that at first given the large inaccuracy of the geometrical configuration and time stamping of the low-cost receivers the speed is initialized with an extremely high value for a civilian target. When disregarding first few minutes the bottom graph shows a stabilized estimate. Although there are still large deviations given the initial estimated value and the accuracy of the measurements it is a relatively good estimate.

The figure 34 shows the estimate of the acceleration magnitude. Although there are some spikes most likely caused by the measurement accuracy most of the time it is near zero value. The figure 35 shows the progress of the mode probabilities. It shows a high probability of a constant velocity motion with influence of the constant accelerated motion probably caused by the measurement accuracy. The probability of the coordinated turn is negligible.

The figure 36 shows the estimated height. The first few minutes are most likely caused by the large measurement error and correspond to the error in the horizontal plane for the velocity. During this time the aircraft was furthest away from the aerials. In the figure 37 the vertical speed is visible. Same as for the horizontal situation the error is large at the beginning but after zooming later the velocity is relatively stable and indicates descent. This is supported by progress of the height. The acceleration and the mode probabilities for vertical motion can be seen in the figure 38 and 39 respectively.

The estimated trajectory of the second selected track can be seen in the figure 40 and is probably of a smaller aircraft during training or sightseeing. This track was selected as there are multiple manoeuvres which could better show the mode transition.

In the figure 41 can be seen the estimated speed the aircraft. Certain number of larger deviations can be seen during the manoeuvres but overall the value corresponds to a smaller aircraft. The graph of the acceleration magnitude can be seen in the figure 42. The mode probabilities can be seen in the figure 43. From its progress the mode switching is noticeable. Although from the trajectory it can be seen that the aircraft performs turns the influence of the nearly coordinated turn model is negligible. This is probably caused because the discrete Wiener process acceleration model can also be used to model manoeuvres such as turns and in combination with the very large measurement errors the possibility of effectively estimating correlated manoeuvres is decreasing. With further tuning it might be possible to boost the nearly coordinated turn model for this track but when tuning a tracker it must be possible to efficiently track all types of aircraft in its surveillance area which brings necessary compromises to the estimation process. In the tracking literature this is typically presented as a problem of how to track military and civilian aircraft most efficiently. For the tracker at the laboratory this problem consist of a necessity to find compromise between tracking near the aerials and at the edge of the surveillance area where the measurement error is in the range of tens of kilometres.

Because of this the discrete Wiener process acceleration model is used to cover large deviations due to the measurement accuracy and thus is more relaxed and preferred even during turns. This should be improved when the aerial configuration is modified.

The corresponding graphs for the height, velocity, acceleration and mode probabilities can be seen in the figures 44,45,46 and 47.

It is to be noted that in both cases the shown trajectory is not the full trajectory of the aircraft while being tracked by the tracker. Both times the trajectory was split into multiple segments due to the track deletion process. This was done even for the ten seconds time limit before track deletion. This is caused by the insufficient coverage of the airspace by the aerials and should not be present with better aerial configuration.

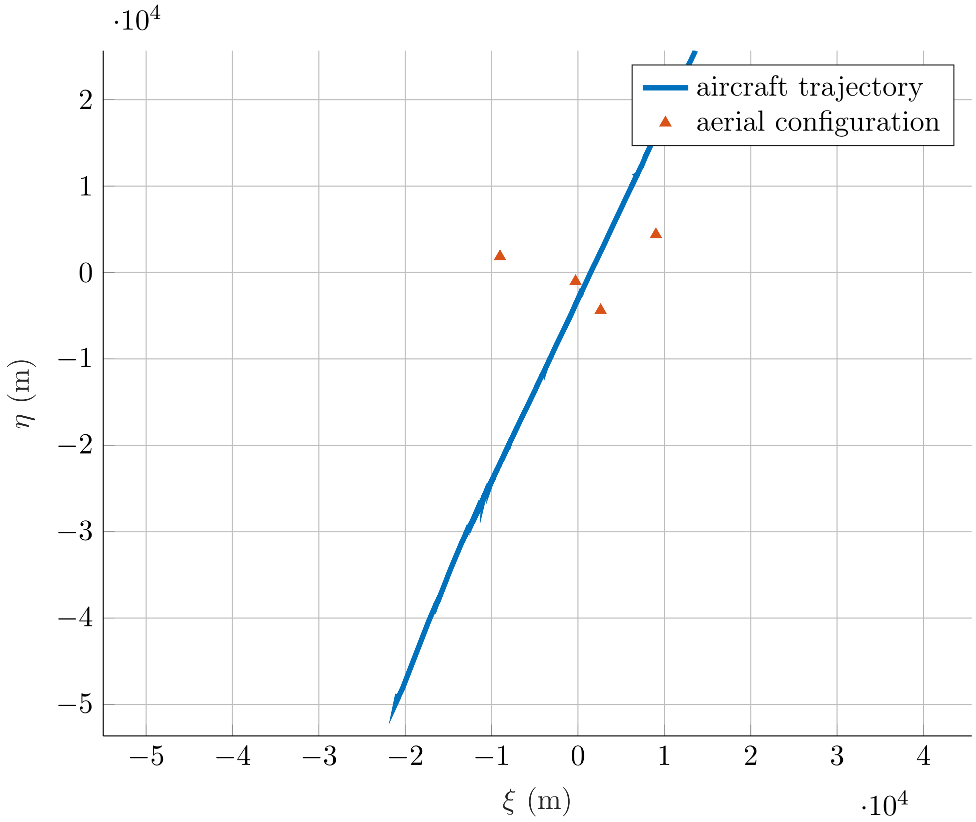

Figure 32: Track 1 - estimated trajectory

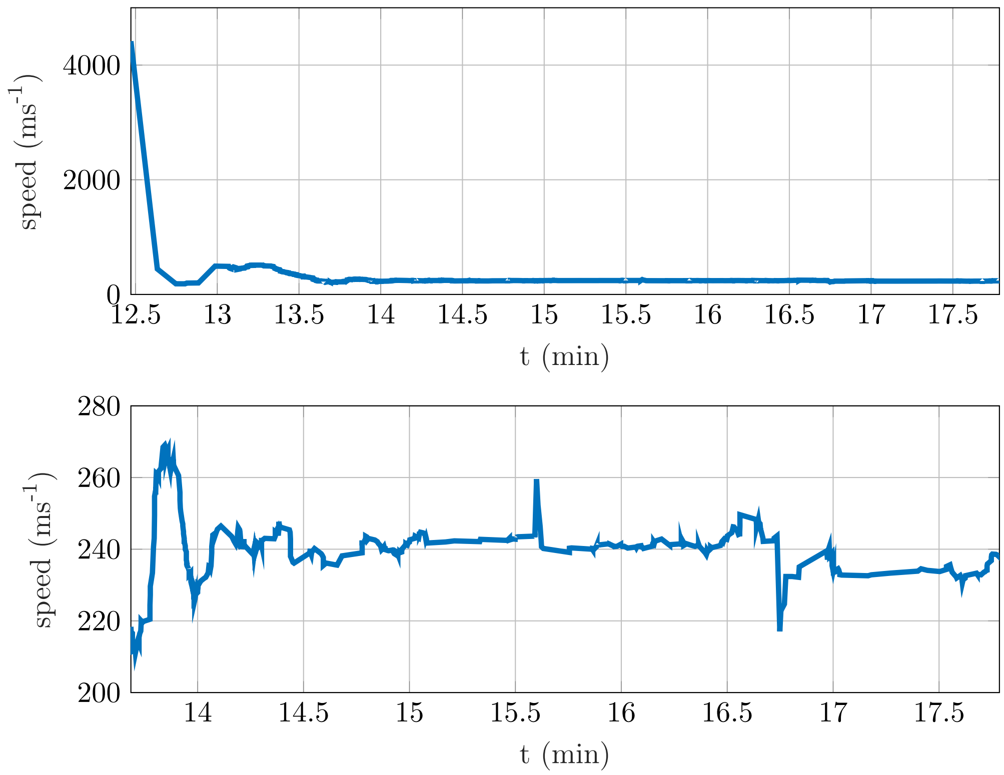

Figure 33: Track 1 - horizontal speed

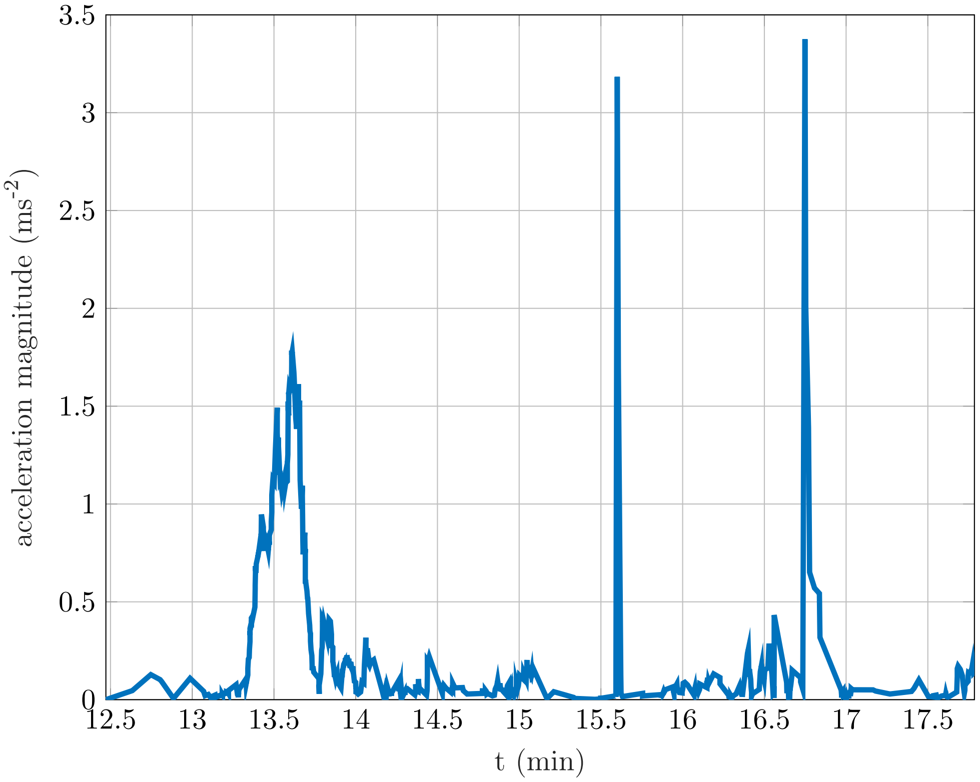

Figure 34: Track 1 - horizontal acceleration magnitude

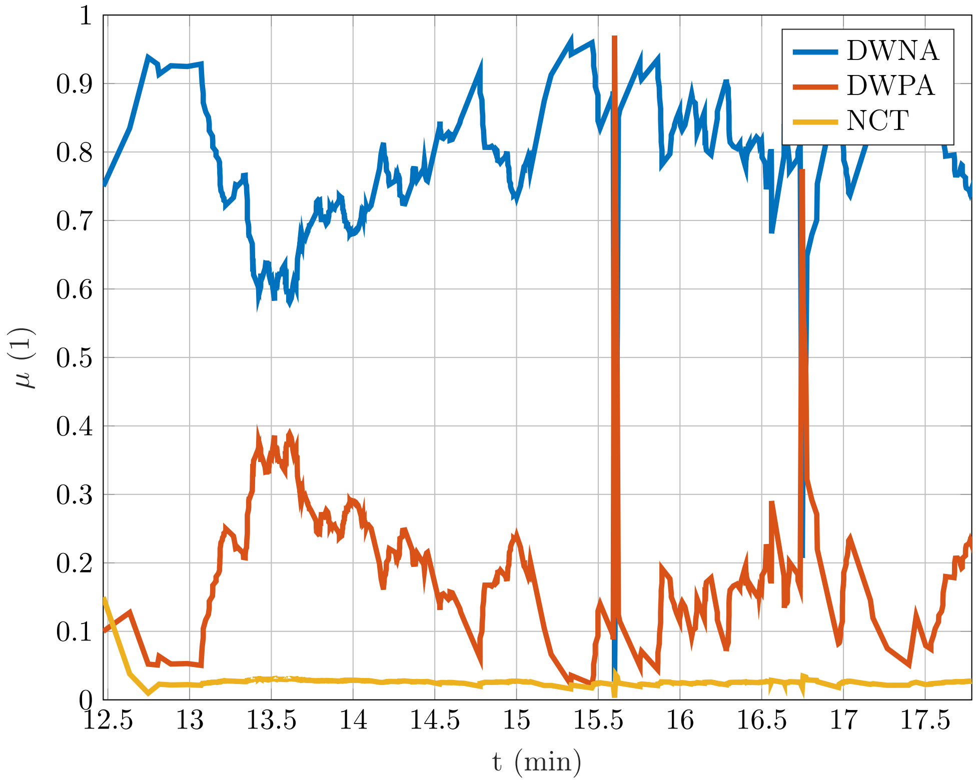

Figure 35: Track 1 - horizontal mode probabilities

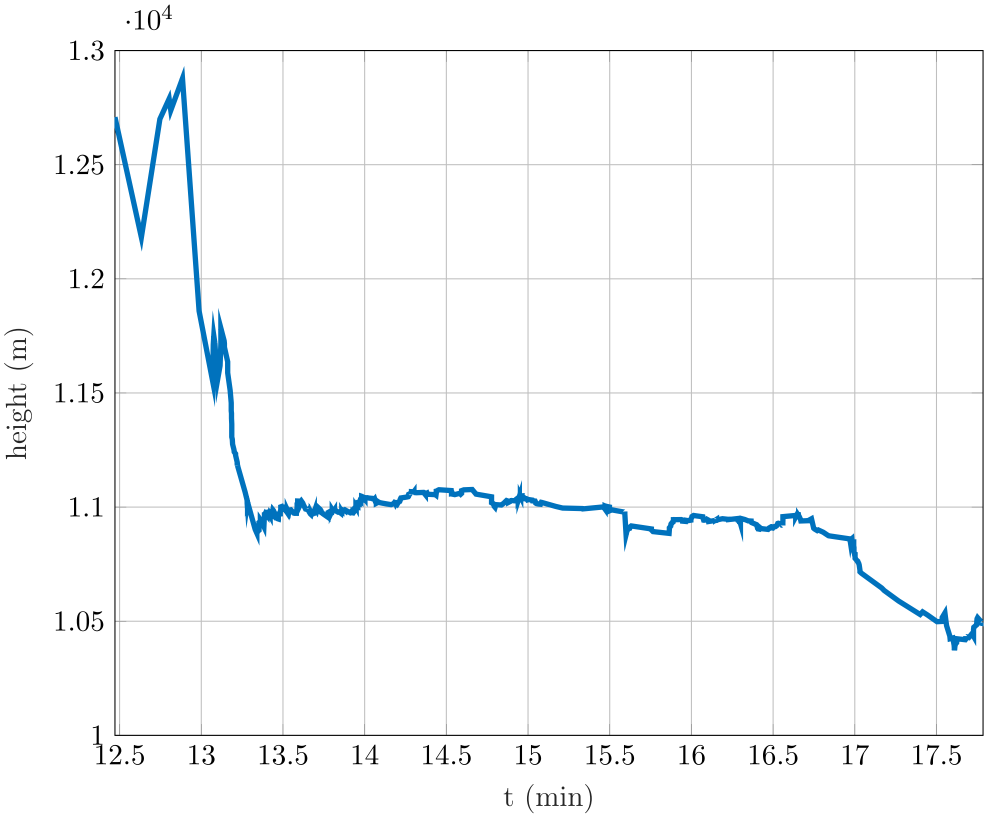

Figure 36: Track 1 - height

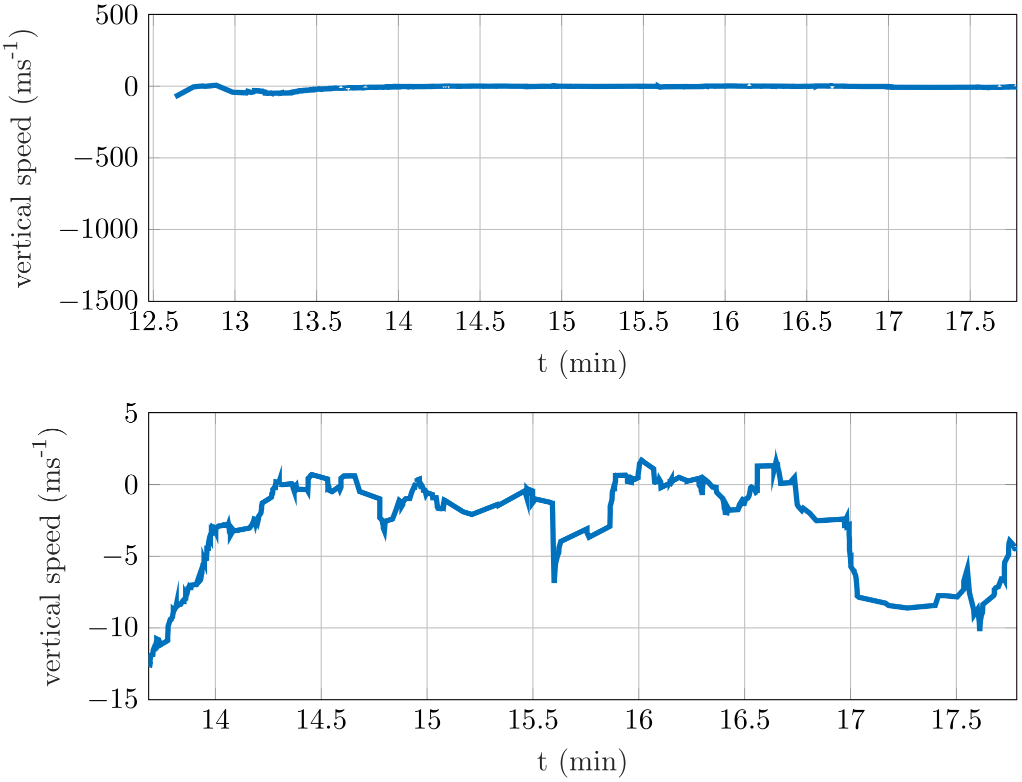

Figure 37: Track 1 - vertical speed

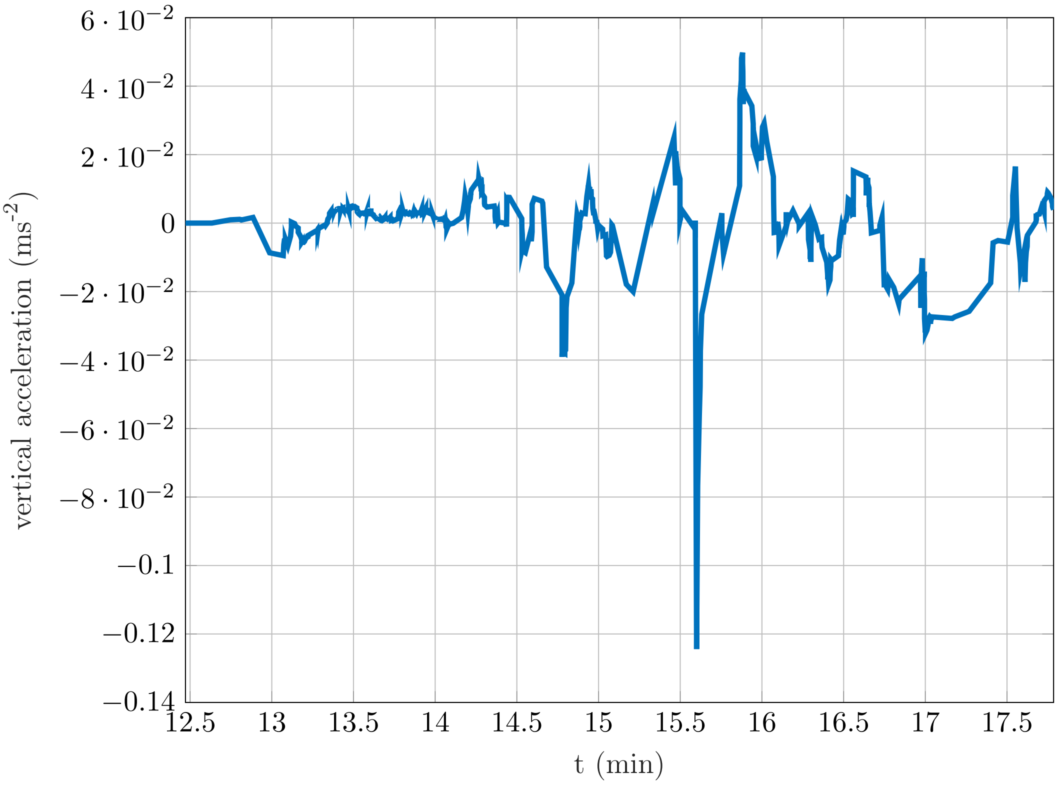

Figure 38: Track 1 - vertical acceleration

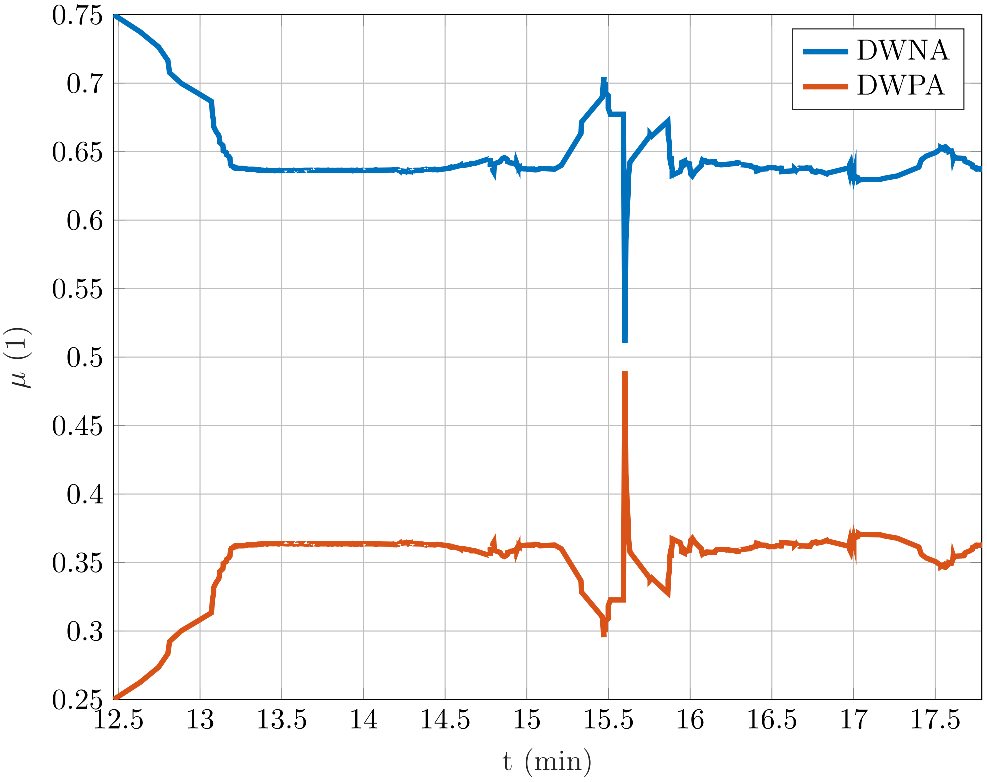

Figure 39: Track 1 - vertical mode probabilities

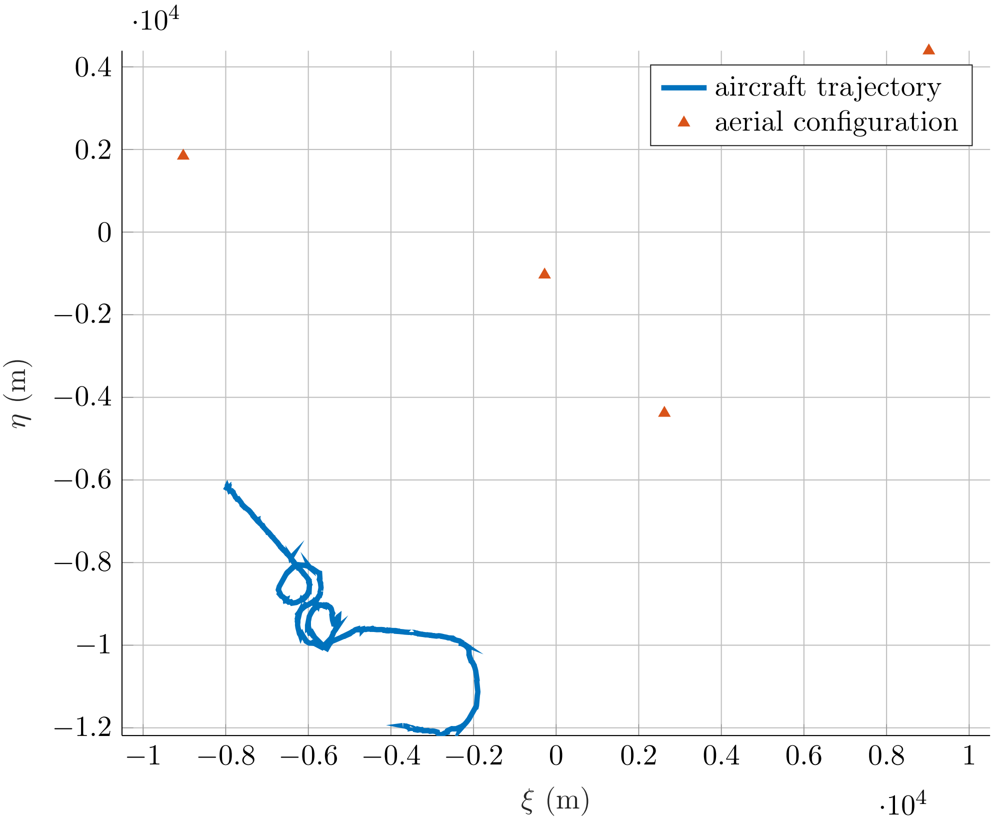

Figure 40: Track 2 - estimated trajectory

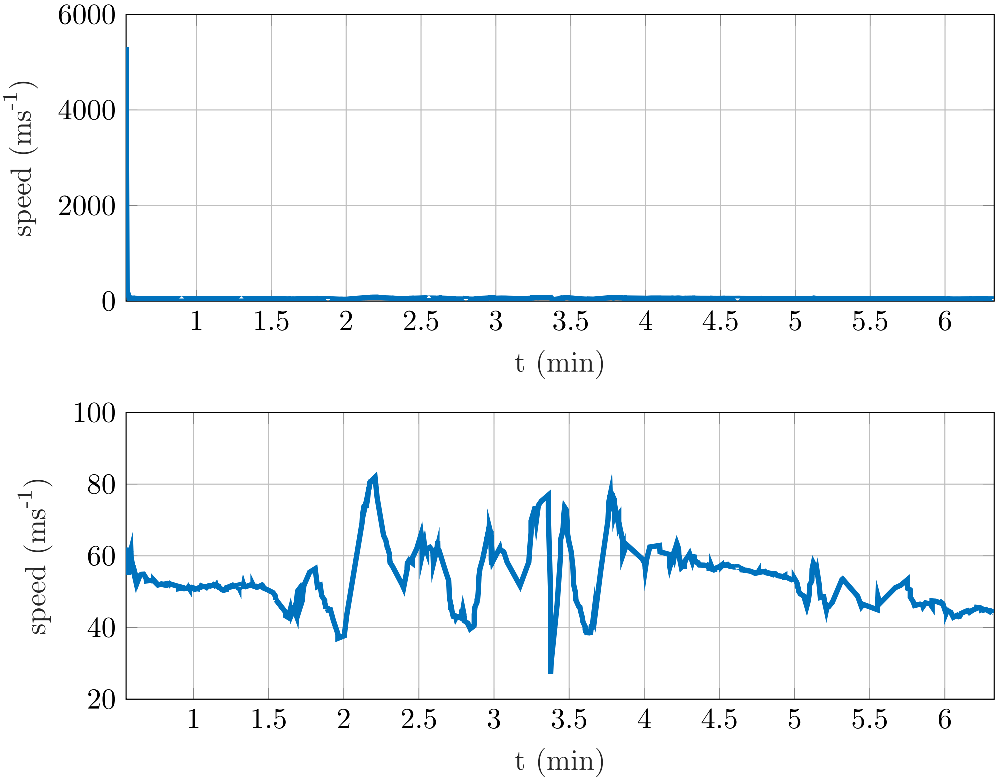

Figure 41: Track 2 - horizontal speed

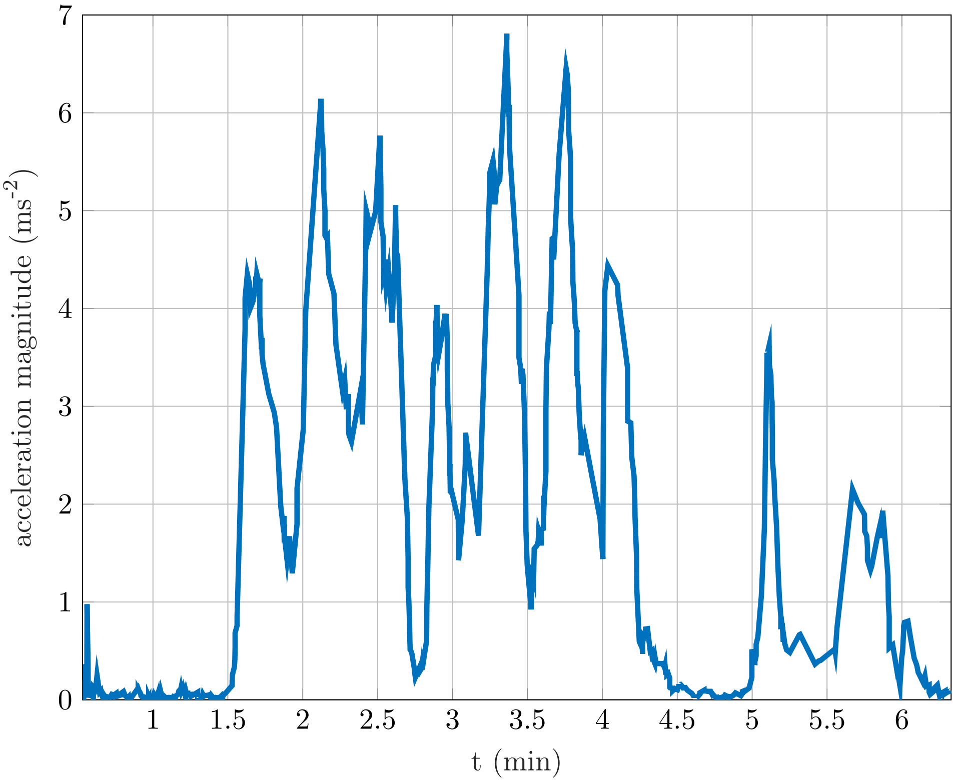

Figure 42: Track 2 - horizontal acceleration magnitude

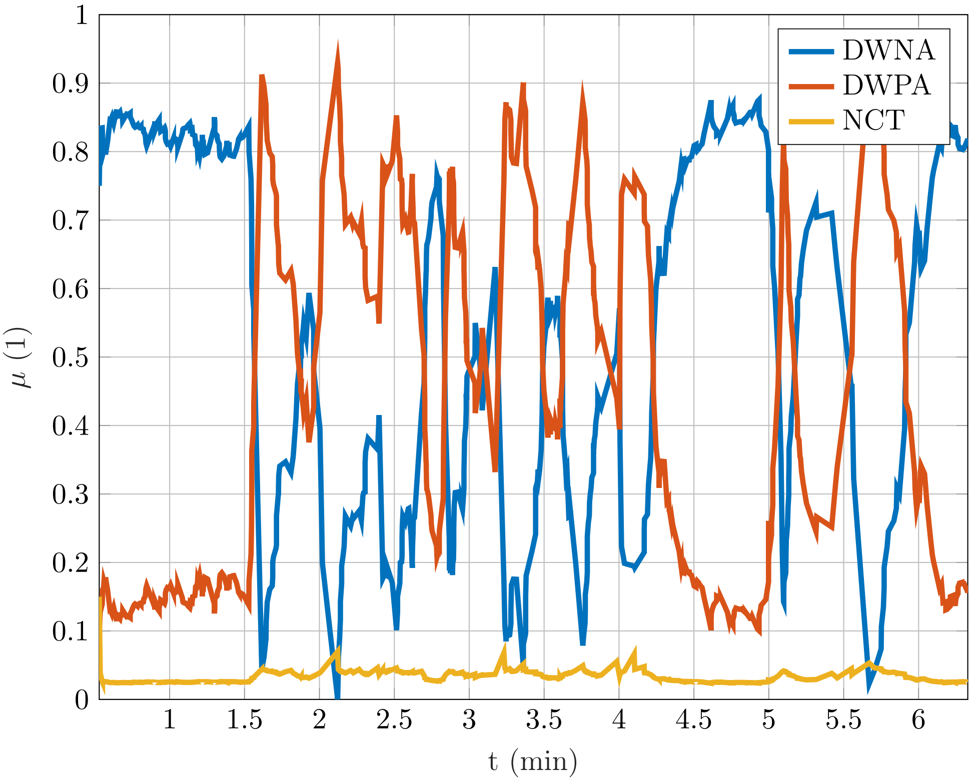

Figure 43: Track 2 - horizontal mode probabilities

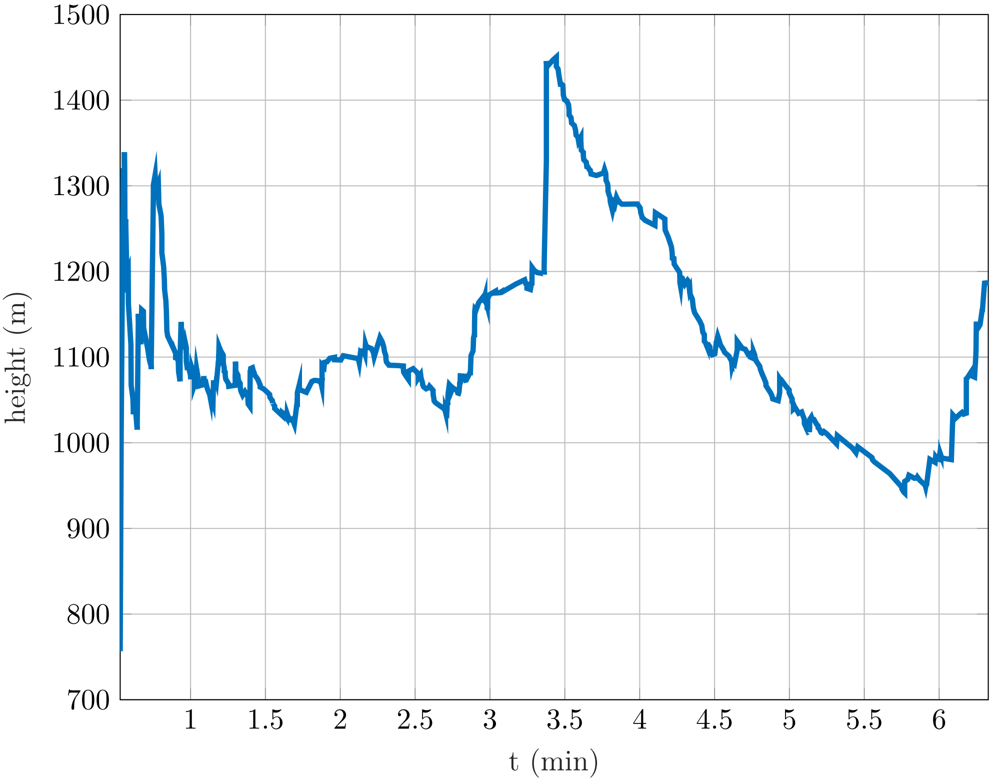

Figure 44: Track 2 - height

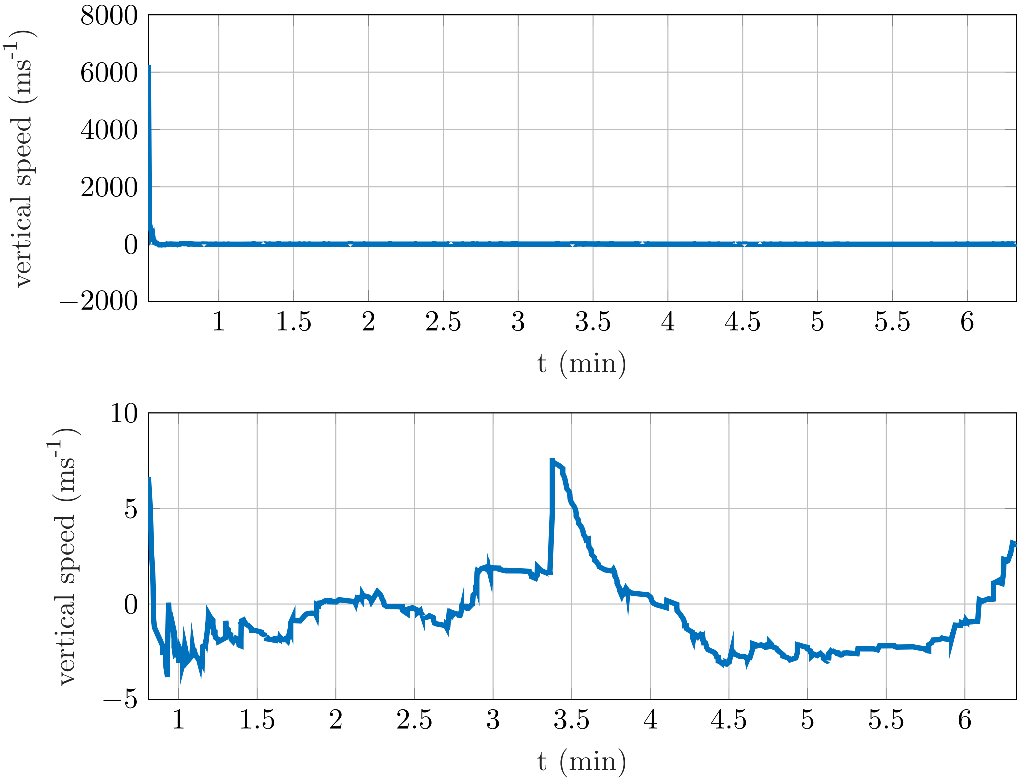

Figure 45: Track 2 - vertical speed

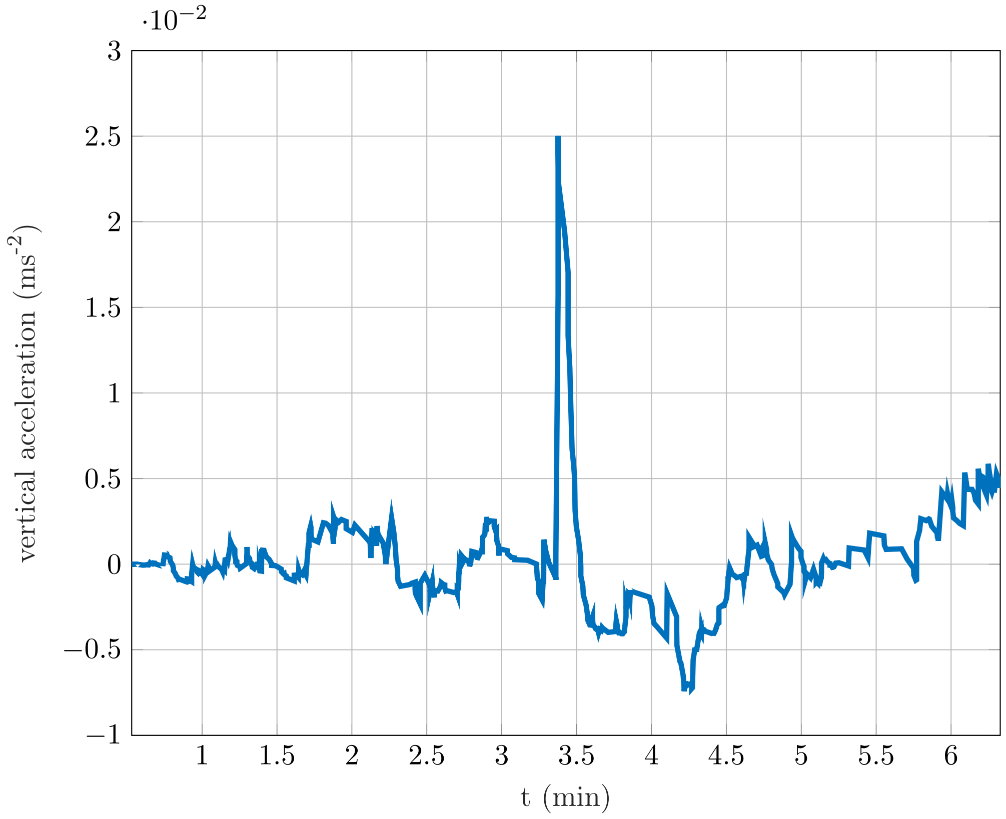

Figure 46: Track 2 - vertical acceleration

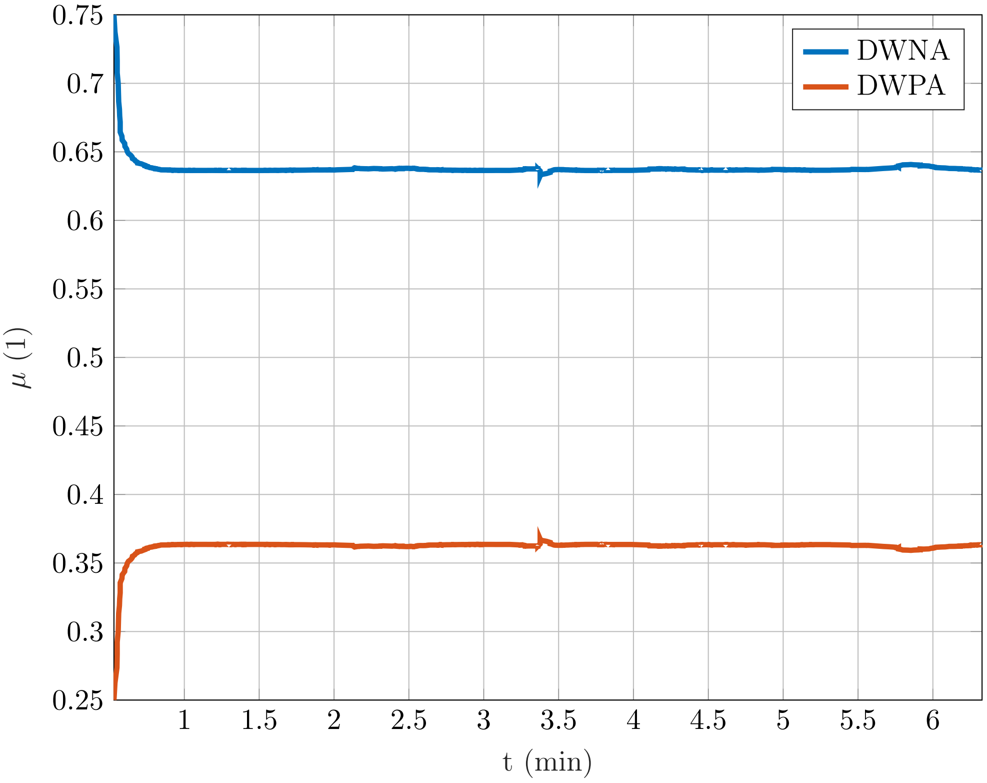

Figure 47: Track 2 - vertical mode probabilities

10 Conclusions

The purpose of the thesis was to design and develop a tracker for the requirements of the ATM Laboratory at the Department of Air Transport, Faculty of Transportation Sciences, Czech Technical University in Prague. The tracker was designed to process measurements obtained from the passive 3D Wide Area Multilateration system. The measurements are generated using four low-cost mode S receivers located in Prague. The designed tracker provides surveillance function by compounding the obtained measurements into tracks but will also be used for other scientific purposes since when working with positional data the most wanted position is the true one or its best estimate.

The thesis fist focuses on the ATM Laboratory itself. The description was done to present the software and hardware segment of the MLT system and to identify the requirements the laboratory has.

A brief introduction into target tracking was done to introduce problems the target tracking is trying to solve and what tools are being used to solve it.

One of the biggest part of the target tracking is the filtering and prediction function. The thesis thoroughly described this function starting from the Kalman filter up to adaptive algorithms for tracking of manoeuvring targets. The current state of the art method for tracking manoeuvring targets is the interacting multiple model algorithm. This method combines multiple Kalman filters or its extended versions to allow for transition between different motion modes and its corresponding models. The ever increasing computational resources allow for a use of advanced tracking methods such as this this even on moderate hardware.

After deriving the interacting multiple model estimator its design with all the necessary decisions was covered. Since the tracker is made for tracking civilian aircraft three motion models corresponding to the expected manoeuvres were selected. First the discrete white noise acceleration model which can, depending on the selected process noise, model nearly constant velocity motion. Second the discrete Weiner process acceleration model which can model nearly constant acceleration motion and should be prioritized when the aircraft is accelerating. Lastly the nearly coordinated turn model was selected. The first two models are decoupled between the coordinates. The nearly coordinated turn model allows for better estimation during constant turn rate manoeuvres at the expense of increased computational complexity since with unknown turn rate it is described by a non-linear function. For the tracking in vertical coordinate only the first two models were selected.

The interacting multiple model estimator is in literature mostly presented for periodic or nearly periodic sources such as radars. It was presented during the introduction of the ATM Laboratory that the passive WAM is highly aperiodic. Two possible modifications to the estimator were evaluated. First if time dependent computation of the Markov chain transition matrix could result in better mode transition. Through the testing it was concluded that the constant Markov chain transition is sufficient enough and any of the tested alternative methods did not result in significant improvement. Second tested modification was to compare if the estimate and covariance computation of the final state estimate using hard decision algorithm based on probability threshold is better than using Gaussian mixture for the same task. The testing showed that there is performance improvement while aircraft is in a single mode for longer time but this does not outweigh performance decrease during manoeuvre transition.

After designing the filtering and prediction function of the tracker focus was on track maintenance. Since the laboratory currently processes only mode S measurement methods for track maintenance can be based on the unique ICAO 24 bit address.

The measurement to track association function is also mostly based on the unique ICAO 24 bit address. Although its use could bring the full solution to the measurement to track association the principle behind multilateration still presents unresolved association situations. An algorithm for how to resolve measurement and track duality was presented.

The designed tracker was developed for testing purposes in the Matlab programming language and for real time processing in C++. The high level design was covered in the thesis and the full implementation can be seen in the provided source codes.

In the end the developed tracker was tested on data from real traffic recording. The results showed that the data association function proved sufficient for the required application with some exceptions where track duality was held for couple of seconds. The filtering of the tracker was mostly degraded due to a very large measurement noise. The standard deviation of the position for the current geometrical configuration of aerials combined with time stamping accuracy of the low-cost receivers reaches up to several kilometres or even tens of kilometres. This causes large initial error but after a while it greatly reduces to provide a reasonable estimate given the situation.

The implementation of the tracker was in general successful and provides the necessary function the ATM Laboratory requires. There is a space for improvement but the main purpose to provide the basics for further research and to evaluate the use of modern tracking methods on low-cost surveillance sensors was accomplished.

Bibliography

1. *EUROCONTROL SEVEN - YEAR FORECAST FEBRUARY 2019: Flight Movements and Service Units 2019 - 2025*. EUROCONTROL, [online] 2019. No. 19/01/15. [visited on 2019-03-02]. Available also from: `https://www.eurocontrol.int/articles/forecasts`.
2. *SESAR - Deployment Programme*. 2018th ed. SESAR Deployment Manager, [online] 2018. [visited on 2019-03-02]. Available also from: `https://www.sesardeploymentmanager.eu/wp-content/uploads/2019/02/Sesar-Deployment-Programme-edition-2018-_FINAL.pdf`.
3. *EUROCONTROL Specificationfor Advanced-Surface Movement Guidance and Control System (A-SMGCS) Services*. .01st ed. EUROCONTROL, [online] 2018. No. 01/03/2018. [visited on 2019-03-02]. Available also from: `https://www.eurocontrol.int/sites/default/files/content/documents/single-sky/specifications/edsp_17_003_spec_asmgcs_v1.0.pdf`.
4. BAR-SHALOM, Yaakov.; LI, Xiao-Rong.; KIRUBARAJAN, Thiagalingam. *Estimation with applications to tracking and navigation*. New York: Wiley, 2001. ISBN 978-0471416555.
5. *Doc 9683: Airborne CollisionAvoidance System(ACAS) Manual*. 1st ed. Montreal, Quebec, Canada: International Civil Aviation Organization, 2006.
6. ARTAS SMSP ARTICLE [online] [visited on 02/03/2019]. Available from: `https://www.eurocontrol.int/sites/default/files/publication/files/artas-article-paper-update-08032017.pdf`.
7. ZACH, Martin. *Návrh nízkonákladového MLAT systému*. Praha, 2015. Diplomová práce. České vysoké učení technické v Praze, Fakulta dopravní.
8. UMLAUF, Lukáš. *Aplikace MLAT metody nad sítí low-cost ADS-B přijímačů*. Praha, 2016. Diplomová práce. České vysoké učení technické v Praze, Fakulta dopravní.
9. TURKOVÁ, Eliška. *MLAT metoda v rámci sítě ADS-B přijímačů na FD*. Praha, 2019. Diplomová práce. České vysoké učení technické v Praze, Fakulta dopravní.
10. *Receiver Range* [online]. Prague: Czech Technical University in Prague, 2019 [visited on 09/03/2019]. Available from: `http://atm-lab.fd.cvut.cz/`.
11. LIPTÁK, Tomáš; LUKEŠ, Petr. METHOD FOR EVALUATION OF LOW-COST ADS-B RECEIVERS TIMESTAMPING LATENCY. In: *7th International Scientific Conference for Doctoral Students May 30 - 31, 2018 in Kosice*. Slovakia: TECHNICAL UNIVERSITY OF KOSICE, FACULTY OF AERONAUTICS, 2018.
12. LANGLEY, Richard B. The Mathematics of GPS. *GPS WORLD*. 1991, vol. July/August 1991.
13. BUCHER, Ralph; MISRA, D. A Synthesizable VHDL Model of the Exact Solution for Three-dimensional Hyperbolic Positioning System. *VLSI Design*. 2002-01-01, vol. 15, no. 2, pp. 507–520. ISSN 1065-514X. Available from DOI: `10.1080/1065514021000012129`.

14. GAVIRIA, Ivan A. Mantilla. *New Strategies to Improve Multilateration Systems in the Air Traffic Control*. Valencia, Spain, 2013. Doctoral Thesis. Universitat Politécnica de Valéncia.

15. FANG, B. T. Simple solutions for hyperbolic and related position fixes. *IEEE Transactions on Aerospace and Electronic Systems*. 1990, vol. 26, no. 5, pp. 748–753. ISSN 0018-9251. Available from DOI: 10.1109/7.102710.

16. BAR-SHALOM, Yaakov. *Multitarget-multisensor tracking: applications and advances*. Boston: Artech House, 2000. ISBN 978-1580530910.

17. BLACKMAN, Samuel S.; POPOLI, Robert. *Design and analysis of modern tracking systems*. Boston: Artech House, 1999. ISBN 15-805-3006-0.

18. COLLINS, J. B.; UHLMANN, J. K. Efficient gating in data association with multivariate Gaussian distributed states. *IEEE Transactions on Aerospace and Electronic Systems*. 1992, vol. 28, no. 3, pp. 909–916. ISSN 0018-9251. Available from DOI: 10.1109/7.256316.

19. DUAN, Y.; TAN, X.; QU, Z.; WANG, H.; WANG, J. Adaptive tracking gate design for maneuvering target in clutter environment. In: *2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. 2017, pp. 1449–1455. Available from DOI: 10.1109/ITNEC.2017.8285037.

20. LIU, X.; WANG, K.; LI, D.; XU, J.; PAN, J. A two-stage gating algorithm for joint probability data association filter. In: *IEEE 10th INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING PROCEEDINGS*. 2010, pp. 381–384. ISSN 2164-5221. Available from DOI: 10.1109/ICOSP.2010.5657198.

21. BAR-SHALOM, Y.; DAUM, F.; HUANG, J. The probabilistic data association filter. *IEEE Control Systems Magazine*. 2009, vol. 29, no. 6, pp. 82–100. ISSN 1066-033X. Available from DOI: 10.1109/MCS.2009.934469.

22. BROOKNER, Eli. *Tracking and Kalman filtering made easy*. New York: Wiley, c1998. ISBN 04-711-8407-1.

23. KIRUBARAJAN, T.; BAR-SHALOM, Y. Kalman filter versus IMM estimator: when do we need the latter? *IEEE Transactions on Aerospace and Electronic Systems*. 2003, vol. 39, no. 4, pp. 1452–1457. ISSN 0018-9251. Available from DOI: 10.1109/TAES.2003.1261143.

24. RIBEIRO, Maria Isabel. *Kalman and Extended Kalman Filters: Concept, Derivation and Properties* [online]. 2004th ed. Lisboa PORTUGAL: Institute for Systems and Robotics, 2004 [visited on 29/03/2019]. Available from: http://users.isr.ist.utl.pt/~mir/pub/kalman.pdf.

25. NAGY, Ivan. *Stochastické Systémy* [online]. Praha: Fakulta dopravní, České vysoké učení technické v Praze [visited on 29/03/2019]. Available from: http://staff.utia.cas.cz/suzdaleva/pdfka/StSysTexty.pdf.

26. KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*. 1960, vol. 82, no. 1. ISSN 00219223. Available from DOI: 10.1115/1.3662552.

27. LI, Q.; LI, R.; JI, K.; DAI, W. Kalman Filter and Its Application. In: *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*. 2015, pp. 74–77. Available from DOI: `10.1109/ICINIS.2015.35`.

28. LACEY, Tony. Tutorial: The Kalman Filter. In: [online]. Boston,Massachusetts: MIT, Chapter 11,133–140 [visited on 29/03/2019].

29. SINGER, R. A. Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets. *IEEE Transactions on Aerospace and Electronic Systems*. 1970, vol. AES-6, no. 4, pp. 473–483. ISSN 0018-9251. Available from DOI: `10.1109/TAES.1970.310128`.

30. MAZOR, E.; AVERBUCH, A.; BAR-SHALOM, Y.; DAYAN, J. Interacting multiple model methods in target tracking: a survey. *IEEE Transactions on Aerospace and Electronic Systems*. 1998, vol. 34, no. 1, pp. 103–123. ISSN 0018-9251. Available from DOI: `10.1109/7.640267`.

31. BARAM, Y.; SANDELL, N. Consistent estimation on finite parameter sets with application to linear systems identification. *IEEE Transactions on Automatic Control*. 1978, vol. 23, no. 3, pp. 451–454. ISSN 0018-9286. Available from DOI: `10.1109/TAC.1978.1101745`.

32. ROSS, Sheldon M. *Introduction to probability models*. 10th ed. Boston: Academic Press, c2010. ISBN 978-012-3756-862.

33. BAR-SHALOM, Y.; CHALLA, S.; BLOM, H. A. P. IMM estimator versus optimal estimator for hybrid systems. *IEEE Transactions on Aerospace and Electronic Systems*. 2005, vol. 41, no. 3, pp. 986–991. ISSN 0018-9251. Available from DOI: `10.1109/TAES.2005.1541443`.

34. P. BLOM, H. A. An efficient filter for abruptly changing systems. In: *The 23rd IEEE Conference on Decision and Control*. 1984, pp. 656–658. Available from DOI: `10.1109/CDC.1984.272089`.

35. VIVONE, G.; BRACA, P.; HORSTMANN, J. Variable structure interacting multiple model algorithm for ship tracking using HF surface wave radar data. In: *OCEANS 2015 - Genova*. 2015, pp. 1–8. Available from DOI: `10.1109/OCEANS-Genova.2015.7271644`.

36. BLAIR, W.D.; WATSON, G.A. *Benchmark Problem for Radar Resource Allocation and Tracking maneuvering Targets in the Prsence of ECM: Technical Report NSWCDD/TR-96/10*. Dahlgren, Virginia: Naval Surface Warfare Center, Dahlgren Division, 1996.

37. LI, X. R.; BAR-SHALOM, Y. Design of an interacting multiple model algorithm for air traffic control tracking. *IEEE Transactions on Control Systems Technology*. 1993, vol. 1, no. 3, pp. 186–194. ISSN 1063-6536. Available from DOI: `10.1109/87.251886`.

38. M. T. BUSCH, Samuel S. Blackman. *Evaluation of IMM filtering for an air defense system application*. 1995. Available from DOI: `10.1117/12.217717`.

39. RONG LI, X.; JILKOV, V. P. Survey of maneuvering target tracking. Part I. Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*. 2003, vol. 39, no. 4, pp. 1333–1364. ISSN 0018-9251. Available from DOI: `10.1109/TAES.2003.1261132`.

40. BAUSCHLICHER, J.; ASHER, R.; DAYTON, D. A comparison of Markov and constant turn rate models in an adaptive Kalman filter tracker. In: *Proceedings of the IEEE National Aerospace and Electronics Conference*. 1989, 116–123 vol.1. Available from DOI: 10.1109/NAECON.1989.40200.

41. W. DALE BLAIR, Gregory A. Watson. *IMM algorithm and aperiodic data*. 1992. Available from DOI: 10.1117/12.138208.

42. VERDE-STAR, Luis. On linear matrix differential equations. *Advances in Applied Mathematics*. 2007, vol. 39, no. 3, pp. 329–344. ISSN 01968858. Available from DOI: 10.1016/j.aam.2006.06.002.

43. *LETECKÝ PŘEDPIS O CIVILNÍ LETECKÉ TELEKOMUNIKAČNÍ SLUŽBĚSVAZEK III - KOMUNIKAČNÍ SYSTÉMY*. Úřad pro civilní letectví: MINISTERSTVO DOPRAVY ČESKÉ REPUBLIKY, 2003. No. 285/2003-220-SP/1. Available also from: http://lis.rlp.cz/predpisy/predpisy/index.htm.

List of Figures

List of Tables

A Attachment

A.1 Matlab implementation

```
ctEkfInitWithTransform.m
ctEkfStep.m
ctEkfStepWithTransform.m
dwnaKfInit.m
dwnaKfStep.m
dwpaKfInit.m
dwpaKfStep.m
ekfStep.m
estimateAndCovarianceCombination.m
getCjOverline.m
getIdealTransitionMatrix.m
getIdealTransitionMatrixLimit.m
getInputStateVector.m
immConst.m
immExpLimited.m
immIdeal.m
immLinearLimited.m
initializeKfs.m
kfStep.m
mixing.m
mixingProbabilities.m
modeMatchedFiltering.m
modeProbabilityUpdate.m
transformCartesianToTrun.m
transformTurnToCartesian.m
```

A.2 C++ implementation

Constants.h

CtEkf.cpp

CtEkf.h

CtEkfUtilities.cpp

CtEkfUtilities.h

DwnaKf.cpp

DwnaKf.h

DwnaXYKf.cpp

DwnaXYKf.h

DwpaKf.cpp

DwpaKf.h

DwpaXYKf.cpp

DwpaXYKf.h

ImmAlgorithm.cpp

ImmAlgorithm.h

ImmHorizontal.cpp

ImmHorizontal.h

ImmVertical.cpp

ImmVertical.h

KalmanFilterAlgorithm.cpp

KalmanFilterAlgorithm.h

KfBase.cpp

KfBase.h

MathUtils.cpp

MathUtils.h

SequenceGenerator.cpp

SequenceGenerator.h

Time.cpp

Time.h

Track.cpp

Tracker.cpp

Tracker.h

Track.h

XYKfBase.cpp

XYKfBase.h