UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

CONTRIBUTIONS TO
SURVEILLANCE AND AIR
TRAFFIC MANAGEMENT ON
THE AIRPORT AREA

TESIS DOCTORAL

JAIME LÓPEZ-ARAQUISTAIN LÓPEZ

INGENIERO EN TELECOMUNICACIÓN
2018

DEPARTMENT:

Departamento de Señales, Sistemas y Radiocomunicaciones

Escuela Técnica Superior de Ingenieros de Telecomunicación

Universidad Politécnica de Madrid

PH.D. THESIS:

**Contributions to Surveillance and Air Traffic Management on
the Airport Area**

Author:

Jaime López-Araquistain López

Advisor:

Juan Alberto Besada Portas

Committee named by the Rector of Universidad Politécnica de Madrid,
on the ____ of _____ 2018.

President: _____

Member: _____

Member: _____

Member: _____

Secretary: _____

After the defence of the Ph.D. Thesis on the ____ of _____
2018, at the Escuela Técnica Superior de Ingenieros de Telecomunicación,
the committee agrees to grant the following qualification:

President

Secretary

Members

Acknowledgements

I would like to start this thesis by thanking all the people who have helped me and supported me throughout this stage of my life.

First of all, I would like to thank Prof. Juan Besada for all these years in which he has been my tutor, both since I joined the group as an intern until these last years as a Ph.D. student. Thank you for all the help and time you have given me. Besides, I would like to thank the other professors of the group - José Ramón, Gonzalo, Ana and Javier - for all their advice.

I would like to thank all the colleague, have not been few, whom I have been getting to know over the years in the GPDS for the help and time they gave me during my first year. In particular, I am grateful to Angel who has been with me throughout the whole doctorate. Who would have thought that his good and (more common) bad jokes were going to make all these years more enjoyable.

I am grateful to my friends, who Thursday after Thursday have helped to me switch off and recharge my batteries to keep moving forward. Hopefully these traditions will never disappear.

A very special gratitude goes out to my family, in particular to my parents - Gracia and Javier - and to my brother Daniel, because without their teachings and patience I would not have gotten so far. I am who I am thanks to them.

A special mention to my aunt Rocío, how right you were when you told me that drawing would be useful in life.

Finally, last but by no means least, I would like to wholeheartedly thank Sandra for all the things she has had to endure throughout these hard years and because without her support I would not have been able to achieve this challenge without getting lost along the way.

PS: thanks to you too, Internet.

Abstract

In this Ph.D thesis, certain improvements for the *Air Traffic Management* (ATM) on the airport area are proposed. Specifically, a series of improvements has been proposed for the *Advanced Surface Movement Guidance and Control System* (A-SMGCS) system which is in charge of the routing, guidance, surveillance, and aircraft control functions in that area. The work done has been focused on these last two functions.

The surveillance function is mainly in charge of, from the available sensors measures, obtaining accurately the position of the targets as well as their identification. To do this, this function has to perform the following tasks: association of measures, correction of biases, and filtering of measures. A bias correction architecture is proposed based on the division of them according to the source that generates them. To be more specific, the biases estimated are: the bias dependent only on the sensor, the bias dependent on the target-sensors pair, and finally, the offset that exists in the vertical measurements according to whether the sensor is geometric or barometric. Regarding the filtering of the measurements, a system that uses two filters in parallel with a selection logic to obtain the estimation of the position of the targets is proposed. The first filter uses the information contained in the airport map to improve the performance. This filter is based on a Bayesian Multiple Hypotheses approach in order to determine the segments being followed by the target after road junctions. In addition, this filter decouples the target movement in two components, one related to along road movement, and another one designed to track movement transversal to road. The second filter of the tracking system is a robust filter that does not use the map information and is therefore capable of obtaining the estimate in situations

where the map-based filter is not (landing aircraft, runway excursions, etc.). In addition, both the complete tracking system and each filter have been evaluated and their performance has been compared to other filters in the literature. Hence, it has been possible to demonstrate that the designed tracking system is competitive and presents a better performance than other solutions.

The control function is responsible for detecting possible conflicts that may exist on the airport area. It uses the filtered positions of the targets obtained by the surveillance function, as well as the established routes obtained by the guidance function. In this thesis a safety net capable of detecting the following conflicts has been designed and analysed: breaching the minimum horizontal separation between aircraft, exceeding the maximum speed of the taxiways, and the runways incursions. The designed safety nets are based on a rigorous characterization of the problem and can work either with only the filtered positions of the targets or also using the map information, thus improving the benefits. In addition, the designed safety nets have been evaluated, demonstrating their correct performance as well as the drastic reduction of the false alarm probability and the early detection of conflicts when the information on the map is being used.

Finally, the main conclusions of this thesis are presented as well as the possible future lines of research.

Resumen

En el trabajo de esta tesis doctoral se proponen ciertas mejoras para el *Air Traffic Management* (ATM) en el área del aeropuerto. Concretamente se ha propuesto una serie de mejoras para el sistema *Advanced Surface Movement Guidance and Control System* (A-SMGCS) que es el encargado de las funciones de enrutado, guiado, vigilancia y control de las aeronaves en dicha área. Es en estas dos últimas funciones en las que se ha centrado el trabajo.

La función de vigilancia se encarga principalmente de, a partir de las medidas de los sensores disponibles, obtener con precisión la posición de los blancos, así como su identificación. Para ello, esta función tiene que realizar las tareas de asociación de las medidas, corrección de sesgos y filtrado de las medidas. Se propone una arquitectura de corrección de sesgos basada en la división de éstos en tres, según la fuente que los genera. En concreto se estima el sesgo dependiente solo del sensor, el sesgo dependiente de la pareja blanco-sensores y por último el offset que existe en las medidas verticales según el sensor sea geométrico o barométrico. Respecto al filtrado de las medidas, se propone un sistema que utiliza dos filtros en paralelo con una lógica de selección para obtener la estimación de la posición de los blancos. El primer filtro utilizado es uno que utiliza la información contenida en el mapa del aeropuerto para mejorar las prestaciones. Para ello, este filtro está basado en un enfoque bayesiano de múltiples hipótesis para determinar las carreteras por las que se mueve el blanco. Además, el sistema descompone el movimiento del blanco en su componente longitudinal y transversal. El segundo filtro que compone el sistema es un filtro robusto que no utiliza la información del mapa y por tanto es capaz de obtener la estimación en situaciones en las que el filtro basado en el mapa no es capaz (aviones aterrizando, salidas de pista,

etc.). Además, tanto el sistema de seguimiento completo como cada filtro por separado han sido evaluados y sus prestaciones han sido comparadas con otros filtros de literatura. De esta manera se ha podido demostrar que el sistema de seguimiento es competitivo y presenta una mejor calidad respecto a otras soluciones.

La función de control se encarga de detectar los posibles conflictos que pueda haber en el área del aeropuerto. Utiliza las posiciones filtradas de los blancos, obtenidas por la función de vigilancia, así como las rutas establecidas, obtenidas por la función de guiado. En esta tesis se ha diseñado y analizado una safety net capaz de detectar los siguientes conflictos: incumplir la separación horizontal mínima entre aeronaves, exceder la velocidad máxima de las taxiways y las incursiones en las runways. Las safety nets diseñadas se basan en una caracterización rigurosa del problema y pueden funcionar o sólo con las posiciones de los blancos o utilizando también la información del mapa, mejorando así las prestaciones. Adicionalmente, se ha medido la calidad de las safety nets desarrolladas, demostrando su correcto funcionamiento así como la reducción drástica de probabilidad de falsa alarma y detección anticipada de los conflictos cuando se usa la información del mapa.

Por último, se presentan las principales conclusiones de esta tesis, así como las posibles futuras líneas de investigación.

Contents

Acknowledgements	iii
Abstract	v
Resumen	vii
Contents	ix
Acronyms	xiii
List of Figures	xvii
List of Tables	xxiii
1 Introduction	1
1.1 Air Traffic Control	1
1.2 Airport Surface Operations	2
1.3 A-SMGCS	4
1.3.1 Surveillance Function	5
1.3.2 Control Function	14
1.3.3 Routing Function	16
1.3.4 Guidance Function	17
1.4 Thesis Objectives and Structure	17

I	Surveillance Function	21
2	Map-Based Tracking on Airport Area	23
2.1	Introduction	23
2.2	Map-Based Tracking Problem Statement	26
2.3	Map-Based Tracking Filter Structure	29
2.4	Coordinate Projection Methods	34
2.4.1	Projection from 2D to Longitudinal ($s_{ID}(m)$)	34
2.4.2	Projection from Longitudinal ($s_{ID}(m)$) to Kinetic state	43
2.4.3	Projection from 2D to Transversal ($s_{ID}(m)$)	47
2.4.4	Projection from Transversal ($s_{ID}(m)$) to 2D offset	48
2.5	Along-segment IMM filter description	49
2.6	Map-Based Multiple Hypothesis Filtering	51
2.6.1	Search for close segments	51
2.6.2	Tracking hypotheses management	52
2.6.3	Tracking hypotheses probability calculation	56
2.6.4	Tracking hypotheses combination and pruning	57
2.7	Lateral Drift Tracker	58
2.8	Map-Based Vertical Tracking	60
3	Robust Estimation Filters on Airport Area	65
3.1	Introduction	65
3.2	Horizontal Robust IMM	68
3.2.1	Horizontal Dynamic Models	69
3.2.2	Horizontal Transition Probabilities	78
3.3	Vertical Robust IMM	79
3.3.1	Vertical Dynamic models	80
3.3.2	Vertical Transition Probabilities	83
3.4	Integration of Robust Filter and Map-Based Filters	84
4	Bias Estimation and Correction	87
4.1	Introduction	87
4.2	Association and Track Management	89
4.3	Sensor Biases	89

4.3.1	Local Bias Estimation	91
4.3.2	Barometric-geometric offset estimation	96
4.3.3	Global Bias Estimation	100
5	Surveillance Function Results	113
5.1	Introduction	113
5.2	Horizontal Tracking Performance	114
5.2.1	Bias Estimation Results	116
5.2.2	Map-Based Tracker Performance	119
5.2.3	Robust Tracker Performance	131
5.2.4	Map-Based Tracking plus Robust Tracking	136
5.2.5	Real Data Results	140
5.3	Vertical Tracking Performance	146
II	Control Function	153
6	Conflict Prediction and Detection	155
6.1	Introduction	155
6.2	Minimum Separation Conflict	160
6.2.1	Mathematical Analysis	160
6.2.2	Separation Safety Net	175
6.3	Separation Safety Net Results	180
6.4	Runway Incursion Conflict	187
6.4.1	Mathematical Analysis	187
6.4.2	Runway Incursion Safety Net	191
6.5	Runway Incursion Safety Net Results	194
III	Conclusions and Bibliography	197
7	Conclusions	199
7.1	Discussion	200
7.2	Contributions	204
7.3	Further Work	206

Bibliography	209
IV Appendices	225
A Estimation Methods	227
A.1 Kalman Filter	227
A.2 IMM Algorithm	231

Acronyms

A-SMGCS Advanced Surface Movement Guidance and Control System

A-VDGS Advanced-Visual Guidance Docking Systems

ADS-B Automatic Dependent Surveillance - Broadcast

AIS Aeronautical Information Service

APM Approach Path Monitoring

APW Area Proximity Warning

ATC Air Traffic Control

ATCO Air Traffic Controller

ATFM Air Traffic Flow Management

ATM Air Traffic Management

CATC Conflicting Air Traffic Control Clearance

CDR Conflict Detection and Resolution

CKF Cubature Kalman Filters

CMAC Conformance Monitoring Alerts for Controllers

EHS Enhanced Surveillance

EKF Extended Kalman Filter

ELS Elementary Surveillance

FOM Figure of Merit

FRUIT False Replies Unsynchronised to Interrogator Transmission

GBPA Generalized Pseudo-Bayesian Approach

GNSS Global Navigation Satellite System

GS Ground Speed

IAS Indicated Air Speed

IATA International Air Transport Association

ICAO International Civil Aviation Organization

ILS Instrument Landing System

IMM Interacting Multiple Model

ISA International Standard Atmosphere

KF Kalman Filter

MBMHF Map-Based Multiple Hypothesis Filtering

MCMC Markov Chain Montecarlo

MHT Multiple Hypothesis Tracker

MLAT Multilateration

MLE Maximum Likelihood Estimator

MMSE Minimum Mean Square Error

MSAW Minimum Safe Altitude Warning

MTI Moving Target Indicator

NAC_p Navigation Accuracy Category for Position

NAC_v Navigation Accuracy Category for Velocity

NN Nearest Neighbour

PDF Probability Density Function

RMCA Runway Monitoring and Conflict Alerting

SMGCS Surface Movement Guidance and Control System

SMR Surface Movement Radar

SSR Secondary Surveillance Radar

STCA Short-term Conflict Alert

TCAS Traffic alert and Collision Avoidance System

TCL Taxiway Centreline Lights

TDOA Time Difference of Arrival

TMA Terminal Manoeuvring Area

UKF Unscented Kalman Filters

VS-IMM Variable Structure IMM

WAM Wide Area Multilateration

List of Figures

1.1	Phases of surface operation. Extracted from [Cassell and Smith, 1995] ©1999 IEEE	3
1.2	Relation between the A-SMGCS functionalities	5
1.3	Diagram of: (a) centralized architecture, (b) distributed architecture	7
1.4	MLAT system	11
1.5	ADS-B system	13
1.6	Thesis objectives architecture	18
2.1	Map example with the three basic elements	27
2.2	Global filter structure	30
2.3	Target position representation	31
2.4	Longitudinal estimator diagram	33
2.5	Transversal estimator diagram	33
2.6	Workflow of the MBMHF	52
2.7	Effect of the sensor noise	53
2.8	Flow diagram of the hypotheses update	54
2.9	Distance and speed transformation depending on the common node	56
2.10	Lateral drift estimation	59
2.11	Lateral drift correction	60
2.12	Heathrow Instrument Approach Chart Detail	61
3.1	Robust filter block diagram	67
3.2	Surface Robust Filter	68
3.3	Robust Vertical Tracker	80

3.4	Map-based filter with robust filter	85
4.1	Basic elements of tracking system	88
4.2	Block diagram of the biases correction	90
4.3	Differences in the centroids of the SMR and ADS-B	92
4.4	Estimation of bias at X Coordinate	95
4.5	Estimation of bias at Y Coordinate	95
4.6	Geometric and barometric measures from Barcelona airport sensors	96
4.7	Estimated barometric offset from Barcelona airport sensors . .	100
4.8	Radar global biases estimation	102
4.9	Range bias estimations for the SMR 1 from each track	109
4.10	Range bias estimation from measures difference	109
4.11	Azimuth bias estimations from measures difference	110
5.1	Taxi trajectory at Barcelona El Prat Airport	115
5.2	Estimated Local Bias	117
5.3	X-Coordinate Estimated Local Bias	118
5.4	Y-Coordinate Estimated Local Bias	118
5.5	Estimated Lateral Drift	119
5.6	Projection to a single road	120
5.7	Projection result in an intersection	121
5.8	Tracking result in an intersection without lateral drift	122
5.9	Lateral drift estimation	122
5.10	Tracking result in an intersection with lateral drift	123
5.11	Mean and 95th distance error	124
5.12	Distance estimation error	125
5.13	Ground speed estimation error	125
5.14	Heading estimation error	126
5.15	Drift estimation error	126
5.16	Ideal trajectory	128
5.17	Distance error for the different trackers	129
5.18	Ground speed error for the different trackers	129
5.19	Heading error for the different trackers	130

5.20	Taxi trajectory distance error for the Interacting Multiple Model (IMM) in the monosensor scenario	132
5.21	Taxi trajectory ground speed error for the IMM in the monosensor scenario	133
5.22	Taxi trajectory heading error for the IMM in the monosensor scenario	133
5.23	No local bias multisensor scenario: distance error	135
5.24	No local bias multisensor scenario: ground speed error	135
5.25	No local bias multisensor scenario: heading error	136
5.26	Local bias multisensor scenario: distance error	137
5.27	Local bias multisensor scenario: ground speed error	137
5.28	Local bias multisensor scenario: heading error	138
5.29	Map-Based Tracking + Robust Tracking: distance error	138
5.30	Map-Based Tracking + Robust Tracking: ground speed error	139
5.31	Map-Based Tracking + Robust Tracking: heading error	139
5.32	Barcelona El Prat Airport	141
5.33	Residual of the position	142
5.34	Ground Speed tracking result	143
5.35	Absolute difference between ground speed updates	144
5.36	Detail of ground speed tracking when no local bias estimation	145
5.37	Heading tracking error	145
5.38	Takeoff trajectory at Barcelona El Prat Airport	147
5.39	No barometric offset multisensor scenario: height error	149
5.40	No barometric offset multisensor scenario: vertical speed error	150
5.41	Barometric offset multisensor scenario: height error	150
5.42	Barometric offset multisensor scenario: vertical speed error	151
5.43	Vertical pseudomeasures scenario: height error	151
5.44	Vertical pseudomeasures scenario: vertical speed error	152
6.1	Complete CDR diagram block	156
6.2	Proposed conflict detector diagram block	159
6.3	Example of different distance conflict situations	160
6.4	d_{min} and distance PDF	162
6.5	Distance threshold and P_d with $\sigma = 5$	165

6.6	Distance threshold and P_d with $\sigma = 10$	166
6.7	Distance threshold and P_d with $\sigma = 20$	166
6.8	Probability of measurements breaking distance threshold with $\sigma = 5$	167
6.9	Probability of measurements breaking distance threshold with $\sigma = 10$	167
6.10	Probability of measurements breaking distance threshold with $\sigma = 20$	168
6.11	P_d and P_{fa} relation with $\sigma = 5$	169
6.12	P_d and P_{fa} relation with $\sigma = 10$	169
6.13	P_d and P_{fa} relation with $\sigma = 20$	170
6.14	P_d and P_{da} for same runway with relative speed of $5m/s$	172
6.15	P_{fa} for parallel runways with relative speed of $5m/s$	173
6.16	P_{fa_a} for parallel runways with relative speed of $5m/s$	173
6.17	P_d with 2-of-3 rule for same runway	174
6.18	P_{fa} with 2-of-3 rule for same runway	174
6.19	Safety net without map	175
6.20	Robust filter with map safety net	176
6.21	Map filter with map safety net	176
6.22	Route calculation example at Barcelona El Prat Airport	178
6.23	Offset route calculation example	179
6.24	Two targets trajectory in parallel runways	181
6.25	Separation Measured between parallel targets	182
6.26	Conflict detection with parallel targets	182
6.27	Parallel runways MonteCarlo results: (a) Robust Filter + Safety Net without Map; (b) MBMHF + Safety Net without Map; (c) Robust Filter + Safety Net with Map; (d) MBMHF + Safety Net with Map.	183
6.28	Two targets trajectory in conflict scenario	184
6.29	Measured distance for conflict scenario	185
6.30	Measured distance for conflict scenario	185

6.31	Same runway MonteCarlo results: (a) Robust Filter + Safety Net without Map; (b) MBMHF + Safety Net without Map; (c) Robust Filter + Safety Net with Map; (d) MBMHF + Safety Net with Map.	186
6.32	Position Probability Density Function (PDF) and segment . .	189
6.33	Rotated position PDF and rotated segment	190
6.34	Runway holding point	192
6.35	Trajectory for runway intrusion scenario	195
6.36	Runway intrusion alarms in one iteration	195
6.37	Runway intrusion alarms in a MonteCarlo	196
7.1	Proposed surveillance architecture	205
7.2	Diagram of a safety net exploiting the map information: (a) tracker with target localization, (b) tracker without target localization	206
A.1	IMM general structure	232

List of Tables

1.1	Typical taxi speeds	3
1.2	Position accuracy according to NACp	13
1.3	Speed accuracy according to NACv	14
4.1	Parameters used by the vertical offset estimator	100
4.2	Characteristics of the scenario sensors	108
5.1	Taxi trajectory description	115
5.2	SMR sensor characteristics	116
5.3	MLAT sensor characteristics	116
5.4	Parameters used by the estimators	117
5.5	MBMHF Constants values	120
5.6	Trajectory description	127
5.7	Computational load (number of track updates per second) . .	131
5.8	Centralized/Distributed IMM: manoeuvre distance error . . .	136
5.9	Absolute residual of the position	142
5.10	Ground speed absolute difference	143
5.11	Ground speed absolute difference without local bias estimation	144
5.12	Heading error	146
5.13	Takeoff trajectory description	147
5.14	SSR sensor characteristics	148
5.15	MLAT vertical characteristics	148
5.16	Landing trajectory description	149
6.1	Hypothesis testing: error types	164
6.2	Values of d_{th} and P_{fa} for different P_d and σ	170

6.3	Minimum distance from the runway centre line to a holding bay, runway-holding position or road-holding position	194
-----	---	-----

Chapter 1

Introduction

1.1 Air Traffic Control

Air Traffic Management (ATM), as defined by the International Civil Aviation Organization (ICAO) is “the dynamic, integrated management of air traffic and airspace - safely, economically and efficiently through the provision of facilities and seamless services in collaboration with all parties” [ICAO, 2005]. ATM is made up of three different parts: Air Traffic Flow Management (ATFM), Aeronautical Information Service (AIS), and Air Traffic Control (ATC). ATFM takes place before the flights begin, optimizing the traffic flow according to the capacity of the ATC and maintaining the safety and efficiency of the flights. AIS is responsible for transmitting all necessary information to airspace users, including information about safety, navigation, technical or even legal matters. Finally, ATC is the service by which aircraft are separated as they fly and at the airports where they land and take off again. It is in this last aspect where this thesis mainly focuses.

To be more specific, the ATC service is a process that is responsible for preventing collisions and expediting and maintaining an orderly flow of air traffic [ICAO, 2001]. These functions are performed for the following situations:

1. Aircraft flying within the designated control tower's responsibility area
2. Aircraft operating on the manoeuvring area
3. Aircraft landing and taking off
4. Aircraft moving on airport surface

Some of the functions of the ATC service on the airport vicinity are the following. First, the ATC has to be able to provide information to aircraft such as start up time procedures, e.g. expected time to take off, or the aerodrome and meteorological information, e.g. runway to be used. In addition, the ATC must also take action in the event of a runway incursion or obstructed runway, reporting the affected aircraft and modifying traffic as necessary. The ATC is also in charge of separating the aircraft thereby having no problems with wake turbulence and jet blast hazards. Finally, the ATC has to provide any other type of information on aerodrome conditions that may be relevant to any aircraft. Before discussing some ATC tools, the next section introduces the typical airport surface operations.

1.2 Airport Surface Operations

As previously mentioned, the scope of this thesis is located on airport surface and its surroundings. This extends from the airport surface to a radius of 5 and 10 nautical miles. In order to be able to design surveillance and safety net systems it is necessary to know the types of movements that exist in these areas, which can be summarized in the figure 1.1 (extracted from [Cassell and Smith, 1995]).

From the moment a given airplane lands, its rollout movement starts. This movement covers from the touchdown point to where it reduces the speed to approximately 60 knots (111 kmph). Between the rollout movement and the next phase there may be a phase called high speed taxi that occurs when the airplane uses a rapid exit taxiway. These rapid taxiways are linked to the runway at an angle that permits aircraft to exit the landing runway at

Figure 1.1: Phases of surface operation. Extracted from [Cassell and Smith, 1995] ©1999 IEEE

high speeds. After the rollout movement or the highspeed taxi, it begins the taxi phase that terminates upon arrival at the apron or parking area. Lastly, the apron movement is the final manoeuvre by which the aircraft parks. In the opposite direction, for the departing aircraft, the order of the phases go from the stand to taxi and to take off roll. The table 1.1 shows the typical average speeds of these phases.

Taxi Phase	Speed (kmph)
Rapid Exit	55 - 130
Normal / Apron taxiway (straight)	18 - 92
Normal / Apron taxiway (turn)	18 - 37
Stand / Stand taxilane	0 - 18

Table 1.1: Typical taxi speeds

At airports, the main location from which traffic is controlled is in the control tower. From there the ATCOs are responsible of the traffic on the surface of the airport and in the airport area, which generally comprises

5 to 10 nautical miles around it. Among the most common tools for controllers to carry out airport surface control are the SMGCS and its update the A-SMGCS. These two tools are now explained.

1.3 A-SMGCS

Originally, airport procedures were covered by the system known as Surface Movement Guidance and Control System (SMGCS) [ICAO, 1986]. This system was in charge of specifying the guidance and control systems for the airport surface. On the one hand, the guidance system was responsible for finding a path for vehicles and aircraft and maintaining them on that path. On the other hand, the control system was in charge of preventing collisions and ensuring smooth traffic flow. This was achieved using a series of painted guidelines and signs or more advanced tools such as switched taxiway centre lines and stop bars. The SMGCS provided guidance on the road from the landing runway to the parking position and backwards to the take-off runway. Additionally, it also provided guidance from a maintenance area to an apron, or from an apron to another apron.

Due to their relative simplicity, SMGCS systems are based primarily on the principle of “seeing and being seen” to maintain separation between the different aircraft and vehicles at the airport. But the need to reduce accidents and specially runway intrusion incidents, to increase total traffic capacity and to provide the ability to operate in low visibility conditions led to improved procedures at airports, i.e. the development of the Advanced Surface Movement Guidance and Control System (A-SMGCS) systems [ICAO, 2004a].

The A-SMGCS system must have four basic functionalities: surveillance, routing, control (also called safety net by EUROCONTROL [EUROCONTROL, 2010]) and guidance. At a high level, the surveillance function is responsible for providing positioning and identification of vehicles and aircraft on the airport area. The routing function is responsible for generating individual routes for each vehicle taking into account the restrictions and

Figure 1.2: Relation between the A-SMGCS functionalities

capacities of the airport. The control function is responsible for ensuring security by preventing conflicts and enforcing resolutions in the case of these events. Finally, the guidance function is responsible for providing clear indications to pilots to follow their routes, managing the Taxiway Centreline Lights (TCL), the stop bars, and other visual aids. Figure 1.2 shows how each of these four functionalities relates to each other. Each of these four functions is explained in more detail below.

1.3.1 Surveillance Function

The surveillance function as defined by ICAO has the following functions. First, it has to provide accurate position information for all movements that are performed in addition to the identification of all targets. It has to be able to deal with not only moving targets but also with static targets. Furthermore, it must provide this information to the guidance, control and routing systems so they can carry out their functions. Lastly, it has to be capable to function in all weather conditions.

In order to guarantee the accuracy requirements of the surveillance sys-

tem, it is often necessary to deploy different sensors since a single sensor technology would not achieve the required performance or there would be coverage problems. The most common sensor on the airport surface is the Surface Movement Radar (SMR) [Schwab and Rost, 1985]. In addition to this radar, there are usually Multilateration (MLAT) based sensors [Xu et al., 2010] and Automatic Dependent Surveillance - Broadcast (ADS-B) systems available [Hicok and Lee, 1998]. Finally, among the usual sensors there may be available Secondary Surveillance Radar (SSR) [Skolnik, 1970] although, due to coverage aspects, it usually only provides measures of aircraft flying around the airport.

The use of different complimentary sensors demands the definition of a complete multisensor data fusion architecture, addressing problems such as the way of integrating the measurements in the tracker, the transformation to a common coordinate system or the appearance of biases, etc. These problems have been the subject of much research work. For example, regarding trackers, the sensor measures can be merged through a centralized system [Chen et al., 2000] [Cho et al., 2013] or a distributed (autonomous) system [Liggins et al., 1997] [Cherchar et al., 2010] [Zhou et al., 2006]. Centralized data processing means that all the sensor measures are sent to a central tracker where the data are fused. Distributed approach architecture implies that each sensor has a local tracker and then a central tracker associates the local tracks and fuses it. These two architectures can be seen in figure 1.3. In addition, there are other hybrid architectures, such as those based on the hierarchical clustering models [Wang et al., 2016].

Also in the field of trackers, the quality of the airport surface tracking systems has been improved by incorporating the information from the airport map into the system [Herrero et al., 2003] [Farina et al., 2003]. Using the map information adds an additional process consisting on locating vehicles on the airport roads. This location information can be used either to improve the quality of measurements by applying some kind of pre-processing [Oh et al., 2015] [Streller, 2008], or to improve the prediction movement models by applying constraints [Simon and Chia, 2002] [Xinyan and Wei, 2007]. Finally, once the tracker has been implemented it is checked, using a set of

Figure 1.3: Diagram of: (a) centralized architecture, (b) distributed architecture

metrics that have to take into account if the target is manoeuvring and the quality of the available sensors [Besada et al., 2014] [Ristic et al., 2011], if the performance meets the requirements.

Another typical problem of multisensor systems is the estimation and correction of sensor measurement biases. The non-correction of these biases has an impact on both the accuracy of the position estimation as well as on the association between the tracks and the measures. These biases have multiple sources such as bad sensor calibration, propagation effects, etc. The introduction of new sensors has made that simple correction models [Lin et al., 2004] do not perform well in real-life scenarios. For this reason, the work in this field has developed new models that correct both the typical biases of each sensor and the typical biases associated with the use of different sensors [Pu et al., 2017] [Jarama et al., 2017].

The usual sensors in an airport, especially radars, are typically very expensive to deploy and there may also be some areas without coverage. Therefore, there are additional technologies that are used as gap fillers and that

are more economical. Examples of these sensors are the cameras or magnetic sensors. Cameras are used to detect the presence of targets and also to obtain additional information, such as aircraft identification (reading it, for example, from the tail of the aircraft) or traffic load statistics [Besada et al., 2005a] [Semertzidis et al., 2010] [García et al., 2005]. Regarding the magnetic detection sensors, they are used because their deployment is very cheap and can be a very functional solution, especially for airports with little traffic [Gao et al., 2009]. Finally, work has also been done to improve the sensors already deployed, such as the improvement of the SMR processing chain [Galati et al., 2010] or the algorithms to calculate the target position of the MLAT sensor [Mantilla-Gaviria et al., 2015].

The following subsections explain the most common sensors available on the airport surface and its surroundings. This sensors provides the main input of the surveillance systems.

1.3.1.1 Surface Movement Radar

Surface Movement Radar (SMR) [Schwab and Rost, 1985] [Herrero et al., 2001] is a primary radar that detects aircraft and vehicles on the airport surface. It is a non-cooperative sensor that detects the targets illuminating them with an electromagnetic wave and receiving back the reflected waves from targets within that area. SMR is normally used as the basis for A-SMGCS design for large airports [ICAO, 2004a] and its main design considerations [Perl, 2006] [Galati et al., 1999] are the following:

1. All targets on the runways and taxiways (movement area) must be reliably detected with a 90% probability even in severe weather situations. Stationary targets have to also be detected leading to the discarding of the use of Moving Target Indicator (MTI) or Doppler processing.
2. Detection volume covers from the ground to about 200 to 300 feet above ground
3. The required update rate should be no less than once per second. Antenna rotation rates of at least 60 rpm are, therefore, required.

4. The object location shall be measured with a precision better than 20m having an antenna main lobe width less than 0.4°

The SMR has three main problems [Galati et al., 2004]. The first one is the control of false alarms, especially in clutter situations. Second, the rain attenuation at highest frequencies. Finally, the plot extraction since aircraft targets have extensions up to 60-70m which, for targets near the radar, result in the detection in dozens or even hundreds of contiguous cells. This problems are being improved using new bands [Mauro et al., 2001] or even eliminating rotation of an antenna but using instead several non-moving synthetic aperture antennas [Lukin et al., 2009].

To summarize, the main features of this sensor that directly affect the quality of surveillance and safety nets on the airport area are the following:

- It has a typical horizontal range of 6 km and a vertical range from 0 to 100 m.
- Its typical deviation is usually of the order of 5 m for the range measurement and 0.02° for the azimuth measurement.
- It only provides synthetic vertical measurements such as the ground bit.
- Its update period is usually 1 second.
- It does not provide velocity measurements.

1.3.1.2 Secondary Surveillance Radar

The Secondary Surveillance Radar (SSR) [Skolnik, 1970] [Stevens, 1988] is a radar system that emits encoded pulses (interrogations) through a rotating directional antenna. Aircraft, upon receiving the signal, reply by emitting their identification or barometric altitude. Hence, while primary radars work with radio frequency reflections, the SSR needs active replies which make it a cooperative sensor. Therefore, SSR is able to obtain not only the target's position but also its identifier and barometric height.

The SSR interrogation signals are formed by encoded sequences of pulses standardized by the ICAO. Three different types of interrogations, or modes, are harnessed, mainly for civil use: Mode A, Mode C and Mode S. Mode A and Mode C use a coded sequence of pulses to interrogate the aircraft and their responses contain its identification and barometric height, respectively. The Mode S [Trim, 1990] [EUROCONTROL, 2003] allows to selectively set up a data link with an aircraft, improving the quality and integrity of the detection. Mode S has two operational modes, Elementary Surveillance (ELS) and Enhanced Surveillance (EHS), that differ in the content of the uplink and downlink data. The ELS mode reports the altitude with higher precision (25ft interval) while the EHS also reports additional data: track angle rate, ground speed, Indicated Air Speed (IAS), etc. Their availability is subject to the version of the registers. This data link also allows to establish a communication that can be used by other systems, such as the Traffic alert and Collision Avoidance System (TCAS). This will be seen in greater depth in chapter 6.

The SSR has numerous advantages such as the ability to obtain the target's identifier as well as its increased effective range due to the use of the response of the interrogations instead of the echo of the signal. Amongst the SSR deficiencies are the False Replies Unsynchronised to Interrogator Transmission (FRUIT) and the Garble [Harris, 1956]. Mode S corrects almost all these deficiencies but nonetheless, overlaps may also occur in this protocol [Petrochilos and van der Veen, 2004].

To summarize, the main features of this sensor when used in Terminal Manoeuvring Area (TMA) that directly affect the quality of surveillance and safety nets on the airport area are the following:

- It has a typical horizontal range of 100 km and a vertical range from 50 to 5000 m.
- Its typical deviation is usually of the order of 20 m for the range measurement and 0.09° for the azimuth measurement.
- It provides barometric vertical measurements.
- Its update period is between 4 seconds.

- It can provide horizontal velocity measurements but only if the Mode S is being used.

1.3.1.3 Multilateration

Multilateration (MLAT) [Galati et al., 2008] [Xu et al., 2010] is a cooperative technology used to accurately locate aircraft by using Time Difference of Arrival (TDOA). MLAT is formed by several receiving stations deployed throughout the airport that receive at different times the aircraft's responses to sensor interrogations. With this response delay the 3D position of the aircraft can be obtained. The listened replies can come from Mode A, C and S transponders as well as ADS-B transponders. The main MLAT advantages are its accuracy, its high update rate, and its low cost to install and maintain.

Figure 1.4: MLAT system

There also exists another related technology called Wide Area Multilateration (WAM) [Abbud et al., 2011] [Galati et al., 2005] that extends the use of MLAT technology to TMA and En-route surveillance. The higher coverage is achieved by putting the receiving stations much further apart, up to 100 km between each other.

To summarize, the main features of this sensor that directly affect the quality of surveillance and safety nets on airport area are the following:

- It has a vertical range from 0 to 100 m. The horizontal range depends on each particular deployment of receiving stations.
- Its typical deviation is usually of the order of 5 m for both Cartesian coordinates.
- It provides geometric vertical measurements.
- Its update period is usually less than 1 second.
- It does not provide horizontal velocity measurements.

1.3.1.4 Automatic Dependent Surveillance - Broadcast

ADS-B [Hicok and Lee, 1998] [Donohue, 1995] is a cooperative surveillance technology that allows an aircraft to emit its identity, position and other information to other aircraft or ground stations over a data link. The position is obtained in latitude and longitude through a radio navigation system such as Global Navigation Satellite System (GNSS) [Hofmann-Wellenhof et al., 2012].

The sensor is made up of two subsystems, ADS-B Out and ADS-B In. The first one is responsible for transmitting information and the second is responsible for collecting this information. ADS-B out periodically transmits information about the flight, such as identification, current position, altitude and speed. ADS-B In, also installed on aircraft, has as its main function to receive and interpret other aircraft ADS-B Out.

It should be noted that ADS-B technology generates a network of sensors composed of each navigation system. This means that there is a great variety of sensors, each one with different features: magnitude of the error, time correlation, etc. For this reason, ADS-B usually characterizes the quality of its measures using a series of parameters called Figure of Merit (FOM). From the horizontal FOM, the ADS-B derives parameters for the position accuracy

Figure 1.5: ADS-B system

(Navigation Accuracy Category for Position (NACp)) and for the horizontal speed (Navigation Accuracy Category for Velocity (NACv)) [Hempe, 2010] [Jones, 2009]. The FOM of these parameters reflects the measure estimated Uncertainty with a 95% percentile. The values of this uncertainty can be seen in the tables 1.2 and 1.3.

NACp	95% Estimated Position Uncertainty
6	< 55.6 m
7	< 185.2 m
8	< 92.6 m
9	< 30 m
10	< 10 m
11	< 3 m

Table 1.2: Position accuracy according to NACp

ADS-B is therefore a cooperative sensor, which provides position information in geodetic coordinates and can provide velocity measurements. The measurement error of this sensor is either very small or strongly correlated over time. The time between measurements of this sensor is usually less than one second.

NACv	95% Estimated Velocity Uncertainty
0	> 10 m/s or unknown
1	< 10 m/s
2	< 3 m/s
3	< 1 m/s
4	< 0.02 m/s

Table 1.3: Speed accuracy according to NACv

To summarize, the main features of this sensor that directly affect the quality of surveillance and safety nets on airport area are the following:

- Its coverage is variable due to the technology used.
- Its accuracy depends on the FOM.
- It provides geometric vertical measurements.
- Its update period is usually 1 second.
- It can provide horizontal and vertical velocity measurements.

1.3.2 Control Function

The control function (or safety function as called by EUROCONTROL) is in charge of detecting conflicts and resolving them, and also of providing alerts for incursions in runways, taxiways and critical areas. In addition, it is also in charge of calculating the necessary separation between surface aircraft, taking into account their speed, direction, dimensions, and jet blast effects. Therefore, in order to calculate these conflicts, the control function needs the target positions estimations (tracks) and route plans that are provided by the surveillance and routing functions respectively. EUROCONTROL divides these functions into three: Runway Monitoring and Conflict Alerting (RMCA), Conflicting Air Traffic Control Clearance (CATC), and Conformance Monitoring Alerts for Controllers (CMAC) [EUROCONTROL, 2018].

RMCA is responsible for providing the Short-term Conflict Alert (STCA) tools between an aircraft and other vehicles using the information provided by the surveillance system and certain predefined rules and parameters: runway configurations, meteorological conditions, procedures, etc. In other words, it is in charge of detecting if any pair of vehicles, or their short time prediction (approximately two minutes), does not meet the minimum separation. In-depth work has been done in this field due to its importance and the different approaches available. For example, the prediction of the position of the targets can be made in three different ways. The first one is to make the prediction using only the current speed and heading of the target [Bilimoria et al., 2000]. The second one is based on a worst-case scenario approach and consists of making a prediction based on the assumption that the target will perform any from a range of manoeuvres and if any of these manoeuvres causes a conflict, then an alarm is arised [Liu and Hwang, 2011]. The third one is a mix of the previous two and it consists of calculating a set of possible trajectories, weighting them by their relative probability and then estimating the conflict probability [Prandini et al., 2000].

CATC provides alerts when the controller introduces an electronic clearance that is not allowed from an operational and safety point of view. A clearance is an authorization for an aircraft to proceed under conditions specified by an air traffic control unit. Therefore, the CATC allows to warn about clearances that, were they not to be corrected, could lead to dangerous outcomes. This system is capable of detecting possible runway incursions in advance [Biella et al., 2013]. Although the usual approach is to directly compare the clearances with the surveillance data [Dubuisson et al., 2004], other solutions also add ground routes as an intermediate layer, which simplifies the process [Weiß et al., 2013].

Finally, CMAC provides alerts when a vehicle's movement that is not in accordance with procedures or clearances is detected. Among the most important alerts this system has to arise are the runway incursion, red stop bar crossed or route deviation. In this field almost all the work has been related to the detection of runway incursion since it is the most dangerous conflict as stated by the International Air Transport Association (IATA) [IATA, 2017]. The causes of runway incursions and the impact of new sensors on conflict

detection performance have been analysed in [Schönefeld and Möller, 2012]. In addition, some studies analyse how runway width influences the conflict predictions [Shi et al., 2017]. Also, new conflict prediction algorithms using stochastic linear hybrid system models have been developed [Brinton et al., 2007] [Seah et al., 2010] which, aside to improving performance, also reduces the workload of the controllers.

1.3.3 Routing Function

The routing system, as defined by ICAO, is a system that either manually or automatically has to be able to designate and modify a route for each aircraft or vehicle at the airport to guarantee performance in dense traffic scenarios. In addition, it has to send the routes to the other three main blocks of the A-SMGCS so they can perform their functions. Since it is a system that also may minimize the distance or time to be travelled by the vehicles, it has a mathematical component of optimization. Therefore, as the optimization of this block is an important aspect, it is where the research has been focused in the last few years. For example, new proposals for routing are to use discrete event dynamic systems (DEDS) [Xin-min et al., 2010], mixed-integer linear programming optimization method [Clare and Richards, 2011] [Keith et al., 2008] or A* algorithm [Li et al., 2011].

An example of an implementation of the routing system together with the guidance system can be seen at [Piazza, 2000]. In this implementation, the airport layout is modelled as an N-ary tree and the time it takes to move along the routes is calculated assuming a constant speed movement model and minimizing the time of the chosen route. Subsequently, the rest of the routes are checked to make sure the minimum separation criteria are met and that there is no intersection with any occupied runway.

1.3.4 Guidance Function

The guidance system, as defined by the ICAO, has to provide (through visual indications and controller inputs) indications to authorized vehicles to follow their routes in a clear and precise manner. It must also be able to clearly provide the restricted areas and indicate changes in the routes. To work correctly, it receives all the information from the other three functions of the A-SMGCS since it needs to know the vehicles' position, the possible conflicts, and the routes assigned. These indications are mainly made up with 3 elements: Taxiway Centreline Lights (TCL), stop bars, and Advanced-Visual Guidance Docking Systems (A-VDGS).

TCL individually guide targets who have a cleared route. They illuminate a certain distance in front of the target and they can change dynamically in the event of changes in the route. The automatic version of this concept is also known as Follow-the-Greens (FtG) [Biella et al., 2015] and it is implemented and evaluated at various airports such as Riga International Airport [Ellejmi, 2016]. The stop bars are points where the aircraft must be stopped until they obtain clearances and these bars can be in a runway holding position or across a taxi. The improvement of this system is aimed at automatizing the shutdown of these stop bars [Meier et al., 2005]. Finally, A-VDGS is a visual system which provides assistance when docking to the correct parking position for each aircraft, providing collision avoidance from static objects [Svensson, 2002].

1.4 Thesis Objectives and Structure

This section sets out the objectives of the proposed work. As introduced before, this thesis proposes improvements to some of the functionalities of the A-SMGCS system. Basically, the objectives of the thesis are related to the surveillance and control functions already defined in section 1.3. This can be seen in figure 1.6.

Specifically, concerning the surveillance functionality, the work has been

done on the estimation and correction of biases and on different tracking filters. Regarding to the estimation of biases, the designed system is able to correct the biases associated with different sources. To be more specific, the system can correct biases associated to both centroid difference between each type of sensor and to poor radar calibrations. In addition, an estimator of the offset between the barometric vertical measurements and geometric ones has also been designed.

Regarding the trackers of the surveillance function, two types of tracking systems have been implemented. The first one is a robust tracking system that depends only on the measurements for both the horizontal and the vertical components. The second type is a tracking system that exploits the airport map information, which allows a pre-processing of the measures to greatly improve the quality of the estimations. With these functions, the surveillance is now able to obtain the estimated position of the vehicles on the airport area.

Figure 1.6: Thesis objectives architecture

With regards to the control functionality, work has been done to design and analyse a safety net capable of detecting various conflicts. In particular, a safety net of the following conflicts has been developed: not maintaining

the minimum horizontal separation, exceeding the taxiway speed limits, and runway incursions. The designed safety nets are able to work using only the airport setting and the sensor measures or also exploiting the map information, which improves the safety net performance.

The structure of the thesis reflects the objectives. After this introduction to the problem of traffic control and to the usual sensors on the airport area, chapter 2 proposes to improve the quality of the horizontal tracking of targets on the airport surface by using the map information. In order to be able to use this information, it is proposed to pre-process the measures dividing the problem into its two components: transversal and longitudinal. Therefore, in this chapter first the basic elements to model the airport map as well as the calculations involved in the pre-processing of the measurements are defined. Afterwards, the tracking filter that follows a multi hypothesis approach is defined to monitor all the possible routes of the target at the intersections. Therefore, all the logic of the maintenance of the hypothesis as well as the models used in the filtering are defined. Lastly, the map information is also used to improve the vertical tracking, to be more specific in landing aircraft that follow the glide path.

In chapter 3, filters capable of estimating both the horizontal and vertical positions of aircraft are designed. The proposed filters are robust filters optimized to have good performance in most cases without using any map information. Finally, the performance of these filters are analysed. These robust filters would run in parallel to the map-based filters in order to obtain the estimation when the targets are not following the map.

In chapter 4, the initialization and track maintenance techniques are introduced. Then, a series of estimators that allow to correct the most common biases on the airport area are proposed. Specifically, the corrected bias terms can be divided into sensor related bias, target related bias, or barometric vertical offset. First, sensor related biases are those whose value is the same for all targets as it depends only on the sensor, e.g. antenna deviation. Second, target related biases are those whose value also depends the target, e.g. transponder delay. Lastly, the bias that is estimated and corrected is the barometric offset associated to how the height is obtained by the sensor.

Therefore, in this chapter different models are proposed for each biases as well as its associated estimator.

In chapter 5, the performance of the filters and the biases estimators are discussed. First, the biases estimation are analysed, checking their effectiveness when all the biases are present. Second, the performance of the filters is evaluated, comparing it with other solutions and using both real and simulated data.

In chapter 6, a safety net system capable of detecting a number of different conflicts is proposed. First, the conflict detection of targets that are not maintaining the minimum separation is analysed. In order to do this, the statistical problem behind this conflict is analysed and the threshold that allows to maintain a certain quality in terms of probability of detection and false alarm is calculated. Afterwards, the detection of the runway incursion is analysed obtaining the probability that the target is located in a runway occupied by another vehicle. In addition, the conflict of targets exceeding the ground speed requirements of the route is also analysed. After analysing these three conflicts, it is proposed to exploit the map information, presented in chapter 2, to improve the performance of the safety net system. First, the case that the position of the aircraft is obtained by a tracker that does not use the map information is considered and there is therefore a need to locate the targets on the map. Then the map is used to decrease the probability of false alarm in the minimum separation conflict, calculating the separation taking into account the roads. In addition, the map information is used in the runway incursion conflict detecting targets that skip stop bars, making the runway incursion detected faster and thus improving the safety. Lastly, a series of scenarios are proposed in which all these safety nets and the impact of the use of the map are evaluated.

Finally, the conclusions drawn from this thesis are presented in chapter 7.

Part I

Surveillance Function

Chapter 2

Map-Based Tracking on Airport Area

2.1 Introduction

Tracking of targets following a road network on a 2D surface (i.e. ground) is a classical data-fusion application [Kirubarajan et al., 1998] [Shea et al., 2000] [Blasch et al., 2014]. The use of road geometrical information as contextual data enabling an improvement of the tracking performance has shown its effectivity in practical cases in the literature [Pannetier et al., 2005] [Cheng and Singh, 2007]. Applications of filters using this kind of constraint, among others, are tracking of aircraft taxiing in an airport surface [Herrero et al., 2003], military surveillance of ground targets following a road network [Bar-Shalom et al., 2004] [Kirubarajan et al., 2000] or indoor mobile tracking of users moving along corridors [Yang et al., 2014].

Many different methods for incorporating map constraints to target tracking have, therefore, been explored in the literature. Among them, we could define two distinct approaches:

1. Maintaining a full-state vector filtering, tracking at the same time along-road and transversal to road movement.

2. Decoupling the movement tracking in two components, one associated to along-road movement, and another one related to movement transversal to road (including here the capability to distinguish the road segment where the target lies).

Regarding the first approach, some potential ways to integrate road information in the full tracking filter would be: including heading/track angle pseudo-measures aligned with the road [Oh et al., 2015] [Pyo et al., 2001] [Tahk and Speyer, 1990]; forcing the transition model to follow the track angle [Herrero et al., 2003] [Caicai and Wei, 2011]; modifying the velocity-related state vector components to fulfil the road associated heading constraint [Xinyan and Wei, 2007]; including non-linearities making the state vector compatible with both the position and the heading constraints imposed by the road [Pannetier et al., 2005] [Streller, 2008] [Agate and Sullivan, 2003] [Gattein et al., 2005] [Pannetier et al., 2004] [Xu et al., 2013]; or making input measurements compatible with the road network by projecting them in suitable road segments [Herrero et al., 2003]. In all those approaches, the state vector used for 2D tracking has two positional components, two velocity components and (quite often) two acceleration components. Depending on the application and on the filter mode they can be expressed in different coordinate frames or using the most suitable mathematical representation (i.e. position in Cartesian coordinates, velocity in Cartesian or polar coordinates, acceleration in inertial Cartesian or along-track plus cross-track components, or converted to longitudinal acceleration and turn rate, etc.). Of course, in multiple model filters making use of this approach some of the model tuned filters could have reduced state vectors, without all the actual components of the 2D movement (e.g. constant velocity models without accelerations, stop models without velocity and acceleration, etc.).

The second approach (decoupled longitudinal and transversal tracking) is based on making a definition of the movement process compatible with the road network by design. It is based on the observation that in a road network the usual target movement is no longer an unconstrained 2D movement but can be better understood as a 1D movement along the road, with minor transversal piloting corrections. Those corrections, made by the pilot to keep the target in its lane or in the road define the lateral movement of the

target (to be called lateral drift in this thesis) when it is moving along a road, making the target trajectory almost parallel to the road axis. An example in the literature of this kind of approach would be [Ulmke and Koch, 2006], where a filter somehow similar to our proposal, based on a Variable Structure IMM (VS-IMM) is described. The map definition proposed in this thesis is more general since other representations, such as the described in [Herrero et al., 2003] [Streller, 2008], are not able to model generic curves. Therefore, novel measurement projection processes, based on a maximum likelihood approach, were designed in this thesis. The filter proposed in this thesis uses a road map description based on three different kinds of road segments: straight, circle arcs, and Bézier curves.

An additional problem to be analysed in both approaches is how the proposed filters should work when the target approaches road junctions. Additional uncertainty arises from the fact the output road to be followed by the target is not known by the filter. Therefore, we may make use of two strategies to address this problem: either disabling the non-linearities described previously until the output road segment is clear [Herrero et al., 2001], or branching the filter into different filters adapted to each of the alternative road segments leaving the junction [Streller, 2008]. The first strategy is especially appropriate for full-state filtering approaches, while the second is suited for decoupled longitudinal and transversal tracking. The Multiple Hypothesis Tracker (MHT) procedure has typically used for association [Cuiping et al., 2010] [Thomaidis et al., 2013] [Herrero et al., 2001] but this thesis proposes to use a MHT inspired procedure for following the branching at junctions, to be called Map-Based Multiple Hypothesis Filtering (MBMHF). This approach leads to additional issues related to the hypotheses formation and evaluation processes, including new terms in the evaluation process resulting in improved performance in junctions vicinity, and making it different to the one described in [Ulmke and Koch, 2006]. Also, this method tracks the lateral drift, which is neglected in the described work (and in the rest of the relevant literature).

Summarising, the proposed filter follows the decoupled longitudinal and transversal tracking approach, implementing branching at junctions through an MBMHF approach, along-segment tracking through IMM filters, lateral

drift-tracking through a Kalman filter, and measurement projections into generic curve road axes using a maximum likelihood approach.

The chapter is organised as follows. In section 2.2 the 2D tracking problem for targets following a road network is presented. Then, section 2.3 explains the main components of the tracking system at a high level, and specially how they interact. After that, we follow a bottom-up approach to detail those components. First, the coordinates transformation needed to project the sensor measures in a road segment and to project tracks following the segments back into 2D are explained in section 2.4. Afterwards, section 2.5 presents the along-segment IMM filter, valid for a road without junctions. To continue with the description of the algorithms section 2.6 describes, based on previous explanations, the projection processes in the road network and the MBMHF based branching. Later, the transversal offset tracker, and the associated coordinate transformations are explained in section 2.7. With the previous sections the horizontal map-based filter is fully defined. Lastly, section 2.8 applies the ideas of the map tracking to a special case of vertical filtering, specifically to the landing following the glide path. It should be noted that the analysis of the performance of this approach is in chapter 5.

2.2 Map-Based Tracking Problem Statement

In this section we will describe the main interface and functional requirements of a multisensor tracking system following targets in 2D making use of map of road networks constraining the tracked objects movements. To do so, next we will first detail system inputs and outputs, and road network map modelling.

The system has 2D position measures as inputs. It is assumed that all the available sensors provide full 2D information in Cartesian coordinates, or that those measures are transformed in the Cartesian plane (i.e. from polar measures in the case of radar sensors). It is also assumed that the measures are perfectly associated beforehand target, so that we are not dealing with

the association process. Additionally, the sensor biases are assumed corrected using the techniques to be explained in chapter 4.

The system provides as output the identification of the segment of the road network where the target lies with highest probability (and the associated probability, as assessed by the system), and its position and velocity in Cartesian coordinates.

A key aspect of the system is the way to describe the kinematic constraints imposed by the presence of the road network. To describe those constraints, we will start by describing the model of the network map. Any road-network map, whether it belongs to an airport or a highway network, or even a plan describing a set of corridors within a building, can be parameterised with simpler elements which allow the description of different parts of the map. In particular, in our approach, each road segment may be represented with three bidimensional basic elements: straight lines, circular arcs and splines [Hasberg et al., 2008]. These three kinds of segments are general enough to represent any map with precision. In Figure 2.1 examples of those three elements are drawn. Specifically, the segment going from point A to point B is a straight line, the segment from point B to C is a circular arc, and the segment between B and D is defined through a spline. There are many options for the splines implementation. In this research we have selected the Bézier splines, and specifically cubic Bézier splines [Prautzsch et al., 2013], due to their simplicity.

Figure 2.1: Map example with the three basic elements

Every segment is delimited by an origin node and a destination node, being the algorithms to be described capable of performing tracking in both movement directions (from origin to destination node, or from destination to origin node). In fact, the system output is not affected by the decision of which end of each segment is designated as origin or destination.

Next, we will describe the representation of each of the types of segments:

1. Straight lines: They are represented by an origin and a destination, defined in their Cartesian coordinates.
2. Circular arc: This element allows to represent simple map curves, through the origin point, the destination point, the centre of the curve and (implicitly) the radius.
3. Cubic Bézier Curves: They are used to define more complex map curves interpolating a collection segment samples, being defined in parametric form and smooth C2. Being smooth C2 implies that the derivatives f' and f'' exist and are continuous. A Bézier curve is formed by different consecutive sections, one between each pair of segment samples. In each section, s_i , the parametric form of the curve is as follows:

$$\begin{cases} x = a_{x1,i}\lambda^3 + a_{x2,i}\lambda^2 + a_{x3,i}\lambda + a_{x4,i} \\ y = a_{y1,i}\lambda^3 + a_{y2,i}\lambda^2 + a_{y3,i}\lambda + a_{y4,i} \end{cases} \quad \lambda \in [0, 1] \quad (2.1)$$

where $(a_{xj,i}, a_{yj,i})_{j=1\dots 4}$ are the coefficients associated to the polynomial defining the coordinates (x, y) in the section s_i , while λ is the independent parameter allowing to traverse the section s_i as it changes from 0 to 1. So, for $\lambda = 0$ we have the segment sample at the beginning of the section, and $\lambda = 1$ may be used to obtain the final segment sample of the section. The coefficients have to guarantee the smooth between sections, hence they are chosen carefully, following the procedures described in [Hasberg et al., 2008]. It should be noted that in order to define a point in this kind of segments both the relevant section and the λ parameter for the relevant section need to be identified.

In our conceptual model of the tracking problem we assume the map

model to be complete (it models all relevant real roads, and no segment is unrelated to any real road). Also, the filter assumes the segment-based geometric model of the road is perfect. Note that due to the possibility to link collections of Cubic Bézier Curves, very reduced error is attainable.

2.3 Map-Based Tracking Filter Structure

The filter decouples the tracking in two components, one related to along-road movement, and another one designed to track movement transversal to road. The along-road tracking process assumes that the target moves in parallel to the centre line of the road, focusing in tracking the projection of target position on this centre line. Therefore, it preprocesses the measures by projecting them to this centre line. The transversal movement (perpendicular to this centre line) is estimated by a second part of the system, using as input the difference between the sensor measure and the projected measure.

The 2D position of the target may be represented by three values:

1. A segment identifier (s_{ID}), unambiguously defining the segment where the target position is being projected.
2. The distance along the segment, measured from the segment origin node, to be called d_{long} .
3. The signed transversal position (d_{trans} in the following) allows to completely define the target position. Positive d_{trans} means the offset is to the right of movement direction, while negative d_{trans} means the offset is to the left.

Summarising, as can be seen in Figure 2.3, for two different positions, given the three values (s_{ID1} , d_{long1} , and d_{trans1}) or (s_{ID2} , d_{long2} , and d_{trans2}), the target position is completely defined.

Figure 2.2: Global filter structure

The overall recursive procedure for tracking a target following a road segment is as follows: it receives the unbiased sensor measures (input measures), which are corrected with the estimated transversal offset obtained from the previous measures. These 2D offset-corrected measures would be over the road segment skeleton, if offset terms were perfectly estimated and there was no noise. As there is residual error in transversal offset estimator, and measures also suffer from noise, once the measure is offset-corrected, it gets projected (using a maximum likelihood approach) to the segment skeleton. This 1D projection is used as input measure of the IMM based longitudinal estimator, which tracks the traversed distance in this segment, and also its 1D velocity. An alternative to the use of an IMM estimator could be to use a VS-IMM but in this case it has been discarded since there are only three modes involved (low computational cost) and the projection of the measures, the use of the map, and the target dynamics is more related with an IMM. In parallel to the longitudinal tracking, the lateral drift estimation is updated through a 1D Kalman filter whose input is the distance between the input measure and the road segment. Finally, both the updated lateral drift and the tracked longitudinal position are combined to obtain the final target kinematic state (position and velocity), taking into account the segment specification. This flow can be summarised in figure 2.2. Relating it with figure 2.3, the Longitudinal Tracking Filter tracks both d_{long} and its associated velocity, while the Transversal Filter tracks d_{trans} , using the target heading obtained by the longitudinal filter. Both filters are based on prediction models to be described in detail in sections 2.5 and 2.7.

Figure 2.3: Target position representation

To complete the segment-adapted filtering description, a measurement error model must be established. The 2D measurement error is modelled as bias free, white, and following a bivariate normal distribution with potentially non constant covariance R_k (being k the number of measure). We also assume that the system has perfect measurement-to-track association.

The filter, as described up to now, would be able to track targets following an only road segment or an only road composed of several segments. To extend it to tracking on a road network we should not only know (d_{long} and d_{trans}), but also s_{ID} . The problem appears near junctions, due to noisy measures and tracking, it is difficult quite often to make a decision on which is the value of s_{ID} . Our approach to do this is to maintain a list of the s_{ID} values in the network with an assessment of the associated probability of each s_{ID} . To keep the problem tractable, only the segments whose probability is above a certain threshold, P_{Hmin} , are kept in this collection. Those segments would be the ones in the vicinity of the last few measures, and compatible with preceding target movement. Therefore, using an Multiple Hypothesis Tracker (MHT) [Blackman, 1986] inspired approach (the Map-Based Multiple Hypothesis Filtering (MBMHF)) approach introduced in section 2.1), this list is updated along time with new segment hypotheses as the target arrives to junctions. Their relative probabilities are updated as new evidence is obtained based on the compatibility of measures with expected target movements, and finally the hypothesis list is pruned as the target leaves the junction and the segment followed by the target (and therefore its s_{ID}) becomes unambiguous. Then, a hypothesis ($H(j)$, with index j) is composed of four elements:

1. A segment identifier ($s_{ID}(j)$), specifying the segment ($S(j)$) where the target is assumed to be.
2. Another segment identifier ($s_{prevID}(j)$), specifying the segment ($S(j)$) where the target was before (the previous segment). The reason for including this second identifier has to do with dealing with ambiguities in the vicinity of segments junctions, as will be described in section 2.6.
3. A 1D IMM based segment-adapted filter ($F(j)$), tracking d_{long} , its velocity and acceleration.
4. The probability of the hypothesis ($P(j)$).

After the definition of what we understand as a hypothesis, the general functional structure for the Longitudinal Tracking Filter will be clarified through the description of the items shown in the figure 2.4. The longitudinal filter makes use of a data base containing all the necessary information about the road segments (segment type and parameters), used for locating and projecting the measures. The filter, following the MBMHF approach, keeps the list of hypotheses. After receiving a new measure, the filter projects it in the skeleton of all the segments in the vicinity of the segments included in collection of hypotheses. This projection is done in the functional blocks called Projection 2D to Longitudinal($s_{ID}(j)$). Then, the projected measure is processed by the segment-adapted filter ($F(j)$), called Longitudinal IMM($s_{ID}(j)$). The MBMHF has a key element, called in the figure Hypotheses Management, which manages all the hypotheses associated to the possible trajectories, creating new hypotheses as the target arrives to junctions, calculating the probability of each hypothesis, and pruning the collection of hypotheses (and the associated segment-adapted filters) as the probability of a given hypothesis becomes lower than P_{Hmin} . To calculate the probability of each hypothesis, it makes use of the kinematic state in all IMM filters in the hypotheses list ($\{H(j)\}$), the map and the measure, assessing its compatibility. IMM filters outputs are then projected back to 2D using segment geometric information (block Projection Longitudinal to 2D($s_{ID}(j)$) in the figure), and then they are combined using its probabilities. Section 2.6 describes this process in further detail.

Figure 2.4: Longitudinal estimator diagram

Lateral drift tracking is much simpler. It should be noted this estimation is only performed when there is no ambiguity on the segment being followed ($\{H(1)\}$ is the only hypothesis in the list after pruning). As summarised in figure 2.5, it is separated in three consecutive processes. The first one is projecting the 2D measure in the transversal-to-segment direction, through the functional block Projection 2D to Transversal ($s_{ID}(1)$), where 1 is the index of the only hypothesis. This projected measure of d_{trans} is used as the input for a Kalman filter in charge of estimating the piloting-induced drift. Then, the filtered d_{trans} estimate is projected back to 2D, using functional block Projection Transversal to 2D($s_{ID}(1)$). Additional details of the projections and the Kalman filter are given in section 2.7.

Figure 2.5: Transversal estimator diagram

Finally, note that after Longitudinal and Transversal trackers obtain their filtered estimates, both are transformed into 2D. Then, the output kinematic state can be obtained through the following additive process:

1. Position estimation is obtained through direct summation of 2D position estimates of both trackers.
2. Velocity estimation is directly Longitudinal tracker filtered estimate transformed to 2D, as there is no velocity component in the lateral tracker.
3. The output covariance (maybe needed for other outer algorithms, such as outlier detection or association) would be the addition of the longitudinal tracker covariance and the lateral tracker covariance (for 2D position). The lateral tracker does not add any covariance to velocity terms of the estimation.

2.4 Coordinate Projection Methods

In this section we are describing the different projection methods used in all parts of our filter. It should be emphasised not only position (and sometimes also velocity) should be projected, but also (measure or filter-calculated) covariances matrices should be projected, as our Bayesian filter approach needs them. It should be noted there are many different coordinate transformations, as there are three types of segments (straight lines, circular arcs and splines) and all the different types of projections needed: Projection from 2D to Longitudinal($s_{ID}(j)$) and Projection Longitudinal ($s_{ID}(j)$) Kinetic State for longitudinal tracking; and Projection from 2D to Transversal ($s_{ID}(j)$) and Projection Transversal to 2D ($s_{ID}(j)$) for lateral drift tracking. In the next subsections we will cover all those coordinate transformations in a systematic way.

2.4.1 Projection from 2D to Longitudinal ($s_{ID}(m)$)

In order to obtain the longitudinal projection from the 2D Cartesian Coordinates, we have implemented a Maximum Likelihood Estimator (MLE). In our model, the measurement is modelled as:

$$\vec{z}_m = \vec{x}_r + \vec{\epsilon} \quad (2.2)$$

where $\vec{z}_m = [x_m \ y_m]^T$ is the 2D measure, $\vec{x}_r = [x \ y]^T$ is the 2D real position and $\vec{\epsilon} = [\epsilon_x \ \epsilon_y]^T$ is the 2D sensor measurement error, assumed to be zero-mean, Gaussian and time-uncorrelated. To obtain the MLE estimator, we constrain the possible longitudinal position of the target to the interest segment ($S(j)$, whose identifier is $s_{ID}(j)$), and therefore the 2D position results:

$$\begin{aligned} \hat{x}_{MLE} = \operatorname{argmax}_{\vec{x}_r \in S(j)} p(\vec{z}_m / \vec{x}_r) = \operatorname{argmax}_{(x,y) \in S(j)} \frac{1}{2\pi |R_m|^{1/2}} \\ \cdot \exp \left(-\frac{1}{2} \begin{bmatrix} x_m - x \\ y_m - y \end{bmatrix}^T \cdot R_m^{-1} \cdot \begin{bmatrix} x_m - x \\ y_m - y \end{bmatrix} \right) \end{aligned} \quad (2.3)$$

where R_m is the sensor covariance, whose terms are:

$$R_m = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{pmatrix} \quad (2.4)$$

This problem is equivalent to minimise the Mahalanobis distance between the measure and the projection:

$$\hat{x}_{MLE} = \operatorname{argmin}_{(x,y) \in S(j)} \begin{bmatrix} x_m - x \\ y_m - y \end{bmatrix}^T \cdot R_m^{-1} \cdot \begin{bmatrix} x_m - x \\ y_m - y \end{bmatrix} \quad (2.5)$$

Now, we have to particularise this expression to the three different segments: straight lines, circular arcs and splines. We do so by finding a suitable parametric definition of the segment and solving the aforementioned minimisation problem. Then, this projection has to be transformed into a distance to feed the filters, and also the measurement covariance need to be projected to obtain the distance measurement variance. In next subsections we will do this process for the different types of segments.

Finally, another interesting parameter, common to the three kinds of

segments, is the likelihood of the projection. It can be calculated as follows:

$$\lambda_{proj} = \frac{1}{2\pi|R_m|^{1/2}} \exp\left(-\frac{1}{2} \begin{bmatrix} x_m - x_{proj} \\ y_m - y_{proj} \end{bmatrix}^T \cdot R_m^{-1} \cdot \begin{bmatrix} x_m - x_{proj} \\ y_m - y_{proj} \end{bmatrix}\right) \quad (2.6)$$

where (x_{proj}, y_{proj}) is the 2D result of the projection (in other words, the constrained MLE). This likelihood will be use in the calculation of the probability of each hypothesis in the MBMHF, as it is a probabilistic measure of the spatial compatibility between the 2D measurement and the associated segment.

2.4.1.1 Straight segments

In the case of the straight lines, the expression of the two coordinates of line can be written in term of an only parameter (x) as:

$$x = x \ ; \quad y = m \cdot (x - x_0) + y_0 \quad (2.7)$$

where (x_0, y_0) is the initial point of the segment, and m is the slope. Substituting (2.7) in (2.5), we have the following equivalent 1D minimisation problem to find the X component of the MLE:

$$x_{proj} = \underset{x}{\operatorname{argmin}} \left[\begin{bmatrix} x - x_m \\ m(x - x_0) + y_0 - y_m \end{bmatrix}^T \cdot R_m^{-1} \cdot \begin{bmatrix} x - x_m \\ m(x - x_0) + y_0 - y_m \end{bmatrix} \right] \quad (2.8)$$

Finding the minimum with respect to x, as the zero of the derivative of this function, it can be obtained:

$$x_{proj} = \frac{\sigma_y^2 x_m + \sigma_{xy}(y_0 - y_m - m(x_0 + x_m)) + \sigma_x^2(m(y_m - y_0) + m^2 x_0)}{\sigma_y^2 - 2\sigma_{xy}m + \sigma_x^2 m^2} \quad (2.9)$$

Then, the 2D MLE results:

$$\hat{x}_{MLE} = \begin{bmatrix} x_{proj} \\ m(x_{proj} - x_0) + y_0 \end{bmatrix} \quad (2.10)$$

Finally, we have to transform the projected Cartesian coordinates into longitudinal ones. In order to do this, the distance from the beginning of the segment, which is the projected longitudinal distance, can be calculated with the following expression:

$$d_{long} = (x_{proj} - x_0)\sqrt{1 + m^2} \quad (2.11)$$

The previous equation can be rewritten applying (2.9) and collecting terms, as:

$$d_{long} = \left(\frac{(\sigma_y^2 - m\sigma_{xy})(x_m - x_0) + (m\sigma_x^2 - \sigma_{xy})(y_m - y_0)}{\sigma_y^2 - 2m\sigma_{xy} + m^2\sigma_x^2} \right) \sqrt{1 + m^2} \quad (2.12)$$

For the filter, it is also needed to calculate the longitudinal covariance of the projected longitudinal measure. As it is shown in (2.12) the projection from 2D measure to 1D projected longitudinal distance is a linear transformation, and therefore the exact longitudinal covariance can be obtained. After some algebra, it results:

$$\sigma_{long}^2 = \frac{(m^2 + 1)(\sigma_x^2\sigma_y^2 - \sigma_{xy}^2)}{\sigma_y^2 - 2m\sigma_{xy} + m^2\sigma_x^2} \quad (2.13)$$

2.4.1.2 Circular arcs

Next the projection to the circular arc segment will be explained. Equivalently to the projection to a straight line, to project to a circular arc the Mahalanobis distance between the measure and the projection has to be minimised (according to (2.5)). The position in polar coordinates in a circular

arc is given by:

$$x = x_c + R_c \cdot \cos(\varphi) \quad ; \quad y = y_c + R_c \cdot \sin(\varphi) \quad (2.14)$$

where (x_c, y_c) is the centre of the curve, R_c is its radius and φ is the angle of the position with respect to X axis. Applying the previous equation in (2.5) we obtain the 1D minimisation problem:

$$\varphi_{proj} = \arg \min_{\varphi} \begin{bmatrix} x_c + R_c \cos \varphi - x_m \\ y_c + R_c \sin \varphi - y_m \end{bmatrix}^T R_m^{-1} \begin{bmatrix} x_c + R_c \cos \varphi - x_m \\ y_c + R_c \sin \varphi - y_m \end{bmatrix} \quad (2.15)$$

First, calculating the derivative of the function to be minimised with respect to φ , and equating to zero we obtain the following implicit definition of φ_{proj} :

$$A \cos \varphi_{proj} + B \sin \varphi_{proj} + C \cos \varphi_{proj} \sin \varphi_{proj} + D(\cos^2 \varphi_{proj} - \sin^2 \varphi_{proj}) = 0 \quad (2.16)$$

where:

$$\begin{aligned} A &= (y_c - y_m)P_{2,2} + (x_c - x_m)P_{1,2} \\ B &= -((x_c - x_m)P_{1,1} + (y_c - y_m)P_{1,2}) \\ C &= R_c(P_{2,2} - P_{1,1}) \\ D &= R_c P_{1,2} \end{aligned} \quad (2.17)$$

with:

$$P = \begin{pmatrix} P_{1,1} & P_{1,2} \\ P_{2,1} & P_{2,2} \end{pmatrix} = R_{med}^{-1} \quad (2.18)$$

Defining the sine in function of the cosine and substituting $\beta = \cos(\varphi_{proj})$ we obtain:

$$A\beta + B\sqrt{1 - \beta^2} + C\beta\sqrt{1 - \beta^2} + D(2\beta^2 - 1) = 0 \quad (2.19)$$

Regrouping:

$$A\beta + D(2\beta^2 - 1) = -\sqrt{1 - \beta^2}(B + C\beta) \quad (2.20)$$

Raising both sides of the equation to the square, and grouping terms we obtain a fourth degree polynomial:

$$a\beta^4 + b\beta^3 + c\beta^2 + d\beta + e = 0 \quad (2.21)$$

where

$$\begin{aligned} a &= C^2 + 4D^2 \\ b &= 4AD + 2BC \\ c &= A^2 - 4D^2 - C^2 + B^2 \\ d &= -2(AD + BC) \\ e &= D^2 - B^2 \end{aligned} \quad (2.22)$$

Resolving (2.21) and applying $\varphi_{proj} = \arccos(\beta)$, eight different possible solutions can be found. The solution is chosen by discarding the solutions not in the range of φ in the circular arc segment, and also by calculating the Mahalanobis distance for each solution (after converting the solution to Cartesian in 2D, calculating (x_{proj}, y_{proj}) from φ_{proj} using equation (2.14)), and selecting the minimum one. To translate the projected position angle into longitudinal, first the angle difference $\Delta\varphi$ with the start of the circular segment is calculated as:

$$\Delta\varphi = \varphi_{proj} - \varphi_0 \quad (2.23)$$

where φ_0 is the the angle of the origin node of the curve:

$$\varphi_0 = \text{atan2}(y_0 - y_c, x_0 - x_c) \quad (2.24)$$

It should be noted that, depending on the rotation direction, the absolute value in (2.23) will keep the angle sign (counter-clockwise) or change it (clockwise). Of course, angle difference is protected against angle definitions discontinuities. Afterwards, this angular difference is translated into distance (with sign) as:

$$d_{long} = R_c \cdot \Delta\varphi \quad (2.25)$$

In this case, the longitudinal covariance is obtained in the tangent of the projected position, in the same way that the previously explained for the straight line case, hence it is calculated with (2.13) and taking into account that the slope of the tangent is given by:

$$m = \frac{-1}{\tan(\varphi_{proj})} \quad (2.26)$$

2.4.1.3 Bézier curves

Finally, the projection to a Bézier curve will be explained. It should be remembered this kind of segments is defined as a collection of consecutive sections, each one defined using a parametric form described in (2.1). To obtain the projection into this kind of segment, projections on all sections are calculated, and then the one with minimum Mahalanobis distance is selected.

To calculate the projection on a given section (s_i), we use the parametric form of the curve according to (2.1) in (2.5), and therefore we obtain the minimisation problem, where the segment section subindex i has been discarded to reduce equations complexity.

$$res_{bez} = \begin{pmatrix} a_{x1}\lambda^3 + a_{x2}\lambda^2 + a_{x3}\lambda + a_{x4} - x_m \\ a_{y1}\lambda^3 + a_{y2}\lambda^2 + a_{y3}\lambda + a_{y4} - y_m \end{pmatrix}^T \quad (2.27)$$

$$\lambda_{proj} = \arg \min_{\lambda} (res_{bez} \cdot R_m^{-1} \cdot res_{bez}^T) \quad (2.28)$$

Calculating the derivative and equating to zero the function to be minimized, a quintic function in λ , is obtained.

$$A\lambda_{proj}^5 + B\lambda_{proj}^4 + C\lambda_{proj}^3 + D\lambda_{proj}^2 + E\lambda_{proj} + F = 0 \quad (2.29)$$

with:

$$\begin{aligned} A &= 6(a_{x1}^2\sigma_x^2 + a_{y1}^2\sigma_y^2 + 2a_{x1}a_{y1}\sigma_{xy}) \\ B &= 10(a_{x1}a_{x2}\sigma_x^2 + (a_{x1}a_{y1} + a_{x2}a_{y1})\sigma_{xy} + a_{y1}^2\sigma_y^2) \\ C &= 4(2a_{x1}a_{x3} + a_{x2}^2)\sigma_x^2 + 8(a_{x1}a_{y1} + a_{x2}a_{y1} + a_{x3}a_{y1})\sigma_{xy} + 12a_{y1}^2\sigma_y^2 \\ D &= 6(a_{x1}a_{x4} - a_{x1}x_m + a_{x2}a_{x3})\sigma_x^2 + 6(2a_{y1}^2 - a_{y1}y_m)\sigma_y^2 \\ &\quad + 6(a_{x1}a_{y1} - a_{x1}y_m + a_{x2}a_{y1} + a_{x3}a_{y1} + a_{x4}a_{y1} - a_{y1}x_m)\sigma_{xy} \\ E &= 2(2a_{x2}a_{x4} - 2a_{x2}x_m + a_{x3}^2)\sigma_x^2 + 2(3a_{y1}^2 - 2a_{y1}y_m)\sigma_y^2 \\ &\quad + 4(a_{x2}a_{y1} - a_{x2}y_m + a_{x3}a_{y1} + a_{x4}a_{y1} - a_{y1}x_m)\sigma_{xy} \\ F &= 2(a_{x3}a_{x4} - a_{x3}x_m)\sigma_x^2 + 2(a_{x3}a_{y1} - a_{x3}y_m + a_{x4}a_{y1} - a_{y1}x_m)\sigma_{xy} \\ &\quad + 2(a_{y1}^2 - a_{y1}y_m)\sigma_y^2 \end{aligned} \quad (2.30)$$

It can be solved using numerical methods, such as the Newton's method. As in each section λ_{proj} must be between 0 and 1, the projection is passed through a non linearity to ensure the projection is compatible with segment definition:

$$\lambda_{proj} = \begin{cases} 0 & \lambda_{proj} < 0 \\ 1 & \lambda_{proj} > 1 \\ \lambda_{proj} & otherwise \end{cases} \quad (2.31)$$

Afterwards, the λ_{proj} parameter is introduced in equation (2.1) for each section of the Bézier curve, to get the 2D projection in each of the sections defining the segment $(x_{proj,i}, y_{proj,i})$. In order to identify the projection in the whole segment, the projection $(x_{proj,i}, y_{proj,i})$ with lowest Mahalanobis distance to the measure (x_m, y_m) is selected. This allows to identify both the section s_i and the λ_{proj} univocally defining the 2D position of the projection in the whole Bézier curve segment (x_{proj}, y_{proj}) .

The next step is to transform the projected λ_{proj} into longitudinal coordinates (along segment distances). Using Composite Simpson's rule, the distance in the section s_i would result:

$$d_{long,i}(\lambda_{proj}) = \int_0^{\lambda_{proj}} \sqrt{\left(\frac{\partial x}{\partial \lambda}\right)^2 + \left(\frac{\partial y}{\partial \lambda}\right)^2} d\lambda \quad (2.32)$$

This approach demands resolving a numerical integration for each measure projection, due to the lack of a simple expression that relates distance with λ , resulting in high computational load. To reduce this load our proposal is sampling equation (2.32) offline, and approximating it with a polynomial. We use a polynomial curve fitting of degree 4, which our tests have shown be capable of providing a good enough fit of the aforementioned equation. So we have:

$$d_{long,i}(\lambda_{proj}) \approx p_{1,i}\lambda_{proj}^4 + p_{2,i}\lambda_{proj}^3 + p_{3,i}\lambda_{proj}^2 + p_{4,i}\lambda_{proj} + p_{5,i} \quad (2.33)$$

As the longitudinal distance is zero when λ is zero, the coefficient $p_{5,i}$ has to be zero. The full distance from the beginning of the Bézier curve can be obtained adding the length of the previous sections (up to a total of $i-1$). As these sections whole length can be calculated from their respective distance interpolation function (equation (2.33)), for $\lambda = 1$, the complete distance can be calculated as:

$$d_{long} = \sum_{k=1}^{i-1} d_{long,k}(1) + d_{long,i}(\lambda_{proj}) \quad (2.34)$$

The longitudinal covariance of the projection is obtained in the tangent of the projected position, in a similar way that the previously explained for the straight line case, hence it is calculated again using (2.13), knowing that the slope of the point is given by:

$$m = \frac{\partial y}{\partial x} = \frac{\partial y}{\partial \lambda} \cdot \frac{\partial x^{-1}}{\partial \lambda} \quad (2.35)$$

From (2.1), taking derivatives, and using the parameters of the section where the projection lies, we have:

$$\begin{aligned}\frac{\partial x}{\partial \lambda} &= 3a_{x1,i}\lambda^2 + 2a_{x2,i}\lambda + a_{x3,i} \\ \frac{\partial y}{\partial \lambda} &= 3a_{y1,i}\lambda^2 + 2a_{y2,i}\lambda + a_{y3,i}\end{aligned}\quad (2.36)$$

To calculate the slope in the projection position we substitute $\lambda = \lambda_{proj}$ in previous equations, resulting:

$$m = \frac{3a_{y1,i}\lambda_{proj}^2 + 2a_{y2,i}\lambda_{proj} + a_{y3,i}}{3a_{x1,i}\lambda_{proj}^2 + 2a_{x2,i}\lambda_{proj} + a_{x3,i}} \quad (2.37)$$

2.4.2 Projection from Longitudinal ($s_{ID}(m)$) to Kinetic state

The different along segment tracking filters estimate target position and velocity in terms of the along-segment tracking estimation vector. This 2D vector contains the distance along the segment (d_{long}), and its associated time derivative; the longitudinal velocity ($v_d = \frac{dd_{long}}{dt}$). The along-segment tracking estimation vector ($\vec{d}$) then results:

$$\vec{d} = [d_{long} \quad v_d]^T \quad (2.38)$$

From this data, the process to obtain the full kinetic state ($\vec{\Theta}$) of a target (i.e. predicted or filtered estimates) will be described next. This process depends on the type of segment, although the output kinematic state of the process always contains the same components: X and Y Cartesian position, ground speed v_{gs} and target heading (θ , defined as angle with Y axis, in clockwise direction):

$$\vec{\Theta} = \begin{bmatrix} x & y & v_{gs} & \theta \end{bmatrix}^T \quad (2.39)$$

X and Y Cartesian coordinates are calculated from d_{long} and from the parametric definition of the segments, and therefore it is dependent on the

type of segment. Meanwhile, it is clear v_d modulus is directly v_{gs} :

$$v_{gs} = |v_d| \quad (2.40)$$

Regarding target heading θ , the method to actually derive it is slightly different for each kind of segment, and will be described in next subsections.

To assess filter output variance, it is also important to calculate $\vec{\Theta}$ estimation covariance for each along-segment tracking filter. This covariance can be related to the 2x2 covariance matrix of the error of the along segment tracking estimation vector $\vec{d}$, which has the following form:

$$P_{long} = \begin{pmatrix} \sigma_{long}^2 & \sigma_{long-vel} \\ \sigma_{long-vel} & \sigma_{vel}^2 \end{pmatrix} \quad (2.41)$$

The projection from longitudinal to kinematic state can be approximated by a linear transformation, and therefore the approximate covariance of $\vec{\Theta}$ (P_S) can be obtained. After some algebra, it results:

$$P_{\Theta} = \begin{pmatrix} \frac{\sigma_{long}^2}{m_{local}^2+1} & \frac{m_{local}\sigma_{long}^2}{m_{local}^2+1} & \frac{\sigma_{long-vel}}{\sqrt{m_{local}^2+1}} & 0 \\ \frac{m_{local}\sigma_{long}^2}{m_{local}^2+1} & \frac{m_{local}^2\sigma_{long}^2}{m_{local}^2+1} & \frac{m_{local}\sigma_{long-vel}}{\sqrt{m_{local}^2+1}} & 0 \\ \frac{\sigma_{long-vel}}{\sqrt{m_{local}^2+1}} & \frac{m_{local}\sigma_{long-vel}}{\sqrt{m_{local}^2+1}} & \sigma_{vel}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\theta}^2 \end{pmatrix} \quad (2.42)$$

where m_{local} is the slope of the curve and σ_{θ} is a default value for the track angle covariance. It should be noted m_{local} is related with target heading θ as follows:

$$m_{local} = \frac{1}{\tan \theta} \quad (2.43)$$

Again, to fully specify the transformation of both states and covariances, the following procedures need to be defined:

1. How to obtain XY coordinates from d_{long}

2. How to obtain the heading from d_{long} , v_d and curve parameters.

This will be detailed in next subsections for each type of segment.

2.4.2.1 Straight segments

In the case of the straight lines the calculation of X and Y coordinates can be directly made by inverting (2.11), resulting in the following expressions:

$$x = x_0 + \frac{d_{long}}{\sqrt{1 + m^2}} \quad (2.44)$$

$$y = y_0 + \frac{m \cdot d_{long}}{\sqrt{1 + m^2}} \quad (2.45)$$

The heading can be obtained as follows:

$$\theta = \arctan(m^{-1}) + \pi \cdot u(-v_d \cdot m) \quad (2.46)$$

where $u(x)$ is the Heaviside step function. The reason v_d is negative means the movement goes in descending X direction, and therefore a π constant need to be added to the arctangent. It is clear that, in this case, combining equations (2.46) and (2.43) the slope m_{local} remains constant and equal to m .

2.4.2.2 Circular arc

In the case of the circular arc, the measure transformation from longitudinal to Cartesian coordinates is made with the following procedure:

$$\Delta\varphi_{abs} = \frac{d_{long}}{R_c} \quad (2.47)$$

where $\Delta\varphi_{abs}$ is the signed value of the rotation angle difference on the curve. To calculate x and y we have:

$$x = x_c + R_c \cdot \cos(\varphi_0 + \Delta\varphi_{abs}) \quad (2.48)$$

$$y = y_c + R_c \cdot \sin(\varphi_0 + \Delta\varphi_{abs}) \quad (2.49)$$

The heading can be calculated with the following expression:

$$\theta = -(\varphi_0 + \Delta\varphi_{abs}) + \pi \cdot u(-v_d) \quad (2.50)$$

where $u(x)$ is again the Heaviside step function and the term v_d is used to check if the rotation is counter-clockwise rotation or clockwise and changes the heading accordingly.

2.4.2.3 Bézier curve

In the Bézier curve case, the measure transformation from d_{long} to Cartesian Coordinates is made obtaining λ from d_{long} and then substituting the resulting λ in (2.1). The first process has to be done in several steps:

1. Identify the relevant Bézier curve section s_i . This is done by successively comparing d_{long} with the accumulated length of curve sections, and finding the first section whose accumulated length is larger than d_{long} . From the description of (2.34), and remembering the definition of $d_{long,k}(\lambda)$ in (2.33), the accumulated length of i-th curve section (s_i) is:

$$d_{acum}(i) = \sum_{k=1}^i d_{long,k}(1) \quad (2.51)$$

2. Obtain the remaining distance $d_{long,i}(\lambda)$ within this section from d_{long} , subtracting the accumulated length of the previous curve section (s_i):

$$d_{long,i}(\lambda) = d_{long} - d_{acum}(i - 1) \quad (2.52)$$

3. Solve the implicit 4th degree equation relating the previously calculated $d_{long,i}(\lambda)$ and the needed λ :

$$p_{1,i}\lambda^4 + p_{2,i}\lambda^3 + p_{3,i}\lambda^2 + p_{4,i}\lambda + p_{5,i} - d_{long,i}(\lambda) = 0 \quad (2.53)$$

Finally, the resulting λ is used in equation (2.1) to obtain X and Y coordinates.

The heading is obtained using the local slope of the point (see equation (2.36)), resulting:

$$\theta = \text{atan2}(3a_{x1,i}\lambda^2 + 2a_{x2,i}\lambda + a_{x3,i}, 3a_{y1,i}\lambda^2 + 2a_{y2,i}\lambda + a_{y3,i}) + \pi \cdot u(-v_d) \quad (2.54)$$

where $u(x)$ is again the Heaviside step function, used to change heading for movements with decreasing distance (negative v_d).

2.4.3 Projection from 2D to Transversal ($s_{ID}(m)$)

In order to estimate the lateral drift, the sensor measure has to be transformed also to a measure of the distance between the segment s_{ID} and the target (d_{trans}). Again, this calculation is different for different types of segments. Also, for the later filter processing it is important to calculate the associated error covariance (to be called σ_{trans}^2).

To calculate the distance to the segment we approximate it by the distance from the point to the line tangent to the projection calculated in section 2.4.1. The distance sign is positive to the right of target movement direction and is negative to the left. Using the equation of distance between point and segment, and the target heading described in section 2.4.2, it results:

$$d_{trans}(m) = \cos \theta \cdot (x_m - x_{proj}) - \sin \theta \cdot (y_m - y_{proj}) \quad (2.55)$$

It should be noted that this projection uses the target heading θ , and therefore it is made after longitudinal filtering. The index (m) in $d_{trans}(m)$ is used to refer to the measured lateral drift.

With the equation (2.43) that relates θ with m_{local} and assuming the impact of the error in the calculation of m_{local} , x_{proj} and y_{proj} from x_m and y_m in the error of $d_{trans}(m)$ is negligible (which is true for straight segments, and also for curves with not too small local radius), we can derive the transversal covariance as:

$$\sin \theta = \frac{1}{\sqrt{1+m_{local}^2}} \quad \cos \theta = \frac{m_{local}}{\sqrt{1+m_{local}^2}} \quad (2.56)$$

$$\sigma_{trans}^2 = \frac{\sigma_x^2 m_{local}^2 + \sigma_y^2 - 2\sigma_{xy} m_{local}}{m_{local}^2 + 1} \quad (2.57)$$

2.4.4 Projection from Transversal ($s_{ID}(m)$) to 2D offset

To convert lateral drift tracking results into 2D, in order to add the results with those of longitudinal tracking, as depicted in Fig. 2.2, the lateral drift d_{trans} must be projected to Cartesian coordinates, using the same sign convention in section 2.4.3. The calculation is based on calculating the vector in the direction orthogonal to the heading. This heading is calculated following the methods described in section 2.4.2, calculated differently for each type of segment. From this result, the Cartesian 2D offset vector, defined as $[\Delta x, \Delta y]^T$, can be calculated as:

$$\Delta x = d_{trans}(f) \cos \theta \quad (2.58)$$

$$\Delta y = -d_{trans}(f) \sin \theta \quad (2.59)$$

Here, The index (f) in $d_{trans}(f)$ is used to refer to the filtered lateral drift. Again, in order to estimate the overall filter output covariance, it is necessary to define a covariance term (R_{drift}) associated to the Cartesian 2D offset vector. From equations (2.56), (2.58) and (2.59), and assuming lateral drift estimator has an standard deviation equal to σ_{trans} , we get:

$$R_{drift} = \frac{1}{m_{local}^2 + 1} \begin{pmatrix} m_{local}^2 \sigma_{trans}^2 & -m_{local} \sigma_{trans}^2 \\ -m_{local} \sigma_{trans}^2 & \sigma_{trans}^2 \end{pmatrix} \quad (2.60)$$

2.5 Along-segment IMM filter description

After describing the coordinate projections, we will focus in the longitudinal (along-segment) filter design. The longitudinal movement along a given segment is modelled using the following simplified state vector:

$$x_{long} = (d_{long}, v_{long}, a_{long})^T \quad (2.61)$$

where d_{long} is the signed distance measured from the segment origin node (calculated as described in previous section), v_{long} is the signed longitudinal velocity at which the target moves away from the origin node, and a_{long} is the signed acceleration of the target in this direction. To track the movement along each segment (associated to an hypothesis within the MBMHF) is estimated by a longitudinal IMM filter [Blom and Bar-Shalom, 1988]. The designed IMM filter uses two Kalman filters defined by two dynamic motion models. The first one is in charge of segments without longitudinal accelerations, so it models a constant velocity movement with the following state variables:

$$x_1 = (d_{long}, v_{long})^T \quad (2.62)$$

The second mode assumes constant longitudinal acceleration, and therefore has the following state variables:

$$x_2 = (d_{long}, v_{long}, a_{long})^T \quad (2.63)$$

This second filter also has higher process noise in order to be able to follow not fully modelled manoeuvres, using plant noise model as the indicated by the following equation:

$$Q_{KA} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_{KA}^2 T^2 \end{pmatrix} \quad (2.64)$$

where σ_{KA} is the comfort standard deviation. This plant noise is derived from Singer model assuming that the target suffer small random perturbations [Bar-Shalom et al., 2011]. The parameters used to model the transition probabilities are those proposed in for a two mode structure [Kastella and Biscuso, 1995].

The measurements feeding the IMM filter are those obtained, after lateral-drift removal, using the projection processes described in section 2.4.1. In that section also the measurement covariances derivation was described.

The filter tracks three states, but a 2D vector only containing distance and velocity is provided as output and finally transformed to kinetic state coordinates according to the procedures in section 2.4.2. Also, the filter covariance is of size 3x3, but only the 2x2 submatrix related to distance and velocity estimates is provided as output and further processed.

An important parameter to be estimated for the MBMHF is the IMM filter likelihood. This likelihood is a measure of the compatibility of longitudinal projected measures with the prior IMM state. It is proposed to use the likelihood of the constant speed filter as the IMM likelihood. This is because each IMM mode have different dimensionality and therefore their likelihoods cannot be directly compared. The likelihood can be calculated as follows:

$$\lambda_{IMM} = \frac{1}{2\pi|\hat{S}|^{1/2}} \exp\left(-\frac{1}{2} \cdot res^T \cdot \hat{S}^{-1} \cdot res\right) \quad (2.65)$$

where res is the measurement residual and $\hat{S}$ is its covariance.

2.6 Map-Based Multiple Hypothesis Filtering

Once the road map representation, the measure projection process and the along-segment tracker have been described, the longitudinal tracking, denoted by MBMHF, is going to be defined. In a general way the tracking system starts associating the sensor measure with the close segments (process to be outlined in section 2.6.1), and managing the associated tracking hypothesis (as will be described in section 2.6.2). This is followed by projecting the (lateral drift-corrected) measure to each of these segments using the methods described in section 2.4.1. Later, a longitudinal IMM estimator (described in section 2.5) is used to smooth the longitudinal distance measurement. Afterwards, each of the potential hypothesis probabilities are calculated (following the method to be described in section 2.6.3). Finally, each hypothesis is unprojected into Cartesian coordinates (as outlined in section 2.4.2) and they are combined using the hypotheses probabilities, as will be described in section 2.6.4. Also, very low probability hypotheses get pruned. The whole process is summarised in figure 2.6.

2.6.1 Search for close segments

As previously explained, the measures need to be projected to the nearby segments. When the first measure is received the distance to all the segments of the airport is calculated and only if the distance is below a certain threshold the segment is included in the list of potential target localizations. This procedure computational cost can be reduced using a spatial cell ordering of the segments. For later measures, the close segments list is obtained from the one associated to the previous measures. They will either be the same segments or adjacent to the ones in the previous localization process. Again, all segments in the need to be at a distance from the measure lower than a predefined threshold (th_{loc}), higher than the expected measurement error but not too high to keep computational load low.

Figure 2.6: Workflow of the MBMHF

2.6.2 Tracking hypotheses management

The multiple hypotheses management process within the MBMHF is in charge of associating close segments list with previous hypotheses. New hypotheses will be created in ambiguity situations as for example intersections. Each hypothesis (with index j) assumes the target moves in a specific segment so it is formed, as previously described, by: 1) the current segment identifier ($s_{ID}(j)$); 2) the previous segment identifier ($s_{prevID}(j)$); 3) the 1D IMM based segment-adapted filter ($F(j)$); and 4) the probability of the hypothesis ($P(j)$).

In order to clarify the hypotheses list initialization and its management, it is assumed that the MBMHF is initialised at a moment where the localization is unambiguous (close-segment list has only one element), so it is assumed that the target is known to be in a specific segment. This way, it is created the hypothesis associated to that location, with both $s_{ID}(1)$ and $s_{prevID}(1)$ equal to that segment, the $F(1)$ filter initialised with an only measure (assuming velocity and acceleration to be zero, but with extremely high associated variances), and hypothesis probability $P(1)$ equals to 1. Of

course, alternative initialisations are also managed.

With the following projected measures, there are different options:

1. A localization (close-segment derived in 2.6.1, and included in the close-segments list) matches with an existing hypothesis $s_{ID}(j)$ parameter. In this case, the localization is simply associated with the hypothesis, which will be updated with this measurement. So, in this case, $s_{ID}(j)$, $s_{prev_{ID}}(j)$, $F(j)$, and $P(j)$ remain unmodified.
2. The localization matches with the previous segment of an existing hypothesis. It is important to take into account this case in order to prevent situations as the one shown in figure 2.7. In that example, measures 1, 2 and 3 are time ordered, and the target moves quite slowly through segment 1. Because of the sensor noise measure 2 is associated with segment 2 when the target is still moving in the previous segment. Measure 3 projections returns to segment 1 but the hypothesis associated to the segment 1 has already evolved to segment 2. In this case the localization is associated with the hypothesis, but after the projection to $s_{prev_{ID}}(j)$ the resulting d_{long} need to be modified adding/subtracting the length of the implicated segments, making its distance measured from the current hypothesis segment $s_{ID}(j)$ (segment 2 in the example). In this case, $s_{ID}(j)$, $s_{prev_{ID}}(j)$, $F(j)$, and $P(j)$ also remain unmodified. The details of the d_{long} transformation can be found in the figure 2.9.

Figure 2.7: Effect of the sensor noise

3. The localization does not match with any created hypothesis $s_{ID}(j)$ parameter but it is a valid transition from one of them (there is a junction between a previous hypothesis $s_{ID}(j)$ segment and the localization one). In this case a new hypothesis is created with the following parameters: $s_{ID}(j)$ will be the localization close-segment; $s_{prevID}(j)$ will be the previous hypothesis $s_{ID}(j)$ value; $F(j)$ state values are modified to the new segment (d_{long} gets added/subtracted the length of the implicated segments, and the velocity and accelerations may change the sign depending on the orientations of both segments); and $P(j)$ is maintained. The details of the d_{long} transformation can be found in the figure 2.9.
4. Finally, in case that the localization does not match with any of the previously explained cases, the measure is discarded assuming that it is a sensor outlier.

Figure 2.8: Flow diagram of the hypotheses update

The previous logic is summarised in figure 2.8. Now, the transformation that change the distance from one segment to the adjacent is explained. This

transformation can be simplified to an offset, d_{offset} , and a sign change, d_{sign} , as can be seen in equation 2.66.

$$\begin{aligned} d'_{long} &= d_{sign} \cdot d_{long} + d_{offset} \\ v'_{long} &= d_{sign} \cdot v_{long} \\ a'_{long} &= d_{sign} \cdot a_{long} \end{aligned} \quad (2.66)$$

The values of the d_{offset} and d_{sign} depend on the common node of the segments, on the length of each segment and on the sign of that length. Denoting the length of each segment as d_1 and d_2 , the segment origin as o and its destination as d then the values of d_{offset} and d_{sign} are summarized in figure 2.9.

$$\begin{aligned} d_{offset} &= -d_1 \operatorname{sign}(d_1 \cdot d_2) \\ d_{sign} &= \operatorname{sign}(d_1 \cdot d_2) \end{aligned}$$

$$\begin{aligned} d_{offset} &= d_2 + d_1 \operatorname{sign}(d_1 \cdot d_2) \\ d_{sign} &= -\operatorname{sign}(d_1 \cdot d_2) \end{aligned}$$

$$\begin{aligned} d_{offset} &= 0 \\ d_{sign} &= -\operatorname{sign}(d_1 \cdot d_2) \end{aligned}$$

Figure 2.9: Distance and speed transformation depending on the common node

2.6.3 Tracking hypotheses probability calculation

Once the phase of associate locations with hypotheses finishes, the system makes projections to all hypothesis segments, the IMM filters are updated with the new projected measures, and the longitudinal state gets unprojected into kinetic state vector coordinates as described in 2.4.2. After this, the probability of each hypothesis is calculated. This probability is obtained taking into account the following parameters: the hypothesis probability according to prior information ($\hat{P}(j)$) value, the projection likelihood (as described in section 2.4.1) and the IMM likelihood (described in section 2.5). The higher the projection likelihood the most compatible the 2D measure is with the segment, and the higher the IMM likelihood the most consistent the segment filter is with the projected longitudinal measures. Hence, neglecting dependency between those two probabilistic measures, the following expression (obtained from the Bayes' theorem assuming likelihood and prior probabilities independence) is used to update the probabilities:

$$P(j) = \frac{\hat{P}(j) \cdot \lambda_{proj}(j) \cdot \lambda_{IMM}(j)}{\sum_{s=1}^M \hat{P}(s) \cdot \lambda_{proj}(s) \cdot \lambda_{IMM}(s)} \quad (2.67)$$

where $P(j)$ is the posteriori probability, $\lambda_{proj}(j)$ is the projection likelihood (see equation (2.6)), $\lambda_{IMM}(j)$ is the filter likelihood (see equation (2.65)), and M is the number of updated hypotheses.

Before calculating the probabilities according to (2.67), very unlikely pro-

jections are discarded using the following procedure: the calculated likelihood of each projection ($\lambda_{proj}(j)$) is compared to a threshold (calculated using chi-square tables for two degrees of freedom, $\lambda_{proj_{max}}$) in order to discard the not likely ones. If this is the case, the hypothesis would be in the same case as if it had no close-segment associated for the current measurement. Additionally, the associated IMM filter ($F(j)$) would need to go back to the estimates available before the current measure was processed.

If any hypothesis has not been updated with the last measure but it has not been pruned, the projection likelihood is the same as the projection threshold and the filter likelihood is the minimum of the updated hypothesis divided by a constant ($N_{like_{miss}}$).

2.6.4 Tracking hypotheses combination and pruning

After each hypothesis has its probability calculated and the coordinates transformed, the MBMHF combines the unprojected kinetic state vectors for each hypothesis to obtain the final output with a probability-weighted average. If any hypothesis has not been updated with the last measure, its estimation is predicted to the current time assuming constant velocity.

Before this combination, an optional heading protection can be applied for low velocities. As the hypothesis are formed by the current segment location ($s_{ID}(j)$) and the previous one ($s_{prevID}(j)$), the heading can be obtained by the transition sense between them. This protection allows to estimate correctly the heading avoiding jumps of π radians when the target is stopped. It is important to remark that this should be only used in scenarios where the targets are not expected to perform a sharp change of direction without the use of multiple segments, e.g. airports.

Finally, the output kinetic state estimate of the MBMHF would result:

$$\vec{\Theta}_{MBMHF} = \sum_{i=1}^M P(i) \cdot \vec{\Theta}(i) \quad (2.68)$$

where $\vec{\Theta}(i)$ is the estimated kinetic state of the hypothesis with index i ,

calculated as described in section 2.4.2. It should be noted the fourth component of the state is an angle in radians, and therefore this average must be protected against angle discontinuities.

The associated output covariance (P_{MBMHF}) is obtained with the following expression of the covariance of the probabilistic mix of random vectors:

$$P_{MBMHF} = \sum_{i=1}^M P(i) \cdot \left[P_{\Theta}(i) + \left(\vec{\Theta}(i) - \vec{\Theta}_{MBMHF} \right) \left(\vec{\Theta}(i) - \vec{\Theta}_{MBMHF} \right)^T \right] \quad (2.69)$$

where $P_{\Theta}(i)$ is the estimated covariance of the hypothesis i , calculated according to the procedures described in 2.4.2.

Finally, those hypotheses that have not been updated for the last few measures get removed from the hypothesis list (T_{max}).

2.7 Lateral Drift Tracker

As described previously, the system also tracks the transversal movement (lateral drift) following the distance between the target and the centre of the road: as can be seen in Figure 2.10, the humane-piloted (or automatic-piloted) target will, in general, not perfectly follow the road axis.

In this figure the 2D position of the target (x, y) and the 2D measurement vector (x_m, y_m) are depicted. Also, the actual target lateral drift (d_{trans}) is depicted, and also the drift measurement ($d_{trans}(m)$). To obtain the filtered drift estimation ($d_{trans}(f)$) it is assumed drift dynamic behaviour can be modelled as a slowly changing signal. In our implementation this is done through a one dimensional vector state Kalman filter, with an exponential time-correlation, although higher order movement models could be implemented with almost no changes in the rest of the complete filtering architecture. So, this filter state vector is:

Figure 2.10: Lateral drift estimation

$$x_{trans} = (d_{trans}) \quad (2.70)$$

This model can be expressed as follows, where $[k]$ is the time index.

$$d_{trans}[k] = \rho \cdot d_{trans}[k - 1] + w[k] \quad (2.71)$$

In this model, ρ is a constant related to the decorrelation time of the process and measurement time and $w[k]$ is a stochastic discrete white process with normal distribution of zero mean and variance equals to $\sigma_{drift}^2(1 - \rho^2)$. This model assumes that in stationary state d_{trans} has a normal distribution of zero mean and variance equals to σ_{drift}^2 . This estimation is obtained with a Kalman Filter with only one state, following this prediction model and assuming measurement error ($v[k]$) have zero mean and variance equal to σ_{trans}^2 , as calculated according to equation (2.57). The measurement model then results:

$$d_{trans}(m)[k] = d_{trans}[k] + v[k] \quad (2.72)$$

The lateral drift is only calculated in time intervals where the segment

being followed by the target is not ambiguous (MBMHF has only one hypothesis), although it is corrected at all times but at MBMHF input (to reduce the impact of lateral drift on hypothesis formation and management), and after MHMBF, as the drift is part of the actual state of the target. So, once the offset is calculated, the sensor measures are corrected with it ($d_{trans}(f)$ is translated to $(\Delta x, \Delta y)$ and this vector is subtracted from (x_m, y_m) before localization and projection) so the MBMHF localization performance is not deteriorated. Once the longitudinal map-constrained position of target has been estimated by the MBMHF, the 3D drift $(\Delta x, \Delta y)$ is added again so the position estimation (first two elements of the MBMHF state vector $\vec{S}_{MBMHF}$, calculated as described in equation (2.68)) does not get biased in relation to the real target position. Figure 2.11 outlines the described process.

Figure 2.11: Lateral drift correction

2.8 Map-Based Vertical Tracking

So far, this chapter has been related to how the map information can be used to improve the horizontal tracking. Obviously, the vertical movement is not constrained by the road maps but there is a particular case in which it is possible to use a similar idea. Specifically, when aircraft are landing (in final approach) they may have to follow the glide path. When this happens, the planes descend with a fixed slope, usually 3° respect the horizontal speed. The glide path is indicated with the Instrument Landing System (ILS) [ICAO, 1996], which is a system that provides lateral and vertical guidance to maintain this rate of descent. This information can be incorporated into the vertical tracking system to improve the estimation which may be interesting as it does not involve much additional computationally cost. The glide slope inclination value and its starting position varies for

each airport and can be extracted from the airport approach charts as shown in figure 2.12.

Before discussing how this information can be exploited, it should be noted that the vertical measures have an offset caused by the use of geometric and barometric measures. This offset has to be corrected before applying the glide slope information. An estimator for this offset is discussed in section 4.3.2.

Figure 2.12: Heathrow Instrument Approach Chart Detail

The glide path information can be added to the estimator in three different ways:

1. Vertical speed pseudomeasures
2. Height pseudomeasures
3. Projection to glide slope and vertical drift estimation

The first approach generates vertical speed pseudomeasures that are added to the height measure and then processed by an IMM filter. The vertical speed would be obtained from the horizontal speed, estimated by the horizontal filter, multiplying it by the glide slope. This technique has the advantage that it is simple but the calculation of the pseudomeasure noise covariance has to be done carefully because it is correlated as it is a filter output. This problem can be alleviated by adding an additional term to the covariance.

The second option is to directly add height pseudo measures. These pseudomeasures would be calculated in a similar manner to the ones associated with vertical speed, that is, taking into account the slope between vertical and horizontal movement. This option also presents the problem of the correlated covariance and can also bias the estimation since the aircraft descend with certain inclination but not all of them follow exactly the path from the ILS point.

The third option projects the measures to the glide path and estimates the drift between the ILS glide path and the glide path followed by the target. This approach would be very similar to the one selected to the MBMHF.

For simplicity it has been chosen the solution of vertical speed pseudo measures. The architecture of this system is the same as a classic tracking filter but, when a height measurement is received, it is checked if the target trajectory is compatible with the glide path. This is done by confirming whether it is compatible both horizontally and vertically. The horizontal verification consists in checking if the horizontal position obtained by some horizontal tracker statistically fits with the horizontal plane of the ILS. This is carried out calculating the Mahalanobis distance between the glide slope, projected to the ground, and the horizontal target position. Then, the calculated distance is compared with a threshold. This is equivalent to projecting the horizontal measure to the glide slope using the equations defined in section 2.4 and then comparing the likelihood of the projection against a threshold.

The vertical check consists in testing that the height measurement fits statistically with the ILS glide slope. For this purpose, the distance between the target and the ILS localizer is calculated and then the height that corresponds to that distance in the glide slope is obtained. Then, the Mahalanobis distance between the expected height and the current target height is calculated. After this, the Mahalanobis distance is compared against a threshold.

Lastly, if the target is compatible both horizontally and vertically with the glide path, it is checked that the target height is above a certain threshold

because the aircraft stop following the glide path at the end of the landing. If these three tests are positive, then the vertical velocity pseudo-measurement is generated as follows:

$$V_v = -V_{horiz} * \tan(3\pi/180) \quad (2.73)$$

Finally, the measure vector that is processed by the tracking filter is generated adding a second state containing the vertical speed pseudomeasure. The vertical tracker used for processing the vertical measurements is explained in the next chapter, in section 3.3. It is a robust vertical IMM but capable of performing the previous explained checks and of filtering the measures with speed information. The improvements derived from the use of these vertical pseudomeasures are also explained in chapter, in section 5.

Chapter 3

Robust Estimation Filters on Airport Area

3.1 Introduction

Target tracking has been a classic problem that has been constantly evolving over the last 40 years. Traditionally, estimators assumed that targets moved according to one model and used this model to obtain a better estimation, as it is the case with Kalman's filters [Kalman, 1960] [Kalman and Bucy, 1961] [Gelb, 1974]. When the target stops following that model, that is when the target performs a manoeuvre, the quality of the tracker gets worse. The first solution to this problem was the adaptive filters for manoeuvre detection [Wang and Varshney, 1993] [Bar-Shalom and Li, 1993] [Bar-Shalom and Birmiwal, 1982]. In these systems, when a manoeuvre is detected additional variables related to that manoeuvre are estimated. Therefore, its tracking quality is improved.

Afterwards, the solution to the problem of estimating targets manoeuvring improves to an approach that maintains a set of sub filters each one associated with a different movement model. This strategy is called multiple model algorithms. After each filter has obtained its estimation adapted to its mode, a final estimation is obtained by combining these modes. There are two

families of multiple-model algorithms: the Generalized Pseudo-Bayesian Approach (GBPA) [Tugnait, 1982] [Watanabe and Tzafestas, 1993] and the Interacting Multiple Model (IMM) algorithms [Li and Bar-Shalom, 1993] [Cho et al., 2013]. Due to the versatility of the IMM solution over the GBPA and its balance between performance and computational cost, it is one of the most popular approaches.

The IMM systems are divided into two types depending on whether they have a fixed or a variable number of models. IMM systems with a number of variable models are called VS-IMM. These systems can be used for target estimation without using any contextual information [Jilkov et al., 1999] although it is more common to use them when using contextual information such as a road map [Pannetier et al., 2005].

Nowadays, as the computational capacity of the systems in charge of surveillance improves, other solutions are becoming popular, such as particle filters [Jing and Vadakkepat, 2010] [Gustafsson et al., 2002]. These filters are based on a genetic sampling approach, with a set of particles to represent the posterior distribution of some stochastic process. The advantage of these systems over the traditional ones is that they do not depend on the noise distribution of the measurements being Gaussian and they have better performance when the target movement pattern is not known in detail.

In our particular problem, the targets to estimate are either on the surface of the airport, or landing, or taking off. This fact restricts the aircraft's movement, constraining the type of manoeuvres that can be made. As it has been already explained in section 1.4, the system proposed in this chapter does not use any information from the road map while the one explained in chapter 2 uses that information. Due to this fact and to the knowledge of the target movement patterns, a solution of the type IMM has been chosen.

The main reason why this system needs to be designed is to work as backup to the MBMHF (explained in chapter 2) in a way that it obtains the estimation of the targets that are not following the map. This can happen for a variety of reasons, such as targets going out of the runway (runway excursion), or when the map is poorly defined and does not contain all the

roads. The use of this filter also makes possible to alleviate the worsening of the MBMHF performance in the junctions since it is possible to implement a logic that, in the event that the MBMHF has multiple hypotheses and none predominates, it selects as system output the robust filter estimation. In addition, the robust filter is also necessary for vertical movements since the map is only used for landings following the glide slope.

Figure 3.1: Robust filter block diagram

The robust filter architecture can be seen in the figure 3.1. The motion estimation has been divided into its two independent parts, the horizontal movement and the vertical movement, each one with associated IMM filters (appendix A). As it is mentioned in chapter 4, the measurements at the entrance of this system may present local biases or barometric offsets that must be corrected before being used. This correction of the biases is represented by the blocks "Local Bias Estimator" (section 4.3.1) and "Geo-Baro offset estimator" (section 4.3.2). After this correction, the measurements are filtered obtaining the target state variables. Together, both systems obtain the 3D position of the target in Cartesian coordinates, the ground speed (value and heading), and the vertical velocity.

The structure of the chapter explaining this robust tracking system is as follows. First, in section 3.2 the IMM is detailed for the horizontal tracking, discussing each of the modes as well as the calculation of the transition probabilities between them. Next, in section 3.3 the operation of the vertical IMM is discussed in the same manner. Finally, section 3.4 explains how the robust filter complements the map-based filters explained in chapter 2. It should be noted that the performance analysis of the systems described here is in chapter 5.

3.2 Horizontal Robust IMM

In this section the designed robust surface IMM is detailed, that is, the one in charge of estimating the horizontal position, ground speed value, and the heading of the targets on the surface. Its block diagram can be seen in the figure 3.2. As it has already been introduced, this is a system in which the input is the measurements of the sensors already corrected from global biases (as will be explained in section 4.3.3) and correctly associated. First, the local biases are corrected (as will be described in section 4.3.1) and then, the measurements are processed by the robust IMM filter.

Figure 3.2: Surface Robust Filter

The horizontal tracker has to be able to process measurements from different sensors. Specifically, it has to process the measures from SMR, MLAT, ADS-B, WAM and SSR. It should be noted that the availability of SSR measures is usually limited, as for coverage reasons it only covers a small part of the take-off phase and landing phase. As a multi-sensor system there are

two ways to implement the architecture. It is possible to choose a centralized architecture where all sensor measurements are processed by the same filter [Chen et al., 2000] or to choose a distributed one where each sensor has its own filter [Liggins et al., 1997].

The main advantage of the centralized system is that there is less time between measurements, which allows the system to have less error in the manoeuvres. This is due to the fact that the higher the number of measurements, the earlier the mode change in the IMM is detected. On the other hand, a centralized architecture is very sensitive to uncorrected sensor related biases, deteriorating the performance. The implementation of both architectures is very similar. The main difference is that in the distributed architecture a process that combines the estimated position of each sensor's tracker must be added. Different types of combinations can be found in the literature [Bar-Shalom, 1990] [Khaleghi et al., 2013].

In the design of this robust filter, a centralized architecture has lastly been chosen due to the improvement already mentioned in the detection of manoeuvres, and because of its lower computational costs. This cost is an important issue since the number of simultaneous tracks that a system of this type can work with is usually a critical aspect. Anyhow, chapter 5 presents and analyses results comparing the centralized architecture with the distributed one with a Bayesian combination of tracks [Kay, 1993].

In the following section, there is a more detailed explanation of the designed IMM, discussing each of the modes and the calculation of transition probabilities.

3.2.1 Horizontal Dynamic Models

As it has been previously explained, each of the modes of an IMM is associated with a dynamic model and implemented by a Kalman Filter (KF) [Kay, 1993] [Gelb, 1974]. The KF is an optimal estimator that processes sensor measurements to estimate a set of variables that tend to be more accurate than those based on a single measurement alone. This is done by using

knowledge of the dynamic behaviour of the system and its measurements. It is important to emphasise that if either the dynamic or the measurement model is non-linear, it would be necessary to linearize them. From the different linearization approaches, such as Unscented Kalman Filters (UKF) [Wan and Van Der Merwe, 2000] or Cubature Kalman Filters (CKF) [Bhaumik et al., 2013], an Extended Kalman Filter (EKF) [Ribeiro, 2004] has been selected for simplicity. For more details about the KF algorithm or its extended version, the appendix A can be consulted.

Since the objective is to implement an IMM filter that is robust and works in Cartesian coordinates, the following dynamic models that cover the typical airport surface movements will be used:

1. Constant ground speed mode
2. Constant acceleration mode
3. Turn speed mode
4. Stop mode

It is important to highlight the stop mode since it is not usually used. This is because the constant speed mode normally includes the stop mode, the reason being that a stopped target is the same as a target at zero constant speed. However, using the stop mode has several advantages. First, the speed tends to oscillate around zero making the target move forward and backward around the actual point where it has stopped. This speed error generates additional position and heading errors. Second, the stop mode has the advantage that the mode change is detected earlier and therefore making the peak error generated by the manoeuvre last less time. These results will be discussed in detail in chapter 5. Next, each of these models will be explained.

3.2.1.1 Constant ground speed mode

The constant ground speed model is in charge of representing the movement of an aircraft at constant speed over the airport surface. Therefore, the parameters to be estimated are the position of the aircraft in Cartesian coordinates, the ground speed, and its heading. The dynamic of this motion model is defined by the following equations:

$$\frac{dv}{dt} = 0 \quad (3.1)$$

$$\frac{d\theta}{dt} = 0 \quad (3.2)$$

$$\frac{dx}{dt} = v \sin(\theta) \quad (3.3)$$

$$\frac{dy}{dt} = v \cos(\theta) \quad (3.4)$$

where v is the ground speed module and θ its heading. Considering that they have a non-linear relationship between the state variables, the mode filter is an EKF. The vector state is as follows:

$$x_f = \begin{bmatrix} x & y & v & \theta \end{bmatrix}^T \quad (3.5)$$

The transition function is defined following a constant speed movement:

$$f(\vec{x}_{k+1}) = \begin{pmatrix} x_k + T \sin(\theta_k) \cdot v_k \\ y_k + T \cos(\theta_k) \cdot v_k \\ v_k \\ \theta_k \end{pmatrix} \quad (3.6)$$

where T is the time since the last measurement. As we have an EKF we also need its associated Jacobian matrix:

$$F = Jacob(f) = \begin{pmatrix} 1 & 0 & T \sin(\theta) & -v \cdot T \cos(\theta) \\ 0 & 1 & T \cos(\theta) & -v \cdot T \sin(\theta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.7)$$

As it has been explained in section 1.2, the sensors measures are transformed to Cartesian coordinates and may or may not have speed information (measured as ground speed and heading). Therefore, the measure matrix takes the following form depending on whether there is speed information (H_2) or not (H_1):

$$H_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad (3.8)$$

$$H_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.9)$$

In this model a zero plant noise is considered since other models can take care of the movements that do not fit with this mode. Therefore, the Q matrix takes the following expression:

$$Q = 0_{4 \times 4} \quad (3.10)$$

where $0_{m \times n}$ is n-by-n matrix of zeros.

3.2.1.2 Constant acceleration mode

The constant acceleration mode is the one in charge of representing the movement of an aircraft accelerating/decelerating through the surface of the airport. This is a uniformly accelerated straight-line movement and since there is a non-linear relationship between the state variables, the mode filter is also an EKF. The dynamic of this motion model is defined by the following

equations

$$\frac{dv}{dt} = a_{long} \quad (3.11)$$

$$\frac{d\theta}{dt} = 0 \quad (3.12)$$

$$\frac{dx}{dt} = (v + a_{long} \cdot t) \sin(\theta) \quad (3.13)$$

$$\frac{dy}{dt} = (v + a_{long} \cdot t) \cos(\theta) \quad (3.14)$$

The vector state has five components: both Cartesian coordinates of the position, the ground speed, the heading, and the longitudinal acceleration.

$$x_f = \begin{bmatrix} x & y & v & \theta & a_{long} \end{bmatrix}^T \quad (3.15)$$

The transition function is defined as follows:

$$f(\vec{x}_{k+1}) = \begin{pmatrix} x_k + T \sin(\theta_k) v_k + T^2/2 \cdot \sin(\theta_k) \cdot a_{long_k} \\ y_k + T \cos(\theta_k) v_k + T^2/2 \cdot \cos(\theta_k) \cdot a_{long_k} \\ v_k \\ \theta_k \\ a_{long_k} \end{pmatrix} \quad (3.16)$$

As we have an EKF we also need its associated Jacobian matrix:

$$F = Jacob(f) = \begin{pmatrix} 1 & 0 & T \sin(\theta) & (v \cdot T + a_{long} \cdot T^2/2) \cos(\theta) & T^2/2 \cdot \sin(\theta) \\ 0 & 1 & T \cos(\theta) & -(v \cdot T - a_{long} \cdot T^2/2) \sin(\theta) & T^2/2 \cdot \cos(\theta) \\ 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.17)$$

The input measures are in Cartesian coordinates and may or may not have speed information (expressed as module and direction). Therefore, the measure matrix takes the following form, depending on whether there is speed information (H_2) or not (H_1):

$$H_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (3.18)$$

$$H_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (3.19)$$

This filter has a process noise in order to be able to follow non fully modelled manoeuvres. Therefore, the plant noise model is modelled by the following equation:

$$Q = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{KA}^2 T^2 \end{pmatrix} \quad (3.20)$$

where σ_{KA} is the comfort standard deviation and T is the time since the last measurement. This plant noise is derived from the Singer's model, assuming that the target suffers small random perturbations [Bar-Shalom et al., 2011].

3.2.1.3 Turn speed mode

The turn speed mode is responsible for representing the movement of an aircraft when turning. The dynamic of this motion model is defined by the following equations:

$$\frac{dv}{dt} = 0 \quad (3.21)$$

$$\frac{d\theta}{dt} = \frac{a_{trans}}{v} \quad (3.22)$$

$$\frac{dx}{dt} = v \cdot \sin(\theta) + t \cdot a_{trans} \cdot \cos(\theta) \quad (3.23)$$

$$\frac{dy}{dt} = v \cdot \cos(\theta) - t \cdot a_{trans} \cdot \sin(\theta) \quad (3.24)$$

In this case the vector state will have five components. Namely, the position with its X and Y Cartesian coordinates, the ground speed value and its associated heading, and the transverse acceleration. As it is non-linear, it is an EKF.

$$x_f = \begin{bmatrix} x & y & v & \theta & a_{trans} \end{bmatrix}^T \quad (3.25)$$

The transition function is defined as follows:

$$f(\vec{x}_{k+1}) = \begin{pmatrix} x_k + T \sin(\theta_k) \cdot v_k + T^2/2 \cdot \cos(\theta_k) \cdot a_{t_k} \\ y_k + T \cos(\theta_k) \cdot v_k - T^2/2 \cdot \sin(\theta_k) \cdot a_{t_k} \\ v_k \\ \theta_k \\ a_{t_k} \end{pmatrix} \quad (3.26)$$

$$T = t_k - t_{k-1} \quad (3.27)$$

As it is an EKF its associated Jacobian will also be used:

$$F = Jacob(f) = \begin{pmatrix} 1 & 0 & T \sin(\theta) & v \cdot T \cos(\theta) - aT^2/2 \sin(\theta) & T^2/2 \cos(\theta) \\ 0 & 1 & T \cos(\theta) & -v \cdot T \sin(\theta) - aT^2/2 \cos(\theta) & -T^2/2 \sin(\theta) \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -aT/v^2 & 1 & T/v \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.28)$$

Depending on the sensor which is being worked with, the measurements may or may not contain speed information. Therefore, the H matrix varies depending on the availability of speed information (expressed as a module and heading) or not:

$$H_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (3.29)$$

$$H_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (3.30)$$

Similar to the constant acceleration mode, this filter also has a process noise in order to be able to follow not fully modelled manoeuvres, using plant noise model as the indicated by the following equation:

$$Q = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{KT}^2 T^2 \end{pmatrix} \quad (3.31)$$

where σ_{KT} is the comfort standard deviation and T is the time, since the last measurement.

3.2.1.4 Stop mode

The stop mode is responsible for representing an aircraft stopped on the surface of the airport. Note that when the target is static, the state variables remain constant and therefore the dynamic model is linear, which allows to use a Kalman filter. The motion model is defined by the following equations:

$$\frac{dv}{dt} = 0 \quad (3.32)$$

$$\frac{d\theta}{dt} = 0 \quad (3.33)$$

$$\frac{dx}{dt} = 0 \quad (3.34)$$

$$\frac{dy}{dt} = 0 \quad (3.35)$$

In this case the vector state will have three components. Namely, the position with its X and Y coordinates, and the heading previous to the stop.

$$x_f = \begin{bmatrix} x & y & \theta \end{bmatrix}^T \quad (3.36)$$

$$F = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.37)$$

Depending on the sensor, the measurements may or may not contain speed information. The H matrix is thus depending on the availability of this information (expressed as a module and heading) or not:

$$H_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad (3.38)$$

$$H_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.39)$$

Again, in this model is considered a null plant noise because other models can take care of the movements that do not fit with this mode. Therefore, the Q matrix takes the following form:

$$Q = 0_{3 \times 3} \quad (3.40)$$

3.2.2 Horizontal Transition Probabilities

As it has been previously introduced, the transition probabilities are those used to premix the state estimators for each of the motion models after the previous update. This takes into account the possibility that the target may have changed the motion mode in the time interval between measurements. These probabilities are a critical part of the IMM design since its performance is very influenced by how they are designed.

These probabilities are not constant, as they depend on the time that has passed between the measurements. Low transition probabilities will make the system to take a longer time to detect manoeuvres and high transition probabilities will make the steady-state performance deteriorate since the system will assume manoeuvres between each measure.

In classical literature, a common solution for transition probabilities is the one defined in [Kastella and Biscuso, 1995]. There, the transition probabilities are defined for an IMM with only two modes: one associated to the target manoeuvring and the other associated to the target at constant speed. When the update period between measurements is low, which happens on the airport surface, the transition probabilities model can be approached by the next expression:

$$T[k] = \begin{pmatrix} 1 - T/T_{bm} & T/T_{bm} \\ T/T_m & 1 - T/T_m \end{pmatrix} \quad (3.41)$$

where T is the time that has passed since the last measurement, T_{bm} is a constant that reflects the average time between manoeuvres and T_m represents the average manoeuvring time. Therefore, the transition probabilities are defined as a relationship among the time between manoeuvres, their duration, and the time that has passed since the last measurement. Keeping this in mind, the transition probabilities are defined for our IMM with 4 modes. Each row/column of the transition matrix represent the following modes: constant speed, turn speed, constant acceleration, and stop. Then the matrix can be written like this:

$$T[k] = \begin{pmatrix} 1 - T/T_{bt} - T/T_{ba} & T/T_{bt} & T/T_{ba} & 0 \\ T/T_t & 1 - T/T_t - T/T_{ba} & T/T_{ba} & 0 \\ T/T_a & T/T_{bt} & 1 - T/T_a - T/T_{bt} - T/T_{ts} & T/T_{ts} \\ 0 & 0 & T/T_s & 1 - T/T_s \end{pmatrix} \quad (3.42)$$

where T_{bt} and T_{ba} represent the time between turns and between accelerations respectively, while T_t and T_a represent their average time. Moreover, T_s represents the average stop time and T_{ts} represents the average time the target takes to stop at the current speed. The zeros of the matrix represent the changes that are not allowed, for example the direct transition from constant speed mode to stop mode since it would be necessary to go through the acceleration mode to stop the target.

3.3 Vertical Robust IMM

The other tracking subsystem that is part of the robust tracker is the one in charge of processing the vertical measurements, estimating the height and

vertical speed of the targets. The block diagram of this system can be seen in the figure 3.3.

Figure 3.3: Robust Vertical Tracker

The robust vertical filter implemented is very similar to the robust architecture already explained above, in section 3.2 for the robust horizontal filter. The biggest difference is that this is a one-dimensional system since it only processes height and vertical speed. It is a multisensor system that processes vertical measurements from different sources and therefore these measurements come with an offset between those of geometric and barometric origin. Hence, the bias corrector explained in section 4.3.2 is needed in order to correct that offset. Specifically, the filter must be able to process vertical measurements from the following sensors: SMR, SSR, MLAT and ADS-B. Just like the robust horizontal filter, this system has a centralised architecture.

3.3.1 Vertical Dynamic models

The vertical tracking filter is based on a centralized IMM filter. Since it has to only follow the vertical component it is a one-dimensional tracking system. The IMM filter has 3 modes: constant vertical speed, vertical acceleration, and stop. These 3 modes are explained in detail below.

3.3.1.1 Vertical constant speed mode

The vertical speed mode is in charge of representing the height change at constant rate. Hence, the parameters to be estimated are the height and the vertical speed. As the state variables have a linear relationship this filter is a KF. The vector state is as follows:

$$x_f = \begin{bmatrix} h & v_v \end{bmatrix}^T \quad (3.43)$$

where h is the height and v_v is the vertical speed module. The transition matrix is defined following a constant speed movement:

$$F = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} \quad (3.44)$$

where T is the time since the last measurement.

The sensors provide position measures in Cartesian coordinates and may or may not have speed information. Thus, the measure matrix takes the following form depending on whether there is speed information or not:

$$H_1 = \begin{pmatrix} 1 & 0 \end{pmatrix} \quad (3.45)$$

$$H_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (3.46)$$

As it has been previously explained with the horizontal constant speed mode in section 3.2.1.1, this model is considered with zero plant noise because other models can take care of the movements that do not fit with this mode. Therefore, the plant noise matrix is as follows:

$$Q = 0_{2 \times 2} \quad (3.47)$$

3.3.1.2 Vertical acceleration mode

This mode is the one in charge of representing a vertically accelerating/decelerating aircraft. Since this model represents a uniformly accelerated vertical movement the relationship between the state variables are linear and therefore, the used filter is a KF. The vector state has three components: height, vertical speed, and vertical acceleration.

$$x_f = \begin{bmatrix} h & v_v & a_{vert} \end{bmatrix}^T \quad (3.48)$$

The transition matrix is defined as follows:

$$\begin{pmatrix} 1 & T & 0.5 \cdot T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix} \quad (3.49)$$

Depending on the sensor, the measurements may or may not contain vertical speed information. Then, the H matrix will depend on whether such speed information exists. Matrix H_1 is used when there is not speed measures and H_2 when it is available.

$$H_1 = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \quad (3.50)$$

$$H_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad (3.51)$$

In this case, a plant noise process is used in order to take into account the deviations on the modes implemented in the robust vertical filter. The noise model used is the Singer one [Singer, 1970], which models uncertainty as Gaussian white spectrum stochastic accelerations and constants between updates. Therefore, the Q matrix takes the following form:

$$Q = \begin{pmatrix} 0.5T^2 \\ T \end{pmatrix} \cdot \sigma_{AH}^2 \cdot \begin{pmatrix} 0.5T^2 & T \end{pmatrix} \quad (3.52)$$

where σ_{AH} is the plant noise standard deviation.

3.3.1.3 Vertical stop mode

This model is responsible for representing an aircraft in flight at constant height, i.e. with zero vertical speed. As the target is static, the state variables remain constant and the filter is a linear KF. In this case the vector state will have one component: the height.

$$x_f = \begin{bmatrix} h \end{bmatrix}^T \quad (3.53)$$

$$F = \begin{pmatrix} 1 \end{pmatrix} \quad (3.54)$$

As this mode does not estimate vertical speed, if the sensor provides that measure then it is discarded. Therefore, the H matrix is the following:

$$H = \begin{pmatrix} 1 \end{pmatrix} \quad (3.55)$$

In this mode the plant noise used is zero and therefore the Q matrix takes the following form:

$$Q = \begin{pmatrix} 0 \end{pmatrix} \quad (3.56)$$

3.3.2 Vertical Transition Probabilities

The transition probabilities for the vertical IMM filter are deduced using the same argument as for the horizontal filter, as seen in section 3.2.2. In this manner, the transition probability matrix is formed by rows/columns that represents the constant speed, constant acceleration and stop. This matrix is the following:

$$T[k] = \begin{pmatrix} 1 - T/T_{bva} & T/T_{bva} & 0 \\ T/T_{va} & 1 - T/T_{va} - T/T_{tvs} & T/T_{tvs} \\ 0 & T/T_{vs} & 1 - T/T_{vs} \end{pmatrix} \quad (3.57)$$

where T_{bva} represents the time between vertical accelerations, while T_{va} represents the average time of vertical acceleration. In addition, T_{vs} represents the average vertical stop time and T_{tvs} represents the average time it takes for the target to vertically stop at the current vertical speed. The zeros of the matrix represent the changes that are not allowed, for example the direct transition from constant speed mode to stop mode since it would be necessary to go through the acceleration mode to stop the target.

3.4 Integration of Robust Filter and Map-Based Filters

The robust horizontal and vertical filters that have been explained can be used as independent systems, but in this thesis it is proposed to use them as a backup for the map-based filters, as it can be seen in the figure 3.4. Therefore, the robust filters are in charge of providing the surveillance output in the situations where the map-based filters cannot obtain the target estimated positions. In the horizontal case the situations in which the map-based filters do not obtain the estimated position are when either the targets are out of the roads or the map is not updated with all the roads. In the vertical case the situation is when an aircraft is landing without following the glide path.

The logic of the horizontal combined system is the following. When the horizontal tracking system receives a measurement, the biases are corrected and then it is processed in parallel by both filters. Then, a selection is made between the estimated position of the robust filter and the MBMHF. If successive measurements have not been located on the map or no hypothesis of the MBMHF is predominant then the robust filter output is used. This selection allows to obtain the position, for instance, in case of runway excursion and to alleviate notably the worsening of the performance of the MBMHF in the first moments of the junctions. In the

Figure 3.4: Map-based filter with robust filter

vertical case the logic is also very simple. If the estimation of the trajectory obtained by the robust vertical filter is compatible with a landing following the glide path then vertical velocity pseudomeasures are generated. Otherwise, the robust vertical filter output is used.

Chapter 4

Bias Estimation and Correction

4.1 Introduction

The sensor measurements cannot be used directly by the surveillance systems as first they have to be preprocessed. This pre-processing is in charge of associating the measurements with the tracks and of correcting the biases. The association stage consists in grouping the measurements coming from different sensors with the track corresponding to each target. Once they have been associated, they go through the functionality associated with the maintenance of the tracks, which is responsible for creating, confirming or deleting them. After the tracks have been created and updated they are used to estimate and correct the different biases that the measurements may have. When this pre-processing has been completed, the corrected measures are used by the rest of the surveillance functions and sent to the control functions. This process can be seen in figure 4.1.

This chapter explains how to perform these different preprocessing stages. First, keeping in mind the ideas of the characteristics of the sensors explained in section 1.2, different association and track management techniques are briefly explained. Afterwards, it will be explained how to estimate and correct the three types of biases that most affect the measures on the airport area. The designed biases estimators are simple because in this thesis scope the important thing about these algorithms is their integration with the rest of the tracking filters.

Figure 4.1: Basic elements of tracking system

Specifically, the estimated biases are the global bias, the local bias and the offset in vertical measurements. Specifically, the global biases [Jarama et al., 2017] are the the bias of the sensor that do not depend on the target that the sensor is measuring. This bias is typically caused by poor sensor calibrations. The local biases [Besada et al., 2012] are the ones that they depend on the target being measured. The division of the horizontal biases into these two groups is not usual in the literature and this is a disadvantage since the global biases vary very slowly over time but the local bias are very fast varying according to the dynamics of the aircraft, the geometry of the scenario and the available sensors. Therefore, this division of the biases is an architectural solution to the bias estimation. Last, the third type of bias is the one presented in the vertical measurements [Tang et al., 2005] are the offset between the geometric and barometric height measures. In conclusion, using as reference the figure 4.1, this chapter introduces the stages of "Association and Track Management" and proposes an architecture to the "Bias Estimation and Correction".

The structure of the chapter is as follows. Section 4.2 briefly introduces the most common techniques for using the sensor measurements to initialize the tracks, and associates the following measurements to them. Afterwards, the most common biases in the measures available at the airport as well as the techniques to correct them will be explained. Section 4.3 discusses the designed architecture and section 4.3.1 explains what the so-called local biases are and how to correct them. Then, in section 4.3.2 the second type of bias is explained, which consists in the correction of the offset between geometric and barometric height measurements. Subsequently, section 4.3.3 similarly explains how to estimate and correct global biases. It should be noted that the performance analysis of the biases correction described here is in chapter 5.

4.2 Association and Track Management

Once the sensors measurements have been obtained, they have to be used to create new tracks or to update tracks already created. This process of matching measurements with existing or new tracks is called association. Many techniques can be found in the literature on how to perform this process [Gad et al., 2002] [Cox, 1993]. One of the simplest techniques is called Nearest Neighbour (NN) [Konstantinova et al., 2003] [Bar-Shalom and Li, 1995]. This technique consists in making the match minimizing a certain distance, usually normalizing this distance with the noise covariance matrices of the measures.

A more complex association technique is the MHT [Blackman, 2004] [Reid, 1977]. In this technique hypotheses are created generating all possible combinations of association. Once the hypotheses have been generated, a score is calculated, typically the likelihood, in order to select only the best hypotheses. Other techniques of association are the Markov Chain Montecarlo (MCMC) [Oh et al., 2009] that tries to reconstruct the target PDF by a collecting of samples and the optimal Bayesian technique [Panta et al., 2007] that uses the Bayes rule to simplify the target state PDF.

Once the measurements have been associated with the existing tracks, these are filtered and updated. If the measurements have been marked to create new tracks, they are not created directly. Instead a series of checks must be performed. A review of these algorithms can be found in [Leung et al., 1996] being the most common techniques the M of N rule [Bar-Shalom et al., 1990a] and the logic-based approach [Bar-Shalom et al., 1990b].

To conclude this brief review of the techniques of association and track initialization, it should be highlighted that the work made in this thesis does not include association and track management. Therefore, in our designs it is assumed that previous systems have performed these stages correctly.

4.3 Sensor Biases

The measurements provided by the sensors are affected by biases that have to be corrected before filtering them. If the biases are not corrected this can lead to

high error and the tracks not being correctly initialised or even being split. The sources of these biases are multiple: bad antenna calibrations, antenna-centroid displacements, transponder delays, etc. These bias terms can be divided in sensor related and target related. On the one hand, sensor related biases, also called global biases, are those whose value is the same for all targets as it depends only on the sensor, e.g. antenna deviation. On the other hand, target related biases, also called local biases, are those whose value also depends the target, e.g. transponder delay.

In the last years, the topic of the bias estimation and correction in a multisensor and multitarget environment has been widely researched with different techniques. [Besada et al., 2012] proposes to use a Kalman filter that estimates the biases obtaining three components: target dependent, sensor dependent and target-sensor dependent. [You et al., 2013] corrects radar systematics biases using least squares with radar and ADS-B measures and taking into account the error in the ADS-B timestamps. [He et al., 2013] uses joint maximum likelihood to solve problems regarding track correlation and sensor bias estimation. [Ristic and Clark, 2012] presents a particle filter implementation to estimate the bias.

Figure 4.2: Block diagram of the biases correction

Figure 4.2 shows the designed architecture for the correction and estimation of the biases for both the horizontal and vertical cases. When a vertical measurement is received it only has to be corrected from the barometric offset. In addition, the measurement before this correction is used to update the existing estimate of the

offset. Instead, horizontal measures need to be corrected of both global and local biases. First, the global bias is corrected, which estimation has been obtained from the previous measures and from reference targets. Once the global bias is corrected, the local bias is corrected and then the measurement is sent to the tracking filter. Additionally, the measure with only the global bias corrected is the one that feeds the local bias estimator. Lastly, it should be noted that if the horizontal tracker is map-based then there will be an additional estimator, specifically the lateral drift estimator.

4.3.1 Local Bias Estimation

The local bias estimator is oriented to the estimation/cancellation of target specific biases, such as those due to the centroid difference between each type of sensor, the specific geometry of SMR occlusions, possible navigator biases for ADS-B measurements, etc. These biases are very common on the airport area, with one of the most common being the difference between the centroid of the position obtained by the SMR and the position obtained by MLAT or ADS-B, as it can be seen in the figure 4.3. To be more specific, the positions obtained by the MLAT and ADS-B systems correspond to the positions associated to their respective antennas while the SMR measure is the centroid of all the detections associated with the target. Therefore, there is a longitudinal offset between the positions obtained by the three sensors that varies when the aircraft changes its direction. Before explaining this estimator, highlight that it is assumed that it is assumed that the global bias has already been corrected using the techniques explained in section 4.3.3.

To correct this type of bias, it is assumed that biases can be modelled as a slowly changing signal over time, with an exponential correlation in time. This exponential correlation model can be summarized by an equivalent formulation based on a Gauss-Markov model:

$$b_{x_m}(k) = \rho \cdot b_{x_m}(k-1) + v_{x_m}(k) \quad (4.1)$$

where ρ is a constant related to the time of decorrelation of the process and $v_{x_m}(k)$ is a stochastic discrete process, white and with Gaussian distribution. This

Figure 4.3: Differences in the centroids of the SMR and ADS-B

process has zero mean and variance $\sigma_b^2(1 - \rho^2)I_{2nx2n}$. This model assumes that, when stationary, $b_{x_m}(k)$ has a Gaussian distribution with zero mean and variance σ_b^2

With this bias model, a Kalman filter (appendix A) can be used for its estimation. Other methods such as the batch processing are discarded due to the high temporal variability of bias since the centroid of the SMR changes when the target manoeuvres and therefore the bias is not constant. This estimator calculates the local biases for the n sensors that are generating measurements of the same target. There are two states of bias for each sensor, corresponding to its bias at the X and Y coordinates. Denoting b as the filter status vector containing the bias for the two components of each sensor, it can be written as follows:

$$b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} b_{x_1} & b_{y_1} & b_{x_2} & b_{y_2} & \cdots & b_{x_n} & b_{y_n} \end{pmatrix}^T \quad (4.2)$$

This estimator will use as filter input the position measurement difference of each sensor with respect to another that will be taken as reference. For example, if sensor 1 measure is taken as a reference, $(\vec{x}_1)$, the measure vector can be defined as:

$$z = \begin{pmatrix} \Delta \vec{x}_{21} \\ \Delta \vec{x}_{31} \\ \vdots \\ \Delta \vec{x}_{j1} \end{pmatrix} = \begin{pmatrix} b_2 - b_1 \\ b_3 - b_1 \\ \vdots \\ b_j - b_1 \end{pmatrix} + \begin{pmatrix} n_2 - n_1 \\ n_3 - n_1 \\ \vdots \\ n_j - n_1 \end{pmatrix} \quad (4.3)$$

where n_i denotes the measurement noise associated to each sensor. Then, the measure covariance matrix R and the observation matrix H are defined. The R matrix is formed by the error of the reference sensor, that affects all the measurements, and the one of each sensor. If we denoted as R_i to each sensor covariance, the R matrix can be written as follows:

$$R = \begin{pmatrix} R_2 & 0 & \cdots & 0 \\ 0 & R_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_j \end{pmatrix} + \begin{pmatrix} R_1 & R_1 & \cdots & R_1 \\ R_1 & R_1 & \cdots & R_1 \\ \vdots & \vdots & \ddots & \vdots \\ R_1 & R_1 & \cdots & R_1 \end{pmatrix} \quad (4.4)$$

The H matrix is formed of a first column made up of the 2×2 dimension identity matrix with negative sign, because the first measurement is used as a reference. Then, another identity matrix appears, with a positive sign, in each of the positions relative to the first term of each difference:

$$H = \begin{pmatrix} -I_{2 \times 2} & I_{2 \times 2} & 0 & 0 \\ -I_{2 \times 2} & 0 & I_{2 \times 2} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -I_{2 \times 2} & 0 & 0 & I_{2 \times 2} \end{pmatrix} \quad (4.5)$$

The rest of the filter equations would be the usual ones for the Kalman filter:

$$K = P_p H^T (H P_p H^T + R)^{-1} \quad (4.6)$$

$$b_f = b_p + K(z - H b_p) \quad (4.7)$$

$$P_f = (I_{2j \times 2j} - K \cdot H)P_p \quad (4.8)$$

This estimator has a weakness due to the fact that it cannot estimate the bias of all the sensors at once, since the measurements of one of them are used to calculate this bias, making the system unobservable [Lee et al., 1982]. Therefore, it is proposed to assume that the reference sensor has null bias. This would cause that the bias for the rest of the sensors will be aligned with the one used as reference. In order to perform this alignment, it is generated a pseudo measure. This pseudo measure is a 2D vector with its two zero components and zero measurement noise:

$$x_m = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (4.9)$$

$$R = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad (4.10)$$

Using a matrix of zeros as plant noise can cause problems and it is necessary to check that this model does not generate invertibility problems. The observation matrix contains the identity in the columns associated with the sensor whose bias is going to be nullified, and the rest is zero:

$$H = \begin{pmatrix} I_{2 \times 2} & 0 & \cdots & 0 \end{pmatrix} \quad (4.11)$$

The filter update equations are the same as those of the normal Kalman filter and therefore they can be consulted in the appendix A.

Next a very simple simulated scenario is presented in which a target moves at a constant speed of 8 m/s, performs a turn with radius of 120 m and then continues at the same constant speed. Two sensors take measurements from it: a SMR and MLAT. The measurements of these two sensors have a longitudinal local bias with value 10 m. This bias can be expressed with the following equation, if translated to Cartesian coordinates:

$$b_{xy} = \begin{pmatrix} 10 \sin \theta_{tray}(t) \\ 10 \cos \theta_{tray}(t) \end{pmatrix} \quad (4.12)$$

Figure 4.4: Estimation of bias at X Coordinate

Figure 4.5: Estimation of bias at Y Coordinate

where $\theta_{tray}(t)$ is the trajectory heading for time t . Figure 4.4 shows the result for the X coordinate estimation, taking as reference the SMR, and the

95% interval. Figure 4.5 shows the same type of results for the Y coordinate. It can be concluded that the proposed estimator correctly tracks these local biases.

4.3.2 Barometric-geometric offset estimation

The vertical measurements provided by the sensors on the airport area are of two sources: geometric and barometric. The barometric measures come from the airborne barometer, which uses atmospheric pressure to determine the altitude. The conversion between pressure and altitude is made assuming the International Standard Atmosphere (ISA) model (temperature 288.15 K and pressure 101,325 Pa, at mean sea level) [Poles, 2010]. On the other hand, the geometric altitude is the Euclidean distance between the mean sea level and the position of the aircraft. Both altitudes are the same only when the atmospheric conditions are the ISA conditions. In the rest on cases, there is a non constant offset between them. An example of this offset can be seen in the figure 4.6. This figure shows the raw height measures from Mode S and MLAT sensors deployed in the Barcelona Airport.

Figure 4.6: Geometric and barometric measures from Barcelona airport sensors

Logically, the offset between both types of vertical measurements means that they cannot be used directly without correcting the offset. There are many ways to make this estimation, such as ordinary least squares [Zaliva and Franchetti, 2014], weighted least squares [Yao et al., 2009] or Federal Kalman filters [Hao and Huang, 2009]. In this thesis it is proposed an intermediate solution similar to the estimation of the local bias explained in section 4.3.1. This approach consists of estimating the offset using a Kalman filter and modelling this offset as a slowly changing signal over time, with an exponential correlation in time. This exponential correlation model can be summarized by an equivalent formulation based on a Gauss-Markov model:

$$\Delta h_p(k) = \rho \cdot \Delta h_p(k-1) + v_{h_p}(k) \quad (4.13)$$

where ρ is a constant related to the time of decorrelation of the process and $v_{h_p}(k)$ is a stochastic discrete process, white and with Gaussian distribution. This process has zero mean and variance $\sigma_h^2(1 - \rho^2)I_{n \times n}$. This model assumes that, when stationary, $\Delta h_p(k)$ has a Gaussian distribution with zero mean and variance σ_h^2 .

The vertical measurements have an additional problem which is that the error generated by quantization is usually bigger than the noise of the measurements. Additionally each sensor also has very different values of quantization step size. This means that no single value of the offset can be estimated since depending on the quantification step of the sensors on which the subtraction is made, one value or another will be obtained. Therefore, it is not possible to use a source of vertical measurement as a reference, instead a sensor has to be used as reference. With this in mind and the bias model, a Kalman filter is used for the estimation. As the offset is calculated with respect to the reference sensor, in the filter there are one state of offset for each sensor. The estimator state vector is as follows:

$$\Delta h_p = \begin{pmatrix} \Delta h_1 \\ \Delta h_2 \\ \vdots \\ \Delta h_n \end{pmatrix} \quad (4.14)$$

This estimator will use as filter input the position measurement difference of all possible pairs of geometric measurements with the reference barometric measurement. Following a similar reasoning to the one made for the local bias estimator, the measurement and its noise covariance matrix have the following form:

$$z = \begin{pmatrix} \Delta h_1 \\ \Delta h_2 \\ \vdots \\ \Delta h_n \end{pmatrix} = \begin{pmatrix} h_{g_2} - h_{p_1} \\ h_{g_3} - h_{p_1} \\ \vdots \\ h_{g_j} - h_{p_1} \end{pmatrix} + \begin{pmatrix} n_2 - n_1 \\ n_3 - n_1 \\ \vdots \\ n_j - n_1 \end{pmatrix} \quad (4.15)$$

$$R = \begin{pmatrix} R_2 & 0 & \cdots & 0 \\ 0 & R_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_j \end{pmatrix} + \begin{pmatrix} R_1 & R_1 & \cdots & R_1 \\ R_1 & R_1 & \cdots & R_1 \\ \vdots & \vdots & \ddots & \vdots \\ R_1 & R_1 & \cdots & R_1 \end{pmatrix} \quad (4.16)$$

where n_i is the measurement noise of each sensor and R_i its associated covariance matrix. The H matrix is formed of a first column made up of ones with negative sign, because the measurement is used as a reference. Then, another one appears, with a positive sign, in each of the positions relative to the first term of each difference:

$$H = \begin{pmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -1 & 0 & 0 & 1 \end{pmatrix} \quad (4.17)$$

The rest of the filter equations would be the usual ones for the Kalman filter:

$$K = P_p H^T (H P_p H^T + R)^{-1} \quad (4.18)$$

$$b_f = b_p + K(z - H b_p) \quad (4.19)$$

$$P_f = (I_{2j \times 2j} - K \cdot H) P_p \quad (4.20)$$

As in the case of the local bias estimator, a pseudomeasure is inserted in the filter to make null the bias of the reference sensor and align the rest of the sensors with it. The equations are the same as in the previous case but for the one-dimensional case:

$$x_m = \begin{pmatrix} 0 \end{pmatrix} \quad (4.21)$$

$$R = \begin{pmatrix} 0 \end{pmatrix} \quad (4.22)$$

The observation matrix contains the identity in the columns associated with the sensor whose bias is going to be nullified, and the rest is zero:

$$H = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix} \quad (4.23)$$

The filter update equations are the same as those of the normal Kalman filter, and they can be consulted in the appendix A. Next the figure 4.7 shows the offset estimated results for the scenario previously introduced in the figure 4.6. In that scenario, the height measures are geometric from MLAT and barometric from Mode-S. The parameters used by the estimator can be seen in the table 4.1. The estimator calculates correctly the offset value, which is around 30m. The peak observed around the second 650 is caused by an error in the raw vertical measurements that are the input of the system. This error can be also observed the figure 4.6.

Figure 4.7: Estimated barometric offset from Barcelona airport sensors

Parameter	Value
ρ	0.999
σ_b	5 m

Table 4.1: Parameters used by the vertical offset estimator

4.3.3 Global Bias Estimation

As previously explained, the local bias is due to sensors measuring the same target differently. In addition to this bias, there are other biases independent of the target, so they are not associated with the pairs of sensors with the sensor itself. We call these biases global biases. These biases have multiple sources such as poor radar calibrations, signal propagation in the atmosphere, etc.

Global biases usually require different correction models for each type of sensor. For example for the SSR, a complete and rigorous mathematical model can be found in [Jarama et al., 2017]. In that research the model analyses the physical effects that generate the biases for the range, azimuth and altitude measures. This model can be also implemented in classic bias estimation processes found in the literature such as [Lin et al., 2004] [Lin

et al., 2005] correcting most of these biases. For other types of common airport sensors such as MLAT, ADS-B or WAM other models can be found, such as those explained in [Besada et al., 2012], [Herrero et al., 2007] and [Abbud and De Miguel, 2014].

In this thesis it is proposed to make an estimator of the global biases for the radars available at the airport, i.e. SMR and SSR. The reason to focus on these two is that they are the sensors with the greatest error on the airport surface. To make this correction it is assumed that the radar bias model divide it into two components: distance and azimuth. The distance bias is split into two terms, one constant and the other depending on the distance the target is at. The azimuth bias has only one constant term. Mathematically the measurements can be modelled as:

$$R_m = (1 + K)R + \Delta R + n_R \quad (4.24)$$

$$\theta_m = \theta + \Delta\theta + n_\theta \quad (4.25)$$

where R_m and R are the measured range and the real one, K is the range gain, ΔR is the range constant bias term and n_R is the noise in the range measure. θ_m and θ are the measured azimuth and the ideal one, $\Delta\theta$ is the azimuth constant term and n_θ is the azimuth measure noise. Noise terms are modelled typically as uncorrelated additive white noises considered Gaussian distributed, while systematic error terms can be considered constants or slow time variant (bias terms). Since this thesis is focused on the airport area, the range measure of the targets does not change much and therefore it is assumed that the gain bias is constant so that the distance bias model is simplified:

$$R_m = R + \Delta R' + n_R \quad (4.26)$$

where $\Delta R'$ includes both bias terms. The estimation of this global bias is proposed to be made from two different sources and combine them, as it is shown in the figure 4.8.

Figure 4.8: Radar global biases estimation

First, being on the airport surface, so-called reference targets can be used. These targets are either reflectors or transponders, depending on whether it is for the SMR or the SSR, which their actual position is known with accuracy. Second, use each of the tracks to obtain a different estimation of the biases associated with each track. Subsequently, each of these biases associated with the track is passed through a system in charge of obtaining the final estimation of the bias through this source. Finally, it is proposed to combine the estimation from both sources. Each of these systems is explained below.

4.3.3.1 Global bias from track estimator

The estimation associated with each track ("Global bias from track estimator" in the figure 4.8) consists of exploiting the raw measurements of the sensors, taking into account that the differences of the measurements of two sensors on the same target are composed of the noises of the sensors and the global and local biases of these sensors. Therefore, a Kalman filter is applied on these position differences, assuming a dynamic behaviour of the biases, and modelling it as a slowly changing signal over time (with a temporal correlation of exponential type). It is important to note that in order not to worsen this estimation, the measures being processed must correspond to the times when the target is not manoeuvring, i.e., when it is either stationary or at constant speed. In addition, it should be noted that from each of the tracks a single estimation of the biases is obtained, which is then used together with the rest of the tracks to obtain the final estimation of the global

bias.

Therefore, in this model, the vector state has two components for each radar, corresponding to each of the two components of the bias, and other two for each non-polar sensor (usually MLAT and ADS-B) that would correspond to their local bias. It should be noted that although this estimator provides an estimation of the local biases, their parameters are adjusted to estimate the biases modelled as very slow signals, i.e. those that are almost constant. This implies that the filter assumes a much higher time correlation constant. Conversely, local bias changes fast as it changes when the target manoeuvres. Therefore, the local bias estimation obtained here is not exploited. Assuming there are k radars the vector state takes the following form:

$$b = \begin{pmatrix} \Delta R_1 \\ \Delta \theta_1 \\ \vdots \\ \Delta R_k \\ \Delta \theta_k \\ \Delta x_{mlat} \\ \Delta y_{mlat} \\ \Delta x_{adsb} \\ \Delta y_{adsb} \end{pmatrix} \quad (4.27)$$

The exponential correlation model can be summarised by an equivalent formulation based on a Gauss-Markov model, as follows:

$$b_i = \rho \cdot b_{i-1} + w_i \quad (4.28)$$

where ρ is a constant matrix related to the decorrelation times of stochastic processes with which the biases are modelled and w is a stochastic process discrete, white, and Gaussian distributed, with zero mean and Q covariance matrix. In more detail, ρ is a diagonal matrix that in each position has the correlation coefficient in polar or Cartesian coordinates as depending on the sensor. Equivalently, the covariance matrix Q is diagonal and in each posi-

tion has the covariance in polar or Cartesian coordinates, depending on the sensor. Specifically, they take the following form

$$\rho = \begin{pmatrix} \rho_R & 0 & \cdots & 0 & 0 \\ 0 & \rho_\theta & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \rho_x & 0 \\ 0 & 0 & \cdots & 0 & \rho_y \end{pmatrix} \quad (4.29)$$

$$Q = \begin{pmatrix} (1 - \rho_R^2)\sigma_{b_R}^2 & 0 & \cdots & 0 & 0 \\ 0 & (1 - \rho_\theta^2)\sigma_{b_\theta}^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & (1 - \rho_x^2)\sigma_{b_x}^2 & 0 \\ 0 & 0 & \cdots & 0 & (1 - \rho_y^2)\sigma_{b_y}^2 \end{pmatrix} \quad (4.30)$$

This model assumes that in stationary state b has a Gaussian distribution with zero mean and C_b covariance:

$$C_b = \begin{pmatrix} \sigma_{b_R}^2 & 0 & \cdots & 0 & 0 \\ 0 & \sigma_{b_\theta}^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_{b_x}^2 & 0 \\ 0 & 0 & \cdots & 0 & \sigma_{b_y}^2 \end{pmatrix} \quad (4.31)$$

Once the model is defined, the other equations are similar to those of the Kalman filter. By calling b_f and b_p the filtered and predicted bias respectively, and P_f and P_p their respective covariance matrices, the prediction stage is as follows:

$$b_p = \rho \cdot b_f \quad (4.32)$$

$$P_p = \rho \cdot P_f \rho^T + Q \quad (4.33)$$

The filter has as inputs the difference in Cartesian coordinates of measurements of the same target made by pairs of sensors. These pairs must be formed by a sensor with other that it is assumed as a reference. If we call the sensor i represents the reference sensor, the measurement vector is defined as follows:

$$z = \begin{pmatrix} \Delta x_{1i} \\ \Delta y_{1i} \\ \vdots \\ \Delta x_{ki} \\ \Delta y_{ki} \end{pmatrix} = \begin{pmatrix} x_1 - x_i \\ y_1 - y_i \\ \vdots \\ x_k - x_i \\ y_k - y_i \end{pmatrix} + \begin{pmatrix} n_{x_1} + n_{x_i} \\ n_{y_1} + n_{y_i} \\ \vdots \\ n_{x_k} + n_{x_i} \\ n_{y_k} + n_{y_i} \end{pmatrix} \quad (4.34)$$

Regarding the measurement matrix, the 2 columns corresponding to the sensor taken as reference contain the identity matrix, in the case of a non polar sensor, or the negative matrix that linearises the projection of the biases in polar to Cartesian, repeated vertically, due to the fact that the reference measurement appears subtracting in the previous equation. Then, another identity matrix appears, in the case that the sensor is non polar, or the matrix that linearises the projection of the polar to Cartesian biases, with a positive sign, in each of the positions relative to the sensor that produced the measurement that appears in the first term of each difference. For example, if we only have one non polar sensor and it is selected as reference, it looks like this:

$$H = \begin{pmatrix} \sin(\theta_1) & R_1 \cos(\theta_1) & \cdots & 0 & 0 & -1 & 0 \\ \cos(\theta_1) & -R_1 \cos(\theta_1) & \cdots & 0 & 0 & 0 & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \\ 0 & 0 & \cdots & \sin(\theta_k) & R_k \cos(\theta_k) & -1 & 0 \\ 0 & 0 & \cdots & \cos(\theta_k) & -R_k \cos(\theta_k) & 0 & -1 \end{pmatrix} \quad (4.35)$$

where R_m and θ_m represent the range and azimuth measured by the radar. Finally, the noise matrix shows that all measurements suffer the same error (that of the reference measurement), and have another component, independent for each measurement difference, coming from the other measurement. It takes the following form:

$$R = \begin{pmatrix} R_1 & 0 & \cdots & 0 \\ 0 & R_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_k \end{pmatrix} + \begin{pmatrix} R_i & R_i & \cdots & R_i \\ R_i & R_i & \cdots & R_i \\ \vdots & \vdots & \ddots & \vdots \\ R_i & R_i & \cdots & R_i \end{pmatrix} \quad (4.36)$$

Finally, the rest of the filtering equations would be those typical of the Kalman filter:

$$K = P_p H^T (H P_p H^T + R)^{-1} \quad (4.37)$$

$$b_f = b_p + K(z - H b_p) \quad (4.38)$$

$$P_f = (I_{2(k+1) \times 2(k+1)} - K H) P_p \quad (4.39)$$

4.3.3.2 Global bias from measures differences

As it has been indicated before, after making this filtering, it is obtained a unique estimation of the global bias that will be used together with the rest of the ones obtained from the other tracks. This estimation is made using another Kalman filter and is explained below. This filter discards the estimations of local biases so its estimation vector has only the components associated with the global biases. The estimator uses the same decorrelation model as the one shown in the equation 4.28. Therefore, it takes the following form:

$$b = \begin{pmatrix} \Delta R_1 \\ \Delta \theta_1 \\ \vdots \\ \Delta R_k \\ \Delta \theta_k \end{pmatrix} \quad (4.40)$$

Since it is the same model, only the matrices involved in the estimator are exposed. They are obtained by simply eliminating from the above matrices those terms related to non-polar sensors:

$$\rho = \begin{pmatrix} \rho_R & 0 & \cdots & 0 & 0 \\ 0 & \rho_\theta & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \rho_R & 0 \\ 0 & 0 & \cdots & 0 & \rho_\theta \end{pmatrix} \quad (4.41)$$

$$Q = \begin{pmatrix} (1 - \rho_R^2)\sigma_{b_R}^2 & 0 & \cdots & 0 & 0 \\ 0 & (1 - \rho_\theta^2)\sigma_{b_\theta}^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & (1 - \rho_R^2)\sigma_{b_R}^2 & 0 \\ 0 & 0 & \cdots & 0 & (1 - \rho_\theta^2)\sigma_{b_\theta}^2 \end{pmatrix} \quad (4.42)$$

The inputs of this estimator are directly the global biases obtained from tracks:

$$z = \begin{pmatrix} \Delta R_1 \\ \Delta \theta_1 \\ \vdots \\ \Delta R_k \\ \Delta \theta_K \end{pmatrix} \quad (4.43)$$

The measurement matrix will be a matrix composed of 2x2 identity matrices at the diagonal in the positions of the sensors from which bias estimation has been obtained:

$$H = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \quad (4.44)$$

Next, a simple scenario to evaluate the quality of this system is presented. This scenario simulates an airport where measurements of two SMR and one MLAT system are available. The sensor characteristics are shown in the table 4.2.

	SMR 1	SMR 2	MLAT
Bias	$\Delta R = -16m$	$\Delta R = 9m$	$\Delta X = 0m$
	$\Delta \theta = 0.25^\circ$	$\Delta \theta = 0.05^\circ$	$\Delta Y = 0m$
Accuracy	$\sigma_R = 5m$	$\sigma_R = 5m$	$\sigma_X = 5m$
	$\sigma_\theta = 0.02^\circ$	$\sigma_\theta = 0.02^\circ$	$\sigma_Y = 5m$

Table 4.2: Characteristics of the scenario sensors

In this scenario it is assumed that over a certain period of time 600 tracks have been available. First of all, it is obtained the estimation associated to each of the tracks. Figure 4.9 shows for each track the estimated the range bias for SMR 1. As expected, the estimations obtained have on average the expected value but are very disperse.

Then the estimates of the global biases coming from the tracks are passed through the filter in charge of smoothing these results. The figures 4.10 and 4.11 show the estimations of the biases in range and in azimuth, respectively. In this case the X axis represents the amount of tracks that has been used to

Figure 4.9: Range bias estimations for the SMR 1 from each track

Figure 4.10: Range bias estimation from measures difference

obtain this estimation. It can be seen that the estimates are quite accurate even with few tracks.

Figure 4.11: Azimuth bias estimations from measures difference

4.3.3.3 Global bias from reference targets

This estimation is based on filtering the difference in polar coordinates between the actual position of the reference target and the measurement by the sensor. This difference is then processed in a Kalman filter that smooths the value of the bias assuming a dynamic behaviour that can be modelled as a slowly changing signal over time. Therefore, it is a simple system, with the same model as the one explained above with the equation 4.28. Trivially it can be seen that the measurements that a radar takes on a reference target have the following expression:

$$R_{med} = R_{real} + \Delta R + n_R \quad (4.45)$$

$$\theta_{med} = \theta_{real} + \Delta\theta + n_\theta \quad (4.46)$$

In this way, knowing exactly the real position of the reference targets an measure of the radar global biases are obtained. Then, the Kalman filter would use the following model:

$$b = \begin{pmatrix} \Delta R \\ \Delta \theta \end{pmatrix} \quad (4.47)$$

The exponential correlation model used has the following transition matrix:

$$\rho = \begin{pmatrix} \rho_R & 0 \\ 0 & \rho_\theta \end{pmatrix} \quad (4.48)$$

The filter measurements would be directly the difference between the actual range and azimuth and the measured by the sensor. The measurement projection matrix H would simply be the identity matrix with dimension 2. Therefore, the measurements and its noise matrix are as follows:

$$z = \begin{pmatrix} \Delta R_{med} \\ \Delta \theta_{med} \end{pmatrix} = \begin{pmatrix} R_{med} - R_{real} \\ \theta_{med} - \theta_{real} \end{pmatrix} \quad (4.49)$$

$$R = R_{sensor} = \begin{pmatrix} \sigma_R^2 & 0 \\ 0 & \sigma_\theta^2 \end{pmatrix} \quad (4.50)$$

The rest of equations would be the usual ones from the Kalman filter, just as they were exposed for the previous case.

4.3.3.4 Combination

When the global bias has been estimated from the two available sources (measures differences and reference targets) they can be combined to try to improve the estimation. Because the covariance of both estimates is known, it is proposed to make the combination equivalent to the optimal Bayesian combination of vectors. This combination is done separately for each sensor and follows the equation below:

$$b_{i_{comb}} = \left(\sum_{l=0}^{r+1} P_l^{-1} \right)^{-1} \left(\sum_{l=0}^{r+1} P_l^{-1} b_l \right) \quad (4.51)$$

where $b_{i_{comb}}$ is the combined bias of sensor i , r is the number of available reference targets, and b_l and P_l represent the estimation of bias and its covariance from source l .

Chapter 5

Surveillance Function Results

5.1 Introduction

This chapter aims to analyse and discuss the performance of the different designed surveillance systems for the airport area. To be more specific, the following systems are evaluated: the biases estimator, the map-based filters, the robust filters, and the filters that combine both last approaches.

The analysed results correspond to either one iteration scenario or MonteCarlo simulations. In these simulations the following 95th percentile magnitudes are calculated:

1. Horizontal position error (Euclidean distance between simulated and predicted 2D positions).
2. Absolute value of ground speed error.
3. Absolute value of heading error
4. Vertical position error (absolute difference between simulated and predicted height).
5. Absolute value of vertical speed error.

In order to be able to accurately estimate a safe upper bound for this percentile with an associated confidence value of 95%, irrespective of error distribution, an approach using order statistics methods [Bain and Engelhardt, 1987], and a Gaussian approximation of binomial distribution is selected. Hence, the samples of the interesting magnitude are sorted in increasing order (from 1 to N) and then the index of the limit sample are calculated [Besada et al., 2014]. Therefore, the index is calculated as follow:

$$i = N \cdot 0.95 + 1.645 \cdot \sqrt{N \cdot 0.95 \cdot 0.05} \quad (5.1)$$

The structure of the chapter is as follows. Section 5.2 discusses the horizontal surveillance system performance. To be more specific, in subsection 5.2.1 the estimator of the horizontal biases is analysed. Then, in subsection 5.2.2 the MBMHF is evaluated, verifying the quality of the measures projection, the filtering, the computational cost, and its performance compared to other filters in the literature. Subsequently, in subsection 5.2.3 the performance of the robust horizontal filter is analysed. In subsection 5.2.4 the complete horizontal filter is discussed, i.e. the one that generates the output as a selection between the robust filter and the MBMHF. Lastly, in subsection 5.2.5 an additional scenario is analysed though using real measures obtained from the Barcelona airport. In section 5.3 the vertical system is analysed in an equivalent way to the horizontal filter, evaluating the correction of the barometric offset as well as the robust filter and the one that uses the glide path information.

5.2 Horizontal Tracking Performance

In this section the performance of the horizontal tracking filter is evaluated with different configurations in a simulated taxiway trajectory at the Barcelona airport. The trajectory is composed of a target that is stopped for 30 seconds. Then, it starts to travel along the surface of the airport. Then, it accelerates more and then stops at a holding point before entering the runway. The summary of longitudinal accelerations and speeds can be

found in the table 5.1 and the trajectory can be observed in the figure 5.1.

Time [s]	Speed [m/s]	Longitudinal Acceleration [m/s^2]
0 - 30	0	-
30 - 49	-	0.4
49 - 249	7.6	-
249 - 253	-	0.5
253 - 403	9.6	-
403 - 422	-	-0.5
422 - 452	0	-
452 - 460	-	0.5
460 - 520	4	-

Table 5.1: Taxi trajectory description

Figure 5.1: Taxi trajectory at Barcelona El Prat Airport

The available sensors in the scenario are a SMR and a MLAT system, whose characteristics are shown in the tables 5.2 and 5.3 respectively.

Parameter	Value	Description
σ_ρ	5 m	Range standard deviation
σ_θ	0.2 degrees	Azimuth standard deviation
T_s	1 sec	Update rate
Position	[1405, -146]	Cartesian coordinates

Table 5.2: SMR sensor characteristics

Parameter	Value	Description
σ_x	5 m	X standard deviation
σ_y	5 m	Y standard deviation
T_s	1 sec	Update rate

Table 5.3: MLAT sensor characteristics

5.2.1 Bias Estimation Results

Before analysing the tracking filters, the performance of the bias estimator is discussed. The aim is to see how they interact with each other especially when the biases are not being completely cancelled. The available sensors are a SMR and a MLAT system. It is assumed that these sensors have a longitudinal bias of 10m caused by the different centroids of the measurements. In turn, the SMR has global biases due to a bad calibration, namely a bias of 5m in distance and 0.3° in azimuth. The target of this trajectory does not move through the centre of the track but has a lateral bias of 2m.

Since the estimator of global biases obtains the estimate from multiple tracks, it is assumed that it has been working prior to this scenario and has already obtained an estimate of the global bias. Specifically, it estimated that the distance bias is 3m and in azimuth it is 0.25° . This causes that the uncorrected global bias projected to Cartesian coordinates is the one seen in figure 5.2.

Therefore, before the measurements are processed by the filter, the global bias is corrected with the previously mentioned estimate. This estimate of

Figure 5.2: Estimated Local Bias

the global bias is constant for the entire scenario. Subsequently the measurements are processed by a local bias estimator and then by the MBMHF which has an estimator of lateral drift. The constants used by both estimators are those shown in table 5.4.

	Local Bias	Lateral Drift
ρ	0.975	0.999
σ_b	3	2

Table 5.4: Parameters used by the estimators

The results of the local bias estimation are shown in figures 5.3 and 5.4. This figure shows the real local bias, the uncorrected global bias added to the local bias, and the estimated local bias. As expected, this estimator tries to calculate the local bias but, since it can not distinguish between the uncorrected global bias and the local bias, it estimates both. It was also foreseeable that the performances worsen slightly since it gets harder to follow the changes in the projection of the bias in the turns, but even so the performances are acceptable.

Figure 5.3: X-Coordinate Estimated Local Bias

Figure 5.4: Y-Coordinate Estimated Local Bias

Figure 5.5 shows the estimate of the lateral drift. As well as in the case of local bias, this estimator not only estimates the lateral bias itself but also the local and global offsets that have not been fully corrected in the previous stages. This causes that the estimated lateral bias to oscillate around the real

Figure 5.5: Estimated Lateral Drift

value of the latter, but still does not worsen the performances excessively.

5.2.2 Map-Based Tracker Performance

In this section only the performance of the MBMHF, explained in chapter 2, is discussed. Table 5.5 shows the constants used in this system, each one has the corresponding section in which it has been explained.

First some results corresponding a one iteration are discussed. To be more specific, figure 5.6 shows some results regarding to the localization and projection of a set of measurements of a target following part of the described trajectory. In this case, only the longitudinal projection is depicted, assuming that the target has no lateral drift, and the lateral drift tracker is disabled. It should be noted the along-segment projection is not normal, it is affected by the measurement covariance matrix, and therefore is different for each segment in the map and position.

Name	Symbol	Value	Section
Kalman Comfort Deviation	σ_{KA}	0.01 m/s^3	2.5
Segment Search Threshold	th_{loc}	$5 \cdot \max(\sigma_\rho, \rho \cdot \sigma_\theta) \text{ m}$	2.6.1
Mahalanobis Projection Threshold	$\lambda_{proj_{max}}$	9.21	2.6.3
Projection Like Miss	$N_{like_{miss}}$	6	2.6.3
Hypothesis Maximum Time without Update	T_{max}	3 sec	2.6.4
Minimum Hypothesis Probability	P_{Hmin}	0.02	2.3

Table 5.5: MBMHF Constants values

Figure 5.6: Projection to a single road

Next, figure 5.7 shows a projection to multiple segments. The target is moving and it arrives to an intersection. The system projects the measures to both roads until one of the segments is too far from the measures. After a little time, the associated hypotheses would be pruned. Again, in this case,

the target has no lateral drift, and the lateral drift tracker is disabled.

Figure 5.7: Projection result in an intersection

Next some one-run results concerning the longitudinal tracking are presented. Figure 5.8 shows the result of the complete longitudinal tracking system in an intersection point. The target, coming from the right of the figure, is turning to its left and with the first measures the probability of going straight and turning are similar so the MBMHF combines both resulting in an intermediate target position, which is out of the road centrelines. With the following measures the probability of turning to the left rapidly grows, and finally the estimated target position changes to the associated curve, as the other hypotheses get pruned. Once again, the target has no lateral drift, and the lateral drift tracker is disabled.

Figure 5.9 and 5.10 show one-run results, including lateral drift with a constant value of 5 meters and enabling the lateral drift tracker: the target is not following the centre of the road. Hence the lateral drift is estimated and corrected. Specifically, figure 5.9 shows the convergence in the estimation of lateral drift ($d_{trans}(f)$), and the associated 95% error band along the central estimate, according the lateral drift Kalman filter estimator.

Figure 5.8: Tracking result in an intersection without lateral drift

Figure 5.9: Lateral drift estimation

Figure 5.10 shows the results of the whole tracking filter (MBMHF plus lateral drift tracking) in 2D, for a part of the trajectory with intersections and associated hypothesis. Here it is clear how the lateral drift is back in the output estimate and the system is robust against this drift for the localization/longitudinal tracking processes.

Figure 5.10: Tracking result in an intersection with lateral drift

5.2.2.1 MonteCarlo Simulation

Now the MBMHF is analysed by a Montecarlo simulation with 500 iterations. The following figures show the 95th percentile error of the distance, ground speed and heading errors, comparing it with the using the system without transversal tracking.

First, only the MonteCarlo distance error result is analysed. As it can be observed in figure 5.11, it is noticeable that at each intersection the performance has a peak error. This is because, depending on the measurements, the filter may think the target is taking a curve that actually it is not in the route. Figure 5.11 also shows the average value of the error which is much lower.

In this scenario there are six distance peak errors as it can be seen in figure 5.12. The first of these peaks is the error of both the initial acceleration manoeuvre and the first turn. Being a mix of two manoeuvres involving two different modes of the IMM worsens the performance. The second peak is associated with intersections on the path. The peak error number 3 corresponds to a curve while the peak number 4 corresponds to a longitudinal acceleration manoeuvre. The peak error number 5 is a mixed error between a

Figure 5.11: Mean and 95th distance error

curve and a subsequent braking manoeuvre. The peak error number 6 corresponds to the acceleration after the target has stopped. As it was previously mentioned, the MBMHF presents peak errors at each intersection associated to the hypotheses uncertainty of these situations. This is solved using the tracker that combines the MBMHF with the robust filter as it is shown in section 5.2.4.

In the case of a speed error, figure 5.13, the four peaks simply correspond to the four longitudinal acceleration manoeuvres. The MBMHF presents a remarkable speed estimation performance both in manoeuvre and stationary. Finally, the heading error, shown in figure 5.14, is a mix between non error and small peak errors. The heading estimation using the map information greatly improves the performance but each intersection of the trajectory causes a small peak error which lasts until the correct hypothesis predominates. Again, this effects are reduced using a mixed approach of robust filter and MBMHF (section 3.4). Lastly, figure 5.15 shows the mean error in the drift estimation.

Figure 5.12: Distance estimation error

Figure 5.13: Ground speed estimation error

Figure 5.14: Heading estimation error

Figure 5.15: Drift estimation error

5.2.2.2 Comparison with Other Trackers

In order to demonstrate the quality of the MBMHF is compared with several trackers in the literature, both making use of map or not. The following results have been simulated in a different scenario because it would be necessary to adapt the complex map of Barcelona airport with the particularities that these systems need. For this reason, a simpler scenario is created in which the trajectory represents a target moving by the airport surface, using a flat earth model and 2D simulation, enough for our analysis. At the beginning, the target is stopped at [100,300] and it starts to accelerate to a speed of 7.5 m/s. Then, it moves at constant velocity until it decelerates and keep stopped for 25 seconds. After that, the target performs another acceleration to 15 m/s. At the end, the target stops again. Table 5.6 and figure 5.16 summarise this trajectory as a set of longitudinal acceleration segments and through the ideal path on the road segments.

Time [s]	Speed [m/s]	Longitudinal Acceleration [m/s^2]
0-15	-	0.5
15-75	7.5	0
75-90	-	-0.5
90-115	0	0
115-145	-	0.5
145-180	15	0
180-210	-	-0.5
210-240	0	0

Table 5.6: Trajectory description

The techniques taken as reference for comparison are: a standard IMM tracker without map information (two constant velocity models with different plant noise) [Bar-Shalom et al., 2004] that from now on it will be denoted by “IMM without Map”; an efficient particle filter constrained to roads proposed in [Cheng and Singh, 2007] that from now on it will be denoted by “Particle Filter”; a longitudinal Kalman filter that incorporates road map information, proposed in [Streller, 2008] that from now on it will be denoted

Figure 5.17: Distance error for the different trackers

Figure 5.18: Ground speed error for the different trackers

peak of the two previous trackers but is less accurate than the proposed MBMHF.

Figure 5.19: Heading error for the different trackers

5.2.2.3 Computational Load Results

Some results analysing the adequacy of MBMHF for real-time applications are shown in table 5.7. All the competing filters have been coded in a MATLAB, using a mono-thread approach, and have being executed in a computer with an Intel i5-2500K processor (release date 2011) and 8GB-DDR3 of RAM memory. The scenario being executed is that of the real data.

We have analysed the amount of time needed to process each track update, and calculated the number of measures (plots/track updates) that the different filters could process per second. This, assuming a sensor update period of 1 second is equivalent to the number of tracks that can be updated in real time, even in this low-end computer and with not too optimised code. Also, we calculated the worst-case minimum number of measures (plots/track updates) that the different filters could process per second by doing the inverse of the time needed to process the worst-case plot in the trajectory.

It can be observed that our filter, coded in not the most efficient programming language and executed in an average computer, has a reduced

Filter	Track update rate (Hz)	Min. Track update rate (Hz)
MBMHF	392	80
IMM with Map	7	6
Longitudinal KF with Map	950	293
IMM without Map	1200	290
Particle Filter	16	7

Table 5.7: Computational load (number of track updates per second)

computational load, only higher to that of the Longitudinal KF with Map and the IMM without Map approaches. The low performance of the particle filter is due to the amount of particles and the lack of parallel programming, while the low performance of the IMM with map is due to the amount of intersections between noise ellipses and segments that have to be calculated. In any case, it is clear our filter is suitable for real time tracking of large amounts of targets.

5.2.3 Robust Tracker Performance

Now the performance of the horizontal robust filter, explained in chapter 3, is analysed. The scenario is the same as the previously explained, table 5.1 and figure 5.1. First, the performance of the single-sensor scenario (SMR) will be evaluated. In addition, the performance according to whether there is a stop mode or not is discussed. Afterwards, a multisensor scenario (SMR and MLAT) with no local bias between the measurements will be analysed and then the same scenario but with local biases. In both multisensor scenarios the performance will be compared depending on whether the filter architecture is centralized or distributed. For these 3 analyses, a Monte Carlo simulation will be performed with 700 iterations measuring the distance, ground speed, and heading errors.

5.2.3.1 Monosensor Scenario

In this scenario it is assumed that only SMR sensor measurements are available and that there is no global bias or association errors. The performance of the robust IMM filter is analysed with 4 modes and with 3 modes (without stop mode). Both trackers are also compared with the use of the sensor plots without filter. The ground speed and the heading of the plots are obtained by performing the difference of each 2D Cartesian measure with the previous one, dividing by the measurement period, and translating this velocity estimate to polar coordinates. These results can be observed in figures 5.20, 5.21 and 5.22.

Figure 5.20: Taxi trajectory distance error for the IMM in the monosensor scenario

A series of error peaks have been numbered in the figures. In the figure 5.20, there are six error peaks corresponding to the error in distance. The first of these peaks is the one associated with the error of both initial manoeuvres: the initial acceleration and the first turn. Being a mix of two manoeuvres involving two different modes of the IMM worsens the performance. The error peaks number 2 and number 4 correspond to the second and third curve of

Figure 5.21: Taxi trajectory ground speed error for the IMM in the monosensor scenario

Figure 5.22: Taxi trajectory heading error for the IMM in the monosensor scenario

the path. Error peak number 3 corresponds to a longitudinal acceleration manoeuvre. Error peak number 5 is a mixed error between a curve and a subsequent braking manoeuvre. The error peak number 6 corresponds to the acceleration after the target has stopped.

In the case of a speed error, the four peaks simply correspond to the four longitudinal acceleration manoeuvres of the distance covered. Finally, the heading error has 4 error peaks, the first one corresponds to when the target is stopped and the filter is initialized, hence it is unknown whether the target will go in one direction or the opposite. The remaining 3 peaks correspond to the remaining turns of the trajectory. Note that this analysis of the manoeuvring error peaks is common for all taxi scenarios analysed in this section.

From these figures it can be inferred that, as expected, when the target is stopped the stop mode improves in a considerable way the stationary error in the three magnitudes. The exception is at the beginning of the scenario since the filter has just been initialized with a stopped target. Also, the stop mode does not worsen the performance when the target is moving which indicates that the transition probability model is correctly calculated.

5.2.3.2 Multisensor scenario: centralized and distributed architecture

In this section the performance of the robust filter in the multisensor case is analysed by comparing the two possible architectures: centralized or distributed. The measurements received are from the SMR and MLAT sensors with the characteristics shown in the tables 5.2 and 5.3. First, it is assumed that this is an ideal scenario in which there is no local bias. These results are shown in the figures 5.23, 5.24 and 5.25.

It can be seen that the results are similar with a few small exceptions. First, the performance of the distributed filter looks slightly better in stationary. Additionally, it has a much better performance at manoeuvres reducing both the magnitude of the error and its duration. A summary of this improvement can be seen in the table 5.8 for the distance case:

Figure 5.23: No local bias multisensor scenario: distance error

Figure 5.24: No local bias multisensor scenario: ground speed error

The performance of these two architectures is now analysed when there are local biases. Specifically, there is a 20m longitudinal bias between the centroid of SMR and MLAT positions. These results are shown in figures 5.26, 5.27 and 5.28.

Figure 5.25: No local bias multisensor scenario: heading error

	Maximum distance error [m]					
	Peak 1	Peak 2	Peak 3	Peak 4	Peak 5	Peak 6
Distributed IMM	19.8	12.7	7.1	16.3	18.1	11.1
Centralized IMM	13.9	10.1	6.5	12.5	15.3	9.6

Table 5.8: Centralized/Distributed IMM: manoeuvre distance error

The results are similar to the previous scenario without local biases, indicating that the bias estimator is working correctly. It can be also appreciated that the distributed system works slightly better when there are no manoeuvres but the centralised one works much better when these occur.

5.2.4 Map-Based Tracking plus Robust Tracking

Now the performance of the full horizontal filter, i.e. the one that switches between the MBMHF and the robust filter (explained in section 3.4), is analysed. The selection logic is based on two aspects. First, if the MBMHF cannot locate the measures in the map or it has not any hypotheses, then the robust filter output is selected. Second, if the MBMHF reaches an intersection and does not have a hypothesis with a probability greater than a

Figure 5.26: Local bias multisensor scenario: distance error

Figure 5.27: Local bias multisensor scenario: ground speed error

threshold, the system selects as output the estimation of the robust filter. This allows to improve the performance reducing the error peaks in each intersection.

The proposed scenario is the monosensor taxi version previously explained. It is assumed that the measures are correctly associated and the biases are

Figure 5.28: Local bias multisensor scenario: heading error

already corrected. No lateral drift is being estimated.

Figure 5.29: Map-Based Tracking + Robust Tracking: distance error

Figure 5.30: Map-Based Tracking + Robust Tracking: ground speed error

Figure 5.31: Map-Based Tracking + Robust Tracking: heading error

Figure 5.29, 5.30 and 5.31 show the performance of the system. As expected, the error associated to each junction is greatly reduced, maintaining the best performance of the MBMHF for the rest of situations. The heading estimation, once all the junction error peaks have been eliminated, has a remarkable performance because it is extracted directly from the road. The distance error is also significantly reduced which, added to the smaller error in the turns, makes the estimation of the position very accurate. The ground speed estimation has similar performance because it does not get affected at intersections.

5.2.5 Real Data Results

In order to demonstrate the performance of our system with real data, even in a multisensor deployment, a scenario from the Barcelona El Prat Airport (IATA: BCN, OACI: LEBL) is also analysed. The filtered data come from an aircraft performing a taxi movement. The sensors available are a SMR and MLAT system. The SMR noise covariance used in the filter is built from the standard deviations shown in table 5.2 while the MLAT is built assuming an independent noise for both coordinates of value $5m$. The airport map, the SMR position and the trajectory are shown in figure 5.32. Also, the rest of filter parameters are those described in table 5.5.

To check the performance of the system we compare the results from the different filters. The problem now is that the ideal trajectory is not available, so we use some indirect measures of the performance. Specifically, we use the following performance measures:

1. To assess position accuracy, the residual between the measures and the predicted position is measured. This residual, ideally, has two components, one related to measurement noise (equal for all filters), and another one related to predicted filter error. So, in general, smaller residual means better predicted filter error and therefore a better filter with respect to position.

Figure 5.32: Barcelona El Prat Airport

2. To assess ground speed accuracy, the difference between successive filtered ground speed values is also measured. In the actual trajectory, the ground speed changes smoothly as the aircraft moves, but any filter will produce a somehow noisy ground speed and may also have rebounds near acceleration manoeuvres. So, a reduced difference between filtered ground speed values, in general, means better filter with respect to velocity.
3. To assess heading accuracy, the different between the filtered heading and the ideal heading of the segment being followed by the target is calculated. The segment can be trivially obtained offline after aircraft operation. For turning (or spline) segments the methods to obtain the ideal heading are described in section 2.4.

Figure 5.33 shows the absolute value of the residual of the position along time, where it can be seen that our filter has reduced error peaks than most competing algorithms and one of the smallest average values. Table 5.9 shows the average value of this magnitude and its 95th percentile along the trajectory. In those results the quality of the MBMHF and that of most of the competing filters are similar, but it is much better than the IMM with Map, and slightly better than the Longitudinal KF with Map.

Figure 5.33: Residual of the position

Filter	Average Residual [m]	95th Percentile Residual [m]
MBMHF	0.25	8.91
Robust IMM	0.36	10.03
MBMHF + Robust IMM switch	0.27	9.2

Table 5.9: Absolute residual of the position

Figure 5.34 shows the estimated ground speed for each of the filters, and figure 5.35 the absolute difference between two consecutive ground speed updates. It can be seen that the MBMHF and the IMM without Map have a good and stable performance while the other filters present problems in the intersections, making the ground speed oscillate. Then, table 5.10 shows the average value of the absolute difference between ground speed updates and

its 95th percentile along the trajectory. In those results the quality of the MBMHF and the IMM without Map show to be clearly better than those of the rest of the filters.

Figure 5.34: Ground Speed tracking result

Filter	Aver. GS diff [m/s]	95th Percent GS diff. [m/s]
MBMHF	0.380	0.631
Robust IMM	0.378	0.638
MBMHF + Robust IMM	0.406	0.726

Table 5.10: Ground speed absolute difference

Figure 5.36 shows a detail of the ground speed estimation when the local bias is not being estimated or corrected. The bias between the SMR and MLAT measures makes the tracking to have form of sawtooth wave since the measurements move forward and backward constantly. The presence of this bias also impacts into the MBMHF measure projection. Table 5.11 shows the average value of the absolute difference between ground speed updates and

Figure 5.35: Absolute difference between ground speed updates

its 95th percentile along the trajectory. Obviously, the position estimation also deeply worsens.

Filter	Aver. GS diff [m/s]	95th Percent GS diff. [m/s]
MBMHF	1.200	2.64
Robust IMM	0.861	2.072
MBMHF + Robust IMM	1.109	2.522

Table 5.11: Ground speed absolute difference without local bias estimation

Figure 5.36: Detail of ground speed tracking when no local bias estimation

Figure 5.37: Heading tracking error

Filter	Aver. Abs. Error [°]	95th Abs. Error [°]
MBMHF	1.39	6.57
Robust IMM	3.82	21.81
MBMHF + Robust IMM	1.33	6.81

Table 5.12: Heading error

Finally, the results showing the quality of the heading estimation are depicted and analysed. In figure 5.37, the absolute heading error is depicted for all competing filters. It can be observed that the error in our system is much lower than that of all the competing filters. This is clearly seen in the Table 5.12, summarising along-trajectory average value of the absolute heading error and its 95th percentile.

5.3 Vertical Tracking Performance

In this section the performance of the vertical tracking filter is evaluated with different configurations. Two multisensor scenarios will be used: one with a take-off and another with a landing. In these two scenarios three aspects are checked: the performance of the vertical filter, the performance of the barometric bias estimator, and the improvement of incorporating the glide slope information to the vertical tracker (section 2.8). To this end, the two scenarios are used as follows: first the take-off scenario without any biases is evaluated, then the same scenario is evaluated but with barometric offset, and lastly the landing scenario with the tracker exploiting the glide slope information is evaluated.

The take-off trajectory is composed of a target that is stopped for 30 seconds. Then it starts to move until it reaches the runway. Then, it performs a strong acceleration movement and then it takes off. The summary of accelerations and speeds can be found in table 5.13 and it can be observed

Time [s]	Horizontal Speed [m/s]	Longitudinal Acceleration [m/s^2]	Vertical Speed [m/s]
0-30	0	-	-
30-38	-	0.5	-
38-78	4	-	-
78-118	-	1	-
118-153	45	-	-
153-273	45	-	19

Table 5.13: Takeoff trajectory description

Figure 5.38: Takeoff trajectory at Barcelona El Prat Airport

in figure 5.38. The sensor data can be found in the previous tables, 5.2 and 5.3, and in tables 5.14 and 5.15.

The landing trajectory is just a target descending at a constant vertical and horizontal speed. Once the target hits the ground, it brakes and then it moves at constant speed. The summary of accelerations and speeds can be found in table 5.16. The trajectory is the same that the one shown in figure 5.38 but in reverse order.

Parameter	Value	Description
σ_ρ	20 m	Range standard deviation
σ_θ	0.09 degrees	Azimuth standard deviation
σ_h	5 m	Vertical standard deviation
Position	[0, 4000]	Cartesian coordinates
T_s	1 sec	Update Rate

Table 5.14: SSR sensor characteristics

Parameter	Value	Description
σ_h	5 m	Vertical standard deviation
T_s	1 sec	Update Rate

Table 5.15: MLAT vertical characteristics

5.3.0.1 Multisensor takeoff with no barometric offset

In this scenario it is assumed that vertical measures from MLAT and SSR are available and that there is no global bias, barometric offset, or association errors. The performance of the vertical robust IMM filter is analysed and it is also compared with the use of the sensor plots without filter. These results can be observed in figures 5.39 and 5.40.

The results have only one manoeuvring peak corresponding to the transition between the three IMM modes: from stop to acceleration and then to constant speed. Hence, it may seem like a rather abrupt peak, but it is to be anticipated. The results in stationary mode are as expected with this type of estimation system.

5.3.0.2 Multisensor takeoff with barometric offset

This scenario is the same that the previous one but with barometric offset between the measures provided by the MLAT and the ones provided by the SSR. It is assumed that there is no global bias or association errors. The

Time [s]	Horizontal Speed [m/s]	Longitudinal Acceleration [m/s^2]	Vertical Speed [m/s]
0-190	40	-	-2.09
190-227.5	-	-0.8	0
227.5-427.5	10	-	-

Table 5.16: Landing trajectory description

Figure 5.39: No barometric offset multisensor scenario: height error

performance of the vertical robust IMM filter is analysed and also compared with the use of the sensor plots without filter. These results can be observed in figures 5.41 and 5.42.

It can be seen that for coverage reasons when the target is still on the airport surface there are only geometrical measurements and the height that is taken as reference is the barometric. Since there are only measurements from one source at the beginning, the offset cannot be estimated and therefore there is an additional constant error of $20m$. Afterwards, when the target is taking off, the coverages overlap and the offset is estimated. There is only one manoeuvring peak which corresponds to a transition from constant height to vertical speed modes of movement.

Figure 5.40: No barometric offset multisensor scenario: vertical speed error

Figure 5.41: Barometric offset multisensor scenario: height error

5.3.0.3 Multisensor landing with vertical speed pseudomeasures

This subscenario represents a landing trajectory performed in compliance with the glide path. The sensors available are the MLAT and the SSR with the same characteristics as in the previous scenarios. The objective of this subscenario is to test the improvement as a result of using vertical velocity

Figure 5.42: Barometric offset multisensor scenario: vertical speed error

pseudomeasures, as it has been explained in section 2.8. It is assumed that the barometric offset has been corrected and therefore is negligible. The results can be seen in the figures 5.43 and 5.44.

Figure 5.43: Vertical pseudomeasures scenario: height error

Figure 5.44: Vertical pseudomeasures scenario: vertical speed error

As expected the results improve using these pseudo measures. The error in the speed estimation is significantly reduced, going from an average error of $0.5m/s^2$ to $0.05m/s^2$. This impacts on the height estimation by reducing the 95th percentile of height error to approximately $1.3m$. When the plane ceases to be on the glide path, the filter becomes identical to the robust one and therefore presents the same performance.

Part II

Control Function

Chapter 6

Conflict Prediction and Detection

6.1 Introduction

One of the main objectives of an A-SMGCS system should be to ensure minimum safety conditions on which all operations are carried out. As stated in the IATA annual report on safety [IATA, 2017] it remains one of the biggest areas for improvement, despite having improved its overall safety performance by 54%. The most common type of accident that occurs on the airport surface is Runway/Taxiway Excursion, usually caused by the weather, which in 2016 it represents at 19% of the total. IATA also states that although runway excursions are the most common type of accident, the associated fatality rate was less than one percent. For this reason, IATA recognizes the need for continued improvement in runway in order to reduce the runway incursions and the collisions.

To reduce the risk of accidents, there are systems called Safety Nets. These systems may be ground-based or airborne and they alert controllers or pilots to an increased risk to flight safety. Its operation is based on the use of planned routes and surveillance data to detect possible conflicts. IATA defines 4 key pillars of security on how to improve conflicts. They are the

following:

1. Improve technology
2. Regulatory harmonization
3. Training
4. Awareness

In this thesis, it is going to focus on improving conflict detection by proposing improvements in technology, i.e. in Conflict Detection and Resolution (CDR) systems. A complete summary of the operation of these algorithms can be found in the figure 6.1. In this figure we can see how the positions of the airport vehicles are obtained by a network of sensors to later obtain their estimated positions by some tracking system. These estimated states, in general position and speed, form the main input of the CDR algorithm. The CDR system uses them together with the airport context which can include the occupied runways, the assigned routes, the maximum speeds of the ways, the minimum separation between aircraft, etc.

Figure 6.1: Complete CDR diagram block

Once the CDR receives the estimated states of the vehicles they are processed by three systems. First of it is checked if there is a conflict going on at the moment. Then, it will predict whether there will be any conflict in a given time window. Finally, in the event of a conflict, the system will propose a resolution of the conflict and will notify the operator. The system

as it has been described so far is a complete system, but not all the CDR perform the 3 stages (detection, prediction and resolution) and some only perform the first stage.

An extensive review of different algorithms and methods of this field can be consulted at [Kuchar and Yang, 2000]. This review ranks the different algorithms according to their following characteristics. First, the algorithms are classified according to the dimension in which they work, as they can detect conflicts horizontally, vertically or both simultaneously. Algorithms are also classified according to whether apart from detecting conflicts they also predict them and, if so, how they make such predictions. This prediction can be calculated from a simple approach using the speed vector to more complex approaches using statistical models of manoeuvres. The third aspect of the classification is whether the system has conflict resolution and if so, how it is obtained. Finally, the review also takes into account how the algorithms work with simultaneous aircraft, that is, how it calculates the pairs with whom it has to detect conflicts. This aspect is often critical due to its impact on the time it takes the system to analyse traffic.

One of the most commonly used systems to detect conflicts in airplanes in flight is the TCAS [FAA, 2011] which is an airborne CDR widely deployed based on SSR transponder signals. The TCAS interrogates the Mode C and Mode S transponders of nearby aircraft and from the replies tracks their altitude and range and issues alerts. This leads to the disadvantage that non-transponding aircraft are not detected. In the event of a conflict, it proposes one pilot to ascend and the pilot of the other aircraft to descend in order to avoid a possible collision. In [Kuchar and Drumm, 2007] different changes are proposed to improve the performance reducing the missed or late threat detections that can lead to collisions.

Another CDR used is the STCA [EUROCONTROL, 2009b]. This system is based on detecting separation conflicts in the various types of airspace, e.g.: route airspace, terminal areas, approach sequencing, departure regions and holding areas. The STCA uses the information provided by the surveillance sensors, typically ADS-B, MLAT and SMR. It notifies the ATC when the minimum separation between two aircraft is not met or has been predicted

not to be met. Due to the high manoeuvrability of the targets, this prediction is usually limited to two minutes.

Other common safety nets are the Area Proximity Warning (APW), the Approach Path Monitoring (APM) and the Minimum Safe Altitude Warning (MSAW). APW [EUROCONTROL, 2017b] warns the controller about unauthorised penetration of controlled flights into restricted airspace or penetration of uncontrolled flights into controlled airspace. APM [EUROCONTROL, 2017a] try to avoid controlled flight into terrain accidents using the aircraft altitude during final approach. MSAW [EUROCONTROL, 2009a] warns the Air Traffic Controller (ATCO) about the increased risk of controlled flight into terrain by generating an alert of aircraft proximity to terrain or obstacles. APM and MSAW are different but complementary functionalities. In the final approach phase MSAW will generate nuisance alerts. The APM functionality prevents this. APM then continues to provide protection until close to the runway.

On airport surface, the safety nets are quite different to the ones associated to in-air aircraft, since on surface the movement is reduced to 2D and the areas through the aircraft can move are more limited. There, the usual accidents are runway collision, ground damage, undershoot (a touchdown off the runway surface), hard landing, tailstrike and off-airport ditching. As can be deduced they are very different accidents and with many different contributing factors. As indicated earlier in this thesis, it will focus mainly on runway collisions and runway incursions.

The runway collision is prevented detecting the targets that are not maintaining the minimum separation. The conflicts related with the runway incursions are detected checking if the targets are entering occupied runways or if the targets are not respecting the holding points.

In this chapter an airport surface conflict detector, specifically the one shown in the figure 6.2, is analysed. This conflict detector is capable of detecting three types of conflicts: failure to maintain minimum separations between aircraft and other vehicles, failure to comply with the indicated speeds of the ways, and runway incursions. For this purpose, two implemen-

tations are proposed. On the one hand, to use directly the positions obtained by the surveillance systems. On the other hand, to add the information of the map to improve the detection of these conflicts. The map information is not used in other solutions except to calculate the target pairs [Zhu et al., 2012] [Mario and Rife, 2012]. The use of the map into the evaluation of the conflict allows to reduce considerably the false alarm probability since the separation between targets is obtained taking into account the roads and not using directly the Cartesian distance. In this chapter, the use of the map is also used to implement new and faster ways to detect the runway incursion conflict

Figure 6.2: Proposed conflict detector diagram block

This chapter is organized as follows. First, section 6.2 discusses about the safety net that detects that minimum separation requirements are not met. For this purpose, section 6.2.1 analyses the mathematics involved in this conflict detector. Then, in section 6.2.2 the safety net is designed and it is also proposed to incorporate the map information to improve its performance. Lastly, in section 6.3 the results of the minimum separation safety net are analysed. The runway incursion safety net is discussed in section 6.4. To this end, in section 6.4.1 the mathematics involved in this safety net are analysed. Afterwards, in section 6.4.2 the runway incursion safety net is designed and how the map information can be incorporated using the holding points. Finally, section 6.5 shows the results on this safety net

6.2 Minimum Separation Conflict

A typical case of a conflict alert is to check that two targets are no closer than a minimum distance. This minimum distance may depend on many factors, such as typical operating speeds, visibility and braking conditions, etc. In this section a safety net capable of detecting targets that do not meet the minimum separation is designed. To this purpose, first the mathematics involved are analysed.

Figure 6.3: Example of different distance conflict situations

6.2.1 Mathematical Analysis

6.2.1.1 Distribution analysis

It is assumed that the target measures, obtained directly from the sensor or from the estimator, are unbiased and under the effect of an additive Gaussian noise. These noise samples are identically distributed and drawn from a zero-mean normal distribution with covariance Σ . Denoting the position coordinates of the target i as $[x_i, y_i]^t$ and its measures as $[x_{m_i}, y_{m_i}]^t$ then the distribution can be written as:

$$[x_{m_i}, y_{m_i}]^t \sim N([x_i, y_i]^t, \Sigma_i) \quad (6.1)$$

where Σ_i denotes the position covariance matrix and can be written as following with ρ_i denoting the correlation coefficient between both coordinates:

$$\Sigma_i = \begin{bmatrix} \sigma_{x_i}^2 & \sigma_{xy_i} \\ \sigma_{xy_i} & \sigma_{y_i}^2 \end{bmatrix} = \begin{bmatrix} \sigma_{x_i}^2 & \rho_i \sigma_{x_i} \sigma_{y_i} \\ \rho_i \sigma_{x_i} \sigma_{y_i} & \sigma_{y_i}^2 \end{bmatrix} \quad (6.2)$$

The subtraction of two measures from different targets is the measured distance for each Cartesian coordinate, $[d_{m_x}, d_{m_y}]^t$. The subtraction of two normal distributions is also normally distributed, with mean the difference between both measures and covariance matrix their addition [Peebles and Shi, 2001], if they are independent:

$$\begin{aligned} \begin{bmatrix} d_{m_x} \\ d_{m_y} \end{bmatrix} &= \begin{bmatrix} x_{m1} - x_{m2} \\ y_{m1} - y_{m2} \end{bmatrix} \\ \begin{bmatrix} d_{m_x} \\ d_{m_y} \end{bmatrix} &\sim N \left(\begin{bmatrix} x_1 - x_2 \\ y_1 - y_2 \end{bmatrix}, \Sigma_1 + \Sigma_2 \right) = N \left(\begin{bmatrix} d_x \\ d_y \end{bmatrix}, \Sigma_{D_{xy}} \right) \end{aligned} \quad (6.3)$$

where $[d_x, d_y]^t$ represents the real difference vector between both targets and $\Sigma_{D_{xy}}$ represents the covariance of the measures difference. This covariance can be written as:

$$\Sigma_{D_{xy}} = \begin{bmatrix} \sigma_{x_1}^2 + \sigma_{x_2}^2 & \sigma_{xy_1} + \sigma_{xy_2} \\ \sigma_{xy_1} + \sigma_{xy_2} & \sigma_{y_1}^2 + \sigma_{y_2}^2 \end{bmatrix} = \begin{bmatrix} \sigma_x^2 & \rho \sigma_x \sigma_y \\ \rho \sigma_x \sigma_y & \sigma_y^2 \end{bmatrix} \quad (6.4)$$

As the expression of the PDF for a bivariate normal distribution is known, the probability of existing conflict given two measures can be directly calculated.

$$f_{D_{xy}}(d_x, d_y) = \frac{1}{2\pi |\Sigma_{D_{xy}}|^{1/2}} \exp \left(-\frac{1}{2} \begin{bmatrix} d_x - \mu_{d_x} \\ d_y - \mu_{d_y} \end{bmatrix}^t \Sigma_{D_{xy}}^{-1} \begin{bmatrix} d_x - \mu_{d_x} \\ d_y - \mu_{d_y} \end{bmatrix} \right) \quad (6.5)$$

Denoting as d_{min} the minimum distance separation, the conflict probability is calculated integrating the PDF in the region defined by the circle centred at the origin and radius d_{min} :

$$p_{conflict} = \int_{-d_{min}}^{d_{min}} \int_{-\sqrt{d_{min}^2 - d_x^2}}^{\sqrt{d_{min}^2 - d_x^2}} f_{D_{xy}}(d_x, d_y) dd_x dd_y \quad (6.6)$$

In the figure 6.4 there is a graphic example of the problem. There, the region of interest is shown in red i.e. the ranges of less than d_{min} , the contour lines of the PDF is also shown with a σ_x of value 10 and a σ_y of value 5.

Figure 6.4: d_{min} and distance PDF

The problem of directly evaluating the equation 6.6 is that this integral must be solved numerically since it has no analytical expression. This can be a costly and critical process if the probability of conflict for many pairs of targets needs to be quickly assessed.

For certain situations the expression 6.6 can be simplified. If the covariance of the difference of measures fulfils that for both coordinates the covariance is the same ($\sigma_x = \sigma_y$) and also, there is no correlation between them ($\rho = 0$), then the Euclidean norm of $[d_x, d_y]$ follows a Rice distribution [Kay, 1998].

$$\sigma_x = \sigma_y = \sigma \wedge \rho = 0 \Rightarrow d = \sqrt{d_x^2 + d_y^2} \sim R(\nu, \sigma) \quad (6.7)$$

where ν represents the distance between the reference point and the center of the bivariate distribution and is calculated as follow:

$$\nu = \sqrt{d_x^2 + d_y^2} \quad (6.8)$$

The Rice distribution has the following PDF:

$$f_D(d \mid \nu, \sigma) = \frac{d}{\sigma^2} \exp\left(-\frac{(d^2 + \nu^2)}{2\sigma^2}\right) I_0(d\nu/\sigma) \quad (6.9)$$

Therefore, the conflict probability can be obtained in a similar way to the equation 6.6 and it results in the following expression:

$$p_{conflict} = \int_0^{d_{min}} f_{D_{xy}} dd \quad (6.10)$$

where $I_0(z)$ is the modified Bessel function of the first kind with order zero. This integral has an analytical expression and is the following:

$$p_{conflict} = 1 - Q_1\left(\frac{\nu}{\sigma}, \frac{d_{min}}{\sigma}\right) \quad (6.11)$$

where Q_1 is the Marcum Q-function with order one:

$$Q_1(a, b) = \int_b^\infty x \exp\left(-\frac{x^2 + a^2}{2}\right) I_0(ax) dx \quad (6.12)$$

The equation 6.12 is calculated using a defined algorithm that can be consulted in [Shnidman, 1989].

6.2.1.2 Distance between two targets test

Once the distribution is known the only thing left is to determine how the breach of separation is checked. To do so, a hypothesis test [Peebles and Shi, 2001] is established where the null hypothesis H_0 establishes that the two targets are separated greater than the minimum distance (no conflict should be raised, ideally), that we will denote d_{min} , and the alternative hypothesis H_1 the opposite situation:

$$\begin{aligned} H_0 &: \sqrt{d_x^2 + d_y^2} > d_{min} \\ H_1 &: \sqrt{d_x^2 + d_y^2} \leq d_{min} \end{aligned} \quad (6.13)$$

	H_0 is true	H_1 is true
Accept H_0	Correct inference	$1 - P_d$
Reject H_0	P_{fa}	Correct inference

Table 6.1: Hypothesis testing: error types

As it happens with all the contrasts of hypothesis, this test has associated two types of error. On the one hand, the type I error which consists of rejecting the null hypothesis when is the true one. From usual radar notation this error is defined as the false alarm probability denoted by P_{fa} , which consists of detect a target/conflict when there is not.

On the other hand, the type II error consists of accepting H_0 when it is false. Following the same radar notation, this error is the complimentary of the detection probability, denoted by P_d . A summary of these errors can be seen in table 6.1.

Applying this to our test, the probability of detection is the probability of detecting a conflict when the pair of targets is effectively at less distance than the minimum. Instead, the probability of false alarm is given when a conflict is detected with two targets that are at more than the minimum separation. Both types of error are related and cannot be minimized at the same time since improving the probability of detection also implies increasing the probability of false alarm.

To design our conflict detector one of the two parameters will be fixed, P_d or P_{fa} , and a threshold distance that guarantees that this parameter is met has to be calculated. In this test both P_d and P_{fa} depend on the actual distance of the targets. The probability of a false alarm is not the same when the two targets are close to d_{min} (high P_{fa}) than when they are much farther away (low P_{fa}). The same happens with the probability of detection, since if the real distance is very low there is more P_d than if it is close to the minimum distance.

In this design, the probability of detection has been set for the worst-case scenario. This implies that the threshold distance over which the statistical

test would be applied has been calculated in such a way that a given detection probability is guaranteed when the real distance between the targets is the minimum distance. Therefore, to calculate the value of the threshold distance denoted by d_{th} , the expression 6.12 or 6.6 has to be solved with the desired detection probability when the real distance is the minimum. To simplify it, the results will be presented when the conditions to assume a Rice distribution are met, so expression 6.12 will be used. This leads to the following expression:

$$P_d = 1 - Q_1\left(\frac{d_{min}}{\sigma}, \frac{d_{th}}{\sigma}\right) \quad (6.14)$$

Next, the figures 6.5, 6.6 and 6.7 show the detection probability for a minimum distance of 120m, a detection probability of 0.9 and different values of standard deviation σ , specifically 5m, 10m and 20m. From these figures it can be easily inferred that the more uncertainty in the distance between the targets, the greater must be the threshold to be able to detect the conflicts. To put it briefly, the value of the threshold distance is obtained by setting a detection probability in the expression 6.14.

Figure 6.5: Distance threshold and P_d with $\sigma = 5$

Once the statistical threshold of the distance has been determined, d_{th} , the false alarm probability and the detection probability can be calculated. As it has been previously mentioned, said probability depends on the distance

Figure 6.6: Distance threshold and P_d with $\sigma = 10$

Figure 6.7: Distance threshold and P_d with $\sigma = 20$

to which the targets are located as long as it is greater than the threshold distance. Therefore, an expression that relates the real distance of the targets with the probability that the measurements are below the threshold can be found. If the real distance is less than the minimum distance then that probability will be that of detection. On the other hand, if it is above the minimum distance it will be that of false alarm. The expression is the following:

$$\begin{cases} P_d = 1 - Q_1\left(\frac{d_i}{\sigma}, \frac{d_{th}}{\sigma}\right) & d_i \leq d_{min} \\ P_{fa} = 1 - Q_1\left(\frac{d_i}{\sigma}, \frac{d_{th}}{\sigma}\right) & d_i > d_{min} \end{cases} \quad (6.15)$$

where d_i is the real separation between the targets. Next, it is shown

in the figures 6.8, 6.9 and 6.10 the results of detection probability and false alarm probability for the three values of standard deviation previously used. In these graphics it can also be seen that, in fact, when d_i takes the value of d_{min} the probability of detection is 0.9. Obviously, it is a continuous graphic since the probability of detection becomes a false alarm when the real distance of the targets exceeds the minimum distance, therefore and as explained above, the benefits of a value can not be improved without worsening those of the other.

Figure 6.8: Probability of measurements breaking distance threshold with $\sigma = 5$

Figure 6.9: Probability of measurements breaking distance threshold with $\sigma = 10$

Figure 6.10: Probability of measurements breaking distance threshold with $\sigma = 20$

Another way of representing the performance of a conflict detector is to directly compare the detection probability with the false alarm probability. From the equation 6.14 the distance threshold which guarantees a certain detection probability when the targets are separated the minimum distance is calculated. Once the distance threshold has been determined for the selected detection probability, it is also possible to calculate the false alarm probability for targets separated a certain distance greater than the threshold. This false alarm probability is calculated using the equation 6.15. Therefore, the result of this comparison can be seen in the figures 6.11, 6.12 and 6.13. In these figures, the X axis represents the selected detection probability at the threshold, while the Y axis shows the false alarm probability at the distance indicated by each plot. Note that for small standard deviations less distance separations are analysed due to the false alarm probability becomes insignificant.

A summary of the numerical results of the detection probability, the false alarm probability and the distance threshold can be observed in the table 6.2

So far, every performance analysis has been focused on the detection probability or the false alarm probability by using only one measure. Therefore,

Figure 6.11: P_d and P_{fa} relation with $\sigma = 5$

Figure 6.12: P_d and P_{fa} relation with $\sigma = 10$

now the analysis is going to be focused on the performance using multiple measures. In fact, on airport surface surveillance the refresh rate of the targets positions is high, for example SMR provides measures each second. Hence, in order to keep analysing the performance, the cumulative detection probability and the cumulative false alarm probability are going to be calculated.

 Figure 6.13: P_d and P_{fa} relation with $\sigma = 20$

P_d	σ [m]	d_{th} [m]	P_{fa}		
			150 m	200 m	250 m
0.9	5	126.51	1.2e-6	0	0
0.9	10	133.21	0.04	9.7e-12	0
0.9	20	147.13	0.41	3.4e-3	1e-7
0.95	5	128.32	6.71e-6	0	0
0.95	10	136.83	0.088	1.1e-10	0
0.95	20	154.36	0.56	9.7e-3	6.75e-7
0.99	5	131.73	1.2e-4	0	0
0.99	10	143.64	0.25	7.34e-9	0
0.99	20	167.93	0.79	0.048	1.64e-5

 Table 6.2: Values of d_{th} and P_{fa} for different P_d and σ

The cumulative detection probability, denoted by Pd_a , is the probability there is at least one detection in the N last assessments of the conflict, that is, in the last N measures. This probability is complementary to the probability of no detect the conflict in any of the last N measures. In this calculation, in order to simplify the expressions it is assumed that consecutive detections are independent. It is important to remark that this assumption is not always true since for example the consecutive positions obtained by a Kalman filter are correlated. Therefore, assuming independence, the Pd_a could be obtained

as follows:

$$P_{da}(N) = 1 - \prod_{i=1}^N (1 - P_{d_i}) \quad (6.16)$$

where P_{d_i} is the conflict detection probability of the i -th measure. Assuming a constant refresh rate of 1 Hz, equation 6.16 can also be used to calculate the cumulative probability detection as function of how long the potential conflict has been evaluated.

On the other hand, the cumulative false alarm probability, denoted by P_{fa} , is the probability that there is at least one false alarm in N assessment of the conflict. It is calculated in a way equivalent to P_{da} . Assuming again that the consecutive detections are independent, this could be written as:

$$P_{fa}(N) = 1 - \prod_{i=1}^N (1 - P_{fa_i}) \quad (6.17)$$

where P_{fa_i} is the false alarm probability of the i -th measure. Now, two simple scenarios showing the results of P_{fa} and P_{da} are presented.

The first scenario is composed by two targets going on the same runway approaching each other. The minimum distance is 120m and the initial separation is 400m. The two targets approach each other at a relative speed of 5m/s. Since this is a scenario where there will be conflict, the probability of cumulative detection is analysed and calculated with the equation 6.16. The sensor is unbiased and it provides measures with the same standard deviation, 10m, for both Cartesian coordinates. Each target position is estimated by a simple Kalman filter with a constant speed model and null plant noise. Since the covariance of the estimated target positions changes after each filtering, the distance threshold is recalculated for each time instant using the equation 6.14. This scenario is evaluated with a MonteCarlo simulation with 400 iterations.

Figure 6.14 shows the cumulative detection probability for this scenario. Both the P_{da} until instant t_i and the P_d for that time instant can be observed.

In addition, the moment in which the aircraft are at a minimum separation is shown with a vertical line. As expected, the P_{da} grows at a higher rate than the P_d guaranteeing faster conflict detection. It can also be seen that when the aircraft are at the minimum separation the P_d obtained in the simulation (0.91) almost coincides with the desired detection probability (0.9), validating the threshold calculation.

Figure 6.14: P_d and P_{da} for same runway with relative speed of $5m/s$

The second scenario is formed by two targets moving closer but in parallel runways so there is no risk of collision. The runway separations analysed are: 125m, 127m and 130m. The rest of the scenario configuration is the same as in the previous one. Since this is a scenario with no conflict, the cumulative false alarm probability is analysed with the equation 6.17. Figures 6.15, and 6.16 show the result for different runway separation. As expected both probabilities are high as the distance between the runways is very close to the minimum distance.

Although the false alarm probabilities may seem high it should be noted that is the probability or at least one false alarm along the time. Systems usually follow the "M-of-N" rule to avoid this behaviour [Boland and Proshan,

 Figure 6.15: P_{fa} for parallel runways with relative speed of $5m/s$

 Figure 6.16: P_{fa_a} for parallel runways with relative speed of $5m/s$

1983], where this criterion states that at least M detections must be made in the last N updates. For example, in the case that the criterion 2-of-3 is applied in the same runway scenario previously analysed, the detector does not get much worse and it only takes an additional second to reach the imposed detection probability. Specifically, the detection probability in d_{min} would go from 89% to 43% but in the next second reach 93%, as it is shown in the

figure 6.17. In addition, the false alarm probability is considerably reduced, except for the first separation because its P_{fa} is bigger than 5%. But for example for the runways separated 127m is reduced from 32% to 23%, as can be seen in the figure 6.18.

Figure 6.17: P_d with 2-of-3 rule for same runway

Figure 6.18: P_{fa} with 2-of-3 rule for same runway

6.2.2 Separation Safety Net

So far, a hypothesis test that detects breaches of the minimum aircraft separation has been defined. By directly using this test, a safety net can be generated that simply uses the filtered positions of the targets, following an architecture like the one shown in the figure 6.19. This safety net follows a logic like the one shown in the algorithm 1. This safety net calculates the distance between the filtered position of the targets, calculates the distance threshold associated to the desired detection probability and compares them.

Figure 6.19: Safety net without map

Algorithm 1 Separation Safety Net (without map)

```

1: procedure SAFETY NET
2:   for each pair of aircraft do
3:     get Target 1 filtered position and standard deviation
4:     get Target 2 filtered position and standard deviation
5:     calculate targets separation and covariance (equation 6.4)
6:     calculate distance threshold (equation 6.16)
7:     if distance < threshold then
8:       generate alarm

```

Although this safety net has an acceptable performance, it can be easily improved adapting it to use de map information. This improvement drastically reduces the false alarm probability as it will be discussed in the next section.

6.2.2.1 Map-Based Separation Safety Net

First of all, to be able to use the map information the aircraft location on the map is needed. Using the map that was defined in section 2.2, locating in which segment the targets are, is done two ways depending on which tracking filter is implemented. If the tracking filter does not use the map information, such as the explained in section 3.2, it is enough to simply project the estimated filter position, X and Y coordinates, to the nearest segment. This would be equivalent to the location of the first measure on the MBMHF explained in section 2.6. As explained in that section, to locate the nearest segment, it would be enough to project the position of the sector's segments and keep the projection at less distance. With this the position of the robust tracking filter is located on the map. In the case that the filter implemented already exploits the map information, such as the MBMHF, the target location is usually part of the tracking information and therefore, no additional operation is necessary. Graphically, these two situations can be seen in figures 6.20 and 6.21. Their logics are shown in the algorithm 2 and 3 respectively.

Figure 6.20: Robust filter with map safety net

Figure 6.21: Map filter with map safety net

Algorithm 2 Separation Safety Net with map (filter without map)

```

1: procedure SAFETY NET
2:   for each pair of aircraft do
3:     get Target 1 filtered position and standard deviation
4:     locate Target 1 on map
5:     get Target 2 filtered position and standard deviation
6:     locate Target 2 on map
7:     calculate targets separation using the map
8:     calculate distance covariance (equation 6.4)
9:     calculate distance threshold (equation 6.16)
10:    if distance < threshold then
11:      generate alarm

```

Algorithm 3 Separation Safety Net with map (map-based filter)

```

1: procedure SAFETY NET
2:   for each pair of aircraft do
3:     get Target 1 filtered position, standard deviation, and location
4:     get Target 2 filtered position, standard deviation, and location
5:     calculate targets separation using the map
6:     calculate covariance (equation 6.4)
7:     calculate distance threshold (equation 6.16)
8:     if distance < threshold then
9:       generate alarm

```

Applying the map information has a great impact of the performance of this safety net. One way to apply the map information is instead of directly calculating the distance between the two estimated positions it is calculated considering the shortest path between them using the road map. This eliminates false alarms in case of parallel runways. To calculate this shortest path, different algorithms can be used, such as Dijkstra's algorithm [Dijkstra, 1959], A* search algorithm [Lerner et al., 2009] or Floyd–Warshall algorithm [Cormen, 2009]. Since in this case the weight of going from one node to another is the distance between them and it is always positive, Dijkstra's algorithm is used for its simplicity. It should be noted that one way to optimize this calculation is to pre-calculate the distance between all the nodes in the safety net initialization and therefore not having to calculate it in real time. The disadvantage of this approach is that it does not take into account possible

runway closures that cause routes to be changed.

An example of the route calculation can be seen in the figure 6.22 where the positions of two targets are marked with asterisks and the route is drawn with a line. There it can be seen how the real distance separation is very high but a safety net that measures only the Euclidean distance between them could generate a false alarm.

Figure 6.22: Route calculation example at Barcelona El Prat Airport

The Dijkstra's algorithm allows to calculate the distance between a node to another but the targets are not located in the nodes but in segments formed by two nodes. Therefore to calculate the distance between two targets the distance between the 4 combinations of nodes has to be calculated and then the distance from the target to that node is added.

Denoting as A and B the nodes of the segment in which the first target is located, C and D for those of the second target, the route and its distance are calculated for the 4 combinations: A-C, A-D, B-C and B-D. Subsequently, for each of the route lengths, two offsets are applied that take into account the distance of the targets to the origin or destination node. This offset can be positive or negative depending on whether the target is ahead of the node in the path or not. To better illustrate this, an example is given in which it

is assumed the map is the shown in the figure 6.23. In that figure the two crosses mark the position of the targets. When the distance between the A-C nodes is calculated, the Dijkstra algorithm returns the following distance and path:

$$\text{route : } ABC, d_{A-C} = d_{AB} + d_{BC} \quad (6.18)$$

Figure 6.23: Offset route calculation example

The distance d_{A-C} is the distance between the two nodes but not the distance between the two targets. Therefore, two offsets must be added, therefore the separation of the targets by the A-C path is denoted as d_{1-2A-C} . This can be written as the following expression:

$$d_{1-2A-C} = d_{A-C} + offset_1 + offset_2 = d_{AB} + d_{BC} - d_{A1} + d_{C2} \quad (6.19)$$

where d_{A1} is the distance from node A to target 1 and d_{C2} is the distance from node C to target 2. Once the 4 target separations have been calculated for the 4 routes, the one with the shortest distance is chosen. Therefore, and keeping this notation, the minimum distance between the two targets can be calculated as:

$$d_{1-2} = \min(d_{1-2A-C}, d_{1-2A-D}, d_{1-2B-C}, d_{1-2B-D}) \quad (6.20)$$

This distance would be the one that the safety net would later compare to the previously calculated d_{th} , as explained in section on 6.2.1.2. Results analysing the improved performance of this conflict detection depending on if the tracking filter also uses the map information are shown in the following section.

6.3 Separation Safety Net Results

In this section a comparison between the separation safety net with map information and without it is shown. As it has already seen in chapter 2, the map information can be applied not only to the safety net but also to the tracking system. For this reason, the performance of the safety nets is evaluated with four different system configurations. The first configuration uses the robust horizontal tracking without map information and then applies both safety net options: the one that uses the map information and the one that does not. The same proposal is made for the tracking filter that exploits the map information, evaluating the safety net with map and without it. Therefore, the following 4 combinations remain:

1. Robust IMM and Safety Net without map
2. Robust IMM and Safety Net with map
3. MBMHF and Safety Net without map
4. MBMHF and Safety Net with map

Two scenarios are proposed. The first one has two targets approaching each other on parallel tracks and therefore there is no real conflict. In this scenario, false alarms associated with the minimum separation conflict are evaluated. The second scenario has two targets also approaching each other but this time there is real conflict since they are on the same taxiway. In this scenario, the detection probability is evaluated.

Figure 6.24: Two targets trajectory in parallel runways

The first scenario can be seen in the figure 6.24. There the targets move at a constant speed of $5.25m/s$ and the minimum separation distance is 185. It is assumed that in this scenario only the measurements of the MLAT are available. The sensor characteristics were shown in the table 5.3. In addition, it is assumed that there are no global biases or association errors. Finally, it should be noted that the targets move along the centre of the track and therefore the lateral drift in the MBMHF approach is not estimated.

The figure 6.25 shows the measured distances between the two targets in a iteration of the scenario. This distance is the one that is then used by the safety net to compare it with the distance threshold. The first thing is noticed is that safety nets that do not use the map information erroneously measure much less separation between targets. It is also observed that the robust filter, since it does not have the segments location associated to the tracking, presents error peaks associated to erroneous segment localizations.

Figure 6.26 shows the result of safety net alarms calculated in one iteration. As expected, the two safety nets that use the map information do not generate any false alarm while the ones that do not use it generate false

Figure 6.25: Separation Measured between parallel targets

Figure 6.26: Conflict detection with parallel targets

alarms. These results are seen in more detail in the figure 6.27 , in which the scenario is evaluated with a MonteCarlo simulation with 500 iterations.

As expected, the minimum distance safety nets that exploit the map information do not generate any false alarm in the scenario. The only the safety net that has a not null false alarm probability is the one without map and that uses the robust filter . This issue is due to the fact that the two runways are separated 195m and the safety requirement of aircraft minimum

Figure 6.27: Parallel runways MonteCarlo results: (a) Robust Filter + Safety Net without Map; (b) MBMHF + Safety Net without Map; (c) Robust Filter + Safety Net with Map; (d) MBMHF + Safety Net with Map.

separation is 185m. As the robust filter estimated positions oscillate from one side of the runway to the other, sometimes a false alarm is generated. Since the MBMHF positions are in the centre of the runway, no alarms are generated.

Figure 6.28: Two targets trajectory in conflict scenario

The second scenario can be seen in figure 6.28. Target 1 moves at a constant speed of $8m/s$ and target 2 at a speed of $5m/s$. This scenario simulates a collision since the separation between targets reach zero meters. All other assumptions and scenario settings are the same that in the previous case.

The figure 6.29 shows for one iteration, the measured distance by the different safety nets before the comparison with the statistical threshold. As in the previous case, it can be seen the effect of taking into account the road map information and not using a Euclidean distance. It is also observed how the distance measured by the MBMHF is more stable due to the quality of the tracking as well as less error in the location. It should be noted that there is only one real situation of conflict, specifically when one of the targets joins the same taxiway that the other.

Figure 6.29: Measured distance for conflict scenario

Figure 6.30: Measured distance for conflict scenario

Figure 6.30 shows the alarms generated by the safety net in one iteration. In this case, the safety nets using the map are marked with asterisks. Several conclusions can be drawn. First of all, the safety nets that do not use the map information generate the alarm before the conflict actually happens. This is simply due to the geometry of the trajectory and the fact that they use the Euclidean distance. In this particular scenario this is advantageous because it generates the detection of the conflict a few seconds before the

Figure 6.31: Same runway MonteCarlo results: (a) Robust Filter + Safety Net without Map; (b) MBMHF + Safety Net without Map; (c) Robust Filter + Safety Net with Map; (d) MBMHF + Safety Net with Map.

other safety net but, as seen in the previous scenario, in general more false alarms are generated than anticipated detections. On the other hand, it is observed as the safety nets that use the map information do not generate false alarms at the end of the scenario. Hereunder the scenario is evaluated with a MonteCarlo simulation with 500 iterations.

Figure 6.31 shows the results of the detection probability obtained by a MonteCarlo simulation. There each safety net performance is drawn with a line and the ideal result is drawn with a dashed line. It can be observed that the safety nets with map information does not have any false alarm but depending on the tracker being used the detection probability varies. The MBMHF with the map-based safety net has an almost perfect conflict detec-

tion while the robust tracker loses some detections due to the performance of the localization. When the safety net does not exploits the map information, false alarms appears at the end of the scenario.

6.4 Runway Incursion Conflict

In this section the runway incursion conflict detector is designed and analysed. For this purpose, a mathematical analysis is first performed to directly obtain the probability of a target entering an obscured runway. As it was done in section 6.2.2.1, it is proposed to add the map information to the safety net, exploiting the holding points.

6.4.1 Mathematical Analysis

The probability of a target being inside a segment can be calculated directly by simply integrating the PDF of its position along the segment, similar to what was done in the expression 6.6. In this case, the PDF is the one of a single target and it takes the following form:

$$f_{XY}(x, y) = \frac{1}{2\pi|\Sigma_{xy}|^{1/2}} \exp \left(-\frac{1}{2} \begin{bmatrix} x - \mu_x \\ y - \mu_y \end{bmatrix}^t \Sigma_{xy}^{-1} \begin{bmatrix} x - \mu_x \\ y - \mu_y \end{bmatrix} \right) \quad (6.21)$$

where $[\mu_x, \mu_y]$ denotes the real target position in Cartesian coordinates, and Σ denotes the measure covariance matrix, Integrating the expression 6.21 along the segment the probability of invading an occupied segment is obtained:

$$P_{conflict_S} = \int \int_S f_{XY}(x, y) dx dy \quad (6.22)$$

This integral must be solved with numerical methods. Straight segments are a simple case because if the segment is aligned with the horizontal axis, the integral limits can be immediately obtained making easier to calculate

the integral. Therefore, to help with its calculation both the segment as well as the PDF of the target position can be rotated. Given a straight segment with initial coordinates $[x_0, y_0]^t$, final coordinate $[x_1, y_1]^t$, and width w , its four vertices can be calculated with the following expression:

$$S = \begin{bmatrix} x_1 \dots x_4 \\ y_1 \dots y_4 \end{bmatrix} = \begin{bmatrix} x_0 - w/2 \cdot \sin(\theta_s) & y_0 + w/2 \cdot \cos(\theta_s) \\ x_0 + w/2 \cdot \sin(\theta_s) & y_0 - w/2 \cdot \cos(\theta_s) \\ x_1 - w/2 \cdot \sin(\theta_s) & y_1 + w/2 \cdot \cos(\theta_s) \\ x_1 + w/2 \cdot \sin(\theta_s) & y_1 - w/2 \cdot \cos(\theta_s) \end{bmatrix}^T \quad (6.23)$$

where θ_s is the heading angle of the segment. It can be obtained with the following expression:

$$\theta_s = \arctan \frac{y_1 - y_0}{x_1 - x_0} \quad (6.24)$$

The rotation matrix that allows the alignment is calculated as follows:

$$R(\theta) = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (6.25)$$

Therefore, to calculate the probability of conflict in an equivalent way, the rotation matrix has to be applied to the segment coordinates as well as to the covariance matrix of the target PDF. This is achieved with the following expressions:

$$S_{rot} = R(\theta_s) \cdot S \quad (6.26)$$

$$\Sigma'_{xy} = R(\theta_s) \cdot \Sigma_{xy} \cdot R(\theta_s)^t \quad (6.27)$$

The integration limits are obtained from the rotated coordinates and they are the following:

$$\begin{aligned}
 x'_{min} &= \min(x_0 \cos(\theta_s) + y_0 \sin(\theta_s), x_1 \cos(\theta_s) + y_1 \sin(\theta_s)) \\
 x'_{max} &= \max(x_0 \cos(\theta_s) + y_0 \sin(\theta_s), x_1 \cos(\theta_s) + y_1 \sin(\theta_s)) \\
 y'_{min} &= -w/2 + y_0 \cos(\theta_s) - x_0 \sin(\theta_s) \\
 y'_{max} &= w/2 + y_0 \cos(\theta_s) - x_0 \sin(\theta_s)
 \end{aligned} \tag{6.28}$$

Finally, the only thing left to do would be to calculate the probability of being in the wrong segment by using the following expression:

$$p_{conflict_{segment}} = \int_{-x'_{min}}^{x'_{max}} \int_{-y'_{min}}^{-y'_{max}} f'_{XY} dx dy \tag{6.29}$$

The above explained can be seen graphically in the figures 6.32 and 6.33. In the first one, the PDF of a target with $\sigma_X = 10$ and $\sigma_Y = 5$ and an inclined segment can be seen. In the second one, both the segment and the PDF have been rotated, easing the calculation of the integral.

Figure 6.32: Position PDF and segment

Summarizing, applying the rotation matrix 6.28 and the resolving the equation 6.29, the incursion probability is obtained.

Figure 6.33: Rotated position PDF and rotated segment

6.4.1.1 Ground Speed Conflict

Another conflict detector that can be easily implemented is the associated to know if the aircraft are exceeding the maximum allowed speed of the taxiway. This detector is discussed now because in order to improve the detection of runway incursions it will be proposed to make a detector that combines both a ground speed conflict and a distance conflict.

In the ground speed conflict the statistics thresholds are much simpler than the associated to the minimum separation. The speed information of the aircraft is given by the tracking system and therefore the estimated speed and also its standard deviation is available. Similar to the aircraft separation test, this alarm is checked by performing a hypothesis test:

$$\begin{aligned} H_0 : v_m &\leq v_{max} \\ H_1 : v_m &> v_{max} \end{aligned} \tag{6.30}$$

where v_m is the aircraft's speed module and v_{max} the speed limit of the taxiway. To calculate the threshold value of the hypothesis contrast, the following expression must be solved for a confidence level given, α .

$$z_{\alpha} = \frac{v_m - v_{max}}{\sigma_{v_m}} \quad (6.31)$$

where σ_{v_m} is the standard deviation of the speed and $z_{\alpha/2}$ is the critical value for the given confidence level, taking into account the normal distribution. Therefore, conflict will be detected with a given confidence level provided the following condition is met:

$$v_m > v_{max} + z_{\alpha/2} \cdot \sigma_{v_m} \quad (6.32)$$

6.4.2 Runway Incursion Safety Net

So far, a hypothesis test that detects runway incursions has been defined. By directly using this test, a safety net that simply uses the target position and the list of occupied runways can be generated. This safety net follows a logic like the one shown in the algorithm 4.

Algorithm 4 Runway Incursion Safety Net (without map)

```

1: procedure SAFETY NET
2:   for each occupied runway do
3:     get target filtered position and standard deviation
4:     calculate runway rotation matrix (equation 6.28)
5:     calculate runway incursion probability (equation 6.29)
6:     if incursion probability > threshold then
7:       generate alarm
```

Although this safety net has an acceptable performance, it can be easily improved adapting it to use the map information. This improvement drastically reduces the false alarm probability as it will be discussed in the next section.

6.4.2.1 Runway Incursion Map-Based Safety Net

The information on the map can also be used to improve detection performance in runway incursion conflicts. As previously mentioned, the runway incursion consists of the unauthorized presence of vehicles in the aircraft takeoff/landing protected area. In addition, ICAO establishes the requirement that every certified airport must have runway-holding positions since the requirement ICAO 3.12.2 [ICAO, 2004b] said “A runway-holding position or positions shall be established:

1. on the taxiway, at the intersection of a taxiway and a runway;
2. at an intersection of a runway with another runway when the former runway is part of a standard taxi-route”

These runway-holding points are used to prevent aircraft and vehicles penetrating the designated protection zone either side of a runway, which needs to be kept clear during runway use and the vehicles have to be stopped until cleared to proceed by ATC. They are printed in the ground and they have the form as the seen in figure 6.34. More modern approaches to the holding points and stop bars include to show them using augmented reality tools [Arthur III et al., 2009].

Figure 6.34: Runway holding point

These positions can be incorporated into the airport map used by safety net and therefore it can be used to check that the aircraft that are near a

runway holding point and are moving at more than a certain speed will not be able to stop and will skip this point generating a risk situation. Additionally, the clearance obtained by the ATC can be manually introduced as input to the safety net. Therefore, this safety net combines two of the conflict detectors, distance detector explained in section 6.2.1.2), and ground speed detector explained in section 6.4.1.1.

To check the distance to the point a separation test will be done using the information on the map. The only difference from the explained in section 6.2.1.2 is that in this case one of the two targets is fixed and its exact position is known so its noise covariance is zero. The speed test would be equivalent to the one already explained in section 6.4.1.1. Reference values to implement this test could be arise a conflict if the target is less than $15m$ and its speed is greater than $1m/s + k \cdot \sigma_{gs}$. This can be seen in the algorithms 5 and 6.

Algorithm 5 Runway Incursion Safety Net with map (filter without map)

```

1: procedure SAFETY NET
2:   for each occupied runway do
3:     get target filtered position and standard deviation
4:     locate target on map
5:     calculate target separation to the holding point
6:     calculate distance and ground speed thresholds
7:     if distance < threshold1 and ground speed > threshold2 then
8:       generate holding point alarm
9:       calculate runway rotation matrix (equation6.28)
10:      calculate runway incursion probability (equation6.29)
11:      if incursionprobability > threshold then
12:        generate runway intrusion alarm

```

Finally, the advantage of this safety net implementation is that it can detect a runway incursion well before it occurs because runway-holding points have to be located as shown in the table 6.3 obtained from the ICAO requirement 3.12.6 [ICAO, 2004b]. In that table Code Number refers to the ICAO Aerodrome Reference Code.

Algorithm 6 Runway Incursion Safety Net with map (map-based filter)

```

1: procedure SAFETY NET
2:   for each occupied runway do
3:     get target filtered position, standard deviation and location
4:     calculate target separation to the holding point
5:     calculate distance and ground speed thresholds
6:     if distance < threshold1 and ground speed > threshold2 then
7:       generate holding point alarm
8:       calculate runway rotation matrix (equation6.28)
9:       calculate runway incursion probability (equation6.29)
10:      if incursionprobability > threshold then
11:        generate runway intrusion alarm

```

Type of Runway	Code Number			
	1	2	3	4
Non-instrument	30 m	40 m	75 m	75 m
Non-precision approach	40 m	40 m	75 m	75 m
Precision approach category I	60 m	60 m	90 m	90 m
Precision approach category II and III	-	-	90 m	90 m
Take-off runway	30 m	40 m	75 m	75 m

Table 6.3: Minimum distance from the runway centre line to a holding bay, runway-holding position or road-holding position

6.5 Runway Incursion Safety Net Results

The third scenario can be seen in the figure 6.35. In this scenario a target is headed towards a runway when it is already being used by another aircraft. This target moves at speed of $4m/s$ and will pass through a holding point at which it will not stop. This will generate a conflict situation. All other assumptions and scenario settings are the same at the previous ones (section 6.3). In order to detect the conflict caused by not respecting a holding point, it is necessary to have information from the map in this scenario.

In this scenario the available tracking systems are the MBMHF approach using the safety net with the map information and the robust IMM filter

Figure 6.35: Trajectory for runway intrusion scenario

using the safety net without this information. The reason for using only these two systems is that in this scenario the results are similar to the other two combinations. Obviously, the safety net that does not have the map information cannot detect the fact of skipping the holding point, it can simply arise the alarm associated with a target entering an occupied runway. Figure 6.36 shows the results of these alarms obtained in one iteration.

Figure 6.36: Runway intrusion alarms in one iteration

Lastly, the scenario is analysed using a MonteCarlo simulation and the detection probabilities are obtained, as the 6.37 shows. First, it can be seen that the safety net without map information, i.e. the one that only detects if the target is inside the occupied runway, has practically the same features for both filters. This is due to the fact that the MBMHF and the robust filter have a similar longitudinal performance when the target is not manoeuvring. On the other hand, the performance of safety net based on the holding point can be observed. As expected, it correctly detects that the target has not stopped at the holding point which means that the target has not waited to the clearance. This detection makes possible to detect the runway incursion in advance.

Figure 6.37: Runway intrusion alarms in a MonteCarlo

Part III

Conclusions and Bibliography

Chapter 7

Conclusions

In this Ph.D. thesis, a system able to perform some ATC functionalities on the airport area has been designed and analysed. To be more specific, the system focuses on both the surveillance and the control functions of the A-SMGCS.

The contributions to the surveillance function are divided into two sub-blocks: aircraft tracking, and bias estimation and correction. In the first field, a tracking system that processes the horizontal measures in parallel with a robust filter and another tracking system that exploits the map information have been designed. The robust filter follows a classic approach using an IMM filter. The second filter uses the map information to project the measures into the centre of the road and then performs a decoupled tracking of the longitudinal and transverse components. The vertical measurement tracking also follows a similar logic but the map information can only be applied when aircraft are landing following the glide path.

Regarding the bias estimation, a system capable of correcting three of the most common biases on the airport area has been designed and evaluated. The estimated biases are the sensor dependent bias (global bias), the target-sensor pair dependent bias (local bias), and the vertical offset between geometric measures and barometric ones.

Concerning the control function, a safety net capable of detecting different conflicts has been designed and evaluated. Specifically, the detected conflicts are the ones related with the minimum separation distance, taxiway speed excess, and runway incursions. The designed safety net can work either by simply using the target position provided by the surveillance system or by also using the map information.

In the following section the conclusions of each thesis chapter are deeply discussed.

7.1 Discussion

In chapter 2 the horizontal map-based filter and its vertical equivalent are designed. The motivation for using the map information is clear: on the airport surface, not only do planes have to follow the tracks but they also have to try to go through their centre. This restricts the freedom of movement of the aircraft which can be used by the tracker. Therefore, first it is proposed to represent the airport map using three bidimensional basic elements: straight lines, circular arcs, and Bézier splines. These three kinds of segments are generally enough to represent any map with precision. Regarding the target tracking, this filter decouples the tracking in two components: one related to along-road movement, and another one designed to track transversal movement. The along-road tracking assumes that the target moves parallel to the centre line of the road, focusing in tracking the projection of target position on this centre line. Therefore, it preprocesses the measures by projecting them to this centre line. The transversal movement (perpendicular to this centre line) is estimated using as input the difference between the sensor measure and the projected measure. This decoupled movement has involved the design of expressions that allow to project the measurements to the segments, separating the longitudinal and transversal component of the measurements. Once the two components are obtained, the longitudinal one is processed by IMM filter optimized for longitudinal tracking, while the transverse component is processed by a Kalman filter. Since this tracker uses the road map information, it has to be able to deal with the intersections. For this pur-

pose, it is proposed an approach that opens up different hypotheses at the road intersections, each one indicating a possible route, i.e. following a Multiple Hypothesis Tracker (MHT) approach. Then, the tracker calculates the probability of each hypothesis and combines them to obtain the filtered longitudinal position. Once the longitudinal position is obtained, it is projected to Cartesian coordinates and the estimated lateral offset, also projected to Cartesian coordinates, is added to obtain the final position.

The use of map information has also been applied to vertical movement. Clearly in this situation the map information can only be applied in a very specific case which is when the aircraft are landing following the glide path. In that situation, the vertical speed of the aircraft is proportional to the horizontal speed and this can be exploited by the surveillance system. In this thesis, several ways of exploiting this information are exposed and it has been ultimately decided to generate pseudo vertical velocity measurements from the filtered horizontal velocities and the glide path angle. After that, the measures are filtered using a IMM approach.

The horizontal and vertical map-based filters cannot be the only ones in charge of the surveillance on the airport area because, although they have better performance than other approaches, they do not fully cover the possible situations.. In the horizontal case, these situations would be aircraft that are not following roads, i.e. aircraft that have already taken off, that have left the road or when the map does not contain all the roads. In the vertical case, as the map filter is only suitable for aircraft landing following the glide path, the rest of the cases are not covered by this system. For these situations, in chapter 3 robust filters that do not use any kind of map information are proposed to work in parallel with these map-based filters and thus they serve as backups of the tracking in the previous explained situations. These robust filters follow a classic IMM approach.

In chapter 4, a system for estimating and correcting the sensor biases is designed and discussed. The designed biases corrections focus on the three components that have the greatest impact on the airport area. The horizontal position measurements are affected by biases associated with the sensor (denoted as global biases) and those associated with the target (denoted as

local biases). To be more specific, the global biases are those that are sensor-dependent and independent of the target they measure while local biases are those that depend on the target. The third type of bias that is estimated is the barometric offset, it is presented in vertical measurements and caused by the fact that some sensors calculate height using geometric sources and others do it using barometric information.

Regarding the way to estimate and correct the biases in this thesis, the following approach is followed. When an already associated measure is received, it is divided into its horizontal and vertical parts. In the case of horizontal measures, first their global bias is corrected and then the same is done with their local bias. The estimation of the global biases is obtained from differences in measurements from two sources: reference targets and track traces. The reference targets are reflectors or transponders which position is known with precision, while the track traces are a unique estimation of the bias made from the tracking system. Regarding the estimation of local biases, it is done once the global biases have been corrected and it also uses measurements differences of pairs of sensors that are measuring the same target. Once both biases have been corrected, the horizontal measure is sent to the tracking system.

Vertical measurements go through a simpler bias correction system since they only need to be corrected from the offset associated with using both geometric and barometric height measurements. The architecture of this correction (section 4.3.2) consists in subtracting the offset before sending the measurements to the vertical tracker. The estimation is also done by the difference of pairs of geometrical and barometric measures and applying a Kalman filter.

In chapter 5 the performance results of the designed surveillance function are discussed, analysing both the biases estimation and the tracking system. First, the correction of biases is evaluated, in a scenario in which all the biases are present. Therefore, it can be concluded that the biases estimator is capable of correcting them properly, allowing the tracking system to receive corrected measurements. Regarding the surveillance system, both robust and map-based filters are analysed when they work separately and when they

work together. The performance results are obtained using simulated data and real data. In addition, their performance is compared with other existing approaches in the literature. The implemented robust filter has an acceptable performance with the typical error peaks associated with the manoeuvres. The map-based filter shows a slightly better longitudinal performance than other approaches in the literature and than the robust filter but above all improves in the transversal estimation and therefore, it reduces the error in turns. On the other hand, when the target arrives at intersection when it keeps moving straight forward the map-based filter has short-time peaks of distance error. This is caused to the fact that at each intersection the filter opens new hypotheses associated to each possible route and until the hypothesis associated to moving straight forward becomes dominant, the position performance worsens. When both filters are working in parallel then the quality of the system is noticeably better than other solutions, since the junction peaks explained before are eliminated using the robust filter as output in these situations, and the map-based filter performs better in the rest of situations.

In chapter 6, a safety net capable of detecting different conflicts has been designed and evaluated. To be more specific, this safety net detects conflicts associated with breaching the minimum separation between targets, taxiway speed excesses, and runway incursions. These detectors can both work with only the sensors measurements or also by using the map model defined in chapter 2 to improve their performance. The use of the map information implies that if the tracking system is not map-based, it is necessary to add a logic to locate the estimated target positions on the map. For the conflict of not keeping the minimum separation, the use of the map allows to calculate with much greater precision the real separation since it takes into account the possible routes between the targets and the geometry of the roads. In the case of the runway incursion, the use of the map allows to also detect the targets that are not respecting the holding points of the runway, allowing a prior conflict detection and increasing the response margin to solve the conflict.

7.2 Contributions

This thesis presents a surveillance system for the airport area capable of correcting the most common biases and accurately tracking the targets. The architecture of this system can be seen in figure 7.1. Regarding the bias estimation, this thesis parts from the approach discussed in [Besada et al., 2012] and extends it to correct the existing biases on the airport area and its sensors. Therefore, the system is capable of differentiating between two horizontal biases: local bias and global bias. In addition, the system corrects the vertical offset between barometric and geometric measures. Each estimator is optimized for one specific bias and hence, each one models processes with different decorrelation times.

Regarding the tracking system, this thesis proposes to use an architecture where a robust filter and a map-based filter run in parallel. The map-based filter includes an estimator that has to estimate the drift between the real target position and the centre of the road, using as input the difference between the sensor measure and the projected measure. The map-based filter has a slightly better longitudinal performance whereas its transversal performance is considerably better than other approaches. In addition, the robust filter allows to obtain the estimated position of targets that do not follow the road map. This performance is analysed using real and simulated data, showing the benefits of this architecture over other proposals in the literature for tracking.

Several key ideas are the main responsible of this benefits over other approaches. First, the adaptation of the airport map has been improved from other approaches that only use straight lines or circular arcs, by adding the use of Bézier splines to the typical segments: straight lines and arcs of circumference. This allows to represent the road map with more freedom and precision. Also, a better performance of the longitudinal tracking has been achieved by using projected measures to the road centre and a multiple hypotheses approach capable of tracking all the possible longitudinal trajectories in the intersections. The system is also capable of distinguishing the lateral drift of the local bias and the global bias. Regarding the use of the

Figure 7.1: Proposed surveillance architecture

map in the vertical surveillance, exploiting the glide path information, also improves the tracking performance when the aircraft are landing.

This thesis also presents a safety net that exploits the road map defined for the surveillance function and uses it in order to improve the performance. This architecture can be seen in figure 7.2. There, the different options, depending if the tracker provides the target location or not, are shown. The use of the map allows to speed up the conflict detection, drastically reduce

Figure 7.2: Diagram of a safety net exploiting the map information: (a) tracker with target localization, (b) tracker without target localization

the false alarm probability on the safety net and detect new conflicts, e.g. the related with not respecting the holding points.

7.3 Further Work

Future work should investigate the capability to deal with erroneous maps and to incorporate additional information to the map road such as closed roads, one-way roads, etc. This could reduce the effect of the junctions into the accuracy of the map-based filters.

An obvious limitation of the map-based filter is that the described approach is only valid when neither the measurement noise nor the target movement make measurements jump from a given segment area to the surroundings of another segment not connected with the previous one. As the hypothesis formation procedure just looks in connected segments, the filter

would fail if this kind of jump is present. This results in limitations linking the maximum speed of the target, the measurement noise variances, and the minimum length of segments. Also, limits on the minimum radius of curves related to measurement noise variances, could appear.

Another line of investigation could be the addition of different sensor technologies to the developed tracker such as cameras or magnetic sensors. Further work should also cover the estimation of more global biases such as the related with MLAT or ADS-B.

Bibliography

- [Abbud and De Miguel, 2014] Abbud, J. and De Miguel, G. (2014). Joint target tracking and systematic error correction for wide area multilateration. In *Air Traffic Management and Systems*, pages 175–191. Springer.
- [Abbud et al., 2011] Abbud, J. M., Miguel, G. d., and Besada, J. (2011). Correction of systematic errors in wide area multilateration. In *2011 Tyrrhenian International Workshop on Digital Communications - Enhanced Surveillance of Aircraft and Vehicles*, pages 173–178.
- [Agate and Sullivan, 2003] Agate, C. S. and Sullivan, K. J. (2003). Particle filtering algorithm for tracking multiple road-constrained targets. In *Signal Processing, Sensor Fusion, and Target Recognition XII*, volume 5096, pages 256–267. International Society for Optics and Photonics.
- [Arthur III et al., 2009] Arthur III, J. T. J., Prinzel III, L. J., Shelton, K. J., Kramer, L. J., Williams, S. P., Bailey, R. E., and Norman, R. M. (2009). Synthetic vision enhanced surface operations with head-worn display for commercial aircraft. *The International Journal Of Aviation Psychology*, 19(2):158–181.
- [Bain and Engelhardt, 1987] Bain, L. J. and Engelhardt, M. (1987). *Introduction to probability and mathematical statistics*. Brooks/Cole.
- [Bar-Shalom, 1990] Bar-Shalom, Y. (1990). Multitarget-multisensor tracking: advanced applications. *Norwood, MA, Artech House, 1990, 391 p.*
- [Bar-Shalom and Birmiwal, 1982] Bar-Shalom, Y. and Birmiwal, K. (1982). Variable dimension filter for maneuvering target tracking. *IEEE transactions on Aerospace and Electronic Systems*, (5):621–629.
- [Bar-Shalom et al., 1990a] Bar-Shalom, Y., Campo, L., and Luh, P. (1990a). From receiver operating characteristic to system operating characteristic: Evaluation of a track formation system. *IEEE Transactions on Automatic Control*, 35(2):172–179.

- [Bar-Shalom et al., 1990b] Bar-Shalom, Y., Fortmann, T. E., and Cable, P. G. (1990b). *Tracking and data association*. ASA.
- [Bar-Shalom and Li, 1993] Bar-Shalom, Y. and Li, X.-R. (1993). Estimation and tracking- principles, techniques, and software. *Norwood, MA: Artech House, Inc, 1993*.
- [Bar-Shalom and Li, 1995] Bar-Shalom, Y. and Li, X.-R. (1995). *Multitarget-multisensor tracking: principles and techniques*, volume 19. YBs London, UK:.
- [Bar-Shalom et al., 2004] Bar-Shalom, Y., Li, X. R., and Kirubarajan, T. (2004). *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons.
- [Bar-Shalom et al., 2011] Bar-Shalom, Y., Willett, P. K., and Tian, X. (2011). *Tracking and data fusion*. YBS publishing.
- [Besada et al., 2012] Besada, J., De Miguel, G., Bernardos, A., and Casar, J. (2012). Generic multisensor multitarget bias estimation architecture. *IET Radar, Sonar & Navigation*, 6(5):365–378.
- [Besada et al., 2004] Besada, J., Herrero, J. G., and de Miguel Vela, G. (2004). New approach to online optimal estimation of multisensor biases. *IEE Proceedings-Radar, Sonar and Navigation*, 151(1):31–40.
- [Besada et al., 2005a] Besada, J. A., García, J., Portillo, J., Molina, J. M., Varona, A., and Gonzalez, G. (2005a). Airport surface surveillance based on video images. *IEEE transactions on aerospace and electronic systems*, 41(3):1075–1082.
- [Besada et al., 2014] Besada, J. A., López-Araquistain, J., de Miguel, G., and Soto, A. (2014). Performance requirements specification design for airport surface surveillance. In *Digital Communications-Enhanced Surveillance of Aircraft and Vehicles (TIWDC/ESAV), 2014 Tyrrhenian International Workshop on*, pages 23–28. IEEE.
- [Besada et al., 2005b] Besada, J. A., Soto, A., de Miguel, G., and Portillo, J. (2005b). Design of an a-smgcs prototype at barajas airport: airport surveillance sensors bias estimation. In *Information Fusion, 2005 8th International Conference on*, volume 2, pages 8–pp. IEEE.
- [Bhaumik et al., 2013] Bhaumik, S. et al. (2013). Cubature quadrature kalman filter. *IET Signal Processing*, 7(7):533–541.

- [Biella et al., 2015] Biella, M., Hahn, K., Zilz, J., Lafferton, H., Vieten, B. D., and Danello, F. (2015). Follow-the-greens: Towards increased safety and efficiency by the use of airfield ground lighting (agl). results from a sesar real time simulation. In *3rd International Conference on Human Factors in Transportation*.
- [Biella et al., 2013] Biella, M., Straube, K., Helms, M., Straub, S., Weiß, B., Schmitt, F., Lafferton, H., Dubuisson, S., and Lane, R. (2013). How can a future safety net successfully detect conflicting atc clearances—yet remain inconspicuous to the tower runway controller? first results from a sesar exercise at hamburg airport. In *International Conference on Engineering Psychology and Cognitive Ergonomics*, pages 65–74. Springer.
- [Bilimoria et al., 2000] Bilimoria, K., Lee, H., Mao, Z.-H., and Feron, E. (2000). Comparison of centralized and decentralized conflict resolution strategies for multiple-aircraft problems. In *18th Applied Aerodynamics Conference*, page 4268.
- [Blackman, 1986] Blackman, S. S. (1986). Multiple-target tracking with radar applications. *Dedham, MA, Artech House, Inc., 1986, 463 p.*
- [Blackman, 2004] Blackman, S. S. (2004). Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1):5–18.
- [Blasch et al., 2014] Blasch, E., Wang, Z., Shen, D., Ling, H., and Chen, G. (2014). Surveillance of ground vehicles for airport security. In *SPIE Defense+ Security*, pages 90890B–90890B. International Society for Optics and Photonics.
- [Bloem et al., 2001] Bloem, E. A., Blom, H. A. P., and van Schaik, F. J. (2001). Advanced data fusion for airport surveillance. In *Proceedings JISSA 2001 International Conference on Airport Surveillance Sensors*. Citeseer.
- [Blom and Bar-Shalom, 1988] Blom, H. A. and Bar-Shalom, Y. (1988). The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE transactions on Automatic Control*, 33(8):780–783.
- [Boland and Proschan, 1983] Boland, P. J. and Proschan, F. (1983). The reliability of k out of n systems. *The Annals of Probability*, pages 760–764.
- [Brinton et al., 2007] Brinton, C. R., Atkins, S. C., and Hall, A. (2007). Analysis of taxi conformance monitoring algorithms and performance.

- In *Integrated Communications, Navigation and Surveillance Conference, 2007. ICNS'07*, pages 1–8. IEEE.
- [Caicai and Wei, 2011] Caicai, G. and Wei, C. (2011). Ground moving target tracking with VS-IMM using mean shift unscented particle filter. *Chinese Journal of Aeronautics*, 24(5):622–630.
- [Cassell et al., 2000] Cassell, R., Evers, C., and Yang, Y. E. (2000). Pathprox-a runway incursion alerting system. In *Digital Avionics Systems Conference, 2000. Proceedings. DASC. The 19th*, volume 2, pages 7D3–1. IEEE.
- [Cassell and Smith, 1995] Cassell, R. and Smith, A. (1995). Development of required navigation performance (rnp) requirements for airport surface movement guidance and control. In *Proceedings of 14th Digital Avionics Systems Conference*, pages 57–64.
- [Chen et al., 2000] Chen, H., Kirubarajan, T., and Bar-Shalom, Y. (2000). Comparison of centralized and distributed tracking algorithms using air-to-air scenarios. In *Signal and Data Processing of Small Targets 2000*, volume 4048, pages 440–452. International Society for Optics and Photonics.
- [Cheng and Singh, 2007] Cheng, Y. and Singh, T. (2007). Efficient particle filtering for road-constrained target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 43(4).
- [Cherchar et al., 2010] Cherchar, A., Belouchrani, A., and Chonavel, T. (2010). A new approach in distributed multisensor tracking systems based on kalman filter methods. In *Information Fusion (FUSION), 2010 13th Conference on*, pages 1–8. IEEE.
- [Cho et al., 2013] Cho, T., Lee, C., and Choi, S. (2013). Multi-sensor fusion with interacting multiple model filter for improved aircraft position accuracy. *Sensors*, 13(4):4122–4137.
- [Clare and Richards, 2011] Clare, G. and Richards, A. G. (2011). Optimization of taxiway routing and runway scheduling. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1000–1013.
- [Cormen, 2009] Cormen, T. H. (2009). *Introduction to algorithms*. MIT press.
- [Cox, 1964] Cox, H. (1964). On the estimation of state variables and parameters for noisy dynamic systems. *IEEE Transactions on automatic control*, 9(1):5–12.

- [Cox, 1993] Cox, I. J. (1993). A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10(1):53–66.
- [Cuiping et al., 2010] Cuiping, L., Jinping, S., Shiyi, M., and Desheng, L. (2010). Tracking ground targets with road constraint using multiple hypotheses tracking. In *Signal Processing Systems (ICSPS), 2010 2nd International Conference on*, volume 2, pages V2–265. IEEE.
- [Delille and Muller, 2010] Delille, F. and Muller, D. (2010). Advanced solutions for integrated surveillance. In *Integrated Communications Navigation and Surveillance Conference (ICNS), 2010*, pages 1–16. IEEE.
- [Dijkstra, 1959] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.
- [Donohue, 1995] Donohue, G. (1995). A visionary look at aviation surveillance systems. *IEEE Aerospace and Electronic Systems Magazine*, 10(10):8–14.
- [Dubuisson et al., 2004] Dubuisson, S., Lane, R., Bonnier, M., Hermand, H., and Biella, M. (2004). Updated osed for conflicting atc clearances sesar p6.7.1.
- [Ellejmi, 2016] Ellejmi, M. (2016). Evaluation of sesar concepts for airport guidance function in live trial environment. In *Integrated Communications Navigation and Surveillance (ICNS), 2016*, pages 1–32. IEEE.
- [EUROCONTROL, 2003] EUROCONTROL (2003). Principles of mode s operation and interrogator codes.
- [EUROCONTROL, 2009a] EUROCONTROL (2009a). EUROCONTROL guidance material for minimum safe altitude warning.
- [EUROCONTROL, 2009b] EUROCONTROL (2009b). EUROCONTROL specification for short term conflict alert. Rev. 1.1.
- [EUROCONTROL, 2010] EUROCONTROL (2010). Eurocontrol operational concept and requirements for a-smgcs implementation.
- [EUROCONTROL, 2017a] EUROCONTROL (2017a). EUROCONTROL guidelines for approach path monitor.
- [EUROCONTROL, 2017b] EUROCONTROL (2017b). EUROCONTROL guidelines for area proximity warning.

- [EUROCONTROL, 2018] EUROCONTROL (2018). Eurocontrol specification for advanced-surface movement guidance and control system (a-smgcs) services.
- [FAA, 2011] FAA (2011). Introduction to TCAS II version 7.1.
- [Farina et al., 2003] Farina, A., Ferranti, L., and Golino, G. (2003). Constrained tracking filters for a-smgcs. In *Proceedings of the sixth international conference of information fusion*, volume 1, pages 414–421.
- [Gad et al., 2002] Gad, A., Majdi, F., and Farooq, M. (2002). A comparison of data association techniques for target tracking in clutter. In *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, volume 2, pages 1126–1133. IEEE.
- [Galati et al., 2010] Galati, G., Leonardi, M., Cavallin, A., and Pavan, G. (2010). Airport surveillance processing chain for high resolution radar. *IEEE Transactions on Aerospace and Electronic Systems*, 46(3):1522–1533.
- [Galati et al., 2005] Galati, G., Leonardi, M., Magarò, P., and Paciucci, V. (2005). Wide area surveillance using ssr mode s multilateration: advantages and limitations. In *Radar Conference, 2005. EURAD 2005. European*, pages 225–229. IEEE.
- [Galati et al., 2008] Galati, G., Leonardi, M., and Tosti, M. (2008). Multilateration (local and wide area) as a distributed sensor system: lower bounds of accuracy. In *Radar Conference, 2008. EuRAD 2008. European*, pages 196–199. IEEE.
- [Galati et al., 2004] Galati, G., Magarò, P., Leonardi, M., and Cavallin, A. (2004). Data extraction in high resolution surface movement radar for a-smgcs. *Proceedings IRS 2004*, pages 19–21.
- [Galati et al., 1999] Galati, G., Naldi, M., and Ferri, M. (1999). Airport surface surveillance with a network of miniradars. *IEEE Transactions on Aerospace and Electronic Systems*, 35(1):331–338.
- [Gao et al., 2009] Gao, H., Heuer, T., Dimitropoulos, K., Grammalidis, N., Weinmann, M., Huhnold, M., Astheimer, T., Kirrane, P., Stockhammer, C., Kurz, A., et al. (2009). Safe airport operation based on innovative magnetic detector system. *IET Intelligent Transport Systems*, 3(2):236–244.

- [García et al., 2005] García, J., Molina, J. M., Besada, J. A., and Portillo, J. I. (2005). A multitarget tracking video system based on fuzzy and neuro-fuzzy techniques. *EURASIP Journal on Advances in Signal Processing*, 2005(14):640283.
- [Gattein et al., 2005] Gattein, S., Pannetier, B., and Vannoorenberghe, P. (2005). Analysis and integration of road projection methods for multiple ground target tracking. In *Information Fusion, 2005 8th International Conference on*, volume 1, pages 8–pp. IEEE.
- [Gelb, 1974] Gelb, A. (1974). *Applied optimal estimation*. MIT press.
- [Gustafsson et al., 2002] Gustafsson, F., Gunnarsson, F., Bergman, N., Forsell, U., Jansson, J., Karlsson, R., and Nordlund, P.-J. (2002). Particle filters for positioning, navigation, and tracking. *IEEE Transactions on signal processing*, 50(2):425–437.
- [Hao and Huang, 2009] Hao, Z. and Huang, S. (2009). Data fusion approach based on high precision barometric altimeter and gps. In *Intelligent Systems and Applications, 2009. ISA 2009. International Workshop on*, pages 1–4. IEEE.
- [Harris, 1956] Harris, K. (1956). Some problems of secondary surveillance radar systems. *Journal of the British Institution of Radio Engineers*, 16(7):355–382.
- [Hasberg et al., 2008] Hasberg, C., Hensel, S., Westenkirchner, M., and Bach, K. (2008). Integrating spline curves in road constraint object tracking. In *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pages 1009–1014. IEEE.
- [He et al., 2013] He, Y., Zhu, H., and Tang, X. (2013). A joint maximum likelihood algorithm for simultaneous track correlation and sensor bias estimation. *Science China Information Sciences*, 56(10):1–11.
- [Hempe, 2010] Hempe, D. (2010). Airworthiness approval of automatic dependent surveillance-broadcast (ads-b) out systems. *US Department of Transportation Federal Aviation Administration, AC-20-165*.
- [Herrero et al., 2007] Herrero, J. G., Portas, J. A. B., and Corredera, J. R. C. (2007). On-line multi-sensor registration for data fusion on airport surface. *IEEE Transactions on Aerospace and Electronic Systems*, 43(1).

- [Herrero et al., 2003] Herrero, J. G., Portas, J. B., and Corredera, J. C. (2003). Use of map information for tracking targets on airport surface. *IEEE Transactions on Aerospace and Electronic Systems*, 39(2):675–693.
- [Herrero et al., 2001] Herrero, J. G., Portas, J. B., Rodriguez, F. J., and Corredera, J. C. (2001). Surface movement radar data processing methods for airport surveillance. *IEEE Transactions on Aerospace and Electronic Systems*, 37(2):563–585.
- [Hicok and Lee, 1998] Hicok, D. S. and Lee, D. (1998). Application of ads-b for airport surface surveillance. In *Digital Avionics Systems Conference, 1998. Proceedings., 17th DASC. The AIAA/IEEE/SAE*, volume 2, pages F34–1. IEEE.
- [Hofmann-Wellenhof et al., 2012] Hofmann-Wellenhof, B., Lichtenegger, H., and Collins, J. (2012). *Global positioning system: theory and practice*. Springer Science & Business Media.
- [IATA, 2017] IATA (2017). IATA safety report 2016 53rd edition.
- [ICAO, 1986] ICAO (1986). Manual of surface movement guidance and control systems (smgcs) (doc 9476).
- [ICAO, 1996] ICAO (1996). Radio navigation aids, annex 10 to the convention on international civil aviation.
- [ICAO, 2001] ICAO (2001). Air traffic management, doc 4444.
- [ICAO, 2004a] ICAO (2004a). Advanced surface movement control and guidance systems (a-smgcs) manual, 408 doc 9830 an/452.
- [ICAO, 2004b] ICAO (2004b). Aerodrome design and operations, annex 14 to the convention on international civil aviation.
- [ICAO, 2005] ICAO (2005). Global air traffic management operational concept, doc 9854 an/458.
- [Jarama et al., 2017] Jarama, Á. J., López-Araquistain, J., Miguel, G. d., and Besada, J. A. (2017). Complete systematic error model of ssr for sensor registration in atc surveillance networks. *Sensors*, 17(10):2171.
- [Jilkov et al., 1999] Jilkov, V., Angelova, D., and Semerdjiev, T. A. (1999). Design and comparison of mode-set adaptive imm algorithms for maneuvering target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 35(1):343–350.

- [Jing and Vadakkepat, 2010] Jing, L. and Vadakkepat, P. (2010). Interacting mcmc particle filter for tracking maneuvering target. *Digital Signal Processing*, 20(2):561–574.
- [Jones, 2009] Jones, S. R. (2009). Ads-b surveillance separation error sensitivity analysis. *Mitre Product MP*, 90254.
- [Julier and Uhlmann, 2004] Julier, S. J. and Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422.
- [Kalman, 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45.
- [Kalman and Bucy, 1961] Kalman, R. E. and Bucy, R. S. (1961). New results in linear filtering and prediction theory. *Journal of basic engineering*, 83(1):95–108.
- [Kastella and Biscuso, 1995] Kastella, K. and Biscuso, M. (1995). Tracking algorithms for air traffic control applications. *Air Traffic Control Quarterly*, 3(1):19–43.
- [Kay, 1993] Kay, S. M. (1993). *Fundamentals of statistical signal processing, volume I: estimation theory*. Prentice Hall.
- [Kay, 1998] Kay, S. M. (1998). *Fundamentals of statistical signal processing: Detection theory, vol. 2*. Prentice Hall Upper Saddle River, NJ, USA:.
- [Keith et al., 2008] Keith, G., Richards, A., and Sharma, S. (2008). Optimization of taxiway routing and runway scheduling. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, page 6827.
- [Khaleghi et al., 2013] Khaleghi, B., Khamis, A., Karray, F. O., and Razavi, S. N. (2013). Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28–44.
- [Kirubarajan et al., 1998] Kirubarajan, T., Bar-Shalom, Y., Pattipati, K., Kadar, I., Abrams, B., and Eadan, E. (1998). Tracking ground targets with road constraints using an IMM estimator. In *Aerospace Conference, 1998 IEEE*, volume 5, pages 5–12. IEEE.
- [Kirubarajan et al., 2000] Kirubarajan, T., Bar-Shalom, Y., Pattipati, K. R., and Kadar, I. (2000). Ground target tracking with variable structure IMM estimator. *IEEE Transactions on Aerospace and Electronic Systems*, 36(1):26–46.

- [Konstantinova et al., 2003] Konstantinova, P., Udvarov, A., and Semerdjiev, T. (2003). A study of a target tracking algorithm using global nearest neighbor approach. In *Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech'03)*, pages 290–5.
- [Kuchar and Drumm, 2007] Kuchar, J. and Drumm, A. C. (2007). The traffic alert and collision avoidance system. *Lincoln Laboratory Journal*, 16(2):277.
- [Kuchar and Yang, 2000] Kuchar, J. K. and Yang, L. C. (2000). A review of conflict detection and resolution modeling methods. *IEEE Transactions on intelligent transportation systems*, 1(4):179–189.
- [Lee et al., 1982] Lee, T., Dunn, K., and Chang, C. (1982). On observability and unbiased estimation of nonlinear systems. In *System Modeling and Optimization*, pages 258–266. Springer.
- [Lerner et al., 2009] Lerner, J., Wagner, D., and Zweig, K. (2009). *Algorithmics of large and complex networks: design, analysis, and simulation*, volume 5515. Springer.
- [Leung et al., 1996] Leung, H., Hu, Z., and Blanchette, M. (1996). Evaluation of multiple target track initiation techniques in real radar tracking environments. *IEEE Proceedings-Radar, Sonar and Navigation*, 143(4):246–254.
- [Li et al., 2011] Li, N., Zhao, Q., and hao Xu, X. (2011). Research of taxiing optimization for aircraft. In *Transportation, Mechanical, and Electrical Engineering (TMEE), 2011 International Conference on*, pages 25–28. IEEE.
- [Li and Bar-Shalom, 1993] Li, X.-R. and Bar-Shalom, Y. (1993). Design of an interacting multiple model algorithm for air traffic control tracking. *IEEE Transactions on Control Systems Technology*, 1(3):186–194.
- [Liggins et al., 1997] Liggins, M. E., Chong, C.-Y., Kadar, I., Alford, M. G., Vannicola, V., and Thomopoulos, S. (1997). Distributed fusion architectures and algorithms for target tracking. *Proceedings of the IEEE*, 85(1):95–107.
- [Lin et al., 2004] Lin, X., Bar-Shalom, Y., and Kirubarajan, T. (2004). Exact multisensor dynamic bias estimation with local tracks. *IEEE Transactions on Aerospace and Electronic Systems*, 40(2):576–590.

- [Lin et al., 2005] Lin, X., Bar-Shalom, Y., and Kirubarajan, T. (2005). Multisensor multitarget bias estimation for general asynchronous sensors. *IEEE Transactions on Aerospace and Electronic Systems*, 41(3):899–921.
- [Liu and Hwang, 2011] Liu, W. and Hwang, I. (2011). Probabilistic trajectory prediction and conflict detection for air traffic control. *Journal of Guidance, Control, and Dynamics*, 34(6):1779–1789.
- [Lukin et al., 2009] Lukin, K., Mogila, A., Vyplavin, P., Galati, G., and Pavan, G. (2009). Novel concepts for surface movement radar design. *International Journal of Microwave and Wireless Technologies*, 1(3):163–169.
- [Mantilla-Gaviria et al., 2015] Mantilla-Gaviria, I. A., Leonardi, M., Galati, G., and Balbastre-Tejedor, J. V. (2015). Localization algorithms for multilateration (mlat) systems in airport surface surveillance. *Signal, Image and Video Processing*, 9(7):1549–1558.
- [Mario and Rife, 2012] Mario, C. and Rife, J. (2012). Integrity and continuity for automated surface conflict-detection monitoring. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1179–1187.
- [Mauro et al., 2001] Mauro, F., Galati, G., and Naldi, M. (2001). A novel w-band radar for airport traffic monitoring: implementation, first operational results, perspectives. In *Physics and Engineering of Millimeter and Sub-Millimeter Waves, 2001. The Fourth International Kharkov Symposium on*, volume 1, pages 27–32. IEEE.
- [Mazor et al., 1998] Mazor, E., Averbuch, A., Bar-Shalom, Y., and Dayan, J. (1998). Interacting multiple model methods in target tracking: a survey. *IEEE Transactions on aerospace and electronic systems*, 34(1):103–123.
- [Meier et al., 2005] Meier, C., Jakobi, J., Adamson, P., Lozito, S., and Martin, L. (2005). Benefits of advanced surface movement guidance and control systems (a-smgcs). *Air Traffic Control Quarterly*, 13(4):329–356.
- [Oh et al., 2015] Oh, H., Kim, S., and Tsourdos, A. (2015). Road-map-assisted standoff tracking of moving ground vehicle using nonlinear model predictive control. *IEEE Transactions on Aerospace and Electronic Systems*, 51(2):975–986.
- [Oh et al., 2009] Oh, S., Russell, S., and Sastry, S. (2009). Markov chain monte carlo data association for multi-target tracking. *IEEE Transactions on Automatic Control*, 54(3):481–497.

- [Pannetier et al., 2004] Pannetier, B., Benameur, K., Nimier, V., and Rombaut, M. (2004). Ground moving target tracking with road constraint. In *Signal Processing, Sensor Fusion, and Target Recognition XIII*, volume 5429, pages 138–150. International Society for Optics and Photonics.
- [Pannetier et al., 2005] Pannetier, B., Benameur, K., Nimier, V., and Rombaut, M. (2005). VS-IMM using road map information for a ground target tracking. In *Information Fusion, 2005 8th International Conference on*, volume 1, pages 8–pp. IEEE.
- [Panta et al., 2007] Panta, K., Vo, B.-N., and Singh, S. (2007). Novel data association schemes for the probability hypothesis density filter. *IEEE Transactions on Aerospace and Electronic Systems*, 43(2).
- [Peebles and Shi, 2001] Peebles, P. Z. and Shi, B. E. (2001). *Probability, random variables, and random signal principles*, volume 3. McGraw-Hill New York, NY.
- [Perl, 2006] Perl, E. (2006). Review of airport surface movement radar technology. *IEEE Aerospace and Electronic Systems Magazine*, 21(10).
- [Petrochilos and van der Veen, 2004] Petrochilos, N. and van der Veen, A.-J. (2004). Algorithms to separate overlapping secondary surveillance radar replies. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings (ICASSP'04). IEEE International Conference on*, volume 2, pages ii–49. IEEE.
- [Piazza, 2000] Piazza, E. (2000). A-smgcs routing and guidance functions. *IEEE Aerospace and Electronic Systems Magazine*, 15(7):15–23.
- [Poles, 2010] Poles, D. (2010). Revision of atmosphere model in bada aircraft performance model.
- [Prandini et al., 2000] Prandini, M., Hu, J., Lygeros, J., and Sastry, S. (2000). A probabilistic approach to aircraft conflict detection. *IEEE Transactions on intelligent transportation systems*, 1(4):199–220.
- [Prautzsch et al., 2013] Prautzsch, H., Boehm, W., and Paluszny, M. (2013). *Bézier and B-spline techniques*. Springer Science & Business Media.
- [Pu et al., 2017] Pu, W., Liu, Y.-F., Yan, J., Liu, H., and Luo, Z.-Q. (2017). Optimal estimation of sensor biases for asynchronous multi-sensor registration. *arXiv preprint arXiv:1710.02808*.

- [Pyo et al., 2001] Pyo, J.-S., Shin, D.-H., and Sung, T.-K. (2001). Development of a map matching method using the multiple hypothesis technique. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pages 23–27. IEEE.
- [Reid, 1977] Reid, D. B. (1977). *A multiple hypothesis filter for tracking multiple targets in a cluttered environment*. Lockheed Missiles & Space Company, Incorporated.
- [Ribeiro, 2004] Ribeiro, M. I. (2004). Kalman and extended kalman filters: Concept, derivation and properties. *Institute for Systems and Robotics*, 43.
- [Ristic and Clark, 2012] Ristic, B. and Clark, D. (2012). Particle filter for joint estimation of multi-object dynamic state and multi-sensor bias. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 3877–3880. IEEE.
- [Ristic et al., 2011] Ristic, B., Vo, B.-N., Clark, D., and Vo, B.-T. (2011). A metric for performance evaluation of multi-target tracking algorithms. *IEEE Transactions on Signal Processing*, 59(7):3452–3457.
- [Schönefeld and Möller, 2012] Schönefeld, J. and Möller, D. (2012). Runway incursion prevention systems: A review of runway incursion avoidance and alerting system approaches. *Progress in Aerospace Sciences*, 51:31–49.
- [Schwab and Rost, 1985] Schwab, C. and Rost, D. (1985). Airport surface detection equipment. *Proceedings of the IEEE*, 73(2):290–300.
- [Seah et al., 2010] Seah, C. E., Aligawesa, A., and Hwang, I. (2010). Algorithm for conformance monitoring in air traffic control. *Journal of Guidance, Control, and Dynamics*, 33(2):500–509.
- [Semertzidis et al., 2010] Semertzidis, T., Dimitropoulos, K., Koutsia, A., and Grammalidis, N. (2010). Video sensor network for real-time traffic monitoring and surveillance. *IET intelligent transport systems*, 4(2):103–112.
- [Shea et al., 2000] Shea, P. J., Zadra, T., Klammer, D. M., Frangione, E., and Brouillard, R. (2000). Improved state estimation through use of roads in ground tracking. In *Signal and Data Processing of Small Targets 2000*, volume 4048, pages 321–333. International Society for Optics and Photonics.

- [Shi et al., 2017] Shi, K., Yu, H., Zhu, Z., and Cheng, J. (2017). Anticipatory runway incursion prevention based on inaccurate position surveillance information. In *Asian Conference on Intelligent Information and Database Systems*, pages 93–104. Springer.
- [Shnidman, 1989] Shnidman, D. A. (1989). The calculation of the probability of detection and the generalized Marcum Q-function. *IEEE Transactions on Information Theory*, 35(2):389–400.
- [Simon and Chia, 2002] Simon, D. and Chia, T. L. (2002). Kalman filtering with state equality constraints. *IEEE transactions on Aerospace and Electronic Systems*, 38(1):128–136.
- [Singer, 1970] Singer, R. A. (1970). Estimating optimal tracking filter performance for manned maneuvering targets. *IEEE Transactions on Aerospace and Electronic Systems*, (4):473–483.
- [Skolnik, 1970] Skolnik, M. I. (1970). *Radar handbook*. McGraw-Hill Education.
- [Stevens, 1988] Stevens, M. (1988). *Secondary Surveillance Radar*. Artech House.
- [Streller, 2008] Streller, D. (2008). Road map assisted ground target tracking. In *Information Fusion, 2008 11th International Conference on*, pages 1–7. IEEE.
- [Svensson, 2002] Svensson, J. (2002). Automated docking systems. *Airports International*, 35(9).
- [Tahk and Speyer, 1990] Tahk, M. and Speyer, J. L. (1990). Target tracking problems subject to kinematic constraints. *IEEE transactions on automatic control*, 35(3):324–326.
- [Tang et al., 2005] Tang, W., Howell, G., and Tsai, Y.-H. (2005). Barometric altimeter short-term accuracy analysis. *IEEE Aerospace and Electronic Systems Magazine*, 20(12):24–26.
- [Thomaidis et al., 2013] Thomaidis, G., Tsogas, M., Lytrivis, P., Karaseitanidis, G., and Amditis, A. (2013). Multiple hypothesis tracking for data association in vehicular networks. *Information Fusion*, 14(4):374–383.
- [Trim, 1990] Trim, R. (1990). Mode s: an introduction and overview (secondary surveillance radar). *Electronics & Communication Engineering Journal*, 2(2):53–59.

- [Tugnait, 1982] Tugnait, J. K. (1982). Detection and estimation for abruptly changing systems. *Automatica*, 18(5):607–615.
- [Ulmke and Koch, 2006] Ulmke, M. and Koch, W. (2006). Road-map assisted ground moving target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 42(4).
- [Wan and Van Der Merwe, 2000] Wan, E. A. and Van Der Merwe, R. (2000). The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. Ieee.
- [Wang et al., 2016] Wang, H., Liu, T., Bu, Q., and Yang, B. (2016). An algorithm based on hierarchical clustering for multi-target tracking of multi-sensor data fusion. In *Control Conference (CCC), 2016 35th Chinese*, pages 5106–5111. IEEE.
- [Wang and Varshney, 1993] Wang, T. and Varshney, P. (1993). A tracking algorithm for maneuvering targets. *IEEE transactions on Aerospace and Electronic Systems*, 29(3):910–925.
- [Watanabe and Tzafestas, 1993] Watanabe, K. and Tzafestas, S. G. (1993). Generalized pseudo-bayes estimation and detection for abruptly changing systems. *Journal of Intelligent and Robotic Systems*, 7(1):95–112.
- [Weiß et al., 2013] Weiß, B., Centarti, F., Schmitt, F., and Straub, S. (2013). Route-based detection of conflicting atc clearances on airports. *arXiv preprint arXiv:1304.6494*.
- [Xin-min et al., 2010] Xin-min, T., Yu-ting, W., and Song-chen, H. (2010). Aircraft taxi route planning for a-smgcs based on discrete event dynamic system modeling. In *Computer Modeling and Simulation, 2010. ICCMS’10. Second International Conference on*, volume 1, pages 224–228. IEEE.
- [Xinyan and Wei, 2007] Xinyan, Z. and Wei, Z. (2007). Application of road information in ground moving target tracking. *Chinese Journal of Aeronautics*, 20(6):529–538.
- [Xu et al., 2013] Xu, L., Li, X. R., Duan, Z., and Lan, J. (2013). Modeling and state estimation for dynamic systems with linear equality constraints. *IEEE Transactions on Signal Processing*, 61(11):2927–2939.

- [Xu et al., 2010] Xu, N., Cassell, R., Evers, C., Hauswald, S., and Langhans, W. (2010). Performance assessment of multilateration systems-a solution to nextgen surveillance. In *Integrated Communications Navigation and Surveillance Conference (ICNS), 2010*, pages D2–1. IEEE.
- [Yang et al., 2014] Yang, Y., Zhao, Y., and Kyas, M. (2014). Recursive bayesian estimation using a topological map for indoor position tracking. In *Vehicular Technology Conference (VTC Spring), 2014 IEEE 79th*, pages 1–5. IEEE.
- [Yao et al., 2009] Yao, Y., Huang, Z.-g., and Li, R. (2009). Data fusion algorithm of baro/gps based on delta-altitude. In *Information, Computing and Telecommunication, 2009. YC-ICT'09. IEEE Youth Conference on*, pages 522–525. IEEE.
- [You et al., 2013] You, H., Hongwei, Z., and Xiaoming, T. (2013). Joint systematic error estimation algorithm for radar and automatic dependent surveillance broadcasting. *IET Radar, Sonar & Navigation*, 7(4):361–370.
- [Young and Jones, 2000] Young, S. D. and Jones, D. R. (2000). Runway incursion prevention using an advanced surface movement guidance and control system (a-smgcs). In *Digital Avionics Systems Conference, 2000. Proceedings. DASC. The 19th*, volume 1, pages 2E2–1. IEEE.
- [Zaliva and Franchetti, 2014] Zaliva, V. and Franchetti, F. (2014). Barometric and gps altitude sensor fusion. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 7525–7529. IEEE.
- [Zhou et al., 2006] Zhou, J., Zhu, Y., You, Z., and Song, E. (2006). An efficient algorithm for optimal linear estimation fusion in distributed multisensor systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 36(5):1000–1009.
- [Zhu et al., 2012] Zhu, X., Tang, X., and Han, S. (2012). Aircraft intersection collision conflict detection and resolution under the control of a-smgcs. In *Modelling, Identification & Control (ICMIC), 2012 Proceedings of International Conference on*, pages 120–125. IEEE.

Part IV

Appendices

Appendix A

Estimation Methods

A.1 Kalman Filter

A Kalman filter [Kalman, 1960] [Kalman and Bucy, 1961] [Gelb, 1974] is a well known estimator that minimizes the Minimum Mean Square Error (MMSE). It operates on linear dynamic systems modelled as processes of Gauss-Markov. Therefore, the dynamic systems are represented as follows:

$$x_k = F_k \cdot x_{k-1} + w_k \quad (\text{A.1})$$

where x_k is the state estimate at t_k , F_k is the state transition model from the previous state x_{k-1} and w_k is the process noise which is assumed a zero mean multivariate normal distribution with covariance Q_k : $N(0, Q_k)$.

The measure model is represented with the following equation:

$$z_k = H_k \cdot x_k + v_k \quad (\text{A.2})$$

where H_k is the observation model and v_k is the measure noise which is assumed to be zero mean Gaussian white noise with covariance R_k : $N(0, R_k)$.

The Kalman filter recursively estimates the dynamic state defined by the

two previous equations. Algorithmically, it has two steps: prediction and update. Both of these steps are explained below.

Prediction

In this step it is obtained the predicted state vector for the current time instant, $x_{k|k-1}$, using the filtered state vector from the previous time instant, $x_{k-1|k-1}$. This prediction is made using the transition matrix and it can be written as:

$$x_{k|k-1} = F_k \cdot x_{k-1|k-1} \quad (\text{A.3})$$

Similarly, the covariance of the predicted state vector is obtained from the filtered one for the previous instant time. It is predicted using the transition matrix and the plant noise covariance of the system:

$$P_{k|k-1} = F_k \cdot P_{k-1|k-1} \cdot F_k^T + Q_k \quad (\text{A.4})$$

Update

In this phase the predicted state is combined with the current measure in order to improve the state estimate. To do this, first the residual between the predicted state and the measurement, y_k , is calculated.

$$y_k = z_k - H_k \cdot x_{k|k-1} \quad (\text{A.5})$$

After this, the residual covariance denoted by S_k is calculated:

$$S_k = R_k + H_k \cdot P_{k|k-1} \cdot H_k^T \quad (\text{A.6})$$

Now, the Kalman filter gain K_k is calculated. This gain is the one that minimizes the covariance of the error in the estimation.

$$K_k = P_{k|k-1} \cdot H_k^T \cdot S_k^{-1} \quad (\text{A.7})$$

Finally, the filtered state vector and its associated covariance are obtained:

$$x_{k|k} = x_{k|k-1} + K \cdot y \quad (\text{A.8})$$

$$P_{k|k} = (I - K_k \cdot H_k) P_{k|k-1} \quad (\text{A.9})$$

It can be also calculated the residual between the estimated state and the measurement:

$$v = z_k - H_k \cdot x_{k|k} \quad (\text{A.10})$$

Extended Kalman Filter (EKF)

The previously explained Kalman filter is applied on linear dynamic systems. When the dynamic system is not linear it has to be linearized obtaining what is called an Extended Kalman Filter (EKF). The EKF linearizes the estimation around the current estimate using the partial derivatives of the functions that describe the system and the relationship of the state to the measures.

Unlike the linear Kalman filter, the extended version is not generally an optimal estimator. In addition, if the initial state estimation is incorrect, or if the process is modelled incorrectly, the filter may quickly diverge. Another problem is that the estimated covariance matrix tends to underestimate the true matrix of covariance and therefore become statistically inconsistent.

The equations of the extended Kalman filter are briefly explained below. Writing the non-linear dynamic model as:

$$x_k = f(x_{k-1}) + w_k \quad (\text{A.11})$$

$$z_k = h(x_k) + v_k \quad (\text{A.12})$$

If the matrix of transition and measurements are linearized with their respective Jacobian matrix, we obtain:

$$F_k = \frac{df(x)}{dx} \quad (\text{A.13})$$

$$H_k = \frac{dh(x)}{dx} \quad (\text{A.14})$$

Then the prediction and update steps are done with the following equations:

$$x_{k|k-1} = f(x_{k-1|k-1}) \quad (\text{A.15})$$

$$P_{k|k-1} = F_k \cdot P_{k-1|k-1} \cdot F_k^T + Q_k \quad (\text{A.16})$$

$$y_k = z_k - h(x_{k|k-1}) \quad (\text{A.17})$$

$$S_k = R_k + H_k \cdot P_{k|k-1} \cdot H_k^T \quad (\text{A.18})$$

$$K_k = P_{k|k-1} \cdot H_k^T \cdot S_k^{-1} \quad (\text{A.19})$$

$$x_{k|k} = x_{k|k-1} + K \cdot y \quad (\text{A.20})$$

$$P_{k|k} = (I - K_k \cdot H_k) P_{k|k-1} \quad (\text{A.21})$$

A.2 IMM Algorithm

The Interacting Multiple Model (IMM) filter [Li and Bar-Shalom, 1993] [Blom and Bar-Shalom, 1988] [Kirubarajan et al., 2000] [Mazor et al., 1998] is a multi-mode estimator that obtains estimated state by the weighted sum of states from a set of estimators, usually Kalman Filter (KF). Each of these estimators represents an IMM mode and they are specifically designed to achieve the appropriate tracking for each of the aircraft's movement models. Examples of these movements may include: constant velocity movement, turns, longitudinal accelerations/decelerations, stopping, etc.

The flow diagram with the main blocks of an IMM can be seen in figure A.1. The filter takes as input both the target position measures at time instant k , $z[k]$, and its associated covariance matrix, $R[k]$. As output, it provides the estimated target state, $\hat{x}[k]$, and its associated covariance matrix at that instant, $P[k]$.

The IMM algorithm obtains a state estimation by the weighted sum of states from a set of Kalman filters associated with different models (modes). The weights change over time as the movement of the target changes. Thereby, the mode matched with target movement is maintained as dominant and the rest as negligible, but still allowing a quick switching between modes

Before proceeding to explain the IMM algorithm, the different probabilities involved in this process will be defined. After this explanation, the four algorithm steps will be explained: premixing, filtering adapted to each mode, mode probability calculation and, finally, combination of the modes estimators. Remark that to minimize the use of sub-incidents the notation $[k]$ indicates that the probability refers to the instant t_k .

IMM Probabilities

Before explaining the IMM steps, we will define the probabilities involved.

Figure A.1: IMM general structure

Transition probability: probability $p_{ij}[k]$ of switching from i mode at instant $k - 1$ to j mode at instant k

$$p_{ij}[k] = P \{m_j[k] | m_i[k - 1]\} \quad (\text{A.22})$$

Model probability: probability $\mu_i[k - 1]$ that the target follows a particular motion model at the instant $k-1$.

$$\mu_i[k - 1] = P \{m_i[k - 1]\} \quad (\text{A.23})$$

Predicted model probability: predicted probability $\mu_{ip}[k]$ that the target follows a particular model of motion at instant k .

$$\mu_{ip}[k] = P \{m_i[k] | z[k-1]\} \quad (\text{A.24})$$

Model mixing probability: probability p_{mixij} that the target was in mode i of the previous update given the hypothesis that the current mode is j , and taking into account the measures received up to the instant $k-1$.

$$p_{mixij} = P \{m_i[k-1] | m_j(k), z[k-1]\} \quad (\text{A.25})$$

It is also important to clarify the different types of state vectors involved in the IMM. There are three: premixed, predicted, and filtered. The predicted and filtered vector states match those defined for the Kalman filter. For the premixed state, the filtered estimators (and their covariances) generated by each of the filters are mixed after the previous measurement, providing mixed state and covariance estimators that represent the input to each of the modes, and which will be updated with the current measurement. The aim of this mix is to consider the possibility of a change of mode (type of target movement) in the time elapsing between two measurements.

Premixing step

At this step the filtered estimators are mixed after the previous measure. The result of this process will be the mixed state estimators, $\hat{x}_0[k-1]$, which will be the input to each of the modes that set up the IMM. Thereby, different weights are applied to the estimators depending on which mode is dominant and on the probability that the mode has switched since the last measure.

First, the transition matrix is generated, which consists of the transition probabilities $p_{ij}[k]$. Each row of the matrix has the transition probabilities for the same mode.

$$T[k] = \begin{bmatrix} p_{11}[k] & p_{12}[k] & \cdots & p_{1M}[k] \\ p_{21}[k] & p_{22}[k] & \cdots & p_{2M}[k] \\ \vdots & \cdots & \ddots & \vdots \\ p_{M1}[k] & p_{M2}[k] & \cdots & p_{MM}[k] \end{bmatrix} \quad (\text{A.26})$$

where M is the amount of IMM modes. Then, the predicted mode probabilities are calculated:

$$\mu_{ip}[k] = \sum_{i=1}^M \mu_i[k-1] p_{ij}[k] \quad (\text{A.27})$$

This can be rewritten as follows:

$$\begin{bmatrix} \mu_{1p}[k] \\ \mu_{2p}[k] \\ \vdots \end{bmatrix} = (T[k])^T \begin{bmatrix} \mu_1[k-1] \\ \mu_2[k-1] \\ \vdots \end{bmatrix} \quad (\text{A.28})$$

Now, the model mixing probabilities are calculated:

$$p_{mix_{ij}} = \mu_i[k-1] \cdot p_{ij}[k] / \mu_{jp}[k] \quad (\text{A.29})$$

Finally, the state estimators and their covariances are mixed for each mode:

$$x_{0j} = \sum_{i=1}^M p_{mix_{ij}} \cdot x_j[k-1] \quad (\text{A.30})$$

$$P_{0j} = \sum_{i=1}^M p_{mix_{ij}} \cdot (P_j[k] + (x_i[k-1] - x_{0j})(x_i[k-1] - x_{0j})^T) \quad (\text{A.31})$$

Filtering

In this step each estimator will be updated using its KF or EKF. Each of the different filters adapted to a mode and they will receive as input the premixed state estimators, $x_{0j}[k-1]$, their premixed covariance, $P_{0j}[k-1]$ and the measurement, $z[k]$.

As it has been already explained in the previous section each filter will provide as output the filtered state $x_j[k]$ and its covariance matrix, $P_j[k]$. The residual covariance, S_i , and the residual between the measure and the filtered state, v_i .

Model Probability Updating

The IMM then calculates the mode probability for the estimators that have been calculated in the previous step. This probability can be calculated as follows:

$$\mu_i = \frac{(\mu_{ip} \cdot \Lambda_i)}{\sum_{i=1}^M \mu_{ip} \cdot \Lambda_i} \quad (\text{A.32})$$

where $like_i$ denotes the estimation likelihood. This can be calculated as follows:

$$\Lambda_i = \frac{1}{(2 \cdot \pi)^{N/2}} \cdot \frac{1}{\sqrt{|S_i|}} \exp \left(-\frac{1}{2} \cdot v_i^T [k] \cdot S_i^{-1} [k] \cdot v_i [k] \right) \quad (\text{A.33})$$

where N is the length of the measure vector.

State Estimate Combination

At this step the mode estimators and their covariances are combined, using the mode probabilities previously obtained. This way the most likely

modes contributes more to the combination.

$$x[k] = \sum_{i=1}^M \mu_i[k] \cdot x_i[k] \quad (\text{A.34})$$

$$P[k] = \sum_{j=1}^M \mu_j \cdot (P_j[k] + (x_j[k] - x[k])(x_j[k] - x[k])^T) \quad (\text{A.35})$$

This would end the calculation of the IMM. As the IMM is a recursive algorithm, the calculated estimate would feed back the filter input at the pre-mixing stage.