Linköping studies in science and technology. Dissertations.
No. 1231

Efficient Estimation and Detection Methods for Airborne Applications

Per-Johan Nordlund

Department of Electrical Engineering
Linköping University, SE–581 83 Linköping, Sweden

Linköping 2008

Linköping studies in science and technology. Dissertations.
No. 1231

Efficient Estimation and Detection Methods for Airborne Applications

Per-Johan Nordlund

perno@isy.liu.se
www.control.isy.liu.se
Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE–581 83 Linköping
Sweden

ISBN 978-91-7393-720-7 ISSN 0345-7524

Printed by LiU-Tryck, Linköping, Sweden 2008

To Caroline and Elliot

Abstract

The overall purpose with this thesis is to investigate and provide computationally efficient methods for estimation and detection. The focus is on airborne applications, and we seek estimation and detection methods which are accurate and reliable yet effective with respect to computational load. In particular, the methods shall be optimized for terrain-aided navigation and collision avoidance respectively. The estimation part focuses on particle filtering and the in general much more efficient marginalized particle filter. The detection part focuses on finding efficient methods for evaluating the probability of extreme values. This is achieved by considering the, in general, much easier task to compute the probability of level-crossings.

The concept of aircraft navigation using terrain height information is attractive because of the independence of external information sources. Typically terrain-aided navigation consists of an inertial navigation unit supported by position estimates from a terrain-aided positioning (TAP) system. TAP integrated with an inertial navigation system is challenging due to its highly nonlinear nature. Today, the particle filter is an accepted method for estimation of more or less nonlinear systems. At least when the requirements on computational load are not rigorous. In many on-line processing applications the requirements are such that they prevent the use of the particle filter. We need more efficient estimation methods to overcome this issue, and the marginalized particle filter constitutes a possible solution. The basic principle for the marginalized particle filter is to utilize linear and discrete substructures within the overall nonlinear system. These substructures are used for efficient estimation by applying optimal filters such as the Kalman filter. The computationally demanding particle filter can then be concentrated on a smaller part of the estimation problem.

The concept of an aircraft collision avoidance system is to assist or ultimately replace the pilot in order to to minimize the resulting collision risk. Detection is needed in aircraft collision avoidance because of the stochastic nature of the sensor readings, here we use information from video cameras. Conflict is declared if the minimum distance between two aircraft is less than a level. The level is given by the radius of a safety sphere surrounding the aircraft. We use the fact that the probability of conflict, for the process studied here, is identical to the probability for a down-crossing of the surface of the sphere. In general, it is easier to compute the probability of down-crossings compared to extremes. The Monte Carlo method provides a way forward to compute the probability of conflict. However, to provide a computationally tractable solution we approximate the crossing of the safety sphere with the crossing of a circular disc. The approximate method yields a result which is as accurate as the Monte Carlo method but the computational load is decreased significantly.

Populärvetenskaplig sammanfattning

Det främsta syftet med den här avhandlingen är att undersöka och tillhandahålla beräkningseffektiva metoder för estimering och detektering. Fokus är på luftburna applikationer, och vi söker estimerings- och detekteringsmetoder som är noggranna och tillförlitliga samtidigt som dem är effektiva med avseende på beräkningsbelastningen. Framförallt så ska metoderna vara optimerade för terrängbaserad navigering och kollisionundvikning. Estimeringsdelen fokuserar på partikelfiltret och det i allmänhet mycket mer effektiva marginaliserade partikelfiltret. Detekteringsdelen fokuserar på att hitta effektiva metoder för att utvärdera sannolikheten för extremvärden. Det uppnår vi genom att betrakta den i allmänhet mycket lättare uppgiften att beräkna sannolikheten för att passera en viss nivå.

Konceptet för att navigera ett flygplan med hjälp av information från terränghöjden är attraktiv på så sätt att systemet blir oberoende av yttre informationskällor. Typiskt består terrängbaserad navigering av ett tröghetsnavigeringssystem stöttat med positionsskattningar från ett terrängbaserat positioneringssystem. Att integrera ett terrängbaserat positioneringssystem med ett tröghetsnavigeringssystem är utmanande på grund av dess kraftigt olinjära karaktär. Idag är partikelfiltret att betrakta som en accepterad estimeringsmetod för mer eller mindre olinjära system. Åtminstone när kraven på beräkningsbelastning inte är alltför stränga. För många realtidstillämpningar är kraven emellertid sådana att dem förhindrar användandet av partikelfiltret. Vi behöver därför mer beräkningseffektiva metoder för att överbrygga problemet, och det marginaliserade partikelfiltret utgör en möjlig lösning. Principen för det marginaliserade partikelfiltret är att utnyttja linjär och diskret struktur inom det totalt sett olinjära systemet. Dessa strukturer går att utnyttja för effektiv estimering genom att skatta dessa delar optimalt med hjälp av tex kalmanfiltret. Det beräkningskrävande partikelfiltret koncentrerar man sedan till en mindre del av skattningsproblemet.

Konceptet för kollisionsundvikning för flygplan är att stötta och i förlängningen ersätta piloten för att minimera kollisionsrisken. Detektering är nödvändig inom kollisionsundvikning på grund av den stokastiska naturen hos sensormätningar. Här använder vi oss av videokamera. Konflikt uppstår om det minsta avståndet mellan flygplanen passerar under en viss nivå. Nivån ges av radien till en säkerhetssfär som omger flygplanen. Vi använder oss av det faktum att sannolikheten för konflikt, för den process vi studerar här, är identisk med sannolikheten för skärning av sfärens yta. Generellt så är det lättare att beräkna sannolikheten för skärning jämfört med extremvärden. Monte Carlo metoden tillhandahåller en väg framåt för att beräkna sannolikheten för konflikt. För att tillhandahålla en beräkningseffektiv lösning approximerar vi emellertid skärningen av säkerhetssfären med skärningen av en cirkulär skiva. Den approximativa metoden ger ett resultat som är lika noggrannt som Monte Carlo metodens men beräkningsbelastningen sjunker signifikant.

Acknowledgments

First of all I would like to thank my supervisor Prof Fredrik Gustafsson for his support throughout my research. Without his never ending patience reading and commenting my articles this thesis would certainly not look the same. I would like to thank Lic Jan Palmqvist for giving me the chance to combine work at Saab with part time PhD studies during $1999-2002$, and Dr Predrag Pucar for giving me the opportunity to resume my research in 2006. Dr Sören Molander deserves attention for all our discussions on collision avoidance. I am also very grateful to Prof Lennart Ljung for giving me the opportunity to join the control & communication group. Moreover, secretary Ulla Salaneck deserves extra gratitude and attention for handling administrative work with ease.

The combination of working at Saab and doing research at the university have been most stimulating. The combination means that encountered technical problems can get the attention they deserve. Working in a project with a tight time schedule means that there is seldom time to solve problems on a theoretical level. This is not always necessary, but often it is. If these problems are solved only partially they have a tendency of emerging again in the future.

Several people have read different versions of this thesis, Prof Fredrik Gustafsson, Dr David Törnqvist, Dr Sören Molander, Dr Umut Orguner, Bengt-Göran Sundqvist, Zoran Sjanic, Roger Larsson and Fredrik Lindsten. Their valuable comments and suggestions have improved the quality of this thesis significantly. Ass Prof Thomas Schön should be credited for our article on the marginalized particle filter. Dr Gustaf Hendeby deserves appreciation for solving my LaTeX-issues, and Fredrik Lindsten for letting me use his illustrative picture on the sense and avoid concept.

My work has been financed partially by the competence center Information Systems for Industrial Control and Supervision ISIS, the National Aerospace Research Funding NFFP and partially by Saab Aerosystems. All are gratefully acknowledged.

I would also like to take the chance and thank my parents, sister, relatives, friends and colleagues for believing in me and my ambitions. Finally, I would like to dedicate this thesis to Caroline and Elliot for putting up with me during this autumn.

Linköping, December 2008
Per-Johan Nordlund

Contents

I Background **1**

1 Introduction **3**
1.1 Estimation for Aircraft Navigation 3
1.2 Detection for Collision Avoidance 6
1.2.1 Manned Air Traffic 6
1.2.2 Unmanned Aerial Vehicles 7
1.3 Outline 8
1.3.1 Outline of Part I 8
1.3.2 Outline of Part II 8
1.4 Other Publications 11

2 Efficient State Estimation **13**
2.1 Terrain-Aided Navigation 13
2.2 Recursive Estimation 14
2.2.1 Linear Models 16
2.2.2 Multiple Models 17
2.2.3 Nonlinear Models 19
2.3 Particle Filter 20
2.3.1 Importance Sampling 20
2.3.2 Recursive Importance Sampling 22
2.3.3 Resampling 23
2.3.4 The Algorithm 24
2.3.5 Asymptotic Properties 25
2.4 Marginalized Particle Filter 28
2.4.1 General Description 28

2.4.2 Filter for Terrain-Aided Navigation . . . 30
2.4.3 Asymptotic Properties . . . 31

3 Efficient Detection 35
3.1 Sense and Avoid . . . 35
3.1.1 Sensor . . . 35
3.1.2 Tracking . . . 36
3.1.3 Near Mid-Air Collision Avoidance . . . 38
3.2 Probability of Near Mid-Air Collision . . . 40
3.3 Detection using hypothesis testing . . . 41
3.4 Probability of level-crossings . . . 44
3.4.1 One dimension and constant velocity . . . 45
3.4.2 One dimension and piecewise constant velocity . . . 47
3.4.3 Two dimensions . . . 48
3.4.4 Three dimensions . . . 50

4 Concluding Remarks 55
4.1 Summary . . . 55
4.2 Future Research . . . 56

A Derivation of $p_\tau(t)$ 57

Bibliography 59

II Publications 67

A Marginalized Particle Filters for Mixed Linear/Nonlinear State-space Models 69
1 Introduction . . . 71
2 Marginalization . . . 73
2.1 The Standard Particle Filter . . . 73
2.2 Diagonal Model . . . 75
2.3 Triangular Model . . . 77
2.4 The General Case . . . 78
3 Important Special Cases and Extensions . . . 81
3.1 Generalized Noise Assumptions . . . 81
3.2 An Important Model Class . . . 82
4 An Illustrating Example . . . 83
5 Integrated Aircraft Navigation . . . 84
5.1 The Dynamic Model . . . 85
5.2 Result . . . 85
6 Conclusions . . . 86
A Appendix . . . 88
A.1 Proof for Theorem 1 . . . 88
References . . . 90

B Marginalized Particle Filter for Accurate and Reliable Aircraft Navigation 95
1 Introduction . . . 97
2 Outline . . . 100
3 INS Error Dynamics . . . 100
4 Terrain-Aided Positioning . . . 104
5 The Marginalized Particle Filter . . . 105
6 Blended INS/TAP using MPF . . . 110
6.1 The applied algorithm . . . 110
6.2 Convergence Analysis of Algorithm 1 . . . 113
7 Simulation Results . . . 115
8 Conclusions . . . 118
A Appendix . . . 118
A.1 INS Motion Equations . . . 118
A.2 Discrete-Time Propagation Matrices . . . 121
References . . . 122

C Probabilistic Conflict Detection for Piecewise Straight Paths 129
1 Introduction . . . 131
2 Problem formulation . . . 133
3 Crossing in case of constant velocity . . . 134
3.1 Time-to-go . . . 134
3.2 Conflict probability for a time interval . . . 136
3.3 Monte-Carlo Approximation . . . 138
3.4 Numerical Approximation . . . 139
4 Conflict in case of piecewise constant velocity . . . 141
4.1 Crossing of the line segment . . . 142
4.2 Implementation . . . 144
5 Simulation results . . . 145
6 Conclusions . . . 146
A Appendix . . . 147
A.1 Proof of Lemma 1 . . . 147
A.2 Proof of Theorem 1 . . . 147
A.3 Derivation of (43) . . . 149
References . . . 149

D Probabilistic Near Mid-Air Collision Avoidance 151
1 Introduction . . . 153
2 Problem formulation . . . 155
3 Crossing of the safety zone . . . 156
3.1 Constant velocity . . . 156
3.2 Piecewise constant velocity . . . 158
4 Monte-Carlo Approximation . . . 159
5 Approximate the safety zone with a disc . . . 160
5.1 Constant velocity . . . 160
5.2 Piecewise constant velocity . . . 163
6 Implementation of the disc approximation . . . 165

6.1 Computing $\mathrm{P}\big(s_{\perp}^{(j,m)}(t) < R\big)$ 165
6.2 Computing $\mathrm{P}(\widehat{\mathrm{NMAC}}_{(0,T)})$ 168
7 Simulation results 169
8 Conclusions 170
A Appendix 171
A.1 Proof of Theorem 1 171
References 175

Notation

Abbreviations and Acronyms

Abbreviation	Meaning
ATC	Air Traffic Control
CLT	Central Limit Theorem
CPA	Closest Point of Approach
EKF	Extended Kalman Filter
ELOS	Equivalent Level of Safety
GNSS	Global Navigation Satellite System
GPB	Generalized Pseudo Bayesian
GPS	Global Positioning System
ILS	Instrument Landing System
IMM	Interacting Multiple Model
INS	Inertial Navigation System
KF	Kalman Filter
MCMC	Markov Chain Monte Carlo
MPF	Marginalized Particle Filter
MS	Modified Spherical
MSC	Modified Spherical Coordinates
NMAC	Near Mid-Air Collision
PF	Particle Filter
RA	Radar Altimeter
RBPF	Rao-Blackwellized Particle Filter
RMSE	Root Mean Square Error
RNP	Required Navigation Performance
RP	Range Parameterized
RPV	Remotely Piloted Vehicle
RVR	Runway Visual Range

Abbreviation	Meaning
TAN	Terrain-Aided Navigation
TAP	Terrain-Aided Positioning
TCAS	Traffic Alert and Collision Avoidance System
TTG	Time-To-Go
UAV	Unmanned Aerial Vehicle
VFR	Visual Flight Rules

Symbols and Mathematical Notation

Notation	Meaning
$\alpha_{t\|t-k}^{(k)}$	GPB weight for model k at discrete time t based on $\mathbb{Y}_k$.
e_t	Measurement noise vector at discrete time t.
F_t	Discrete time state transition matrix.
$f(\cdot)$	State propagation mapping.
G_t	Discrete time noise to state propagation matrix.
$g(x)$	An arbitrary integrable function of the random variable x.
$\hat{g}_t$	Expected value of $g(x_t)$, $\hat{g}_t = \mathbb{E}_{p(\cdot)}[g(x_t)]$.
$\hat{g}_t^N$	Estimated value on $\hat{g}_t$ using N i.i.d. samples of x_t.
$\hat{g}_{\mathrm{R},t}^N$	Estimated value on $\hat{g}_t$ using N i.i.d. samples of a subset of x_t.
$h(\cdot)$	State to noise-free measurement mapping.
H_t	State to noise-free measurement transition matrix.
K_t	Kalman filter gain matrix.
λ_t	Discrete time Markov chain at time t.
Λ_t	Stacked vector of λ_t, $\Lambda_t = \{\lambda_0, \ldots, \lambda_t\}$.
N	Number of samples, realizations or particles.
N_{eff}, $\hat{N}_{\mathrm{eff}}$	Exact and estimated number of efficient samples.
n_x	Dimension of the vector x or x_t.
$P_{t\|t-k}$, P_x	Covariance of $\hat{x}_{t\|t-k}$ and x respectively.
Q_t	Process noise covariance matrices.
R	Radius of safety circle or sphere.
R_t	Measurement noise covariance matrix.
s, $s(t)$	Relative position.
S_t	Kalman filter residual covariance matrix.
Σ_t, $\Sigma_{\mathrm{R},t}$	Sampling covariances.
τ	Time-to-go.
u_t	Process noise vectors at discrete time t.
v, $v(t)$	Relative velocity.
$v_\perp$	Relative velocity perpendicular to line-of-sight.
$w(\mathbb{X}_t)$	Importance weight.
$w_t^{(i)}$	Importance weight evaluated at $\mathbb{X}_t^{(i)}$.
$\bar{w}_t^{(i)}$	Normalized importance weight, $\sum_i \bar{w}_t^{(i)} = 1$.
$\dot{x}$	Time derivative of x.

Notation	Meaning
x_t	State vector at discrete time t.
$x(t)$	State vector at continuous time t.
$\mathbb{X}_t$	Stacked vector of states, $\mathbb{X}_t = \{x_0, \ldots, x_t\}$.
$x_t^{(i)}$, $\mathbb{X}_t^{(i)}$	A sample/realization of x_t and $\mathbb{X}_t$ respectively.
$\hat{x}_{t\|t-k}$	Least mean square estimate of x_t given the measurements $\mathbb{Y}_k$.
y_t	Measurement vector at discrete time t.
$\mathbb{Y}_t$	Stacked vector of measurements, i.e. $\mathbb{Y}_t = \{y_0, \ldots, y_t\}$.
$\mathcal{N}(m, P)$	Gaussian probability density with mean m and covariance P.
χ_l^2	Central χ^2 distribution with l degrees of freedom.
$\chi_l^2(\lambda)$	Non-central χ^2 distribution with l degrees of freedom and non-centrality parameter λ.
$q(x)$	Importance function used for creating samples of a random variable x.
$p(x)$, $p_x(\cdot)$	Probability density function for the random variable x.
$p(x\|y)$, $p_{x\|y}(\cdot)$	Conditional probability density function for x given y.
$\mathrm{P}(\cdot)$, $\Pr(\cdot)$	Probability function.
$\mathbb{E}_{p(x)}[g(x)]$	Expectation of the function $g(x)$ with respect to the probability density $p(x)$.
$\xrightarrow{d}$	Convergence in distribution.
$\xrightarrow{a.s.}$	Almost surely convergence.
$\det P$	The determinant of matrix P.
P^T	The transpose of matrix P.
$\delta_{x^{(i)}}(x)$	Dirac delta function with mass located in $x^{(i)}$.
$\propto$	Proportional relation.
$\sim$	Distributed according to relation.

Part I

Background

1

Introduction

Autonomy is a concept that has received a lot of attention during recent years. To define autonomy is not easy, since it includes a wide spectrum of concepts which differ from case to case. Usually an autonomous vehicle means that there is some degree of self-guidance inherent in the vehicle. This provides the capability to move from one location to another in a more or less predetermined environment. However, the challenge for autonomous vehicles is how to deal with unknown and dynamic environments. This put requirements on the system to be capable of creating situational awareness, and that the vehicle is capable of reacting on unforeseen situations. The main driver for overcoming the technical issues is that it should be possible to significantly cut costs by using autonomous vehicles for monotonous, time-consuming and dangerous missions. Two areas are known as enabling technologies for autonomous vehicles; navigation and collision avoidance. The purpose of navigation is to estimate the own vehicle's kinematic state, e.g. position and velocity. The principle of collision avoidance is to estimate other vehicle's kinematic states and then to find a trajectory which constitutes low risk of collision with respect to the tracked vehicles.

The purpose of this thesis is to introduce the reader to the concepts of estimation and detection in general, and applied to navigation and collision avoidance for airborne systems in particular. Both estimation and detection are about obtaining accurate values of (functions of) quantities based on noisy readings of a surrounding environment. Although the material in this thesis is focused on airborne navigation and collision avoidance the theory should be applicable to many other areas.

1.1 Estimation for Aircraft Navigation

Today the Air Navigation Service (ANS) for civil air traffic still relies on ground based navigation aids, e.g. Non-Directional Beacons (NDB), VHF Omni-directional radio Range (VOR) and Distance Measuring Equipment (DME). Many of existing ground based navi-

gation aids are currently being phased out [108, 15] for cost reasons. The ability to replace them is made possible through the concept of Required Navigation Performance (RNP). RNP authorizes access through requirements on the navigation system performance regardless of hardware. The requirements are among other that the system shall have a certain accuracy and integrity. Integrity is a statistical measure of a navigation system's capability to stay within a containment region, or actually to provide a warning if the aircraft position can not be guaranteed to be within the region [87]. Loss of integrity can either be caused by undetected faults or that the fault-free accuracy is too low [92]. The systems that are currently acting as the primary source of information, and are expected to do so in the foreseeable future, are satellite based system often referred to as Global Navigation Satellite Systems (GNSS). The most known is probably the Global Positioning System (GPS), but there are others e.g. GLONASS and Galileo. A GNSS operates using satellites with known positions, and the satellites transmit signals making it possible for receivers to compute their own positions through triangulation [58].

For phases of flight where severe requirements are put on the navigation system accuracy, e.g. during precision approach and landing, dedicated systems are most often used. The currently most widespread system is the Instrument Landing System (ILS), which guides the pilot to touchdown in limited visibility conditions [58]. ILS landings are usually categorized as Cat I, II, IIIa, IIIb and IIIc, corresponding to successively lower decision heights and shorter runway visual range (RVR) in the order given. Decision height and RVR represent the point from where the pilot must be able to see the runway. Efforts are being made to find complements to ILS, and again GNSS are considered as the primary substitute.

There are however concerns with GNSS related to their reliability or integrity. GNSS is sensitive to disturbances, e.g. athmospheric phenomena and undetected hardware failures. To overcome the integrity issue GNSS must be augmented and monitored. This can be done using different techniques, usually referred to as either ground, aircraft or satellite based augementation systems. Here we will focus on Aircraft Based Augmentation Systems (ABAS), where the principle is to use other onboard navigation sensors. A typical sensor for augmentation with GNSS is the Inertial Navigation System (INS). The INS operates through a set of rather complicated differential equations describing the vehicle's motion. Input to the equations are measured angular rate and acceleration, making INS independent of external sources.

Accurate and reliable navigation systems are becoming even more important due to the introduction of Unmanned Aerial Vehicles (UAVs). Requirements on navigation accuracy and reliability are amplified since the pilot in manned aircraft also function as a monitor of system performance. Requirements are also put on sensor cost and weight, at least for small tactical UAVs. No single, stand-alone sensor is capable of meeting all requirements. The remedy is to adopt the concept of integrated navigation. Integrated navigation means that the output from two or more navigation sensors are blended to obtain better accuracy and robustness than what the individual sensors can achieve. In many cases, the Inertial Navigation System (INS) is typically seen as the primary source of navigation data. One of the reasons is that INS besides position and velocity also provides the vehicle's orientation, which is important for autonomous control. However, its navigation accuracy degrades with time as sensor errors are mathematically integrated through the navigation equations. The standard system for stabilizing the INS drift is currently the

Global Positioning System (GPS) [18, 21, 11]. The GPS signal is however weak making it sensitive to intentional or unintentional disturbances. A possible and interesting supplement to GPS is Terrain-Aided Positioning (TAP), since it can serve as a monitor and back-up system to GPS. The principle of TAP is to use terrain height information to estimate position, see Figure 1.1.

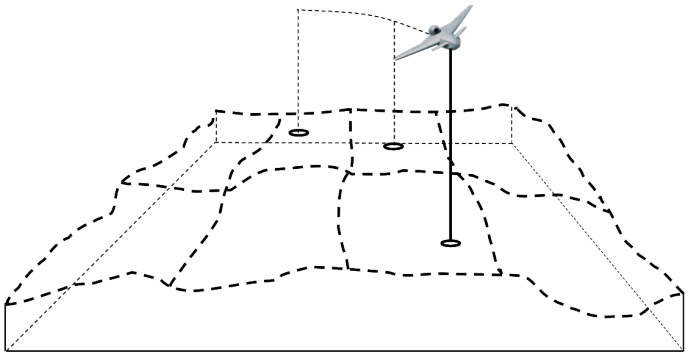

Figure 1.1: *Terrain-aided positioning.*

Here we will study INS integrated with TAP. The challenge with TAP is how to deal with its highly nonlinear, non-analytical nature. The terrain height measurements can either be processed sequentially one by one, which is the method under consideration here, or collected into batches and then processed using profile matching [39]. The particle filter constitutes a generic tool for recursive state estimation of arbitrary systems at the expense of a high computational load. In Paper A the marginalized particle filter is described. It is, when applicable, superior to the standard particle filter. It uses linear or nearly linear substructures which, conditionally upon the nonlinear part, is estimated using linear filters. The remaining nonlinear part is estimated by the particle filter. It is shown that the computational load can be reduced significantly.

There are similar results to be found in the literature on the marginalized particle filter as those presented in this thesis, although obtained independently and of slightly different forms. The most equivalent result can be found in [24], where the aim is also to partition the state vector and apply the particle filter on the truly nonlinear part only. They refer to the resulting filter as the mixture Kalman filter. In [27, 4] similar results are also obtained, although the formulations are rather different. An application is given in [5], where they apply the partitioning technique on amplitude and phase modulated signals. Moreover, similar techniques applied on jump Markov linear systems are given in [28, 32]. For early versions of the result presented here see [79, 78], which are refined in [97].

For integrated INS/TAP we can extend the idea of marginalization to handle a third discrete substructure. In Paper B we provide the details of the filter applied to INS/TAP and show that excellent performance is achieved for a tractable amount of computational load.

1.2 Detection for Collision Avoidance

The purpose of collision avoidance systems is to minimize the risk of collision between vehicles, see Figure 1.2. In an encounter between two manned vehicles, where a collision is imminent, the pilot of each vehicle will normally initiate actions to avoid the collision. There are also procedures to follow and in a near-collision scenario the right-of-way rule applies. In a scenario where one of the vehicles is an UAV the situation is different. If no

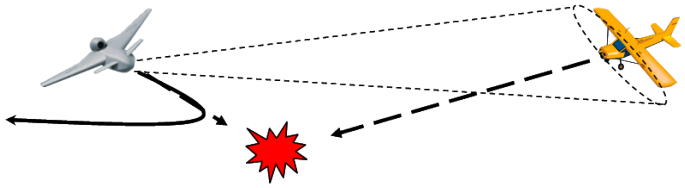

Figure 1.2: *Collision scenario.*

measures are taken there is only one pilot to detect a hazardous situation and react accordingly. To put all responsibility on the pilot in the manned vehicle is deemed unacceptable.

1.2.1 Manned Air Traffic

For manned aircraft today there are different layers of safety, as illustrated by Figure 1.3. The safety measures described below are applicable either partially or as a whole depending on airspace and aircraft. The outer layer consists of procedures. All flights have to comply with rules and procedures as decided by authorities. For example in some cases one needs to file a flight plan, describing among other things your destination, and receive an Air Traffic Control (ATC) clearance. The next layer is separation, a service provided by ATC, where an air traffic controller keeps track of aircraft by surveillance radar and/or transponder. The controller detects aircraft on collision course and informs the pilots how to maneuver in order to avoid the potential conflict. The third layer consists of a cooperative collision aviodance through the use of Traffic Alert and Collision Avoidance System (TCAS) [67]. TCAS automatically detects and evaluates cooperative traffic by interrogating transponders on speed, height and bearing and may advise the pilot to climb or descend to avoid a collision. The inner layer consists of see and avoid, a function which relies on the pilot to see an incoming aircraft and maneuver if necessary. A close encounter or a Near Mid-Air Collision (NMAC) is declared if the minimum distance between two aircraft is less than 150 m [35, 10].

The airspace is partitioned into classes depending on, among other things, altitude [33, 35, 104]. The classification is rather complex and can differ from country to country. One of the most important airspaces for tactical UAVs is class E. Class E includes the major part of the lower airspace up to roughly 5000 m. Within this airspace for aircraft flying by Visual Flight Rules (VFR) [35, 104] radio communication with ATC is not required and separation assistance from ATC is not likely. This implies additional requirements on the capability of see and avoid.

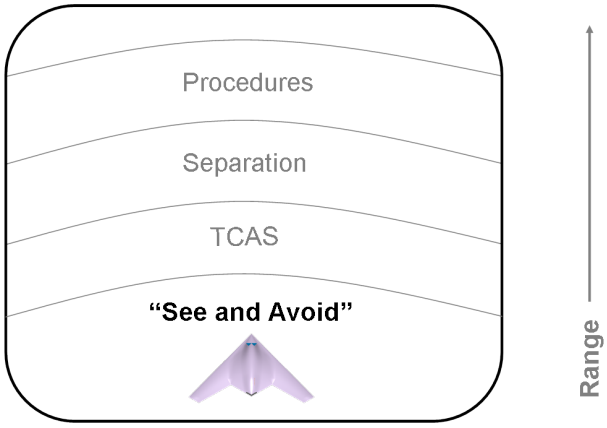

Figure 1.3: *Layers of safety in controlled airspace.*

1.2.2 Unmanned Aerial Vehicles

For the introduction of UAVs into civil airspace the authority requires an Equivalent Level Of Safety (ELOS). This means that the overall safety accomplished for manned aircraft as described above is also required for an unmanned aircraft. This implies that it is not sufficient, for example, to replace the human see and avoid capability with an automatic sense and avoid system. The whole chain of safety measures, from procedures down to see and avoid, needs to be considered to be able to claim ELOS. For example, a seemingly simple task of implementing TCAS on Remotely Piloted Vehicles (RPVs) is in reality not easy [66, 110]. The concerns of implementing TCAS on RPVs are of such a magnitude that the recommendation is *not* to equip with TCAS, at least until the opposite is demonstrated [66]. The concerns are primarily additional delays due to communication links for manual response and criticality of TCAS for autonomous response [110].

There are conceptually two different ways to quantify ELOS. Either we study the human capability and try to mimic it, or we analyze historical statistics on NMACs and empirically determine what a system must achieve. An example on the first method is given in [76], where they present established requirements on parameters such as detection ranges for the US military RPVs Global Hawk and Predator. Examples on the latter method are given by [96] based on 10-year statistics for mid-air collisions in the United States airspace, and [61] based on surveillance radar readings over United States airspace and extracting those encounters which involve VFR aircraft. In [7] a study on TCAS performance is reported, based on an assumed NMAC rate for European airspace. The result of the study indicates a TCAS conflict resolution capability of the original NMACs around 90%, although a great deal new, induced NMACs were also created.

Here we will focus on the inner layer, i.e. to replace the pilot with an automatic sense and avoid in order to avoid NMACs. Although the equivalent level of safety put requirements on a sense and avoid system for unmanned aircraft in any controlled airspace, it is within class E and against aircraft flying VFR we have the most obvious need for sense and avoid. There are many proposals for collision avoidance or conflict resolution, see the survey given by [69]. We are interested in the probability of NMAC for a predicted

trajectory. The most similar result to the ones presented in this thesis is found in the part on short range conflict detection in [91]. The main difference is that in [91] the initial condition is assumed known.

The process for certification of TCAS involved detailed safety studies, numerous encounter simulations and exhaustive field tests. The certification of a sense and avoid system must surely pass a similar process [70]. All actions taken to shorten the process are of course valuable and means that the cost to develop a system is decreased. The development and verification would benefit from a system design which minimize the amount of encounter simulations. For example having a system which per design has the capability of providing correct collision risk measure for a given encounter would be highly beneficial. As will be shown this is one of the main motivators why it is of interest to compute probability of NMAC for a predicted trajectory.

Based on conditions defining a near mid-air collision (NMAC) anytime in the future we can compute the probability of NMAC using the Monte Carlo method. Monte Carlo means that we draw samples from a probability density function defining the initial relative state between the vehicles. The samples are then simulated along the predicted flight trajectory and for each sample the outcome, NMAC or not NMAC, is recorded. The mean of the outcome of the samples constitutes an empirical estimate of the probability of NMAC. The Monte Carlo method is however computer intensive mainly due to the small probabilities that the application is required to detect. Typically the Monte Carlo method is not possible to use for real-time processing. Instead, in Paper C we propose a method for computing collision risk as accurate as the Monte Carlo method but with a significantly reduced computational load. The result is extended to three dimensions in Paper D.

1.3 Outline

1.3.1 Outline of Part I

Part I consists of two main subparts; estimation in Section 2 and detection in Section 3. The estimation part starts in Section 2.1 by providing a more detailed description of terrain-aided navigation. Recursive state estimation is presented in Section 2.2, and for linear, multiple and nonlinear models in particular. Section 2.3 deals with the particle filter including asymptotic properties. Section 2.4 is devoted to the marginalized particle filter. The detection part starts with giving a more detailed description of the sense and avoid system in Section 3.1. Definition of probability of NMAC is given in Section 3.2 and detection principles using hypotheses is provided in Section 3.3. In Section 3.4 we introduce the reader to the concept of using level-crossings as the mean to compute probability of conflict. We start with one dimension, then two dimensions, and finally three dimensions. Part I is concluded in Section 4, including a discussion on possible future research.

1.3.2 Outline of Part II

Part II consists of a collection of papers which constitute the main contributions of the thesis.

Paper A: Marginalized Particle Filters for Mixed Linear/Nonlinear State-space Models

T.B. Schön, F. Gustafsson, and P-J. Nordlund. Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing*, 53(7):2279–2289, July 2005.

Summary: The particle filter offers a general numerical tool to approximate the filtering density function for the state in nonlinear and non-Gaussian filtering problems. While the particle filter is fairly easy to implement and tune, its main drawback is that it is quite computer intensive, with the computational complexity increasing quickly with the state dimension. One remedy to this problem is to marginalize out the states appearing linearly in the dynamics. The result is that one Kalman filter is associated with each particle. The main contribution in this paper is to derive the details for the marginalized particle filter for a general nonlinear state-space model. Several important special cases occurring in typical signal processing applications are also discussed. The marginalized particle filter is applied to a simplified integrated aircraft navigation system assuming a known altitude. It is demonstrated that the complete high-dimensional system can be based on a particle filter using marginalization for all but two states. Excellent performance on real flight data is reported.

Background and contribution: The article is based on the theory given in [78], which provides the derivation of the marginalized particle filter. The main parts of the article is credited to T.B. Schön who has also refined the results.

Paper B: Marginalized Particle Filter for Accurate and Reliable Aircraft Navigation

P-J. Nordlund and F. Gustafsson. Marginalized particle filter for accurate and reliable terrain-aided navigation. *Accepted for publication in IEEE Transactions on Aerospace and Electronic Systems*, 2008.
http://www.control.isy.liu.se/research/reports/2008/2870.pdf.

Summary: This paper details an approach to the integration of INS (Inertial Navigation System) and TAP (Terrain-Aided Positioning). The solution is characterized by a joint design of INS and TAP, meaning that the highly nonlinear TAP is not designed separately but jointly with the INS using one and the same filter. The applied filter extends the theory of the MPF (Marginalized Particle Filter) given by [98]. The key idea with MPF is to estimate the nonlinear part using the particle filter and the part which is linear, conditionally upon the nonlinear part, is estimated using the Kalman filter. The extension lies in the possibility to deal with a third multi-modal part, where the discrete mode variable is also estimated jointly with the linear and nonlinear parts. Conditionally upon the mode and the nonlinear part, the resulting subsystem is linear and estimated using the Kalman filter. Given the nonlinear motion equations which the INS uses to compute navigation data, the INS equations must be linearized for the MPF to work. A set of linearized equations is derived and the linearization errors are shown to be insignificant with respect to the final

result. Simulations are performed and the result indicates near-optimal accuracy when compared to the Cramer-Rao lower bound.

Background and contribution: Here the marginalized particle filter is extended to cover systems which, besides linear and nonlinear substructures, also contains a discrete unknown mode. The linearization errors when applying Taylor expansion on the INS navigation equations are shown to be insignificant. The resulting filter is applied on a complete integrated INS/TAP system.

Paper C: Probabilistic Conflict Detection for Piecewise Straight Paths

P-J. Nordlund and F. Gustafsson. Probabilistic conflict detection for piecewise straight paths. *Submitted to Automatica*, 2008.
http://www.control.isy.liu.se/research/reports/2008/2871.pdf.

Summary: We consider probabilistic methods for detecting conflicts as a function of predicted trajectory. A conflict is an event representing collision or imminent collision between vehicles or objects. The computations use state estimate and covariance from a target tracking filter based on sensor readings. Existing work is primarily concerned with risk estimation at a certain time instant, while the focus here is to compute the integrated risk over the critical time horizon. This novel formulation leads to evaluating the probability for level-crossing. The analytic expression involves a multi-dimensional integral which is hardly tractable in practice. Further, a huge number of Monte Carlo simulations would be needed to get sufficient reliability for the small risks that the applications often require. Instead, we propose a sound numerical approximation that leads to evaluating a one-dimensional integral which is suitable for real-time implementations.

Background and contribution: Here we derive an efficient method for computing probability of conflict in two dimensions using theory for level-crossings. The method is derived for not only linear motion but also a relative motion which follows a piecewise straight path.

Paper D: Probabilistic Near Mid-Air Collision Avoidance

P-J. Nordlund and F. Gustafsson. Probabilistic near mid-air collision avoidance. *Submitted to IEEE Transactions on Aerospace and Electronic Systems*, 2008.
http://www.control.isy.liu.se/research/reports/2008/2872.pdf.

Summary: We propose a probabilistic method to compute the near mid-air collision risk as a function of predicted flight trajectory. The computations use state estimate and covariance from a target tracking filter based on angle-only sensors such as digital video cameras. The majority of existing work is focused on risk estimation at a certain time instant. Here we derive an expression for the integrated risk over the critical time horizon. This is possible using probability for level-crossing, and the expression applies to a three-dimensional piecewise straight flight trajectory. The Monte Carlo technique provides a

method to compute the probability, but a huge number of simulations is needed to get sufficient reliability for the small risks that the applications require. Instead we propose a method which through sound geometric and numerical approximations yield a solution suitable for real-time implementations. The algorithm is applied to realistic angle-only tracking data, and shows promising results when compared to the Monte Carlo solution.

Background and contribution: Here we derive the conditions for a near mid-air collision to occur. Through a sound geometrical approximation and the extension of Paper C to piecewise linear motion in three dimensions we derive an efficient method for computing probability of near mid-air collision. The resulting method is applied to realistic angle-only tracking data.

1.4 Other Publications

Publications not included but of related interest are:

T. Hektor, H. Karlsson, and P-J. Nordlund. A marginalized particle filter approach to an integrated INS/TAP system. In *Proceedings of the IEEE/ION Position, Location and Navigation Symposium*, pages 766–770, May 2008.

P-J. Nordlund and F. Gustafsson. The probability of near midair collisions using level-crossings. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008.

F. Gustafsson, T.B. Schön, R. Karlsson, and P-J. Nordlund. State-of-the-art for the marginalized particle filter. In *Proceedings of the IEEE Nonlinear Statistical Signal Processing Workshop*, pages 172–174, Sept 2006.

P-J. Nordlund, F. Gunnarsson, and F. Gustafsson. Particle filters for positioning in wireless networks. In *Procedings of EUSIPCO*, Toulouse, France, September 2002.

F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, R. Karlsson, and P-J. Nordlund. Particle filters for positioning, navigation and tracking. *IEEE Transactions on signal processing*, 50(2):425–437, Feb 2002.

P-J. Nordlund and F. Gustafsson. Recursive estimation of three-dimensional aircraft position using terrain-aided positioning. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Proceesing*, volume 2, pages 1121–1124, 2002.

F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, R. Karlsson, and P-J. Nordlund. A framework for particle filtering in positioning, navigation and tracking problems. In *Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing*, pages 34–37, Aug 2001.

P-J. Nordlund and F. Gustafsson. Sequential Monte Carlo filtering techniques applied to integrated navigation systems. In *Proceedings of the 2001 American Control Conference*, volume 6, pages 4375–4380, 2001.

2

Efficient State Estimation

2.1 Terrain-Aided Navigation

The purpose of Terrain-Aided Navigation (TAN), as for any other navigation system, is to provide an accurate and reliable estimate of the kinematic state of the own platform, where the state typically consists of position and its derivatives. The navigation solution from TAN is based on sensors which do not rely on external information sources. This makes TAN resistant to disturbances and jamming. The principle for TAN is to integrate information from a Inertial Navigation System (INS) and a Terrain-Aided Positioning (TAP) system, see Figure 2.1. The TAP system is basically a Radar Altimeter (RA) measuring distance to ground and a database with stored terrain height. The RA measurement subtracted from INS altitude provides a terrain height measurement. The database gives terrain height for sampled horizontal positions, ranging from a couple of meters to several hundred depending on the database. In the case studied here terrain height is given at every 50 meter. Terrain height measurements are matched with stored terrain height. The points where stored height matches measured height yield aircraft position candidates. Gradually the measurements form a terrain height profile with fewer and fewer matches in the database. Thereby the position candidates become fewer and fewer until there is only one left, see Figure 2.2 for an illustration. The idea of using terrain as a navigation aid is not new, see [19] for a interesting historical survey on terrain navigation systems.

The task of navigating using terrain height can be cast as recursive state estimation problem. Let the navigation quantities, such as position, velocity and orientation, be comprised in the state vector x_t. The expression for terrain height provided by the terrain information system is

$$h(x_t^{\mathrm{pos}}), \tag{2.1}$$

where $h(x_t^{\mathrm{pos}})$ is the terrain height given by the database as a function of horizontal position. Note that no analytical expression exists for $h(\cdot)$. Also, usually several different

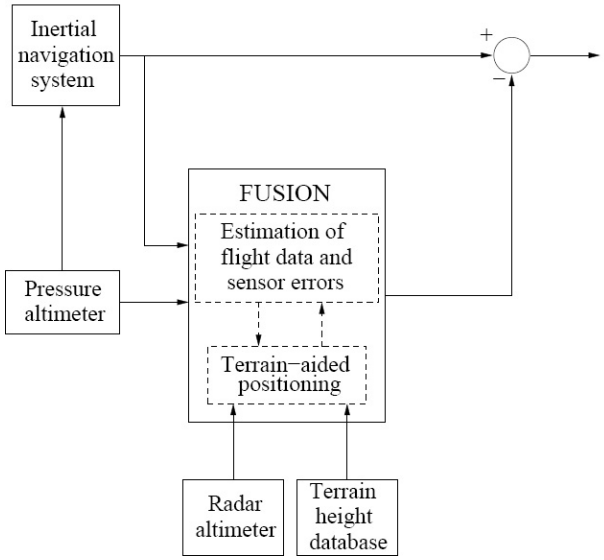

Figure 2.1: *Concept of integrated INS/TAP.*

positions yield the same height making the problem multimodal. This makes the particle filter ideal for this application. The inertial navigation system computes the state x_t from measured acceleration and angular rate u_t. The computations are based on a set of rather complicated nonlinear differential equations, in discrete time given by $f(x_t, u_t)$. However, in paper B we show that the nonlinearities are weak. The expression for time update of INS data then become

$$x_{t+1} = f(x_t, u_t) \approx F_t x_t + G_t u_t. \tag{2.2}$$

By splitting the state vector

$$x_t = \begin{bmatrix} x_t^{\text{pos}} \\ x_t^{\text{nav}} \end{bmatrix}, \tag{2.3}$$

the prerequisites for applying the marginalized particle filter are at place.

2.2 Recursive Estimation

We consider discrete-time state space descriptions with additive noise

$$\begin{aligned} x_{t+1} &= f(x_t) + G_t(x_t)u_t, \\ y_t &= h(x_t) + e_t, \end{aligned} \tag{2.4}$$

where $x_t \in \mathbb{R}^{n_x}$ represents the unknown state vector and $y_t \in \mathbb{R}^{n_y}$ is the observation. The subscript t denotes a discrete-time index assuming a sampling time T sec. The process noise u_t and measurement noise e_t are stochastic processes with known probability

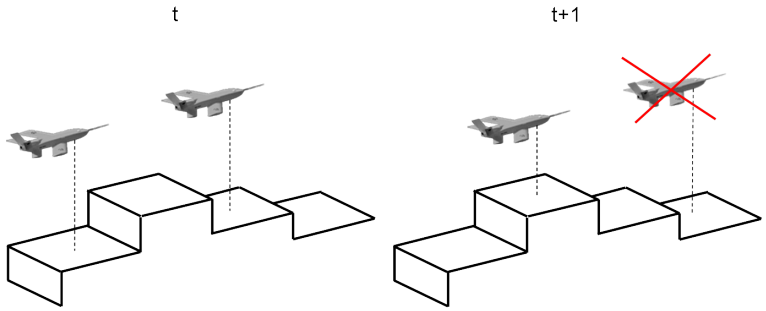

Figure 2.2: *Principle of terrain-aided positioning, with two position candidates at time t but only one at time t+1.*

densities $p(u_t)$ and $p(e_t)$. The noise sequences are both assumed white and independent of each other, i.e.

$$\begin{aligned} p(u_{t+k}, u_t) &= p(u_{t+k})p(u_t), \quad \forall k \neq 0, \\ p(e_{t+l}, e_t) &= p(e_{t+l})p(e_t), \quad \forall l \neq 0, \\ p(u_t, e_t) &= p(u_t)p(e_t). \end{aligned} \tag{2.5}$$

We denote by $\mathbb{X}_t = \{x_0, \ldots, x_t\}$ and $\mathbb{Y}_t = \{y_0, \ldots, y_t\}$ the stacked vector of all the states and measurements up to time t. By the assumptions on the noise according to (2.5) and the definition of conditional density [44]

$$p(x|y) = \frac{p(x,y)}{p(y)} \tag{2.6}$$

one can show that x_t is a Markov process

$$p(\mathbb{X}_t) = \prod_{k=0}^{t} p(x_k|x_{k-1}), \tag{2.7}$$

where $p(x_0|x_{-1}) = p(x_0)$, and that the conditional density of the measurements given the states are independent

$$p(\mathbb{Y}_t|\mathbb{X}_t) = \prod_{k=0}^{t} p(y_k|x_k) = \prod_{k=0}^{t} p_{e_t}(y_k - x_k). \tag{2.8}$$

The goal is to compute or estimate the *posterior* probability density function $p(x_t|\mathbb{Y}_t)$. When a new measurement is available we want our estimate to be updated using the new information. To avoid having to re-calculate everything we need recursive expressions for how the new information should be incorporated. Using Bayes' formula [52] and the assumptions (2.7)–(2.8) we can derive the following recursions for the posterior probability

$$p(x_t|\mathbb{Y}_t) = \frac{p(y_t|x_t)p(x_t|\mathbb{Y}_{t-1})}{p(y_t|\mathbb{Y}_{t-1})}, \tag{2.9a}$$

$$p(y_t|\mathbb{Y}_{t-1}) = \int_{\mathbb{R}^{n_x}} p(y_t|x_t)p(x_t|\mathbb{Y}_{t-1})dx_t, \tag{2.9b}$$

referred to as the *measurement update*. The *time update* is provided by the law of total probability

$$p(x_{t+1}|\mathbb{Y}_t) = \int_{\mathbb{R}^{n_x}} p(x_{t+1}|x_t)p(x_t|\mathbb{Y}_t)dx_t. \tag{2.10}$$

Based on the posterior probability we can compute the point estimate $\hat{x}_{t|t}$ which minimizes the expectation of the estimation error squared, i.e.

$$\begin{aligned}\hat{x}_{t|t} &= \arg\min_{\hat{x}_t} \mathbb{E}_{p(x_t|\mathbb{Y}_t)}\left[(x_t - \hat{x}_t)(x_t - \hat{x}_t)^T\right] \\ &= \arg\min_{\hat{x}_t} \int_{\mathbb{R}^{n_x}} (x_t - \hat{x}_t)(x_t - \hat{x}_t)^T p(x_t|\mathbb{Y}_t)dx_t.\end{aligned} \tag{2.11}$$

The estimate $\hat{x}_{t|t}$ is given by Theorem 2.1.

Theorem 2.1 (Least-mean-square estimation)
Given two random variables (scalar- or vector-valued) x_t and $\mathbb{Y}_t$, the optimal least-mean-square estimate of x_t given $\mathbb{Y}_t$ is

$$\hat{x}_{t|t} = \mathbb{E}_{p(x_t|\mathbb{Y}_t)}[x_t]. \tag{2.12}$$

Proof: See [55]. □

2.2.1 Linear Models

For linear estimation we have the Kalman filter, thoroughly described in e.g. [55]. Consider the linear state-space model

$$\begin{aligned}x_{t+1} &= F_t x_t + G_t u_t, \\ y_t &= H_t x_t + e_t,\end{aligned} \tag{2.13}$$

where

$$\begin{aligned}&\mathbb{E}\begin{bmatrix}u_t \\ e_t \\ x_0\end{bmatrix}\begin{bmatrix}u_{t+k}^T & e_{t+k}^T & x_0^T & 1\end{bmatrix} = \begin{bmatrix}Q_t\delta_k & 0 & 0 & 0 \\ 0 & R_t\delta_k & 0 & 0 \\ 0 & 0 & P_0 & \hat{x}_0\end{bmatrix}, \\ &\delta_k = \begin{cases}1 & \text{if } k = 0 \\ 0 & \text{otherwise.}\end{cases}\end{aligned} \tag{2.14}$$

With Gaussian distributed u_t, e_t and x_0, i.e.

$$u_t \sim \mathcal{N}(0, Q_t), \quad e_t \sim \mathcal{N}(0, R_t), \quad x_0 \sim \mathcal{N}(\hat{x}_0, P_0), \tag{2.15}$$

the optimal filter is the Kalman filter, see Theorem 2.2.

Theorem 2.2 (The Kalman filter)
Consider the linear state-space model according to (2.13)*. Moreover, assume that the noise processes have the properties according to* (2.14)–(2.15) *Then* x_t *and* x_{t+1}*, conditionally upon* $\mathbb{Y}_t$*, are Gaussian distributed for any time* $t \geq 0$*, i.e.*

$$\begin{aligned} p(x_t|\mathbb{Y}_t) &= \mathcal{N}(\hat{x}_{t|t}, P_{t|t}), \\ p(x_{t+1}|\mathbb{Y}_t) &= \mathcal{N}(\hat{x}_{t+1|t}, P_{t+1|t}), \end{aligned} \tag{2.16}$$

where the mean and covariance propagate through the measurement update

$$\begin{aligned} \hat{x}_{t|t} &= \hat{x}_{t|t-1} + P_{t|t-1}H_t^T S_t^{-1}(y_t - H_t\hat{x}_{t|t-1}), \\ P_{t|t} &= P_{t|t-1} - P_{t|t-1}H_t^T S_t^{-1} H_t P_{t|t-1}, \\ S_t &= R_t + H_t P_{t|t-1} H_t^T, \end{aligned} \tag{2.17}$$

and the time update

$$\begin{aligned} \hat{x}_{t+1|t} &= F_t\hat{x}_{t|t}, \\ P_{t+1|t} &= F_t P_{t|t} F_t^T + G_t Q_t G_t^T, \end{aligned} \tag{2.18}$$

with initial values $\hat{x}_{0|-1} = \hat{x}_0$ *and* $P_{0|-1} = P_0$*.*

Proof: See [40, 3, 55]. □

Note that for non-Gaussian distributed u_t, e_t and x_0 the best *linear* least-mean-square solution is still provided by the Kalman filter [55].

2.2.2 Multiple Models

In the multiple model approach we extend the linear model to cover the case with multiple linear models. The multitude is handled by a random variable λ_t which can take a finite number of discrete values $\{1, 2, \ldots, M\}$. The model under consideration is

$$\begin{aligned} x_{t+1} &= F_t(\lambda_{t+1})x_t + G_t(\lambda_{t+1})u_t(\lambda_{t+1}), \\ y_t &= H_t(\lambda_t)x_t + e_t(\lambda_t), \end{aligned} \tag{2.19}$$

and we assume λ_t follows a discrete Markov process

$$\Pr(\Lambda_t) = \prod_{k=0}^{t} \Pr(\lambda_k|\lambda_{k-1}), \tag{2.20}$$

where $\Pr(\lambda_0|\lambda_{-1}) = \Pr(\lambda_0)$. The mode transition probabilities $\Pr(\lambda_t|\lambda_{t-1})$ are given by the matrix

$$\begin{bmatrix} \Pr(1|1) & \ldots & \Pr(1|M) \\ \vdots & & \vdots \\ \Pr(M|1) & \ldots & \Pr(M|M) \end{bmatrix}. \tag{2.21}$$

From (2.19) we see that if we know the mode sequence, $\Lambda_t = \{\lambda_0, \ldots, \lambda_t\}$, we are back to the original linear model according to (2.13), with the solution given by the Kalman filter. In practice, we usually do not have access to the mode sequence and we are confined to use the probability of each possible sequence $\Pr(\Lambda_t|\mathbb{Y}_t)$. Using the law of total probability we can write the filtering posterior probability density as

$$p(x_t|\mathbb{Y}_t) = \sum_{\Lambda_t} p(x_t|\mathbb{Y}_t, \Lambda_t)\Pr(\Lambda_t|\mathbb{Y}_t). \tag{2.22}$$

The number of possible mode sequences grows exponentially with time, there are M^{t+1} different possibilities to account for at time t, starting from time $t = 0$.

One way to limit the number of mode sequences is to only account for the most recent time instances, say the last L. This means that we only need to account for M^L different hypotheses, and it enables us to keep the computational load constant over time. At each update of the filters we first split each mode sequence into M new ones. Each mode sequence is now $L + 1$ time instances long. We then apply a Kalman filter on each extended mode sequence. Next we merge M sequences which are now equal when considering only the L most recent time instances. This is the principle behind the *Generalized Pseudo-Bayesian* (GPB) filter, see Algorithm 2.1 for $L = 1$. The measurement update of the weight $\alpha_{t|t}^{(\lambda_t)} = \Pr(\lambda_t|\mathbb{Y}_t)$ is given by

$$\begin{aligned}\Pr(\lambda_t|\mathbb{Y}_t) &\propto \Pr(\lambda_t|\mathbb{Y}_{t-1}) \int p(y_t|\lambda_t, x_t)p(x_t|\lambda_t, \mathbb{Y}_{t-1})dx_t \\ &= e^{-\frac{1}{2}(y_t - H_t(\lambda_t)\hat{x}_{t|t-1}^{(\lambda_t)})^T S_t^{-1}(\lambda_t)(y_t - H_t(\lambda_t)\hat{x}_{t|t-1}^{(\lambda_t)})}\Pr(\lambda_t|\mathbb{Y}_{t-1}).\end{aligned} \tag{2.23}$$

Algorithm 2.1 (The GPB1 filter).

1. For $j = 1, \ldots, M$, initialize the weights $\bar{\alpha}_0^{(\lambda_0)} = \Pr(\lambda_0)$ and the estimates $\hat{x}_0^{(\lambda_0)}$, $P_0^{(\lambda_0)}$.

2. Estimate measurement update: Compute $\hat{x}_{t|t}^{(\lambda_t)}$, $P_{t|t}^{(\lambda_t)}$ using (2.17) conditionally upon λ_t.

3. Weight measurement update: Compute new weights using

$$\begin{aligned}\alpha_{t|t}^{(\lambda_t)} &= e^{-\frac{1}{2}(y_t - H_t(\lambda_t)\hat{x}_{t|t-1}^{(\lambda_t)})^T S_t^{-1}(\lambda_t)(y_t - H_t(\lambda_t)\hat{x}_{t|t-1}^{(\lambda_t)})}\bar{\alpha}_{t|t-1}^{(\lambda_t)}, \\ \bar{\alpha}_{t|t}^{(\lambda_t)} &= \frac{\alpha_{t|t}^{(\lambda_t)}}{\sum_{\lambda_t=1}^{M} \alpha_{t|t}^{(\lambda_t)}}\end{aligned}$$

4. Merge:

$$\begin{aligned}\hat{x}_{t|t} &= \sum_{\lambda_t=1}^{M} \bar{\alpha}_{t|t}^{(\lambda_t)}\hat{x}_{t|t}^{(\lambda_t)}, \\ P_{t|t} &= \sum_{\lambda_t=1}^{M} \bar{\alpha}_{t|t}^{(\lambda_t)}(P_{t|t}^{(\lambda_t)} + (\hat{x}_{t|t}^{(\lambda_t)} - \hat{x}_{t|t})(\hat{x}_{t|t}^{(\lambda_t)} - \hat{x}_{t|t})^T).\end{aligned}$$

5. Estimate time update: Compute $\hat{x}_{t+1|t}^{(\lambda_{t+1})}$, $P_{t+1|t}^{(\lambda_{t+1})}$ using (2.18) conditionally upon λ_{t+1}.

6. Weight time update: Compute new weights using

$$\bar{\alpha}_{t+1|t}^{(\lambda_{t+1})} = \sum_{\lambda_t=1}^{M} \Pr(\lambda_{t+1}|\lambda_t)\bar{\alpha}_{t|t}^{(\lambda_t)}$$

7. Iterate from step 2.

The conceptually similar *Interacting Multiple Model* (IMM) filter uses a slighly more sophisticated way of computing the weights [16, 12, 56]. See [106] for a nice survey and comparison of different multiple model approaches.

2.2.3 Nonlinear Models

When we are facing a nonlinear estimation problem given by (2.4) we can still use the Kalman filter recursions given by (2.17)–(2.17). The difference is that we use a set of linearized equations given by the first order terms in the Taylor expansion around the current estimate

$$H_t = \left.\frac{\partial h(x_t)}{\partial x_t}\right|_{x_t=\hat{x}_{t|t-1}}, \quad F_t = \left.\frac{\partial f(x_t)}{\partial x_t}\right|_{x_t=\hat{x}_{t|t}}. \tag{2.24}$$

The result is referred to as the extended Kalman filter (EKF) [55, 3, 93]. More sophisticated versions of the EKF exist which account for the second order terms in the Taylor expansion [49].

The Gaussian sum filter [102, 2, 3] can be seen as an extension to the EKF. The idea is to split the estimation using several EKFs working in parallel, where each EKF is concentrated on a subspace of the overall posterior probability density. The overall posterior probability is approximately given by a mixture of Gaussians

$$p(x_t|Y_t) \approx \sum_{i=1}^{N} \bar{w}_t^{(i)} \mathcal{N}(\hat{x}_{t|t}^{(i)}, P_{t|t}^{(i)}), \quad \sum_{i=1}^{N} \bar{w}_t^{(i)} = 1, \tag{2.25}$$

where the mean and covariance of each Gaussian is provided by a dedicated EKF. Typically the splitting is based on the initial values $\hat{x}_0$ and P_0 such that the covariance for each EKF $P_0^{(i)}$ is small enough. Small enough here means that the linearized equations should provide an accurate description around $\hat{x}_{t|t}^{(i)}$ as long as we do not deviate more than given by $P_0^{(i)}$. For an example on a Gaussian sum filter application see Example 2.1.

Example 2.1

The Range Parameterized EKF (RP-EKF) is a method for tracking other vehicles based on angle-only measurements. Tracking with angle-only measurements normally results in highly uncertain range estimates. The principle for RP-EKF is to split the applicable range into subintervals, each having the same so-called coefficient of variation, C_r. The

C_r is defined as the ratio between standard deviation and mean of each range segment. The performance of angle-only tracking is highly dependent on C_r, the lower the better [88]. A large uncertainty in range compared to the mean yields a high C_r. Suppose the range is uniformly distributed over the interval $(r_{\min}, r_{\max})$, which yields

$$C_r = \frac{\sigma_r}{\hat{r}} = \frac{\frac{r_{\max}-r_{\min}}{\sqrt{12}}}{\frac{r_{\max}+r_{\min}}{2}} \tag{2.26}$$

We can achieve a smaller C_r by splitting the range interval into a number N_f of subintervals, where subinterval i is defined by $(r_{\min}\rho^{i-1}, r_{\min}\rho^i)$ and $\rho = \left(r_{\max}/r_{\min}\right)^{1/N_f}$. Now it is straightforward to show that $C_r^{(i)}$ is the same for each segment and is given by [9]

$$C_r^{(i)} = \frac{\sigma_r^{(i)}}{\hat{r}^{(i)}} = \frac{2(\rho-1)}{\sqrt{12}(\rho+1)}. \tag{2.27}$$

The principle of the point-mass filter [64, 65, 13] is to discretize the support of the posterior density into a finite set of points. The expression for the posterior probability becomes

$$p(x_t|Y_t) \approx \sum_{i=1}^{N} \bar{w}_t^{(i)} \delta_{x_t^{(i)}}(x_t), \tag{2.28}$$

where $\delta(\cdot)$ is the Dirac delta function.

2.3 Particle Filter

Particle filters, or sequential Monte Carlo methods [31, 38], can be seen as simulation-based methods, i.e. we simulate a large number of possible state trajectories using e.g. the time propagation equations from (2.4). Based on the measurements we associate a weight to each trajectory, using the measurement equation from (2.4). The realizations together with the associated weigths provide an empirical approximation of the joint posterior distribution. Probably two of the first references on particle filtering are [45] and [46]. In these papers importance sampling, described in Section 2.3.1 and 2.3.2, was introduced on sequential estimation problems. The crucial step of resampling, described in Section 2.3.3, was however first introduced in [38]. The two ingredients, importance sampling and resampling, form the basis for the particle filter algorithm given in Section 2.3.4.

2.3.1 Importance Sampling

Suppose that we have drawn N *independent identically distributed* (i.i.d.) samples of x_t from the probability density $q(x_t|y_t)$. In the literature, the function $q(x_t|y_t)$ is usually

referred to as the *importance function*, see e.g. [30]. The first requirement on this distribution is that it should be possible to draw samples from it. See e.g. [53] for details on how to create samples from a distribution. We can use the set of samples to form an empirical approximation of $q(x_t|y_t)$, i.e.

$$q(x_t|y_t) \approx \frac{1}{N}\sum_{i=1}^{N} \delta_{x_t^{(i)}}(x_t), \tag{2.29}$$

where $\delta(\cdot)$ is the Dirac delta function. What (2.29) really means is that the probability mass for a small volume is approximated by the relative frequency of samples that are located within the volume. A more correct notation would therefore be

$$q(dx_t|y_t) \approx \frac{1}{N}\sum_{i=1}^{N} \delta_{x_t^{(i)}}(dx_t), \tag{2.30}$$

but this is not pursued further. What we really would like is to be able to draw realizations from $p(x_t|y_t)$, referred to as *perfect sampling*, but unfortunately this is not possible in general. The best we can hope for is that $q(x_t|y_t)$ is as close to $p(x_t|y_t)$ as possible. Let us introduce the importance weight defined as

$$w(x_t) = \frac{p(x_t|y_t)}{q(x_t|y_t)}, \tag{2.31}$$

or in the Bayesian framework, where the normalization constant $p(y_t)$ in general is analytically intractable,

$$w(x_t) = \frac{p(y_t|x_t)p(x_t)}{q(x_t|y_t)}. \tag{2.32}$$

See Example 2.2 for the most basic choice of importance function.

Example 2.2

A natural choice of importance function is to use the prior probability density for the state vector

$$q(x_t|y_t) = p(x_t). \tag{2.33}$$

The weights are then computed according to

$$w(x_t) = \frac{p(y_t|x_t)p(x_t)}{p(x_t)} = p(y_t|x_t). \tag{2.34}$$

The importance weight, $w(x_t)$, can be looked upon as a measure of the skewness of $q(x_t|y_t)$ relative to $p(x_t|x_t)$. From (2.31) or (2.32) we see that for the importance weight to be defined the support of $p(x_t|y_t)$ must be included in the support of $q(x_t|y_t)$. Based

on the realizations from (2.29) and the importance weights computed using (2.32) we can express an empirical approximation of the posterior density $p(x_t|y_t)$ according to[1]

$$\begin{aligned} p(x_t|y_t) &= \frac{w(x_t)q(x_t|y_t)}{\int w(x_t)q(x_t|y_t)dx_t} \\ &\approx \frac{\sum_{i=1}^{N} w_t^{(i)}\delta_{x_t^{(i)}}(x_t)}{\sum_{i=1}^{N} w_t^{(i)}} = \sum_{i=1}^{N} \bar{w}_t^{(i)}\delta_{x_t^{(i)}}(x_t), \end{aligned} \tag{2.35}$$

where we have defined

$$w_t^{(i)} = \frac{p(y_t|x_t^{(i)})p(x_t^{(i)})}{q(x_t^{(i)}|y_t)}, \quad \bar{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^{N} w_t^{(j)}}. \tag{2.36}$$

Using (2.35) we can now perform any kind of operation that previously was intractable, e.g. computing any kind of estimate. For any integrable function $g(x_t)$ we can estimate its expected value according to

$$\begin{aligned} \hat{g}_t &= \mathbb{E}_{p(x_t|y_t)}[g(x_t)] = \int g(x_t)p(x_t|y_t)dx_t \\ &\approx \int g(x_t)\sum_{i=1}^{N} \bar{w}_t^{(i)}\delta_{x_t^{(i)}}(x_t)dx_t = \sum_{i=1}^{N} \bar{w}_t^{(i)}g(x_t^{(i)}) = \hat{g}_t^N. \end{aligned} \tag{2.37}$$

Using (2.37) we see that the least-mean-square estimate $\hat{x}_{t|t}$ and its covariance are estimated using $g(x_t) = x_t$ and $g(x_t) = (x_t - \hat{x}_{t|t})(x_t - \hat{x}_{t|t})^T$ respectively, i.e.

$$\hat{x}_{t|t} \approx \sum_{i=1}^{N} \bar{w}_t^{(i)} x_t^{(i)}, \tag{2.38a}$$

$$P_{t|t} \approx \sum_{i=1}^{N} \bar{w}_t^{(i)}(x_t^{(i)} - \hat{x}_{t|t})(x_t^{(i)} - \hat{x}_{t|t})^T. \tag{2.38b}$$

2.3.2 Recursive Importance Sampling

The question now is how to recursively obtain a set of realizations

$$\{\mathbb{X}_t^{(i)}\}_{i=1}^N = \{x_0^{(i)}, \dots, x_t^{(i)}\}_{i=1}^N \tag{2.39}$$

which together with the weights $\{\bar{w}_t^{(i)}\}_{i=1}^N$ are distributed approximately according to $p(\mathbb{X}_t|\mathbb{Y}_t)$. The new set of realizations and weights should be based upon the realizations $\{\mathbb{X}_{t-1}^{(i)}\}_{i=1}^N$ and importance weights $\{\bar{w}_{t-1}^{(i)}\}_{i=1}^N$.

Suppose the realizations $\{\mathbb{X}_{t-1}^{(i)}\}_{i=1}^N$ have been drawn from some arbitrary but known density $q(\mathbb{X}_{t-1}|\mathbb{Y}_{t-1})$, i.e.

$$q(\mathbb{X}_{t-1}|\mathbb{Y}_{t-1}) \approx \frac{1}{N}\sum_{i=1}^{N} \delta_{\mathbb{X}_{t-1}^{(i)}}(\mathbb{X}_{t-1}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{x_0^{(i)}}(x_0)\cdots\delta_{x_{t-1}^{(i)}}(x_{t-1}), \tag{2.40}$$

[1] Note that we will in the sequel not explicitly state the space on which the integration is taken place, here $(\mathbb{R}^{n_x})^{(t+1)}$, assuming it is clear from the context.

which together with the importance weights are such that

$$p(\mathbb{X}_{t-1}|\mathbb{Y}_{t-1}) \approx \sum_{i=1}^{N} \bar{w}_{t-1}^{(i)} \delta_{\mathbb{X}_{t-1}^{(i)}}(\mathbb{X}_{t-1}). \tag{2.41}$$

We now wish to extend the existing realizations and modify the existing weights, such that we obtain a set of realizations which, together with the modified weights, is approximately distributed according to $p(\mathbb{X}_t|\mathbb{Y}_t)$. The importance function can be rewritten recursively, using Bayes' rule, according to

$$q(\mathbb{X}_t|\mathbb{Y}_t) = q(x_t|\mathbb{X}_{t-1}, \mathbb{Y}_t) q(\mathbb{X}_{t-1}|\mathbb{Y}_t). \tag{2.42}$$

In order to be a practical method the new measurement y_t should not affect the existing realizations $\{\mathbb{X}_{t-1}^{(i)}\}_{i=1}^{N}$. Otherwise, we would be forced to create new realizations for the state history at each time t, which would imply a computational load which grows as $\mathcal{O}(t)$. Therefore, we require that

$$q(\mathbb{X}_{t-1}|\mathbb{Y}_t) = q(\mathbb{X}_{t-1}|\mathbb{Y}_{t-1}). \tag{2.43}$$

Inserting (2.40) into (2.42) we obtain an expression for how to produce new samples for x_t

$$q(x_t|\mathbb{Y}_t) = \int q(\mathbb{X}_t|\mathbb{Y}_t) d\mathbb{X}_{t-1} \approx \sum_{i=1}^{N} q(x_t|\mathbb{X}_{t-1}^{(i)}, \mathbb{Y}_t). \tag{2.44}$$

To create a new set of samples $\{x_t^{(i)}\}_{i=1}^{N}$ based on (2.44) we draw N times from the importance function according to

$$x_t^{(i)} \sim q(x_t|\mathbb{X}_{t-1}^{(i)}, \mathbb{Y}_t), \quad i = 1, \ldots, N. \tag{2.45}$$

The last step is to obtain a properly weighted set of realizations with respect to $p(\mathbb{X}_t|\mathbb{Y}_t)$. What we need is an expression for how the new weight $w(\mathbb{X}_t)$ is computed recursively in terms of $w(\mathbb{X}_{t-1})$. We have

$$w(\mathbb{X}_t) = \frac{p(\mathbb{Y}_t|\mathbb{X}_t)p(\mathbb{X}_t)}{q(\mathbb{X}_t|\mathbb{Y}_t)} = \frac{p(y_t|x_t)p(x_t|x_{t-1})}{q(x_t|\mathbb{X}_{t-1}, \mathbb{Y}_t)} w(\mathbb{X}_{t-1}). \tag{2.46}$$

2.3.3 Resampling

If the weights $\{w_t^{(i)}\}_{i=1}^{N}$ are very skewed, i.e. most of the weights are close to zero and only a few of them are significant, most of the samples do not contribute to the approximation. To create a set of samples with equal weights we can perform a so-called resampling [101]. This means that we multiply those samples with a large importance weight, and samples with small weights are discarded. Call the number of offsprings from each sample $N_t^{(i)}$. To meet the requirement that the unweighted set of samples is still approximately distributed according to $p(\mathbb{X}_t|\mathbb{Y}_t)$ we need

$$\mathbb{E}[N_t^{(i)}] = N\bar{w}_t^{(i)}. \tag{2.47}$$

Note however that one could be deceived to think that resampling achieves the goal of having a set of samples perfectly matched to $p(\mathbb{X}_t|\mathbb{Y}_t)$. But the resampling introduces another problem, we obtain a set of *dependent* samples. However, to make the sequential importance sampling technique operational the resampling procedure is necessary.

The standard resampling method is the *multinomial resampling*, originally introduced in [38]. The principle is to pick a realization from $\{\mathbb{X}_t^{(i)}\}_{i=1}^N$, with replacement, where the probability to pick $\mathbb{X}_t^{(i)}$ is equal to $\bar{w}_t^{(i)}$. This procedure can be achieved in $\mathcal{O}(N)$ operations, see e.g. [20] for the details.

A second method known to introduce less additional variance than multinomial resampling, is *residual resampling* [74]. The principle behind residual resampling is to first multiply/discard particles deterministically according to $\lfloor N\bar{w}_t^{(i)} \rfloor$, where the notation $\lfloor x \rfloor$ corresponds to the integer part of x. The second step is to perform multinomial resampling on the rest, i.e. on

$$N_t^{\text{rest}} = N - \sum_{i=1}^{N} \lfloor N\bar{w}_t^{(i)} \rfloor. \tag{2.48}$$

A third strategy is to multiply and discard samples entirely deterministically, a *deterministic resampling* [60]. This can be achieved by, for each $i = 1, \ldots, N$, finding the largest integer $N_t^{(i)}$ such that

$$\frac{N_t^{(i)} - \Delta}{N} < \bar{w}_t^{(i)}, \quad 0 \leq \Delta < 1. \tag{2.49}$$

2.3.4 The Algorithm

For a summary of the particle filter see Algorithm 2.2.

Algorithm 2.2 (The particle filter).

1. For $i = 1, \ldots, N$, sample $x_0^{(i)} \sim p(x_0)$ and set $w_{-1}^{(i)} = \frac{1}{N}$
2. For each $i = 1, \ldots, N$, update $w_t^{(i)} = \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|\mathbb{X}_{t-1}^{(i)}, \mathbb{Y}_t)} \bar{w}_{t-1}^{(i)}$ and normalize $\bar{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_j w_t^{(j)}}$.
3. If resampling (e.g. if $\hat{N}_{\text{eff}} < N_{\text{th}}$), apply one of the resampling procedures described in Section 2.3.3 on $\{\bar{w}_t^{(i)}\}_{i=1}^N$.
4. For $i = 1, \ldots, N$, sample $x_{t+1}^{(i)} \sim q(x_{t+1}|\mathbb{X}_t^{(i)}, \mathbb{Y}_{t+1})$.
5. Iterate from step 2.

A simple and intuitive choice for importance function, which is also the choice made in [38], is to use

$$q(x_t|\mathbb{X}_{t-1}, \mathbb{Y}_t) = p(x_t|x_{t-1}). \tag{2.50}$$

The importance weights are then updated with

$$w(\mathbb{X}_t) = p(y_t|x_t)w(\mathbb{X}_{t-1}). \tag{2.51}$$

The problem with choosing the weights according to (2.51) is that we could end up with very skewed weights, particularly if the measurement y_t is very informative (informative in the sense that the support of $p(y_t|x_t)$ is concentrated compared to $p(x_t|\mathbb{Y}_{t-1})$).

From (2.46) we see that if we choose

$$q(x_t|\mathbb{X}_{t-1}, \mathbb{Y}_t) = p(x_t|x_{t-1}, y_t) = \frac{p(y_t|x_t)p(x_t|x_{t-1})}{p(y_t|x_{t-1})}, \tag{2.52}$$

the weights according to (2.46) are updated with

$$w(\mathbb{X}_t) = p(y_t|x_{t-1})w(\mathbb{X}_{t-1}), \tag{2.53}$$

which is independent of x_t. This means that the optimal choice of $q(x_t|\mathbb{X}_{t-1}, \mathbb{Y}_t)$ with respect to making the weights as uniform as possible with respect to x_t, is given by (2.52) [30]. The problem with the choice in (2.53) is that it only works for certain systems. More specifically we must be able to compute

$$p(y_t|x_{t-1}) = \int p(y_t|x_t)p(x_t|x_{t-1})dx_t, \tag{2.54}$$

which is not possible in the general case. Examples on when it is analytically tractable are when the function $h(\cdot)$ in (2.4) is linear with respect to x_t and the process and measurement noises are Gaussian distributed (or distributed as Gaussian mixtures). Another special case where (2.54) is analytically tractable is when x_t only takes on a finite number of different values, changing the integral to a summation.

A third alternative is to use only those realizations $\mathbb{X}_{t-1}^{(i)}$ which provide a high likelihood for y_t. This is possible, before we draw new samples for x_t, by evaluating e.g. $p(y_t|f(x_{t-1}^{(i)}))$, where $f(x_{t-1}^{(i)})$ is the prediction of $x_{t-1}^{(i)}$ given by the state-space model (2.4). A resampling at this stage, where the probability to pick $\mathbb{X}_{t-1}^{(i)}$ is $p(y_t|f(x_{t-1}^{(i)}))$, would yield the desired result. This is the principle behind the auxiliary particle filter, see [89], and we have

$$w(\mathbb{X}_t) = \frac{p(y_t|x_t)}{p(y_t|f(x_{t-1}))}w(\mathbb{X}_{t-1}). \tag{2.55}$$

2.3.5 Asymptotic Properties

One can show that, under mild assumptions, a central limit theorem holds for the importance sampling method, see Theorem 2.3.

Theorem 2.3 (Central limit theorem for importance sampling)
Assume i.i.d. samples $\{\mathbb{X}_t^{(i)}\}_{i=1}^N$*, the support of* $q(\mathbb{X}_t|\mathbb{Y}_t)$ *includes that of* $p(\mathbb{X}_t|\mathbb{Y}_t)$*, and the expectations*

$$\mathbb{E}_{p(\mathbb{X}_t|\mathbb{Y}_t)}[g(x_t)], \quad \mathbb{E}_{p(\mathbb{X}_t|\mathbb{Y}_t)}[w(\mathbb{X}_t)], \quad \mathbb{E}_{p(\mathbb{X}_t|\mathbb{Y}_t)}[g(x_t)g(x_t)^T w(\mathbb{X}_t)],$$

exists and are finite. Then we have convergence in distribution for $\hat{g}_t^N$ according to

$$\begin{aligned}\sqrt{N}(\hat{g}_t^N - \hat{g}_t) &\xrightarrow{d} \mathcal{N}(0, \Sigma_t),\\ \Sigma_t &= \frac{\mathbb{E}_{q(\mathbb{X}_t|\mathbb{Y}_t)}\left[w(\mathbb{X}_t)^2(g(x_t) - \hat{g}_t)(g(x_t) - \hat{g}_t)^T\right]}{\left(\mathbb{E}_{q(\mathbb{X}_t|\mathbb{Y}_t)}\left[w(\mathbb{X}_t)\right]\right)^2},\end{aligned} \tag{2.56}$$

as $N \to \infty$. Moreover, a consistent estimate of the covariance Σ_t is given by

$$\widehat{\mathrm{cov}}(\hat{g}_t^N) = \sum_{i=1}^{N} \left(\bar{w}_t^{(i)}\right)^2 (g(x_t^{(i)}) - \hat{g}_t^N)(g(x_t^{(i)}) - \hat{g}_t^N)^T, \tag{2.57a}$$

$$N\widehat{\mathrm{cov}}(\hat{g}_t^N) \xrightarrow{a.s.} \Sigma_t. \tag{2.57b}$$

Proof: See [36]. □

We see from (2.57b) that the sample covariance of $\hat{g}_t^N$ tends to zero as $\mathcal{O}(N^{-1})$.

Remark 2.1. Note that in the perfect sampling case, i.e. $q(\mathbb{X}_t|\mathbb{Y}_t) = p(\mathbb{X}_t|\mathbb{Y}_t)$, and with $g(x_t) = x_t$ we have

$$\Sigma_t = P_{t|t}, \tag{2.58}$$

where $P_{t|t}$ is the covariance of the estimate $\hat{x}_{t|t}$.

As a measure of sampling efficiency we can use the ratio between the sample variance when using a set of samples drawn from the posterior directly (perfect sampling) and the sample variance obtained through the use of importance sampling. The sampling efficiency in its turn gives an expression for the effective sample size, $N_{\mathrm{eff}} \leq N$. In [63] it is shown that the sampling efficiency, assuming a scalar-valued function $g(x_t)$, is approximately given by

$$\hat{N}_{\mathrm{eff}} = \frac{\sum_i (g(x_t^{(i)}) - \hat{g}_t^N)^2 \bar{w}_t^{(i)}}{\sum_i (g(x_t^{(i)}) - \hat{g}_t^N)^2 (\bar{w}_t^{(i)})^2}. \tag{2.59}$$

When we apply resampling we will lose the independence among the samples, and thereby destroying the basic property for the convergence results to hold. A reasonable question is whether it is necessary to perform any kind of resampling at all. The problem here is that the variance of the importance weights can only increase with time. This property is shown in [63] and it implies that as time progresses the weights will become more and more nonuniform. A standard method for deciding when to resample is to use the expression for approximate sampling efficiency according to (2.59). As soon as $\hat{N}_{\mathrm{eff}} < N_{\mathrm{th}}$, where N_{th} is some predefined threshold, a resampling is carried out. Note however that after the first resampling this expression must be used with care since it assumes independent samples.

For the resampling to have its intended effect, i.e. to increase the effective sample size, the sampling step must be such that it scatters the samples. Otherwise we will experience what is referred to as the *depeletion* problem. To overcome the depletion problem we can add artificial noise to the system, u_t^{add}. The idea here is to use more process noise in our

model than what the true system really exhibits. Note that the additional noise should only be added after a resampling. A standard choice for the distribution for u_t^{add} is

$$u_t^{\text{add}} \sim \mathcal{N}(0, \kappa P_{t|t}), \tag{2.60}$$

where $P_{t|t}$ is estimated covariance of $\hat{x}_{t|t}$ computed for example according to (2.38b) and the discount factor κ is a small number typically in the order of $0.01 - 0.001$. The idea of introducing additional noise is not such a bad idea, considering the fact that we discretize the state vector and in turn the densities involved. A way of dealing with discretization errors is to add uncertainty in form of additional noise. The κ should therefore be such that it accounts for the discretization error, i.e. κ should be a function of N. The drawback with introducing additional noise is that we will inevitably lose information, meaning that the estimated covariance of $\hat{x}_{t+1|t}$ is larger than the actual covariance. One way of dealing with this is to introduce correlation between x_t and u_t^{add} such that the covariances for $\hat{x}_{t+1|t}$ and $\hat{x}_{t|t}$ are the same, see [73] for more information on this subject. More sophisticated methods for handling the depletion problem is to use kernel theory on the samples [77], or to apply Markov Chain Monte Carlo (MCMC) methods [37].

Some convergence results for the particle filter exist. In [37] a central limit theorem is presented for the case of sequential importance sampling with resampling and MCMC moves. A more interesting result is given in [25] on mean-square convergence, see Theorem 2.4.

Theorem 2.4 (Convergence for bounded functions)
Assume $\sup_{x_t \in \mathbb{R}^{n_x}} |p_{e_t}(y_t - h(x_t))| < \infty$. *Then there exists* c_t *independent of* N *such that for any bounded function* $g(x_t)$

$$\mathbb{E}\big[(\hat{g}_t^N - \hat{g}_t)^2\big] \leq c_t \frac{\sup_{x_t \in \mathbb{R}^{n_x}} |g(x_t)|}{N}, \tag{2.61}$$

where $\hat{g}_t^N$ *and* $\hat{g}_t$ *are given by* (2.37).

Proof: See [25]. □

There are two problems with Theorem 2.4. First of all, it only includes *bounded* functions $g(x_t)$, i.e. the standard least-mean-square estimate using $g(x_t) = x_t$ is not covered. Secondly, c_t can very well grow with time, implying that an ever increasing number of particles has to be used. To ensure a uniform convergence ($c_t = c$), requirements must be put on the dynamic model to exponentially forget any error. In practice this means that for example the more noise the system exhibits, the better the particle filter should be working. A very interesting result on convergence for the particle filter is given by [50]. This result includes *unbounded* functions $g(x_t)$, e.g. the least-mean-square estimate $\hat{x}_{t|t}$, and is given by Theorem 2.5.

Theorem 2.5 (Convergence for unbounded functions)
For mild assumptions on the unbounded function $g(x_t)$ *and involved densities, see [50] for details, there exists* c_t *independent of* N *such that*

$$\mathbb{E}\big[(\hat{g}_t^N - \hat{g}_t)^4\big] \leq c_t(\mathbb{Y}_t) \frac{\max_{s=0,\ldots,t}\{1, \mathbb{E}_{p(x_s|\mathbb{Y}_s)}[g^4(x_s)]\}}{N^2}, \tag{2.62}$$

where $\hat{g}_t^N$ *and* $\hat{g}_t$ *are given by* (2.37).

Proof: See [50]. □

A problem with Theorem 2.5 is that it is not uniform with respect to the measurements $\mathbb{Y}_s$. Moreover, the applicability of c_t is not obvious as it typically grows rather quickly with time t.

2.4 Marginalized Particle Filter

Theorems 2.4 and 2.5 show that asymptotically the rate of convergence as a function of N is $\mathcal{O}(N^{-1/2})$. Note that this does not mean that the particle filter is independent of the state dimension. On the contrary, in [26] they demonstrate that computational complexity with respect to dimension is between polynomial and exponential, depending on the problem and chosen particle filter algorithm. This is also confirmed in practice, where it is easily shown by comparing problems of different dimensions that there is a dependence. One possibility that can achieve a significant reduction of the sampling variance is to take advantage of potential structure within the state-space model. If there is any structure that can be solved using analytical, closed-form estimation techniques then we should of course do so. For the remaining parts to be estimated we can resort to sampling techniques. The remaining parts are however of lower dimension, meaning that we should not have to use as many samples as before for the same accuracy.

2.4.1 General Description

Consider a state-space model which can be written on the form

$$x_{t+1}^{\mathrm{n}} = f_t^{\mathrm{n}}(x_t^{\mathrm{n}}) + F_t^{\mathrm{n}} x_t^{\mathrm{l}} + G_t^{\mathrm{n}} u_t^{\mathrm{n}}, \tag{2.63a}$$

$$x_{t+1}^{\mathrm{l}} = f_t^{\mathrm{l}}(x_t^{\mathrm{n}}) + F_t^{\mathrm{l}} x_t^{\mathrm{l}} + G_t^{\mathrm{l}} u_t^{\mathrm{l}}, \tag{2.63b}$$

$$y_t = h_t(x_t^{\mathrm{n}}) + H_t x_t^{\mathrm{l}} + e_t, \tag{2.63c}$$

where $x_t = \begin{bmatrix}(x_t^{\mathrm{n}})^T & (x_t^{\mathrm{l}})^T\end{bmatrix}^T$. The superscripts n and l denote what part of the state vector is nonlinear and linear respectively. We assume the noises and initial distribution for x_t^{l} have the properties given by (2.14)– (2.15). The essence of the marginalized particle filter is the factorization of the pdf

$$p(x_t^{\mathrm{l}}, \mathbb{X}_t^{\mathrm{n}} | \mathbb{Y}_t) = p(x_t^{\mathrm{l}} | \mathbb{X}_t^{\mathrm{n}}, \mathbb{Y}_t) p(\mathbb{X}_t^{\mathrm{n}} | \mathbb{Y}_t). \tag{2.64}$$

By using the Rao-Blackwellization technique we use the fact that $p(x_t^{\mathrm{l}} | \mathbb{X}_t^{\mathrm{n}}, \mathbb{Y}_t)$ exists as a closed-form expression, and that $\mathbb{E}_{p(x_t^{\mathrm{l}} | \mathbb{X}_t^{\mathrm{n}}, \mathbb{Y}_t)}[g(x_t^{\mathrm{l}})]$ exists and is possible to evaluate. Recall the expression for the expected value of an integrable function $g(x_t)$ from (2.37) repeated here for convenience

$$\hat{g}_t = \mathbb{E}_{p(x_t | \mathbb{Y}_t)}[g(x_t)] \approx \sum_{i=1}^{N} w_t^{(i)} g(x_t^{(i)}) = \hat{g}_t^N. \tag{2.65}$$

The expected value of $g(x_t) = g(x_t^{\mathrm{l}}, x_t^{\mathrm{n}})$ can be written according to

$$\begin{aligned}
\hat{g}_t &= \mathbb{E}_{p(x_t|\mathbb{Y}_t)}[g(x_t)] = \mathbb{E}_{p(\mathbb{X}_t^{\mathrm{n}}, x_t^{\mathrm{l}}|\mathbb{Y}_t)}[g(x_t^{\mathrm{n}}, x_t^{\mathrm{l}})] \\
&= \int \mathbb{E}_{p(x_t^{\mathrm{l}}|\mathbb{X}_t^{\mathrm{n}}, \mathbb{Y}_t)}[g(x_t^{\mathrm{n}}, x_t^{\mathrm{l}})] p(\mathbb{X}_t^{\mathrm{n}}|\mathbb{Y}_t) d\mathbb{X}_t^{\mathrm{n}} \\
&= \frac{\mathbb{E}_{q(\mathbb{X}_t^{\mathrm{n}}|\mathbb{Y}_t)}\left[\mathbb{E}_{p(x_t^{\mathrm{l}}|\mathbb{X}_t^{\mathrm{n}}, \mathbb{Y}_t)}[g(x_t^{\mathrm{n}}, x_t^{\mathrm{l}})] w(\mathbb{X}_t^{\mathrm{n}})\right]}{\mathbb{E}_{q(\mathbb{X}_t^{\mathrm{n}}|\mathbb{Y}_t)}[w(\mathbb{X}_t^{\mathrm{n}})]} \\
&\approx \sum_{i=1}^{N} \bar{w}_t^{(i)} \mathbb{E}_{p(x_t^{\mathrm{l}}|\mathbb{X}_t^{\mathrm{n},(i)}, \mathbb{Y}_t)}[g(x_t^{\mathrm{n},(i)}, x_t^{\mathrm{l}})] = \hat{g}_{\mathrm{R},t}^{N}.
\end{aligned} \tag{2.66}$$

From (2.66) we see that the mean and covariance of $g(x_t) = x_t^{\mathrm{n}}$ are the same as before, i.e. computed according to (2.38a) and (2.38b). However, the mean and covariance of x_t^{l} are from (2.66) given by

$$\begin{aligned}
\hat{x}_{t|t}^{\mathrm{l}} &\approx \sum_{i=1}^{N} \bar{w}_t^{(i)} \mathbb{E}_{p(x_t^{\mathrm{l}}|\mathbb{X}_t^{\mathrm{n},(i)}, \mathbb{Y}_t)}[x_t^{\mathrm{l}}] \\
&= \sum_{i=1}^{N} \bar{w}_t^{(i)} \hat{x}_{t|t}^{\mathrm{l},(i)}, \\
P_{t|t}^{\mathrm{l}} &\approx \sum_{i=1}^{N} \bar{w}_t^{(i)} \mathbb{E}_{p(x_t^{\mathrm{l}}|\mathbb{X}_t^{\mathrm{n},(i)}, \mathbb{Y}_t)}[(x_t^{\mathrm{l}} - \hat{x}_{t|t}^{\mathrm{l}})(x_t^{\mathrm{l}} - \hat{x}_{t|t}^{\mathrm{l}})^T] \\
&= \sum_{i=1}^{N} \bar{w}_t^{(i)} \big(P_{t|t}^{\mathrm{l},(i)} + (\hat{x}_{t|t}^{\mathrm{l},(i)} - \hat{x}_{t|t}^{\mathrm{l}})(\hat{x}_{t|t}^{\mathrm{l},(i)} - \hat{x}_{t|t}^{\mathrm{l}})^T\big).
\end{aligned} \tag{2.67}$$

In the expressions (2.67) we have used that the estimate and covariance of x_t^{l}, *conditionally upon a given realization* $\mathbb{X}_t^{\mathrm{n},(i)}$, denoted by $\hat{x}_{t|t}^{\mathrm{l},(i)}$ and $P_{t|t}^{\mathrm{l},(i)}$ respectively, where

$$P_{t|t}^{\mathrm{l},(i)} = \mathbb{E}_{p(x_t^{\mathrm{l}}|\mathbb{X}_t^{\mathrm{n},(i)}, \mathbb{Y}_t)}[(x_t^{\mathrm{l}} - \hat{x}_{t|t}^{\mathrm{l},(i)})(x_t^{\mathrm{l}} - \hat{x}_{t|t}^{\mathrm{l},(i)})^T], \tag{2.68}$$

are analytically tractable. Algorithm 2.3 provides the conceptual steps in the marginalized particle filter. For details the reader is referred to Paper A.

Algorithm 2.3 (Marginalized particle filter).

1. Initialization: Initialize the particles and set initial values for the linear state variables, to be used in the Kalman filter.
2. Particle filter measurement update: evaluate the importance weights and normalize.
3. Resample with replacement (if necessary).
4. Kalman filter measurement update
5. Particle filter time update: Predict new particles.
6. Kalman filter time update.
7. Iterate from step 2.

2.4.2 Filter for Terrain-Aided Navigation

The terrain-navigation application can be posed as a recursive state estimation problem according to

$$x_{t+1}^{\mathrm{n}} = f_t^{\mathrm{n}}(x_t^{\mathrm{n}}) + F_t^{\mathrm{n}} x_t^{\mathrm{l}} + G_t^{\mathrm{n}} u_t^{\mathrm{n}}, \tag{2.69a}$$

$$x_{t+1}^{\mathrm{d}} = f_t^{\mathrm{d}}(x_t^{\mathrm{n}}) + F_t^{\mathrm{d}} x_t^{\mathrm{d}} + G_t^{\mathrm{d}} u_t^{\mathrm{d}}, \tag{2.69b}$$

$$x_{t+1}^{\mathrm{l}} = f_t^{\mathrm{l}}(x_t^{\mathrm{n}}) + F_t^{\mathrm{l}} x_t^{\mathrm{l}} + G_t^{\mathrm{l}} u_t^{\mathrm{l}}, \tag{2.69c}$$

$$y_t = h_t(x_t^{\mathrm{n}}) + H_t x_t^{\mathrm{d}} + e_t(\lambda_t). \tag{2.69d}$$

Here x_t^{n}, x_t^{d} and x_t^{l} correspond to horizontal position, altitude and remaining navigation quantities respectively. Note the mode dependent measure noise $e_t(\lambda_t)$. The purpose of the mode variable λ_t is to model the bimodal character of the radar measurements. For example, flying over dense forrest yields radar measurements which can either correspond to distance to ground or the tree tops. We could apply Algorithm 2.3 to estimate $[x_t^{\mathrm{l}}\ x_t^{\mathrm{d}}]^T$ and x_t^{n} respectively. However, using the GPB1 filter according to Algorithm 2.1, we would need two multi-dimensional Kalman filters running in parallel. The remedy is to use the structural independence in (2.69) between x_t^{d} and x_t^{l}, which makes it possible to further split the estimation. The posterior probability density can be factorized according to

$$p(x_t^{\mathrm{l}}, x_t^{\mathrm{d}}, \mathbb{X}_t^{\mathrm{n}} | \mathbb{Y}_t) = p(x_t^{\mathrm{l}} | \mathbb{X}_t^{\mathrm{n}}, \mathbb{Y}_t) p(x_t^{\mathrm{d}} | \mathbb{X}_t^{\mathrm{n}}, \mathbb{Y}_t) p(\mathbb{X}_t^{\mathrm{n}} | \mathbb{Y}_t). \tag{2.70}$$

This means that, conditionally upon $\mathbb{X}_t^{\mathrm{n}}$, we can estimate x_t^{l} and x_t^{d} independently of each other. We can estimate the mode dependent x_t^{d} with two one-dimensional Kalman filters and apply a single Kalman filter for estimation of x_t^{l}. Algorithm 2.4 provides the conceptual steps in the Terrain-Aided Navigation Filter. For details the reader is referred to Paper B.

Algorithm 2.4 (Terrain-Aided Navigation Filter).

1. Initialization: Initialize the particles and set initial values for the linear state variables, to be used in the Kalman filter.
2. Particle filter measurement update: evaluate the importance weights and normalize.
3. Resample with replacement (if necessary).
4. GPB filter measurement update and merge
5. Kalman filter measurement update
6. Particle filter time update: Predict new particles.
7. Kalman filter time update.
8. GPB filter time update.
9. Iterate from step 2.

The filter from paper B has been tested on authentic flight data as described in [47, 48].

2.4.3 Asymptotic Properties

Similar to Theorem 2.3 one can derive a central limit theorem for the case with the Rao-Blackwellization technique applied. The result is given by Theorem 2.6

Theorem 2.6 (CLT for Rao-Blackwellization)
Assume i.i.d. samples $\{\mathbb{X}_t^{(i)}\}_{i=1}^N$*, the support of* $q(\mathbb{X}_t|\mathbb{Y}_t)$ *includes that of* $p(\mathbb{X}_t|\mathbb{Y}_t)$ *and that the involved expectations exist and are finite. Then we have convergence in distribution for* $\hat{g}_t^N$ *according to*

$$\begin{aligned}
&\sqrt{N}(\hat{g}_{\mathrm{R},t}^N - \hat{g}_t) \xrightarrow{d} \mathcal{N}(0, \Sigma_{\mathrm{R},t}), \\
&\Sigma_{\mathrm{R},t} = \frac{\mathbb{E}_{q(\mathbb{X}_t^{\mathrm{n}}|\mathbb{Y}_t)}\Big[\mathbb{E}_{p(x_t^{\mathrm{l}}|\mathbb{X}_t^{\mathrm{n}},\mathbb{Y}_t)}[(g(x_t)-\hat{g}_t)(g(x_t)-\hat{g}_t)^T]w(\mathbb{X}_t^{\mathrm{n}})^2\Big]}{\big(\mathbb{E}_{q(\mathbb{X}_t^{\mathrm{n}}|\mathbb{Y}_t)}\big[w(\mathbb{X}_t^{\mathrm{n}})\big]\big)^2},
\end{aligned} \tag{2.71}$$

as $N \to \infty$*. A consistent estimate of* $\Sigma_{\mathrm{R},t}$ *and the sample covariance* $\mathrm{cov}(\hat{g}_{\mathrm{R},t}^N)$ *are given by*

$$\begin{aligned}
&\widehat{\mathrm{cov}}(\hat{g}_{\mathrm{R},t}^N) \\
&= \sum_{i=1}^N \big(\mathbb{E}_{p(x_t^{\mathrm{l}}|\mathbb{X}_t^{\mathrm{n},(i)},\mathbb{Y}_t)}[g^{(i)}(x_t) - \hat{g}_t]\big)\big(\mathbb{E}_{p(x_t^{\mathrm{l}}|\mathbb{X}_t^{\mathrm{n},(i)},\mathbb{Y}_t)}[g^{(i)}(x_t) - \hat{g}_t]\big)^T (\bar{w}_t^{(i)})^2, \\
&N\widehat{\mathrm{cov}}(\hat{g}_{\mathrm{R},t}^N) \xrightarrow{a.s.} \Sigma_{\mathrm{R},t}
\end{aligned} \tag{2.72}$$

where $g^{(i)}(x_t) = g(x_t^{\mathrm{n},(i)}, x_t^{\mathrm{l}})$.

Proof: See [29]. □

One can also show that, in accordance with our intuition, the variance obtained when applying Rao-Blackwellization is always less than or equal to the variance when not applying it, see Corollary 2.1.

Corollary 2.1 (Rao-Blackwellization variance reduction)
Assume that the assumptions given in Theorem 2.3 are valid and that the expectations involved exist and are finite. Then

$$\Sigma_t - \Sigma_{\mathrm{R},t} = \frac{\mathbb{E}_{q(\mathbb{X}_t^{\mathrm{n}}|\mathbb{Y}_t)}\Big[\mathrm{cov}_{q(\mathbb{X}_t^{\mathrm{l}}|\mathbb{X}_t^{\mathrm{n}},\mathbb{Y}_t)}\big((g(x_t)-\hat{g}_t)w(\mathbb{X}_t)\big)\Big]}{\big(\mathbb{E}_{q(\mathbb{X}_t|\mathbb{Y}_t)}\big[w(\mathbb{X}_t)\big]\big)^2} \geq 0, \tag{2.73}$$

where Σ_t *and* $\Sigma_{\mathrm{R},t}$ *are given by* (2.56) *and* (2.71) *respectively.*

Proof: See [28]. □

From Lemma 2.1 we conclude that, when the average conditional variance of $g(x_t)$ is high we can gain a lot using Rao-Blackwellization.

To be more specific about what we can gain using Rao-Blackwellization, consider a system according to (2.63) where $H_t \equiv 0$. Compare with terrain-aided navigation (2.69) where $H_t = 0$ with respect to x_t^{l}. In this case we have

$$p(\mathbb{X}_t|\mathbb{Y}_t) = p(\mathbb{X}_t^{\mathrm{l}}|\mathbb{X}_t^{\mathrm{n}}, \mathbb{Y}_t)p(\mathbb{X}_t^{\mathrm{n}}|\mathbb{Y}_t) = p(\mathbb{X}_t^{\mathrm{l}}|\mathbb{X}_t^{\mathrm{n}})p(\mathbb{X}_t^{\mathrm{n}}|\mathbb{Y}_t), \tag{2.74}$$

where the second equality stems from the fact that if $\mathbb{X}_t^{\mathrm{n}}$ is given then $\mathbb{Y}_t$ provides no further information. Based on (2.74) and the assumption that we use $q(\mathbb{X}_t|\mathbb{Y}_t) = p(\mathbb{X}_t)$ as importance function we can simplify the expression for the importance weight according to

$$\begin{aligned} w(\mathbb{X}_t) &= p(\mathbb{Y}_t)\frac{p(\mathbb{X}_t|\mathbb{Y}_t)}{p(\mathbb{X}_t)} = p(\mathbb{Y}_t)\frac{p(\mathbb{X}_t^{\mathrm{l}}|\mathbb{X}_t^{\mathrm{n}})p(\mathbb{X}_t^{\mathrm{n}}|\mathbb{Y}_t)}{p(\mathbb{X}_t^{\mathrm{l}}|\mathbb{X}_t^{\mathrm{n}})p(\mathbb{X}_t^{\mathrm{n}})} \\ &= p(\mathbb{Y}_t)\frac{p(\mathbb{X}_t^{\mathrm{n}}|\mathbb{Y}_t)}{p(\mathbb{X}_t^{\mathrm{n}})} = w(\mathbb{X}_t^{\mathrm{n}}). \end{aligned} \tag{2.75}$$

This means that the importance weight only depends on $\mathbb{X}_t^{\mathrm{n}}$ whether we apply Rao-Blackwellization or not. We can now rewrite (2.73) according to

$$\begin{aligned} \Sigma_t - \Sigma_{\mathrm{R},t} &= \frac{\mathbb{E}_{p(\mathbb{X}_t^{\mathrm{n}})}\left[w(\mathbb{X}_t^{\mathrm{n}})^2 \mathrm{cov}_{p(x_t^{\mathrm{l}}|\mathbb{X}_t^{\mathrm{n}})}\left(g(x_t) - \hat{g}_t\right)\right]}{\left(\mathbb{E}_{p(\mathbb{X}_t^{\mathrm{n}})}\left[w(\mathbb{X}_t^{\mathrm{n}})\right]\right)^2} \\ &\approx N\sum_{i=1}^{N}(\bar{w}_t^{(i)})^2 \mathrm{cov}_{p(x_t^{\mathrm{l}}|\mathbb{X}_t^{\mathrm{n},(i)})}\left(g(x_t^{\mathrm{n},(i)}, x_t^{\mathrm{l}}) - \hat{g}_t\right). \end{aligned} \tag{2.76}$$

From (2.76) we see that for $g(x_t) = x_t^{\mathrm{n}}$ we gain nothing using Rao-Blackwellization in this case, because $\mathrm{cov}_{p(x_t^{\mathrm{l}}|\mathbb{X}_t^{\mathrm{n}})}(x_t^{\mathrm{n}} - \hat{x}_{t|t}^{n}) = 0$. On the other hand, for $g(x_t) = x_t^{\mathrm{l}}$ we have that

$$\mathrm{cov}_{p(x_t^{\mathrm{l}}|\mathbb{X}_t^{\mathrm{n},(i)})}(x_t^{\mathrm{l}} - \hat{x}_{t|t}^{\mathrm{l}}) = P_t^{\mathrm{l},(i)}, \tag{2.77}$$

which inserted into (2.76) yields

$$\Sigma_t - \Sigma_{\mathrm{R},t} \approx N\sum_{i=1}^{N}(\bar{w}_t^{(i)})^2 P_t^{\mathrm{l},(i)}. \tag{2.78}$$

Before we can draw any conclusions we have to consider the resulting variance from (2.72) for $g(x_t) = x_t^{\mathrm{l}}$, which is

$$\Sigma_{\mathrm{R},t} \approx N\sum_{i=1}^{N}(\bar{w}_t^{(i)})^2(\hat{x}_{t|t}^{\mathrm{l},(i)} - \hat{x}_{t|t}^{\mathrm{l}})(\hat{x}_{t|t}^{\mathrm{l},(i)} - \hat{x}_{t|t}^{\mathrm{l}})^T. \tag{2.79}$$

By comparing (2.78) with (2.79) we see that if the covariances $P_t^{\mathrm{l},(i)}$ provided by the Kalman filters are large compared to the corresponding spread of the mean terms $(\hat{x}_{t|t}^{\mathrm{l},(i)} - \hat{x}_{t|t}^{\mathrm{l}})(\hat{x}_{t|t}^{\mathrm{l},(i)} - \hat{x}_{t|t}^{\mathrm{l}})^T$ we should be able to gain a lot by applying Rao-Blackwellization.

In practice we are confined to use a finite number of samples in combination with resampling. In this case, including when the Kalman filter covariances $P_t^{\mathrm{l},(i)}$ are small or even zero, there is a potentially even higher gain in using Rao-Blackwellization. For an illustration see Example 2.3.

Example 2.3: Rao-Blackwellization

Consider the system

$$\begin{aligned} x_{t+1} = \begin{bmatrix} x_t^{\mathrm{n}} \\ x_t^{\mathrm{l}} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_t + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_t. \\ y_t = h(x_t^{\mathrm{n}}) + e_t, \end{aligned} \tag{2.80}$$

In this case

$$x_{t+1}^{\mathrm{l}} = x_t^{\mathrm{l}} + u_t = x_t^{\mathrm{l}} + x_{t+1}^{\mathrm{n}} - x_t^{\mathrm{n}} - x_t^{\mathrm{l}} = x_{t+1}^{\mathrm{n}} - x_t^{\mathrm{n}}, \tag{2.81}$$

where we have used $u_t = x_{t+1}^{\mathrm{n}} - x_t^{\mathrm{n}} - x_t^{\mathrm{l}}$. In words, x_{t+1}^{l} is known given x_{t+1}^{n} and x_t^{n} meaning that $P_{t+1|t}^{\mathrm{l},(i)} = 0$. From (2.78) and (2.79), we can conclude that here we gain nothing from Rao-Blackwellization. However, this is not necessarily true for a finite N in combination with resampling. A study with 10000 simulations and

$$h(x_t^{\mathrm{n}}) = x_t^{\mathrm{n}}, \quad Q = 2^2, \quad R = 5^2, \quad P_0 = \begin{bmatrix} 50^2 & 0 \\ 0 & 10^2 \end{bmatrix}, \tag{2.82}$$

yields the result as shown in Figure 2.3. Here we use a linear measurement equation to be able compute and compare with the optimal estimate given by the Kalman filter. The Root Mean Square Error (RMSE) is computed according to

$$\mathrm{RMSE} = \left(\frac{1}{10000} \sum_{m=1}^{10000} (\hat{x}_{3|2}^{(m)} - x_{3|2}^{\mathrm{true},(m)})(\hat{x}_{3|2}^{(m)} - x_{3|2}^{\mathrm{true},(m)})^T \right)^{1/2}. \tag{2.83}$$

We see that here it is most effective to use both resampling and Rao-Blackwellization. The reason is that resampling by itself, despite introducing dependence, improves estimation performance. Moreover, a resampling in combination with Rao-Blackwellization increases the performance even more. The reason is that if we do not use Rao-Blackwellization we initialize by sampling both x_0^{n} and x_0^{l}. A resampling will result in many samples where $x_0^{\mathrm{n},(i)}$ are the same but also many $x_0^{\mathrm{l},(i)}$. The time update of $x_0^{\mathrm{n},(i)}$ is given by

$$x_{1|0}^{\mathrm{n},(i)} \sim \mathcal{N}\big(x_0^{\mathrm{n},(i)} + x_0^{\mathrm{l},(i)}, Q\big). \tag{2.84}$$

Here many of the samples will still be approximately the same, particularly if Q is small. For the case with Rao-Blackwellization we time update the samples according to

$$x_{1|0}^{\mathrm{n},(i)} \sim \mathcal{N}\big(x_0^{\mathrm{n},(i)}, Q + P_0^{\mathrm{l}}\big). \tag{2.85}$$

We see that the diversity among the samples increases because of the additional covariance term P_0^{l} in (2.85).

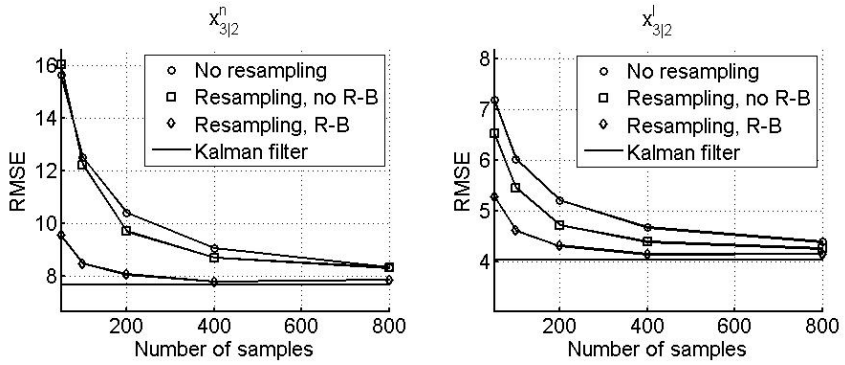

Figure 2.3: *A comparison between* $\hat{x}^{\mathrm{n}}_{3|2}$ *and* $\hat{x}^{\mathrm{l}}_{3|2}$ *for the system* (2.80) *with parameters given by* (2.82) *computed by using no resampling, resampling, resampling and Rao-Blackwellization (R-B), and the Kalman filter. The RMSE is based on* 10000 *simulations. Resampling is applied after the first measurement.*

3

Efficient Detection

3.1 Sense and Avoid

The principle for the sense and avoid system under consideration is depicted in Figure 3.1. The intruder must first be detected by a sensor, in this case a video camera. After detection the sensor feeds a tracking filter with readings of azimuth and elevation angles relative to the intruder. Based on the angle measurements the tracking filter estimates target position and velocity. Finally we compute the probability of near mid-air collision using estimated relative position and velocity. If the probability is large an avoidance maneuver is initiated.

3.1.1 Sensor

There are many sensors applicable for collision avoidance. We will focus on electro-optical sensors, and more specifically video cameras. The advantage with video cameras is that they are much less expensive and readily available as Commercial Off-The-Shelf (COTS) compared to for example radars. Moreover, the bearing accuracy is very high (similar to or better than the human eye). The resolution of a video camera is today typically $0.1 - 1$ mrad. As an example the human eye normally has a spatial resolution of about 0.3 mrad. Video cameras with silicon Charged Coupled Devices (CCDs) have sensitivities well into the near infrared area ($\approx 1 \cdot 10^{-6}$ m), which exceeds the capability of the human eye. The most prominent problems with cameras are that range is not measured and that they are limited to high visibility atmospheric conditions and daytime light conditions. Note however that the visibility conditions apply to the human see and avoid capability as well. Several sensors can be mounted on a rack to cover required field of regard or view, ± 110 deg horisontally and ± 15 deg vertically, see Figure 3.2.

Background movement is measured by analyzing the optical flow between successive frames. An object can then be detected which moves in a manner that deviates from the background [75]. For example, a vehicle on collision course is approximately still in

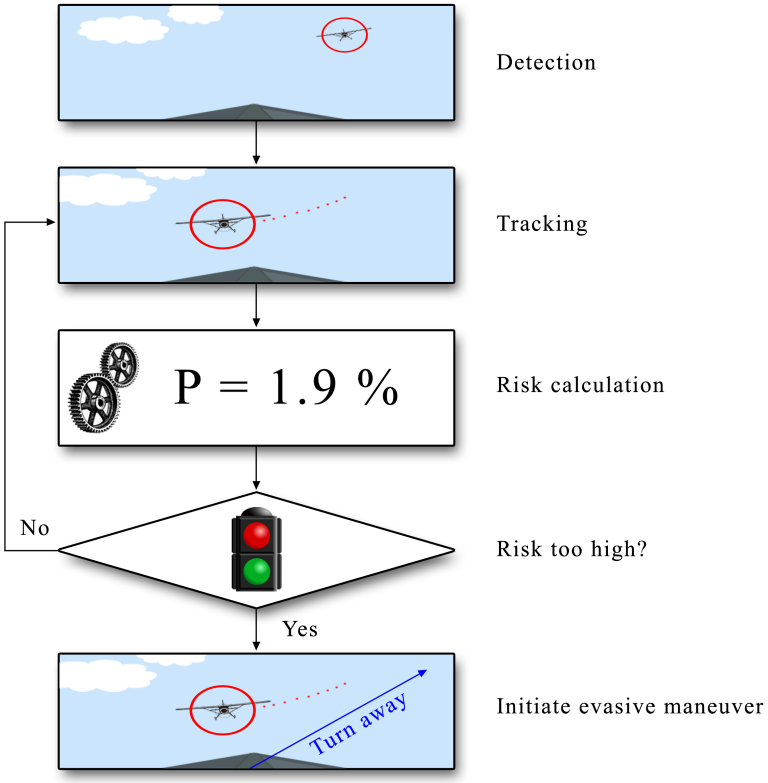

Figure 3.1: *Concept for sense and avoid.*

successive frames and thereby provides the means to distinguish it from the background. Typical detection distance for a video camera with a resolution of 0.3 mrad, assuming good visibility conditions and a target cross-section of $3\ \mathrm{m}^2$, is around 4000 m with 90% probability of detection. This is confirmed by the demonstration described in [75]. The demonstration showed a probability of detection, here defined as the number of correct detections divided by number of frames processed, using off-line processing close to 100% at a distance of 4000 m. At the same time probability of false alarm, here defined as number of false detections divided by number of frames and pixels per frames, was approximately 0.01%. For a comparison the human eye is capable of detecting an intruder at a distance of approximately 3000 m (F-16 with probability 90%) [75]. Intutively, on-line processing with platform vibrations will worsen the result [107].

3.1.2 Tracking

The purpose of target tracking is to estimate the kinematic state of a target or intruder based on noisy measurements. This is a recursive state estimation problem, and the theory in Section 2 is applicable. The state is usually confined to relative position s and velocity

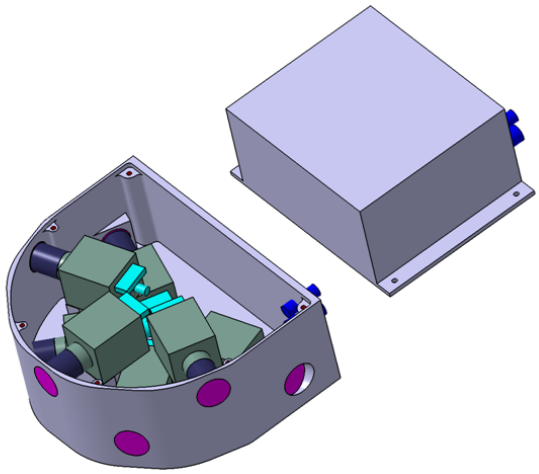

Figure 3.2: *Example on camera mounting.*

v, in Cartesian coordinates given by

$$x^{\text{car}} = \begin{bmatrix} s_X & s_Y & s_Z & v_X & v_Y & v_Z \end{bmatrix}^T. \tag{3.1}$$

Relative position is obtained by subtracting intruder position and own position, i.e.

$$s = s^{\text{int}} - s^{\text{own}}, \tag{3.2}$$

and equivalently for relative velocity. In angle-only tracking the measurements are azimuth and elevation, denoted by η and ϵ respectively. These angles are defined by

$$\eta = \arctan\left(\frac{s_Y}{s_X}\right), \tag{3.3}$$

$$\epsilon = \arctan\left(\frac{s_Z}{\sqrt{s_X^2 + s_Y^2}}\right). \tag{3.4}$$

For good tracking performance it is important to model the motion of the intruder as accurately as possible. The simplest model is based on the assumption that the target has approximately constant velocity, i.e.

$$\begin{bmatrix} \dot{v}_X^{\text{int}} & \dot{v}_Y^{\text{int}} & \dot{v}_Z^{\text{int}} \end{bmatrix}^T \approx \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T. \tag{3.5}$$

If it is inappropriate to assume constant velocity there are other possible models which deal with motion in one way or another [94]. The own vehicle is subject to acceleration particularly during an avoidance maneuver

$$\begin{bmatrix} \dot{v}_X^{\text{own}} & \dot{v}_Y^{\text{own}} & \dot{v}_Z^{\text{own}} \end{bmatrix}^T = \begin{bmatrix} a_X & a_Y & a_Z \end{bmatrix}^T, \tag{3.6}$$

where the acceleration is measured accurately typically by an Inertial Navigation System (INS).

It is a well known fact that distance or range is unobservable for angle-only measurements and a constant relative velocity. To gain full observability the observer platform must out-maneuver the target, i.e. there must be higher non-zero derivative of the observer motion compared to the target [14, 99]. Very inaccurate range estimates can yield large linearization errors. This can affect the other, observable states such that they diverge. In other words, the unobservable range can affect tracking filter stability.

There are two widely used methods for angle-only tracking, Range Parameterized EKF (RP-EKF) and Modified Spherical EKF (MS-EKF). Both are known to provide stable estimates of the target state vector. The principle behind MS-EKF is to decouple the observable states from the non-observable range. This is accomplished by representing the state vector in modified spherical coordinates according to [34]

$$\begin{aligned} x^{\mathrm{msc}} &= \begin{bmatrix} \frac{1}{r} & \eta & \epsilon & \frac{\dot{r}}{r} & \dot{\eta}\cos\epsilon & \dot{\epsilon} \end{bmatrix}^T, \\ r &= \sqrt{s_X^2 + s_Y^2 + s_Z^2}. \end{aligned} \tag{3.7}$$

From the motion equations for x^{msc} it is clear that the first state representing the inverse of range is decoupled from the other states during periods with constant velocity [34]. The principle of RP-EKF is to apply a Gaussian sum filter as demonstrated by Example 2.1. For examples on typical tracking filter performance when using an angle-only sensor see e.g. [56, 34, 72]. Note that it is highly important to accurately estimate not only the mean but the covariance matrix, denoted by P, as well. This is due to the fact that when we compute probability of Near Mid-Air Collision (NMAC) based on tracking data we need the probability density function (pdf) for the state vector. In case of a Gaussian distributed state vector the mean and covariance give a complete description of the pdf. A justified question is whether the output from the tracking filter is normally distributed or not. The results using MS-EKF given by [72] indicate that the output from the tracking filter is indeed normally distributed. Due to the low observability along line-of-sight, angle-only tracking typically results in large covariances for the variables along line-of-sight (range and closing speed).

The natural extension of the standard EKF to deal with a maneuvering target, where the maneuvering is well modeled by a discrete parameter, is to apply the IMM filter [16]. Note however it is not clear that the IMM outperforms EKF merely based on whether the target maneuvers or not [59]. Another important issue to deal with is data or measurement association in case of multiple-target tracking [14].

3.1.3 Near Mid-Air Collision Avoidance

There are numerous methods which provide collision avoidance, see for example [69] for a survey of different methods for conflict detection and resolution. Note that many of the methods investigated in [69] are primarily for separation assurance and not NMAC avoidance, although the basic principle is the same. The main difference is with respect to time, separation conflict is usually detected and resolved at a much earlier stage compared to a last resort collision avoidance maneuver.

We adopt a probabilistic framework, since the uncertainties are of the same magnitude or larger compared to the safety zone. The uncertainties come from the tracking filter estimating the intruder's position and velocity. Given an estimated state vector and

covariance from the tracking filter we must avoid NMACs with a predefined probability. At the same time we must not initiate an avoidance maneuver if not really necessary, i.e. the probability for nuisance maneuvers shall not exceed a predefined level. A nuisance maneuver causes an unnecessary interruption to the mission of the vehicle which is undesirable.

We are interested in the probability of NMAC for a predicted trajectory. As long as there is at least one trajectory which complies with an acceptable maximum risk nothing happens, i.e. the flight continues along its predetermined trajectory. If the probability for a particular trajectory exceeds the acceptable level, we need to find another trajectory which has a probability less than the acceptable level, compare with Figure 3.3. If the trajectory

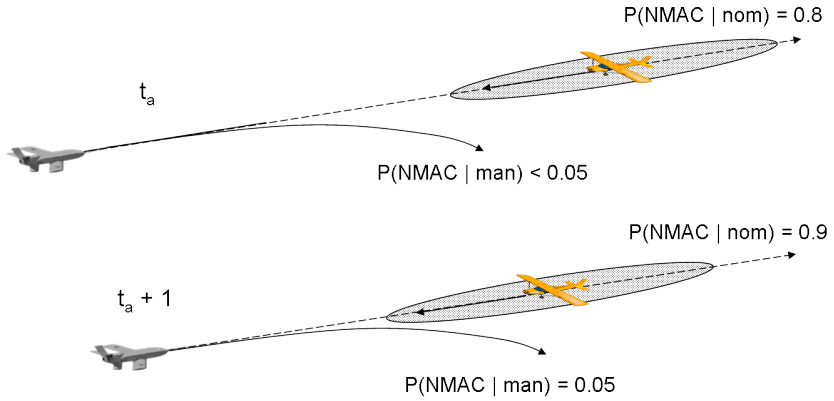

Figure 3.3: *An illustration of an encounter with the ellipse representing the uncertainty of intruder position. Here two trajectories, corresponding to a maneuver (man) and no maneuver (nom), are checked. At time t_a the maneuver trajectory results in an acceptable $\mathrm{P}(\mathrm{NMAC}|\mathrm{man})$, here less than 0.05, and the flight continues along nom. At time $t_a + 1$ the maneuver trajectory yields $\mathrm{P}(\mathrm{NMAC}|\mathrm{man}) = 0.05$. If no other maneuver yields a lower probability man is initiated.*

corresponding to a last resort avoidance maneuver exceeds the level of acceptable risk at the subsequent time instant the evasive maneuver is executed immediately. The main reason for executing an avoidance maneuver as late as possible is to minimize the amount of nuisance maneuvers [103]. The tracking filter accuracy increases over time since the estimate is based on more measurements. The more certain estimate of intruder position and velocity we have access to the better we can judge whether an avoidance maneuver is really necessary or not.

For an accurate estimate of probability of NMAC we must be able to accurately predict the relative trajectory. The predicted trajectory must also be feasible, i.e. the trajectory must be such that it corresponds to a maneuver which the vehicle can actually perform. This means that the acceleration in (3.6) is limited, where the actual limit dependence on vehicle platform and power.

3.2 Probability of Near Mid-Air Collision

Denote the predicted relative position with $s(t)$, where $t = 0$ represents the present time. For a predicted trajectory the probability of NMAC is

$$\mathrm{P}(\mathrm{NMAC}_{t>0}) = \mathrm{P}\big(\min_{t>0}|s(t)|<\mathrm{R}\big), \tag{3.8}$$

where $|s(t)|$ corresponds to the distance between the vehicles and R is the safety zone radius. Here we assume the safety zone is given by a sphere. This is not necessary, but it makes the notation simpler. The probability (3.8) is a prediction from current time into the future. This means that (3.8) is actually a function of current time, denoted with t_a (a as in absolute time). Usually we omit the dependence on t_a for notational convenience.

Closest point of approach (CPA) is the point where the distance between two vehicles is at its minimum. For an illustration see Example 3.1.

Example 3.1

For a straight trajectory CPA occurs when the vectors representing relative position and relative velocity respectively are perpedendicular. Time to CPA, t_cpa, is for a constant velocity defined by the equation

$$v^T s(t_\mathrm{cpa}) = v^T(s(0) + vt_\mathrm{cpa}) = 0. \tag{3.9}$$

Using the closest point of approach we can write the probability of NMAC as

$$\mathrm{P}(\mathrm{NMAC}_{t>0}) = \mathrm{P}\big(|\mathrm{CPA}|<\mathrm{R}\big). \tag{3.10}$$

There are other suggestions on risk measures, where the most common found in literature is the instantaneous probability of NMAC (or conflict with respect to separation) given by [90, 91, 22, 23]

$$\mathrm{P}(\mathrm{NMAC}_t) = \mathrm{P}\big(|s(t)|<\mathrm{R}\big), \tag{3.11}$$

which is then typically maximized over t. A collision detection system based on bearing measurements explicitly is described in [6], without providing any details about parameter settings. In [62] they investigate the performance of a collision avoidance system based on line-of-sight rate, and come to the conclusion that range information is highly beneficial.

The definition of probability of NMAC according to (3.8) and (3.10) yields the probability for a predicted trajectory. We can compute probability of NMAC for a predicted trajectory by sampling N times from estimated intruder state vector and for each sample i compute $\mathrm{CPA}^{(i)}$. Then $\mathrm{P}(\mathrm{NMAC}_{t>0})$ is approximately given by

$$\mathrm{P}(\mathrm{NMAC}_{t>0}) \approx \frac{1}{N}\sum_{i=1}^{N} I\big(|\mathrm{CPA}^{(i)}|<\mathrm{R}\big), \tag{3.12}$$

where $I(\cdot)$ is the indicator function. For an illustration see Figure 3.4.

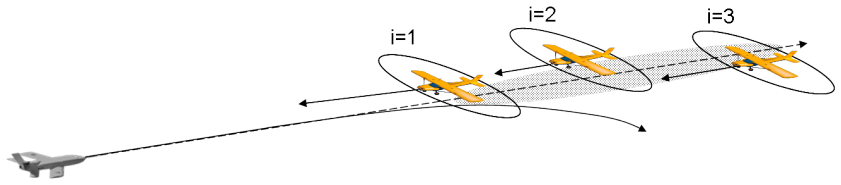

Figure 3.4: *Sample N times (in the picture $N = 3$) from estimated intruder state vector and compute $\{|\mathrm{CPA}^{(i)}|\}_{i=1}^{N}$. Then $\mathrm{P}(\mathrm{NMAC})$ is approximately given by the frequency of $|\mathrm{CPA}^{(i)}| < \mathrm{R}$.*

3.3 Detection using hypothesis testing

Near mid-air collision avoidance can be cast as a hypothesis testing [57, 105, 51], where the hypotheses are

$$\begin{aligned} \mathcal{H}_0 &: \text{no NMAC encounter} \Leftrightarrow \overline{\mathrm{NMAC}}_{t>0} \Leftrightarrow |\mathrm{CPA}| > \mathrm{R}, \\ \mathcal{H}_1 &: \text{NMAC encounter} \Leftrightarrow \mathrm{NMAC}_{t>0} \Leftrightarrow |\mathrm{CPA}| < \mathrm{R}. \end{aligned} \tag{3.13}$$

These hypotheses are either outcomes or assumed known depending on the context. The actions of the NMAC avoidance system is either an avoidance maneuver initiated at any absolute time t_a (= avoidance) or no avoidance maneuver at all (= no avoidance). Since we consider relative position and velocity to be stochastic we do not know, in advance, what the outcome of the encounter will be exactly. We need to resort to the probability of the outcome. The characterization of the performance of the NMAC avoidance system is possible through [109]

$$\text{avoidance unsuccessful} \Leftrightarrow \mathcal{H}_1 \,\big|\, \text{avoidance}, \tag{3.14a}$$

$$\text{avoidance necessary} \Leftrightarrow \mathcal{H}_1 \,\big|\, \text{no avoidance}. \tag{3.14b}$$

The events in (3.14) means that the encounter results in a NMAC given an avoidance maneuver or no avoidance maneuver respectively. It is desirable to have a system which achieves a low probability of unsuccessful avoidance when the necessity of avoidance is high. On the other hand, when there is no need for avoidance, i.e. probability of avoidance necessary is low, then the system should not initate any avoidance maneuver.

Remark 3.1. Compare with standard notation in detection theory, i.e. missed detection and false alarm, which here corresponds to

$$\text{missed detection} \Leftrightarrow \text{no avoidance} \,\big|\, \mathcal{H}_1, \tag{3.15a}$$

$$\text{false alarm} \Leftrightarrow \text{avoidance} \,\big|\, \mathcal{H}_0. \tag{3.15b}$$

The second event (3.15b) is also referred to as a nuisance maneuver. The main problem with (3.15) is that it does not provide any information regarding the success of any applied avoidance maneuver.

The principle of computing the probability of NMAC for a predicted trajectory fits nicely in a setting where there is a requirement on maximum acceptable probability of

NMAC. Assume the requirement is that the collision avoidance system must have a certain capability of avoiding NMACs, i.e.

$$\mathrm{P}(\text{avoidance unsuccessful}) = \mathrm{P}_{\mathrm{req}}. \tag{3.16}$$

Assume that we, at a certain absolute time t_a, have a computed probability of NMAC that is $\mathrm{P}_{\mathrm{req}}$. The computed value is based on the relative state probability density function (pdf) and a predicted relative trajectory. Consider now that we simulate a large number of encounter simulations at the same time t_a using (3.12). The initial conditions are given by the same relative state pdf and the predicted trajectory is the same as well. The outcome, i.e. the frequency of NMACs, will tend to $\mathrm{P}_{\mathrm{req}}$ as the number of simulations tend to infinity. This is quite intuitive, and is also confirmed in [72]. To simplify further analysis and be able to focus on the risk assessment computations we need to make three assumptions

- The video camera has detected the intruder at a minimum distance,
- The tracking filter estimate of relative state is sufficiently accurate,
- The vehicle has a minimum maneuverability.

The extent of these assumptions (minimum distance, sufficiently accurate and minimum maneuverability) must be such that $\mathrm{P}_{\mathrm{req}}$ is larger than computed $\mathrm{P}(\mathcal{H}_1 \mid \text{avoidance})$ at the point in time t_a when the first risk assessment computation is performed. Then for successive computations of $\mathrm{P}(\mathcal{H}_1 \mid \text{avoidance})$, if it at any point reaches $\mathrm{P}_{\mathrm{req}}$ we initiate the avoidance maneuver, compare with Figure 3.3. The outcome of encounter simulations, the frequency of unsuccessful avoidance, will then tend to

$$\mathrm{P}(\text{avoidance unsuccessful}) = \mathrm{P}_{\mathrm{req}}. \tag{3.17}$$

The fact that we in practice compute $\mathrm{P}(\mathcal{H}_1 \mid \text{avoidance})$ at discrete points in time t_{a} means that the result will never be identical to $\mathrm{P}_{\mathrm{req}}$. Note that (3.17) does not make any distinction between whether the encounter is NMAC or non-NMAC for the original, no avoidance trajectory. The case with a non-NMAC encounter resulting in avoidance unsuccessful is often referred to as an *induced* NMAC. We also know that $\mathrm{P}_{\mathrm{req}}$ will not be exceeded if no avoidance maneuver is executed, i.e. the outcome of encounter simulations will tend to

$$\mathrm{P}(\text{avoidance necessary}) < \mathrm{P}_{\mathrm{req}}, \tag{3.18}$$

This result is explained by the fact that the probability of avoidance unsuccessful and avoidance necessary will approach each other and eventually coincide at or near closest point of approach, see Example 3.2. Any avoidance maneuver becomes less effective the closer the relative position is to CPA. Since an avoidance maneuver is never initiated and the probabilities coincide the relation follows.

Compare the result above with instantaneous probability, or any other method for that matter, where we need to find thresholds P_{th} which yield as a result the required probability because $\mathrm{P}_{\mathrm{th}} \neq \mathrm{P}_{\mathrm{req}}$. The encounter simulations would now have to include

the verification of the thresholds. Moreover, the thresholds are in general neither easy to find or constant with respect to e.g. tracking performance [72].

Using the probability of NMAC computed for a predicted trajectory the encounter simulations can focus on verifying sensor and tracking filter performance. For example we need camera probability of detection and tracking filter convergence, both as functions of distance. The result must be such that there is enough distance left for at least one avoidance maneuver, with a collision risk below the requirement, to be available. Moreover, camera probability of false alarm and tracking filter performance must be such that maximum level of false alarm rate is not exceeded. For example, the more uncertain the tracking filter is the more nuisance avoidance maneuvers will be executed because the probability of NMAC will be larger for non-NMAC encounters, i.e.

$$\mathrm{P}(\text{avoidance} \mid \mathcal{H}_0) = \mathrm{P}(\text{nuisance}) \tag{3.19}$$

will be larger for a worse tracking performance. As noted earlier this is the argument for waiting as long as possible before initiating an avoidance maneuver. The longer we can wait the more time will the sensor and tracking filter have to detect and converge respectively. To obtain the probability of nuisance maneuvers we typically have to resort to simulations. For an illustration of probability of avoidance unsuccessful and avoidance necessary see Example 3.2.

Example 3.2

Consider two near head-on collision scenarios in two dimensions, see Figure 3.5, with

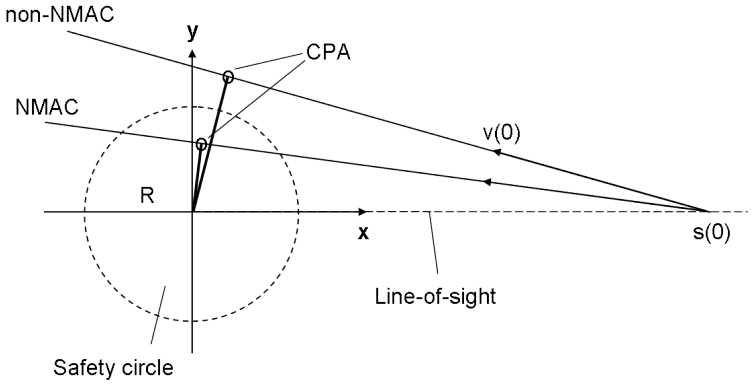

Figure 3.5: *A NMAC and a non-NMAC encounter in a relative coordinate system with the $x-$axis pointing along line-of-sight. The ratio of mean and standard deviation is set to five for distance and closing speed. Standard deviation for distance and speed orthogonal to line-of-sight are both two. Initial distance $|s(0)|$ is 1500 meters and speed $|v(0)|$ is 100 m/s.*

$|\mathrm{CPA}| = 100$ and 200 meters respectively. The two scenarios correspond to a NMAC and non-NMAC encounter respectively. Probability of NMAC with and without avoidance maneuver is shown in Figure 3.6 as a function of absolute time t_{a}. The avoidance maneuver is simplified such that an instantaneous turn of 60 degrees is performed at time

$t_a + 3$. Note that the two curves eventually coincide which is quite intuitive considering the effect of an avoidance maneuver is gradually reduced and eventually it has no effect at all at or near closest point of approach. Here we assume $\mathrm{P}_{\mathrm{req}} = 0.05$, which means that $\mathrm{P}(\mathcal{H}_1 \mid \text{avoidance})$ is 0.05. In the left plot we conclude that the $\mathrm{P}(\mathcal{H}_1 \mid \text{no avoidance})$ is approximately $0.98-0.99$, which is obtained from the gap between zero and the no avoidance trajectory curve at the time the avoidance trajectory curve crosses the $\mathrm{P}_{\mathrm{req}}-$level. In the right plot the avoidance trajectory curve never exceeds 0.05 and therefore no avoidance maneuver will be executed. This is desirable because probability of avoidance necessary, represented by the no avoidance trajectory curve, tends to a small value below the $\mathrm{P}_{\mathrm{req}}-$level. If a smaller $\mathrm{P}_{\mathrm{req}}$ is used then probability of avoidance unsuccessful will be smaller. However, we would also obtain a larger probability of avoidance unnecessary if the lower $\mathrm{P}_{\mathrm{req}}-$level cuts the avoidance trajectory curve in the non-NMAC scenario. For more information and several examples on the performance of collision avoidance systems see [68, 109].

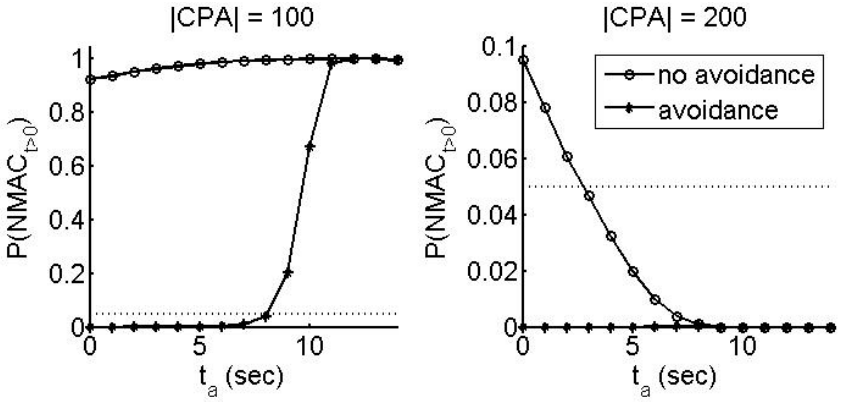

Figure 3.6: *Probability of NMAC as a function of absolute time, no maneuver and avoidance maneuver (instantaneous after 3 sec) respectively. Left plot shows the NMAC encounter and the right the non-NMAC encounter from Figure 3.5.*

3.4 Probability of level-crossings

The problem with the Monte-Carlo method (3.12) is that it requires a too large number of samples, see Paper C. We need to find another method for computing probability of NMAC which is computationally tractable for real-time processing. A way forward is to look upon NMAC as a level-crossing, or down-crossing, of the surface of the safety sphere, see Figure 3.7. Here we use a coordinate system which is rotated such that

$$\hat{s}_y(0) = \hat{s}_z(0) = 0, \tag{3.20}$$

i.e. the $x-$axis is aligned with line-of-sight.

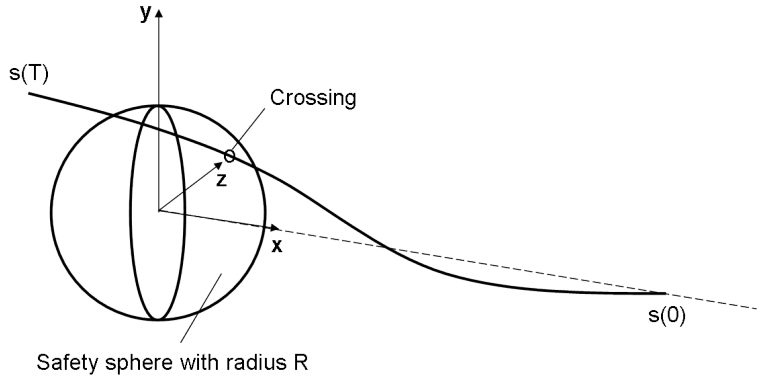

Figure 3.7: *A NMAC occurs if the relative position crosses the surface of the sphere.*

We assume the predicted relative trajectory is deterministic conditionally upon initial intruder state. That is, given the initial state the future trajectory of the intruder is known. Typically we assume the intruder follows a straight path. Here we consider short periods of time, say prediction times up to a minute. For disturbances along the path which are not excessive the assumption on a known future relative trajectory given the initial state should not be too far from the truth. For the own vehicle we have the possibility to use more or less complex motion models for prediction due to known dynamics. It is probably useless though to use too complex models since the true trajectory will deviate more or less from the predicted. The result on short-range conflict detection given in [91] can be used to determine the effect of disturbances and modelling errors. To be able to compute probability of NMAC for a predicted trajectory we will assume that the relative position follows a piecewise straight path. In practice this is obviously not correct, but it is possible to make the error arbitrarily small by making the straight segments short enough.

For computational tractability we need to limit the predicition time t to a finite T, i.e. we consider the event $\text{NMAC}_{(0,T)}$ instead of $\text{NMAC}_{t>0}$. This is no restriction as long as T is larger than the time it takes to perform an avoidance maneuver and some additional time for the effect of the maneuver to become evident in the probability.

3.4.1 One dimension and constant velocity

In one dimension the process describing relative position $s_x(t)$ is confined to lie on a straight line. Assume we wish to compute the probability

$$\mathrm{P}\big(\min_{0<t<T} s_x(t) < 0\big). \tag{3.21}$$

Recall that t represents prediction time, and that (3.21) is also a function of absolute time t_a. This probability can be split using two mutually exclusive events, $s_x(0) < 0$ and $s_x(0) > 0$. At the same time we note that if $s_x(0) < 0$ we automatically have that the

minimum of the process is less than zero, i.e.

$$\begin{aligned}&\mathrm{P}\big(\min_{0<t<T} s_x(t) < 0\big)\\ &= \mathrm{P}\big(\min_{0<t<T} s_x(t) < 0 \cap s_x(0) > 0\big) + \mathrm{P}\big(s_x(0) < 0\big).\end{aligned} \tag{3.22}$$

The first probability in the sum in (3.22) can be seen as a level-crossings problem. The position $s_x(t)$ must cross the level zero since it starts with $s_x(0) > 0$, i.e. a down-crossing. Let us start by considering a constant velocity

$$\dot{s}_x(t) = v_x(t), \quad 0 < t < T, \tag{3.23a}$$
$$v_x(t) = v_x(0), \tag{3.23b}$$

which means that we can reformulate the down-crossings probability according to

$$\mathrm{P}\big(\min_{0<t<T} s_x(0) + v_x(0)t < 0 \cap s_x(0) > 0\big). \tag{3.24}$$

If a down-crossing is to occur the relative velocity must be negative, otherwise $s_x(t)$ will only increase. Moreover, the minimum of $s_x(0) + v_x(0)t$ with a negative velocity and $0 < t < T$ is attained at $t = T$. We can therefore write the probability as

$$\mathrm{P}\big(s_x(0) + v_x(0)T < 0 \cap s_x(0) > 0 \cap v_x(0) < 0\big). \tag{3.25}$$

This event is depicted in Figure 3.8. The ellipse represents the joint probability density function (pdf) for $s_x(0)$ and $v_x(0)$ and is denoted by $p_{s_x(0),v_x(0)}(s, v)$. We see from

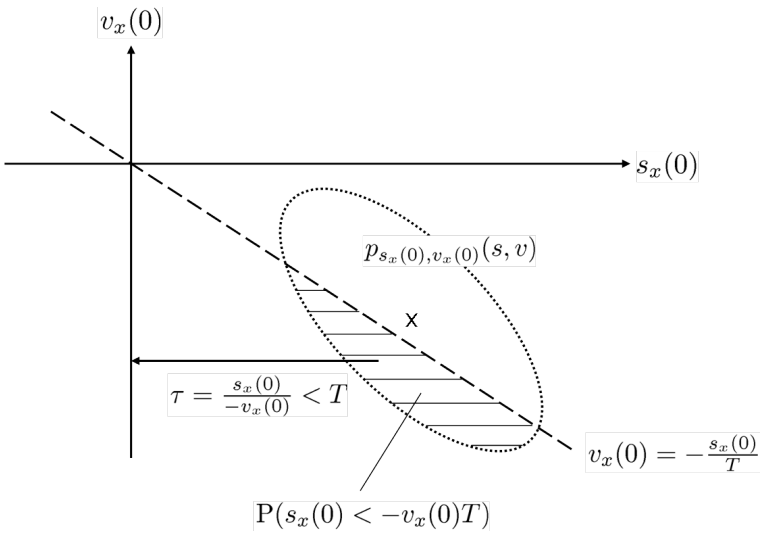

Figure 3.8: *The probability density for $s_x(0)$ and $v_x(0)$ integrated over the area $s_x(0) > 0$ and $v_x(0) < -s_x(0)/T$ yields the probability of $\tau < T$.*

Figure 3.8 that the sought probability is given by the integral of $p_{s_x(0),v_x(0)}(s, v)$ over the

area limited by $s_x(0) > 0$ and $v_x(0) < -s_x(0)/T$. To simplify the expression for the probability we can define a time τ according to

$$\tau = \begin{cases} \frac{s_x}{-v_x} & \text{if } s_x > 0 \cap v_x < 0, \\ \infty & \text{otherwise,} \end{cases} \tag{3.26}$$

which yields the expression

$$\mathrm{P}\big(s_x(0) + v_x(0)T < 0 \cap s_x(0) > 0 \cap v_x(0) < 0\big) = \mathrm{P}(\tau < T). \tag{3.27}$$

The time τ can be interpreted as a time-to-go (ttg), i.e. the time left before $s_x(\tau) = 0$. Based on the distribution for τ it is straightforward to obtain the expression for the probability density function

$$p_\tau(t) = \int_{-\infty}^{0} -v p_{s_x, v_x}(-vt, v) dv, \quad t < \infty. \tag{3.28}$$

See Example 3.3 for the case with Gaussian distributed variables.

Example 3.3: Gaussian distributed $s_x(0)$ and $v_x(0)$

If $s_x(0)$ and $v_x(0)$ are jointly Gaussian distributed,

$$\begin{bmatrix} s_x \\ v_x \end{bmatrix} \sim \mathcal{N}\Bigg(\begin{bmatrix} \hat{s}_x \\ \hat{v}_x \end{bmatrix}, \underbrace{\begin{bmatrix} \sigma_{sx}^2 & \rho_x \sigma_{sx} \sigma_{vx} \\ \rho_x \sigma_{sx} \sigma_{vx} & \sigma_{vx}^2 \end{bmatrix}}_{P_{sx}} \Bigg), \tag{3.29}$$

we have

$$\begin{aligned} &\mathrm{P}(\tau < T) \\ &= \frac{1}{2\pi \sigma_{sx} \sigma_{vx} (1 - \rho_x)^{1/2}} \int_{-\infty}^{0} \int_{0}^{-vT} e^{-\frac{1}{2} \begin{bmatrix} s - \hat{s}_x \\ v - \hat{v}_x \end{bmatrix}^T P_{sx}^{-1} \begin{bmatrix} s - \hat{s}_x \\ v - \hat{v}_x \end{bmatrix}} ds dv \end{aligned} \tag{3.30}$$

The expression for $p_\tau(t)$ is derived in Appendix A.

Note that it is possible to obtain the expression for $\mathrm{P}(\tau < T)$ through the use of Rice's formula [71]. Rice's formula gives a method for computing the intensity of level-crossings for a stochastic process.

3.4.2 One dimension and piecewise constant velocity

The result for a straight path is possible to extend to a piecewise straight path given by

$$\dot{s}(t) = v^{(j)}, \quad T_j < t < T_{j+1}, \tag{3.31a}$$

$$v^{(j)} = v(0) + \sum_{l=1}^{j} \Delta v^{(l)}, \tag{3.31b}$$

where all $\Delta v^{(j)}$ are known. An important observation is that if it is unlikely that $v_x^{(j)}$ changes sign, i.e.

$$\mathrm{P}\big(\mathrm{sign}(v_x^{(j)}) = \mathrm{sign}(v_x^{(0)})\big) = 1, \tag{3.32}$$

or very close to one, then $\mathrm{NMAC}_{(T_j,T_{j+1})}$ for different j:s are mutually exclusive. With this assumption we can write

$$\begin{aligned}\mathrm{P}(\mathrm{NMAC}_{(0,T)}) &= \mathrm{P}(\cup_{j=0}^{J-1}\mathrm{NMAC}_{(T_j,T_{j+1})}) \\ &= \sum_{j=0}^{J-1}\mathrm{P}(\mathrm{NMAC}_{(T_j,T_{j+1})}).\end{aligned} \tag{3.33}$$

Now we can focus the computation on each segment with constant velocity $v^{(j)}$. By considering what the initial conditions at $t = 0$ would be if the position and velocity are $s(T_j)$ and $v^{(j)}$ at $t = T_j$ we can reuse the results from the case with constant velocity. Denote the new relative position at $t = 0$ by $s^{(j)}$, and we have

$$s^{(j)} = s(T_j) - T_j v^{(j)}. \tag{3.34}$$

An advantage using $s^{(j)}$ instead of $s(T_j)$ comes from the fact that the covariance of $s^{(j)}$ is equal to the covariance of $s(0)$ for all $j = 0, \ldots, J-1$. This is clear from the expression

$$s^{(j)} = s(0) - \sum_{l=1}^{j} T_l \Delta v^{(l)}, \tag{3.35}$$

which is derived in Paper C. The probability of down-crossing is now given by

$$\mathrm{P}(\mathrm{NMAC}_{(0,T)}) = \sum_{j=0}^{J-1}\mathrm{P}(T_j < \tau^{(j)} < T_{j+1}), \tag{3.36}$$

where

$$\tau^{(j)} = \begin{cases} \frac{s_x^{(j)}}{-v_x^{(j)}} & \text{if } s_x^{(j)} > 0 \cap v_x^{(j)} < 0, \\ \infty & \text{otherwise,} \end{cases} \tag{3.37}$$

3.4.3 Two dimensions

In two dimensions the probability of NMAC is

$$\mathrm{P}(\mathrm{NMAC}_{(0,T)}) = \mathrm{P}\big(\min_{0<t<T}\sqrt{s_x^2(t) + s_y^2(t)} < \mathrm{R}\big), \tag{3.38}$$

that is the relative position shall not cross the circle with radius R. A reasonable approximation is to consider the crossing of a line perpendicular to line-of-sight, see Figure 3.9. For notational convenience we place the line at $x = 0$, keeping in mind that we have the possibility to change the outcome by changing the location of the line. The

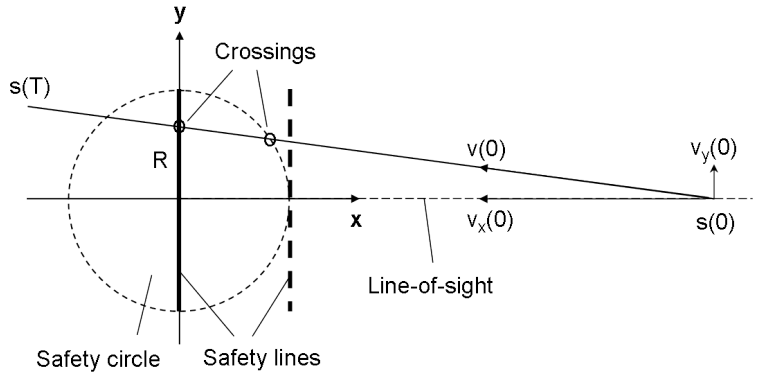

Figure 3.9: *The crossing of the circle is approximated by the crossing of a line.*

only requirement is that the line must be orthogonal to line-of-sight ($x-$axis). The reason is that we can reuse the results from Section 3.4.1. The stochastic time τ provides a measure on time-to-go, i.e. the time left before the relative position crosses the $y-$axis. If the velocity $|v_y(0)|$ is too small there will be a crossing, too small in the sense that $|v_y(0)\tau + s_y(0)| < \mathrm{R}$. This means that we can express the probability of NMAC within T seconds according to

$$\mathrm{P}(\widehat{\mathrm{NMAC}}_{(0,T)}) = \mathrm{P}(|v_y(0)\tau + s_y(0)| < \mathrm{R} \cap \tau < T). \tag{3.39}$$

The event $\widehat{\mathrm{NMAC}}_{(0,T)}$ denotes the crossing of the safety line instead of the safety circle, see Figure 3.9. In the case where τ is independent of $s_y(0)$ and $v_y(0)$ it is possible to express the sought probability according to

$$\mathrm{P}(\widehat{\mathrm{NMAC}}_{(0,T)}) = \int_0^T \mathrm{P}(|v_y(0)t + s_y(0)| < \mathrm{R}) p_\tau(t) dt. \tag{3.40}$$

For an illustration see Example 3.4.

Example 3.4: Gaussian distributed state vector in two dimensions

If $s_y(0)$ and $v_y(0)$ are Gaussian distributed

$$\begin{bmatrix} s_y(0) \\ v_y(0) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \hat{s}_y(0) \\ \hat{v}_y(0) \end{bmatrix}, P_y \right), \tag{3.41}$$

then $v_y(0)t + s_y(0)$ is also Gaussian distributed

$$s_y(0) + v_y(0)t \sim \mathcal{N}\left(\hat{s}_y(0) + \hat{v}_y(0)t, \begin{bmatrix} 1 & t \end{bmatrix} P_y \begin{bmatrix} 1 \\ t \end{bmatrix} \right). \tag{3.42}$$

Moreover, if $s_x(0)$ and $v_x(0)$ are Gaussian distributed as in Example 3.3, then the expression for $p_\tau(t)$ is given by Appendix A. If $s_x(0)$ and $v_x(0)$ are uncorrelated with

$s_y(0)$ and $v_y(0)$ the expression for probability of NMAC according to (3.40) is applicable. Probability of NMAC can then computed by applying Simpson's rule [1] on (3.40).

The extension to piecewise straight paths is equivalent to the one-dimensional case. By using $s^{(j)}$ (3.35) and $v^{(j)}$ (3.31b) we have

$$\mathrm{P}(\widehat{\mathrm{NMAC}}_{(0,T)}) = \sum_{j=0}^{J-1} \mathrm{P}\big(|v_y^{(j)}\tau^{(j)} + s_y^{(j)}| < \mathrm{R} \cap T_j < \tau^{(j)} < T_{j+1}\big). \tag{3.43}$$

As for a straight path we can simplify the expression in (3.43) for the independent case according to

$$\mathrm{P}(\widehat{\mathrm{NMAC}}_{(0,T)}) = \sum_{j=0}^{J-1} \int_{T_j}^{T_{j+1}} \mathrm{P}\big(|v_y^{(j)}t + s_y^{(j)}| < \mathrm{R}\big) p_\tau(t) dt. \tag{3.44}$$

For the computation of (3.39) and (3.43) in the general, dependent case see Paper C.

3.4.4 Three dimensions

In three dimensions the probability of NMAC is

$$\mathrm{P}(\mathrm{NMAC}_{(0,T)}) = \mathrm{P}\big(\min_{0<t<T} \sqrt{s_x^2(t) + s_y^2(t) + s_z^2(t)} < \mathrm{R}\big), \tag{3.45}$$

that is the relative position shall not cross a sphere with radius R. The approximation is to consider the crossing of a circular disc perpendicular to line-of-sight, see Figure 3.10. Similar to the two-dimensional case we can express the approximate probability of

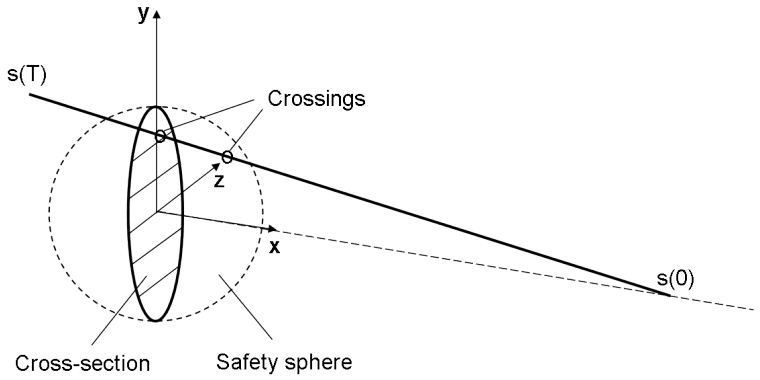

Figure 3.10: *The crossing of the sphere is approximated by the crossing of a disc.*

NMAC within T seconds according to

$$\mathrm{P}(\widehat{\mathrm{NMAC}}_{(0,T)}) = \mathrm{P}(\tau \underbrace{\sqrt{v_y^2(0) + v_z^2(0)}}_{v_\perp} < \mathrm{R} \cap \tau < T). \tag{3.46}$$

For notational brevity we assume here that $s_y(0) \equiv 0$ and $s_z(0) \equiv 0$. The variance of $s_y(0)$ and $s_y(0)$ are given by [8]

$$\begin{aligned} \sigma_{sy} &= \mathrm{var}\big(s_x(\eta - \hat{\eta})\big) = \hat{s}_x^2\sigma_\eta^2 + \sigma_{sx}^2\sigma_\eta^2, \\ \sigma_{sz} &= \mathrm{var}\big(s_x(\epsilon - \hat{\epsilon})\big) = \hat{s}_x^2\sigma_\epsilon^2 + \sigma_{sx}^2\sigma_\epsilon^2. \end{aligned} \tag{3.47}$$

For angle-only tracking with an angle measurement error less than 0.5 mrad and an initial distance less than 4000 meters we can neglect $s_y(0)$ and $s_z(0)$ without introducing significant errors. If the simplification is not appropriate it is however straightforward to include $s_y(0)$ and $s_z(0)$, compare with two dimensions in Section 3.4.3 and Paper C. In the case where τ is independent of $v_y(0)$ and $v_z(0)$ the probability is given by

$$\mathrm{P}(\widehat{\mathrm{NMAC}}_{(0,T)}) = \int_0^T \mathrm{P}(v_\perp t < \mathrm{R}) p_\tau(t) dt. \tag{3.48}$$

The main difference between two and three dimensions is, for a given $\tau = t$, how to compute the probability of $v_\perp$ to be within a circle with radius R/t. If $v_y(0) = v_y$ and $v_z(0) = v_z$ are Gaussian distributed

$$v_y \sim \mathcal{N}(\hat{v}_y, \sigma_y^2), \quad v_z \sim \mathcal{N}(\hat{v}_z, \sigma_z^2) \tag{3.49}$$

we can use the fact that the sum of the squares is possible to express as a weighted sum of two non-central χ^2- variables each with one degree of freedom, denoted by $\chi_1^2(\lambda)$. Here λ is the non-centrality parameter. Consider the simplest case where v_y and v_z are uncorrelated. This is no restriction since the correlated case is handled equivalently [95, 100]. Then the velocities squared and normalized with their respective variances are

$$\begin{aligned} \frac{v_y^2}{\sigma_y^2} &\sim \chi_1^2(\lambda_y), \quad \lambda_y = \frac{\hat{v}_y^2}{\sigma_y^2}, \\ \frac{v_z^2}{\sigma_z^2} &\sim \chi_1^2(\lambda_z), \quad \lambda_z = \frac{\hat{v}_z^2}{\sigma_z^2}. \end{aligned} \tag{3.50}$$

One method to compute the probability of the weighted sum is via the characteristic function, for details see Paper D. Another method is to express the weighted sum as an infinite series of central χ^2- variables [95, 100], denoted by χ_l^2, i.e.

$$\mathrm{P}(v_\perp^2 < \frac{\mathrm{R}^2}{t^2}) = \sum_{l=0}^{\infty} c_l \mathrm{P}\big(\xi_l < \frac{\mathrm{R}^2}{\kappa t^2}\big), \quad \xi_l \sim \chi_{2+2l}^2. \tag{3.51}$$

In practice we need to truncate the sum after a finite number of terms L, and the positive scale parameter κ can be used for better convergence. We know from [95] that

$\kappa = \min(\sigma_y^2, \sigma_z^2)$ provides a fast convergence. Moreover, it guarantees a mixture representation, i.e.

$$\sum_{l=0}^{\infty} c_l = 1, \text{ and } c_l \geq 0, \ \forall l. \tag{3.52}$$

A mixture representation together with the fact that the central χ^2- distribution is a decreasing function with respect to its degree of freedom, $\mathrm{P}\big(\xi_{l+1} < x\big) \leq \mathrm{P}\big(\xi_l < x\big)$, yield an upper bound on the truncation error according to

$$\begin{aligned} &|\hat{\mathrm{P}}_L(v_\perp^2 < \frac{\mathrm{R}^2}{t^2}) - \mathrm{P}(v_\perp^2 < \frac{\mathrm{R}^2}{t^2})| = \\ &\sum_{l=L+1}^{\infty} c_l \mathrm{P}\big(\xi_l < \frac{\mathrm{R}^2}{\kappa t^2}\big) \leq \\ &\mathrm{P}\big(\xi_L < \frac{\mathrm{R}^2}{\kappa t^2}\big)\big(1 - \sum_{l=0}^{L} c_l\big). \end{aligned} \tag{3.53}$$

See Algorithm 3.1 for details on the realization.

Algorithm 3.1 ($\mathrm{P}(v_\perp^2 < \frac{\mathrm{R}^2}{t^2})$ *as an infinite series).*

1. Initialization: Assume $\sigma_z \leq \sigma_y$

$$\begin{aligned} &\kappa = \sigma_z^2 \\ &c_0 = \frac{\kappa}{\sigma_y \sigma_z} e^{-\frac{\lambda_y + \lambda_z}{2}}, \\ &\mathrm{P}\big(\xi_0 < \frac{\mathrm{R}^2}{\kappa t^2}\big) = 1 - e^{-\frac{\mathrm{R}^2}{2\kappa t^2}}, \\ &\hat{\mathrm{P}}_0(v_\perp^2 < \frac{\mathrm{R}^2}{t^2}) = c_0 \mathrm{P}\big(\xi_0 < \frac{\mathrm{R}^2}{\kappa t^2}\big). \end{aligned}$$

2. Repeat until $\mathrm{P}\big(\xi_L < \frac{\mathrm{R}^2}{\kappa t^2}\big)\big(1 - \sum_{l=0}^{L} c_l\big) < \varepsilon$

$$\begin{aligned} &c_l = \frac{1}{l} \sum_{r=1}^{l} g_r c_{l-r}, \\ &g_r = \frac{r\lambda_y}{2}(1 - \frac{\kappa}{\sigma_y^2})^{r-1} + \frac{r\lambda_z}{2}(1 - \frac{\kappa}{\sigma_z^2})^{r-1} + \\ &\frac{1 - r\lambda_y}{2}(1 - \frac{\kappa}{\sigma_y^2})^{r} + \frac{1 - r\lambda_z}{2}(1 - \frac{\kappa}{\sigma_z^2})^{r}, \\ &\mathrm{P}\big(\xi_l < \frac{\mathrm{R}^2}{\kappa t^2}\big) = \mathrm{P}\big(\xi_{l-1} < \frac{\mathrm{R}^2}{\kappa t^2}\big) - \frac{\mathrm{R}^{2l}}{(2\kappa t^2)^l l!} e^{-\frac{\mathrm{R}^2}{2\kappa t^2}}, \\ &\hat{\mathrm{P}}_l(v_\perp^2 < \frac{\mathrm{R}^2}{t^2}) = \hat{\mathrm{P}}_{l-1}(v_\perp^2 < \frac{\mathrm{R}^2}{t^2}) + c_l \mathrm{P}\big(\xi_l < \frac{\mathrm{R}^2}{\kappa t^2}\big). \end{aligned}$$

Remark 3.2. Simulations indicate that

$$\kappa = \frac{2\sigma_y^2\sigma_z^2}{\sigma_y^2 + \sigma_z^2} \tag{3.54}$$

yields a faster convergence than $\kappa = \min(\sigma_y^2, \sigma_z^2)$. To be able to quarantee a mixture representation the conditions in (3.52) must be met. We know from [95] that for $\sum_{l=0}^{\infty} c_l = 1$ a sufficient requirement is that $\kappa < 2\min(\sigma_y^2, \sigma_z^2)$. Moreover, a sufficient condition for $c_l \geq 0, \forall l$ is that $g_r \geq 0, \forall r$. It is possible to show that this is the case, assuming $\sigma_z \leq \sigma_y$, if

$$\lambda_y \geq \lambda_z \frac{\sigma_y^2}{\sigma_z^2} \Leftrightarrow |\hat{v}_y| \geq |\hat{v}_z| \frac{\sigma_y^2}{\sigma_z^2}. \tag{3.55}$$

Remark 3.3. For large $\lambda_y + \lambda_z$ the number of terms L needs to be large for the truncation error to be small. This can also be deduced from Algorithm 3.1 where $c_0 \approx 0$ for large $\lambda_y + \lambda_z$.

Example 3.5 illustrates how to compute the probability in (3.48) for the Gaussian case.

Example 3.5: Gaussian distributed state vector in three dimensions

If $v_y(0)$ and $v_z(0)$ are Gaussian distributed

$$\begin{bmatrix} v_y(0) \\ v_z(0) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \hat{v}_y(0) \\ \hat{v}_z(0) \end{bmatrix}, P_{yz} \right), \tag{3.56}$$

we apply a change of variables according to

$$\begin{bmatrix} v_{y'}(0) \\ v_{z'}(0) \end{bmatrix} = U \begin{bmatrix} v_y(0) \\ v_z(0) \end{bmatrix}, \quad P_{yz} = U \begin{bmatrix} \sigma_{y'}^2 & 0 \\ 0 & \sigma_{z'}^2 \end{bmatrix} U^T, \tag{3.57}$$

where U is a unitary matrix. We can then compute $\mathrm{P}(v_\perp t < \mathrm{R})$ using Algorithm 3.1 with mean and variance for $v_{y'}(0)$ and $v_{z'}(0)$ inserted. If $s_x(0)$ and $v_x(0)$ are Gaussian distributed as well the expression for $p_\tau(t)$ is given by Appendix A. The probability $\mathrm{P}(\widehat{\mathrm{NMAC}}_{(0,T)})$ can then computed by applying Simpson's rule [1] on (3.40), assuming τ is independent of $v_\perp$. Note that when applying Simpson's formula we need to compute $\mathrm{P}(v_\perp t_k < \mathrm{R})$ for a number K of discretized time instants t_k. This means that we need to keep both K and L small for a low total computational load.

The extension to piecewise straight paths is equivalent to the one-dimensional case. By using $s^{(j)}$ (3.35) and $v^{(j)}$ (3.31b) we have

$$\begin{aligned} &\mathrm{P}(\widehat{\mathrm{NMAC}}_{(0,T)}) \\ &= \sum_{j=0}^{J-1} \mathrm{P}\big(\sqrt{(v_y^{(j)}\tau^{(j)} + s_y^{(j)})^2 + (v_z^{(j)}\tau^{(j)} + s_z^{(j)})^2} < \mathrm{R} \cap T_j < \tau^{(j)} < T_{j+1} \big). \end{aligned} \tag{3.58}$$

The computation of (3.46) and (3.58) are given by Paper D.

4

Concluding Remarks

In this section we provide a summary of the results of this thesis and discuss possible future research.

4.1 Summary

In this thesis we have provided efficient methods for estimation and detection in general, and for terrain-aided navigation and collision avoidance in particular.

The marginalized particle filter, presented in Paper A, utilizes linear substructure for efficient estimation. The part of the state vector which is confined to a linear model, conditionally upon the nonlinear part, is estimated using the Kalman filter. The remaining nonlinear part with lower dimension is estimated by the particle filter. The filter is applied to a simplified terrain-aided navigation in which we assume altitude is known and excellent performance is achieved. The idea of marginalization is extended in Paper B, where we derive a filter which is optimized for the structure inherent in terrain-aided navigation. Here we split the linear substructure in two parts, one unimodal part estimated by the Kalman filter and one multimodal part estimated by the GPB1 filter. The multimodal part corresponds to altitude, which has a separable posterior pdf conditionally upon the nonlinear states. Application of the filter on both simulated and authentic data show excellent performance.

In Paper C we derive the basic theory for efficient conflict detection in two dimensions. The efficiency is obtained by using the concept of level-crossings to describe conflicts. Conflict is declared if the relative position at any point in time crosses a line segment. This makes the computation of probability of conflict computationally tractable as opposed to a sampling based method. The formulation based on level-crossings makes it possible to apply marginalization. For example in the two-dimensional case with independence between variables along and cross line-of-sight we can compute three out of four integrals analytically. The fourth integral is computed numerically using Simpson's formula. The

theory is then extended for near mid-air collision avoidance in three dimensions in Paper D. Here we approximate the crossing of a safety sphere with the crossing of a disc in order to make the idea of level-crossings applicable. Application of the method on simulated data yields promising results.

4.2 Future Research

The marginalized particle filter is, when applicable, a very effective method to decrease the computational load compared to the standard particle filter. There exist theorems which provide measures on what is gained in terms of sampling variance when the marginalized particle filter is applied instead of the particle filter. However, these theorems assume independent samples. This means that resampling is not covered. Intuition and simulations indicate that the marginalized particle filter is even more effective when resampling is introduced. More studies should be conducted to clarify the theory behind this effect and also if any new insights could be used to further increase the effectiveness.

Here we have derived computationally tractable methods for computing the probability of conflict for a predicted trajectory. The method is applicable to three dimensions, uncertain initial conditions and piecewise straight paths. The main restriction concerns potential disturbances and uncertainties along the predicted relative flight trajectory. Typical disturbances are unknown intentions of the intruder and that the vehicles do not follow the assumed path exactly due to wind and simplified aircraft dynamics. The obvious extension is to include the effect of disturbances by combining the results here and the ones given in [91] for short term conflict detection.

The assumption on a normally distributed estimate of the state vector from the tracking filter is usually quite good [72]. This is true at least for the application considered here with short initial distances, around 4000 meters, and high measurement update rate, around 10 Hz. Probably there are cases where the assumption on normality fails. The derived methods for computing probability of conflict are generic in the sense that they do not require the input to be Gaussian distributed. They do however need the probability density function (pdf) for the relative state vector and for the non-Gaussian case this can be difficult to capture. An interesting question is if it is possible to obtain an upper bound on the probability of conflict based on estimated moments, typically mean and covariance, of the state vector instead of the pdf. A potential solution could be convex optimization techniques applied to Chebyshev bounds [17].

The expressions for probability of conflict are rather complicated, especially in the case with dependence between the variables along and cross line-of-sight. It is also difficult to obtain useful bounds on the approximation error, particularly for the geometric approximation but in the general case also for the numerical approximation. The Gram-Charlier expansion, and the similar Edgeworth expansion, constitute methods for expanding a distribution in terms of the normal distribution and its derivatives [54]. It would be interesting to investigate the applicability of these expansions on the probability of conflict and to see if expressions are attainable with bounds on the approximation errors.

A

Derivation of $p_\tau(t)$

Derivation of the expression for $p_\tau(t)$ in the case with normally distributed s_x and v_x.

$$\begin{aligned} p_\tau(t) &= \int_{-\infty}^{0} -v p_{s_x,v_x}(-vt, v)dv \\ &= \frac{1}{2\pi(\det P_x)^{1/2}} \int_{-\infty}^{0} -ve^{-\frac{1}{2}\begin{bmatrix} -vt-\hat{s}_x \\ v-\hat{v}_x \end{bmatrix}^T P_x^{-1} \begin{bmatrix} -vt-\hat{s}_x \\ v-\hat{v}_x \end{bmatrix}} dv. \end{aligned} \tag{A.1}$$

Diagonalize P_x

$$\begin{aligned} P_x &= \begin{bmatrix} 1 & \kappa \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_{sx}^2(1-\rho_x^2) & 0 \\ 0 & \sigma_{vx}^2 \end{bmatrix} \begin{bmatrix} 1 & \kappa \\ 0 & 1 \end{bmatrix}^T, \\ P_x^{-1} &= \begin{bmatrix} 1 & -\kappa \\ 0 & 1 \end{bmatrix}^T \begin{bmatrix} \frac{1}{\sigma_{sx}^2(1-\rho_x^2)} & 0 \\ 0 & \frac{1}{\sigma_{vx}^2} \end{bmatrix} \begin{bmatrix} 1 & -\kappa \\ 0 & 1 \end{bmatrix}, \end{aligned} \tag{A.2}$$

where $\kappa = \rho_x\sigma_{sx}/\sigma_{vx}$. Insert (A.2) into the exponent in (A.1) and write as a polynomial of v

$$\begin{aligned} &\begin{bmatrix} -vt-\hat{s}_x-\kappa(v-\hat{v}_x) \\ v-\hat{v}_x \end{bmatrix}^T \begin{bmatrix} \frac{1}{\sigma_{sx}^2(1-\rho_x^2)} & 0 \\ 0 & \frac{1}{\sigma_{vx}^2} \end{bmatrix} \begin{bmatrix} -vt-\hat{s}_x-\kappa(v-\hat{v}_x) \\ v-\hat{v}_x \end{bmatrix} \\ &= \frac{(vt+\hat{s}_x+\kappa(v-\hat{v}_x))^2}{\sigma_{sx}^2(1-\rho_x^2)} + \frac{(v-\hat{v}_x)^2}{\sigma_{vx}^2} \\ &= \left(\frac{(t+\kappa)^2}{\sigma_{sx}^2(1-\rho_x^2)} + \frac{1}{\sigma_{vx}^2}\right)v^2 - 2\left(\frac{(t+\kappa)(\kappa\hat{v}_x-\hat{s}_x)}{\sigma_{sx}^2(1-\rho_x^2)} + \frac{\hat{v}_x}{\sigma_{vx}^2}\right)v \\ &+ \left(\frac{(\kappa\hat{v}_x-\hat{s}_x)^2}{\sigma_{sx}^2(1-\rho_x^2)} + \frac{\hat{v}_x^2}{\sigma_{vx}^2}\right) = g_2^2v^2 - 2g_1v + g_0'. \end{aligned} \tag{A.3}$$

Complete the square for v

$$g_2^2 v^2 - 2g_1 v + g_0' = (g_2 v - \frac{g_1}{g_2})^2 - \frac{g_1^2}{g_2^2} + g_0', \tag{A.4}$$

and insert into (A.1)

$$\begin{aligned} p_\tau(t) &= \frac{1}{2\pi(\det P_x)^{1/2}} \int_{-\infty}^{0} -v e^{-\frac{1}{2}\left((g_2 v - \frac{g_1}{g_2})^2 - \frac{g_1^2}{g_2^2} + g_0'\right)} dv \\ &= g_0 e^{\frac{g_1^2}{2g_2^2}} \int_{-\infty}^{0} -v e^{-\frac{(g_2 v - \frac{g_1}{g_2})^2}{2}} dv. \end{aligned} \tag{A.5}$$

Change variable to $x = g_2 v - g_1/g_2$

$$\begin{aligned} p_\tau(t) &= \frac{g_0}{g_2^2} e^{\frac{g_1^2}{2g_2^2}} \int_{-\infty}^{-\frac{g_1}{g_2}} -\big(x + \frac{g_1}{g_2}\big) e^{-\frac{x^2}{2}} dv \\ &= \frac{g_0}{g_2^2} e^{\frac{g_1^2}{2g_2^2}} \left(e^{-\frac{g_1^2}{2g_2^2}} - \sqrt{2\pi} \frac{g_1}{g_2} \Phi\big(-\frac{g_1}{g_2}\big) \right). \end{aligned} \tag{A.6}$$

Bibliography

[1] M. Abramowitz and I.A. Stegun, editors. *Handbook of Mathematical Functions With Formulas, Graphs and Mathematical Tables*. US Department of Commerce, 10 edition, 1964.

[2] D. L. Alspach and H. W. Sorenson. Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Transactions on Automatic Control*, 17(4):439–448, Aug 1972.

[3] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice Hall, Englewood Cliffs, N.J. USA, 1979.

[4] C. Andrieu and A. Doucet. Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 64(4), 2002.

[5] C. Andrieu and A. Doucet. Optimal estimation of amplitude and phase modulated signals. Technical Report CUED/F-INFENG/TR395, Department of Engineering, University of Cambridge CB2 1PZ Cambridge, 2000.

[6] P. Angelov, C. Bocaniala, C.D.and Xideas, C. Patchett, D. Ansell, M. Everett, and G. Leng. A passive approach to autonomous collision detection and avoidance in uninhabited aerial systems. In *IEEE Proceedings of the Tenth International Conference on Computer Modeling and Simulation*, 2008.

[7] T Arino, K. Carpenter, S. Chabert, H. Hutchinson, T. Miquel, B. Raynaud, K. Rigotti, and E. Vallauri. ACAS analysis programme ACASA project work package 1 final report on studies on the safety of ACAS in Europe. Technical Report Version 1.3, EEC Brétigny, March 2002.

[8] L.A. Aroian. The probability function of the product of two normally distributed variables. *The Annals of Mathematical Statistics*, 18(2), 1947.

[9] S. Arulampalam and B. Ristic. Comparison of the particle filter with range-parameterised and modified polar ekfs for angle-only tracking. In *Proceedings of SPIE, Signal and Data Processing of Small Target*, 2000.

[10] *F2411-07 Standard Specification for Design and Performance of an Airborne Sense-and-Avoid System*. ASTM, 2007.

[11] B. Azimi-Sadjadi and P. S. Krishnaprasad. Approximate nonlinear filtering and its application in navigation. *Automatica*, 41(6), June 2005.

[12] Y. Bar-Shalom and X-R. Li. *Estimation and Tracking: Principles, Techniques and Software*. Artech House, Boston, London, 1993.

[13] N. Bergman. *Recursive Bayesian Estimation, Navigation and Tracking Applications*. PhD thesis 579, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 1999.

[14] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[15] J.H. Blakey. Navigating towards the future: transitioning from terrestrial radio navigation to satellite navigation and airborne surveillance. In *Proceedings of the 24th Digital Avionics Systems Conference*, volume 2, Nov 2005.

[16] H. A. P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, Aug 1988.

[17] S.P. Boyd and L. Vandenberghe. *Convex Optimization*, volume 1. Cambridge University Press, 2004.

[18] R. G. Brown and P. Y. C. Hwang. *Introduction to random signals and applied Kalman filtering*. John Wiley & Sons, 3rd edition, 1992.

[19] J.L. Campbell. *Application of airborne laser scanner - aerial navigation*. PhD thesis, Dept. of Electrical Engineering, Ohio University, 2006.

[20] J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for nonlinear problems. In *IEE Proceedings in Radar, Sonar and Navigation*, volume 146, 1999.

[21] H. Carvalho, P. del Moral, A. Monin, and G. Salut. Optimal nonlinear filtering in GPS/INS integration. *IEEE Transactions on Aerospace and Electronic Systems*, 33 (3):835–849, July 1997.

[22] K. Chan. Analytical expressions for computing spacecraft collision probabilities. In *Proceedings of the 11th Annual AAS/AIAA Space Flight/Mechanics Meeting*, 2001.

[23] K. Chan. Improved analytical expressions for computing spacecraft collision probabilities. *Advances in the Astronautical Sciences*, 114, Part II: Spaceflight Mechanics, 2003.

[24] R. Chen and J. S. Liu. Mixture Kalman filters. *Journal of the Royal Statistical Society*, 62(3):493–508, 2000.

[25] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50(3), March 2002.

[26] F. Daum and J. Huang. Curse of dimensionality and particle filters. In *Proceedings of the IEEE Aerospace Conference*, 2003.

[27] A. Doucet and C. Andrieu. Particle filtering for partially observed Gaussian state space models. Technical Report CUED/F-INFENG/TR393, Department of Engineering, University of Cambridge CB2 1PZ Cambridge, Sep 2000.

[28] A. Doucet, N. J. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump Markov linear systems. Technical Report CUED/F-INFENG/TR359, Department of Engineering, University of Cambridge CB2 1PZ Cambridge, 1999.

[29] A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. *Uncertainty in Artificial Intelligence*, 2000.

[30] A. Doucet, S. J. Godsill, and C. Andrieu. On sequential simulation-based methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[31] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo methods in practice*. Springer-Verlag, 2001.

[32] A. Doucet, N. J. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 49(3), March 2002.

[33] A.C. Drumm, J.W. Andrews, T.D. Hall, V.M. Heinz, J.K. Kuchar, S.D. Thompson, and J.D. Welch. Remotely piloted vehicles in civil airspace: Requirements and analysis methods for the traffic alert and collision avoidance system (tcas) and see-and-avoid systems. In *Proceedings of the 23rd Digital Avionics Systems Conference*, volume 2, 2004.

[34] T. Erlandsson. Angle-only target tracking. Master's thesis LiTH-ISY-EX-07/3904, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 2007.

[35] *Aeronautical Information Manual - Official Guide to Basic Flight Information and ATC Procedures*. FAA, 2008.

[36] J. Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57(6):1317–1339, November 1989.

[37] W. R. Gilks and C. Berzuini. Following a moving target - Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society*, 63(1):127–146, 2001.

[38] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113, 1993.

[39] P.D. Groves, R.J. Handley, and A.R. Runnalls. Optimising the integration of terrain referenced navigation with INS and GPS. *The Journal of Navigation*, 59(1), 2006.

[40] F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons Inc, 1st edition, 2000.

[41] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, R. Karlsson, and P-J. Nordlund. A framework for particle filtering in positioning, navigation and tracking problems. In *Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing*, pages 34–37, Aug 2001.

[42] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, R. Karlsson, and P-J. Nordlund. Particle filters for positioning, navigation and tracking. *IEEE Transactions on signal processing*, 50(2):425–437, Feb 2002.

[43] F. Gustafsson, T.B. Schön, R. Karlsson, and P-J. Nordlund. State-of-the-art for the marginalized particle filter. In *Proceedings of the IEEE Nonlinear Statistical Signal Processing Workshop*, pages 172–174, Sept 2006.

[44] A. Gut. *An Intermediate Course in Probability*. Springer-Verlag, 1995.

[45] J. E. Handschin. Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. *Automatica*, 6:555–563, 1970.

[46] J. E. Handschin and D. Q. Mayne. Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *International Journal of Control*, 9(5):547–559, 1969.

[47] T. Hektor. Marginalized particle filter for aircraft navigation in 3-D. Master's thesis LiTH-ISY-EX-07/4024, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 2007.

[48] T. Hektor, H. Karlsson, and P-J. Nordlund. A marginalized particle filter approach to an integrated INS/TAP system. In *Proceedings of the IEEE/ION Position, Location and Navigation Symposium*, pages 766–770, May 2008.

[49] G. Hendeby. *Performance and Implementaion Aspects of Nonlinear Filtering*. PhD thesis 1161, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 2008.

[50] X-L. Hu, T.B. Schön, and L. Ljung. A basic convergence result for particle filtering. *IEEE Transactions on Signal Processing*, 56(4), 2008.

[51] J. Jansson and F. Gustafsson. A framework and automotive application of collision avoidance decision making. *Automatica*, 2008.

[52] A. H. Jazwinski. *Stochastic processes and filtering theory*. Academic Press, 1970.

[53] N.L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions*, volume 1. John Wiley & Sons, Inc., second edition, 1994.

[54] N.L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Multivariate Distributions*, volume 1: Models and Applications. John Wiley & Sons, Inc., second edition, 2000.

[55] T. Kailath, A. Sayed, and B. Hassibi. *Linear estimation*. Prentice Hall, 2001.

[56] R. Karlsson. *Particle Filtering for Positioning and Tracking Applications*. PhD thesis 924, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 2005.

[57] S.M. Kay. *Fundamentals of Statistical Signal Processing: Detection Theory*, volume 2. Prentice Hall Inc, 1998.

[58] M. Kayton and W. R. Fried. *Avionics navigation systems*. John Wiley & Sons, 2nd edition, 1997.

[59] T. Kirubarajan and Y. Bar-Shalom. Kalman filter versus imm estimator: When do we need the latter? *IEEE Transactions on Aerospace and Electronic Systems*, 39 (4), 2003.

[60] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[61] M. J. Kochenderfer, L.P. Espindle, J. D. Griffith, and J. K. Kuchar. Encounter modeling for sense and avoid development. In *Proceedings of the Integrated Communications, Navigation and Surveillance Conference*, 2008.

[62] M. J. Kochenderfer, J. D. Griffith, and J. K. Kuchar. Hazard alerting using line-of-sight rate. In *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008.

[63] A. Kong, J. S. Liu, and W. H. Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(524):278–288, March 1994.

[64] S. C. Kramer and H. W. Sorenson. Bayesian parameter estimation. *IEEE Transactions on Automatic Control*, 33(2):217–222, Feb 1988.

[65] S. C. Kramer and H. W. Sorenson. Recursive Bayesian estimation using piece-wise constant approximations. *Automatica*, 24(6):789–801, 1988.

[66] J. Kuchar, J. Andrews, A. Drumm, T. Hall, V. Heinz, and S. Thompson. A safety analysis process for the traffic alert and collision avoidance system (tcas) and see-and-avoid systems on remotely piloted vehicles. In *Proceedings of the AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit*, September 2004.

[67] J. K. Kuchar and A. C. Drumm. The traffic alert and collision avoidance system. *Lincoln Laboratory Journal*, 16(2), 2007.

[68] J.K. Kuchar. *A unified methodology for the evaluation of hazard alerting systems.* PhD thesis, Massachusetts Inst. of Technology, Dept. of Aeronautics and Astronautics, Cambridge, MA, 1995.

[69] J.K. Kuchar and L.C. Yang. A review of conflict detection and resolution methods. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):179–189, 2000.

[70] A. R. Lacher, D. R. Maroney, and A. D. Zeitlin. Unmanned aircraft collision avoidance - technology assessment and evaluation methods. Technical Report 07-0095, The MITRE Corporation, 2007.

[71] G. Lindgren. Lectures on stationary stochastic processes. http://www.maths.lth.se/matstat/staff/georg/Publications/lecture.pdf, 2004.

[72] F. Lindsten. Angle-only based collision risk assessment for unmanned aerial vehicles. Master's thesis LiTH-ISY-EX-08/4192, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 2008.

[73] J. Liu and M. West. Combined parameter and state estimation in simulation-based filtering. In *Sequential Monte Carlo methods in Practice*. Addison-Wesley, 2001.

[74] J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statisticial Association*, 93(443), Sep 1998.

[75] J. McCalmont, J. Utt, and M. Deschenes. Detect and avoid technology demonstration. In *Proceedings of the 1st Unmanned Aerospace Vehicles, Systems, Technologies, and Operations Conference and Workshop*, 2002.

[76] J. McCalmont, J. Utt, and M. Deschenes. Sense and avoid, phase I (man-in-the-loop) advanced technology demonstration. In *Proceedings of Infotech@Aerospace*, 2005.

[77] C. Musso, N. Oudjane, and F. Le Gland. Improving regularised particle filters. In *Sequential Monte Carlo methods in Practice*. Addison-Wesley, 2001.

[78] P-J. Nordlund. *Sequential Monte Carlo Filters and Integrated Navigation.* Licentiate thesis 945, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 2002.

[79] P-J. Nordlund. Recursive state estimation of nonlinear systems with applications to integrated navigation. Technical Report LITH-ISY-R-2321, Department of Electrical Engineering, Linköping University, SE-58183 Linköping, Sweden, Nov 2000.

[80] P-J. Nordlund and F. Gustafsson. The probability of near midair collisions using level-crossings. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008.

[81] P-J. Nordlund and F. Gustafsson. Marginalized particle filter for accurate and reliable terrain-aided navigation. *Accepted for publication in IEEE Transactions on Aerospace and Electronic Systems*, 2008. http://www.control.isy.liu.se/research/reports/2008/2870.pdf.

[82] P-J. Nordlund and F. Gustafsson. Probabilistic conflict detection for piecewise straight paths. *Submitted to Automatica*, 2008.
http://www.control.isy.liu.se/research/reports/2008/2871.pdf.

[83] P-J. Nordlund and F. Gustafsson. Probabilistic near mid-air collision avoidance. *Submitted to IEEE Transactions on Aerospace and Electronic Systems*, 2008.
http://www.control.isy.liu.se/research/reports/2008/2872.pdf.

[84] P-J. Nordlund and F. Gustafsson. Sequential Monte Carlo filtering techniques applied to integrated navigation systems. In *Proceedings of the 2001 American Control Conference*, volume 6, pages 4375–4380, 2001.

[85] P-J. Nordlund and F. Gustafsson. Recursive estimation of three-dimensional aircraft position using terrain-aided positioning. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Proceesing*, volume 2, pages 1121–1124, 2002.

[86] P-J. Nordlund, F. Gunnarsson, and F. Gustafsson. Particle filters for positioning in wireless networks. In *Procedings of EUSIPCO*, Toulouse, France, September 2002.

[87] P. B. Ober. Designing integrity into position estimation. *The Journal of Navigation*, 53(2), 2000.

[88] N. Peach. Bearings-only tracking using a set of range-parameterised extended Kalman filters. In *IEE Proceedings of Control Theory and Applications*, volume 142, 1995.

[89] M. K. Pitt and N. Shephard. Filtering via simulation: auxiliary particle filters. *Journal of the American Statistical Association*, 94:590–599, 1999.

[90] M. Prandini, J. Lygeros, A. Nilim, and S. Sastry. Randomized algorithms for probabilistic aircraft conflict detection. In *Proceedings of the 38th Conference on Decision and Control*, 1999.

[91] M. Prandini, J Hu, J Lygeros, and S. Sastry. A probabilistic approach to aircraft conflict detection. *IEEE Transactions on Intelligent Transportation Systems*, 1(4): 199–220, 2000.

[92] P. Pucar and J. Palmqvist. Saab NINS/NILS - an autonomous landing system for Gripen. In *Proceedings of the Position Location and Navigation Symposium, IEEE 2000*, 2000.

[93] K. Reif, S. Günther, E. Yas, and R. Unbehauen. Stochastic stability of the discrete-time extended Kalman filter. *IEEE Transactions on Automatic Control*, 44(4): 714–728, April 1999.

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

[95] H. Ruben. Probability content of regions under spherical normal distributions, iv: the distribution of homogeneous and non-homogeneous quadratic functions of normal variables. *The Annals of Mathematical Statistics*, 33(2), 1962.

[96] R. J. Schaefer. A standards-based approach to sense-and-avoid technology. In *Proceedings of the 3rd AIAA Unmanned Unlimited Technical Conference, Workshop and Exhibition*, 2004.

[97] T. Schön. *Estimation of Nonlinear Dynamic Systems: Theory and Applications*. PhD thesis 998, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 2006.

[98] T.B. Schön, F. Gustafsson, and P-J. Nordlund. Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing*, 53(7):2279–2289, July 2005.

[99] O. Shakernia, W-Z. Chen, and V. M. Raska. Passive ranging for UAV sense and avoid applications. In *Proceedings of Infotech@Aerospace*, 2005.

[100] J. Sheil and I. O'Muircheartaigh. Algorithm as 106: The distribution of non-negative quadratic forms in normal variables. *Applied Statistics*, 26(1), 1977.

[101] A. F. M. Smith and A. E. Gelfand. Bayesian statistics without tears: A sampling-resampling perspective. *The American Statistician*, 46(2), 1992.

[102] H. W. Sorenson and D. L. Alspach. Recursive Bayesian estimation using Gaussian sums. *Automatica*, 7:465–479, 1971.

[103] B-G. Sundqvist. Auto-ACAS - robust nuisance-free collision avoidance. In *Proceedings of the 44th Conference on Decision and Control and 2005 European Control Conference*, December 2005.

[104] *Aeronautical Information Publication (AIP) Sweden*. Swedish Civil Aviation Authority, 2007. http://www.lfv.se/sv/LFV/Flygtrafiktjansten/FPC/IAIP/.

[105] D. Törnqvist. *Estimation and Detection with Applications to Navigation*. PhD thesis 1216, Dept. of Electrical Engineering, Ohio University, 2008.

[106] J. K. Tugnait. Detection and estimation for abruptly changing systems. *Automatica*, 18(5):607–615, 1982.

[107] J. Utt, J. McCalmont, and M. Deschenes. Development of a sense and avoid system. In *Proceedings of Infotech@Aerospace*, 2005.

[108] C. Whelan. Evolution of air navigation services during this decade. *The Journal of Navigation*, 54(3), 2003.

[109] L.C. Yang and J.K. Kuchar. Prototype conflict alerting system for free flight. *Journal of Guidance, Control and Dynamics*, 20(4):768–773, 1997.

[110] A. D. Zeitlin and M. P. McLaughlin. Safety of cooperative collision avoidance for unmanned aircraft. *IEEE Aerospace and Electronic Systems Magazine*, 22(4), 2007.

Part II

Publications

Paper A

Marginalized Particle Filters for Mixed Linear/Nonlinear State-space Models

Authors: Schön, T.B. and Gustafsson, F. and Nordlund, P-J.

Edited version of the paper: T.B. Schön, F. Gustafsson, and P-J. Nordlund. Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing*, 53(7):2279–2289, July 2005

Preliminary version: Published as Technical Report LiTH-ISY-R-2548, Dept. of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden.

Marginalized Particle Filters for Mixed Linear/Nonlinear State-space Models

Thomas Schön*, and Fredrik Gustafsson*, and Per-Johan Nordlund**

*Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
schon, fredrik@isy.liu.se

**Department of Decision Support
Saab Aerosystems
581 88 Linköping, Sweden
per-johan.nordlund@saabgroup.com

Abstract

The particle filter offers a general numerical tool to approximate the posterior density function for the state in nonlinear and non-Gaussian filtering problems. While the particle filter is fairly easy to implement and tune, its main drawback is that it is quite computer intensive, with the computational complexity increasing quickly with the state dimension. One remedy to this problem is to marginalize out the states appearing linearly in the dynamics. The result is that one Kalman filter is associated with each particle. The main contribution in this article is to derive the details for the marginalized particle filter for a general nonlinear state-space model. Several important special cases occurring in typical signal processing applications will also be discussed. The marginalized particle filter is applied to an integrated navigation system for aircraft. It is demonstrated that the complete high-dimensional system can be based on a particle filter using marginalization for all but two states. Excellent performance on real flight data is reported.

1 Introduction

The nonlinear non-Gaussian filtering problem considered here consists of recursively computing the posterior probability density function of the state vector in a general discrete-time state-space model, given the observed measurements. Such a general model can be formulated as

$$x_{t+1} = f(x_t, w_t), \tag{1a}$$

$$y_t = h(x_t, e_t). \tag{1b}$$

Here, y_t is the measurement at time t, x_t is the state variable, w_t is the process noise, e_t is the measurement noise, and f, h are two arbitrary nonlinear functions. The two noise densities p_{w_t} and p_{e_t} are independent and are assumed to be known.

The posterior density $p(x_t|Y_t)$, where $Y_t = \{y_i\}_{i=0}^t$, is given by the following general measurement recursion

$$p(x_t|Y_t) = \frac{p(y_t|x_t)p(x_t|Y_{t-1})}{p(y_t|Y_{t-1})}, \tag{2a}$$

$$p(y_t|Y_{t-1}) = \int p(y_t|x_t)p(x_t|Y_{t-1})dx_t, \tag{2b}$$

and the following time recursion

$$p(x_{t+1}|Y_t) = \int p(x_{t+1}|x_t)p(x_t|Y_t)dx_t, \tag{2c}$$

initiated by $p(x_0|Y_{-1}) = p(x_0)$ [20]. For linear Gaussian models, the integrals can be solved analytically with a finite dimensional representation. This leads to the Kalman filter recursions, where the mean and the covariance matrix of the state are propagated [1]. More generally, no finite dimensional representation of the posterior density exists. Thus, several numerical approximations of the integrals (2) have been proposed. A recent important contribution is to use simulation based methods from mathematical statistics, sequential Monte Carlo methods, commonly referred to as particle filters [11, 12, 16].

Integrated navigation is used as a motivation and application example. Briefly, the integrated navigation system in the Swedish fighter aircraft Gripen consists of an inertial navigation system (INS), a terrain-aided positioning (TAP) system and an integration filter. This filter fuses the information from INS with the information from TAP. For a more thorough description of this system the reader is referred to [32, 33]. TAP is currently based on a point-mass filter as presented in [6], where it is also demonstrated that the performance is quite good, close to the Cramér-Rao lower bound. Field tests conducted by the Swedish air force have confirmed the good precision. Alternatives based on the extended Kalman filter have been investigated [5], but have been shown to be inferior particularly in the transient phase (the EKF requires the gradient of the terrain profile, which is unambiguous only very locally). The point-mass filter, as described in [6], is likely to be changed to a marginalized particle filter in the future for Gripen.

TAP and INS are the primary sensors. Secondary sensors (GPS and so on) are used only when available and reliable. The current terrain-aided positioning filter has three states (horizontal position and altitude), while the integrated navigation system estimates the accelerometer and gyroscope errors, and some other states. The integration filter is currently based on a Kalman filter with 27 states, taking INS and TAP as primary input signals.

The Kalman filter which is used for integrated navigation requires Gaussian variables. However, TAP gives a multi-modal un-symmetric distribution in the Kalman filter measurement equation and it has to be approximated with a Gaussian distribution before being used in the Kalman filter. This results in severe performance degradation in many cases, and is a common cause for filter divergence and system re-initialization.

The appealing new strategy is to merge the two state vectors into one, and solve integrated navigation and terrain-aided positioning in one filter. This filter should include

all 27 states, which effectively would prevent application of the particle filter. However, the state equation is almost linear, and only two states enter the measurement equation nonlinearly, namely horizontal position. Once linearization (and the use of EKF) is absolutely ruled out, marginalization would be the only way to overcome the computational complexity. More generally, as soon as there is a linear sub-structure available in the general model (1) this can be utilized in order to obtain better estimates and possibly reduce the computational demand. The basic idea is to partition the state vector as

$$x_t = \begin{bmatrix} x_t^l \\ x_t^n \end{bmatrix}, \tag{3}$$

where x_t^l denotes the state variable with conditionally linear dynamics and x_t^n denotes the nonlinear state variable [14, 32]. Using Bayes' theorem we can then marginalize out the linear state variables from (1) and estimate them using the Kalman filter [22], which is the optimal filter for this case. The nonlinear state variables are estimated using the particle filter. This technique is sometimes referred to as Rao-Blackwellization [14]. The idea has been around for quite some time, see e.g., [12, 7, 8, 2, 14, 31]. The contribution of this article is the derivation of the details for a general nonlinear state-space model with a linear sub-structure. Models of this type are common and important in engineering applications, e.g., positioning, target tracking and collision avoidance [18, 4]. The marginalized particle filter has been successfully used in several applications, for instance in aircraft navigation [32], underwater navigation [23], communications [9, 37], nonlinear system identification [28, 36], and audio source separation [3].

Section 2 explains the idea behind using marginalization in conjunction with general linear/nonlinear state-space models. Three nested models are used, in order to make the presentation easy to follow. Some important special cases and generalizations of the noise assumptions are discussed in Section 3. To illustrate the use of the marginalized particle filter a synthetic example is given in Section 4. Finally, the application example is given in Section 5 and the conclusions are stated in Section 6.

2 Marginalization

The variance of the estimates obtained from the standard particle filter can be decreased by exploiting linear sub-structures in the model. The corresponding variables are marginalized out and estimated using an optimal linear filter. This is the main idea behind the marginalized particle filter. The goal of this section is to explain how the marginalized particle filter works by using three nested models. The models are nested in the sense that the first model is included in the second, which in turn is included in the third. The reason for presenting it in this fashion is to facilitate reader understanding, by incrementally extending the standard particle filter.

2.1 The Standard Particle Filter

The particle filter is used to get an approximation of the posterior density $p(x_t|Y_t)$ in the general model (1). This approximation can then be used to obtain an estimate of some

inference function, $g(\cdot)$, according to

$$I(g(x_t)) = E_{p(x_t|Y_t)}[g(x_t)] = \int g(x_t)p(x_t|Y_t)dx_t. \tag{4}$$

In the following the particle filter, as it was introduced in [16], will be referred to as the standard particle filter. For a thorough introduction to the standard particle filter the reader is referred to [11, 12]. The marginalized and the standard particle filter are closely related. The marginalized particle filter is given in Algorithm 1 and neglecting steps 4a and 4c results in the standard particle filter algorithm.

Algorithm 1 (The marginalized particle filter).

1. Initialization: For $i = 1, \ldots, N$, initialize the particles, $x^{n,(i)}_{0|-1} \sim p_{x^n_0}(x^n_0)$ and set $\{x^{l,(i)}_{0|-1}, P^{(i)}_{0|-1}\} = \{\bar{x}^l_0, \bar{P}_0\}$.

2. For $i = 1, \ldots, N$, evaluate the importance weights $q^{(i)}_t = p(y_t|X^{n,(i)}_t, Y_{t-1})$ and normalize

$$\tilde{q}^{(i)}_t = \frac{q^{(i)}_t}{\sum_{j=1}^N q^{(j)}_t}.$$

3. Particle filter measurement update (resampling): Resample N particles with replacement,

$$\Pr(x^{n,(i)}_{t|t} = x^{n,(j)}_{t|t-1}) = \tilde{q}^{(j)}_t.$$

4. Particle filter time update and Kalman filter:

   (a) Kalman filter measurement update:
   Model 1: (10),
   Model 2: (10),
   Model 3: (22).

   (b) Particle filter time update (prediction): For $i = 1, \ldots, N$, predict new particles,

$$x^{n,(i)}_{t+1|t} \sim p(x^n_{t+1|t}|X^{n,(i)}_t, Y_t).$$

   (c) Kalman filter time update:
   Model 1: (11),
   Model 2: (17),
   Model 3: (23).

5. Set $t := t + 1$ and iterate from step 2.

The particle filter algorithm 1 is quite general and several improvements are available in the literature. It is quite common to introduce artificial noise in step 3 in order to counteract the degeneracy problem. Depending on the context various importance functions can be used in step 4b. In [11] several refinements to the particle filter algorithm are discussed.

2.2 Diagonal Model

The explanation of how the marginalized particle filter works is started by considering the following model,

Model 1.

$$x_{t+1}^n = f_t^n(x_t^n) + w_t^n, \tag{5a}$$
$$x_{t+1}^l = A_t^l(x_t^n)x_t^l + w_t^l, \tag{5b}$$
$$y_t = h_t(x_t^n) + C_t(x_t^n)x_t^l + e_t. \tag{5c}$$

The gaps in the equations above are placed there intentionally, in order to make the comparison to the general model (18) easier. The state noise is assumed white and Gaussian distributed according to

$$w_t = \begin{bmatrix} w_t^l \\ w_t^n \end{bmatrix} \sim \mathcal{N}(0, Q_t), \quad Q_t = \begin{bmatrix} Q_t^l & 0 \\ 0 & Q_t^n \end{bmatrix}. \tag{6a}$$

The measurement noise is assumed white and Gaussian distributed according to

$$e_t \sim \mathcal{N}(0, R_t). \tag{6b}$$

Furthermore, x_0^l is Gaussian,

$$x_0^l \sim \mathcal{N}(\bar{x}_0, \bar{P}_0). \tag{6c}$$

The density of x_0^n can be arbitrary, but it is assumed known. The A^l and C matrices are arbitrary. □

Model 1 is called "diagonal model" due to the diagonal structure of the state equation (5a) – (5b). The aim of recursively estimating the posterior density $p(x_t|Y_t)$ can be accomplished using the standard particle filter. However, conditioned on the nonlinear state variable, x_t^n, there is a linear sub-structure in (5), given by (5b). This fact can be used to obtain better estimates of the linear states. Analytically marginalizing out the linear state variables from $p(x_t|Y_t)$ and using Bayes' theorem gives ($X_t^n = \{x_i^n\}_{i=0}^t$)

$$p(x_t^l, X_t^n|Y_t) = \underbrace{p(x_t^l|X_t^n, Y_t)}_{\text{Optimal KF}} \underbrace{p(X_t^n|Y_t)}_{\text{PF}}, \tag{7}$$

where $p(x_t^l|X_t^n, Y_t)$ is analytically tractable. It is given by the Kalman filter (KF), see Lemma 1 below for the details. Furthermore, $p(X_t^n|Y_t)$ can be estimated using the particle filter (PF). If the same number of particles are used in the standard particle filter and the marginalized particle filter the latter will, intuitively, provide better estimates. The reason for this is that the dimension of $p(x_t^n|Y_t)$ is smaller than the dimension of $p(x_t^l, x_t^n|Y_t)$, implying that the particles occupy a lower dimensional space. Another reason is that optimal algorithms are used in order to estimate the linear state variables. Let $\hat{I}_N^s(g(x_t))$ denote the estimate of (4) using the standard particle filter with N particles. When the

marginalized particle filter is used the corresponding estimate is denoted by $\hat{I}_N^m(g(x_t))$. Under certain assumptions the following central limit theorem holds,

$$\sqrt{N}(\hat{I}_N^s(g(x_t)) - I(g(x_t))) \Longrightarrow \mathcal{N}(0, \sigma_s^2), \quad N \to \infty \tag{8a}$$

$$\sqrt{N}(\hat{I}_N^m(g(x_t)) - I(g(x_t))) \Longrightarrow \mathcal{N}(0, \sigma_m^2), \quad N \to \infty \tag{8b}$$

where $\sigma_s^2 \geq \sigma_m^2$. A formal proof of (8) is provided in [14, 13]. For the sake of notational brevity the dependence of x_t^n in A_t, C_t, and h_t are suppressed below.

Lemma 1
Given Model 1, the conditional probability density functions for $x_{t|t}^l$ and $x_{t+1|t}^l$ are given by

$$p(x_t^l|X_t^n, Y_t) = \mathcal{N}(\hat{x}_{t|t}^l, P_{t|t}), \tag{9a}$$

$$p(x_{t+1}^l|X_{t+1}^n, Y_t) = \mathcal{N}(\hat{x}_{t+1|t}^l, P_{t+1|t}), \tag{9b}$$

where

$$\hat{x}_{t|t}^l = \hat{x}_{t|t-1}^l + K_t(y_t - h_t - C_t\hat{x}_{t|t-1}^l), \tag{10a}$$

$$P_{t|t} = P_{t|t-1} - K_t C_t P_{t|t-1}, \tag{10b}$$

$$S_t = C_t P_{t|t-1} C_t^T + R_t, \tag{10c}$$

$$K_t = P_{t|t-1} C_t^T S_t^{-1}, \tag{10d}$$

and

$$\hat{x}_{t+1|t}^l = A_t^l \hat{x}_{t|t}^l, \tag{11a}$$

$$P_{t+1|t} = A_t^l P_{t|t} (A_t^l)^T + Q_t^l. \tag{11b}$$

The recursions are initiated with $\hat{x}_{0|-1}^l = \bar{x}_0$ and $P_{0|-1} = \bar{P}_0$.

Proof: Straightforward application of the Kalman filter [22, 21]. □

The second density, $p(X_t^n|Y_t)$, in (7) will be approximated using the standard particle filter. Bayes' theorem and the Markov property inherent in the state-space model can be used to write $p(X_t^n|Y_t)$ as

$$p(X_t^n|Y_t) = \frac{p(y_t|X_t^n, Y_{t-1}) p(x_t^n|X_{t-1}^n, Y_{t-1})}{p(y_t|Y_{t-1})} p(X_{t-1}^n|Y_{t-1}), \tag{12}$$

where an approximation of $p(X_{t-1}^n|Y_{t-1})$ is provided by the previous iteration of the particle filter. In order to perform the update (12) analytical expressions for $p(y_t|X_t^n, Y_{t-1})$ and $p(x_t^n|X_{t-1}^n, Y_{t-1})$ are needed. They are provided by the following lemma.

Lemma 2
For Model 1 $p(y_t|X_t^n, Y_{t-1})$ and $p(x_{t+1}^n|X_t^n, Y_t)$ are given by

$$p(y_t|X_t^n, Y_{t-1}) = \mathcal{N}(h_t + C_t\hat{x}_{t|t-1}^l,\, C_t P_{t|t-1} C_t^T + R_t), \tag{13a}$$

$$p(x_{t+1}^n|X_t^n, Y_t) = \mathcal{N}(f_t^n,\, Q_t^n). \tag{13b}$$

Proof: Basic facts about conditionally linear models, see e.g., [19, 35]. □

The linear system (5b) – (5c) can now be formed for each particle, $\{x_t^{n,(i)}\}_{i=1}^N$ and the linear states can be estimated using the Kalman filter. This requires one Kalman filter associated with each particle. The overall algorithm for estimating the states in Model 1 is given in Algorithm 1. From this algorithm it should be clear that the only difference from the standard particle filter is that the time update (prediction) stage has been changed. In the standard particle filter the prediction stage is given solely by step 4b in Algorithm 1. Step 4a is referred to as the *measurement update* in the Kalman filter [21]. Furthermore, the prediction of the nonlinear state variables, $\hat{x}^n_{t+1|t}$ is obtained in step 4b. According to (5a) the prediction of the nonlinear state variables does not contain any information about the linear state variables. This implies that $\hat{x}^n_{t+1|t}$ cannot be used to improve the quality of the estimates of the linear state variables. However, if Model 1 is generalized by imposing a dependence between the linear and the nonlinear state variables in (5a) the prediction of the nonlinear state variables can be used to improve the estimates of the linear state variables. In the subsequent section it will be elaborated on how this affects the state estimation.

2.3 Triangular Model

Model 1 is now extended by including the term $A_t^n(x_t^n)x_t^l$ in the nonlinear state equation. This results in a "triangular model", defined below.

Model 2.

$$x^n_{t+1} = f^n_t(x^n_t) + A^n_t(x^n_t)x^l_t + w^n_t, \tag{14a}$$
$$x^l_{t+1} = A^l_t(x^n_t)x^l_t + w^l_t, \tag{14b}$$
$$y_t = h_t(x^n_t) + C_t(x^n_t)x^l_t + e_t, \tag{14c}$$

with the same assumptions as in Model 1. □

Now, from (14a) it is clear that $\hat{x}^n_{t+1|t}$ does indeed contain information about the linear state variables. This implies that there will be information about the linear state variable, x^l_t, in the prediction of the nonlinear state variable, $\hat{x}^n_{t+1|t}$. To understand how this affects the derivation it is assumed that step 4b in Algorithm 1 has just been completed. This means that the predictions, $\hat{x}^n_{t+1|t}$, are available and the model can be written (the information in the measurement, y_t, has already been used in step 4a)

$$x^l_{t+1} = A^l_t x^l_t + w^l_t, \tag{15a}$$
$$z_t = A^n_t x^l_t + w^n_t, \tag{15b}$$

where

$$z_t = x^n_{t+1} - f^n_t. \tag{15c}$$

It is possible to interpret z_t as a measurement and w_t^n as the corresponding measurement noise. Since (15) is a linear state-space model with Gaussian noise, the optimal state estimate is given by the Kalman filter according to

$$\hat{x}_{t|t}^{l*} = \hat{x}_{t|t}^{l} + L_t(z_t - A_t^n \hat{x}_{t|t}^{l}), \tag{16a}$$

$$P_{t|t}^{*} = P_{t|t} - L_t N_t L_t^T, \tag{16b}$$

$$L_t = P_{t|t}(A_t^n)^T N_t^{-1}, \tag{16c}$$

$$N_t = A_t^n P_{t|t}(A_t^n)^T + Q_t^n, \tag{16d}$$

where "$*$" has been used to distinguish this second measurement update from the first one. Furthermore, $\hat{x}_{t|t}^{l}$, and $P_{t|t}$ are given by (10a) and (10b) respectively. The final step is to merge this second measurement update with the time update to obtain the predicted states. This results in

$$\hat{x}_{t+1|t}^{l} = A_t^l \hat{x}_{t|t}^{l} + L_t(z_t - A_t^n \hat{x}_{t|t}^{l}), \tag{17a}$$

$$P_{t+1|t} = A_t^l P_{t|t}(A_t^l)^T + Q_t^l - L_t N_t L_t^T, \tag{17b}$$

$$L_t = A_t^l P_{t|t}(A_t^n)^T N_t^{-1}, \tag{17c}$$

$$N_t = A_t^n P_{t|t}(A_t^n)^T + Q_t^n. \tag{17d}$$

For a formal proof of this the reader is referred to Appendix A.1. To make Algorithm 1 valid for the more general Model 2 the time update equation in the Kalman filter (11) has to be replaced by (17).

The second measurement update is called measurement update due to the fact that the mathematical structure is the same as a measurement update in the Kalman filter. However, strictly speaking it is not really a measurement update, since there does not exist any new measurement. It is better to think of this second update as a correction to the real measurement update, using the information in the prediction of the nonlinear state variables.

2.4 The General Case

In the previous two sections the mechanisms underlying the marginalized particle filter have been illustrated. It is now time to apply the marginalized particle filter to the most general model.

Model 3.

$$x_{t+1}^n = f_t^n(x_t^n) + A_t^n(x_t^n)x_t^l + G_t^n(x_t^n)w_t^n, \tag{18a}$$

$$x_{t+1}^l = f_t^l(x_t^n) + A_t^l(x_t^n)x_t^l + G_t^l(x_t^n)w_t^l, \tag{18b}$$

$$y_t = h_t(x_t^n) + C_t(x_t^n)x_t^l + e_t, \tag{18c}$$

where the state noise is assumed white and Gaussian distributed with

$$w_t = \begin{bmatrix} w_t^l \\ w_t^n \end{bmatrix} \sim \mathcal{N}(0, Q_t), \quad Q_t = \begin{bmatrix} Q_t^l & Q_t^{ln} \\ (Q_t^{ln})^T & Q_t^n \end{bmatrix}. \tag{19a}$$

The measurement noise is assumed white and Gaussian distributed according to

$$e_t \sim \mathcal{N}(0, R_t). \tag{19b}$$

Furthermore, x_0^l is Gaussian,

$$x_0^l \sim \mathcal{N}(\bar{x}_0, \bar{P}_0). \tag{19c}$$

The density of x_0^n can be arbitrary, but it is assumed known. □

In certain cases some of the assumptions can be relaxed. This will be discussed in the subsequent section. Before moving on it is worthwhile to explain how models used in some applications of marginalization relate to Model 3. In [24] the marginalized particle filter was applied to underwater navigation using a model corresponding to (18), save the fact that $G_t^n = I, G_t^l = I, f_t^l = 0, A_t^n = 0$. In [18] a model corresponding to linear state equations and a nonlinear measurement equation is applied to various problems, such as car positioning, terrain navigation, and target tracking. Due to its relevance this model will be discussed in more detail in Section 3. Another special case of Model 3 has been applied to problems in communication theory in [9, 37]. The model used there is linear. However, depending on an indicator variable the model changes. Hence, this indicator variable can be thought of as the nonlinear state variable in Model 3. A good and detailed explanation of how to use the marginalized particle filter for this case can be found in [14]. They refer to the model as a jump Markov linear system.

Analogously to what has been done in (7), the filtering distribution, $p(x_t|Y_t)$ is split using Bayes' theorem,

$$p(x_t^l, X_t^n|Y_t) = p(x_t^l|X_t^n, Y_t)p(X_t^n|Y_t). \tag{20}$$

The linear state variables are estimated using the Kalman filter in a slightly more general setting than which was previously discussed. However, it is still the same three steps that are executed in order to estimate the linear state variables. The first step is a measurement update using the information available in y_t. The second step is a measurement update using the information available in $\hat{x}_{t+1|t}^n$ and finally there is a time update. The following theorem explains how the linear state variables are estimated.

Theorem 1
Using Model 3 the conditional probability density functions for x_t^l and x_{t+1}^l are given by

$$p(x_t^l|X_t^n, Y_t) = \mathcal{N}(\hat{x}_{t|t}^l, P_{t|t}), \tag{21a}$$

$$p(x_{t+1}^l|X_{t+1}^n, Y_t) = \mathcal{N}(\hat{x}_{t+1|t}^l, P_{t+1|t}), \tag{21b}$$

where

$$\hat{x}_{t|t}^l = \hat{x}_{t|t-1}^l + K_t(y_t - h_t - C_t\hat{x}_{t|t-1}^l), \tag{22a}$$

$$P_{t|t} = P_{t|t-1} - K_t M_t K_t^T, \tag{22b}$$

$$M_t = C_t P_{t|t-1} C_t^T + R_t, \tag{22c}$$

$$K_t = P_{t|t-1} C_t^T M_t^{-1}, \tag{22d}$$

and

$$\begin{aligned}\hat{x}^l_{t+1|t} &= \bar{A}^l_t \hat{x}^l_{t|t} + G^l_t (Q^{ln}_t)^T (G^n_t Q^n_t)^{-1} z_t \\ &\quad + f^l_t + L_t(z_t - A^n_t \hat{x}^l_{t|t}), \qquad (23a)\\ P_{t+1|t} &= \bar{A}^l_t P_{t|t} (\bar{A}^l_t)^T + G^l_t \bar{Q}^l_t (G^l_t)^T - L_t N_t L_t^T, \qquad (23b)\\ N_t &= A^n_t P_{t|t} (A^n_t)^T + G^n_t Q^n_t (G^n_t)^T, \qquad (23c)\\ L_t &= \bar{A}^l_t P_{t|t} (A^n_t)^T N_t^{-1}, \qquad (23d)\end{aligned}$$

where

$$\begin{aligned} z_t &= x^n_{t+1} - f^n_t, \qquad (24a)\\ \bar{A}^l_t &= A^l_t - G^l_t (Q^{ln}_t)^T (G^n_t Q^n_t)^{-1} A^n_t, \qquad (24b)\\ \bar{Q}^l_t &= Q^l_t - (Q^{ln}_t)^T (Q^n_t)^{-1} Q^{ln}_t. \qquad (24c)\end{aligned}$$

Proof: See Appendix A.1. □

It is worth noting that if the cross-covariance, Q^{ln}_t, between the two noise sources w^n_t and w^l_t is zero, then $\bar{A}^l_t = A^l_t$ and $\bar{Q}^l_t = Q^l_t$. The first density, $p(x^l_t|X^n_t, Y_t)$, on the right hand side in (20) is now taken care of. In order for the estimation to work the second density, $p(X^n_t|Y_t)$, in (20) is taken care of according to (12). The analytical expressions for $p(y_t|X^n_t, Y_{t-1})$ and $p(x^n_t|X^n_{t-1}, Y_{t-1})$ are provided by the following theorem.

Theorem 2
For Model 3 $p(y_t|X^n_t, Y_{t-1})$ and $p(x^n_{t+1}|X^n_t, Y_t)$ are given by

$$\begin{aligned} p(y_t|X^n_t, Y_{t-1}) &= \mathcal{N}(h_t + C_t \hat{x}^l_{t|t-1},\ C_t P_{t|t-1} C_t^T + R_t), \qquad (25a)\\ p(x^n_{t+1}|X^n_t, Y_t) &= \mathcal{N}(f^n_t + A^n_t \hat{x}^l_{t|t},\ A^n_t P_{t|t} (A^n_t)^T \\ &\quad + G^n_t Q^n_t (G^n_t)^T). \qquad (25b)\end{aligned}$$

Proof: Basic facts about conditionally linear models, see [19]. The details for this particular case can be found in [35]. □

The details for estimating the states in Model 3 have now been derived, and the complete algorithm is Algorithm 1. As pointed out before, the only difference between this algorithm and the standard particle filtering algorithm is that the prediction stage is different. If steps 4a and 4c are removed from Algorithm 1 the standard particle filter algorithm is obtained.

In this article the most basic form of the particle filter has been used. Several more refined variants exist, which in certain applications can give better performance. However, since the aim of this article is to communicate the idea of marginalization in a general linear/nonlinear state-space model the standard particle filter has been used. It is straightforward to adjust the algorithm given in this paper to accommodate e.g., the auxiliary particle filter [34] and the Gaussian particle filter [26, 27]. Several ideas are also given in the articles collected in [11].

The estimates as expected means of the linear state variables and their covariances are given by [32]

$$\hat{x}^l_{t|t} = \sum_{i=1}^{N} \tilde{q}_t^{(i)} \hat{x}^{l,(i)}_{t|t} \approx E_{p(x^l_t|Y_t)} \left[x^l_t \right], \tag{26a}$$

$$\hat{P}_{t|t} = \sum_{i=1}^{N} \tilde{q}_t^{(i)} \left(P^{(i)}_{t|t} + (\hat{x}^{l,(i)}_{t|t} - \hat{x}^l_{t|t})(\hat{x}^{l,(i)}_{t|t} - \hat{x}^l_{t|t})^T \right) \tag{26b}$$

$$\approx E_{p(x^l_t|Y_t)} \left[\left((x^l_t)^2 - E_{p(x^l_t|Y_t)} \left[(x^l_t)^2 \right] \right)^2 \right]. \tag{26c}$$

where $\tilde{q}_t^{(i)}$ are the normalized importance weights, provided by step 2 in Algorithm 1.

3 Important Special Cases and Extensions

Model 3 is quite general indeed and in most applications special cases of it are used. This fact, together with some extensions will be the topic of this section.

The special cases are just reductions of the general results presented in the previous section. However, they still deserve some attention in order to highlight some important mechanisms. It is worth mentioning that linear sub-structures can enter the model more implicitly as well, for example, by modeling colored noise and by sensor offsets and trends. These modeling issues are treated in several introductory texts on Kalman filtering, see e.g., Section 8.2.4 in [17]. In the subsequent section some noise modeling aspects are discussed. This is followed by a discussion of a model with linear state equations and a nonlinear measurement equation.

3.1 Generalized Noise Assumptions

The Gaussian noise assumption can be relaxed in two special cases. First, if the measurement equation (18c) does not depend on the linear state variables, x^l_t, i.e., $C_t(x^n_t) = 0$, the measurement noise can be arbitrarily distributed. In this case (18c) does not contain any information about the linear state variables, and hence cannot be used in the Kalman filter. It is solely used in the particle filter part of the algorithm, which can handle all probability density functions.

Second, if $A^n_t(x^n_t) = 0$ in (18a), the nonlinear state equation will be independent of the linear states, and hence cannot be used in the Kalman filter. This means that the state noise, w^n_t, can be arbitrarily distributed.

The noise covariances can depend on the nonlinear state variables, i.e., $R_t = R_t(x^n_t)$ and $Q_t = Q_t(x^n_t)$. This is useful for instance in terrain navigation, where the nonlinear state variable includes information about the position. Using the horizontal position and a geographic information system (GIS) on-board the aircraft noise covariances depending on the characteristics of the terrain at the current horizontal position can be motivated. This issue will be elaborate upon in Section 5.

3.2 An Important Model Class

A quite important special case of Model 3, is a model with linear state equations and a nonlinear measurement equation. In Model 4 below such a model is defined.

Model 4.

$$x^n_{t+1} = A^n_{n,t}x^n_t + A^n_{l,t}x^l_t + G^n_t w^n_t, \tag{27a}$$

$$x^l_{t+1} = A^l_{n,t}x^n_t + A^l_{l,t}x^l_t + G^l_t w^l_t, \tag{27b}$$

$$y_t = h_t(x^n_t) + e_t, \tag{27c}$$

with $w^n_t \sim \mathcal{N}(0, Q^n_t)$ and $w^l_t \sim \mathcal{N}(0, Q^l_t)$. The distribution for e_t can be arbitrary, but it is assumed known. □

The measurement equation (27c) does not contain any information about the linear state variable, x^l_t. Hence, as far as the Kalman filter is concerned (27c) cannot be used in estimating the linear states. Instead all information from the measurements enter the Kalman filter implicitly via the second measurement update using the nonlinear state equation (27a) and the prediction of the nonlinear state, $\hat{x}^n_{t+1|t}$, as a measurement. This means that in Algorithm 1, step 4a can be left out. In this case the second measurement update is much more than just a correction to the first measurement update. It is the only way in which the information in y_t enters the algorithm.

Model 4 is given special attention as several important state estimation problems can be modeled in this way. Examples include positioning, target tracking and collision avoidance [18, 4]. For more information on practical matters concerning modeling issues, see e.g., [30, 29, 4, 32]. In the applications mentioned above the nonlinear state variable, x^n_t, usually corresponds to the position, whereas the linear state variable, x^l_t, corresponds to velocity, acceleration and bias terms.

If Model 4 is compared to Model 3 it can be seen that the matrices A^n_t, A^l_t, G^n_t, and G^l_t are independent of x^n_t in Model 4, which implies that

$$P^{(i)}_{t|t} = P_{t|t} \qquad \forall\, i = 1, \ldots, N. \tag{28}$$

This follows from (23b) – (23d) in Theorem 1. According to (28) only one instead of N Riccati recursions is needed, which leads to a substantial reduction in computational complexity. This is of course very important in real-time implementations. A further study of the computational complexity of the marginalized particle filter can be found in [25].

If the dynamics in (18a) – (18b) is almost linear it can be linearized to obtain a model described by (27a) – (27b). Then the extended Kalman filter can be used instead of the Kalman filter. As is explained in [30, 29] it is common that the system model is almost linear, whereas the measurement model is severely nonlinear. In these cases use the particle filter for the severe nonlinearities and the extended Kalman filter for the mild nonlinearities.

4 An Illustrating Example

In order to make things as simple as possible the following two dimensional model will be used

$$x_{t+1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x_t + w_t, \tag{29a}$$

$$y_t = h(z_t) + e_t, \tag{29b}$$

where the state vector is $x_t = \begin{bmatrix} z_t & \dot{z}_t \end{bmatrix}^T$. Hence, the state consists of a physical variable and its derivative. Models of this kind are very common in applications. One example is bearings only tracking, where the objective is to estimate the angle and angular velocity and the nonlinear measurement depends on the antenna diagram. Another common application is state estimation in a DC-motor, where the angular position is assumed to be measured nonlinearly. As a final application terrain navigation in one dimension is mentioned, where the measurement is given by a map. A more realistic terrain navigation example is discussed in Section 5.

Model (29) is linear in $\dot{z}_t$ and nonlinear in z_t. The state vector can thus be partitioned as $x_t = \begin{bmatrix} x_t^n & x_t^l \end{bmatrix}^T$, which implies that (29) can be written as

$$x_{t+1}^n = x_t^n \quad + T x_t^l + w_t^n, \tag{30a}$$

$$x_{t+1}^l = \quad x_t^l \; + w_t^l, \tag{30b}$$

$$y_t = h_t(x_t^n) \quad + e_t, \tag{30c}$$

This corresponds to the triangular model given in Model 2. Hence, the Kalman filter for the linear state variable is given by (22) – (24) where the nonlinear state is provided by the particle filter. The estimate of the linear state variable is given by (23a) which for this example is

$$\hat{x}_{t+1|t}^l = (1 - l_t T)\hat{x}_{t|t}^l + l_t T \frac{x_{t+1}^n - x_t^n}{T}, \tag{31}$$

where

$$n_t = T^2 p_{t|t} + q_t^n, \qquad l_t = \frac{T}{n_t} p_{t|t}. \tag{32}$$

Intuitively (31) makes sense, since the velocity estimate is given as a weighted average of the current velocity and the estimated momentary velocity, where the weights are computed from the Kalman filter quantities. In cases where (29a) is motivated by Newtons' force law the unknown force is modeled as a disturbance and $q_t^n = 0$. This implies that (31) is reduced to

$$\hat{x}_{t+1|t}^l = \frac{x_{t+1}^n - x_t^n}{T}. \tag{33}$$

Again this can intuitively be understood, since conditioned on the knowledge of the nonlinear state variable, (30a) can be written

$$x_t^l = \frac{x_{t+1}^n - x_t^n}{T}. \tag{34}$$

Thus, (30b) does not add any information for the Kalman filter, since x_t^l is a deterministic function of the known nonlinear state variable.

5 Integrated Aircraft Navigation

As was explained in the introduction, the integrated navigation system in the Swedish fighter aircraft Gripen consists of an inertial navigation system (INS), a terrain-aided positioning (TAP) system and an integration filter. This filter fuses the information from INS with the information from TAP, see Fig. 1. The currently used integration filter, is

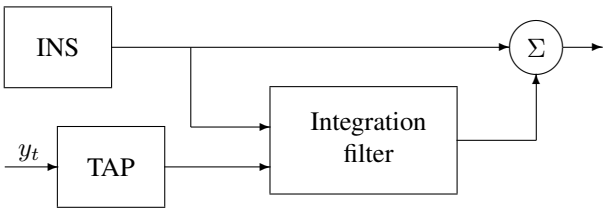

Figure 1: *The integrated navigation system consists of an inertial navigation system (INS), a terrain-aided positioning (TAP) system and an integration filter. The integration filter fuse the information from INS with the information from TAP.*

likely to be changed to a marginalized particle filter in the future for Gripen, see Fig. 2. A

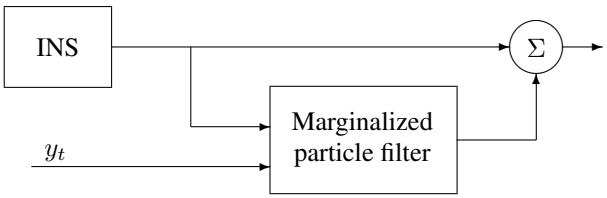

Figure 2: *Using the marginalized particle filter for navigation. The terrain information is now incorporated directly in the marginalized particle filter. The radar altimeter delivers the hight measurement* y_t.

first step in this direction was taken in [18], where a six dimensional model was used for integrated navigation. In six dimensions, the particle filter is possible to use, but better performance can be obtained. As demonstrated in [18], 4000 particles in the marginalized filter outperforms 60000 particles in the standard particle filter.

The feasibility study presented here applies marginalization to a more realistic nine dimensional sub-model of the total integrated navigation system. Already here, the dimensionality has proven to be too large for the particle filter to be applied directly. The example contains all ingredients of the total system, and the principle is scalable to the full 27-dimensional state vector. The model can be simulated and evaluated in a controlled fashion, see [32] for more details. In the subsequent sections the results from field trials are presented.

5.1 The Dynamic Model

In order to apply the marginalized particle filter to the navigation problem a dynamic model of the aircraft is needed. In this article the overall structure of this model is discussed. For details the reader is referred to [32] and the references therein. The errors in the states are estimated instead of the absolute states. The reason is that the dynamics of the errors are typically much slower than the dynamics of the absolute states. The model has the following structure

$$x_{t+1}^n = A_{n,t}^n x_t^n + A_{l,t}^n x_t^l + G_t^n w_t^n, \tag{35a}$$

$$x_{t+1}^l = A_{n,t}^l x_t^n + A_{l,t}^l x_t^l + G_t^l w_t^l, \tag{35b}$$

$$y_t = h\left(\begin{bmatrix} L_t \\ l_t \end{bmatrix} + x_t^n\right) + e_t. \tag{35c}$$

There are 7 linear states, and 2 nonlinear states. The linear states consist of 2 velocity states and 3 states for the aircraft in terms of heading, roll, and pitch. Finally there are 2 states for the accelerometer bias. The nonlinear states correspond to the error in the horizontal position, which is expressed in latitude, L_t, and longitude, l_t.

The total dimension of the state vector is thus 9, which is too large to be handled by the particle filter. The highly nonlinear nature of measurement equation (35c), due to the terrain elevation database, implies that an extended Kalman filter cannot be used. However, the model described by (35) clearly fits into the framework of the marginalized particle filter.

The measurement noise in (35c) deserves some special attention. The radar altimeter, which is used to measure the ground clearance, interprets any echo as the ground. This is a problem when flying over trees. The tree tops will be interpreted as the ground, with a false measurement as a result. One simple, but effective, solution to this problem is to model the measurement noise as

$$p_{e_t}(\cdot) = \pi\mathcal{N}(m_1, \sigma_1) + (1-\pi)\mathcal{N}(m_2, \sigma_2), \tag{36}$$

where π is the probability of obtaining an echo from the ground, and $(1-\pi)$ is the probability of obtaining an echo from the tree tops. The probability density function (36) is shown in Fig. 3. Experiments have shown that this, in spite of its simplicity, is a quite accurate model [10]. Furthermore, m_1, m_2, σ_1, σ_2, and π in (36) can be allowed to depend on the current horizontal position, L_t, l_t. In this way information from the terrain data base can be inferred on the measurement noise in the model. Using this information it is possible to model whether the aircraft is flying over open water or over a forest.

5.2 Result

The flight that has been used is shown in Fig. 4. This is a fairly tough flight for the algorithm, in the sense that during some intervals data are missing, and sometimes the radar altimeter readings become unreliable. This happens at high altitudes and during sharp turns (large roll angle), respectively. In order to get a fair understanding of the algorithms performance, 100 Monte Carlo simulations with the same data have been performed, where only the noise realizations have been changed from one simulation to the

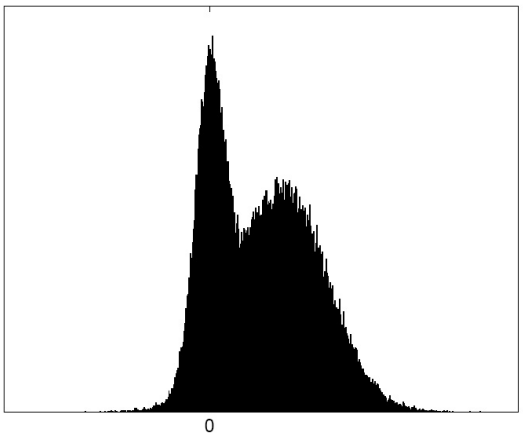

Figure 3: *A typical histogram of the error in the data from the radar altimeter. The first peak corresponds to the error in the ground reading and the second peak corresponds to the error in the readings from the tree tops.*

other. Many parameters have to be chosen, but only the number of particles used are commented here (see [15] for more details). In Fig. 5 a plot of the error in horizontal position as a function of time is presented, for different number of particles. The true position is provided by the differential GPS (DGPS). From this figure it is obvious that the estimate improves as more particles are used. This is natural since the theory states that the densities are approximated better the more particles used. The difference in performance is mainly during the transient, where it can be motivated to use more particles. By increasing the number of particles the convergence time is significantly reduced and a better estimate is obtained. This is true up to 5000 particles. Hence, 5000 particles where used in this study. The algorithm can be further improved, and in [15] several suggestions are given.

The conclusion from this study is that the marginalized particle filter performs well, and provides an interesting and powerful alternative to methods currently used in integrated aircraft navigation systems.

6 Conclusions

The marginalization techniques have systematically been applied to general nonlinear and non-Gaussian state-space models, with linear sub-structures. This has been done in several steps, where each step implies a certain modification of the standard particle filter. The first step was to associate one Kalman filter with each particle. These Kalman filters were used to estimate the linear states. The second step was to use the prediction of the nonlinear state as an additional measurement. This was used to obtain better estimates

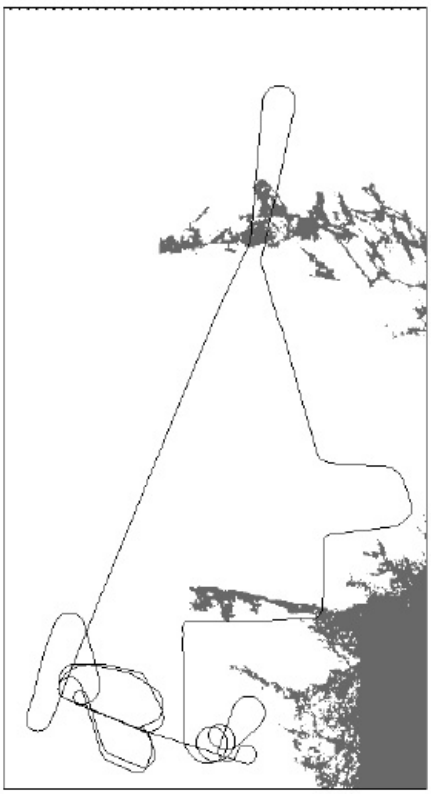

Figure 4: *The flight path used for testing the algorithm. The flight path is clockwise and the dark regions in the figure are open water.*

of the linear state variables. The complete details for the marginalized particle filter were derived for a general nonlinear and non-Gaussian state-space model. Several important special cases were also described. Conditions implying that all the Kalman filters will obey the same Riccati recursion were given.

Finally, a terrain navigation application with real data from the Swedish fighter aircraft Gripen was presented. The particle filter is not a feasible algorithm for the full nine-state model since a huge number of particles would be needed. However, since only two states (the aircrafts horizontal position) appear nonlinearly in the measurement equation, a special case of the general marginalization algorithm can be applied. A very good result can be obtained with only 5000 particles, which is readily possible to implement in the computer currently used in the aircraft.

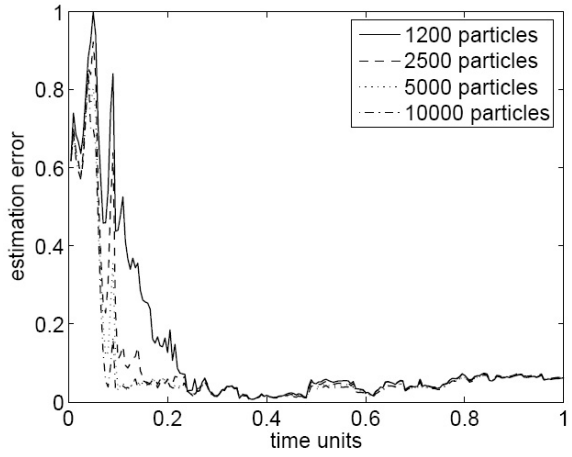

Figure 5: *The horizontal position error as a function of time units for different numbers of particles. The marginalized particle filter given in Algorithm 1 has been used.*

A Appendix

A.1 Proof for Theorem 1

The proof of (16) and (17) is provided as a special case of the proof below.

Proof: For the sake of notational brevity the dependence on x_t^n in (18) is suppressed in this proof. Write (18) as

$$x_{t+1}^l = f_t^l + A_t^l x_t^l + G_t^l w_t^l, \tag{37a}$$

$$z_t^1 = A_t^n x_t^l + G_t^n w_t^n, \tag{37b}$$

$$z_t^2 = C_t x_t^l + e_t, \tag{37c}$$

where z_t^1 and z_t^2 are defined as

$$z_t^1 = x_{t+1}^n - f_t^n, \tag{37d}$$

$$z_t^2 = y_t - h_t. \tag{37e}$$

Inspection of the above equations gives that z_t^1 and z_t^2 can both be thought of as measurements, since mathematically (37b) and (37c) possess the structure of measurement equations. The fact that there is a cross-correlation between the two noise processes w_t^l and w_t^n, since $Q_t^{ln} \neq 0$ in (19a), has to be taken care of. This can be accomplished using the Gram-Schmidt procedure to de-correlate the noise [17], [21]. Instead of w_t^l the

following can be used

$$\begin{aligned}\bar{w}_t^l &= w_t^l - \mathrm{E}[w_t^l(w_t^n)^T](\mathrm{E}[w_t^n(w_t^n)^T])^{-1}w_t^n \\ &= w_t^l - Q_t^{ln}(Q_t^n)^{-1}w_t^n, \end{aligned} \tag{38}$$

resulting in $\mathrm{E}[\bar{w}_t^l(w_t^n)^T] = 0$ and

$$\bar{Q}_t^l = \mathrm{E}[\bar{w}_t^l(\bar{w}_t^l)^T] = Q_t^l - Q_t^{ln}(Q_t^n)^{-1}Q_t^{ln}. \tag{39}$$

Using (37b) and (38), (37a) can be rewritten according to (G_t^n is assumed invertible. The case of a non-invertible G_t^n is treated in [5])

$$\begin{aligned}x_{t+1}^l &= A_t^l x_t^l + G_t^l[\bar{w}_t^l + Q_t^{ln}(Q_t^n)^{-1}(G_t^n)^{-1}(z_t^1 \\ &\quad - A_t^n x_t^l)] + f_t^l, \qquad (40) \\ &= \bar{A}_t^l x_t^l + G_t^l\bar{w}_t^l + G_t^l Q_t^{ln}(G_t^n Q_t^n)^{-1}z_t^1 + f_t^l, \qquad (41)\end{aligned}$$

where

$$\bar{A}_t^l = A_t^l - G_t^l Q_t^{ln}(G_t^n Q_t^n)^{-1}A_t^n. \tag{42}$$

The de-correlated system is

$$x_{t+1}^l = f_t^l + \bar{A}_t^l x_t^l + G_t^l Q_t^{ln}(G_t^n Q_t^n)^{-1}z_t^1 + G_t^l\bar{w}_t^l, \tag{43a}$$

$$z_t^1 = A_t^n x_t^l + G_t^n w_t^n, \tag{43b}$$

$$z_t^2 = C_t x_t^l + e_t, \tag{43c}$$

which is a linear system with Gaussian noise. Moreover, from (37d) and (37e) it can be seen that Z_t^1 and Z_t^2 are known if X_{t+1}^n and Y_t are known. The actual proof, using induction, of the theorem can now be started. At time zero; $p(x_0^l|X_0^n, Y_{-1}) = p(x_0^l|x_0^n) = \mathcal{N}(\bar{x}_0^l, \bar{P}_0)$. Now, assume that $p(x_t^l|X_t^n, Y_{t-1})$ is Gaussian at an arbitrary time, t.

The recursions are divided into three parts. First, the information available in the actual measurement, y_t, i.e., z_t^2 is used. Once the measurement update has been performed the estimates, $\hat{x}_{t|t}^l$ and $P_{t|t}$ are available. These can now be used to calculate the predictions of the nonlinear state, $\hat{x}_{t+1|t}^n$. These predictions will provide new information about the system. Second, this new information is incorporated by performing a second measurement update using the artificial measurement, z_t^1. Finally a time update, using the result from the second step, is performed.

Part 1: Assume that both $p(x_t^l|X_t^n, Y_{t-1}) = \mathcal{N}(\hat{x}_{t|t-1}^l, P_{t|t-1})$ and z_t^2 are available. This means that $p(x_t^l|X_t^n, Y_t)$ can be computed,

$$p(x_t^l|X_t^n, Y_t) = \frac{p(y_t|x_t^n, x_t^l)p(x_t^l|X_t^n, Y_{t-1})}{\int p(y_t|x_t^n, x_t^l)p(x_t^l|X_t^n, Y_{t-1})dx_t^l}. \tag{44}$$

Using the fact that the measurement noise and thereby $p(y_t|x_t^n, x_t^l)$ is Gaussian and the

Kalman filter [1] it can be seen that $p(x_t^l|X_t^n, Y_t) = \mathcal{N}(\hat{x}_{t|t}^l, P_{t|t})$ where

$$\hat{x}_{t|t}^l = \hat{x}_{t|t-1}^l + K_t(z_t^2 - C_t\hat{x}_{t|t-1}^l), \tag{45a}$$
$$P_{t|t} = P_{t|t-1} - K_t M_t K_t^T, \tag{45b}$$
$$K_t = P_{t|t-1}C_t^T M_t^{-1}, \tag{45c}$$
$$M_t = C_t P_{t|t-1} C_t^T + R_t. \tag{45d}$$

Part 2: At this stage z_t^1 becomes available. Use

$$p(x_t^l|X_{t+1}^n, Y_t) = \frac{p(x_{t+1}^n|x_t^n, x_t^l)p(x_t^l|X_t^n, Y_t)}{\int p(x_{t+1}^n|x_t^n, x_t^l)p(x_t^l|X_t^n, Y_t)dx_t^l} \tag{46}$$

analogously to part 1 $p(x_t^l|X_{t+1}^n, Y_t) = \mathcal{N}(\hat{x}_{t|t}^{l*}, P_{t|t}^*)$ where

$$\hat{x}_{t|t}^{l*} = \hat{x}_{t|t}^l + L_t(z_t^1 - A_t^n\hat{x}_{t|t}^l), \tag{47a}$$
$$P_{t|t}^* = P_{t|t} - L_t N_t^* L_t^T, \tag{47b}$$
$$L_t = P_{t|t}(A_t^n)^T(N_t^*)^{-1}, \tag{47c}$$
$$N_t^* = A_t^n P_{t|t}(A_t^n)^T + G_t^n Q_t^n (G_t^n)^T. \tag{47d}$$

Part 3: The final part is the time update, i.e., to compute

$$p(x_{t+1}^l|X_{t+1}^n, Y_t) = \int p(x_{t+1}^l|x_{t+1}^n, x_t^n, x_t^l)p(x_t^l|X_{t+1}^n, Y_t)dx_t^l. \tag{48}$$

Since the state noise is Gaussian this corresponds to the time update handled by the Kalman filter. Hence, $p(x_{t+1}^l|X_{t+1}^n, Y_t) = \mathcal{N}(\hat{x}_{t+1|t}^l, P_{t+1|t})$ where

$$\hat{x}_{t+1|t}^l = \bar{A}_t^l\hat{x}_{t|t}^l + G_t^l(Q_t^{ln})^T(G_t^n Q_t^n)^{-1}z_t^1 + f_t^l + L_t(z_t^1 - A_t^n\hat{x}_{t|t}^l), \tag{49a}$$
$$P_{t+1|t} = \bar{A}_t^l P_{t|t}(\bar{A}_t^l)^T + G_t^l\bar{Q}_t^l(G_t^l)^T - L_t N_t L_t^T, \tag{49b}$$
$$L_t = \bar{A}_t^l P_{t|t}(A_t^n)^T N_t^{-1}, \tag{49c}$$
$$N_t = A_t^n P_{t|t}(A_t^n)^T + G_t^n Q_t^n (G_t^n)^T. \tag{49d}$$

□

References

[1] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Information and system science series. Prentice Hall, Englewood Cliffs, NJ, USA, 1979.

[2] C. Andrieu and A. Doucet. Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society*, 64(4):827–836, 2002.

[3] C. Andrieu and S. J. Godsill. A particle filter for model based audio source separation. In *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation (ICA 2000)*, Helsinki, Finland, June 2000.

[4] Y. Bar-Shalom and X-R. Li. *Estimation and Tracking: Principles, Techniques, and Software*. Artech House, Norwood, MA, USA, 1993.

[5] N. Bergman. *Recursive Bayesian Estimation: Navigation and Tracking Applications*. Phd thesis No 579, Linköping Studies in Science and Technology, SE-581 83 Linköping, Sweden, May 1999.

[6] N. Bergman, L. Ljung, and F. Gustafsson. Terrain navigation using Bayesian statistics. *IEEE Control Systems Magazine*, 19(3):33–40, June 1999.

[7] G. Casella and C. P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

[8] R. Chen and J. S. Liu. Mixture Kalman filters. *Journal of the Royal Statistical Society*, 62(3):493–508, 2000.

[9] R. Chen, X. Wang, and J. S. Liu. Adaptive joint detection in flat-fading channels via mixture Kalman filtering. *IEEE Transactions on Information Theory*, 46(6):2079–2094, 2000.

[10] C. Dahlgren. Non-linear black box modelling of JAS 39 Gripen's radar altimeter. Master's Thesis No LiTH-ISY-EX-1958, Department of Electrical Engineering, Linköping University, Sweden, October 1998.

[11] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, New York, USA, 2001.

[12] A. Doucet, S. J. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[13] A. Doucet, N. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump Markov linear systems. Technical Report CUED/F-INFENG/TR 359, Signal Processing Group, Department of Engineering, University of Cambridge, Trupington street, CB2 1PZ Cambridge, 1999.

[14] A. Doucet, N. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624, 2001.

[15] P. Frykman. Applied particle filters in integrated aircraft navigation. Master's Thesis No LiTH-ISY-EX-3406, Department of Electrical Engineering, Linköping University, Sweden, April 2003.

[16] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings on Radar and Signal Processing*, volume 140, pages 107–113, 1993.

[17] F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons, New York, USA, 2000.

[18] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation and tracking. *IEEE Transactions on Signal Processing*, 50(2):425–437, February 2002.

[19] A. C. Harvey. *Forecasting, structural time series models and the Kalman filter*. Cambridge University Press, Cambridge, UK, 1989.

[20] A. H. Jazwinski. *Stochastic processes and filtering theory*. Mathematics in science and engineering. Academic Press, New York, USA, 1970.

[21] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Information and System Sciences Series. Prentice Hall, Upper Saddle River, NJ, USA, 2000.

[22] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82:35–45, 1960.

[23] R. Karlsson and F. Gustafsson. Particle filter for underwater navigation. In *Proceedings of the Statistical Signal Processing Workshop*, pages 509–512, St. Louis, MO, USA, September 2003.

[24] R. Karlsson, F. Gustafsson, and T. Karlsson. Particle filter and Cramér-Rao lower bound for underwater navigation. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, April 2003.

[25] R. Karlsson, T. Schön, and F. Gustafsson. Complexity analysis of the marginalized particle filter. Technical Report LiTH-ISY-R-2611, Department of Electrical Engineering, Linköping University, 2004.

[26] J. H. Kotecha and P. M. Djuric. Gaussian particle filtering. *IEEE Transactions on Signal Processing*, 51(10):2592–2601, 2003.

[27] J. H. Kotecha and P. M. Djuric. Gaussian sum particle filtering. *IEEE Transactions on Signal Processing*, 51(10):2602–2610, 2003.

[28] P. Li, R. Goodall, and V. Kadirkamanathan. Parameter estimation of railway vehicle dynamic model using Rao-Blackwellised particle filter. In *Proceedings of the European Control Conference*, Cambridge, UK, September 2003.

[29] X. R. Li and V. P. Jilkov. A survey of maneuvering target tracking-part III: Measurement models. In *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, pages 423–446, San Diego, USA, July 2001.

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

[31] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer, New York, USA, 2001.

[32] P.-J. Nordlund. *Sequential Monte Carlo Filters and Integrated Navigation.* Licentiate Thesis No 945, Department of Electrical Engineering, Linköping University, Sweden, 2002.

[33] J. Palmqvist. *On Integrity Monitoring of Integrated Navigation Systems.* Licentiate Thesis No 600, Department of Electrical Engineering, Linköping University, Sweden, 1997.

[34] M. K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, June 1999.

[35] T. Schön. *On Computational Methods for Nonlinear Estimation.* Licentiate Thesis No 1047, Department of Electrical Engineering, Linköping University, Sweden, October 2003.

[36] T. Schön and F. Gustafsson. Particle filters for system identification of state-space models linear in either parameters or states. In *Proceedings of the 13th IFAC Symposium on System Identification (SYSID)*, pages 1287–1292, Rotterdam, The Netherlands, September 2003.

[37] X. Wang, R. R. Chen, and D. Guo. Delayed-pilot sampling for mixture Kalman filter with application in fading channels. *IEEE Transactions on Signal Processing*, 50(2):241–254, February 2002.

Paper B

Marginalized Particle Filter for Accurate and Reliable Aircraft Navigation

Authors: Nordlund, P-J. and Gustafsson, F.

Edited version of the paper: P-J. Nordlund and F. Gustafsson. Marginalized particle filter for accurate and reliable terrain-aided navigation. *Accepted for publication in IEEE Transactions on Aerospace and Electronic Systems*, 2008.
http://www.control.isy.liu.se/research/reports/2008/2870.pdf

Preliminary version: Published as Technical Report LiTH-ISY-R-2870, Dept. of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden.

Marginalized Particle Filter for Accurate and Reliable Aircraft Navigation

Per-Johan Nordlund*, and Fredrik Gustafsson**

*Department of Decision Support
Saab Aerosystems
581 88 Linköping, Sweden
per-johan.nordlund@saabgroup.com

**Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
fredrik@isy.liu.se

Abstract

This paper details an approach to the integration of INS (Inertial Navigation System) and TAP (Terrain-Aided Positioning). The solution is characterized by a joint design of INS and TAP, meaning that the highly nonlinear TAP is not designed separately but jointly with the INS using one and the same filter.

The applied filter extends the theory of the MPF (Marginalized Particle Filter) given by [26]. The key idea with MPF is to estimate the nonlinear part using the particle filter and the part which is linear, conditionally upon the nonlinear part, is estimated using the Kalman filter. The extension lies in the possibility to deal with a third multi-modal part, where the discrete mode variable is also estimated jointly with the linear and nonlinear parts. Conditionally upon the mode and the nonlinear part, the resulting subsystem is linear and estimated using the Kalman filter.

Given the nonlinear motion equations which the INS uses to compute navigation data, the INS equations must be linearized for the MPF to work. A set of linearized equations is derived and the linearization errors are shown to be insignificant with respect to the final result. Simulations are performed and the result indicates near-optimal accuracy when compared to the Cramer-Rao lower bound.

1 Introduction

Accurate and reliable navigation systems have been identified as a critical enabling technology for enhanced aircraft capabilities in the coming 10-20 years. One reason is the foreseen increased use of unmanned aerial vehicles (UAVs). Following the introduction of UAVs the requirements on the navigation system (cost, size and performance) is

strengthened, and no stand-alone navigation sensor is capable of meeting them all. The solution is to blend the output from two or more navigation sensors to achieve an overall good enough accuracy and reliability.

Due to its reliability and short-term accuracy, even for flight conditions involving substantial maneuvering, inertial navigation systems (INS) are usually regarded as the primary source of navigation data. The major drawback with inertial navigation is that initialization and sensor errors cause computed quantities to drift. To stabilize the drift and ensure long-term accuracy the inertial navigation system is integrated with one or more aiding sources. The standard aiding source today is the global positioning system (GPS), see e.g. [7, 9, 3]. Although satellite navigation is seeing a widespread use, problems with the GPS such as reception limitation and interference increase the relevance of other aiding navigation sensors. One example is terrain-referenced navigation, or terrain-aided positioning (TAP). The principle is to measure terrain variations along the flight path and compare it to a database with stored terrain elevation for given positions. Although TAP does not suffer from the limitations applicable for GPS, there are other criteria which must be met. The distance to the ground needs to be within the operating range of the radar altimeter, you need a terrain elevation map over the area of interest and last but not least you need terrain variation along the flight path (which is not always the case e.g. when flying over water). Nevertheless, often the drawbacks of TAP are easier to accept than those for GPS, and the idea of using the terrain height [16], [4] or landmarks [11], [18] for positioning purposes has been around for quite some time now.

The challenge with TAP is to deal with its highly nonlinear, non-analytical characteristics. When facing a nonlinear estimation problem, a standard tool among practitioners is to apply the extended Kalman filter (EKF). Due to TAPs multi-modal character, corresponding to a measured terrain profile matching several profiles in the database, the EKF often fails. Better performance is obtained using grid-based methods, e.g. the point-mass filter [4], where the probability is discretized over the state space. This is possible due to the low dimensionality of TAP (either two or three dimensions if considering altitude besides horizontal position).

Traditionally, integrating TAP with INS has been performed using separate filters, one for TAP estimating position and another for estimating INS quantities using position from TAP as input [23]. Here, we use state-of-the-art joint design, meaning that we blend TAP and INS tightly in one and the same filter, see Figure 1.

Using this tight fusion technique means that we need to solve a nonlinear, high-dimensional problem. Here high-dimensional means that we have to consider not only position but also INS computed quantities such as velocity, attitude and heading. This rules out grid-based methods which, due to the computational load increasing exponentially with the dimension, are tractable only up to three dimensions. Simulation-based methods, such as the particle filter (PF) [12], have the promising feature of theoretically being independent of dimensionality. Simulation results indicate however that this is not the case in practice, although less dependent compared to the grid-based methods. Moreover, based on analysis and simulations [21] we know that a high performance INS with position error typically in the range of one nautical mile per hour (1.825 km/hr) is not very well suited for the particle filter. This has to do with the process noise being so small, making the particles cluster in state space and thereby increasing the discretization error of the particle filter. For the stand-alone particle filter to work on the blended

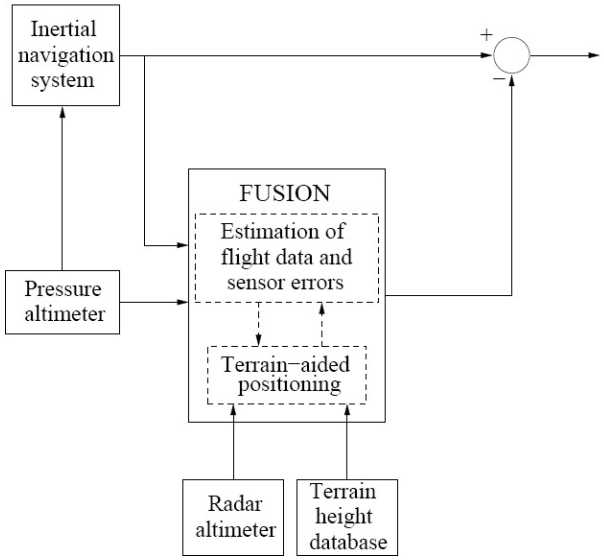

Figure 1: *Chosen configuration for blended INS/TAP. The pressure altimeter is used to stabilize INS vertical channel.*

INS/TAP described here the number of particles needed for filter convergence is simply too large to be computationally tractable [21].

However, we derive a set of linearized equations for the INS errors and we show that the linearization errors are small. Particularly this is true when the errors are kept small either using a high performance INS or by feeding the error back to the INS. Still the problem is highly nonlinear due to TAP, but now with a conditionally linear sub-structure corresponding to the additional INS related quantities. This means that conditionally upon position the INS related quantities can be estimated using the extended Kalman filter while position is estimated using the particle filter. The combined Kalman/particle filter is also known as the marginalized particle filter (MPF) [26] or the Rao-Blackwellized particle filter (RBPF) [2].

For the system to be able to provide accurate estimates of position we need an accurate estimate of altitude. One way forward and the one detailed in this paper is to use measurements from a radar altimeter (RA). The ground clearance measurements from the RA are however subject to a mode dependent error characteristic. The measurement error reflects e.g. whether there are a lot of trees on the ground or not. Conditionally upon the mode and nonlinear horizontal position, altitude is straightforwardly estimated by the Kalman filter. The MPF from [26] is here extended to account for this multi-modal character of the terrain elevation measurements. The extension consists of including estimation of the discrete RA measurement error mode as a third part in the joint filter design. The use of an airborne laser scanner (ALS) to measure ground clearance [8] is an interesting alternative to the radar altimeter. The accuracy and possibility to filter measurements which originate from tree reflections should yield significantly better estimation accuracy. On the other hand, the ALS has problem penetrating fog, rain and clouds which limits its applicability.

2 Outline

This paper begins by a derivation of INS error dynamics in Section 3. The INS nonlinear equations, for the sake of completeness given in Appendix A.1, are linearized and the resulting linear INS error equations are shown to accurately describe the aircraft dynamics. The non-analytical and highly nonlinear terrain-aided positioning system is introduced in Section 4. The derivation of the extended marginalized particle filter (MPF) is given in Section 5. The derivation consists of three lemmas, where each lemma provides result on how to estimate the linear, multi-modal and nonlinear parts respectively. The details on the algorithm for the blended INS/TAP system is given in 6 where we also analyze convergence properties of the filter. The algorithm is tested in a simulation study as described in 7. Finally conclusions are elaborated on in Section 8.

3 INS Error Dynamics

Collect all navigation variables, i.e. latitude L, longitude l, altitude h, velocity in north v_n, east v_e and down v_d directions and attitude and heading represented by a transfomation matrix from body to navigation frame C_b^n, in a state vector

$$z = \begin{bmatrix} L & l & h & v_n & v_e & v_d & \text{vec}(C_b^n) \end{bmatrix}^T. \tag{1}$$

The input variables, i.e. accelerations f^b and angular rates ω^b, are collected in the vector

$$w = \begin{bmatrix} f_x^b & f_y^b & f_z^b & \omega_x^b & \omega_y^b & \omega_z^b \end{bmatrix}^T. \tag{2}$$

For details regarding the navigation and input variables see Appendix A.1. The state dynamics, given by (77), (79) and (85) in Appendix A.1, can compactly be written according to

$$\dot{z} = f(z, w). \tag{3}$$

Denote the corresponding INS state and input vectors by z^{ins} and w^{ins}. Due to initialization and sensor errors the state vector computed by the INS will differ from the true state vector. Define the INS state and sensor errors according to

$$x = z - z^{\text{ins}}, \quad u = w - w^{\text{ins}}. \tag{4}$$

Combining (3)–(4) we can write the error dynamics as

$$\dot{x} = \dot{z} - \dot{z}^{\text{ins}} = f(z, w) - f(z - x, w - u). \tag{5}$$

The goal is to provide a set of linearized equations describing the INS error dynamics,

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + \Delta, \tag{6}$$

such that Δ representing the linearization error is small. Below we derive x, u, A and B in (6) such that

$$\frac{|\Delta_i|}{|[Ax]_i|} < 0.01, \quad i = 1, \ldots, n, \quad \forall x \in \text{Table 1} \tag{7}$$

where n is the number of states in x. We will show that x and u according to

$$x = \begin{bmatrix} \tilde{L} & \tilde{l} & \tilde{h} & \tilde{v}_n & \tilde{v}_e & \gamma_n & \gamma_e & \gamma_d & b_x^a & b_y^a \end{bmatrix}^T, \tag{8}$$

$$u = \begin{bmatrix} u^h & u_x^a & u_y^a & u_x^\gamma & u_y^\gamma & u_z^\gamma & u_x^b & u_y^b \end{bmatrix}^T. \tag{9}$$

and A and B according to Appendix A.2 are adequate. The vector x in (8) is extended with two states for accelerometer biases b_x^a and b_y^a compared to (4). Moreover, u in (9) consists of white noise components including process noise for the accelerometer biases u_x^b and u_y^b. Note that there is no accelerometer bias or noise along body frame $z-$axis. Instead white noise u^h enters the equation for altitude error $\tilde{h}$ directly. The reason for this is that the INS is supported by a pressure altimeter, see Figure 1, which compensates for any drift in the vertical channel.

Table 1: *Ranges on navigation data.*

$\lvert L\rvert \leq 70$ deg	$\lvert\tilde{L}\rvert \leq 2000/r_0$ rad
$h \leq 5000$ m	$\lvert\tilde{h}\rvert \leq 100$ m
$\lvert v_{\text{tot}}\rvert \leq 200$ m/s	$\lvert\tilde{v}_{\text{tot}}\rvert \leq 2$ m/s
$\lvert\gamma_{n,e}\rvert \leq 5 \cdot 10^{-4}$ rad	$\lvert\gamma_d\rvert \leq 1 \cdot 10^{-3}$ rad
$\lvert f_{x,y,z}^b\rvert \leq 20$ m/s^2	$\lvert b_{x,y}^a\rvert \leq 5 \cdot 10^{-3}$ m/s^2

From (77) and (5) the expressions for the latitude and longitude errors become

$$\begin{aligned} \dot{\tilde{L}} &= \frac{v_n}{r_L(L)+h} - \frac{v_n - \tilde{v}_n}{r_L(L-\tilde{L})+h-\tilde{h}}, \\ \dot{\tilde{l}} &= \frac{v_e \cos^{-1} L}{r_l(L)+h} - \frac{(v_e - \tilde{v}_e)\cos^{-1}(L-\tilde{L})}{r_l(L-\tilde{L})+h-\tilde{h}}. \end{aligned} \tag{10}$$

Apply Taylor expansion on (10) around h, ε^2 and L and we can rewrite the equations according to

$$\begin{aligned} \dot{\tilde{L}} &= \frac{1}{r_0}\tilde{v}_n + \Delta_{\tilde{L}} \\ \dot{\tilde{l}} &= \frac{v_e \sin L}{r_0 \cos^2 L}\tilde{L} + \frac{1}{r_0 \cos L}\tilde{v}_e + \Delta_{\tilde{l}}. \end{aligned} \tag{11}$$

Inserting values on the errors involved from Table 1, the magnitude on $\Delta_{\tilde{L}}$ and $\Delta_{\tilde{l}}$ in (11) is

$$\begin{aligned} \frac{|\Delta_{\tilde{L}}|}{|[Ax]_{\tilde{L}}|} &< \varepsilon^2 + \frac{v_n\tilde{h}}{\tilde{v}_n r_0} < 9 \cdot 10^{-3}, \\ \frac{|\Delta_{\tilde{l}}|}{|[Ax]_{\tilde{l}}|} &< \frac{\tilde{v}_e\varepsilon^2 + v_e\tilde{h}/r_0}{v_e\tilde{L}\tan L + \tilde{v}_e} < 8 \cdot 10^{-3}. \end{aligned} \tag{12}$$

From (77) we see that the equation for the altitude error $\tilde{h}$ becomes

$$\dot{\tilde{h}} = -\tilde{v}_d. \tag{13}$$

The INS is unstable in the vertical channel with a time constant of approximately 10 minutes [20]. For an operational INS the vertical channel must therefore be stabilized. Typically this is done using a pressure sensor. A simple and reasonable assumption is that the pressure sensor error drifts according to a random walk process, where the driving noise is described by the u^h-component in (9). Moreover, we assume that the stabilization works through an altitude filter such that the INS altitude error follows the pressure altitude error, resulting in the equation

$$\dot{\tilde{h}} \approx u^h. \tag{14}$$

Applying (5) on (79) gives the velocity error equation

$$\begin{aligned}\dot{\tilde{v}}^n = \tilde{C}_b^n f^b + (C_b^n - \tilde{C}_b^n)\tilde{a}^b - (\tilde{\Omega}_{en}^n + 2\tilde{\Omega}_{ie}^n)v^n - \\ (\Omega_{en}^n - \tilde{\Omega}_{en}^n + 2(\Omega_{ie}^n - \tilde{\Omega}_{ie}^n))\tilde{v}^n + \tilde{g}^n,\end{aligned} \tag{15}$$

with $f^{b,\text{ins}} = f^b - \tilde{a}^b$ and $\tilde{g}^n \approx \begin{bmatrix}0 & 0 & \tilde{g}_d\end{bmatrix}^T$. Taylor expansion on $\tilde{\omega}_{en}^n$ using (81) and (5) around ε^2, h and L and on $\tilde{\omega}_{ie}^n$ around L provides

$$\begin{aligned}\tilde{\omega}_{en}^n + 2\tilde{\omega}_{ie}^n &= \hat{\tilde{\omega}}_{en}^n + 2\hat{\tilde{\omega}}_{ie}^n + \Delta_{\delta\omega}, \\ \hat{\tilde{\omega}}_{en}^n + 2\hat{\tilde{\omega}}_{ie}^n &= \begin{bmatrix} \frac{\tilde{v}_e}{r_0} - 2\omega_{ie}\tilde{L}\sin L \\ -\frac{\tilde{v}_n}{r_0} \\ -\frac{v_e\tilde{L}}{r_0\cos^2 L} - \frac{\tilde{v}_e\tan L}{r_0} - 2\omega_{ie}\tilde{L}\cos L \end{bmatrix}.\end{aligned} \tag{16}$$

We can also simplify $\omega_{en}^n - \tilde{\omega}_{en}^n + 2(\omega_{ie}^n - \tilde{\omega}_{ie}^n)$ in (15) according to

$$\begin{aligned}\omega_{en}^n - \tilde{\omega}_{en}^n + 2(\omega_{ie}^n - \tilde{\omega}_{ie}^n) &= \hat{\omega}_{en}^n + 2\omega_{ie}^n + \Delta_\omega, \\ \hat{\omega}_{en}^n + 2\omega_{ie}^n &= \begin{bmatrix} \frac{v_e}{r_0} + 2\omega_{ie}\cos L \\ -\frac{v_n}{r_0} \\ -\frac{v_e\tan L}{r_0} - 2\omega_{ie}\sin L \end{bmatrix}.\end{aligned} \tag{17}$$

Moreover, define a small-angle transformation $\gamma^n = \begin{bmatrix}\gamma_n & \gamma_e & \gamma_d\end{bmatrix}^T$, in skew-symmetric matrix form denoted by Γ^n, through

$$\begin{aligned}\tilde{C}_b^n &= C_b^n - C_b^{n,\text{ins}} = C_b^n - C_n^{n,\text{ins}}C_b^n \\ &= (I - C_n^{n,\text{ins}})C_b^n = \Gamma^n C_b^n + \Delta_{\tilde{C}_b^n}.\end{aligned} \tag{18}$$

The small-angle transformation describes a rotation of the navigation frame computed by the INS relative to the true navigation frame. The rest term $\Delta_{\tilde{C}_b^n}$ consists of second and higher order terms of γ^n which are obtained after Taylor expansion of $I - C_n^{n,\text{ins}}$. Applying the approximations on the velocity error equation in (15), together with $f^n = C_b^n f^b$, $\Gamma^n f^n = -F^n\gamma^n$ and $\Omega^n v^n = -V^n\omega^n$, yields

$$\begin{aligned}\dot{\tilde{v}}^n = -F^n\gamma^n + C_b^n\tilde{a}^b + \tilde{V}^n(\hat{\omega}_{en}^n + 2\omega_{ie}^n) + \\ V^n(\hat{\tilde{\omega}}_{en}^n + 2\hat{\tilde{\omega}}_{ie}^n) + \tilde{g}^n + \Delta_{\tilde{v}^n}.\end{aligned} \tag{19}$$

The error introduced in $\tilde{v}_n$ and $\tilde{v}_e$ when going from (15) to (19) is upper limited by

$$\begin{aligned}\|\Delta_{\tilde{v}_{n,e}}\|_\infty \leq \|\Delta_{\tilde{C}_b^n} f^b\|_\infty + \|\Gamma^n \tilde{a}^b\|_\infty \\ + \|\tilde{V}^n \Delta_\omega\|_\infty + \|V^n \Delta_{\tilde{\omega}}\|_\infty.\end{aligned} \tag{20}$$

Using (16) and (17) we have

$$\begin{aligned}\|\Delta_{\tilde{C}_b^n} f^b\|_\infty &\leq 2\gamma_d^2 \max(f_x^b, f_y^b, f_z^b) \\ \|\Gamma^n \tilde{a}^b\|_\infty &\leq 2\gamma_d \max(b_x^a, b_y^a, b_z^a), \\ \|\tilde{V}^n \Delta_\omega\|_\infty &\leq \tilde{v}\left(\frac{\tilde{v}\tan L}{r_0} + \frac{\varepsilon^2 v L \tan L}{r_0} + 2\omega_{ie}\tilde{L}\right), \\ \|V^n \Delta_{\tilde{\omega}}\|_\infty &\leq v\left(\frac{\tilde{v}\varepsilon^2 \tan L}{r_0} + \frac{v\tilde{h}L\tan L}{r_0^2} + 2\omega_{ie}\tilde{L}^2\right),\end{aligned} \tag{21}$$

with $v = \max(v_n, v_e)$. Using values from Table 1 we obtain

$$\begin{aligned}\frac{\|\Delta_{\tilde{v}_{n,e}}\|_\infty}{|[Ax]_{\tilde{v}_{n,e}}|} &\leq \frac{(40+10+4+2)\cdot 10^{-6}}{(10000+5000+300+100)\cdot 10^{-6}} \\ &\leq 4 \cdot 10^{-3}.\end{aligned} \tag{22}$$

Finally, using (85) and (18) it is straightforward to show [20] that the linearized equation for γ^n is given by

$$\begin{aligned}\dot{\gamma}^n &= C_b^n \tilde{\omega}_{ib}^b - \tilde{\omega}_{in}^n - \Omega_{in}^n \gamma^n + \Delta_{\gamma^n}, \\ \|\Delta_{\gamma^n}\|_\infty &\leq 2\gamma_d^2\left(\frac{v\tan L}{r_0} + \omega_{ie}\right),\end{aligned} \tag{23}$$

where $v = \max(v_n, v_e)$ and $\tilde{v} = \max(\tilde{v}_n, \tilde{v}_e)$. The rest term Δ_{γ^n} consists of second and higher order terms of γ^n. The error is therefore upper limited by

$$\frac{\|\Delta_{\gamma^n}\|_\infty}{|[Ax]_{\Delta_{\gamma^n}}|} \leq \frac{0.4\cdot 10^{-9}}{(1000+500+100)\cdot 10^{-9}} < 3\cdot 10^{-4}. \tag{24}$$

The error characteristics for the accelerometers and rate gyros are in general involved, see e.g. [19]. The easiest, but for the application adequate, way to model the accelerometer errors is to use a slowly varying offset and white noise. Normally there also exist offsets in the rate gyros, but these are for the application here considered small and therefore neglected. Note however, the algorithm is readily modifiable to include the influence of gyro drift in cases where the gyro offset is larger or the time interval is longer. The accelerometer offset, or bias, can with good accuracy be modelled as a first order Gauss-Markov process,

$$\dot{b}^a = -\frac{1}{\tau} b^a + u^b \approx u^b. \tag{25}$$

The last approximation is valid because the time constant τ is usually rather large. We will here incorporate accelerometer biases acting only in the x- and y-directions in body

frame. This is an implication from the assumption that the INS altitude error follows the pressure altitude, meaning that any z-accelerometer bias is compensated for through the stabilization of the vertical channel. This is true for situations where roll and pitch angles are close to zero. During a turn, climb and/or dive the altitude filter is effected by x- and/or $y-$ accelerometer biases. Here we assume turns, climbs and/or dives are rare such that we can neglect this effect.

To be able to apply our discrete time filter (6) has to be discretized

$$x_{t+1} = F_t x_t + G_t u_t. \tag{26}$$

For a small sampling period T_s the Euler approximation provides

$$\begin{aligned} F_t &= I + T_s A_t, \\ G_t &= T_s(I + \frac{T_s}{2} A_t) B_t, \\ \mathbb{E}[u_t u_t^T] &= Q_t / T_s, \end{aligned} \tag{27}$$

where $A(t) = A_t$, $B(t) = B_t$ and $\mathbb{E}[u(t)u^T(t)] = Q(t) = Q_t$ are considered constant during the sampling period.

4 Terrain-Aided Positioning

The idea behind terrain-aided positioning is to use the terrain height profile, obtained by projecting the path of the aircraft onto the ground. The INS computed altitude provides a measurement of altitude above mean-sea level. At the same time the ground clearance, i.e. the distance between the aircraft and the ground, is measured using a radar altimeter. The difference between these two measurements provides a measurement on the terrain height at the location where the measurement was performed. A number of such measurements build up a measured terrain height profile. The aircraft carries a terrain elevation database where the terrain height is stored as a function of sampled horizontal position. The measured terrain height profile is compared with all possible profiles obtained from the database. The database profile which resembles the measured profile the most is selected, and thereby determines the aircraft's position.

The equation for terrain-aided positioning is

$$y_t = h_t^{\text{ins}} - h_t^{\text{ra}} = h(L_t, l_t) - \tilde{h}_t + e_t, \tag{28}$$

where y_t is measured terrain height and $h(\cdot)$ is the terrain height given by the database as a function of horizontal position, i.e. latitude and longitude. The term $\tilde{h}_t$ is the INS altitude error. Moreover, e_t is the measurement noise, having a probability density which here is given by

$$p(e_t) = \sum_{\lambda_t=1}^{2} \Pr(\lambda_t) \mathcal{N}(m_t^{(\lambda_t)}, R_t^{(\lambda_t)}), \tag{29}$$

i.e. a Gaussian mixture with two modes. The first mode ($\lambda_t = 1$) represents the case where the radar altimeter beam hits the ground and thereby reflecting the true ground

clearance. The second mode ($\lambda_t = 2$) models the case where the beam hits a tree top, giving a measurement of the ground clearance which is too small. The probability for each of the two events is $\Pr(\lambda_t = 1)$ and $\Pr(\lambda_t = 2)$ respectively. The radar altimeter is a pulsed system operating at 4.3 GHz which makes it sensitive to reflections from e.g. trees. Together with a wide beam lobe (≈ 50 deg) the radar altimeter normally measures the closest distance to the ground or any obstacle, even during moderate roll and pitch angles. To eliminate roll and pitch dependent errors the measurements from the radar altimeter are not used when roll or pitch angle is larger than 25 deg.

A simple way of avoiding the mode dependent error characteristics is to approximate the probability density in (29) with a single Gaussian. However, it is shown in [15] that the gain when taking advantage of multi-modal characteristics can be significant. Simulations show that e.g. horizontal position is estimated with approximately 70% better accuracy.

5 The Marginalized Particle Filter

The main idea of the particle filter is to discretize the posterior probability density for the state x_t according to

$$p(x_t|Y_t) \approx \sum_{i=1}^{N} \bar{w}_t^{(i)} \delta_{x_t^{(i)}}(x_t), \tag{30}$$

where δ is the delta-Dirac function and $Y_t = \{y_0, \ldots, y_t\}$ is the stacked vector of measurements. The weights $\bar{w}_t^{(i)}$, where $\sum_{i=1}^{N} \bar{w}_t^{(i)} = 1$, together with the particles $x_t^{(i)}$ are such that they together yield a set of samples approximately drawn from the posterior probability density. Theoretically we can solve almost any estimation problem using the particle filter, as long as the number of particles N is high enough.

In many cases the underlying motion model has structures which can be exploited for the purpose of decreasing N and thereby the computational load. Consider a state-space model which can be written on the form

$$x_{t+1}^{\mathrm{n}} = f_t^{\mathrm{n}}(x_t^{\mathrm{n}}) \qquad + F_t^{\mathrm{n}} x_t^{\mathrm{l}} + G_t^{\mathrm{n}} u_t^{\mathrm{n}}, \tag{31a}$$

$$x_{t+1}^{\mathrm{d}} = f_t^{\mathrm{d}}(x_t^{\mathrm{n}}) + F_t^{\mathrm{d}} x_t^{\mathrm{d}} \qquad + G_t^{\mathrm{d}} u_t^{\mathrm{d}}, \tag{31b}$$

$$x_{t+1}^{\mathrm{l}} = f_t^{\mathrm{l}}(x_t^{\mathrm{n}}) \qquad + F_t^{\mathrm{l}} x_t^{\mathrm{l}} + G_t^{\mathrm{l}} u_t^{\mathrm{l}}, \tag{31c}$$

$$y_t = h_t(x_t^{\mathrm{n}}) + H_t x_t^{\mathrm{d}} \qquad + e_t(\lambda_t), \tag{31d}$$

where $x_t = \begin{bmatrix}(x_t^{\mathrm{n}})^T & (x_t^{\mathrm{d}})^T & (x_t^{\mathrm{l}})^T\end{bmatrix}^T$. The superscripts n, d and l denote which part of the state vector has a nonlinear, discrete and linear structure respectively. Note that spaces in (31) are used to emphasize what parts of the state vector are affected by other parts and the measurements. This is important for the results derived below. Assume that the process noise is Gaussian distributed according to

$$u_t = \begin{bmatrix} u_t^{\mathrm{n}} \\ u_t^{\mathrm{d}} \\ u_t^{\mathrm{l}} \end{bmatrix} \sim \mathcal{N}(0, Q_t), \quad Q_t = \begin{bmatrix} Q_t^{\mathrm{n}} & 0 & 0 \\ 0 & Q_t^{\mathrm{d}} & 0 \\ 0 & 0 & Q_t^{\mathrm{l}} \end{bmatrix}. \tag{32}$$

See [26] on how to deal with a mutually correlated process noise. Also assume that x_0^{d} and x_0^{l} are Gaussian distributed i.e.

$$x_0^{\mathrm{d}} \sim \mathcal{N}(0, P_0^{\mathrm{d}}), \quad x_0^{\mathrm{l}} \sim \mathcal{N}(0, P_0^{\mathrm{l}}). \tag{33}$$

The measurement noise e_t is a sum of M Gaussians according to

$$e_t \sim \sum_{\lambda_t=1}^{M} \Pr(\lambda_t)\mathcal{N}(m_t^{(\lambda_t)}, R_t^{(\lambda_t)}), \tag{34}$$

with mode transition probabilities

$$\pi_{\lambda_{t-1}}^{\lambda_t} = \Pr(\lambda_t|\lambda_{t-1}), \quad \lambda_t, \lambda_{t-1} = 1, \ldots, M. \tag{35}$$

The aim is to recursively estimate the probability density function (pdf) for x_t given all available measurements Y_t. The pdf is then used to compute an estimate of x_t, here the mean value, and the corresponding covariance of the estimate. The direct approach is to apply the particle filter. However, for the class of systems described by (31) there exists a more efficient way. Consider the probability density $p(x_t^{\mathrm{l}}, x_t^{\mathrm{d}}, X_t^{\mathrm{n}}|Y_t)$, where $X_t^{\mathrm{n}} = \{x_0^{\mathrm{n}}, \ldots, x_t^{\mathrm{n}}\}$ is the stacked vector of state history. This pdf can be factorized using Bayes´ rule according to

$$\begin{aligned} &p(x_t^{\mathrm{l}}, x_t^{\mathrm{d}}, X_t^{\mathrm{n}}|Y_t) = \\ &p(x_t^{\mathrm{l}}|x_t^{\mathrm{d}}, X_t^{\mathrm{n}}, Y_t)p(x_t^{\mathrm{d}}|X_t^{\mathrm{n}}, Y_t)p(X_t^{\mathrm{n}}|Y_t) = \\ &p(x_t^{\mathrm{l}}|X_t^{\mathrm{n}}, Y_t)p(x_t^{\mathrm{d}}|X_t^{\mathrm{n}}, Y_t)p(X_t^{\mathrm{n}}|Y_t). \end{aligned} \tag{36}$$

Assume for now that we have an estimate based on the particle filter of $p(X_t^{\mathrm{n}}|Y_t)$ according to

$$p(X_t^{\mathrm{n}}|Y_t) \approx \sum_{i=1}^{N} \bar{w}_t^{(i)} \delta_{X_t^{\mathrm{n},(i)}}(X_t^{\mathrm{n}}). \tag{37}$$

From (37) we have the probability density

$$p(x_t^{\mathrm{n}}|Y_t) \approx \sum_{i=1}^{N} \bar{w}_t^{(i)} \delta_{x_t^{\mathrm{n},(i)}}(x_t^{\mathrm{n}}), \tag{38}$$

by extracting x_t^{n} from X_t^{n}. Moreover, combining (36) and (37) we have estimates of

$$\begin{aligned} p(x_t^{\mathrm{l}}|Y_t) &= \int p(x_t^{\mathrm{l}}|X_t^{\mathrm{n}}, Y_t)p(X_t^{\mathrm{n}}|Y_t)dX_t^{\mathrm{n}} \\ &\approx \sum_{i=1}^{N} \bar{w}_t^{(i)} p(x_t^{\mathrm{l}}|X_t^{\mathrm{n},(i)}, Y_t), \\ p(x_t^{\mathrm{d}}|Y_t) &= \int p(x_t^{\mathrm{d}}|X_t^{\mathrm{n}}, Y_t)p(X_t^{\mathrm{n}}|Y_t)dX_t^{\mathrm{n}} \\ &\approx \sum_{i=1}^{N} \bar{w}_t^{(i)} p(x_t^{\mathrm{d}}|X_t^{\mathrm{n},(i)}, Y_t). \end{aligned} \tag{39}$$

In the forthcoming three Lemmas we derive expressions for how to recursively compute $p(X_t^{\mathrm{n}}|Y_t)$, $p(x_t^{\mathrm{l}}|X_t^{\mathrm{n}}, Y_t)$ and $p(x_t^{\mathrm{d}}|X_t^{\mathrm{n}}, Y_t)$, i.e.

Lemma 1: $p(x_t^{\mathrm{l}}|X_t^{\mathrm{n}}, Y_t)$ by the Kalman filter,
Lemma 2: $p(x_t^{\mathrm{d}}|X_t^{\mathrm{n}}, Y_t)$ by a bank of M^{t+1} Kalman filters,
Lemma 3: $p(X_t^{\mathrm{n}}|Y_t)$ by the particle filter.

The recursions are such that we do not need knowledge of the state history X_t^{n} but only x_{t-1}^{n} and x_t^{n}. The derived expressions are then used together with (38) and (39) to obtain estimates of the posterior pdfs of x_t^{l}, x_t^{d} and x_t^{n}.

Lemma 1 (Conditionally linear single Gaussian)
For the state-space model (31)*, with the assumptions according to* (32)–(33)*, we have that*

$$p(x_t^{\mathrm{l}}|X_t^{\mathrm{n}}, Y_t) = \mathcal{N}(\hat{x}_{t|t}^{\mathrm{l}}, P_{t|t}^{\mathrm{l}}), \tag{40a}$$
$$p(x_{t+1}^{\mathrm{l}}|X_{t+1}^{\mathrm{n}}, Y_t) = \mathcal{N}(\hat{x}_{t+1|t}^{\mathrm{l}}, P_{t+1|t}^{\mathrm{l}}), \tag{40b}$$

where

$$\hat{x}_{t|t}^{\mathrm{l}} = \hat{x}_{t|t-1}^{\mathrm{l}}, \quad P_{t|t}^{\mathrm{l}} = P_{t|t-1}^{\mathrm{l}}, \tag{41}$$

and

$$\begin{aligned}
\hat{x}_{t+1|t}^{\mathrm{l}} &= (F_t^{\mathrm{l}} - K_{p,t}F_t^{\mathrm{n}})\hat{x}_{t|t}^{\mathrm{l}} + \\
&\quad K_{p,t}(x_{t+1}^{\mathrm{n}} - f_t^{\mathrm{n}}(x_t^{\mathrm{n}})) + f_t^{\mathrm{l}}(x_t^{\mathrm{n}}), \\
P_{t+1|t}^{\mathrm{l}} &= F_t^{\mathrm{l}} P_{t|t}^{\mathrm{l}} (F_t^{\mathrm{l}})^T + G_t^{\mathrm{l}} Q_t^{\mathrm{l}} (G_t^{\mathrm{l}})^T - K_{p,t} S_{p,t} K_{p,t}^T, \\
K_{p,t} &= F_t^{\mathrm{l}} P_{t|t}^{\mathrm{l}} (F_t^{\mathrm{n}})^T S_{p,t}^{-1}, \\
S_{p,t} &= G_t^{\mathrm{n}} Q_t^{\mathrm{n}} (G_t^{\mathrm{n}})^T + F_t^{\mathrm{n}} P_{t|t}^{\mathrm{l}} (F_t^{\mathrm{n}})^T.
\end{aligned} \tag{42}$$

Proof: Conditionally upon X_t^{n}, x_t^{l} is independent of Y_t and thereby unaffected by the multi-modal noise e_t given by (34). The result then follows immediately from [26]. □

In practice the above means that we can estimate $p(x_t^{\mathrm{l}}|Y_t)$ as the weighted sum of N Kalman filters applied to each sequence of $\{X_t^{\mathrm{n},(i)}\}_{i=1}^N$.

Lemma 2 (Conditionally linear multi-modal Gaussian)
For the state-space model (31)*, with the assumptions according to* (32)–(35) *and* $\Lambda_t = \{\lambda_0, \ldots, \lambda_t\}$*, we have that*

$$p(x_t^{\mathrm{d}}|X_t^{\mathrm{n}}, Y_t) = \sum_{\Lambda_t} \bar{\alpha}_t^{(\Lambda_t)} \mathcal{N}(\hat{x}_{t|t}^{\mathrm{d},(\Lambda_t)}, P_{t|t}^{\mathrm{d},(\Lambda_t)}), \tag{43a}$$
$$p(x_{t+1}^{\mathrm{d}}|X_t^{\mathrm{n}}, Y_t) = \sum_{\Lambda_t} \bar{\alpha}_t^{(\Lambda_t)} \mathcal{N}(\hat{x}_{t+1|t}^{\mathrm{d},(\Lambda_t)}, P_{t+1|t}^{\mathrm{d},(\Lambda_t)}), \tag{43b}$$

where

$$\begin{aligned}\hat{x}^{\mathrm{d},(\Lambda_t)}_{t|t} &= \hat{x}^{\mathrm{d},(\Lambda_{t-1})}_{t|t-1} + \\ &\quad P^{\mathrm{d},(\Lambda_{t-1})}_{t|t-1} H_t^T (S^{(\Lambda_t)}_{f,t})^{-1}(y_t - \hat{y}^{(\Lambda_t)}_{t|t-1}),\end{aligned} \tag{44a}$$

$$\begin{aligned}P^{\mathrm{d},(\Lambda_t)}_{t|t} &= P^{\mathrm{d},(\Lambda_{t-1})}_{t|t-1} - \\ &\quad P^{\mathrm{d},(\Lambda_{t-1})}_{t|t-1} H_t^T (S^{(\Lambda_t)}_{f,t})^{-1} H_t (P^{\mathrm{d},(\Lambda_{t-1})}_{t|t-1})^T,\end{aligned} \tag{44b}$$

$$\hat{y}^{(\Lambda_t)}_{t|t-1} = h_t(x^{\mathrm{n}}_t) - H_t \hat{x}^{\mathrm{d},(\Lambda_{t-1})}_{t|t-1} - m^{(\lambda_t)}_t, \tag{44c}$$

$$S^{(\Lambda_t)}_{f,t} = R^{(\lambda_t)}_t + H_t P^{\mathrm{d},(\Lambda_{t-1})}_{t|t-1} H_t^T, \tag{44d}$$

$$\begin{aligned}\hat{x}^{\mathrm{d},(\Lambda_t)}_{t+1|t} &= f^{\mathrm{d}}_t(x^{\mathrm{n}}_t) + F^{\mathrm{d}}_t \hat{x}^{\mathrm{d},(\Lambda_t)}_{t|t}, \\ P^{\mathrm{d},(\Lambda_t)}_{t+1|t} &= F^{\mathrm{d}}_t P^{\mathrm{d},(\Lambda_t)}_{t|t} (F^{\mathrm{d}}_t)^T + G^{\mathrm{d}}_t Q^{\mathrm{d}}_t (G^{\mathrm{d}}_t)^T,\end{aligned} \tag{45}$$

and

$$\begin{aligned}\alpha^{(\Lambda_t)}_t &= \mathcal{N}(\hat{y}^{(\Lambda_t)}_{t|t-1}, S^{(\Lambda_t)}_{f,t}) \pi^{\lambda_t}_{\lambda_{t-1}} \bar{\alpha}^{(\Lambda_{t-1})}_{t-1}, \\ \bar{\alpha}^{(\Lambda_t)}_t &= \frac{\alpha^{(\Lambda_t)}_t}{\sum_{\Lambda_t} \alpha^{(\Lambda_t)}_t}\end{aligned} \tag{46}$$

Proof: The probability $p(x^{\mathrm{d}}_t | X^{\mathrm{n}}_t, Y_t)$ can be written according to

$$p(x^{\mathrm{d}}_t | X^{\mathrm{n}}_t, Y_t) = \sum_{\Lambda_t} p(x^{\mathrm{d}}_t | X^{\mathrm{n}}_t, Y_t, \Lambda_t) \Pr(\Lambda_t | X^{\mathrm{n}}_t, Y_t). \tag{47}$$

Conditionally upon λ_t, e_t is a single Gaussian. Together with $\bar{\alpha}^{(\Lambda_t)}_t = \Pr(\Lambda_t | X^{\mathrm{n}}_t, Y_t)$ the result for (43)–(45) follows from [26]. Using Bayes´ rule repeatedly and the principle of induction, the probability $\Pr(\Lambda_t | X^{\mathrm{n}}_t, Y_t)$ is recursively given by

$$\begin{aligned}&\Pr(\Lambda_t | X^{\mathrm{n}}_t, Y_t) = \\ &\frac{p(y_t | X^{\mathrm{n}}_t, Y_{t-1}, \Lambda_t) \Pr(\Lambda_t | X^{\mathrm{n}}_t, Y_{t-1})}{p(y_t | X^{\mathrm{n}}_t, Y_{t-1})} = \\ &\frac{p(y_t | X^{\mathrm{n}}_t, Y_{t-1}, \Lambda_t) \Pr(\lambda_t | \lambda_{t-1}) \Pr(\Lambda_{t-1} | X^{\mathrm{n}}_{t-1}, Y_{t-1})}{\sum_{\Lambda_t} p(y_t | X^{\mathrm{n}}_t, Y_{t-1}, \Lambda_t) \Pr(\lambda_t | \lambda_{t-1}) \Pr(\Lambda_{t-1} | X^{\mathrm{n}}_{t-1}, Y_{t-1})} = \\ &\frac{p(y_t | X^{\mathrm{n}}_t, Y_{t-1}, \Lambda_t) \Pr(\lambda_t | \lambda_{t-1}) \bar{\alpha}^{(\Lambda_{t-1})}_{t-1}}{\sum_{\Lambda_t} p(y_t | X^{\mathrm{n}}_t, Y_{t-1}, \Lambda_t) \Pr(\lambda_t | \lambda_{t-1}) \bar{\alpha}^{(\Lambda_{t-1})}_{t-1}}.\end{aligned} \tag{48}$$

From [26] we know that $p(y_t | X^{\mathrm{n}}_t, Y_{t-1}, \Lambda_t) = \mathcal{N}(\hat{y}^{(\Lambda_t)}_{t|t-1}, S^{(\Lambda_t)}_{f,t})$. Together with the mode transition probability $\pi^{\lambda_t}_{\lambda_{t-1}} = \Pr(\lambda_t | \lambda_{t-1})$ the formulas in (46) follows. □

In practice this means that we have to apply one Kalman filter for each sequence of particles and each sequence of modes. The number of possible mode sequences increases

exponentially with time and must somehow be limited. One way is to include the estimate of the mode sequence in the particle filter, which automatically limits the number such that only the most probable mode sequences survive. Another way is to merge mode sequences which are identical from $t - L$ up to and including t, so as to keep the number constant ($= M^L$), using e.g. the generalized pseudo-Bayesian (GPB) or interacting multiple model (IMM) filter [14, 5, 22].

Lemma 3 (Gaussian distributed likelihood and prior)
The probability $p(X_t^{\mathrm{n}}|Y_t)$ is recursively given by

$$p(X_t^{\mathrm{n}}|Y_t) = \frac{p(y_t|X_t^{\mathrm{n}}, Y_{t-1})p(x_t^{\mathrm{n}}|X_{t-1}^{\mathrm{n}}, Y_{t-1})}{p(y_t|Y_{t-1})} p(X_{t-1}^{\mathrm{n}}|Y_{t-1}). \tag{49}$$

For the state-space model (31)*, with the assumptions according to* (32)–(35)*, we have that*

$$p(y_t|X_t^{\mathrm{n}}, Y_{t-1}) = \sum_{\Lambda_t} \alpha_t^{(\Lambda_t)}, \tag{50a}$$

$$p(x_{t+1}^{\mathrm{n}}|X_t^{\mathrm{n}}, Y_t) = \mathcal{N}(\hat{x}_{t+1|t}^{\mathrm{n}}, P_{t+1|t}^{\mathrm{n}}), \tag{50b}$$

where

$$\begin{aligned} \hat{x}_{t+1|t}^{\mathrm{n}} &= f_t^{\mathrm{n}}(x_t^{\mathrm{n}}) + F_t^{\mathrm{n}} \hat{x}_{t|t}^{\mathrm{l}}, \\ P_{t+1|t}^{\mathrm{n}} &= F_t^{\mathrm{n}} P_{t|t}^{\mathrm{l}} (F_t^{\mathrm{n}})^T + G_t^{\mathrm{n}} Q_t^{\mathrm{n}} (G_t^{\mathrm{n}})^T, \end{aligned} \tag{51}$$

with $\alpha_t^{(\Lambda_t)}$ given by (46)*.*

Proof: Expression (49) is given by repeated use of Bayes' rule. For $p(x_{t+1}^{\mathrm{n}}|X_t^{\mathrm{n}}, Y_t)$ see [26]. For $p(y_t|X_t^{\mathrm{n}}, Y_{t-1})$ rewrite it according to

$$\begin{aligned} &p(y_t|X_t^{\mathrm{n}}, Y_{t-1}) = \\ &\sum_{\Lambda_t} p(y_t|X_t^{\mathrm{n}}, Y_{t-1}, \Lambda_t) \Pr(\Lambda_t|X_{t-1}^{\mathrm{n}}, Y_{t-1}) = \\ &\sum_{\Lambda_t} \mathcal{N}(\hat{y}_{t|t-1}^{(\Lambda_t)}, S_{f,t}^{(\Lambda_t)}) \pi_{\lambda_{t-1}}^{\lambda_t} \bar{\alpha}_{t-1}^{(\Lambda_{t-1})} = \\ &\sum_{\Lambda_t} \alpha_t^{(\Lambda_t)}, \end{aligned} \tag{52}$$

where the last step is given by (46). □

For the particle filter algorithm, we can choose to use $p(x_t^{\mathrm{n}}|X_{t-1}^{\mathrm{n},(i)}, Y_{t-1})$ to update the samples, i.e.

$$x_t^{\mathrm{n},(i)} \sim p(x_t^{\mathrm{n}}|X_{t-1}^{\mathrm{n},(i)}, Y_{t-1}), \tag{53}$$

knowing that this is a Gaussian density and thereby easy to sample from. The weights are then calculated according to

$$\begin{aligned} w_t^{(i)} &= \bar{w}_{t-1}^{(i)} p(y_t^{\mathrm{n}} | X_t^{\mathrm{n},(i)}, Y_{t-1}) = \bar{w}_{t-1}^{(i)} \sum_{\Lambda_t} \alpha_t^{(i,\Lambda_t)}, \\ \bar{w}_t^{(i)} &= \frac{w_t^{(i)}}{\sum_k w_t^{(k)}}, \end{aligned} \tag{54}$$

which together with (53) yield (38).

A very important special case of (31) is when the matrices F_t^{n}, G_t^{n}, F_t^{d}, G_t^{d}, F_t^{l}, G_t^{l} and H_t are independent of x_t^{n}. In this case one can deduce from Lemma 1 and 2 that the Kalman filter covariance matrices are

$$P_{t|t}^{\mathrm{l},(i)} = P_{t|t}^{\mathrm{l}} \text{ and } P_{t|t}^{\mathrm{d},(i,\Lambda_t)} = P_{t|t}^{\mathrm{d},(\Lambda_t)}, \tag{55}$$

i.e. independent of $X_t^{\mathrm{n},(i)}$, and at each time t we only have to update it once for x_t^{l} and Λ_t times (each mode sequence) for $x_t^{\mathrm{d},(\Lambda_t)}$. This implies that, for a given number of samples N, the computational load for the marginalized particle filter is approximately the same as for the stand-alone particle filter. In this case, given that the number of samples needed for MPF is significantly lower than for the particle filter, the gain with respect to computational load can be substantial.

6 Blended INS/TAP using MPF

6.1 The applied algorithm

For the purpose of applying the marginalized particle filter we separate the position and altitude states from the others in (8) according to

$$\begin{aligned} x_t^{\mathrm{n}} &= \begin{bmatrix} \tilde{L}_t & \tilde{l}_t \end{bmatrix}^T, \\ x_t^{\mathrm{d}} &= \tilde{h}_t, \\ x_t^{\mathrm{l}} &= \begin{bmatrix} \tilde{v}_{n,t} & \tilde{v}_{e,t} & \gamma_{n,t} & \gamma_{e,t} & \gamma_{d,t} & b_{x,t}^a & b_{y,t}^a \end{bmatrix}^T. \end{aligned} \tag{56}$$

Using the system equations derived in Sections 3 and 4, the discrete state propagation and measurement equations become

$$\begin{bmatrix} x_{t+1}^{\mathrm{n}} \\ x_{t+1}^{\mathrm{d}} \\ x_{t+1}^{\mathrm{l}} \end{bmatrix} = F_t \begin{bmatrix} x_t^{\mathrm{n}} \\ x_t^{\mathrm{d}} \\ x_t^{\mathrm{l}} \end{bmatrix} + G_t \begin{bmatrix} u_t^{\mathrm{d}} \\ u_t^{\mathrm{l}} \end{bmatrix}. \tag{57a}$$

$$y_t = h\left(\begin{bmatrix} L_t^{\mathrm{ins}} \\ l_t^{\mathrm{ins}} \end{bmatrix} + x_t^{\mathrm{n}} \right) - x_t^{\mathrm{d}} + e_t, \tag{57b}$$

where $h(\cdot)$ in (57b) is the terrain database height with input arguments latitude $L_t = L_t^{\text{ins}} + \tilde{L}_t$ and longitude $l_t = l_t^{\text{ins}} + \tilde{l}_t$ and

$$
\begin{aligned}
F_t &= I + T_s A_t = \begin{bmatrix} F_{\text{n},t}^{\text{n}} & 0_{2\times 1} & F_{\text{l},t}^{\text{n}} \\ 0_{1\times 2} & 1 & 0_{1\times 7} \\ F_{\text{n},t}^{\text{l}} & 0_{1\times 7} & F_{\text{l},t}^{\text{l}} \end{bmatrix}, \\
G_t &= T_s(I + \frac{T_s}{2} A_t) B_t = \begin{bmatrix} 0_{2\times 1} & G_t^{\text{n}} \\ T_s(1+\frac{T_s}{2}) & 0_{1\times 7} \\ 0_{7\times 1} & G_t^{\text{l}} \end{bmatrix}, \\
u_t^{\text{d}} &= u_t^h, \\
u_t^{\text{l}} &= \begin{bmatrix} u_{x,t}^a & u_{y,t}^a & u_{x,t}^\gamma & u_{y,t}^\gamma & u_{z,t}^\gamma & u_{x,t}^b & u_{y,t}^b \end{bmatrix}^T.
\end{aligned}
\tag{58}
$$

Note that the state propagation model in (57a) is linear as opposed to the more general nonlinear model used in Section 5.

To only have to compute one covariance matrix P_t^{l}, the matrices $F_{\text{l},t}^{\text{l}}$, $F_{\text{l},t}^{\text{n}}$, G_t^{n} and G_t^{l} must all be independent of x_t^{n}. This is achieved by not compensating INS computed quantities with estimated errors before entering F_t, i.e. $x_t = x_t^{\text{ins}} + \tilde{x}_t \approx x_t^{\text{ins}}$. An alternative is to compensate using the MPF estimates, meaning that we use the same compensation for all $i = 1, \ldots, N$. The second alternative should be better if the INS errors are large, but for simplicity the first alternative is chosen here.

For the altitude error $x_t^{\text{d}} = \tilde{h}_t$ we choose to estimate it using the GPB filter. This means that we use two Kalman filters, each one conditioned on one of the modes in (29). For each time t the number of modes is always two. The recursions are then given by

$$
\begin{aligned}
\hat{x}_{t|t}^{\text{d},(i,\lambda_t)} &= \hat{x}_{t|t-1}^{\text{d},(i)} + \\
&\quad P_{t|t-1}^{\text{d},(i)} H_t^T (S_{f,t}^{(i,\lambda_t)})^{-1} (y_t - \hat{y}_{t|t-1}^{(i,\lambda_t)}), \\
P_{t|t}^{\text{d},(i,\lambda_t)} &= P_{t|t-1}^{\text{d},(i)} - \\
&\quad P_{t|t-1}^{\text{d},(i)} H_t^T (S_{f,t}^{(i,\lambda_t)})^{-1} H_t (P_{t|t-1}^{\text{d},(i)})^T, \\
\alpha_t^{(i,\lambda_t)} &= \mathcal{N}(\hat{y}_{t|t-1}^{(i,\lambda_t)}, S_{f,t}^{(i,\lambda_t)}) \sum_{\lambda_{t-1}=1}^{2} \pi_{\lambda_{t-1}}^{\lambda_t} \bar{\alpha}_{t-1}^{(\lambda_{t-1})}, \\
\hat{y}_{t|t-1}^{(i,\lambda_t)} &= h(x_t^{\text{n},(i)}) - H_t \hat{x}_{t|t-1}^{\text{d},(i)} - m_t^{(\lambda_t)}, \\
S_{f,t}^{(i,\lambda_t)} &= R_t^{(\lambda_t)} + H_t P_{t|t-1}^{\text{d},(i)} H_t^T.
\end{aligned}
\tag{59}
$$

To keep the number of mode sequences constant the result from the two Kalman filters are merged, using $\bar{\alpha}_t^{(i,\lambda_t)} = \alpha_t^{(i,\lambda_t)} / \sum_{\lambda_t=1}^{2} \alpha_t^{(i,\lambda_t)}$, according to

$$
\begin{aligned}
\hat{x}_{t|t}^{\text{d},(i)} &= \sum_{\lambda_t=1}^{2} \bar{\alpha}_t^{(i,\lambda_t)} \hat{x}_{t|t}^{\text{d},(i,\lambda_t)}, \\
P_{t|t}^{\text{d},(i)} &= \sum_{\lambda_t=1}^{2} \bar{\alpha}_t^{(i,\lambda_t)} (P_{t|t}^{\text{d},(i,\lambda_t)} + (\hat{x}_{t|t}^{\text{d},(i,\lambda_t)} - \hat{x}_{t|t}^{\text{d},(i)})^2).
\end{aligned}
\tag{60}
$$

We will add some artificial process noise u_t^{add} for the latitude and longitude error states, to deal with particle filter discretization errors and to further decrease the number of needed particles. This will change the state propagation equation for horizontal position in (57a) to

$$x_{t+1}^{\mathrm{n}} = F_{\mathrm{n},t}^{\mathrm{n}} x_t^{\mathrm{n}} + F_{\mathrm{l},t}^{\mathrm{n}} x_t^{\mathrm{l}} + \underbrace{G_t^{\mathrm{n}} u_t^{\mathrm{l}} + u_t^{\mathrm{add}}}_{u_t^{\mathrm{n}}}, \tag{61}$$

which should be compared to the propagation equation for x_t^{l}, i.e.

$$x_{t+1}^{\mathrm{l}} = F_{\mathrm{n},t}^{\mathrm{l}} x_t^{\mathrm{n}} + F_{\mathrm{l},t}^{\mathrm{l}} x_t^{\mathrm{l}} + G_t^{\mathrm{l}} u_t^{\mathrm{l}}. \tag{62}$$

The process noises u_t^{n} and u_t^{l} are mutually correlated. On the other hand, here $Q_t^{\mathrm{add}} \gg Q_t^{\mathrm{l}}$ which yields $u_t^{\mathrm{n}} \approx u_t^{\mathrm{add}}$ and the correlation is therefore neglected.

A summary of the applied algorithm is given in Algorithm 1.

Algorithm 1 (The MPF for blended INS/TAP).

1. Initialization:
   For $i = 1, \ldots, N$, sample $x_0^{\mathrm{n},(i)} \sim p(x_0^{\mathrm{n}})$, and set

$$\begin{aligned}
\{\hat{x}_{0|-1}^{\mathrm{l},(i)}, P_{0|-1}^{\mathrm{l}}\} &= \{0, P_0^{\mathrm{l}}\}, \\
\{\hat{x}_{0|-1}^{\mathrm{d},(i)}, P_{0|-1}^{\mathrm{d},(i)}\} &= \{0, P_0^{\mathrm{d}}\}, \\
\{\bar{\alpha}_{-1}^{(i,1)}, \bar{\alpha}_{-1}^{(i,2)}\} &= \{\Pr(\lambda_t = 1), \Pr(\lambda_t = 2)\}.
\end{aligned}$$

2. GPB filter measurement update:
   For $i = 1, \ldots, N$ and $\lambda_t = 1, 2$, compute

$$\begin{aligned}
&\{\hat{x}_{t|t}^{\mathrm{d},(i,\lambda_t)}, P_{t|t}^{\mathrm{d},(i,\lambda_t)}, \alpha_t^{(i,\lambda_t)}\} \textit{ using } (59), \\
&\bar{\alpha}_t^{(i,\lambda_t)} = \alpha_t^{(i,\lambda_t)} / (\alpha_t^{(i,1)} + \alpha_t^{(i,2)}), \\
&\{\hat{x}_{t|t}^{\mathrm{d},(i)}, P_{t|t}^{\mathrm{d},(i)}\} \textit{ using } (60).
\end{aligned}$$

3. Particle filter measurement update:
   For each $i = 1, \ldots, N$, update

$$w_t^{(i)} = \bar{w}_{t-1}^{(i)} \textstyle\sum_{\lambda_t=1}^{2} \alpha^{(i,\lambda_t)}, \; \bar{w}_t^{(i)} = w_t^{(i)} / \textstyle\sum_i w_t^{(i)}.$$

4. Resampling:
   Resample N times with replacement according to

$$\Pr(x_t^{(i)} = x_t^{(k)}) = \bar{w}_t^{(i)}.$$

5. Kalman filter measurement uppdate:
   For each $i = 1, \ldots, N$, set

$$\hat{x}^{\mathrm{l},(i)}_{t|t} = \hat{x}^{\mathrm{l},(i)}_{t|t-1},\, P^{\mathrm{l}}_{t|t} = P^{\mathrm{l}}_{t|t-1}.$$

6. GPB filter time update:
   For each $i = 1, \ldots, N$, compute

$$\begin{aligned}\hat{x}^{\mathrm{d},(i)}_{t+1|t} &= \hat{x}^{\mathrm{d},(i)}_{t|t},\\ P^{\mathrm{d},(i)}_{t+1|t} &= P^{\mathrm{d},(i)}_{t|t} + T_s(1+T_s/2)^2 Q^{\mathrm{d}}_t.\end{aligned}$$

7. Particle filter time update:
   For $i = 1, \ldots, N$, sample

$$x^{\mathrm{n},(i)}_{t+1} \sim p(x^{\mathrm{n}}_{t+1}|X^{\mathrm{n},(i)}_t, Y_t) \text{ using (50b).}$$

8. Kalman filter time update:
   For each $i = 1, \ldots, N$, compute

$$\{\hat{x}^{\mathrm{l},(i)}_{t+1|t}, P^{\mathrm{l},(i)}_{t+1|t}\} \text{ according to (42).}$$

6.2 Convergence Analysis of Algorithm 1

For the estimation of x^{n}_t there are to the authors knowledge not many results which can be used for convergence analysis. The results that do exist e.g. [10] are unfortunately rather conservative. Simulations indicate however that given a large enough number of samples the estimate of x^{n}_t does converge.

We can on the other hand analyze the behaviour of x^{l}_t and x^{d}_t. Below we show that the estimates of both x^{l}_t and x^{d}_t always converge, although they likely converge to something wrong if the estimate of x^{n}_t diverges. Rewrite the model for x^{l}_t according to

$$\begin{aligned}x^{\mathrm{l}}_{t+1} &= F^{\mathrm{l}}_{\mathrm{l},t}x^{\mathrm{l}}_t + F^{\mathrm{l}}_{\mathrm{n},t}x^{\mathrm{n}}_t + G^{\mathrm{l}}_t u^{\mathrm{l}}_t\\ z^{\mathrm{l}}_t &= x^{\mathrm{n}}_{t+1} - F^{\mathrm{n}}_{\mathrm{n},t}x^{\mathrm{n}}_t = F^{\mathrm{n}}_{\mathrm{l},t}x^{\mathrm{l}}_t + u^{\mathrm{n}}_t,\end{aligned} \tag{63}$$

The term $F^{\mathrm{l}}_{\mathrm{n},t}x^{\mathrm{n}}_t$ in (63) can be regarded as a known input signal.

Suppose first that the aircraft is traveling without any turns, at constant speed, at the same altitude, and that the path is located around 60 degrees latitude. In this case, the eigenvalues of $F^{\mathrm{l}}_{\mathrm{l},t}$ all lie on or slightly outside the unit-circle. For the Riccati recursion to converge a necessary condition is that the system is detectable [17]. For detectability in this case we need full observability. To investigate the observability we can use the observability matrix

$$O(t, t+k) = \begin{bmatrix} H_t \\ \vdots \\ H_{t+k-1}F_{t+k-2}\cdots F_t \end{bmatrix}, \tag{64}$$

where in our case $H_t = F^{\mathrm{n}}_{\mathrm{l},t}$ and $F_t = F^{\mathrm{l}}_{\mathrm{l},t}$. We know from [24] that if rank $\big(O(t, t+k)\big) = \dim(x_t)$, then the system is observable. It is straightforward to verify, under the

flight conditions stated above, we actually have full observability after only four steps, i.e. $k = 4$. On the other hand, most of the singular values of $O(t, t+k)$ are very small, and this is true for larger values on k as well.

To clarify, we can simplify the system equations further by discarding those elements which are insignificant during shorter periods of time, say one or two minutes. For these short periods of time we can neglect that the earth rotates and that the surface of the earth is curved, i.e.

$$\omega_{ie}^n \approx 0_{3\times 1}, \quad \omega_{en}^n \approx 0_{3\times 1}. \tag{65}$$

The simplification above means that the state transition matrix will look like

$$F_{\mathrm{l},t}^{\mathrm{l}} = I + T_s \begin{bmatrix} 0_{2\times 2} & \begin{matrix} 0 & f_d \\ -f_d & 0 \end{matrix} & \begin{matrix} -f_e & c_{b,11}^n & c_{b,12}^n \\ f_n & c_{b,21}^n & c_{b,22}^n \end{matrix} \\ 0_{5\times 2} & & 0_{5\times 5} \end{bmatrix}. \tag{66}$$

Moreover, although the term f^n in (19) can be regarded as a known input signal (at least $f^{n,\mathrm{ins}}$ is known), it is convenient for the analysis to rewrite it as

$$f^n = \dot{v}^n + (\Omega_{en}^n + 2\Omega_{ie}^n)v^n - g^n \approx \begin{bmatrix} \dot{v}_n \\ \dot{v}_e \\ \dot{v}_d - g_d \end{bmatrix}. \tag{67}$$

From the simplified system matrix $F_{\mathrm{l},t}^{\mathrm{l}}$ above and the expression for the specific force f^n we can draw two conclusions. First of all, if there is no horizontal acceleration, i.e. $\dot{v}_n = \dot{v}_e = 0$, γ_d will not be observable. This is easily seen from (66), because in this case $f_n \approx f_e \approx 0$ and γ_d will thereby not have any influence on v_n or v_e, hence unobservable. Secondly, flying along a straight path means that only the sum of $-f_d\gamma_n$ and b_e^a, and $f_d\gamma_e$ and b_n^a, where

$$\begin{aligned} b_n^a &= c_{b,11}^n b_x^a + c_{b,12}^n b_y^a, \\ b_e^a &= c_{b,21}^n b_x^a + c_{b,22}^n b_y^a, \end{aligned} \tag{68}$$

are observable. We need a change in C_b^n to be able estimate the individual components in the two sums.

The detectability criteria is only a necessary condition for the Riccati recursion to converge. A necessary and sufficient condition is to also require that the system is unit-circle controllable [17]. Here, it is straightforward to verify, by inspection of G_t^k, that the system is actually controllable, and thereby also unit-circle controllable.

The same reasoning as for x_t^{l} applies to x_t^{d}. Rewrite the model for x_t^{d} according to

$$\begin{aligned} x_{t+1}^{\mathrm{d}} &= x_t^{\mathrm{d}} + G_t^{\mathrm{d}} u_t^{\mathrm{d}} \\ z_t^{\mathrm{d}} &= y_t - h\left(\begin{bmatrix} L_t^{\mathrm{ins}} \\ l_t^{\mathrm{ins}} \end{bmatrix} + x_t^{\mathrm{n}}\right) = -x_t^{\mathrm{d}} + e_t(\lambda_t). \end{aligned} \tag{69}$$

It is obvious that the model is both observable and controllable thereby providing sufficient conditions for the Riccati equation to converge. The mode variable λ_t could possibly cause the estimate to converge to something wrong, but simulations show that this is highly unlikely.

7 Simulation Results

In this section we apply the marginalized particle filter according to Algorithm 1 on simulated inertial navigation data. Terrain elevation data is taken from a commercial database, which contains terrain elevation at discrete points separated with 50 metres in both north and east directions. Elevation data at intermediate points is computed using bilinear interpolation. The flight trajectory projected onto the ground is depicted in Figure 2.

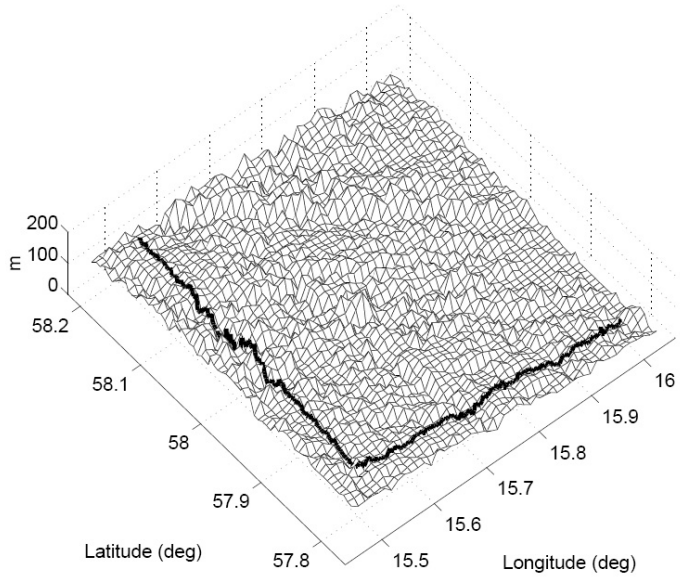

Figure 2: *Terrain elevation profile along the flight trajectory.*

As can be deduced from Figure 2 the flight trajectory makes a turn after about half of the distance. The main reason for this turn is to make γ_n and γ_e distinguishable from b_x^a and b_y^a. The measurements are assumed unavailable during the turn, to imitate the fact that the radar altimeter provides poor ground clearance measurements when the absolute value of the bank angle $|\phi|$ is large. The bank angle during the turn is 60 deg. Moreover, to make γ_d observable, the speed along the path changes from time to time according to Figure 3. Note that the turn and speed changes are used to make attitude, heading and accelerometer biases observable. Position and velocity errors are observable without accelerations meaning that the algorithm does not require accelerations for accurate position and velocity estimates.

To simulate INS data we have used the truth flight profile given by Figure 2 and 3 and worked backwards through the nonlinear motion model given in Appendix A.1. Sensor errors according to Table 2 have then been added to the exact sensor measurements obtained from the backward propagation. Finally the sensor measurements, now with errors added, are run through the nonlinear motion model to yield as close to authentic INS data as possible. Note that INS initial alignment is not simulated but initial errors

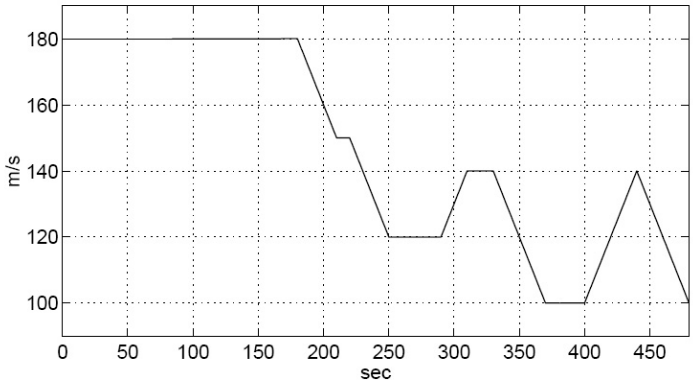

Figure 3: *Ground speed along the flight trajectory.*

given by Table 2 are used to initialize the INS computations. Simulated terrain elevation

Table 2: *Simulated INS errors.*

$p(x_0^{\mathrm{n}})$	$\mathcal{U}\big(-\frac{1000\sqrt{3}}{r_0}, \frac{1000\sqrt{3}}{r_0}\big) \cdot \mathcal{U}\big(-\frac{1000\sqrt{3}}{r_0 \cos L_0}, \frac{1000\sqrt{3}}{r_0 \cos L_0}\big)$
$p(x_0^{\mathrm{d}})$	$\mathcal{N}(0, 50)$
$p(x_0^{\mathrm{l}})$	$\mathcal{N}\big(0, \mathrm{diag}(1, 1, \frac{0.05\pi}{180}, \frac{0.05\pi}{180}, \frac{0.1\pi}{180}, 10^{-3} \cdot 5_{1\times 2})^2\big)$
$p(u_t^{\mathrm{d}})$	$\mathcal{N}(0, 0.2)$
$p(u_t^{\mathrm{l}})$	$\mathcal{N}\big(0, \mathrm{diag}(10^{-4}, 10^{-4}, 10^{-6} \cdot 1_{1\times 5})^2\big)$

measurements have been created along the flight path by adding a random error defined by

$$p(e_t) = 3/4 \cdot \mathcal{N}(0, 3^2) + 1/4 \cdot \mathcal{N}(12, 6^2), \tag{70}$$

to the true terrain elevation. We have assumed mode transition probabilities for the measurement noise from (35) according to

$$\begin{bmatrix} \pi_1^1 & \pi_2^1 \\ \pi_1^2 & \pi_2^2 \end{bmatrix} = \begin{bmatrix} 3/4 & 3/4 \\ 1/4 & 1/4 \end{bmatrix}. \tag{71}$$

Note that these particular parameter values are not authentic but gives an adequate example on the distribution of the radar altimeter measurement error over dense forest. In practice the values are found empirically by comparing measurements from GPS, radar altimeter and terrain height database over different types of terrain.

For the marginalized filter we used sampling period $T_s = 1$ sec and 12000 particles ($N = 12000$). No significant improvement was obtained using more than 12000 particles. For the additional process noise we chose

$$u_t^{\mathrm{add}} \sim \mathcal{N}(0, 2 \cdot 10^{-3}\, P_{t|t-1}^{\mathrm{n,MPF}}), \tag{72}$$

applied to x_t^{n} according to (61). Deterministic resampling [12] were performed if

$$\sum_i 1/(\bar{w}_t^{(i)})^2 < 2N/3 \tag{73}$$

and at least five filter iterations have past since the last resampling.

The result (RMSE$_t$ and $\sqrt{P_t^{\mathrm{MEAN}}}$) based on 100 Monte Carlo simulations is depicted in Figure 4 for position ($s_L = \tilde{L}\, r_0$ and $s_l = \tilde{l}\, r_0 \cos L$) and altitude errors, Figure 5 for velocity and acceleration errors, and Figure 6 for platform orientation errors respectively. The RMSE$_t$ and P_t^{MEAN} are computed according to

$$\mathrm{RMSE}_t = \left(\frac{1}{100} \sum_{m=1}^{100} ||\hat{x}_{t|t-1}^{\mathrm{MPF},(m)} - x_t^{\mathrm{true},(m)}||_2^2 \right)^{1/2}, \tag{74a}$$

$$P_t^{\mathrm{MEAN}} = \frac{1}{100} \sum_{m=1}^{100} \mathrm{tr}(P_{t|t-1}^{\mathrm{MPF},(m)}), \tag{74b}$$

m representing the m:th Monte Carlo simulation.

The RMSE$_t$ for horizontal position decreases to about 30 m after 40 sec, and the stationary error level lies around 20 m. The RMSE$_t$ for the altitude converges to its stationary value, approximately 1 m, after 30 sec. The RMSE$_t$ for horizontal velocity drops below 0.2 m/s, and the INS horizontal platform orientation error (γ_n and γ_e) drop below $0.02/\sqrt{2} \approx 0.015$ deg for each of the two errors. The RMSE$_t$ for b_x^a and b_x^a approach $3 \cdot 10^{-3}/\sqrt{2} \approx 2 \cdot 10^{-3}$ m/s^2, and for γ_d it approaches 0.075 deg. Note also the distinct increase of the position and velocity error and uncertainty during the turn (between time 150 to 175 sec). This is due to that no terrain elevation measurements are used during this period of time.

In the same figures the corresponding Cramer-Rao posterior prediction bounds are shown. The bound is computed according to [20]

$$\begin{aligned} P_0^{\mathrm{CR}} &= P_0, \\ P_{t+1}^{\mathrm{CR}} &= F_t P_t^{\mathrm{CR}} \big(I - (I + \mathcal{R}_t^{-1} P_t^{\mathrm{CR}})^{-1} \mathcal{R}_t^{-1} P_t^{\mathrm{CR}} \big) F_t^T \\ &\quad + G_t Q_t G_t^T, \end{aligned} \tag{75}$$

where F_t and G_t are taken from (58), Q_t and P_0 from Table 2 and $\mathcal{R}_t^{-1}$ is given by

$$\begin{aligned} \mathcal{R}_t^{-1} = \mathbb{E}_{p(e_t)} \left[\left(\tfrac{d}{de_t} \log p(e_t) \right)^2 \right] \cdot \\ \mathbb{E}_{p(x_t^{\mathrm{n}})} \begin{bmatrix} \frac{\partial}{\partial x_t^{\mathrm{n}}} h(x_t^{\mathrm{n}}) \\ -1 \\ 0_{7\times 1} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x_t^{\mathrm{n}}} h(x_t^{\mathrm{n}}) \\ -1 \\ 0_{7\times 1} \end{bmatrix}^T . \end{aligned} \tag{76}$$

$\mathcal{R}_t^{-1}$ is computed by evaluating the expectations in (76) using a large number of samples from $p(e_t)$ according to (70) and $p(x_0^{\mathrm{n}}) \approx p(x_t^{\mathrm{n}})$ according to Table 2. As can be seen from the figures, the RMSE$_t$ of the filter estimates are all slightly larger than the corresponding P$_t^{\mathrm{CR}}$, but the difference is small indicating that the applied filter is close to being optimal with respect to RMSE$_t$, at least after filter convergence.

8 Conclusions

In this paper we have extended the MPF from [26] to account for a multi-modal measurement noise. The extended filter has been applied on a tightly blended INS/TAP navigation system. We have shown that by concentrating on the inertial navigation errors, we can linearize the state transition equations without introducing any significant errors. The MPF takes advantage of the linearized structure, and estimates it using relatively fast Kalman filters. The highly nonlinear terrain-aided positioning system only depends on position, meaning that we can focus the computer intensive particle filter on the position part of the state vector only. Compared to applying a stand-alone particle filter we can decrease the number of samples substantially and thereby making the applied MPF computationally tractable.

Simulations have been performed on simulated inertial navigation data, using a commercial terrain elevation database to simulate the terrain-aided positioning system. The simulation result is compared with the Cramer-Rao lower bound. The comparison shows that we obtain nearly optimal accuracy, at least after filter convergence. The deviation between the lower bound and the simulation result partly depends on the fact that the particle filter still only provides an approximate solution particularly due to discretization errors. Another possible contributing factor to the deviation could be that the Cramer-Rao bound is not a tight bound in this case. There could very well exist other bounds that are tighter, see e.g. [27].

A Appendix

A.1 INS Motion Equations

Based on measured accelerations and angular rates in three dimensions the INS computes position, velocity, attitude and heading. The computations are based on an accurate nonlinear motion model describing the kinematics of the system. These equations will not be derived here, for detailed derivations see e.g. [6, 25, 13]. To be able to characterize the INS mathematically we will need a number of coordinate frames given by:

i Inertial frame, fixed in the inertial space. For navigation periods shorter than days this frame can be approximated with an earth centered non-rotating frame.

e Earth-centered frame, fixed to the earth, i.e it rotates with the earth.

n Navigation frame, with its center attached to the aircraft. The x, y and z-axis are aligned with north, east and the ellipsoid normal respectively. The velocity e.g. is denoted by $v^n = [v_n, v_e, v_d]^T$.

b Body frame, attached to the aircraft, thereby always translating and rotating with the aircraft. The x, y and z-axis points through the nose, right wing and belly respectively. The acceleration e.g. is denoted by $a^b = [a_x, a_y, a_z]^T$.

The horizontal position is usually given as two angles, latitude and longitude. Latitude refers to the angle between the normal to the reference ellipsoid and the equatorial plane, and will be denoted by L. Longitude is the angle between the same normal and a plane

intersecting the Greenwich meridian, and will be denoted by l. The reference ellipsoid is defined by the World Geodetic System 1984 (WGS84), see [1] or Table 3 for numerical values. The equations for latitude, longitude and altitude are

Table 3: *Parameters for WGS84.*

Parameter	**Notation**	**Numerical value**
Semimajor axis	r_0	$6.378137 \cdot 10^6$ m
Reciprocal of flattening	$1/\mathrm{f}$	298.2572
First eccentricity	ε	0.08181919
Angular velocity	ω_{ie}	$7.292115 \cdot 10^{-5}$ rad/sec
Gravity at equator	g_0	9.780325 m/s^2
Gravity formula constant	k	0.001931853
Gravity formula constant	m	0.003449787

$$\begin{aligned} \dot{L} &= \frac{v_n}{r_L + h}, \\ \dot{l} &= \frac{v_e}{(r_l + h)\cos L}, \\ \dot{h} &= -v_d. \end{aligned} \tag{77}$$

In (77), the two radii of curvature are given by

$$r_L = \frac{r_0(1-\varepsilon^2)}{(1-\varepsilon^2 \sin^2 L)^{3/2}}, \quad r_l = \frac{r_0}{(1-\varepsilon^2 \sin^2 L)^{1/2}}, \tag{78}$$

where the constant ε is the earth's first eccentricity, see Table 3.

The velocity of the aircraft relative to the earth, expressed in the navigation (n) frame and denoted by $v^n = \begin{bmatrix} v_n & v_e & v_d \end{bmatrix}^T$, is given as the solution to the differential equation

$$\dot{v}^n = C_b^n f^b - (\Omega_{en}^n + 2\Omega_{ie}^n)v^n + g^n. \tag{79}$$

The vector f^b is the acceleration sensed by the accelerometers (specific force vector) and C_b^n is a transformation matrix from body frame to navigation frame. The matrices Ω_{en}^n and Ω_{ie}^n represent the rotation of the navigation frame relative to earth and earth relative to inertial frame respectively, both expressed in the n-frame. The rotation described by Ω_{en}^n arises when travelling over the curved surface of the earth. The matrices Ω_{en}^n and Ω_{ie}^n are both the skew-symmetric matrix

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{80}$$

of the corresponding vector $\omega = \begin{bmatrix}\omega_x & \omega_y & \omega_z\end{bmatrix}^T$. In vector form the rotations are given by

$$\omega_{ie}^n = \begin{bmatrix}\cos L \\ 0 \\ -\sin L\end{bmatrix}\omega_{ie}, \quad \omega_{en}^n = \begin{bmatrix}\frac{v_e}{r_l+h} \\ -\frac{v_n}{r_L+h} \\ -\frac{v_e \tan L}{r_l+h}\end{bmatrix}, \tag{81}$$

where ω_{ie} is a scalar representing the angular velocity of the earth. For a numerical value see Table 3.

The acceleration f^b includes the effect of the gravity vector, g^n, which represents the sum of the earth's gravitation, G^n, and the centripetal acceleration due to the rotation of the earth, i.e.

$$g^n = G^n - (\Omega_{ie}^n)^2 r^n, \tag{82}$$

where r^n is the position vector of the aircraft measured from the centre of the earth. The WGS84 ellipsoid is defined in such a way that the angle between g^n and the normal to the ellipsoid is minimized. The deflection of the vertical, i.e. the remaining error angle between the ellipsoid's normal and the gravity vector, is usually less than 5 μrad. Therefore, without introducing any significant errors, the gravity is approximately given by [1]

$$\begin{aligned} g^n &\approx \begin{bmatrix}0 & 0 & g_d\end{bmatrix}^T, \\ g_d &= g_0 \frac{1 + k\sin^2 L}{(1 - \varepsilon^2 \sin^2 L)^{1/2}} \cdot \\ &\quad \left(1 - \frac{2h}{r_0}\left(1 + \mathrm{f} + m - 2\mathrm{f}\sin^2 L\right) + \frac{3h^2}{r_0^2}\right). \end{aligned} \tag{83}$$

For numerical values on g_0, ε, f, k, m and r_0 see Table 3.

The attitude and heading of the aircraft are often represented by an orthogonal matrix C_b^n, $(C_b^n)^T C_b^n = I$, relating a vector in the body frame to a vector in the navigation frame. This matrix is referred to as a direction cosine matrix (DCM), and the coupling to the attitude and heading of the aircraft is

$$C_b^n = \begin{bmatrix}c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta\end{bmatrix}\begin{bmatrix}1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi\end{bmatrix}. \tag{84}$$

In (84), ϕ, θ and ψ are the roll, pitch and heading angles, and s and c are short for $\sin$ and $\cos$ respectively. The corresponding matrix differential equation for C_b^n is given by

$$\dot{C}_b^n = C_b^n \Omega_{ib}^b - \Omega_{in}^n C_b^n, \tag{85}$$

where the skew-symmetric matrices Ω_{ib}^b and Ω_{in}^n are again given by their vector counterparts ω_{ib}^b and ω_{in}^n according to (80). The vector ω_{ib}^b corresponds to the angular rates exhibited by the body frame, expressed in the body frame, i.e. the angular rates sensed by the rate gyros. Moreover, $\omega_{in}^n = \omega_{ie}^n + \omega_{en}^n$.

A.2 Discrete-Time Propagation Matrices

$A(t)$ is given by

$$A(t) = \begin{bmatrix} A_{\mathrm{n}}^{\mathrm{n}}(t) & 0_{2\times 1} & A_{\mathrm{l}}^{\mathrm{n}}(t) \\ 0_{1\times 2} & 0 & 0_{1\times 7} \\ A_{\mathrm{n}}^{\mathrm{l}}(t) & 0_{7\times 1} & A_{\mathrm{l}}^{\mathrm{l}}(t) \end{bmatrix}$$

where

$$A_{\mathrm{n}}^{\mathrm{n}}(t) = \begin{bmatrix} 0 & 0 \\ \frac{v_e \sin L}{r_0 \cos^2 L} & 0 \end{bmatrix},$$

$$A_{\mathrm{l}}^{\mathrm{n}}(t) = \begin{bmatrix} \frac{1}{r_0} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{r_0 \cos L} & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$A_{\mathrm{n}}^{\mathrm{l}}(t) = \begin{bmatrix} d_L^{v_n} & d_L^{v_e} & d_L^{\gamma_n} & 0 & d_L^{\gamma_d} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T,$$

$$A_{\mathrm{l}}^{\mathrm{l}}(t) = \begin{bmatrix} d_{v_n}^{v_n} & d_{v_e}^{v_n} & 0 & f_z & -f_e & c_{b,11}^n & c_{b,12}^n \\ d_{v_n}^{v_e} & d_{v_e}^{v_e} & -f_z & 0 & f_n & c_{b,21}^n & c_{b,22}^n \\ 0 & -\frac{1}{r_0} & 0 & d_{\gamma_e}^{\gamma_n} & d_{\gamma_d}^{\gamma_n} & 0 & 0 \\ \frac{1}{r_0} & 0 & d_{\gamma_n}^{\gamma_e} & 0 & d_{\gamma_d}^{\gamma_e} & 0 & 0 \\ 0 & \frac{\tan L}{r_0} & d_{\gamma_n}^{\gamma_d} & d_{\gamma_e}^{\gamma_d} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and

$$\begin{aligned}
d_L^{v_n} &= -\frac{v_e^2}{r_0 \cos^2 L} + 2\omega_{ie} v_e \cos L \\
d_L^{v_e} &= \frac{v_n v_e}{r_0 \cos^2 L} - 2\omega_{ie} v_d \sin L + 2\omega_{ie} v_n \cos L \\
d_L^{\gamma_n} &= \omega_{ie} \sin L \\
d_L^{\gamma_d} &= \frac{v_e}{r_0 \cos^2 L} + \omega_{ie} \cos L \\
d_{v_n}^{v_n} &= \frac{v_d}{r_0} \\
d_{v_e}^{v_n} &= -\frac{2 v_e \tan L}{r_0} - 2\omega_{ie} \sin L \\
d_{v_n}^{v_e} &= \frac{v_e \tan L}{r_0} + 2\omega_{ie} \sin L \\
d_{v_e}^{v_e} &= \frac{v_n \tan L}{r_0} + \frac{v_d}{r_0} \\
d_{\gamma_n}^{\gamma_e} &= -\omega_{ie} \sin L + \frac{v_e \tan L}{r_0}, \quad d_{\gamma_e}^{\gamma_n} = -d_{\gamma_n}^{\gamma_e} \\
d_{\gamma_n}^{\gamma_d} &= \frac{v_n}{r_0}, \quad d_{\gamma_d}^{\gamma_n} = -d_{\gamma_n}^{\gamma_d} \\
d_{\gamma_e}^{\gamma_d} &= \omega_{ie} \cos L + \frac{v_e}{r_0}, \quad d_{\gamma_d}^{\gamma_e} = -d_{\gamma_e}^{\gamma_d}.
\end{aligned}$$

$B(t)$ is given by

$$B(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c^n_{b,11} & c^n_{b,12} & 0 & 0 & 0 & 0 & 0 \\ 0 & c^n_{b,21} & c^n_{b,22} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c^n_{b,11} & c^n_{b,12} & c^n_{b,13} & 0 & 0 \\ 0 & 0 & 0 & c^n_{b,21} & c^n_{b,22} & c^n_{b,23} & 0 & 0 \\ 0 & 0 & 0 & c^n_{b,31} & c^n_{b,32} & c^n_{b,33} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

References

[1] Department of Defense World Geodetic System 1984. Technical Report NIMA TR8350.2, 3rd Ed, National Imagery and Mapping Agency, July 1997.

[2] C. Andrieu and A. Doucet. Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 64(4), 2002.

[3] B. Azimi-Sadjadi and P. S. Krishnaprasad. Approximate nonlinear filtering and its application in navigation. *Automatica*, 41(6), June 2005.

[4] Niclas Bergman, Lennart Ljung, and Fredrik Gustafsson. Terrain navigation using Bayesian statistics. *IEEE Control Systems Magazine*, 19(3):33–40, June 1999.

[5] H. A. P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, Aug 1988.

[6] K. R. Britting. *Inertial Navigation Systems Analysis*. Wiley - Interscience, 1971.

[7] R. G. Brown and P. Y. C. Hwang. *Introduction to random signals and applied Kalman filtering*. John Wiley & Sons, 3rd edition, 1992.

[8] J.L. Campbell, M. Uijt de Haag, and F. van Graas. Terrain-referenced positioning using airborne laser scanner. *Navigation: Journal of The Institute of Navigation*, 52(4), 2005.

[9] H. Carvalho, P. del Moral, A. Monin, and G. Salut. Optimal nonlinear filtering in GPS/INS integration. *IEEE Transactions on Aerospace and Electronic Systems*, 33(3):835–849, July 1997.

[10] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50(3), March 2002.

[11] J. Dezert. Improvement of strapdown inertial navigation using pdaf. *IEEE Transactions on Aerospace and Electronic Systems*, 35(3), July 1999.

[12] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo methods in practice*. Springer-Verlag, 2001.

[13] J. L. Farrell. *Integrated aircraft navigation*. Academic Press, 1976.

[14] F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons Inc, 1st edition, 2000.

[15] G. Hendeby and F. Gustafsson. Fundamental filtering limitations in linear non-Gaussian systems. In *Proceedings of the 16th IFAC World Congress*, 2005.

[16] A. J. Henley. Terrain aided navigation - current status, techniques for flat terrain and reference data requirements. In *Proceedings of IEEE Position, Location and Navigation Symposium*, 1990.

[17] T. Kailath, A. Sayed, and B. Hassibi. *Linear estimation*. Prentice Hall, 2001.

[18] J. Kim and S. Sukkarieh. Autonomous airborne navigation in unknown terrain environments. *IEEE Transactions on Aerospace and Electronic Systems*, 40(3), July 2004.

[19] A. Lawrence. *Modern Inertial Technology: Navigation, Guidance and Control*. Springer-Verlag, New York, USA, 2nd edition, 1998.

[20] P-J. Nordlund. *Sequential Monte Carlo Filters and Integrated Navigation*. Licentiate thesis 945, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 2002.

[21] P-J. Nordlund and F. Gustafsson. Sequential Monte Carlo filtering techniques applied to integrated navigation systems. In *Proceedings of the 2001 American Control Conference*, volume 6, pages 4375–4380, 2001.

[22] P-J. Nordlund and F. Gustafsson. Recursive estimation of three-dimensional aircraft position using terrain-aided positioning. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Proceesing*, volume 2, pages 1121–1124, 2002.

[23] P. Pucar and J. Palmqvist. Saab NINS/NILS - an autonomous landing system for Gripen. In *Proceedings of the Position Location and Navigation Symposium, IEEE 2000*, 2000.

[24] W. J. Rugh. *Linear System Theory*. Prentice Hall, Upper Saddle River, N.J. USA, 2nd edition, 1996.

[25] P. G. Savage. *Strapdown Inertial Navigation - Lecture Notes*. Strapdown Associates Inc., Plymouth, MN, USA, Feb 1990.

[26] T.B. Schön, F. Gustafsson, and P-J. Nordlund. Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing*, 53(7):2279–2289, July 2005.

[27] A. J. Weiss and E. Weinstein. A lower bound on the mean-square error in random parameter estimation. *IEEE Transactions on information theory*, 31(5):680–682, 1985.

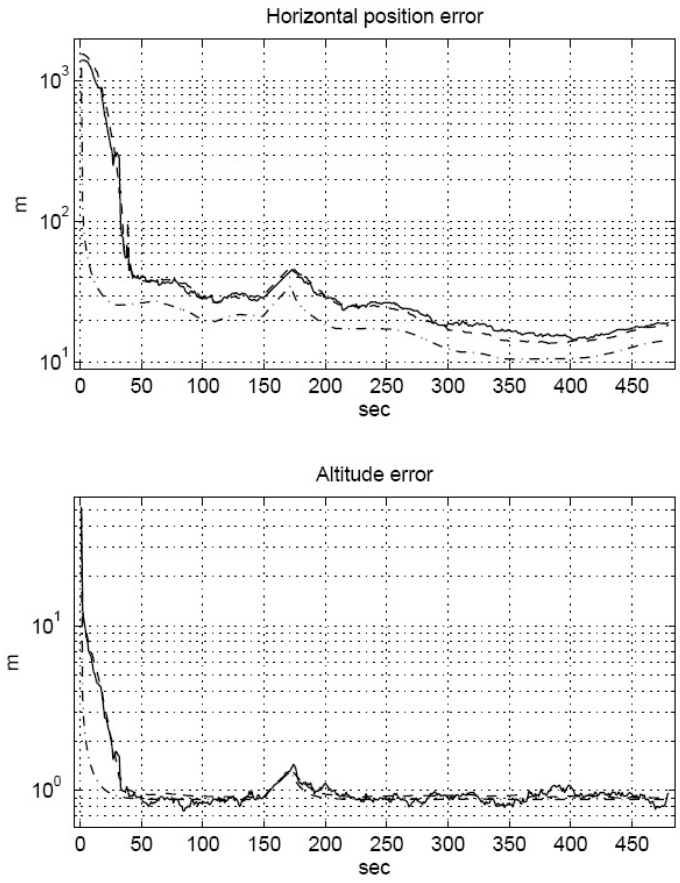

Figure 4: *RMSE$_t$ (solid line), $\sqrt{P_t^{\mathrm{MEAN}}}$ (dashed line) and $\sqrt{P_t^{\mathrm{CR}}}$ (dash-dotted line) for horizontal position $\sqrt{s_L^2 + s_l^2}$ and altitude $\tilde{h}$.*

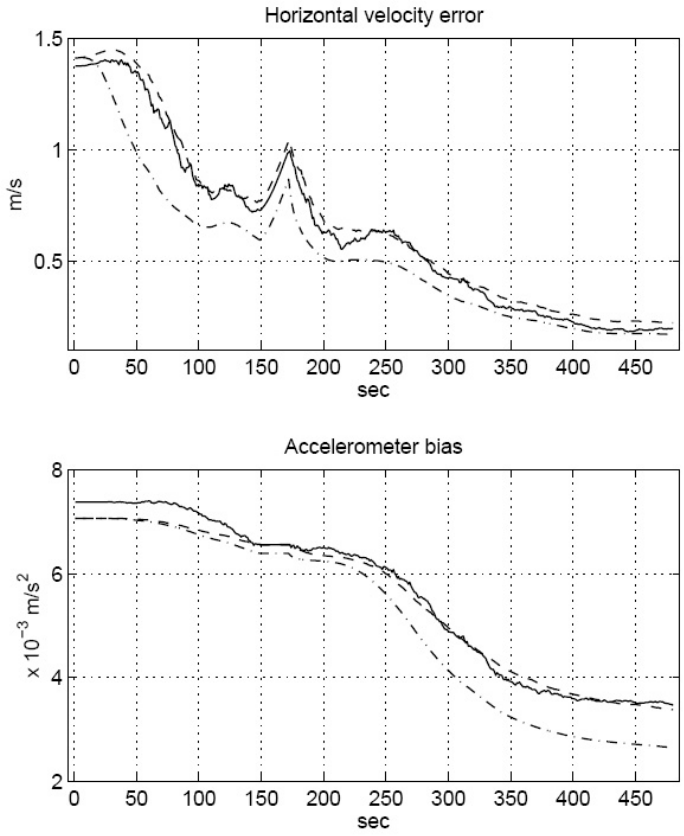

Figure 5: *RMSE$_t$ (solid line), $\sqrt{P_t^{\mathrm{MEAN}}}$ (dashed line) and $\sqrt{P_t^{\mathrm{CR}}}$ (dash-dotted line) for horizontal velocity $\sqrt{\tilde{v}_n^2 + \tilde{v}_e^2}$ and accelerometer bias $\sqrt{(b_x^a)^2 + (b_y^a)^2}$.*

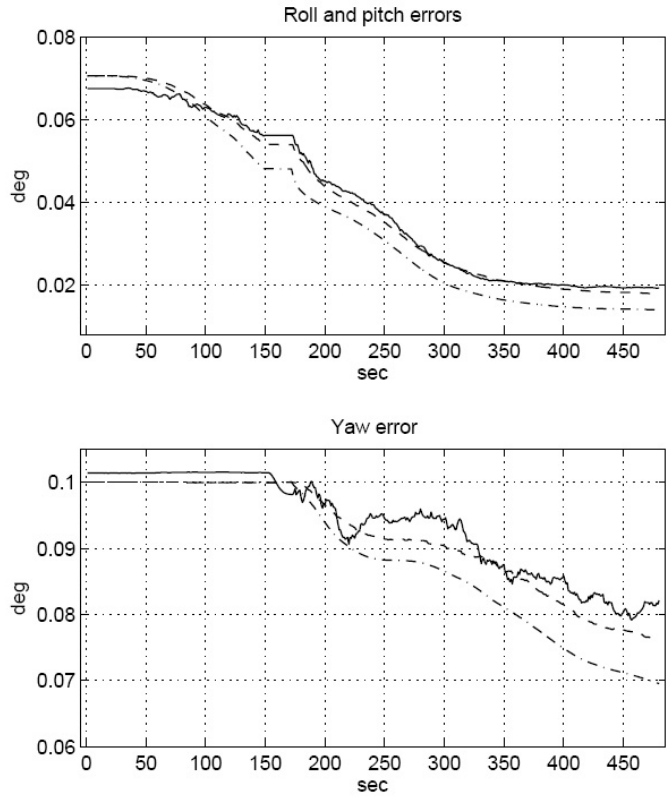

Figure 6: *RMSE$_t$ (solid line), $\sqrt{P_t^{\mathrm{MEAN}}}$ (dashed line) and $\sqrt{P_t^{\mathrm{CR}}}$ (dash-dotted line) for roll and pitch errors $\sqrt{\gamma_n^2 + \gamma_e^2}$ and yaw error γ_d.*

Paper C

Probabilistic Conflict Detection for Piecewise Straight Paths

Authors: Nordlund, P-J. and Gustafsson, F.

Edited version of the paper: P-J. Nordlund and F. Gustafsson. Probabilistic conflict detection for piecewise straight paths. *Submitted to Automatica*, 2008.
http://www.control.isy.liu.se/research/reports/2008/2871.pdf

Preliminary version: Published as Technical Report LiTH-ISY-R-2871, Dept. of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden.

Probabilistic Conflict Detection for Piecewise Straight Paths

Per-Johan Nordlund*, and Fredrik Gustafsson**

*Department of Decision Support
Saab Aerosystems
581 88 Linköping, Sweden
per-johan.nordlund@saabgroup.com

**Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
fredrik@isy.liu.se

Abstract

We consider probabilistic methods for detecting conflicts as a function of predicted trajectory. A conflict is an event representing collision or imminent collision between vehicles or objects. The computations use state estimate and covariance from a target tracking filter based on sensor readings. Existing work is primarily concerned with risk estimation at a certain time instant, while the focus here is to compute the integrated risk over the critical time horizon. This novel formulation leads to evaluating the probability for level-crossing. The analytic expression involves a multi-dimensional integral which is hardly tractable in practice. Further, a huge number of Monte Carlo simulations would be needed to get sufficient reliability for the small risks that the applications often require. Instead, we propose a sound numerical approximation that leads to evaluating a one-dimensional integral which is suitable for real-time implementations.

1 Introduction

Collision or conflict avoidance is a crucial and enabling technology for autonomous vehicles. This is particularly important when autonomous vehicles shall co-exist with manned vehicles in an unregulated environment. Conflict is typically defined as an event where two or more vehicles or objects are closer to each other than given by a safety zone. For air traffic, the extent of the safety zone is defined by authorities [9], [2] and is such that collision is considered imminent if entering the zone. Conflict avoidance can also be utilized in manned vehicles for mitigating or avoiding accidents [12], [8].

This paper describes a method for detecting and avoiding hazardous situations based on uncertain sensor readings. It is assumed that the probability density function for the

state vector describing the relative motion is available. Usually, a tracking filter estimates the state vector, which comprises a (relative) position and velocity in one, two or three dimensions. There are many proposals for solving this problem, see e.g. [17] for a review. Here we adopt a probabilistic point of view, where randomness is a fundamental part all the way from sensor to decision. The majority of probabilistic methods found in literature deal with instantaneous probability of conflict, i.e the probability of conflict at a certain time instant. The time instant could e.g. correspond to the *point of closest approach* [6], [7], [15] or the time instant which maximizes probability of conflict [21]. One problem with instantaneous probability of conflict is how to interpret the result with respect to a predicted, not necessarily straight, trajectory. Note that simply integrating instantaneous probability over time does not yield a correct probability as a function of the time interval. This comes from the fact that the events representing instantaneous conflict are dependent for consecutive time points [16].

In some rare cases there are closed form analytical expressions for the probability of conflict depending on how a conflict is defined and what uncertainties are involved [20]. We propose a novel analytic framework that attacks the, in general, computationally intractable problem of computing the probability of conflict for a given, not necessarily straight, trajectory. Monte-Carlo or sampling methods are known to provide solutions to arbitrary probabilistic problems [13], [24], [22], [21], they are also known to be computer intensive particularly when the underlying probabilities are small [23]. Here we do not rely on Monte-Carlo methods, but instead we make use of theory for stochastic processes and level-crossings. The method is based on the probability density for *time-to-go (ttg)*. Time-to-go is the ratio between distance and closing speed, and is identified as essential for conflict predictions [14]. A level-crossing occurs if the distance perpendicular to line-of-sight is less than a threshold after ttg seconds. A similar approach was applied in [21] for aircraft probability of conflict but for the case of known initial position and velocity. Here we consider the situation with significant initial uncertainties, e.g. as a result of tracking intruders based on angle-only sensors. We consider time horizons up to a couple of minutes, and therefore neglect effects from disturbances on the predicted path. When longer periods of time are considered the effects of for example wind disturbances for aircraft conflict detection can be significant [5].

The result is extended to cover the important case with piecewise linear trajectories, which provides a way forward to deal with arbitrary continuous paths [19]. We focus the presentation to the two-dimensional case, to be able to concentrate on the fundamental ideas.

In Section 2 we formulate the problem mathematically and state the prerequisites to be able to provide a solution. Section 3 details a novel analytic method for computing the probability of conflict when the velocity is constant. A conflict is here defined as the crossing of a line segment. The conditions are based on the probability density for time-to-go. The probability of crossing the line can be expressed as the expected value of the distribution for the distance perpendicular to line-of-sight with respect to the probability density for time-to-go (ttg). Section 4 extends the theory to deal with piecewise linear paths. In Section 5 we give some results comparing the sampling solution with the analytical solution and finally in Section 6 we draw conclusions.

2 Problem formulation

Let $\mathrm{C}_{(0,T)}$ denote the event of a conflict between two objects for the time period $0 < t < T$, i.e. $\mathrm{C}_{(0,T)} = 1$ if a conflict occurs at any time during $0 < t < T$ and $\mathrm{C}_{(0,T)} = 0$ otherwise. If a conflict is to take place or not is a function of future relative position $s(t)$ for $t > 0$, where t is prediction time. Here we define a conflict to occur if the relative position crosses the line segment with endpoints $(0, -h)$ and $(0, h)$, see Figure 1. The

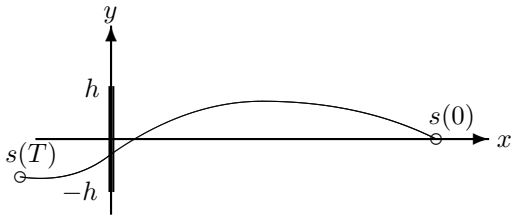

Figure 1: *A conflict occurs if the relative position $s(t)$ crosses the line segment.*

line segment could represent the front/rear end of a car or be the result of approximating the safety zone surrounding an air vehicle. The location and length is a matter of choice, but here we choose to place the line segment at $x = 0$ primarily for notational convenience. While the relative position is a random variable we seek to compute the probability of conflict, $\mathrm{P}\big(\mathrm{C}_{(0,T)}\big)$. The objective is to find an efficient method for computing $\mathrm{P}\big(\mathrm{C}_{(0,T)}\big)$, efficient in the sense that it is computationally tractable for real-time processing.

Let the state vector $x(0)$ be comprised of two-dimensional relative position $s(0)$ and velocity $v(0)$ in Cartesian coordinates. The state vector is rotated so the x-axis is pointing towards the threat, i.e. rotated such that the mean of the initial distance perpendicular to line-of-sight is zero

$$\hat{s}_y(0) = 0. \tag{1}$$

We assume the joint probability density function (pdf) for $x(0)$ is available

$$p_{s_x,v_x,s_y,v_y}(\cdot). \tag{2}$$

Typically, an estimate of $x(0)$ is provided by a tracking filter [4]. Target tracking will not be pursued here in detail, we simply state that based on measurements from a sensor, the tracking filter provides estimates of the state vector $\hat{x}(0)$, together with its covariance $P(0)$, where

$$P(0) = \begin{bmatrix} P_x & P_{xy} \\ P_{xy}^T & P_y \end{bmatrix}. \tag{3}$$

To set explicit expressions we assume the tracking filter output is normally distributed, i.e.

$$x(0) \sim \mathcal{N}(\hat{x}(0), P(0)). \tag{4}$$

It should be stressed that the assumption on a normally distributed state vector is not a requirement for the method to work. In principle, any distribution is applicable as long as the probability density function for $x(0)$ is available.

In the sequel we will leave out the dependency on time when $t = 0$ if unambiguous by the context. Note that we will only deal with a relative time scale, represented by $t = 0$ as the current time on an absolute time scale. At each new time instant on the absolute time scale the tracking filter provides updated estimates $\hat{x}(0)$ and $P(0)$.

3 Crossing in case of constant velocity

Consider the event $\mathrm{C}_{(0,T)}$ when the velocity is constant $v(t) = v(0) = v$. For a $\mathrm{C}_{(0,T)}$ to occur the relative position must cross the line segment. This line segment is orthogonal to the $x-$axis (line-of-sight), see Figure 2.

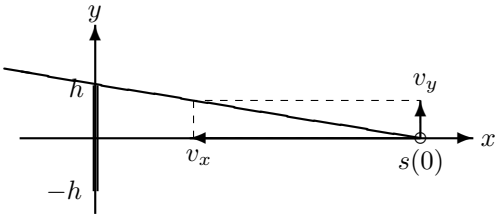

Figure 2: *Geometry for the limit of* $\mathrm{C}_{(0,T)}$ *in two dimensions.*

3.1 Time-to-go

Define a stochastic time variable τ according to

$$\tau = \begin{cases} \frac{s_x(0)}{-v_x} & \text{if } s_x(0) > 0 \ \cap \ v_x < 0, \\ \infty & \text{otherwise.} \end{cases} \tag{5}$$

The τ represents time-to-go, i.e. the time it takes for the relative position to cross the y-axis. A crossing of the $y-$axis occurs within $0 < t < T$ if the closing speed is $-\infty < v_x < 0$ and the initial distance is $0 < s_x < -v_x T$. The probability density for τ is given by Lemma 1.

Lemma 1 (Probability density function for τ)
For $s_x(t)$, $t \in \mathbb{R}_+$ with $\dot{s}_x(t) = v_x$ and a joint probability function for $s_x(0)$ and v_x given by $p_{s_x,v_x}(s, v)$, the probability density function for τ is given by

$$p_\tau(t) = \int_{-\infty}^{0} -v p_{s_x,v_x}(-vt, v)dv, \quad t < \infty. \tag{6}$$

Proof: *See Appendix A.1 or Rice's formula [18].*

With the assumption (4) on normally distributed variables the density $p_\tau(t)$ is given by Corollary 1.

Corollary 1 (Normally distributed $s_x(0)$ and $v_x(0)$)
For $s_x(0)$ and $v_x(0)$ distributed according to

$$\begin{bmatrix} s_x \\ v_x \end{bmatrix} \sim \mathcal{N}\Big(\begin{bmatrix} \hat{s}_x \\ \hat{v}_x \end{bmatrix}, \begin{bmatrix} \sigma_{sx}^2 & \rho_x \sigma_{sx}\sigma_{vx} \\ \rho_x \sigma_{sx}\sigma_{vx} & \sigma_{vx}^2 \end{bmatrix} \Big), \tag{7}$$

the probability density function for τ from (6) *with $\kappa = \rho_x \sigma_{sx}/\sigma_{vx}$ is given by*

$$p_\tau(t) = \frac{g_0}{g_2^2}\left(1 - (2\pi)^{\frac{1}{2}} \frac{g_1}{g_2} e^{\frac{g_1^2}{2g_2^2}} \Phi\big(-\frac{g_1}{g_2}\big)\right), \tag{8}$$

where

$$\begin{aligned} g_0 &= \frac{1}{2\pi\sigma_{sx}\sigma_{vx}(1-\rho_x^2)^{\frac{1}{2}}} e^{-\frac{(\kappa\hat{v}_x - \hat{s}_x)^2}{2\sigma_{sx}^2(1-\rho_x^2)} - \frac{\hat{v}_x^2}{2\sigma_{vx}^2}}, \\ g_1 &= \frac{(t+\kappa)(\kappa\hat{v}_x - \hat{s}_x)}{\sigma_{sx}^2(1-\rho_x^2)} + \frac{\hat{v}_x}{\sigma_{vx}^2}, \\ g_2 &= \left(\frac{(t+\kappa)^2}{\sigma_{sx}^2(1-\rho_x^2)} + \frac{1}{\sigma_{vx}^2}\right)^{\frac{1}{2}}, \end{aligned} \tag{9}$$

and $\Phi(\cdot)$ corresponds to the standard normal distribution

$$\Phi(x) = \int_{-\infty}^{x} \phi(\xi) d\xi = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{\xi^2}{2}} d\xi. \tag{10}$$

Proof: *See [11].*

To compute the one-dimensional normal distribution $\Phi(x)$ the error function in Matlab can be used. If the error function is not available a very accurate result is given by [3]

$$\begin{aligned} &\Phi(x) \approx \\ &\sqrt{\frac{1}{4} - \frac{7e^{-\frac{x^2}{2}} + 16e^{x^2(\sqrt{2}-2)} + (7 + \frac{\pi x^2}{4})e^{-x^2}}{120}} + \frac{1}{2}, \end{aligned} \tag{11}$$

for $x \geq 0$. According to [3] the relative error in (11) is less than 3×10^{-4}.

See Example 1 for an illustration of $p_\tau(t)$.

Example 1: Normal approximation of $p_\tau(t)$

Consider a bearings-only tracking case where the covariance in distance and closing speed is large

$$\begin{aligned} \begin{bmatrix} s_x & v_x \end{bmatrix}^T &= \begin{bmatrix} 1000 & -100 \end{bmatrix}^T, \\ P_x &= \begin{bmatrix} 250^2 & -0.8 \cdot 250 \cdot 25 \\ -0.8 \cdot 250 \cdot 25 & 25^2 \end{bmatrix}. \end{aligned} \tag{12}$$

In Figure 3, $p_\tau(t)$ is compared with the corresponding normal probability density having the same expected value and variance. Here the expected value and standard deviation of τ are 10.2 and 1.86 respectively. As can be seen p_τ is asymmetric with a heavier right tail probability compared to the normal probability.

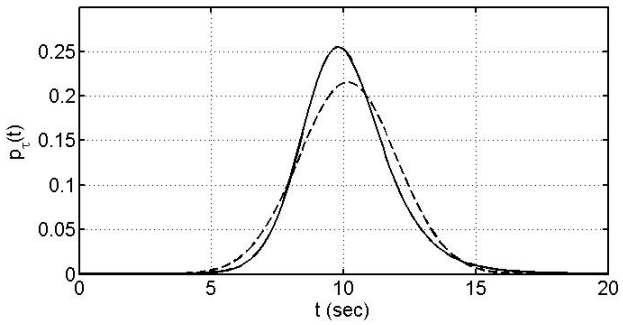

Figure 3: *A comparison between $p_\tau(t)$ (solid line) and the normal probability density with the same mean and variance (dashed line).*

3.2 Conflict probability for a time interval

A conflict will occur if the relative position crosses the line segment, see Figure 2. The condition for conflict is that given $\tau = t < T$, if $|v_y t + s_y|$ is smaller than h, i.e. the distance when crossing the $y-$axis is less than h, there will be a conflict. A motivation of the result when incorporating the fact that we are dealing with random variables is as follows. Assume we know that $\tau = t < T$, and we seek

$$\mathrm{P}\big(|v_y t + s_y| < h\big). \tag{13}$$

Since t is not known but a random variable we replace t with τ, i.e.

$$\mathrm{P}(\mathrm{C}_{(0,T)}) = \mathrm{P}\big(|v_y \tau + s_y| < h \cap \tau < T\big), \tag{14}$$

where the intersection is incorporated because we are only interested in the case $\tau < T$. Split the time interval $(0, T)$ into K subintervals. Events corresponding to different subintervals are mutually exclusive, which means that the probability of conflict is the sum of conflict for each subinterval. For each subinterval $k = 0, \ldots, K-1$ we factorize

$$\begin{aligned} &\mathrm{P}\big(|v_y \tau + s_y| < h \cap T_k < \tau < T_{k+1}\big) = \\ &\mathrm{P}\big(|v_y \tau + s_y| < h \big| T_k < \tau < T_{k+1}\big)\mathrm{P}\big(T_k < \tau < T_{k+1}\big), \end{aligned} \tag{15}$$

where $T_{k+1} - T_k = \Delta T$. Now let $\Delta T \to 0$ which yields

$$\begin{aligned} \mathrm{P}(\mathrm{C}_{(0,T)}) = \int_0^T \mathrm{P}\big(|v_y\tau + s_y| < h | \tau = t\big) p_\tau(t) dt = \\ \int_0^T \mathrm{P}\big(|v_y t + s_y| < h\big) p_\tau(t) dt. \end{aligned} \tag{16}$$

Note that (16) is valid only when $s_x,\ v_x$ are independent of $s_y,\ v_y$. The general case is given by Theorem 1.

Theorem 1 (Conflict for a straight path)
For $s(t) = \begin{bmatrix} s_x(t) & s_y(t) \end{bmatrix}^T$, $t \in \mathbb{R}_+$ *with* $\dot{s}(t) = v$ *and a joint probability function for* $x(0)$ *given by* (2) *the probability of down-crossing a line with end points* $(0, -h)$ *and* $(0, h)$ *within* T *sec is given by*

$$\begin{aligned} \mathrm{P}(\mathrm{C}_{(0,T)}) &= \mathrm{P}\big(|v_y\tau + s_y| < h \cap \tau < T\big) \\ &= \int_0^T \sum_{r=0}^{\infty} \frac{1}{r!} \mathrm{P}_r(\hat{\nu}_x, t) dt. \end{aligned} \tag{17}$$

The terms $P_r(t)$ *are given by the Taylor expansion of* $\mathrm{P}\big(|v_y t + s_y| < h \mid v_x\big)$ *around* $v_x = \hat{\nu}_x$, *i.e.*

$$\begin{aligned} &\mathrm{P}_r(t) = \\ &\left.\frac{\partial^r \mathrm{P}\big(|v_y t + s_y| < h \mid v_x\big)}{\partial v_x^r}\right|_{v_x = \hat{\nu}_x} \left.\frac{\partial^r M(\xi, t)}{\partial \xi^r}\right|_{\xi = 0}. \end{aligned} \tag{18}$$

Here the moment-generating function is defined by

$$M(\xi, t) = \int_{-\infty}^0 -v e^{\xi(v - \hat{\nu}_x)} p_{s_x, v_x}(-vt, v) dv, \tag{19}$$

the conditional expected value of v_x *given* $\tau = t$ *is*

$$\hat{\nu}_x = \hat{\nu}_x(t) = \mathrm{sol}\left\{ \left.\frac{\partial M(\xi, t)}{\partial \xi}\right|_{\xi = 0} = 0 \right\}. \tag{20}$$

Note that $\mathrm{P}_1(t) = 0$ *due to* (20)*, and* $M(0, t) = p_\tau(t)$.
Proof: *See Appendix A.2.*

Under the assumption (4) that $x(0)$ is normally distributed, the probability $\mathrm{P}(|v_y t + s_y| < h \mid v_x = \hat{\nu}_x)$ is given by Corollary 2

Corollary 2 (Conflict for a Gaussian $x(0)$)
For a normally distributed $x(0)$ the probability of conflict from Theorem 1 is given by inserting

$$\begin{aligned}&\mathrm{P}(|v_y t + s_y| < h \,\big|\, v_x = \hat{\nu}_x) = \\ &\Phi\Big(\frac{h - \hat{s}_y(t)}{\sqrt{P_{s_y(t)}}}\Big) - \Phi\Big(\frac{-h - \hat{s}_y(t)}{\sqrt{P_{s_y(t)}}}\Big),\end{aligned} \tag{21}$$

where

$$\begin{aligned}&\hat{s}_y(t) = \hat{s}_y + \hat{v}_y t + \\ &\begin{bmatrix}1\\ t\end{bmatrix}^T P_{xy}^T P_x^{-1}\Big(\begin{bmatrix}-t\\ 1\end{bmatrix}\hat{\nu}_x - \begin{bmatrix}\hat{s}_x\\ \hat{v}_x\end{bmatrix}\Big), \\ &P_{s_y(t)} = \begin{bmatrix}1\\ t\end{bmatrix}^T \Big(P_y - P_{xy}^T P_x P_{xy}\Big)\begin{bmatrix}1\\ t\end{bmatrix},\end{aligned} \tag{22}$$

and

$$\begin{aligned}&M(\xi, t) = \\ &\frac{g_0}{g_2^2} e^{-\hat{\nu}_x \xi}\Big(1 - \frac{g_1 + \xi}{g_2} e^{\frac{(g_1+\xi)^2}{2g_2^2}}\sqrt{2\pi}\Phi\big(-\frac{g_1+\xi}{g_2}\big)\Big), \\ &\hat{\nu}_x = \frac{\frac{g_0}{g_2^3}\Big(\frac{g_1}{g_2} - (1 + \frac{g_1^2}{g_2^2})e^{\frac{g_1^2}{2g_2^2}}\sqrt{2\pi}\Phi\big(-\frac{g_1}{g_2}\big)\Big)}{p_\tau(t)},\end{aligned} \tag{23}$$

and $p_\tau(t)$ from Corollary 1.
Proof: *Replace $p_{s_x,v_x,s_y,v_y}(\cdot)$ with the normal probability density.*

3.3 Monte-Carlo Approximation

The probability according to (17) is in general, e.g. when $x(0)$ is normally distributed, not possible to compute analytically. A straightforward approximate solution is to use a Monte-Carlo method, i.e. to draw N samples of $x(0)$ from (4) and approximate the probability with the outcome of the sampling, i.e.

$$\begin{aligned}&\hat{\mathrm{P}}_{\mathrm{mc}}(\mathbf{C}_{(0,T)}) = \\ &\frac{1}{N}\sum_{i=1}^{N} I\big(|\tau^{(i)} v_y^{(i)} + s_y^{(i)}| < h \cap \tau^{(i)} < T\big),\end{aligned} \tag{24}$$

where $I(\cdot)$ is the indicator function and

$$\tau^{(i)} = \begin{cases} \frac{s_x^{(i)}}{-v_x^{(i)}} & \text{if } s_x^{(i)} > 0 \ \cap \ v_x^{(i)} < 0, \\ \infty & \text{otherwise.}\end{cases} \tag{25}$$

Denote the true value of the sought probability with p. The set of samples is binomially distributed, $\text{Bin}(N, p)$, but for a large enough N, usually $Np(1-p) > 20$ is sufficient, the probability is approximated well by [10]

$$\begin{aligned} &\hat{\text{P}}_{\text{mc}}(\text{C}_{(0,T)}) \sim \mathcal{N}(p, \sigma^2), \\ &\sigma^2 = \frac{p(1-p)}{N}. \end{aligned} \tag{26}$$

For a relative mean square error $\varepsilon_{\text{rel}} \leq \frac{\sigma}{p}$ we can compute needed number of samples according to

$$N \geq \frac{1-p}{\varepsilon_{\text{rel}}^2 p} \approx \frac{1}{\varepsilon_{\text{rel}}^2 p}, \tag{27}$$

where the last approximation is valid for small p. Assume $p = 0.01$ and $3\varepsilon_{\text{rel}} \leq 0.1$, i.e a relative error smaller than 10% with probability 0.997. These values plugged into (27) suggests that we must use $N \geq 90000$. For many on-line applications this means a too high computational load.

3.4 Numerical Approximation

To be able to compute probability of conflict according to (17) the Taylor expansion has to be truncated. For a given accuracy level ε we need to find a R such that

$$\begin{aligned} &\int_0^T \sum_{r=R+1}^{\infty} \frac{1}{r!} \text{P}_r(\hat{\nu}_x, t) dt = \\ &\int_0^T \frac{1}{(R+1)!} \text{P}_{R+1}(\mu, t) dt < \varepsilon, \end{aligned} \tag{28}$$

where the equality is valid for some $-\infty < \mu = \mu(t) < 0$. In the general case the rest term is more or less cumbersome to analyze and we usually have to resort to simulations. Recall that we apply a Taylor expansion due to $P_{xy} \neq 0$. The weaker correlation the fewer terms we need in the expansion.

A sound and simple numerical approximation for computing a one-dimensional integral is given by Simpson's rule [1]

$$\begin{aligned} &\int_{t_0}^{t_K} f(t) dt = F(t_0, t_K; K, f(t)) + R_K \\ &= \frac{\Delta t}{3} \left(f(t_0) + 2 \sum_{k=1}^{K/2-1} f(t_{2k}) + 4 \sum_{k=1}^{K/2} f(t_{2k-1}) + f(t_K) \right) + R_K, \end{aligned} \tag{29}$$

where $\Delta t = (t_K - t_0)/K$ and $t_k = k\Delta t + t_0$. From [1] we know that the approximation error is bounded by

$$R_K < \frac{\Delta t^5}{90} \sum_{k=0}^{K/2-1} \max_{t_{2k} < t < t_{2k+2}} \left| \frac{\partial^4 f}{\partial t^4} \right|. \tag{30}$$

This can used for computing $\mathrm{P}(\mathrm{C}_{(0,T)})$ according to

$$\hat{\mathrm{P}}_{\mathrm{simp}}(\mathrm{C}_{(0,T)}) = F(0,T;K,f(t)), \tag{31}$$

with

$$f(t) = \sum_{r=0}^{R} \frac{1}{r!} \mathrm{P}_r(\hat{\nu}_x, t). \tag{32}$$

Simpson's rule provides a well-known and accurate method for computing (17), although in some cases, e.g. when the variables are normally distributed with small variances, the higher derivatives of $\mathrm{P}(|v_y t + s_y| < h)p_\tau(t)$ can be large. For an illustration see Example 2

Example 2: Two methods for $\mathrm{P}(\mathrm{C}_{(0,T)})$

Consider again the bearings-only case as in Example 1 where $h = 150$ and

$$\begin{aligned} \hat{x}(0) &= \begin{bmatrix} 1000 & -100 & 0 & 10 \end{bmatrix}^T, \\ P(0) &= \begin{bmatrix} \frac{1000^2}{\gamma^2} & -0.8\frac{1000\cdot 100}{\gamma^2} & 0 & 0 \\ -0.8\frac{1000\cdot 100}{\gamma^2} & \frac{1000^2}{\gamma^2} & 0 & 0 \\ 0 & 0 & 2^2 & 0 \\ 0 & 0 & 0 & 4^2 \end{bmatrix}. \end{aligned} \tag{33}$$

Here the correlation matrix P_{xy} is zero and the terms $\mathrm{P}_r(\hat{\nu}_x, t)$ in (17) are therefore zero for $r \geq 1$. A comparison between the Monte-Carlo solution (24) with $N = 1000000$ and the numerical approximation (31) with $\Delta t = 0.5$ is shown in Figure 4 for $\gamma = 4$ and $\gamma = 8$. For comparison we have also computed an instantaneous probability of conflict as a function of time according to

$$\begin{aligned} &\mathrm{P}(\mathrm{C}_t) = \mathrm{P}\big(\sqrt{s_x^2(t) + s_y^2(t)} < h\big) \approx \\ &\frac{1}{M} \sum_{i=1}^{M} I\bigg(\sqrt{\big(s_x^{(i)}(t)\big)^2 + \big(s_y^{(i)}(t)\big)^2} < h\bigg), \end{aligned} \tag{34}$$

with $M = 100000$, plotted in Figure 4 with sampling time 0.1 sec. As described in Section 1 the instantaneous probability of conflict is not straightforward to interpret with respect to a predicted trajectory. From Figure 4 we conclude that $\mathrm{P}(\mathrm{C}_t)$ is highly dependent on estimated covariance $P(0)$. The accuracy (3σ) for the Monte Carlo solution is for $p = 0.85$ approximately

$$\varepsilon_{\mathrm{mc}} = 3\sqrt{\frac{p(1-p)}{N}} \approx 1 \cdot 10^{-3}. \tag{35}$$

The actual difference between Monte Carlo and numerical approximation according to (29) is less than $1 \cdot 10^{-3}$, meaning that (31) yields a result which is at least as good as the sampling method. This is confirmed by analysing the error given by (30). Computing

an approximate fourth derivative of the integrand for $\gamma = 8$, which is the worst case, and finding the maximum for each interval $(t_{2k} < t < t_{2k+2})$ yields

$$\varepsilon_{\text{simp}} < \frac{0.5^5}{90} \sum_{k=0}^{32-1} \max_{t_{2k}<t<t_{2k+2}} \left| \frac{\partial^4 f}{\partial t^4} \right| \approx 1.5 \cdot 10^{-3}. \tag{36}$$

To summarize, in this case we obtain equal accuracy from Simpson's rule compared to the Monte Carlo solution. At the same time, comparison in Matlab shows that the computing time is about 1000 times less using the implementation of Theorem 1 instead of the Monte-Carlo solution.

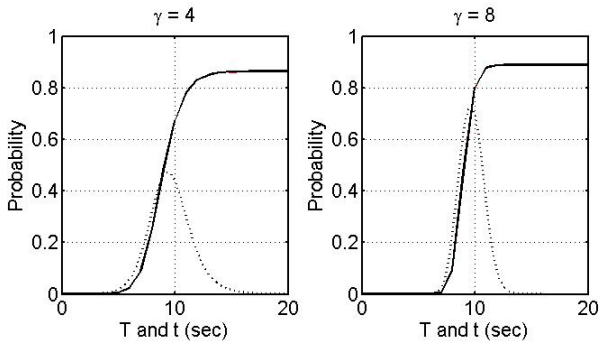

Figure 4: *The left plot shows the result for* $\gamma = 4$ *using Simpson's rule* (31) *with* $\Delta t = 0.5$ *(solid line), the Monte-Carlo solution* (24) *with* $N = 1000000$ *(dashed line) and* $\mathrm{P}(\mathrm{C}_t)$ (34) *(dotted line). The right plot compares the same solutions but for* $\gamma = 8$*. The difference between the numerical solution* (31) *and the Monte-Carlo solution* (24) *is too small to be visible.*

4 Conflict in case of piecewise constant velocity

From now on we assume the relative position follows a piecewise straight path given by

$$\dot{s}(t) = v^{(j)}, \quad T_j < t < T_{j+1}, \tag{37a}$$

$$v^{(j)} = v(0) + \sum_{l=1}^{j} \Delta v^{(l)}, \tag{37b}$$

where all $\Delta v^{(j)}$ are known. Note that the model in (37) is fairly general because we can approximate any curved path arbitrarily well as long as we use a large enough number of straight segments.

4.1 Crossing of the line segment

According to (37) we assume piecewise constant velocity $v(t) = v^{(j)}$ for $T_j < t < T_{j+1}$. The event $\mathrm{C}_{(0,T)}$ can always be expressed according to

$$\mathrm{C}_{(0,T)} = \cup_{j=0}^{J-1} \mathrm{C}_{(T_j,T_{j+1})}, \tag{38}$$

i.e. the union of events for each segment $T_j < t < T_{j+1}$ with $T_0 = 0$ and $T_J = T$. A geometric interpretation of $\mathrm{C}_{(T_j,T_{j+1})}$ in two dimensions is given by Figure 5. From

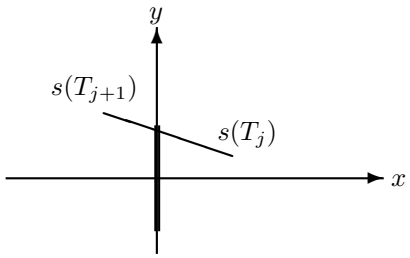

Figure 5: *Geometry for* $\mathrm{C}_{(T_j,T_{j+1})}$ *in two dimensions.*

basic probability theory we know that

$$\mathrm{P}(\cup_{j=0}^{J-1} \mathrm{C}_{(T_j,T_{j+1})}) \leq \sum_{j=0}^{J-1} \mathrm{P}(\mathrm{C}_{(T_j,T_{j+1})}). \tag{39}$$

An important observation is that if it is unlikely that $v_x^{(j)}$ changes sign, i.e.

$$\mathrm{P}\big(\mathrm{sign}(v_x^{(j)}) = \mathrm{sign}(v_x^{(0)})\big) \approx 1, \tag{40}$$

then $\mathrm{C}_{(T_j,T_{j+1})}$ for different j's are mutually exclusive. With this assumption we can write

$$\mathrm{P}(\mathrm{C}_{(0,T)}) = \sum_{j=0}^{J-1} \mathrm{P}(\mathrm{C}_{(T_j,T_{j+1})}). \tag{41}$$

The change to be incorporated for $\mathrm{C}_{(T_j,T_{j+1})}$ compared to the conditions for the constant velocity case is caused by considering $t = T_j$ instead of $t = 0$ as the starting time. This yields changing variables to $s(T_j)$ and $v^{(j)}$. It is possible to derive conditions for $\mathrm{C}_{(T_j,T_{j+1})}$ using $s(T_j)$. However, a more efficient way is to consider the distance obtained by extrapolating from $s(T_j)$ backwards in time T_j sec using the current velocity $v^{(j)}$. This yields the distance at $t = 0$ which would give $s(T_j)$ after T_j sec using a constant velocity $v^{(j)}$. This means that we are back to starting time $t = 0$ and it enables us to reuse results from the case with constant velocity. Denote the distance obtained from extrapolation by $s^{(j)}$, and we have

$$s^{(j)} = s(T_j) - T_j v^{(j)}. \tag{42}$$

Another advantage using $s^{(j)}$ instead of $s(T_j)$ comes from the fact that the covariance of $s^{(j)}$ is equal to the covariance of $s(0)$ for all $j = 0, \ldots, J-1$. This is clear from the expression

$$s^{(j)} = s(0) - \sum_{l=1}^{j} T_l \Delta v^{(l)}, \tag{43}$$

which is derived in Appendix A.3.

Now we can define a random variable $\tau^{(j)}$ according to

$$\tau^{(j)} = \begin{cases} \frac{s_x^{(j)}}{-v_x^{(j)}} & \text{if } s_x^{(j)} > 0 \cap v_x^{(j)} < 0, \\ \infty & \text{otherwise,} \end{cases} \tag{44}$$

and the distribution for $\tau^{(j)}$ is given by Corollary 3.

Corollary 3 (Probability density for $\tau^{(j)}$)
For $s_x^{(j)}(t)$, with $\dot{s}_x^{(j)}(t) = v_x^{(j)}$ and a joint probability function for $s_x^{(j)}$ and $v_x^{(j)}$ given by $p_{s_x^{(j)}, v_x^{(j)}}(s, v)$, the density for $\tau^{(j)}$ is given by

$$p_{\tau^{(j)}}(t) = \int_{-\infty}^{0} -v p_{s_x^{(j)}, v_x^{(j)}}(-vt, v) dv \quad t < \infty. \tag{45}$$

Proof: *See Appendix A.1 with $s_x(0)$ and v_x replaced by $s_x^{(j)}$ and $v_x^{(j)}$.*

In the case where $x(0)$ is normally distributed, the probability density $p_{\tau^{(j)}}(t)$ is given by Corollary 1 with $x(0)$ replaced by $x^{(j)}$.

The probability of $\mathrm{C}_{(0,T)}$ is now given by Corollary 4.

Corollary 4 (Conflict for a piecewise straight path)
For $s^{(j)}(t) = \begin{bmatrix} s_x^{(j)}(t) & s_y^{(j)}(t) \end{bmatrix}^T$, with $\dot{s}_x^{(j)}(t) = v_x^{(j)}$ and a joint probability function for $x^{(j)}$ given by $p_{s_x^{(j)}, v_x^{(j)}, s_y^{(j)}, v_y^{(j)}}(\cdot)$, the probability of down-crossing a line with end points $(0, -h)$ and $(0, h)$ within T sec is given by

$$\begin{aligned} \mathrm{P}(\mathrm{C}_{(0,T)}) &= \sum_{j=0}^{J-1} \mathrm{P}(\mathrm{C}_{(T_j, T_{j+1})}) \\ &= \sum_{j=0}^{J-1} \int_{T_j}^{T_{j+1}} \sum_{r=0}^{\infty} \frac{1}{r!} \mathrm{P}_r^{(j)}(\hat{\nu}_x^{(j)}, t) dt. \end{aligned} \tag{46}$$

The terms $\mathrm{P}_r^{(j)}(\hat{\nu}_x^{(j)}, t)$, $M^{(j)}(\xi, t)$ and $\hat{\nu}_x^{(j)}$ are given by (18)*,* (19) *and* (20) *respectively with $s^{(j)}$ from* (43) *and $v^{(j)}$ from* (37b) *inserted instead of $s(0)$ and $v(0)$.*
Proof: *See Appendix A.2 with s_x, v_x, s_y, v_y replaced by $s_x^{(j)}$, $v_x^{(j)}$, $s_y^{(j)}$, $v_y^{(j)}$.*

Under the assumption that $x(0)$ is normally distributed, the probability $\mathrm{P}\big(|v_y^{(j)} t + s_y^{(j)}| < h \,\big|\, v_x^{(j)} = \hat{\nu}_x^{(j)}\big)$ is given by Corollary 2 with $x(0)$ replaced by $x^{(j)}$.

4.2 Implementation

Two different methods for computing the probability of conflict are given below. The first is the Monte-Carlo implementation as given by Algorithm 1 and the second is the implementation of Corollary 4 as given by Algorithm 2.

Algorithm 1 (Monte-Carlo with accuracy ε).

- Choose N such that $N \geq \frac{9p(1-p)}{\varepsilon^2}$.
- Draw N samples, $x^{(i)}(0) \sim \mathcal{N}(\hat{x}(0), P(0))$.
- For $i = 1, \ldots, N$: Compute

$$\mathrm{C}_{(0,T)}^{(i)} = I\bigg(\cup_{j=0}^{J-1} \left(|\tau^{(i,j)} v_y^{(i,j)} + s_y^{(i,j)}| < h \cap \tau^{(i,j)} < T \right) \bigg), \tag{47}$$

  where

$$\tau^{(i,j)} = \begin{cases} \frac{s_x^{(i,j)}}{-v_x^{(i,j)}} & \text{if } s_x^{(i,j)} > 0 \ \cap \ v_x^{(i,j)} < 0, \\ \infty & \text{otherwise,} \end{cases} \tag{48}$$

  and

$$\begin{aligned} s^{(i,j)} &= s^{(i)}(0) - \sum_{l=1}^{j} T_l \Delta v^{(l)}, \\ v^{(i,j)} &= v^{(i)}(0) + \sum_{l=1}^{j} \Delta v^{(l)}. \end{aligned} \tag{49}$$

- Compute the probability of conflict

$$\hat{\mathrm{P}}_{\mathrm{mc}}(\mathrm{C}_{(0,T)}) = \frac{1}{N} \sum_{i=1}^{N} \mathrm{C}_{(0,T)}^{(i)}. \tag{50}$$

Algorithm 2 (Corollary 4 with accuracy ε).

- For $j = 0, \ldots, J-1$: Choose Δt_j and R such that

$$\frac{\Delta t_j^5}{90} \sum_{k=0}^{K_j/2-1} \max_{t_{2k} < t < t_{2k+2}} \left| \frac{\partial^4 f}{\partial t^4} \right| < \frac{\varepsilon}{J}, \tag{51}$$

  where $\Delta t_j = \frac{T_{j+1} - T_j}{K_j}$ and $f(t) = \sum_{r=0}^{R} \frac{1}{r!} \mathrm{P}_r^{(j)}(\hat{\nu}_x^{(j)}, t)$
- Compute the probability for each segment $j = 0, \ldots, J-1$ using (29)

$$\hat{\mathrm{P}}_{\mathrm{simp}}(\mathrm{C}_{(T_j, T_{j+1})}) = F(T_j, T_{j+1}; K_j, f(t)). \tag{52}$$

- Compute the total probability of conflict

$$\hat{\mathrm{P}}_{\mathrm{simp}}(\mathrm{C}_{(0,T)}) = \sum_{j=0}^{J-1} \hat{\mathrm{P}}_{\mathrm{simp}}(\mathrm{C}_{(T_j, T_{j+1})}). \tag{53}$$

5 Simulation results

There are two simulated bearings-only tracking scenarios, the first one with values according to

$$\hat{x}(0) = \begin{bmatrix} 1000 & -100 & 0 & 10 \end{bmatrix}^T,$$
$$P(0) = \begin{bmatrix} \frac{10^6}{\gamma^2} & -\frac{0.8\cdot 10^5}{\gamma^2} & 0 & -\frac{0.3\cdot 4000}{\gamma} \\ -\frac{0.8\cdot 10^5}{\gamma^2} & \frac{10^4}{\gamma^2} & 0 & \frac{0.3\cdot 400}{\gamma} \\ 0 & 0 & 2^2 & 0 \\ -\frac{0.3\cdot 4000}{\gamma} & \frac{0.3\cdot 400}{\gamma} & 0 & 4^2 \end{bmatrix}. \tag{54}$$

with a scale factor $\gamma = 4$, and the second with the same values as in (54) but with a scale factor set to $\gamma = 8$. A turn of -50 deg is performed in both cases after 3 sec. Absolute quantities are given in Figure 6.

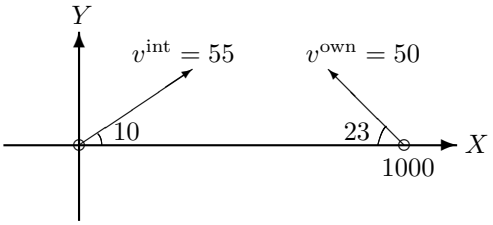

Figure 6: *Absolute geometry for own and intruder (int) vehicle, with angles in deg, speeds in m/s and distance in m.*

Methods for evaluation, i.e. compared to the truth given by the Monte Carlo solution according to Algorithm 1 using $N = 1000000$, are:

- Algorithm 2 using $R = 2$, i.e. two terms in the Taylor expansion, and $\Delta t = 0.5$.
- Algorithm 2 using $R = 1$, i.e. one term in the Taylor expansion, and $\Delta t = 0.5$.

For comparison we have included the instantaneous probability of conflict given by (34) again with $M = 100000$. The probability $\mathrm{P}(\mathrm{C}_t)$ is plotted in Figure 7 with sampling time 0.1 sec.

As can be deduced from Figure 7, Algorithm 2 with $R = 2$ gives a close to identical result compared to the Monte Carlo solution in both scenarios, the relative error is about 0.1% in both cases. Algorithm 2 with $R = 1$ gives a relative error about 2% for the first scenario. For the second scenario the relative error increases to about 6%. The comparison with instantaneous probability of conflict from (34) indicates that the interpretation of $\mathrm{P}(\mathrm{C}_t)$ is not straightforward. First of all, as noted in Section 1, $\mathrm{P}(\mathrm{C}_t)$ yields the conflict risk for a certain point in time, it does not provide the risk for an entire trajectory. Moreover, when we halve σ_{sx} and σ_{vx}, i.e. γ changes from 4 to 8, $\max_t \mathrm{P}(\mathrm{C}_t)$ decreases with 17% while $\mathrm{P}(\mathrm{C}_{(0,T)})$ decreases with 33%. That is, the decrease is more distinct for $\mathrm{P}(\mathrm{C}_{(0,T)})$ than for $\max_t \mathrm{P}(\mathrm{C}_t)$. In this case the more the probability decreases the better because the true minimum distance is 189 m which is outside the range of conflict.

To summarize, in this case we obtain a very accurate result from Algorithm 2. At the same time, the computation time in Matlab is about 1000 times less using Algorithm 2 with $R = 2$ instead of Algorithm 1.

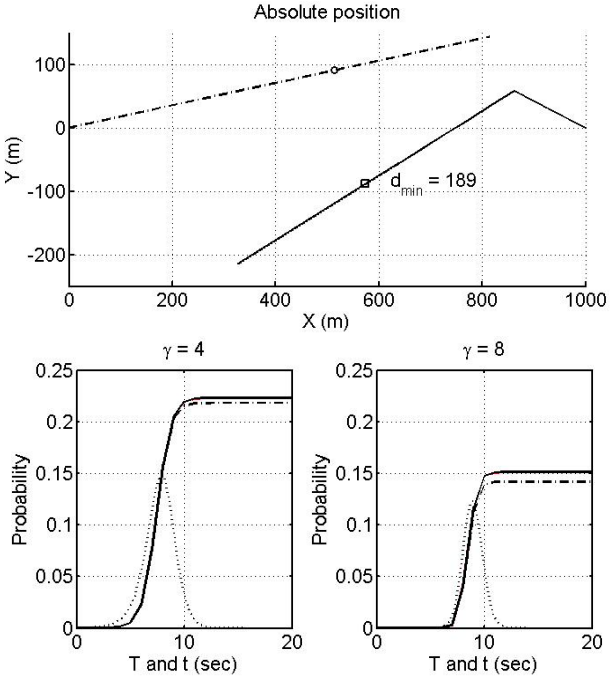

Figure 7: *The upper plot shows the nominal position of the intruder (dash-dotted line) and the own vehicle (solid line). The circle and square denote the positions of the intruder and own vehicle respectively at the point of closest approach. An instantaneous turn of -50 deg is performed after 3 sec. The lower plots show the result using Algorithm 2 with $R = 2$ (solid line), Algorithm 2 with $R = 1$ (dash-dotted line), the Monte-Carlo solution from Algorithm 1 with $N = 1000000$ (dashed line) and $\mathrm{P}(\mathrm{C}_t)$ from* (34) *(dotted with points). Simulation parameters are taken from* (54) *with $\gamma = 4$ (lower left plot) and $\gamma = 8$ (lower right plot).*

6 Conclusions

This paper presents a novel solution to the probability of conflict for a predicted relative trajectory. The method does not rely on sampling techniques, but defining conflict as the crossing of a line segment enable us to derive an analytical expression for the probability of conflict. The analytical expression is a one-dimensional integral which is computed using Simpson's formula. The computation time using the novel solution is 3 orders of magnitude less compared to a sampling based method.

A Appendix

A.1 Proof of Lemma 1

Since $\tau \geq 0$ is true we can write

$$\mathrm{P}(\tau < T) = \mathrm{P}(0 < \tau < T). \tag{55}$$

The probability can be divided into the two mutually exclusive events $v_x < 0$ and $v_x > 0$, i.e.

$$\begin{aligned} &\mathrm{P}(0 < \tau < T) = \mathrm{P}(0 < \tau < T \cap v_x < 0) \\ &+ \mathrm{P}(0 < \tau < T \cap v_x > 0). \end{aligned} \tag{56}$$

From the definition of τ we have that $\tau = \infty$ for $v_x > 0$, which means that the second probability in (56) is zero for finite T and

$$\mathrm{P}(0 < \tau < T) = \mathrm{P}(0 < \tau < T \cap v_x < 0). \tag{57}$$

Inserting $\tau = \frac{s_x}{-v_x}$ yields

$$\begin{aligned} \mathrm{P}(\tau < T) &= \mathrm{P}(0 < s_x < -v_x T \cap v_x < 0) \\ &= \int_{-\infty}^{0} \int_{0}^{-vT} p_{s_x,v_x}(s,v) ds dv. \end{aligned} \tag{58}$$

The probability density for τ for $t < \infty$ is given by

$$p_\tau(t) = \frac{d}{dt}\mathrm{P}(\tau < t) = \int_{-\infty}^{0} -v p_{s_x,v_x}(-vt, v) dv. \tag{59}$$

A.2 Proof of Theorem 1

Using the joint probability density for s_x, v_x, s_y, v_y,

$$\begin{aligned} &p_{s_x,v_x,s_y,v_y}(s,v,y,z) = \\ &p_{s_y,v_y|s_x,v_x}(y,z) p_{s_x,v_x}(s,v), \end{aligned} \tag{60}$$

we have

$$\mathrm{P}(\mathrm{C}_{(0,T)}) = \int_{-\infty}^{0} \int_{0}^{-vT} F(s,v) p_{s_x,v_x}(s,v) ds dv \tag{61}$$

where

$$\begin{aligned} F(s,v) &= \mathrm{P}\big(|s_y + v_y t| < h | s_x = s, v_x = v\big) \\ &= \iint_{|y+tz|<h} p_{s_y,v_y|s_x,v_x}(y,z) dy dz. \end{aligned} \tag{62}$$

Note that $t = \frac{s}{-v}$ in (62). Now changing integration variable from s to $t = \frac{s}{-v}$ yields

$$\begin{aligned} &\mathrm{P}(\mathrm{C}_{(0,T)}) = \\ &\int_{-\infty}^{0} \int_{0}^{T} -vF(-vt, v) p_{s_x, v_x}(-vt, v) dt dv. \end{aligned} \tag{63}$$

Taylor expansion of $F(-vt, v)$ around $\hat{\nu}_x$ yields

$$\begin{aligned} F(-vt, v) &= \mathrm{P}\big(|s_y + v_y t| < h | s_x = -vt, v_x = v\big) \\ &= F(-\hat{\nu}_x t, \hat{\nu}_x) + \sum_{r=1}^{\infty} \frac{1}{r!} \frac{\partial^r F(-vt, v)}{\partial v^r}\bigg|_{v=\hat{\nu}_x} (v - \hat{\nu}_x)^r \end{aligned} \tag{64}$$

Inserting into (63) yields

$$\begin{aligned} \mathrm{P}(\mathrm{C}_{(0,T)}) &= \int_{0}^{T} F(-\hat{\nu}_x t, \hat{\nu}_x) p_\tau(t) dt \\ &+ \int_{0}^{T} \sum_{r=1}^{\infty} \frac{1}{r!} \frac{\partial^r F(-vt, v)}{\partial v^r}\bigg|_{v=\hat{\nu}_x} \\ &\cdot \int_{-\infty}^{0} -v(v - \hat{\nu}_x)^r p_{s_x, v_x}(-vt, v) dv dt. \end{aligned} \tag{65}$$

Choose $\hat{\nu}_x = \hat{\nu}_x(t)$ such that

$$\int_{-\infty}^{0} -v(v - \hat{\nu}_x) p_{s_x, v_x}(-vt, v) dv = 0, \tag{66}$$

which means that the first order term corresponding to $r = 1$ in (65) disappears. Now define a moment-generating function according to

$$M(\xi, t) = \int_{-\infty}^{0} -ve^{\xi(v - \hat{\nu}_x)} p_{s_x, v_x}(-vt, v) dv, \tag{67}$$

and the result follows.

A.3 Derivation of (43)

Using $v^{(j)} = v^{(l)} + \sum_{r=l+1}^{j} \Delta v^{(r)}$, the stochastic variable $s(T_j)$ can be written according to

$$\begin{aligned} s(T_j) &= s(0) + \sum_{l=0}^{j-1} (T_{l+1} - T_l) v^{(l)} \\ &= s(0) + \sum_{l=0}^{j-1} (T_{l+1} - T_l)(v^{(j)} - \sum_{r=l+1}^{j} \Delta v^{(r)}) \\ &= s(0) + T_j v^{(j)} - \sum_{l=1}^{j} T_l \Delta v^{(l)}. \end{aligned} \tag{68}$$

Denote $s^{(j)} = s(0) - \sum_{l=1}^{j} T_l \Delta v^{(l)}$, which yields $s(T_j) = s^{(j)} + T_j v^{(j)}$.

References

[1] M. Abramowitz and I.A. Stegun, editors. *Handbook of Mathematical Functions With Formulas, Graphs and Mathematical Tables*. US Department of Commerce, 10 edition, 1964.

[2] ASTM. *F2411-07 Standard Specification for Design and Performance of an Airborne Sense-and-Avoid System*, 2007.

[3] R.J. Bagby. Calculating normal probabilities. *The American Mathematical Monthly*, 102(1):46–49, Jan 1995.

[4] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[5] G. Chaloulos and J. Lygeros. Effect of wind correlation on aircraft conflict probability. *Journal of Guidance, Control, and Dynamics*, 30(6), 2007.

[6] K. Chan. Analytical expressions for computing spacecraft collision probabilities. In *Proceedings of the 11th Annual AAS/AIAA Space Flight/Mechanics Meeting*, 2001.

[7] K. Chan. Improved analytical expressions for computing spacecraft collision probabilities. *Advances in the Astronautical Sciences*, 114, Part II: Spaceflight Mechanics, 2003.

[8] A. Eidehall, J. Pohl, F. Gustafsson, and J. Ekmark. Toward autonomous collision avoidance by steering. *IEEE Transactions on Intelligent Transportation Systems*, 8(1), 2007.

[9] FAA. *Aeronautical Information Manual - Official Guide to Basic Flight Information and ATC Procedures*, 2008.

[10] A. Gut. *An Intermediate Course in Probability*. Springer-Verlag, 1995.

[11] D.V. Hinkley. On the ratio of two correlated normal random variables. *Biometrika*, 56(3):635–639, 1969.

[12] J. Jansson. *Collision Avoidance Theory with Applications to Automotive Collision Mitigation*. PhD thesis 950, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 2005.

[13] J. Jansson and F. Gustafsson. A framework and automotive application of collision avoidance decision making. *Automatica*, 2008.

[14] M. J. Kochenderfer, J. D. Griffith, and J. K. Kuchar. Hazard alerting using line-of-sight rate. In *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008.

[15] J. Krozel and M. Peters. Strategic conflict detection and resolution for free flight. In *Proceedings of the 36th IEEE Conference on Decision and Control*, 1997.

[16] J.K. Kuchar. *A unified methodology for the evaluation of hazard alerting systems*. PhD thesis, Massachusetts Inst. of Technology, Dept. of Aeronautics and Astronautics, Cambridge, MA, 1995.

[17] J.K. Kuchar and L.C. Yang. A review of conflict detection and resolution methods. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):179–189, 2000.

[18] G. Lindgren. Lectures on stationary stochastic processes. http://www.maths.lth.se/matstat/staff/georg/Publications/lecture.pdf, 2004.

[19] R. A. Paielli. Algorithms for tactical conflict resolution and strategic conflict probability reduction. In *Proceedings of the 1st AIAA Aircraft, Technology Integration and Operations Forum*, 2001.

[20] R.A. Paielli and H. Erzberger. Conflict probability estimation for free flight. *Journal of Guidance, Control and Dynamics*, 20(3):588–596, 1997.

[21] M. Prandini, J Hu, J Lygeros, and S. Sastry. A probabilistic approach to aircraft conflict detection. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):199–220, 2000.

[22] M. Prandini, J. Lygeros, A. Nilim, and S. Sastry. Randomized algorithms for probabilistic aircraft conflict detection. In *Proceedings of the 38th Conference on Decision and Control*, 1999.

[23] L. Yang, J. H. Yang, J. Kuchar, and E. Feron. A real-time Monte Carlo implementation for computing probability of conflict. In *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, 2004.

[24] L.C. Yang and J.K. Kuchar. Prototype conflict alerting system for free flight. *Journal of Guidance, Control and Dynamics*, 20(4):768–773, 1997.

Paper D

Probabilistic Near Mid-Air Collision Avoidance

Authors: Nordlund, P-J. and Gustafsson, F.

Edited version of the paper: P-J. Nordlund and F. Gustafsson. Probabilistic near mid-air collision avoidance. *Submitted to IEEE Transactions on Aerospace and Electronic Systems*, 2008.
http://www.control.isy.liu.se/research/reports/2008/2872.pdf

Preliminary version: Published as Technical Report LiTH-ISY-R-2872, Dept. of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden.

Probabilistic Near Mid-Air Collision Avoidance

Per-Johan Nordlund*, and Fredrik Gustafsson**

*Department of Decision Support
Saab Aerosystems
581 88 Linköping, Sweden
per-johan.nordlund@saabgroup.com

**Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
fredrik@isy.liu.se

Abstract

We propose a probabilistic method to compute the near mid-air collision risk as a function of predicted flight trajectory. The computations use state estimate and covariance from a target tracking filter based on angle-only sensors such as digital video cameras. The majority of existing work is focused on risk estimation at a certain time instant. Here we derive an expression for the integrated risk over the critical time horizon. This is possible using probability for level-crossing, and the expression applies to a three-dimensional piecewise straight flight trajectory. The Monte Carlo technique provides a method to compute the probability, but a huge number of simulations is needed to get sufficient reliability for the small risks that the applications require. Instead we propose a method which through sound geometric and numerical approximations yield a solution suitable for real-time implementations. The algorithm is applied to realistic angle-only tracking data, and shows promising results when compared to the Monte Carlo solution.

1 Introduction

To maintain a safe distance between each other, manned aircraft flying in controlled airspace use the service provided by an Air Traffic Control (ATC). ATC informs and orders human pilots to perform maneuvers in order to avoid Near Mid-Air Collisions (NMAC). A NMAC between two aircraft occurs if the relative distance between them becomes less than a predefined distance [9, 4]. The last decade semi-automatic systems such as TCAS (Traffic Collision Avoidance System) [15] have been implemented that essentially move this responsibility from ATC to the pilot. The TCAS system, however, assumes that both aircraft exchange data on speed, height and bearing over a data link and that both systems cooperate. When operating small UAVs this assumption is often no

longer valid. A typical UAV operates on altitudes where small intruding aircraft are often present that do not carry transponders.

This paper presents a method for detecting and avoiding hazardous situations based on uncertain sensor readings. There are many proposals for solving this problem, see e.g. [16] for a review. Here we consider data from a passive angle-only sensor. A challenge with angle-only measuring sensors is how to deal with the significant uncertainty obtained in estimated distance and relative speed. One approach to increase accuracy in the distance estimate is to perform own platform maneuvers [21]. The method in this paper does not rely on accurate distance estimates. The proposed method is based on computing the probability of NMAC for a predicted trajectory. The majority of existing methods are based on instantaneous probability of NMAC [8, 14, 20]. Instantaneous probability corresponds to the probability that the relative position is within a predefined volume at a certain time instant. It is not straightforward how to interpret instantaneous probability of NMAC with respect to an entire future trajectory.

We present an approximate solution to the in general computationally intractable problem of computing the probability of NMAC for a predicted trajectory. Although Monte-Carlo methods are known to be able to approximate probabilities arbitrarily well [23, 13], they are also known to be computer intensive particularly when the underlying probabilities are small. Here we do not rely on Monte-Carlo methods, but instead we make use of theory for stochastic processes and level-crossings as in [17]. The event corresponding to NMAC can be seen as the crossing of a safety sphere surrounding the vehicle. The concept of avoiding the safety sphere is also adopted in [6] but confined to deterministic trajectories. We derive expressions for the probability that the relative trajectory will cross the safety sphere, both for straight and piecewise straight trajectories. By appropriate approximations of the safety zone the probability of crossing the boundary becomes computationally tractable. The essence of the proposed method is to consider the crossing of a disc instead of the crossing of the sphere. The results for two dimensions given by [17] are extended to three dimensions. The main difference between two and three dimensions is that we need to compute a probability over a circular disc instead of a line. In contrast to the two-dimensional case, there does not exist any method to analytically compute the probability over the circular disc. An alternative approach is then needed, and we use the fact that the characteristic function exists as an analytical expression, and then compute the probability by numerically inverting the characteristic function. The result is extended to cover the important case with piecewise linear trajectories, which provides a way forward to deal with curved paths in general and avoidance maneuvers in particular [19]. We consider time horizons up to a couple of minutes, and therefore neglect effects from disturbances on the predicted path. When longer periods of time are considered the effects of for example wind disturbances can be significant [7].

In Section 2 we formulate the problem mathematically and state the prerequisites to be able to provide a solution. Section 3 details the exact conditions for a NMAC to occur. The conditions are based on the minimum relative distance for a given predicted trajectory. If the minimum relative distance is less than a predefined threshold that particular trajectory will lead to a NMAC. We start by giving the solution for a straight path in Section 3.1 and then continue to the case with piecewise straight paths in Section 3.2. Section 4 provides an approximate solution to compute the probability of NMAC. This solution is based on sampling methods and yields an arbitrarily small approximation error but is

computationally demanding. In Section 5 we present a novel solution based on an approximation of the NMAC geometry. A NMAC will occur if the predicted relative position ever crosses the surface of a predefined sphere. If we instead of a sphere consider the crossing of a circular disc with certain properties the problem becomes computationally tractable. The probability of crossing the circle can be computed using the distribution of the distance perpendicular to line-of-sight weighted with the probability density for time-to-go (ttg) and then integrated over a time interval. Details on the implementation of the disc approximation is given in Section 6. In Section 7 we give some results comparing the sampling solution with the geometric solution and finally in Section 8 we draw conclusions.

2 Problem formulation

The probability of near-midair collision (NMAC) between two aerial vehicles for a given time period $(0, T)$ is defined as

$$\mathrm{P}\big(\mathrm{NMAC}_{(0,T)}\big) = \mathrm{P}\big(\min_{0<t<T}|s(t)|<R\big), \tag{1}$$

where $s(t)$ represents the relative position between the two vehicles at time $t \geq 0$ and t is the prediction time. R is the radius of a safety zone, which we assume has the shape of a sphere, and $R = 150$ m, see Figure 1.

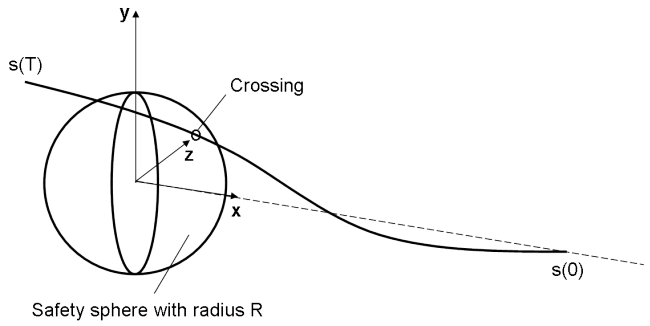

Figure 1: *A NMAC occurs if the relative position crosses the safety sphere.*

The definition according to (1) means that if the distance

$$|s(t)| = \sqrt{s_x^2(t) + s_y^2(t) + s_z^2(t)}$$

for any $0 < t < T$ falls below R, no matter for how long, we have a NMAC.

Typically, an estimate of relative position is provided by an angle-only tracking filter [5]. Target tracking will not be pursued here in detail, we simply state that based on measurements from an angle measurement unit e.g. an electro-optical sensor, the tracking filter estimates three-dimensional relative position $s(0)$ and velocity $v(0)$ in cartesian coordinates together with their covariances. To simplify the problem formulation we assume the angle measurement unit is accurate and the coordinate system is rotated such

that the $x-$ axis is aligned with line of sight. This means that

$$s_y(0) \equiv s_z(0) \equiv 0 \tag{2}$$

and the estimated state vector used for the probability computations is

$$\hat{x}(0) = \begin{bmatrix} \hat{s}_x(0) & \hat{v}_x(0) & \hat{v}_y(0) & \hat{v}_z(0) \end{bmatrix}^T. \tag{3}$$

The corresponding estimated covariance matrix is, using $\mathrm{var}(s_y) = \mathrm{var}(s_z) = 0$ from (2),

$$\begin{aligned} P(0) &= \begin{bmatrix} P_{sx} & C \\ C^T & P_{yz} \end{bmatrix}, \\ P_{sx} &= \begin{bmatrix} \sigma_{sx}^2 & \rho_x \sigma_{sx} \sigma_{vx} \\ \rho_x \sigma_{sx} \sigma_{vx} & \sigma_{vx}^2 \end{bmatrix}, \end{aligned} \tag{4}$$

and similar for P_{yz}. We assume the tracking filter output is normally distributed, i.e.

$$\hat{x}(0) \sim \mathcal{N}(x(0), P(0)). \tag{5}$$

Note that we will only deal with a relative time scale, represented by $t = 0$ as the current time on an absolute time scale. At each new time instant on the absolute time scale the tracking filter provides updated estimates of $x(0)$ and $P(0)$.

We seek a method capable of accurately computing probability of NMAC when the underlying probability of NMAC is 0.01 or larger. The figure 0.01 comes from the performance of TCAS, which based on simulation studies has a failure rate of around 0.1 [3, 15]. Taking into account that the sensor and tracking filter have a limited intruder detection capability makes 0.01 reasonable. We must be able to detect a collision scenario with better accuracy than 0.1, due to e.g. sensor and tracking limitations, to achieve an overall system accuracy of 0.1. The computation accuracy should be 10% or better, i.e. if the probability is 0.01 then the method should provide a result which does not deviate more than 10% from 0.01. The method must be computationally tractable for real-time processing.

3 Crossing of the safety zone

3.1 Constant velocity

Let us first consider the event $\mathrm{NMAC}_{(0,T)}$ assuming a constant velocity

$$\dot{s}(t) = v(0), \quad 0 < t < T, \tag{6a}$$

$$v(t) = v(0), \tag{6b}$$

The definition of $\mathrm{NMAC}_{(0,T)}$ according to (1) can also be written as

$$\begin{aligned} &\mathrm{P}\big(\mathrm{NMAC}_{(0,T)}\big) = \\ &\mathrm{P}\big(\min_{0<t<T} |s(t)| < R \cap |s(0)| > R\big) + \mathrm{P}\big(|s(0)| < R\big). \end{aligned} \tag{7}$$

The definition means that we use two mutually exclusive events for $|s(0)|$ to split the probability, and at the same time noting that if $|s(0)| < R$ we automatically have a distance which is less than R. The first probability term in (7) corresponds to a down-crossing of the surface of the sphere. The situation of a down-crossing is depicted in Figure 2. The minimum relative distance $\min_{t>0} |s(t)|$ is attained when $s(t)$ is orthogonal to v.

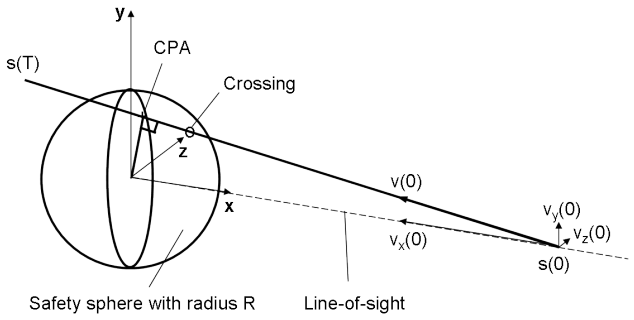

Figure 2: *Crossing of the safety sphere for a straight path.*

This point is called *closest point of approach* and denoted by cpa. The time until cpa is reached, t_{cpa}, can be computed using the equation

$$v^T\big(s(0) + vt_{\mathrm{cpa}}\big) = 0, \tag{8}$$

which yields

$$t_{\mathrm{cpa}} = -\frac{v^T s(0)}{|v|^2}. \tag{9}$$

The main condition for $\mathrm{NMAC}_{(0,T)}$ then becomes

$$\min_{0<t<T} |s(t)| = |s(t_{\mathrm{cpa}})| = |s(0) + vt_{\mathrm{cpa}}| < R, \tag{10}$$

which we denote

$$C_1 = |s(0) + vt_{\mathrm{cpa}}| < R. \tag{11}$$

A finite end time $t < T$ means that t_{cpa} as computed by (9) could yield $t_{\mathrm{cpa}} > T$. As long as $t_{\mathrm{cpa}} < T$, condition C_1 applies, but in case $t_{\mathrm{cpa}} > T$ we can still have a NMAC situation if

$$C_4 = |s(0) + vT| < R. \tag{12}$$

The two remaining conditions needed for a complete description of a down-crossing are $t_{\mathrm{cpa}} > 0$ and $|s(0)| > R$. The first one corresponds to the two vehicles approaching each other, and the second one comes from the definition of a down-crossing. To summarize, the conditions for $\mathrm{NMAC}_{(0,T)}$ are

$$\mathrm{NMAC}_{(0,T)} = (C_1 \cap C_2 \cup C_4) \cap C_3 \cup \bar{C}_3, \tag{13}$$

where $\cap$ denotes intersection, $\cup$ union, $\bar{C}$ complement of C and

$$\begin{aligned} C_1 &= |s(0) + vt_{\text{cpa}}| < R, \\ C_2 &= 0 < t_{\text{cpa}} < T, \\ C_3 &= |s(0)| > R, \\ C_4 &= |s(0) + vT| < R. \end{aligned} \tag{14}$$

3.2 Piecewise constant velocity

From now on we assume the relative position follows a piecewise straight path given by

$$\dot{s}(t) = v^{(j)}, \quad T_j < t < T_{j+1}, \tag{15a}$$

$$v^{(j)} = v(0) + \sum_{l=1}^{j} \Delta v^{(l)}, \tag{15b}$$

where all $\Delta v^{(l)}$ are known. Note that the model in (15) is fairly general because we can approximate any continuous curved path arbitrarily well as long as we use a large enough number of straight segments.

The event $\text{NMAC}_{(0,T)}$ can always be expressed according to

$$\text{NMAC}_{(0,T)} = \cup_{j=0}^{J-1} \text{NMAC}_{(T_j,T_{j+1})}, \tag{16}$$

i.e. the union of events for each segment $T_j < t < T_{j+1}$ with $T_0 = 0$ and $T_J = T$. A geometric interpretation of $\text{NMAC}_{(0,T)}$ for two segments is given by Figure 3.

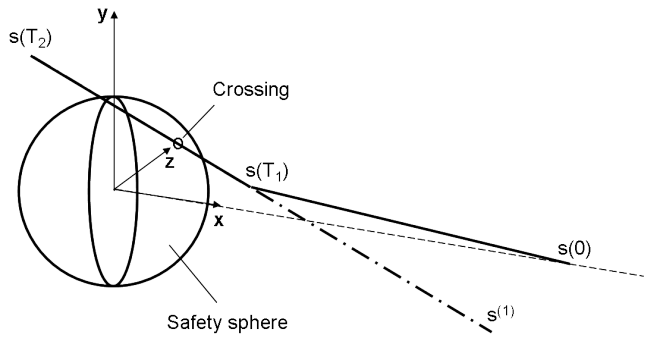

Figure 3: *Crossing of the safety sphere for a piecewise straight path with two segments.*

The change to be incorporated for $\text{NMAC}_{(T_j,T_{j+1})}$ compared to the conditions for the constant velocity case is caused by considering $t = T_j$ instead of $t = 0$ as the starting time. This yields changing variables to $s(T_j)$ and $v^{(j)}$. It is possible to derive conditions for $\text{NMAC}_{(T_j,T_{j+1})}$ using $s(T_j)$. However, a more efficient way is to consider the distance obtained by extrapolating from $s(T_j)$ backwards in time T_j seconds using the current velocity $v^{(j)}$. This yields the distance at $t = 0$ which would give $s(T_j)$ after T_j seconds

using a constant velocity $v^{(j)}$, compare with Figure 3. This means that we are back to starting time $t = 0$ and it enables us to reuse results from the case with constant velocity. Denote the distance obtained from extrapolation by $s^{(j)}$, and we have

$$s^{(j)} = s(T_j) - T_j v^{(j)}. \tag{17}$$

An advantage using $s^{(j)}$ instead of $s(T_j)$ comes from the fact that the covariance of $s^{(j)}$ is equal to the covariance of $s(0)$ for all $j = 0, \ldots, J-1$. This is clear from the expression [17]

$$s^{(j)} = s(0) - \sum_{l=1}^{j} T_l \Delta v^{(l)}. \tag{18}$$

Inserting the new variables into the conditions from (14) yield

$$\begin{aligned} C_1^{(j)} &= |s^{(j)} + v^{(j)} t_{\text{cpa}}^{(j)}| < R, \\ C_2^{(j)} &= T_j < t_{\text{cpa}}^{(j)} < T_{j+1}, \\ C_3^{(j)} &= |s^{(j)} + T_j v^{(j)}| > R, \\ C_4^{(j)} &= |s^{(j)} + v^{(j)} T_{j+1}| < R, \end{aligned} \tag{19}$$

where

$$t_{\text{cpa}}^{(j)} = -\frac{(v^{(j)})^T s^{(j)}}{|v^{(j)}|^2}. \tag{20}$$

Recall that we are dealing with stochastic variables, which means that we compute

$$\begin{aligned} &\mathrm{P}\big(\text{NMAC}_{(0,T)}\big) = \\ &\mathrm{P}\big(\cup_{j=0}^{J-1} (C_1^{(j)} \cap C_2^{(j)} \cup C_4^{(j)}) \cap C_3^{(j)} \cup \bar{C}_3^{(j)}\big). \end{aligned} \tag{21}$$

4 Monte-Carlo Approximation

The probability according to (21) is in general very difficult to compute. A straightforward approximate solution is to use a Monte-Carlo method, i.e. to draw N samples of $x(0)$ from (5) and approximate the probability with the outcome of the sampling, i.e.

$$\begin{aligned} &\hat{\mathrm{P}}_{\text{mc}}\big(\text{NMAC}_{(0,T)}\big) = \\ &\frac{1}{N} \sum_{i=1}^{N} I\Big(\cup_{j=0}^{J-1} (C_1^{(i,j)} \cap C_2^{(i,j)} \cup C_4^{(i,j)}) \cap C_3^{(i,j)} \cup \bar{C}_3^{(i,j)} \Big), \end{aligned} \tag{22}$$

where $I(\cdot)$ is the indicator function. Denote the true value of the sought probability with p. The set of samples is binomially distributed, $\text{Bin}(N, p)$, but for a large enough N, usually $Np(1-p) > 20$ is sufficient, the probability is approximated well by [10]

$$\begin{aligned} &\hat{\mathrm{P}}_{\text{mc}}\big(\text{NMAC}_{(0,T)}\big) \sim \mathcal{N}(p, \sigma^2), \\ &\sigma^2 = \frac{p(1-p)}{N}. \end{aligned} \tag{23}$$

For a relative mean square error $\epsilon \leq \frac{\sigma}{p}$ we can write needed number of samples according to

$$N \geq \frac{1-p}{\epsilon^2 p} \approx \frac{1}{\epsilon^2 p}, \tag{24}$$

where the last approximation is valid for small p. Assume $p = 0.01$ and $3\epsilon \leq 0.1$, i.e a relative error smaller than 10% with probability 0.997. These values plugged into (24) suggests that we must use $N \geq 90000$. For many on-line applications this means a too high computational load.

5 Approximate the safety zone with a disc

5.1 Constant velocity

Let us first consider the constant velocity case according to (6). An approximate near mid-air collision, denoted by $\widehat{\mathrm{NMAC}}_{(0,T)}$, is given by considering the crossing of a circular disc instead of a sphere. The disc can be seen as a cross-section of the sphere perpendicular to line-of-sight as illustrated in Figure 4. Note that the location and radius of the disc is a matter of choice. For example placing the disc at $x = R$, i.e. in front of the safety sphere, and with radius R yields a conservative result for probability of $\mathrm{NMAC}_{(0,T)}$. Below we place the disc at $x = 0$ primarily for notational convenience.

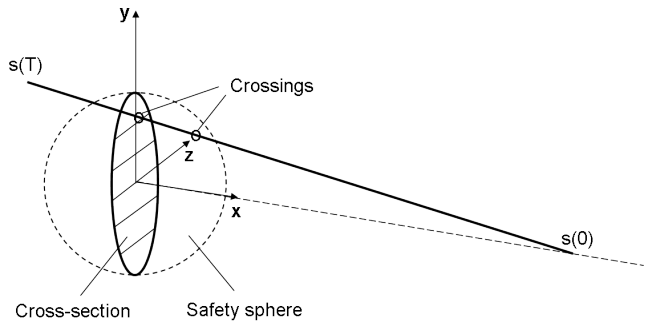

Figure 4: *Approximating the crossing of the safety sphere with a cross-section.*

Following the same principle as in [17] we define a stochastic time variable τ, representing time-to-go (ttg), according to

$$\tau = \begin{cases} \frac{s_x}{-v_x} & \text{if } s_x > 0 \ \cap \ v_x < 0, \\ \infty & \text{otherwise,} \end{cases} \tag{25}$$

Time-to-go is the ratio of distance and closing speed, and corresponds to time left before the relative position crosses the $yz-$plane. The probability density function for τ is given by Lemma 1.

Lemma 1 (Probability density function for τ)
For s_x and v_x distributed according to

$$\begin{bmatrix} s_x \\ v_x \end{bmatrix} \sim \mathcal{N}\Big(\begin{bmatrix} \hat{s}_x \\ \hat{v}_x \end{bmatrix}, \begin{bmatrix} \sigma_{sx}^2 & \rho_x \sigma_{sx}\sigma_{vx} \\ \rho_x \sigma_{sx}\sigma_{vx} & \sigma_{vx}^2 \end{bmatrix} \Big), \tag{26}$$

the probability density function for τ as defined by (25) *is given by*

$$\begin{aligned} p_\tau(t) &= \int_{-\infty}^{0} -v p_{s_x,v_x}(-vt, v)dv \\ &= \frac{g_0}{g_2^2}\Big(1 - (2\pi)^{\frac{1}{2}} \frac{g_1}{g_2} e^{\frac{g_1^2}{2g_2^2}} \Phi\big(-\frac{g_1}{g_2}\big)\Big), \end{aligned} \tag{27}$$

where $\kappa = \rho_x \sigma_{sx}/\sigma_{vx}$ and

$$\begin{aligned} g_0 &= \frac{1}{2\pi\sigma_{sx}\sigma_{vx}(1-\rho_x^2)^{\frac{1}{2}}} e^{-\frac{(\kappa\hat{v}_x - \hat{s}_x)^2}{2\sigma_{sx}^2(1-\rho_x^2)} - \frac{\hat{v}_x^2}{2\sigma_{vx}^2}}, \\ g_1 &= \frac{(t+\kappa)(\kappa\hat{v}_x - \hat{s}_x)}{\sigma_{sx}^2(1-\rho_x^2)} + \frac{\hat{v}_x}{\sigma_{vx}^2}, \\ g_2 &= \left(\frac{(t+\kappa)^2}{\sigma_{sx}^2(1-\rho_x^2)} + \frac{1}{\sigma_{vx}^2}\right)^{\frac{1}{2}}, \end{aligned} \tag{28}$$

and $\Phi(\cdot)$ corresponds to the standard normal distribution

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{\xi^2}{2}} d\xi. \tag{29}$$

Proof: *See [11].*

We also define a distance from line-of-sight, denoted by $s_\perp(t)$. This distance is a function of time-to-go $\tau = t$, according to

$$s_\perp(t) = \sqrt{(tv_y)^2 + (tv_z)^2} = t \left| \begin{bmatrix} v_y \\ v_z \end{bmatrix} \right|, \tag{30}$$

and corresponds to the total displacement perpendicular to line-of-sight after $\tau = t$ seconds. The interpretation of the event $\widehat{\mathrm{NMAC}}_{(0,T)}$ in terms of τ and $s_\perp(t)$ is that, given a time-to-go $0 < \tau = t < T$, if $s_\perp(t) < R$ the event will occur. Since both τ and $s_\perp(t)$ are stochastic we need to compute the probability of $\widehat{\mathrm{NMAC}}_{(0,T)}$. The probability for a given ttg is provided by

$$\mathrm{P}(s_\perp(\tau) < R | \tau = t) = \mathrm{P}(s_\perp(t) < R). \tag{31}$$

By weighting with the probability density for τ we have

$$\mathrm{P}\big(\widehat{\mathrm{NMAC}}_{(0,T)}\big) = \int_0^T \mathrm{P}\big(s_\perp(t) < R\big) p_\tau(t) dt. \tag{32}$$

The above holds when s_x, v_x are independent of $s_\perp(t)$. In the general case a dependency makes it difficult to express the probability of NMAC in terms of $p_\tau(t)$. This is seen from the block diagonalization of the covariance matrix P,

$$P = \begin{bmatrix} I & 0 \\ C^T P_{sx}^{-1} & I \end{bmatrix} \begin{bmatrix} P_{sx} & 0 \\ 0 & P_{vz} - C^T P_{sx}^{-1} C \end{bmatrix} \begin{bmatrix} I & 0 \\ C^T P_{sx}^{-1} & I \end{bmatrix}^T, \tag{33}$$

which results in

$$\begin{aligned} s_\perp(t) &= \left| \begin{bmatrix} tv_y \\ tv_z \end{bmatrix} - C^T P_{sx}^{-1} \begin{bmatrix} ts_x \\ tv_x \end{bmatrix} \right| \\ &= \left| \begin{bmatrix} tv_y \\ tv_z \end{bmatrix} - C^T P_{sx}^{-1} \begin{bmatrix} t^2 v_x \\ tv_x \end{bmatrix} \right|, \end{aligned} \tag{34}$$

where the last step corresponds to a change of variable from s_x to $t = -s_x/v_x$. As can be seen in (34) the dependency on v_x remains.

However, we can split $p_\tau(t)$ into M partitions, where each partition $m = 1, \ldots, M$, corresponds to a subset of closing speeds $a_m < v_x < b_m$. Now, for each partition, we compute the conditional mean of v_x given $\tau = t$ and $a_m < v_x < b_m$. The partitioned pdf, $p_\tau^{(m)}(t)$, and the conditional mean, $\hat{\nu}_x^{(m)}(t)$, are given by Corollary 1.

Corollary 1 (Expressions for $p_\tau^{(m)}(t)$ and $\hat{\nu}_x^{(m)}(t)$)
With the same assumptions as in Lemma 1 we have

$$\begin{aligned} p_\tau^{(m)}(t) &= \int_{a_m}^{b_m} -v p_{s_x, v_x}(-vt, v) dv \\ &= \frac{g_0}{g_2^2} e^{\frac{g_1^2}{2g_2^2}} \Bigg(e^{-\frac{1}{2}(g_2 b_m - \frac{g_1}{g_2})^2} - e^{-\frac{1}{2}(g_2 a_m - \frac{g_1}{g_2})^2} \\ &\quad - \sqrt{2\pi} \frac{g_1}{g_2} \big(\Phi\big(g_2 b_m - \frac{g_1}{g_2}\big) - \Phi\big(g_2 a_m - \frac{g_1}{g_2}\big) \big) \Bigg), \end{aligned} \tag{35}$$

and the conditional expected value of v_x given $\tau = t$ and $a_m < v_x < b_m$ is

$$\begin{aligned} \hat{\nu}_x^{(m)}(t) &= \frac{\int_{a_m}^{b_m} -v^2 p_{s_x, v_x}(-vt, v) dv}{\int_{a_m}^{b_m} -v p_{s_x, v_x}(-vt, v) dv} = \frac{g_0}{g_2^3 p_\tau^{(m)}(t)} e^{\frac{g_1^2}{2g_2^2}} \\ &\cdot \Bigg((g_2 b_m + \frac{g_1}{g_2}) e^{-\frac{1}{2}(g_2 b_m - \frac{g_1}{g_2})^2} - (g_2 a_m + \frac{g_1}{g_2}) e^{-\frac{1}{2}(g_2 a_m - \frac{g_1}{g_2})^2} \\ &\quad - \sqrt{2\pi} \big(1 + \frac{g_1^2}{g_2^2}\big) \big(\Phi\big(g_2 b_m - \frac{g_1}{g_2}\big) - \Phi\big(g_2 a_m - \frac{g_1}{g_2}\big) \big) \Bigg). \end{aligned} \tag{36}$$

Proof: *Straightforward calculations using the same technique as for $p_\tau(t)$.*

The probability $\mathrm{P}(\widehat{\mathrm{NMAC}}_{(0,T)})$ is now given by applying Taylor expansion around $\hat{\nu}_x^{(m)}(t)$ for each partition, see Theorem 1.

Theorem 1 ($\widehat{\mathrm{NMAC}}_{(0,T)}$ for a straight path)
For $s(t) = \begin{bmatrix} s_x(t) & s_y(t) & s_z(t) \end{bmatrix}^T$ with assumptions (2)*,* (5) *and* (6) *the probability of a down-crossing within T sec of a circular disc with $x-$axis as its normal and radius R is given by*

$$\begin{aligned} &\mathrm{P}(\widehat{\mathrm{NMAC}}_{(0,T)}) = \\ &\int_0^T \sum_{m=1}^{M} \mathrm{P}\big(s_\perp^{(m)}(t) < R\big) p_\tau^{(m)}(t) dt + \mathrm{P}_M \end{aligned} \tag{37}$$

where

$$s_\perp^{(m)}(t) = \left| \begin{bmatrix} tv_y \\ tv_z \end{bmatrix} - C^T P_{sx}^{-1} \begin{bmatrix} t^2 \hat{\nu}_x^{(m)}(t) \\ t\hat{\nu}_x^{(m)}(t) \end{bmatrix} \right|. \tag{38}$$

The rest term is upper bounded by

$$\begin{aligned} \mathrm{P}_M \leq \int_0^T \sum_{m=1}^{M} \int_{a_m}^{b_m} &\left| \frac{\partial^2 \mathrm{P}\big(s_\perp(t) < R\big)}{\partial v_x^2} \right|_{v_x = \mu^{(m)}(t)} \\ \cdot &- v(v - \mu^{(m)}(t))^2 p_{s_x, v_x}(-vt, v) dv dt \end{aligned} \tag{39}$$

where $a_m < \mu^{(m)}(t) < b_m$, $a_1 = -\infty$ and $b_M = 0$.
Proof: *See Appendix A.1.*

Remark 1. We can always apply Taylor expansion around a more accessible point compared to $\hat{\nu}_x^{(m)}$. For example, in the case we use $M = 1$ a reasonable choice is $\hat{v}_x$, the unconditional expected value of v_x. The advantage using $\hat{v}_x$ instead of $\hat{\nu}_x^{(m)}$ is that the computational load is decreased. On the other hand, using $\hat{v}_x$ in general makes the rest term P_M larger because the first order term in the Taylor expansion is no longer eliminated.

5.2 Piecewise constant velocity

The extension of the result for a straight path to the case with a piecewise straight path given by (15) is as follows. An important observation is that if it is unlikely that $v_x^{(j)}$ changes sign, i.e.

$$\mathrm{P}\big(\mathrm{sign}(v_x^{(j)}) = \mathrm{sign}(v_x^{(0)})\big) \approx 1, \tag{40}$$

then $\widehat{\mathrm{NMAC}}_{(T_j, T_{j+1})}$ for different j's are mutually exclusive. With this assumption we can write

$$\mathrm{P}(\mathrm{NMAC}_{(0,T)}) = \sum_{j=0}^{J-1} \mathrm{P}(\mathrm{NMAC}_{(T_j, T_{j+1})}), \tag{41}$$

and we can concentrate on each segment $T_j < t < T_{j+1}$ with constant velocity $v^{(j)}$. The principle for computing $\mathrm{P}\big(\widehat{\mathrm{NMAC}}_{(T_j,T_{j+1})}\big)$ is the same as for $\mathrm{P}\big(\widehat{\mathrm{NMAC}}_{(0,T)}\big)$. The difference is that we use the initial values $v^{(j)}$ and $s^{(j)}$ from (15b) and (18) respectively instead of $v(0)$ and $s(0)$. Define a stochastic time variable $\tau^{(j)}$ according to

$$\tau^{(j)} = \begin{cases} \frac{s_x^{(j)}}{-v_x^{(j)}} & \text{if } s_x^{(j)} > 0 \cap v_x^{(j)} < 0, \\ \infty & \text{otherwise,} \end{cases} \tag{42}$$

We also define a distance from line-of-sight, denoted by $s_\perp^{(j)}(t)$, according to

$$s_\perp^{(j)}(t) = \sqrt{(tv_y^{(j)} + s_y^{(j)})^2 + (tv_z^{(j)} + s_z^{(j)})^2}, \tag{43}$$

and corresponds to the total displacement perpendicular to line-of-sight after $\tau^{(j)} = t$ seconds starting from $s^{(j)}$. Note that $s_y^{(j)}$ and $s_z^{(j)}$ are known and deterministic based on (2) and (18). The probability of NMAC for a piecewise straight path is given by Corollary 2.

Corollary 2 ($\widehat{\mathrm{NMAC}}_{(0,T)}$ for a piecewise straight path)
For $s^{(j)}(t) = \begin{bmatrix} s_x^{(j)}(t) & s_y^{(j)}(t) & s_z^{(j)}(t) \end{bmatrix}^T$ with assumptions (2)*,* (5) *and* (15) *the probability of a down-crossing within T sec of a circular disc with $x-$axis as its normal and radius R is given by*

$$\begin{aligned} &\mathrm{P}(\widehat{\mathrm{NMAC}}_{(0,T)}) = \\ &\sum_{j=0}^{J-1} \int_{T_j}^{T_{j+1}} \sum_{m=1}^{M} \mathrm{P}\big(s_\perp^{(j,m)}(t) < R\big) p_{\tau^{(j)}}^{(m)}(t) dt + \mathrm{P}_M^{(j)} \end{aligned} \tag{44}$$

where

$$s_\perp^{(j,m)}(t) = \left| \begin{bmatrix} tv_y^{(j)} + s_y^{(j)} \\ tv_z^{(j)} + s_z^{(j)} \end{bmatrix} - C^T P_{sx}^{-1} \begin{bmatrix} t^2 \hat{\nu}_x^{(j,m)}(t) \\ t\hat{\nu}_x^{(j,m)}(t) \end{bmatrix} \right|. \tag{45}$$

The rest term is upper bounded by

$$\begin{aligned} \mathrm{P}_M^{(j)} \leq & \int_0^T \sum_{m=1}^{M} \int_{a_m}^{b_m} \left| \frac{\partial^2 \mathrm{P}\big(s_\perp(t) < R\big)}{\partial v_x^2} \right|_{v_x = \mu^{(m)}(t))} \\ & \cdot - v(v - \mu^{(m)}(t))^2 p_{s_x, v_x}(-vt, v) dv dt \end{aligned} \tag{46}$$

where $a_m < \mu^{(j,m)}(t) < b_m$, $a_1 = -\infty$ and $b_M = 0$.
Proof: *See Appendix A.1 with $s^{(j)}, v^{(j)}$ instead of $s(0), v(0)$.*

6 Implementation of the disc approximation

6.1 Computing $\mathrm{P}\big(s_{\perp}^{(j,m)}(t) < R\big)$

For notational convenience we ignore index j and study the probability $\mathrm{P}\big(s_{\perp}(t) < R\big)$, keeping in mind that $s_{\perp}(t)$ is actually $s_{\perp}^{(j,m)}(t)$ as given in (45). Let us define a new random variable representing orthogonal displacement per time unit according to

$$v_{\perp}(t) = \frac{s_{\perp}(t)}{t} = \left| \begin{bmatrix} \nu_y(t) \\ \nu_z(t) \end{bmatrix} \right|, \tag{47}$$

where we have using (45)

$$\begin{bmatrix} \nu_y(t) \\ \nu_z(t) \end{bmatrix} = \begin{bmatrix} v_y + \frac{\hat{s}_y}{t} \\ v_z + \frac{\hat{s}_z}{t} \end{bmatrix} - C^T P_{sx}^{-1} \begin{bmatrix} t\hat{\nu}_x(t) \\ \hat{\nu}_x(t) \end{bmatrix}. \tag{48}$$

The covariance matrix for ν_y and ν_z is from (33) given by $P_{yz} - C^T P_{sv}^{-1} C$. We assume that ν_y and ν_z are uncorrelated, i.e. the matrix $P_{yz} - C^T P_{sv}^{-1} C$ is diagonal. This is no restriction since correlation is handled by applying a change of variables

$$\begin{bmatrix} \nu_y' \\ \nu_z' \end{bmatrix} = U \begin{bmatrix} \nu_y \\ \nu_z \end{bmatrix}, \tag{49}$$

where U is a unitary matrix given by

$$P_{yz} - C^T P_{sv}^{-1} C = U D U^T = U \begin{bmatrix} d_y^2 & 0 \\ 0 & d_z^2 \end{bmatrix} U^T. \tag{50}$$

The sought probability can now be written according to

$$\begin{aligned} &\mathrm{P}(v_{\perp} < \frac{R}{t}) = \mathrm{P}\big(v_{\perp}^2 < \frac{R^2}{t^2}\big) = \\ &\mathrm{P}\big(d_y^2 \frac{\nu_y^2(t)}{d_y^2} + d_z^2 \frac{\nu_z^2(t)}{d_z^2} < \frac{R^2}{t^2}\big). \end{aligned} \tag{51}$$

Under the assumption that ν_y and ν_z are normally distributed and uncorrelated, we know that $\nu_y^2(t)/d_y^2$ and $\nu_z^2(t)/d_z^2$ are two independent non-central χ^2-distributed variables with one degree of freedom,

$$\frac{\nu_y^2(t)}{d_y^2} \sim \chi_1^2(\lambda_y), \quad \frac{\nu_z^2(t)}{d_z^2} \sim \chi_1^2(\lambda_z), \tag{52}$$

where the non-centrality parameters λ_y and λ_z are given by

$$\lambda_y = \frac{\hat{\nu}_y^2(t)}{d_y^2}, \quad \lambda_z = \frac{\hat{\nu}_z^2(t)}{d_z^2}. \tag{53}$$

The probability $\mathrm{P}(v_{\perp}^2 < \frac{R^2}{t^2})$ in (51) is not available in closed form. A simple way to compute (51) is to approximate it with a central χ^2 [12, 18]. The problem with this

method is that it is difficult to estimate the approximation error. A better approximation to (51), in the sense that it is possible to control the approximation error, is to use the fact that the characteristic function of the distribution in (51) is available in closed form [22], i.e.

$$\phi_{v_\perp^2}(\xi) = \mathrm{E}\big[e^{i\xi v_\perp^2}\big] = \frac{e^{\frac{i\sigma_y^2\lambda_y\xi}{1-2i\sigma_y^2\xi}}}{\sqrt{1-2i\sigma_y^2\xi}}\frac{e^{\frac{i\sigma_z^2\lambda_z\xi}{1-2i\sigma_z^2\xi}}}{\sqrt{1-2i\sigma_z^2\xi}}. \tag{54}$$

Then use the inverse to the characteristic function to compute the distribution according to

$$\mathrm{P}(v_\perp^2 < \frac{R^2}{t^2}) = \frac{2}{\pi}\int_0^\infty \mathrm{Re}\Big(\phi_{v_\perp^2}(\xi)\Big)\frac{\sin\xi\frac{R^2}{t^2}}{\xi}d\xi. \tag{55}$$

The probability in (55) is computed using [1]

$$\begin{aligned}&\mathrm{P}(v_\perp^2 < \frac{R^2}{t^2}) = \\ &\frac{\Delta\frac{R^2}{t^2}}{2\pi} + \frac{2}{\pi}\sum_{l=1}^{L}\mathrm{Re}\Big(\phi_{v_\perp^2}(\Delta l)\Big)\frac{\sin\Delta l\frac{R^2}{t^2}}{l} + I_D + I_T.\end{aligned} \tag{56}$$

I_D is the discretization error and is controlled by Δ, and I_T is the truncation error and is controlled by ΔL. See Algorithm 1 for details on the realization.

Algorithm 1 ($\mathrm{P}(v_\perp^2 < \frac{R^2}{t^2})$ *as a characteristic function).*

1. Compute Δ:

$$\begin{aligned}\widehat{v_\perp^2} &= \sigma_y^2(1+\lambda_y) + \sigma_z^2(1+\lambda_z),\\ \sigma_{v_\perp^2} &= \big(2\sigma_y^4(1+2\lambda_y) + 2\sigma_z^4(1+2\lambda_z)\big)^{\frac{1}{2}},\\ \Delta &= \frac{\pi}{\widehat{v_\perp^2} + 6\sigma_{v_\perp^2}}\end{aligned}$$

2. Choose L such that:

$$\mathrm{Re}\Big(\phi_{v_\perp^2}(\Delta L)\Big)\frac{2}{\pi L} < 1\cdot 10^{-5}$$

3. Compute $\hat{\mathrm{P}}_L(v_\perp^2 < \frac{R^2}{t^2})$:

$$\begin{aligned}&\hat{\mathrm{P}}_L(v_\perp^2 < \frac{R^2}{t^2})\\ &= \frac{\Delta\frac{R^2}{t^2}}{2\pi} + \frac{2}{\pi}\sum_{l=1}^{L}\mathrm{Re}\Big(\phi_{v_\perp^2}(\Delta l)\Big)\frac{\sin\Delta l\frac{R^2}{t^2}}{l}\end{aligned}$$

Example 1: $\mathrm{P}(v_\perp^2 < \frac{R^2}{t^2})$

Consider the case with $C = 0$, $\hat{s}_y = \hat{s}_z = 0$ and

$$\begin{bmatrix} \hat{v}_y \\ \hat{v}_z \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \end{bmatrix}, \quad P_{yz} = \begin{bmatrix} 2^2 & 0 \\ 0 & 1^2 \end{bmatrix}. \tag{57}$$

This is a typical result of angle-only tracking where the velocity orthogonal to line-of-sight is estimated accurately. The result of computing $\mathrm{P}(v_\perp^2 < \frac{R^2}{t^2})$ with $R = 150$ using Algorithm 1 is given by Figure 5. We see that using no more than $15 - 20$ terms $(= L)$ gives equivalent accuracy compared to using the Monte Carlo solution,

$$\begin{aligned} &\hat{\mathrm{P}}_{\mathrm{mc}}(v_\perp^2 < \frac{R^2}{t^2}) \\ &= \frac{1}{M} \sum_{i=1}^{M} I\left((v_y^{(i)})^2 + (v_z^{(i)})^2 < \frac{R^2}{t^2} \right), \end{aligned} \tag{58}$$

with $M = 1000000$.

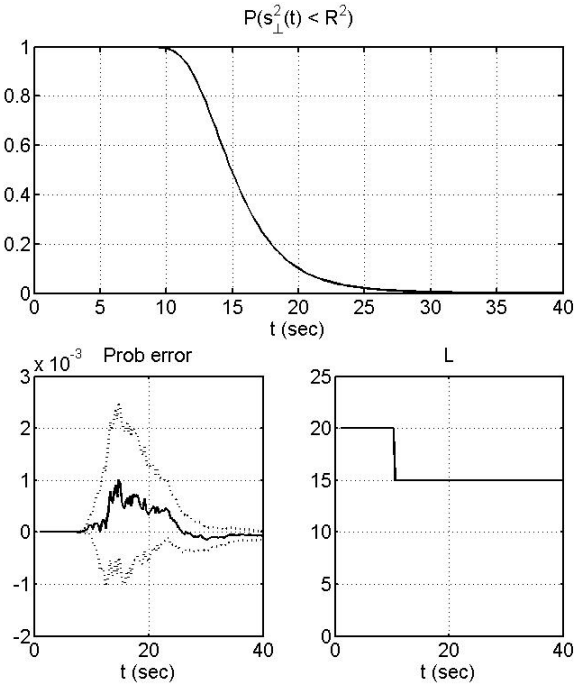

Figure 5: *The upper plot shows estimated* $\mathrm{P}(v_\perp^2 < \frac{R^2}{t^2})$*, the lower left shows the difference compared to Monte Carlo solution (solid line) including the* $3 - \sigma$ *levels for the Monte Carlo solution (dotted lines), and the lower right shows the number of terms used in Algorithm 1.*

6.2 Computing $\mathrm{P}(\widehat{\mathrm{NMAC}}_{(0,T)})$

A sound and simple numerical approximation for computing a one-dimensional integral is given by Simpson's rule [2]

$$\begin{aligned}\mathrm{P}(\mathrm{NMAC}_{(T_j,T_{j+1})}) &= \int_{T_j}^{T_{j+1}} f(t)dt \\ &= \frac{\Delta t}{3}\Big(f(t_0) + 2\sum_{k=1}^{K/2-1} f(t_{2k}) + 4\sum_{k=1}^{K/2} f(t_{2k-1}) + f(t_K)\Big) + R_K,\end{aligned} \tag{59}$$

where $\Delta t = (T_{j+1} - T_j)/K$, $t_k = k\Delta t + T_j$ and

$$f(t) = \sum_{m=1}^{M} \mathrm{P}\big(s_\perp^{(j,m)}(t) < R\big) p_{\tau^{(j,m)}}(t). \tag{60}$$

From [2] we know that the approximation error is bounded by

$$R_K < \frac{\Delta t^5}{90} \sum_{k=0}^{K/2-1} \max_{t_{2k}<t<t_{2k+2}} \Big|\frac{\partial^4 f}{\partial t^4}\Big|. \tag{61}$$

The implementation of Corollary 2 is given by Algorithm 2.

Algorithm 2 (Implementation of Corollary 2).

1. Set $j = 0$.
2. For each $t_k = k\Delta t_j + T_j$ and $m = 1, \ldots, M_j$ compute
$$p_{\tau^{(j,m)}}(t_k), \quad \hat{\nu}_x^{(j,m)}(t_k), \tag{62}$$
given by Corollary 1.
3. For each $t_k = k\Delta t_j + T_j$ and $m = 1, \ldots, M_j$ compute
$$\mathrm{P}\big(s_\perp^{(j,m)}(t_k) < R\big) \tag{63}$$
using Algorithm 1.
4. Compute the probability for segment j
$$\begin{aligned}\hat{\mathrm{P}}_{\mathrm{simp}}(\mathrm{C}_{(T_j,T_{j+1})}) = \frac{\Delta t_j}{3}\Big(&f(T_j) + f(T_{j+1}) \\ &+ 2\sum_{k=1}^{\frac{K_j}{2}-1} f(t_{2k}) + 4\sum_{k=1}^{\frac{K_j}{2}} f(t_{2k-1})\Big),\end{aligned} \tag{64}$$

where

$$f(t) = \sum_{m=1}^{M} \mathrm{P}\big(s_{\perp}^{(j,m)}(t) < R\big) p_{\tau^{(j,m)}}(t). \tag{65}$$

5. Set $j = j + 1$ and iterate from step 2 until $j = J$.

6. Compute the total probability of conflict

$$\hat{\mathrm{P}}_{\mathrm{simp}}(\mathrm{C}_{(0,T)}) = \sum_{j=0}^{J-1} \hat{\mathrm{P}}_{\mathrm{simp}}(\mathrm{C}_{(T_j,T_{j+1})}). \tag{66}$$

7 Simulation results

Using the notation from Figure 6 we let α and ψ deteremine the direction of the intruders and own speeds, v^{int} and v^{own} respectively, relative to line-of-sight. The corresponding relative velocity is then given by

$$\begin{aligned} v_x &= -v^{\mathrm{own}} \cos\psi - v^{\mathrm{int}} \cos\alpha, \\ v_y &= v^{\mathrm{own}} \sin\psi - v^{\mathrm{int}} \sin\alpha. \end{aligned} \tag{67}$$

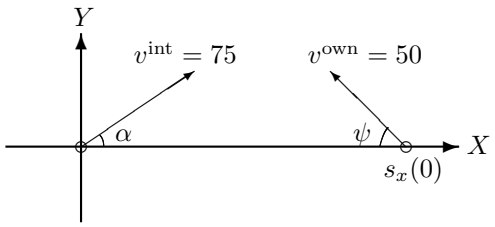

Figure 6: *Collision geometry in absolute coordinates.*

Note that the variables above should be regarded as true quantities. Moreover, we let $s_{\min}$ define the distance when the relative position is at point of closest approach. For a straight trajectory, the relation between $s_{\min}$, initial distance $s_x(0)$ and relative velocity is given by

$$\frac{s_{\min}}{s_x(0)} = -\frac{v_y}{\sqrt{v_x^2 + v_y^2}}. \tag{68}$$

Inserting (67) into (68) yields an expression for ψ according to

$$\begin{aligned} \psi = \arcsin\left(\frac{v^{\mathrm{int}}}{v^{\mathrm{own}}} \sin\big(\alpha + \arcsin\frac{s_{\min}}{s_x(0)}\big)\right) + \\ \arcsin\frac{s_{\min}}{s_x(0)}. \end{aligned} \tag{69}$$

Here we place the circular disc with $R = 150$ m at $x = 22.5$ instead of $x = 0$ to approximate the crossing of the sphere better when performing an avoidance maneuver, see Figure 7. The value $x = 22.5$ corresponds to the $x-$coordinate where a straight path starting from $s_x(0) = 1000$ is tangent to the circle.

There are two simulated scenarios with avoidance maneuvers performed with a $60-$deg turn in the horizontal plane. The first is given by Table 1 and Figure 7 and corresponds to the case with $\alpha = 0$ deg. The second is given by Table 1 and Figure 8 and corresponds to $\alpha = 25$ deg. The estimated state vector and corresponding covariance matrix given in Table 1 are the result from a tracking filter using simulated angle-only sensor measurements. Both simulations assume a reaction time set to 3 sec, and reflects the time from initiation of the avoidance maneuver until the turn actually begins. All turns are performed with 10 deg/sec, meaning that the turning time is 6 seconds. The trajectories numbered 2 and 3 correspond to the situation two and four seconds later respectively during which the constant velocity according to the initial conditions applies.

We evaluate two methods by comparing to the truth given by the Monte Carlo method according to (22) using $N = 1000000$. The two methods are

- Algorithm 2 using $M = 3$ with $a_1 = -\infty$, $a_2 = \hat{v}_x^{(j)} - \sigma_{vx}$ and $a_2 = \hat{v}_x^{(j)} + \sigma_{vx}$, Taylor expansion around $\hat{\nu}_x^{(j,m)}$ and $\Delta t = 0.1$.

- Algorithm 2 using $M = 1$, Taylor expansion around $\hat{v}_x^{(j)}$ and $\Delta t = 0.1$.

As can be deduced from Tables 2 and 3, all relative errors are $< 10\%$ for the method with $M = 3$. Recall the requirement that we seek a method capable of computing probability of NMAC with a relative accuracy at least 10% for probabilities larger than 0.01. The relative error for the method with $M = 1$ is much worse for scenario II. The worst case, ignoring II.1 which has a too small probability, occurs for II.2 which yields a relative error of approximately 75%. If the requirement is to avoid NMAC with a probability of at least 0.01, the first scenario achieves the objective if initiating an avoidance maneuver while $\hat{s}_x(0) > 1351$. In the second scenario we must initiate the turn while $\hat{s}_x(0) > 1399$.

8 Conclusions

This paper presents a novel solution to the probability of NMAC for a three-dimensional predicted relative trajectory. A NMAC occurs if the distance between two aircraft becomes less than a threshold, which is determined by the radius of a safety sphere. The method does not rely on sampling techniques, but uses theory for level-crossings. By appropriate approximations, crossing of a circular disc instead of a sphere and Taylor expansion to deal with correlation, we derive an expression for the probability of NMAC. By using the proposed expression it is possible to decrease the computational load by at least three orders of magnitude compared to the Monte Carlo solution.

Table 1: *Parameters for the simulations.*

	α	$\hat{s}_x(0)$	$\hat{v}_x(0)$	$\hat{v}_y(0)$
I.1	0	1600	-124	7.8
I.2	0	1351	-124	9.2
I.3	0	1103	-124	11.3
II.1	25	1600	-100	6.3
II.2	25	1399	-100	7.2
II.3	25	1198	-101	8.5

$$\hat{x}(0) = \begin{bmatrix} \hat{s}_x(0) & \hat{v}_x(0) & \hat{v}_y(0) & 0 \end{bmatrix}^T$$

$$P(0) = \begin{bmatrix} (\frac{\hat{s}_x(0)}{7})^2 & 0.55\frac{\hat{s}_x(0)\hat{v}_x^{(0)}}{7\cdot 8} & 0.99\frac{\hat{s}_x(0)2}{7} & 0.90\frac{\hat{s}_x(0)}{7} \\ 0.55\frac{\hat{s}_x(0)\hat{v}_x^{(0)}}{7\cdot 8} & (\frac{\hat{v}_x^{(0)}}{8})^2 & 0.59\frac{\hat{v}_x(0)2}{8} & 0.53\frac{\hat{v}_x(0)}{8} \\ 0.99\frac{\hat{s}_x(0)2}{7} & 0.59\frac{\hat{v}_x(0)2}{8} & 2^2 & 0.89\cdot 2 \\ 0.90\frac{\hat{s}_x(0)}{7} & 0.53\frac{\hat{v}_x(0)}{8} & 0.89\cdot 2 & 1^2 \end{bmatrix}$$

Table 2: *Relative error for scenario I when using Algorithm 2 with* $M = 3$, $\hat{\nu}_x^{(j,m)}$ *and* $M = 1$, $\hat{v}_x^{(j)}$.

	$s_{\min}$	$\hat{\mathrm{P}}_{\mathrm{mc}}\big(\mathrm{NMAC}_{(0,30)}\big)$	A.2(3, $\hat{\nu}_x^{(j,m)}$)	A.2(1, $\hat{v}_x^{(j)}$)
I.1	100	0.91	2%	−2%
	454	0.0008	−9%	−6%
I.2	100	0.93	1%	−2%
	357	0.0060	−10%	−2%
I.3	100	0.95	1%	−2%
	259	0.045	−2%	9%

A Appendix

A.1 Proof of Theorem 1

Using the joint probability density for s_x, v_x, v_y, v_z,

$$\begin{aligned} p_{s_x,v_x,v_y,v_z}(s,v,y,z) &= p(s,v,y,z) \\ &= \frac{1}{(2\pi)^2 \det P^{1/2}} e^{\frac{1}{2}(x-\hat{x})^T P^{-1}(x-\hat{x})} \\ &= \frac{1}{(2\pi)^2 \det P^{1/2}} e^{\frac{1}{2}|x-\hat{x}|_{P^{-1}}}, \end{aligned} \tag{70}$$

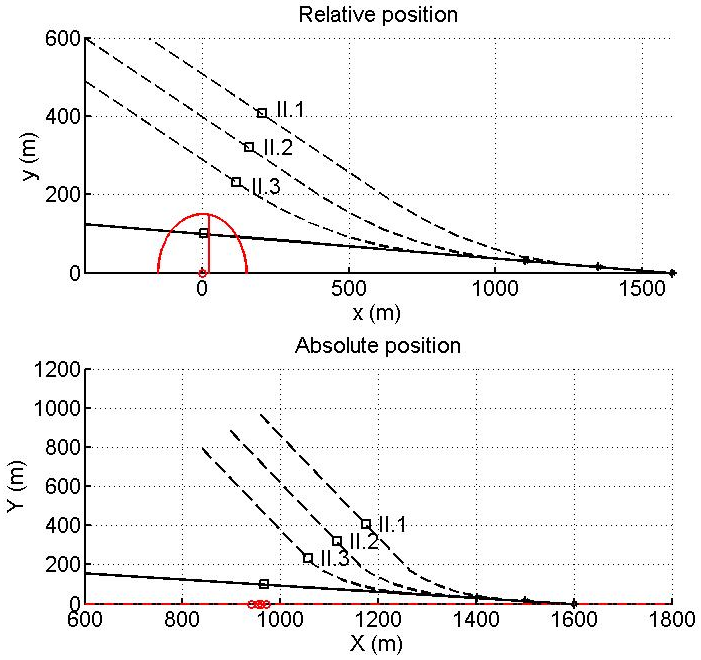

Figure 7: *The upper plot shows the relative trajectory and the lower plot shows the absolute trajectories of the intruder (dash-dotted line) and own vehicle (solid line for the straight paths and dashed for the paths with the* $60-$*deg turn). Points of closest approach are marked with circles and squares for the intruder and own vehicle respectively. Starting points are marked with stars. Simulation parameters are taken from Table 1, I.1 - I.3.*

we have

$$\begin{aligned} &\mathrm{P}(\widehat{\mathrm{NMAC}}_{(0,T)}) = \\ &\int_{-\infty}^{0} \int_{0}^{-vT} \iint_{(ty)^2+(tz)^2<R^2} p(s,v,y,z)\, dydzdsdv, \end{aligned} \tag{71}$$

where $t = -s/v$. Block diagonalize P according to

$$\begin{aligned} P &= \begin{bmatrix} I & 0 \\ K & I \end{bmatrix} \begin{bmatrix} P_{sv} & 0 \\ 0 & P'_{yz} \end{bmatrix} \begin{bmatrix} I & 0 \\ K & I \end{bmatrix}^T, \\ P^{-1} &= \begin{bmatrix} I & 0 \\ -K & I \end{bmatrix}^T \begin{bmatrix} P_{sv}^{-1} & 0 \\ 0 & (P'_{yz})^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -K & I \end{bmatrix}, \end{aligned} \tag{72}$$

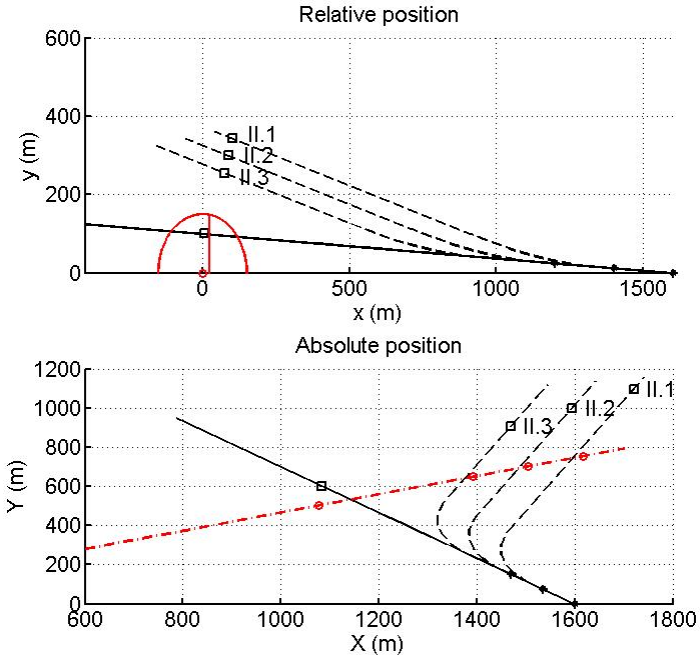

Figure 8: *The upper plot shows the relative trajectory and the lower plot shows the absolute trajectories of the intruder (dash-dotted line) and own vehicle (solid line for the straight paths and dashed for the paths with the* $60-$*deg turn). Points of closest approach are marked with circles and squares for the intruder and own vehicle respectively. Starting points are marked with stars. Simulation parameters are taken from Table 1, II.1 - II.3.*

where $P'_{yz} = P_{yz} - C^T P_{sv}^{-1} C$ and $K = C^T P_{sv}^{-1}$. The inner double integral over y, z is, using the block diagonalized covariance matrix, given by

$$\begin{aligned} F(s,v) &= \iint\limits_{(ty)^2+(tz)^2<R^2} p_{s_y,v_y|s_x,v_x}(y,z|s,v)dydz \\ &= \iint\limits_{(ty)^2+(tz)^2<R^2} p(y,z|s,v)dydz, \end{aligned} \tag{73}$$

where

$$p(y,z|s,v) = \frac{1}{2\pi \det(P'_{yz})^{1/2}} e^{-\frac{1}{2}\left\| \begin{bmatrix} y-\hat{v}_y \\ z-\hat{v}_z \end{bmatrix} - K \begin{bmatrix} s-\hat{s}_x \\ v-\hat{v}_x \end{bmatrix} \right\|_{(P'_{yz})^{-1}}}. \tag{74}$$

Table 3: *Relative error for scenario II when using Algorithm 2 with $M = 3$, $\hat{\nu}_x^{(j,m)}$ and $M = 1$, $\hat{v}_x^{(j)}$.*

	$s_{\min}$	$\hat{\mathrm{P}}_{\mathrm{mc}}\big(\mathrm{NMAC}_{(0,30)}\big)$	A.2(3, $\hat{\nu}_x^{(j,m)}$)	A.2(1, $\hat{v}_x^{(j)}$)
II.1	100	0.87	2%	−1%
	358	0.0028	2%	103%
II.2	100	0.90	2%	−2%
	312	0.0089	−6%	75%
II.3	100	0.92	2%	−2%
	266	0.025	−7%	49%

The probability of $\widehat{\mathrm{NMAC}}_{(0,T)}$ is given by

$$\begin{aligned}&\mathrm{P}(\widehat{\mathrm{NMAC}}_{(0,T)})\\&= \int_{-\infty}^{0}\int_{0}^{-vT} F(s,v)p_{s_x,v_x}(s,v)dsdv.\end{aligned} \tag{75}$$

Change variable from s to $t = -s/v$ which yields

$$\begin{aligned}&\mathrm{P}(\widehat{\mathrm{NMAC}}_{(0,T)})\\&= \int_{-\infty}^{0}\int_{0}^{T} -vF(-vt,v)p_{s_x,v_x}(-vt,v)dtdv\\&= \int_{0}^{T}\int_{-\infty}^{0} -vF(-vt,v)p_{s_x,v_x}(-vt,v)dvdt.\end{aligned} \tag{76}$$

Consider partition the inner integral over v in M intervals, i.e.

$$\begin{aligned}&\int_{-\infty}^{0} -vF(-vt,v)p_{s_x,v_x}(-vt,v)dv\\&= \sum_{m=1}^{M}\int_{a_m}^{b_m} -vF(-vt,v)p_{s_x,v_x}(-vt,v)dvdt,\end{aligned} \tag{77}$$

where $a_1 = -\infty$ and $b_M = 0$. For each partition, Taylor expansion of $F(-vt, v)$ around $\hat{\nu}_x^{(m)}(t)$ yields

$$F(-vt,v) = \sum_{r=0}^{\infty}\frac{1}{r!}\frac{\partial^r F(-vt,v)}{\partial v^r}\bigg|_{v=\hat{\nu}_x^{(m)}(t)}\big(v-\hat{\nu}_x^{(m)}(t)\big)^r. \tag{78}$$

Inserting into (77) yields

$$\begin{aligned}
&\int_{-\infty}^{0} -vF(-vt,v)p_{s_x,v_x}(-vt,v)dv \\
&= \sum_{m=1}^{M}\sum_{r=0}^{\infty}\frac{1}{r!}\frac{\partial^r F(-vt,v)}{\partial v^r}\Big|_{v=\hat{\nu}_x^{(m)}(t)} \\
&\cdot \int_{a_m}^{b_m} -v\big(v-\hat{\nu}_x^{(m)}(t)\big)^r p_{s_x,v_x}(-vt,v)dvdt,
\end{aligned} \tag{79}$$

Choose $\hat{\nu}_x^{(m)}(t)$ such that

$$\int_{a_m}^{b_m} -v\big(v-\hat{\nu}_x^{(m)}(t)\big)p_{s_x,v_x}(-vt,v)dv = 0, \tag{80}$$

which means that the first order terms corresponding to $r = 1$ in (79) are eliminated. The result is

$$\begin{aligned}
&\mathrm{P}(\widehat{\mathrm{NMAC}}_{(0,T)}) \\
&= \int_0^T \sum_{m=1}^{M} F\big(-\hat{\nu}_x^{(m)}(t)t, \hat{\nu}_x^{(m)}(t)\big) \\
&\cdot \int_{a_m}^{b_m} -vp_{s_x,v_x}(-vt,v)dvdt + \mathrm{P}_M \\
&= \int_0^T \sum_{m=1}^{M} F\big(-\hat{\nu}_x^{(m)}(t)t, \hat{\nu}_x^{(m)}(t)\big)p_\tau^{(m)}(t)dt + \mathrm{P}_M,
\end{aligned} \tag{81}$$

where

$$\begin{aligned}
\mathrm{P}_M \leq &\int_0^T \sum_{m=1}^{M}\int_{a_m}^{b_m} \frac{1}{2}\frac{\partial^2 F(-vt,v)}{\partial v^2}\Big|_{v=\mu^{(m)}(t)} \\
&\cdot -v\big(v-\mu^{(m)}(t)\big)^2 p_{s_x,v_x}(-vt,v)dvdt
\end{aligned} \tag{82}$$

for $a_m < \mu^{(m)}(t) < b_m$.

References

[1] J. Abate and W. Whitt. The fourier-series method for inverting transforms of probability distributions. *Queueing Systems*, 10(1-2), 1992.

[2] M. Abramowitz and I.A. Stegun, editors. *Handbook of Mathematical Functions With Formulas, Graphs and Mathematical Tables*. US Department of Commerce, 10 edition, 1964.

[3] T Arino, K. Carpenter, S. Chabert, H. Hutchinson, T. Miquel, B. Raynaud, K. Rigotti, and E. Vallauri. ACAS analysis programme ACASA project work package 1 final report on studies on the safety of ACAS in Europe. Technical Report Version 1.3, EEC Brétigny, March 2002.

[4] ASTM. *F2411-07 Standard Specification for Design and Performance of an Airborne Sense-and-Avoid System*, 2007.

[5] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[6] C. Carbone, U. Ciniglio, F. F. Corraro, and S. Luongo. A novel 3d geometric algorithm for aircraft autonomous collision avoidance. In *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006.

[7] G. Chaloulos and J. Lygeros. Effect of wind correlation on aircraft conflict probability. *Journal of Guidance, Control, and Dynamics*, 30(6), 2007.

[8] K. Chan. Analytical expressions for computing spacecraft collision probabilities. In *Proceedings of the 11th Annual AAS/AIAA Space Flight/Mechanics Meeting*, 2001.

[9] FAA. *Aeronautical Information Manual - Official Guide to Basic Flight Information and ATC Procedures*, 2008.

[10] A. Gut. *An Intermediate Course in Probability*. Springer-Verlag, 1995.

[11] D.V. Hinkley. On the ratio of two correlated normal random variables. *Biometrika*, 56(3):635–639, 1969.

[12] J.P. Imhof. Computing the distribution of quadratic forms in normal variables. *Biometrika*, 48(3/4):419–426, 1961.

[13] J. Jansson and F. Gustafsson. A framework and automotive application of collision avoidance decision making. *Automatica*, 2008.

[14] J. Krozel and M. Peters. Strategic conflict detection and resolution for free flight. In *Proceedings of the 36th IEEE Conference on Decision and Control*, 1997.

[15] J. K. Kuchar and A. C. Drumm. The traffic alert and collision avoidance system. *Lincoln Laboratory Journal*, 16(2), 2007.

[16] J.K. Kuchar and L.C. Yang. A review of conflict detection and resolution methods. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):179–189, 2000.

[17] P-J. Nordlund and F. Gustafsson. Probabilistic conflict detection for piecewise straight paths. *Submitted to Automatica*, 2008.
http://www.control.isy.liu.se/research/reports/2008/2871.pdf.

[18] P-J. Nordlund and F. Gustafsson. The probability of near midair collisions using level-crossings. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008.

[19] R. A. Paielli. Algorithms for tactical conflict resolution and strategic conflict probability reduction. In *Proceedings of the 1st AIAA Aircraft, Technology Integration and Operations Forum*, 2001.

[20] M. Prandini, J Hu, J Lygeros, and S. Sastry. A probabilistic approach to aircraft conflict detection. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):199–220, 2000.

[21] O. Shakernia, W-Z. Chen, and V. M. Raska. Passive ranging for UAV sense and avoid applications. In *Proceedings of Infotech@Aerospace*, 2005.

[22] L.A. Waller, B.W. Turnbull, and J.M. Hardin. Obtaining distribution functions by numerical inversion of characteristic functions with applications. *The American Statistician*, 49(4), 1995.

[23] L. Yang, J. H. Yang, J. Kuchar, and E. Feron. A real-time Monte Carlo implementation for computing probability of conflict. In *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, 2004.

PhD Dissertations
Division of Automatic Control
Linköping University

M. Millnert: Identification and control of systems subject to abrupt changes. Thesis No. 82, 1982. ISBN 91-7372-542-0.
A. J. M. van Overbeek: On-line structure selection for the identification of multivariable systems. Thesis No. 86, 1982. ISBN 91-7372-586-2.
B. Bengtsson: On some control problems for queues. Thesis No. 87, 1982. ISBN 91-7372-593-5.
S. Ljung: Fast algorithms for integral equations and least squares identification problems. Thesis No. 93, 1983. ISBN 91-7372-641-9.
H. Jonson: A Newton method for solving non-linear optimal control problems with general constraints. Thesis No. 104, 1983. ISBN 91-7372-718-0.
E. Trulsson: Adaptive control based on explicit criterion minimization. Thesis No. 106, 1983. ISBN 91-7372-728-8.
K. Nordström: Uncertainty, robustness and sensitivity reduction in the design of single input control systems. Thesis No. 162, 1987. ISBN 91-7870-170-8.
B. Wahlberg: On the identification and approximation of linear systems. Thesis No. 163, 1987. ISBN 91-7870-175-9.
S. Gunnarsson: Frequency domain aspects of modeling and control in adaptive systems. Thesis No. 194, 1988. ISBN 91-7870-380-8.
A. Isaksson: On system identification in one and two dimensions with signal processing applications. Thesis No. 196, 1988. ISBN 91-7870-383-2.
M. Viberg: Subspace fitting concepts in sensor array processing. Thesis No. 217, 1989. ISBN 91-7870-529-0.
K. Forsman: Constructive commutative algebra in nonlinear control theory. Thesis No. 261, 1991. ISBN 91-7870-827-3.
F. Gustafsson: Estimation of discrete parameters in linear systems. Thesis No. 271, 1992. ISBN 91-7870-876-1.
P. Nagy: Tools for knowledge-based signal processing with applications to system identification. Thesis No. 280, 1992. ISBN 91-7870-962-8.
T. Svensson: Mathematical tools and software for analysis and design of nonlinear control systems. Thesis No. 285, 1992. ISBN 91-7870-989-X.
S. Andersson: On dimension reduction in sensor array signal processing. Thesis No. 290, 1992. ISBN 91-7871-015-4.
H. Hjalmarsson: Aspects on incomplete modeling in system identification. Thesis No. 298, 1993. ISBN 91-7871-070-7.
I. Klein: Automatic synthesis of sequential control schemes. Thesis No. 305, 1993. ISBN 91-7871-090-1.
J.-E. Strömberg: A mode switching modelling philosophy. Thesis No. 353, 1994. ISBN 91-7871-430-3.
K. Wang Chen: Transformation and symbolic calculations in filtering and control. Thesis No. 361, 1994. ISBN 91-7871-467-2.
T. McKelvey: Identification of state-space models from time and frequency data. Thesis No. 380, 1995. ISBN 91-7871-531-8.
J. Sjöberg: Non-linear system identification with neural networks. Thesis No. 381, 1995. ISBN 91-7871-534-2.
R. Germundsson: Symbolic systems – theory, computation and applications. Thesis No. 389, 1995. ISBN 91-7871-578-4.
P. Pucar: Modeling and segmentation using multiple models. Thesis No. 405, 1995. ISBN 91-7871-627-6.
H. Fortell: Algebraic approaches to normal forms and zero dynamics. Thesis No. 407, 1995. ISBN 91-7871-629-2.

A. Helmersson: Methods for robust gain scheduling. Thesis No. 406, 1995. ISBN 91-7871-628-4.
P. Lindskog: Methods, algorithms and tools for system identification based on prior knowledge. Thesis No. 436, 1996. ISBN 91-7871-424-8.
J. Gunnarsson: Symbolic methods and tools for discrete event dynamic systems. Thesis No. 477, 1997. ISBN 91-7871-917-8.
M. Jirstrand: Constructive methods for inequality constraints in control. Thesis No. 527, 1998. ISBN 91-7219-187-2.
U. Forssell: Closed-loop identification: Methods, theory, and applications. Thesis No. 566, 1999. ISBN 91-7219-432-4.
A. Stenman: Model on demand: Algorithms, analysis and applications. Thesis No. 571, 1999. ISBN 91-7219-450-2.
N. Bergman: Recursive Bayesian estimation: Navigation and tracking applications. Thesis No. 579, 1999. ISBN 91-7219-473-1.
K. Edström: Switched bond graphs: Simulation and analysis. Thesis No. 586, 1999. ISBN 91-7219-493-6.
M. Larsson: Behavioral and structural model based approaches to discrete diagnosis. Thesis No. 608, 1999. ISBN 91-7219-615-5.
F. Gunnarsson: Power control in cellular radio systems: Analysis, design and estimation. Thesis No. 623, 2000. ISBN 91-7219-689-0.
V. Einarsson: Model checking methods for mode switching systems. Thesis No. 652, 2000. ISBN 91-7219-836-2.
M. Norrlöf: Iterative learning control: Analysis, design, and experiments. Thesis No. 653, 2000. ISBN 91-7219-837-0.
F. Tjärnström: Variance expressions and model reduction in system identification. Thesis No. 730, 2002. ISBN 91-7373-253-2.
J. Löfberg: Minimax approaches to robust model predictive control. Thesis No. 812, 2003. ISBN 91-7373-622-8.
J. Roll: Local and piecewise affine approaches to system identification. Thesis No. 802, 2003. ISBN 91-7373-608-2.
J. Elbornsson: Analysis, estimation and compensation of mismatch effects in A/D converters. Thesis No. 811, 2003. ISBN 91-7373-621-X.
O. Härkegård: Backstepping and control allocation with applications to flight control. Thesis No. 820, 2003. ISBN 91-7373-647-3.
R. Wallin: Optimization algorithms for system analysis and identification. Thesis No. 919, 2004. ISBN 91-85297-19-4.
D. Lindgren: Projection methods for classification and identification. Thesis No. 915, 2005. ISBN 91-85297-06-2.
R. Karlsson: Particle Filtering for Positioning and Tracking Applications. Thesis No. 924, 2005. ISBN 91-85297-34-8.
J. Jansson: Collision Avoidance Theory with Applications to Automotive Collision Mitigation. Thesis No. 950, 2005. ISBN 91-85299-45-6.
E. Geijer Lundin: Uplink Load in CDMA Cellular Radio Systems. Thesis No. 977, 2005. ISBN 91-85457-49-3.
M. Enqvist: Linear Models of Nonlinear Systems. Thesis No. 985, 2005. ISBN 91-85457-64-7.
T. B. Schön: Estimation of Nonlinear Dynamic Systems — Theory and Applications. Thesis No. 998, 2006. ISBN 91-85497-03-7.
I. Lind: Regressor and Structure Selection — Uses of ANOVA in System Identification. Thesis No. 1012, 2006. ISBN 91-85523-98-4.
J. Gillberg: Frequency Domain Identification of Continuous-Time Systems Reconstruction and Robustness. Thesis No. 1031, 2006. ISBN 91-85523-34-8.
M. Gerdin: Identification and Estimation for Models Described by Differential-Algebraic Equations. Thesis No. 1046, 2006. ISBN 91-85643-87-4.

C. Grönwall: Ground Object Recognition using Laser Radar Data – Geometric Fitting, Performance Analysis, and Applications. Thesis No. 1055, 2006. ISBN 91-85643-53-X.
A. Eidehall: Tracking and threat assessment for automotive collision avoidance. Thesis No. 1066, 2007. ISBN 91-85643-10-6.
F. Eng: Non-Uniform Sampling in Statistical Signal Processing. Thesis No. 1082, 2007. ISBN 978-91-85715-49-7.
E. Wernholt: Multivariable Frequency-Domain Identification of Industrial Robots. Thesis No. 1138, 2007. ISBN 978-91-85895-72-4.
D. Axehill: Integer Quadratic Programming for Control and Communication. Thesis No. 1158, 2008. ISBN 978-91-85523-03-0.
G. Hendeby: Performance and Implementation Aspects of Nonlinear Filtering. Thesis No. 1161, 2008. ISBN 978-91-7393-979-9.
J. Sjöberg: Optimal Control and Model Reduction of Nonlinear DAE Models. Thesis No. 1166, 2008. ISBN 978-91-7393-964-5.
D. Törnqvist: Estimation and Detection with Applications to Navigation. Thesis No. 1216, 2008. ISBN 978-91-7393-785-6.