Forward Filtering and Backward Smoothing for Radar Applications

Master of Science Thesis

Viktor Kärnstrand

Department of Electrical Engineering
Division of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY

Department of Systems Design
Division of Concept Development
SAAB ELECTRONIC DEFENCE SYSTEMS

Gothenburg, Sweden 2012
Report No. EX034

Abstract

This study presents and evaluates the performance of various solutions to the filtering and smoothing problems from an air surveillance radar perspective. In the theory part, the Kalman Filter and the Rauch-Tung-Striebel smoother equations are derived for linear models. Thereafter, we derive the corresponding equations for nonlinear models using the Unscented Transform. These solutions are then expanded to the Interactive Multiple Model framework. The resulting algorithms are evaluated through Monte Carlo simulation using six benchmark scenarios representing the classes; large aircraft, agile commercial aircraft, medium bomber and fighter jet. Optimal process model parameters are determined using an evolutionary optimization method. The resulting process model parameters indicates that the use of multiple, and nonlinear models each contribute to a more adaptive and accurate description of the process. The results from the evaluation clearly show that the use of nonlinear process models and multiple models each improve the process state estimates, in particular for smoothing.

Acknowledgements

First of all I would like to thank all the wonderful people I have met during my graduate studies. You actually made it seem like long nights in school was the best of activities. I would also like to thank my parents who have supported me all the way throughout my school years. Special thanks to Daniel Svensson and Johannes Wintenby at Saab Electronic Defence Systems, and Lennart Svensson at Chalmers University, for your knowledge and inspiration. But most of all, thanks to my supervisor Johan Degerman for your commitment, good company and for believing in this study.

Viktor Kärnstrand, Göteborg 2012-04-30

Contents

1 Introduction **1**
1.1 Aim of study . . . 3
1.2 Thesis outline . . . 3

2 Problem statement **5**

3 Conceptual solution **7**
3.1 Filtering . . . 7
3.2 Smoothing . . . 8

4 Optimal solution for linear systems **9**
4.1 The Kalman Filter . . . 10
4.1.1 MLE derivation . . . 10
4.1.2 MMSE derivation . . . 11
4.2 The Rauch-Tung-Striebel smoother . . . 12

5 Approximate solution for nonlinear systems **14**
5.1 The Unscented Transform . . . 15
5.2 The prediction step . . . 16
5.3 Unscented Kalman Gain . . . 17
5.4 Unscented Smoothing Gain . . . 17

6 Solution for multiple models **19**
6.1 IMM Filtering . . . 20
6.1.1 Conceptual solution . . . 20
6.1.2 Algorithm . . . 21
6.2 IMM Smoothing . . . 22
6.2.1 Conceptual solution . . . 22
6.2.2 Algorithm . . . 24

7 Process models **26**
7.1 Discretization of linear differential equations 26
7.2 Discretized linearization 28
7.3 Degrees of freedom for process noise 29

8 Evaluation **31**
8.1 Evaluation data 32
8.2 Process model selection 32
8.3 Process parameter estimation 33
8.4 Evaluated trackers 33

9 Results **38**

10 Conclusion **43**
10.1 Computational complexity 43
10.2 Performance and parameter settings 44
10.3 Future work 45

A Application specific process models **49**
A.1 Constant Acceleration (CA) 49
A.2 Constant Velocity (CV) 50
A.3 Contant Acceleration Local Coordinates (CAL) 52
A.4 Coordinated Turn (CT) 53

B Figures **56**

1

Introduction

The tracker is the software component in a radar system which associates the sensor measurements with the observed objects and keeps track of those objects. In addition, the tracker extracts information about the kinematic parameters of the observed objects over time, typically in a probabilistic manner. In the case where the sensor is a doppler radar, for example, measurements contain information about radial distance to the object from the sensor, the angle of the object relative the sensor, and possibly also the velocity of the object in the radial direction. We have no direct information about the acceleration from such measurements, and neither do we have any information about the velocity of the object in angular direction. On top of that the measurements are typically noisy, which makes it even harder to extract knowledge about kinematics.

Fortunately, we may add one piece to the puzzle, namely our knowledge about the laws of physics, which we may use to construct a model for the underling process that we are observing. In this case the process referres to the kinematics of an oberved airborne object, which is assumed to be a continous markov process described by a dynamical process model in state space. This dynamical process is not assumed to be deterministic, but contains a stochastic term describing the process model uncertainty. We may also have knowledge about the sensor that provides us with the measurements. We may thus also specify a model describing the process in which the sensor maps the probabilistic description of the process state, from state space to measurement space. We assume that this mapping is not deterministic but includes an additional stochastic term to account for measurement uncertainty.

We thus have two stochastic models to describe the process and the measurements. To be able to use these stochastic models we need a Bayesian framework that yields optimal probability distributions in state space for the process state, given measurements, by using the process- and measurement models. The Bayesian framwork that works recursively forward in time to calculate optimal statistical distributions for the process is referred to as filtering. The Bayesian framework that works recursively backwards in time to correct for the information gained through latter measurements is referred to as smoothing. These probability distributions can then be optimized in some sense to produce process state estimates, see figure 1.1 for a visual example.

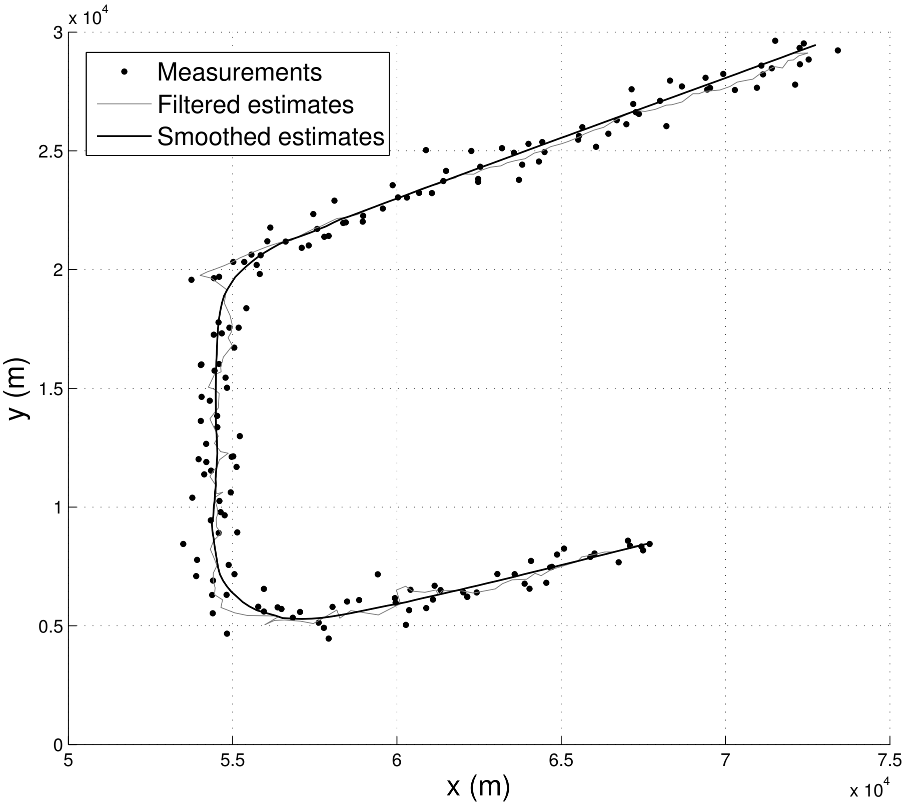


Figure 1.1: Example on filtering and smoothing position estimates for benchmark scenario 1. See section 8.1 for information on the benchmark scenarios.

For linear process and measurement models, and time-uncorrelated Gaussian process- and measurement noise, there are optimal solutions for both the filtering and the smoothing problems, namely the *Kalman Filter* (KF) [1] and the *Rauch-Tung-Striebel* (RTS) [2] smoother, respectively. The application of Bayesian filtering and smoothing in this study is restricted to the domain of air surveillance radar systems, which means that the sensor is typically providing measurements in nonlinear coordinates. And the process, i.e. the kinematics of the observed airborne object in this case, is typically best described by nonlinear models. In addition to this, the airborne object is most certainly in different modes of dynamics during the flight, which might be best described by a collection of process models rather than a single one. Luckily, there are modifications to the *Kalman Filter* that solves these problems.

There are different methods to deal with nonlinear process- and measurement projections, but in this study we will focus on the *Unscented Kalman Filter* (UKF) proposed by *Julier et al* [9]. The UKF uses a sampled version of the probability distribution at hand to produce approximate solutions to the filtering problem. There are a number of different proposed approximate solutions to the smoothing problem for nonlinear models, among which we focus on the Unscented RTS smoothing solution proposed by *Särkkä* [3].

The problem of filtering with multiple models also has a number of approximate solutions which can be grouped into the subclasses of hard decision- and soft descision filters. Basically, hard decision means that one of the multiple filtered probability distributions is chosen at each instant of time, while in soft decision filtering a weighted sum of the filtered probability distributions constitutes the optimal probability distribution. The weights being the filter probabilities at each instant of time. We will focus on the *Interactive Multiple Model* (IMM), a soft decision filter derived from the assumption that the process dynamics behaves as a switch markov system. We will also focus on the proposed solution to the problem of multiple model filtering proposed by *Nadarajah et al* [4], built upon the model of the IMM filter [7]. The figures 1.2(a)-1.2(b) show how probabilities for three filters, each containing one distrinct process model, changes over time, and how smoothing decreases the unscertainty during straight course.

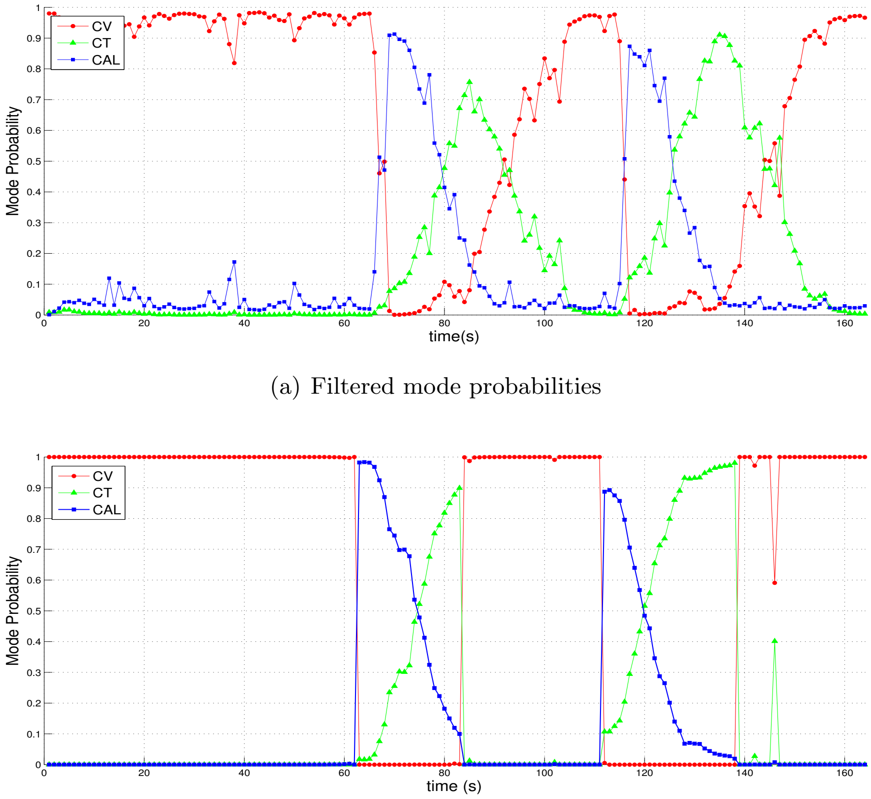
1.1 Aim of study

The aim of this study is to determine the improvement in process state estimates using nonlinear process models, multiple models and smoothing, separately and combined, for the air surveillance radar application.

1.2 Thesis outline

The outline of the thesis is as follows. In Chapter 2, ”Problem statement”, we outlay a formal description of the problem at hand, namely the estimation of the state of a process, given a number of measurements. In Chapter 3, ”Conceptual solution”, we present the Bayesian frameworks which are the general solutions to the filtering and the smoothing problems, respectively.

We then derive the optimal solutions to the filtering and the smoothing problems for linear process models in Chapter 4, ”Optimal solution for linear systems”. It is also in this chapter that we will familiarize ourselves with the *Kalman Filter* and the *Rauch-Tung-Striebel* smoother. In the following Chapter 5, ”Approximate solution for nonlinear systems”, we discuss the *Unscented Transform* and how it is used for nonlinear model



(a) Filtered mode probabilities

(b) Smoothed mode probabilities

Figure 1.2: Example on soft decision multiple model filtering, and smoothing. The figure shows the mode probabilities for three filters over benchmark scenario 1. See section 8.1 for information on the benchmark scenarios.

filtering and smoothing. We thus derive the equations for the *Unscented Kalman Filter* and the *Unscented Rauch-Tung-Striebel* smoother proposed by *Särkkä* [3].

Thereafter, in Chapter 6, "Solution for multiple models", we develop a bayesian framework for a switch markov system which allows us to use multiple filters. First we derive the equations for the *IMM* filter and then we discuss the *IMM* smoother proposed by *Nadarajah et al.* In the last theory Chapter 7, "Process models", we deal with how to discretize differential equations which describe the dynamical process in state space, so as to be applicable for filtering. After a whole lot of theory, we discuss the evaluation procedure in Chapter 8, and present our findings in Chapter 9.

2

Problem statement

We are observing a time continous process, $\mathbf{x}(t)$ over time by measuring the system at times $\{t_i\}_{i=1}^{N}$ generating measurements $\mathbf{Z}^N$,

$$\begin{aligned} \mathbf{Z}^N &= \mathbf{z}(t_1), \mathbf{z}(t_2), ..., \mathbf{z}(t_k), ..., \mathbf{z}(t_N) \\ &= \mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k, ..., \mathbf{z}_N. \end{aligned} \tag{2.1}$$

We assume that measurement $\mathbf{z}_k$ only depends on the process state $\mathbf{x}_k = \mathbf{x}(t_k)$, at the same instant of time, i.e $p(\mathbf{z}_k|\mathbf{x}_k,...) = p(\mathbf{z}_k|\mathbf{x}_k)$. This dependence $p(\mathbf{z}_k|\mathbf{x}_k)$, which we shall refer to as a measurement model, we describe with a relation $\mathbf{z}_k = h_k(\mathbf{x}_k) + \mathbf{w}_k$. The term $\mathbf{w}_k$ represents the measurement noise at time t_k, which we assume is Gaussian. The statement that $\mathbf{z}_k$ only depends on $\mathbf{x}_k$ also implies that the measurement noise does not have any time correlation. These assumption are summarized in equation (2.2).

$$\begin{aligned} &\mathbf{z}_k = h_k(\mathbf{x}_k) + \mathbf{w}_k \\ &\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \\ &\mathbf{E}\{\mathbf{w}_i \mathbf{w}_j^T\} = \delta_{ij} \mathbf{R}_i \end{aligned} \tag{2.2}$$

We further assume that the process $\mathbf{x}$ is markov, i.e $p(\mathbf{x}_k|\mathbf{x}_{k-1},...) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$. The probability density function $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, which we shall call the measurement model, we describe by a relation $\mathbf{x}_k = f_{k-1}(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1}$. The term $\mathbf{v}_{k-1}$ represents the maneuvering noise between times t_{k-1} and t_k, and is assumed to be Gaussian. The markov assumption also implies that the maneuvering noise does not have any time correlation. These assumption are summarized in equation (2.3).

$$\begin{aligned} &\mathbf{x}_k = f_{k-1}(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1} \\ &\mathbf{v}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1}) \\ &\mathbf{E}\{\mathbf{v}_i \mathbf{v}_j^T\} = \delta_{ij} \mathbf{Q}_i \end{aligned} \tag{2.3}$$

See figure 2.1 for schematic explaination of the model assumptions. Given these model assumptions, the problem at hand is to find the distribution $p(\mathbf{x}_k)$, that best describes our knowledge about the process state $\mathbf{x}_k$, at time t_k. The solution to this problem differs depending on which measurements we have access to at the moment. Without loss of generality we may discuss the three cases:

Prediction Find the probability density function, $p(\mathbf{x}_k|\mathbf{Z}^{k-1})$, for the process state $\mathbf{x}_k$ at time t_k, given that we know all measurements up until time t_{k-1}.

Filtering Find the probability density function, $p(\mathbf{x}_k|\mathbf{Z}^{k})$, for the process state $\mathbf{x}_k$ at time t_k, given that we know all measurements up until time t_k.

Smoothing Find the probability density function, $p(\mathbf{x}_k|\mathbf{Z}^{N})$, for the process state $\mathbf{x}_k$ at time t_k, given that we know all measurements up until time t_N (N > k).

To solve the problem at hand for each of these three cases, we first want to construct a general solution, without considering specific classes of process- or measurement models. Secondly we wish to apply the general solution to linear-, and then to nonlinear models, to construct recursive algortihms. Last, we wish to construct a solution for a multiple model description of the system. This means that we have to restate the problem and find an augmented solution that takes into account a number of process- and measurement models, rather than one.

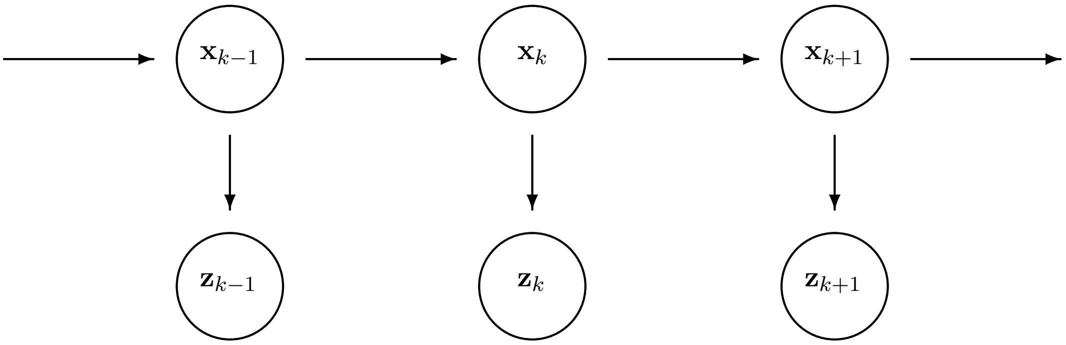


Figure 2.1: Visual schematic for the assumed model. The markov process $\mathbf{x}$ is being measured at a discrete number of times. The measurement $\mathbf{z}_k$ is assumed to be dependent only on the process state $\mathbf{x}_k$ at the same instant of time.

3

Conceptual solution

In order to derive the solution to the filtering and smoothing problems, we may start without making any assumptions about the process model or the measurement model other than the ones made in the problem statement. In this section we use a bayesian approach to form conceptual solutions to the filtering and the smoothing problems.

3.1 Filtering

The filtered probability distribution $p(\mathbf{x}_k|\mathbf{Z}^k)$ can be described as (3.1), according to Bayes' formula.

$$p(\mathbf{x}_k|\mathbf{Z}^k) = p(\mathbf{x}_k|\mathbf{z}_k, \mathbf{Z}^{k-1}) = \frac{p(\mathbf{z}_k, \mathbf{x}_k|\mathbf{Z}^{k-1})}{p(\mathbf{z}_k|\mathbf{Z}^{k-1})} \tag{3.1}$$

The joint distribution in the numerator of (3.1) may be rewritten as (3.3), using the definition (3.2) for joint probability.

$$P(A \cap B) = P(A|B)P(B) \tag{3.2}$$

$$p(\mathbf{z}_k, \mathbf{x}_k|\mathbf{Z}^{k-1}) = p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{Z}^{k-1})p(\mathbf{x}_k|\mathbf{Z}^{k-1}) \tag{3.3}$$

To utilize the knowledge that we have about the markov process $\mathbf{x}$, we marginalize $p(\mathbf{x}_k|\mathbf{Z}^{k-1})$ over $\mathbf{x}_{k-1}$, see (3.4). The second equality is true due to the definition (3.2) for joint probabilities. The factor $p(\mathbf{x}_{k-1}|\mathbf{Z}^{k-1})$ in the integrand is the posteriori distribution from the prior timestep.

$$p(\mathbf{x}_k|\mathbf{Z}^{k-1}) = \int p(\mathbf{x}_k, \mathbf{x}_{k-1}|\mathbf{Z}^{k-1})d\mathbf{x}_{k-1} = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{Z}^{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}^{k-1})d\mathbf{x}_{k-1} \tag{3.4}$$

Since we assumed a first order markov system, the depence on $\mathbf{Z}^{k-1}$, in $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{Z}^{k-1})$ and $p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{Z}^{k-1})$, can be ignored and can thus be written as (3.5). Since if we are filtering forward and assume we know $\mathbf{x}_k$ with absolute certainty there is no need for measurements $\mathbf{Z}^{k-1}$. This is formally states in equation (3.5).

$$\begin{aligned} p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{Z}^{k-1}) &= p(\mathbf{z}_k|\mathbf{x}_k) \\ p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{Z}^{k-1}) &= p(\mathbf{x}_k|\mathbf{x}_{k-1}) \end{aligned} \tag{3.5}$$

$p(\mathbf{z}_k|\mathbf{x}_k)$, in words, is the probability distribution for the measurement $\mathbf{z}_k$ at time step k, given one knows the process state $\mathbf{x}_k$ at the same instant of time. This density is referred to as the measurement model. $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the probability distribution for the process state $\mathbf{x}_k$ at time step k, given that we know the process state $\mathbf{x}_{k-1}$ at the time of the preceding measurement. This density is referred to as the process model.

3.2 Smoothing

The probability distribution of interest is $p(\mathbf{x}_k|\mathbf{Z}^N)$, i.e the smoothed probability distribution for the process at time step k given that we know all measurements up to timestep N. We begin by marginalizing over $\mathbf{x}_{k+1}$.

$$p(\mathbf{x}_k|\mathbf{Z}^N) = \int p(\mathbf{x}_k, \mathbf{x}_{k+1}|\mathbf{Z}^N) d\mathbf{x}_{k+1} \tag{3.6}$$

The joint distribution for $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ can be divided into two factors through the definition of conditional probability (3.2).

$$p(\mathbf{x}_k, \mathbf{x}_{k+1}|\mathbf{Z}^N) = p(\mathbf{x}_k|\mathbf{x}_{k+1}, \mathbf{Z}^N)p(\mathbf{x}_{k+1}|\mathbf{Z}^N) \tag{3.7}$$

We may ignore measurements $\{\mathbf{z}_i\}_{i=k+1}^N$ in $p(\mathbf{x}_k|\mathbf{x}_{k+1}, \mathbf{Z}^N)$, since they are redundant if we know $\mathbf{x}_{k+1}$ for a markov system. This insight, plus Bayes' formula leads to (3.8).

$$p(\mathbf{x}_k|\mathbf{x}_{k+1}, \mathbf{Z}^N) = p(\mathbf{x}_k|\mathbf{x}_{k+1}, \mathbf{Z}^k) = \frac{p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{Z}^k)p(\mathbf{x}_k|\mathbf{Z}^k)}{p(\mathbf{x}_{k+1}|\mathbf{Z}^k)} \tag{3.8}$$

All this added up gives us the joint probability distribution (3.9).

$$p(\mathbf{x}_k, \mathbf{x}_{k+1}|\mathbf{Z}^N) = \frac{p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{Z}^k)p(\mathbf{x}_k|\mathbf{Z}^k)}{p(\mathbf{x}_{k+1}|\mathbf{Z}^k)} p(\mathbf{x}_{k+1}|\mathbf{Z}^N) \tag{3.9}$$

In the joint probability distribution (3.9), $p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{Z}^k) = p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ is the predicted distribution for the process state $\mathbf{x}_{k+1}$ at timestep $k+1$ given that we know $\mathbf{x}_k$ with absolute certainty. $p(\mathbf{x}_{k+1}|\mathbf{Z}^k)$ is the predicted distribution for the process at timestep $k+1$ given the filtered distribution for $\mathbf{x}_k$ and $p(\mathbf{x}_k|\mathbf{Z}^k)$ is the filtered distribution at timestep k. $p(\mathbf{x}_{k+1}|\mathbf{Z}^N)$ is the smoothed distribution at timestep $k+1$, which we know at this point, from the previous backward smoothing step, thus giving us a recursive framework for backward smoothing.

4

Optimal solution for linear systems

THE solution to the filtering problem for linear models is known as the *Kalman Filter*. There also exists a solution for the smoothing problem for linear models, known as the *Rauch-Tung-Striebel* smoother. To derive the filtering and smooting equations, we assume that the process and measurement models are linear and written on the form (4.1). Other than that, we keep the assumptions that the process noise aswell as the measurement noise is Gaussian and uncorrelated over time, as expressed in (4.2). In words, the stochastic normal distributed term $\mathbf{v}_{k-1}$, with covariance $\mathbf{Q}_{k-1}$, is the process noise in the interval in between measurement $k-1$ and k. The stochastic normal distributed term $\mathbf{w}_k$, with covariance $\mathbf{R}_k$, is the measurement noise for measurement k.

$$\begin{aligned} \mathbf{x}_k &= \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_{k-1} \\ \mathbf{z}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{w}_k \end{aligned} \tag{4.1}$$

$$\begin{aligned} \mathbf{v}_{k-1} &\sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1}), \quad \mathbf{E}\{\mathbf{v}_i\mathbf{v}_j^T\} &= \delta_{ij}\mathbf{Q}_i \\ \mathbf{w}_k &\sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k), \quad \mathbf{E}\{\mathbf{w}_i\mathbf{w}_j^T\} &= \delta_{ij}\mathbf{R}_i \end{aligned} \tag{4.2}$$

The moments of the predicted process distribution $p(\mathbf{x}_k|\mathbf{Z}^{k-1}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})$ is calculated as (4.3), according to the model previously described.

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \end{aligned} \tag{4.3}$$

The moments of the predicted measurement distribution $p(\mathbf{z}_k|\mathbf{Z}^{k-1}) = \mathcal{N}(\mathbf{z}_k; \hat{\mathbf{z}}_{k|k-1}, \mathbf{S}_{k|k-1})$ is calculated as (4.4).

$$\begin{aligned}\hat{\mathbf{z}}_{k|k-1} &= \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1} \\ \mathbf{S}_{k|k-1} &= \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k\end{aligned} \tag{4.4}$$

With this said we are ready to derive the filtering and smoothing equations under the assumptions of linear process- and measurement models.

4.1 The Kalman Filter

4.1.1 MLE derivation

To find the optimal filtered distribution $p(\mathbf{x}_k|\mathbf{Z}^k)$ at time k we adopt an *Maximum likelihood estimate* (MLE) approach. This means that we wish to find the estimate $\hat{\mathbf{x}}_{k|k}$ for which the filtered distribution $p(\mathbf{x}_k|\mathbf{Z}^k)$, derived as (3.1), is maximized. The MLE approach lets us utilize the model assumption that the process distributions in state space are Gaussian aswell as the process- and measurement noise. Since the maximum point of a positive function is the same as for the logarithm of the said function we may maximize $\mathcal{L}(\mathbf{x}_k, \mathbf{z}_k)$, the logarithm of (3.1), instead of maximizing the probability distribution function (3.1).

$$\mathcal{L}(\mathbf{x}_k, \mathbf{z}_k) = \log\left[\frac{p(\mathbf{x}_k,\mathbf{z}_k|\mathbf{Z}^{k-1})}{p(\mathbf{z}_k|\mathbf{Z}^{k-1})}\right] = \log\left[p(\mathbf{x}_k,\mathbf{z}_k|\mathbf{Z}^{k-1})\right] - \log\left[p(\mathbf{z}_k|\mathbf{Z}^{k-1})\right] \tag{4.5}$$

The numerator $p(\mathbf{x}_k, \mathbf{z}_k|\mathbf{Z}^{k-1})$ in (3.1) can be described by a product of two Gaussian distributions (4.6) according to (3.3).

$$p(\mathbf{x}_k, \mathbf{z}_k|\mathbf{Z}^{k-1}) \propto \mathcal{N}(\mathbf{z}_k; \mathbf{H}\mathbf{x}_k, \mathbf{R}_k) \times \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \tag{4.6}$$

Since the term $log\left[p(\mathbf{z}_k|\mathbf{Z}^{k-1})\right]$ in (4.5) does not contain $\mathbf{x}_k$, and therefore has a zero partial derivative in relation to $\mathbf{x}_k$, we may write (4.7).

$$\mathcal{L}(\mathbf{x}_k,\mathbf{z}_k) \propto ||\mathbf{z}_k - \mathbf{H}_k\mathbf{x}_k||^2\mathbf{R}_k^{-1} + ||\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}||^2\mathbf{P}_{k|k-1}^{-1} \tag{4.7}$$

We may then just differentiate (4.7) and set this expression to zero. This leaves us with the vector equation (4.8).

$$\nabla_{\mathbf{x}_k}\mathcal{L} = 2\mathbf{H}_k\mathbf{R}_k^{-1}(\mathbf{H}_k\mathbf{x}_k - \mathbf{z}_k) + 2\mathbf{P}_{k|k-1}^{-1}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) = \mathbf{0} \tag{4.8}$$

Solving this equations gives us the explicit expression (4.9) for the *MLE* $\hat{\mathbf{x}}_{k|k}$ of $\mathbf{x}_k$ given $\mathbf{Z}^k$.

$$\hat{\mathbf{x}}_{k|k} = (\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k + \mathbf{P}_{k|k-1}^{-1})(\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{z}_k + \mathbf{P}_{k|k-1}^{-1}\hat{\mathbf{x}}_{k|k-1}) \tag{4.9}$$

Using the matrix inversion lemma we may rewrite this as (4.10) where $\mathbf{K}_k$ is the kalman gain expressed in (4.11).

$$\hat{\mathbf{x}}_{k|k} \quad = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}) \tag{4.10}$$

$$\mathbf{K}_k \quad = \mathbf{P}_{k|k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{S}_{k|k-1}^{-1} \tag{4.11}$$

We now wish to derive the state space covariance matrix for the filtered distribution (3.1), defined by (4.12), which will describe the uncertainty of the process state estimate $\hat{\mathbf{x}}_{k|k}$.

$$\mathbf{P}_{k|k} = cov(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}) \tag{4.12}$$

Using definitions (4.11) gives us (4.13).

$$\mathbf{P}_{k|k} = cov(\mathbf{x}_k - (\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}))) \tag{4.13}$$

Using our measurement model from (4.1) we may write (4.13).

$$\begin{aligned}\mathbf{P}_{k|k} &= cov(\mathbf{x}_k - (\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}))) \\ &= cov((\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) - \mathbf{K}_k\mathbf{v}_k) \\ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)cov(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k cov(\mathbf{v}_k)\mathbf{K}_k^T \\ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T\end{aligned} \tag{4.14}$$

Equation(4.14) is the definition for the posterior state space covariance matrix $\mathbf{P}_{k|k}$.

4.1.2 MMSE derivation

Now we wish to show that the derived Kalman gain is also optimal in the *Minimum Mean Square Error* (MMSE) sence. Insted of assuming Gaussian distribution for the probability distributions in state space, we assume the derived updating equation (4.10). To find the MMSE estimator, we which to choose $\hat{\mathbf{x}}_{k|k}$ such that the square of the *Mean Square Error* (MSE), (4.15) is minimized.

$$MSE(\hat{\mathbf{x}}) = \sum_i \mathbf{E}\{(\mathbf{x}_i - \hat{\mathbf{x}}_i)^2\} = \sum_i \mathbf{P}_{ii} = Tr(\mathbf{P}) \tag{4.15}$$

We therefore wish to find $\mathbf{K}_k$ such that the trace of the posterior state space covariance matrix $\mathbf{P}_{k|k}$, defined as (4.14), equivalent to (4.16), is minimized.

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{K}_k^T + \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T \tag{4.16}$$

Differentiating the trace of (4.16) yields (4.17).

$$\frac{\partial Tr(\mathbf{P}_{k|k})}{\partial \mathbf{K}_k} = -2(\mathbf{H}_k\mathbf{P}_{k|k-1})^T + 2\mathbf{K}_k\mathbf{S}_k \tag{4.17}$$

This yields the optimal Kalman gain (4.18), which is the same as the *MLE* optimal gain (4.11).

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{S}_k^{-1} \tag{4.18}$$

For more details on the *Kalman Filter*, see [1, 2].

4.2 The Rauch-Tung-Striebel smoother

In order to solve the smoothing problem under the assumptions of linear models, we again adopt an MLE approach. We wish to maximize the distribution $p(\mathbf{x}_k|\mathbf{Z}^N)$,which we assume is Gaussian, with respect to $\mathbf{x}_k$. This is equivalent to the maximization of the joint smoothed distribution $p(\mathbf{x}_k, \mathbf{x}_{k+1}|\mathbf{Z}^N)$ with respect to $\mathbf{x}_k$. For the full expression of $p(\mathbf{x}_k, \mathbf{x}_{k+1}|\mathbf{Z}^N)$, see equation (3.9). This in turn is equivalent to maximizing the logarithm (4.19) of the strictly positive function $p(\mathbf{x}_k, \mathbf{x}_{k+1}|\mathbf{Z}^N)$, for $\mathbf{x}_{k+1} = \hat{\mathbf{x}}_{k+1|N}$.

$$\begin{aligned}\mathcal{L}(\mathbf{x}_k, \mathbf{x}_{k+1}) &= \log\left[p(\mathbf{x}_k, \mathbf{x}_{k+1}|\mathbf{Z}^N)\right]\\ &= \log\left[p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{Z}^k)\right] + \log\left[p(\mathbf{x}_k|\mathbf{Z}^k)\right] - \log\left[p(\mathbf{x}_{k+1}|\mathbf{Z}^k)\right] + \log\left[p(\mathbf{x}_{k+1}|\mathbf{Z}^N)\right]\end{aligned} \tag{4.19}$$

Since we are differentiating with respect to $\mathbf{x}_k$, the terms *log* $\left[p(\mathbf{x}_{k+1}|\mathbf{Z}^k)\right]$ and *log* $\left[p(\mathbf{x}_{k+1}|\mathbf{Z}^N)\right]$ in (4.19) can be ignored. The joint distribution (3.9) can therefore be seen as proportial to a product of two Gaussian distributions.

$$p(\mathbf{x}_k, \mathbf{x}_{k+1}|\mathbf{Z}^N) \propto \mathcal{N}(\mathbf{x}_{k+1}; \mathbf{F}_k\mathbf{x}_k, \mathbf{Q}_k) \times \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}) \tag{4.20}$$

This in turn means that the important aspects of $\mathcal{L}(\mathbf{x}_k, \mathbf{x}_{k+1})$ can be summarized as (4.21).

$$\mathcal{L}(\mathbf{x}_k, \hat{\mathbf{x}}_{k+1|N}) \propto ||\hat{\mathbf{x}}_{k+1|N} - \mathbf{F}_k\mathbf{x}_k||^2\mathbf{Q}_k^{-1} + ||\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}||^2\mathbf{P}_{k|k}^{-1} \tag{4.21}$$

We may then just differentiate (4.21) with respect to $\mathbf{x}_k$ and set the gradient (4.22) to zero to find the MLE.

$$\nabla_{\mathbf{x}_k}\mathcal{L} = 2\mathbf{F}_k\mathbf{Q}_k^{-1}(\mathbf{F}_k\mathbf{x}_k - \hat{\mathbf{x}}_{k+1|N}) + 2\mathbf{P}_{k|k}^{-1}(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}) = \mathbf{0} \tag{4.22}$$

Rearranging (4.22) for explicit expression of the optimal posterior MLE estimate $\hat{\mathbf{x}}_{k|N}$ of the state leads to (4.23).

$$\hat{\mathbf{x}}_{k|N} = (\mathbf{F}_k^T\mathbf{Q}_k^{-1}\mathbf{F}_k + \mathbf{P}_{k|k}^{-1})(\mathbf{F}_k^T\mathbf{Q}_k^{-1}\hat{\mathbf{x}}_{k+1|N} + \mathbf{P}_{k|k}^{-1}\hat{\mathbf{x}}_{k|k}) \tag{4.23}$$

Using the matrix inversion lemma we arrive at the *Rauch-Tung-Striebel* smoothing equation (4.24), where matrix (4.25) is the smoothing gain.

$$\hat{\mathbf{x}}_{k|N} = \hat{\mathbf{x}}_{k|k} + \mathbf{A}_k(\hat{\mathbf{x}}_{k+1|N} - \mathbf{F}_k\hat{\mathbf{x}}_{k|k}) \tag{4.24}$$

$$\mathbf{A}_k = \mathbf{P}_{k|k}\mathbf{F}_k^T(\mathbf{F}_k\mathbf{P}_{k|k}\mathbf{F}_k^T + \mathbf{Q}_k)^{-1} \tag{4.25}$$

The smoothed process state covariance matrix $\mathbf{P}_{k|N}$ is then calculated according to (4.26).

$$\begin{aligned}
\mathbf{P}_{k|N} \quad &= cov(\mathbf{x}_k - \hat{\mathbf{x}}_{k|N}) \\
&= cov(\mathbf{x}_k - (\hat{\mathbf{x}}_{k|k} + \mathbf{A}_k(\hat{\mathbf{x}}_{k+1|N} - \mathbf{F}_k\hat{\mathbf{x}}_{k|k}))) \\
&= \mathbf{P}_{k|k} + \mathbf{A}_k(\mathbf{P}_{k+1|N} - \mathbf{P}_{k|k+1})\mathbf{A}_k^T
\end{aligned} \tag{4.26}$$

For more details on the *Rauch-Tung-Striebel* smoother, see [2].

5

Approximate solution for nonlinear systems

There are a vast number of situations in engineering where filtering and smoothing are needed to extract information about an observed process. But for many of these situations, the derived linear solutions just won't do, since the process or the sensor is best described by a nonlinear model. In these cases the *Kalman Filter* and the *Rauch-Tung-Striebel* smoother are no longer applicable. The main problem is that nonlinear transformations are applied to Gaussian distributions; transformations from state space to measurement space, or through predictions in state space, from one point in time to another. These transformations are being made without difficulty for linear models, but for nonlinear models we must use approximations.

Since the beginning of the 60's, back when the first attempts were made with nonlinear kalman filters, the approximation of choice has been the *Extended Kalman Filter* (EKF). In the EKF, the nonlinear function is simply linearized around the estimate of the distribution mean. And all the sudden, the standard Kalman equations are applicable again. This seems convenient, but there is an issue with stability when it comes to extended kalman filters and smoothers [3].

The Oxford mathematician *Uhlmann* claims that *"it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function"*. Instead of approximating the nonlinear map as in *EKF* we may approximate the distribution by a number of discretized units called *sigma points*. The *sigma points* are each transformed using the nonlinear map and then reassembled to approximate the first two moments of the transformed distribution. This approach gives increased stability and is also capable of handling higher order moments. The Unscented transform could be used for coordinate transformation as well as for process- or measurement model transformations.

5.1 The Unscented Transform

This sampling scheme, called the *the Unscented Transform*, was proposed by *Julier* and *Uhlmann* [9]. The first step in this scheme is to construct an extended Gaussian distribution with mean $\tilde{\mathbf{x}}_{k-1}$ and covariance $\tilde{\mathbf{P}}_{k-1}$, by adding the process- and measurement noises as in (5.1). For information on the reduced dimensionality noise covariance matrix $\mathbf{\Sigma}_{k-1}$, see section 7.3.

$$\begin{aligned} \tilde{\mathbf{x}}_{k-1} &= \begin{pmatrix} \hat{\mathbf{x}}_{k-1|k-1} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \\ \tilde{\mathbf{P}}_{k-1} &= \begin{pmatrix} \mathbf{P}_{k-1|k-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_k \end{pmatrix} \end{aligned} \tag{5.1}$$

The sigma points are then calculated according to (5.2), where $(\sqrt{(n_{\mathbf{x}}+\kappa)\tilde{\mathbf{P}}_{k-1}})_i$ means column i of the matrix $\mathbf{L}$, for which $\mathbf{L}\mathbf{L}^T = (n_{\mathbf{x}}+\kappa)\tilde{\mathbf{P}}_{k-1}$. $n_{\mathbf{x}}$ is the dimensionality of the extended distribution (5.1), and κ is a constant parameter which gives dimensional-independent scaling for Gaussian distributions if set to $\kappa = 3 - n_{\mathbf{x}}$. For more information on κ, see [8].

$$\chi_{k-1}^{(i)} = \begin{cases} \tilde{\mathbf{x}}_{k-1}, & i = 0 \\ \tilde{\mathbf{x}}_{k-1} + \left(\sqrt{(n_{\mathbf{x}}+\kappa)\tilde{\mathbf{P}}_{k-1}}\right)_i, & i = 1,...,n_{\mathbf{x}} \\ \tilde{\mathbf{x}}_{k-1} - \left(\sqrt{(n_{\mathbf{x}}+\kappa)\tilde{\mathbf{P}}_{k-1}}\right)_{i-n_{\mathbf{x}}}, & i = (n_{\mathbf{x}}+1),...,2n_{\mathbf{x}} \end{cases} \tag{5.2}$$

Each sigma point also has an importance weight, defined in (5.3), so that the reassembled moments are defined as (5.4) and (5.5).

$$W^{(i)} = \begin{cases} \frac{\kappa}{n_{\mathbf{x}}+\kappa} i = 0 \\ \frac{1}{2(n_{\mathbf{x}}+\kappa)} i = 1,...,n_{\mathbf{x}} \\ \frac{1}{2(n_{\mathbf{x}}+\kappa)} i = (n_{\mathbf{x}}+1),...,2n_{\mathbf{x}} \end{cases} \tag{5.3}$$

$$\bar{\mathbf{x}} = \sum_i W^{(i)} \chi^{(i)} \tag{5.4}$$

$$\tilde{\mathbf{P}} = \sum_i W^{(i)} (\chi^{(i)} - \bar{\mathbf{x}})(\chi^{(i)} - \bar{\mathbf{x}})^T \tag{5.5}$$

The sigma points $\chi_{k-1}^{(i)}$ can be subdivided into parts. In our case one part representing the process state, one representing the process noise and one representing the measurement noise, see (5.6).

$$\chi_{k-1}^{(i)} = \begin{pmatrix} \chi_{k-1}^{(i),\mathbf{x}} \\ \chi_{k-1}^{(i),\Sigma} \\ \chi_{k-1}^{(i),R} \end{pmatrix} \tag{5.6}$$

There is no proof that the Unscented Transform is the optimal representation of a distribution given the two first moments, but arguments can be made that it is a good choice. For more details on the Unscented Transform see [8, 9].

5.2 The prediction step

We start by sampling the posteriori distribution $p(\mathbf{x}_{k-1}|\mathbf{Z}^{k-1})$, as described in section 5.1, giving sigma point-/weight pairs $\{\chi_{k-1}^{(i)}, W_{k-1}^{(i)}\}_{i=0}^{N-1}$. The prediction and transformation into measurement space are then carried out according to (5.7) and (5.8). The matrix $\mathbf{B}$ is projects the reduced dimensional noise term to state space, see secion 7.3.

$$\chi_{k|k-1}^{(i),\mathbf{x}} = f_{k-1}(\chi_{k-1}^{(i),\mathbf{x}}, \mathbf{B} \cdot \chi_{k-1}^{(i),\Sigma}) \tag{5.7}$$

$$\mathcal{Z}_{k|k-1}^{(i)} = h_k(\chi_{k|k-1}^{(i),\mathbf{x}}, \chi_{k-1}^{(i),R}) \tag{5.8}$$

The first two moments of the predicted distributions in state space is then calculated as (5.9), and the predicted distribution in measurement space is calculated as (5.10).

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \sum_{i=0}^{N-1} W_{k-1}^{(i)} \chi_{k|k-1}^{(i),\mathbf{x}} \\ \mathbf{P}_{k|k-1} &= \sum_{i=0}^{N-1} W_{k-1}^{(i)} (\chi_{k|k-1}^{(i),\mathbf{x}} - \hat{\mathbf{x}}_{k|k-1})(\chi_{k|k-1}^{(i),\mathbf{x}} - \hat{\mathbf{x}}_{k|k-1})^T \end{aligned} \tag{5.9}$$

$$\begin{aligned} \hat{\mathbf{z}}_{k|k-1} &= \sum_{i=0}^{N-1} W_{k-1}^{(i)} \mathcal{Z}_{k|k-1}^{(i)} \\ \mathbf{S}_{k|k-1} &= \sum_{i=0}^{N-1} W_{k-1}^{(i)} (\mathcal{Z}_{k|k-1}^{(i)} - \hat{\mathbf{z}}_{k|k-1})(\mathcal{Z}_{k|k-1}^{(i)} - \hat{\mathbf{z}}_{k|k-1})^T \end{aligned} \tag{5.10}$$

The cross correlation between predicted state and measurement, which we will find useful later, is defined as (5.11).

$$\mathbf{P}_{\mathbf{xz}} = \sum_{i=0}^{N-1} W_{k-1}^{(i)} (\chi_{k|k-1}^{(i),\mathbf{x}} - \hat{\mathbf{x}}_{k|k-1})(\mathcal{Z}_{k|k-1}^{(i)} - \hat{\mathbf{z}}_{k|k-1})^T \tag{5.11}$$

5.3 Unscented Kalman Gain

Since we already derived the form (4.10) for the updating equation in section 4.1, using the method of *Maximum Likelihood Estimation*, we shall assume the updating form (4.10) for kalman filtering. Without assuming Gaussian distributions we may then just use the method of *Minimum Mean Square Error* estimation to find the kalman gain for the unscented filter. The posterior covariance matrix, denoted $\mathbf{P}_{k|k}$, is derived in (5.12) using the updating equation (4.10).

$$\begin{aligned}
\mathbf{P}_{k|k} &= E\{(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^2\} \\
&= E\{(\mathbf{x}_k - (\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})))^2\} \\
&= E\{((\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) - \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}))^2\} \\
&= E\{(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^2 - 2\mathbf{K}_k(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}) + \mathbf{K}_k^2(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})^2\} \\
&= E\{(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^2\} - 2\mathbf{K}_k E\{(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})\} + \mathbf{K}_k^2 E\{(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})^2\} \\
&= \mathbf{P}_{k|k-1} - 2\mathbf{K}_k\mathbf{P}_{\mathbf{xz}} + \mathbf{K}_k^2\mathbf{S}_{k|k-1}
\end{aligned} \tag{5.12}$$

As in section 4.1.2 we minimize the trace of the posterior covariance matrix to find the optimal gain.

$$\frac{\partial Tr(\mathbf{P}_{k|k})}{\partial \mathbf{K}_k} = -2\mathbf{P}_{\mathbf{xz}} + 2\mathbf{K}_k\mathbf{S}_{k|k-1} = 0 \tag{5.13}$$

The optimal Unscented Kalman Gain is thus (5.14).

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{xz}}\mathbf{S}_{k|k-1}^{-1} \tag{5.14}$$

5.4 Unscented Smoothing Gain

The conceptual solution (3.9) to the smoothing problem can be written as (5.15).

$$p(\mathbf{x}_k, \mathbf{x}_{k+1}|\mathbf{Z}^N) = p(\mathbf{x}_k|\mathbf{x}_{k+1}, \mathbf{Z}^N)p(\mathbf{x}_{k+1}|\mathbf{Z}^N) = \frac{p(\mathbf{x}_k, \mathbf{x}_{k+1}|\mathbf{Z}^k)}{p(\mathbf{x}_{k+1}|\mathbf{Z}^k)}p(\mathbf{x}_{k+1}|\mathbf{Z}^N) \tag{5.15}$$

We start by describing the joint probability distribution $p(\mathbf{x}_k, \mathbf{x}_{k+1}|\mathbf{Z}^k)$.

$$p(\mathbf{x}_k, \mathbf{x}_{k+1}|\mathbf{Z}^k) = \mathcal{N}\left(\begin{pmatrix} \mathbf{x}_k \\ \mathbf{x}_{k+1} \end{pmatrix}; \begin{pmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{x}}_{k+1|k} \end{pmatrix}, \begin{pmatrix} \mathbf{P}_{k|k} & \mathbf{C}_{k+1} \\ \mathbf{C}_{k+1}^T & \mathbf{P}_{k+1|k} \end{pmatrix}\right) \tag{5.16}$$

$\mathbf{C}_{k+1}$ in equation (5.16) is the cross correlation as described in (5.17).

$$\mathbf{C}_{k+1} = E\{(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})(\mathbf{x}_k - \hat{\mathbf{x}}_{k+1|k})^T\} = \sum_{i=0}^{N-1} W_k^{(i)} (\chi_k^{(i),\mathbf{x}} - \hat{\mathbf{x}}_{k|k})(\chi_{k+1|k}^{(i),\mathbf{x}} - \hat{\mathbf{x}}_{k+1|k})^T \tag{5.17}$$

To compute $p(\mathbf{x}_k|\mathbf{x}_{k+1}, \mathbf{Z}^N)$, we divide (5.16) by $p(\mathbf{x}_{k+1}|\mathbf{Z}^k)$, according to (5.15), which yields (5.18).

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{x}_{k+1}, \mathbf{Z}^N) &= \mathcal{N}\left(\mathbf{x}_k; \mathbf{m}^{'}_{k+1}, \mathbf{P}^{'}_{k+1}\right) \\ \mathbf{A}_k &= \mathbf{C}_{k+1}\mathbf{P}^{-1}_{k+1|k} \\ \mathbf{m}^{'}_{k+1} &= \hat{\mathbf{x}}_{k|k} + \mathbf{A}_k(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k}) \\ \mathbf{P}^{'}_{k+1} &= \mathbf{P}_{k|k} - \mathbf{A}_k\mathbf{P}_{k+1|k}\mathbf{A}_k^T \end{aligned} \tag{5.18}$$

The joint distribution $p(\mathbf{x}_k, \mathbf{x}_{k+1}|\mathbf{Z}^N)$ can then be expressed as (5.19).

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{x}_{k+1}|\mathbf{Z}^N) &= \mathcal{N}\left(\begin{pmatrix} \mathbf{x}_k \\ \mathbf{x}_{k+1} \end{pmatrix}; \mathbf{m}^{''}_{k+1}, \mathbf{P}^{''}_{k+1}\right) \\ \mathbf{m}^{''}_{k+1} &= \begin{pmatrix} \hat{\mathbf{x}}_{k|k} + \mathbf{A}_k(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k}) \\ \hat{\mathbf{x}}_{k+1|N} \end{pmatrix} \\ \mathbf{P}^{''}_{k+1} &= \begin{pmatrix} \mathbf{A}_k\mathbf{P}_{k+1|N}\mathbf{A}_k^T + \mathbf{P}^{'}_{k+1} & \mathbf{A}_k\mathbf{P}_{k+1|N} \\ \mathbf{P}_{k+1|N}\mathbf{A}_k^T & \mathbf{P}_{k+1|N} \end{pmatrix} \end{aligned} \tag{5.19}$$

Marginalizing over $\mathbf{x}_{k+1}$ then gives the unscented smoothing equations (5.20). Note that these are equivalent to the *Rauch-Tung-Striebel* smoothing equations (4.24) and (4.26), but the smoothing gain $\mathbf{A}_k$ is defined according to (5.18).

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{Z}^N) &= \mathcal{N}\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|N}, \mathbf{P}_{k|N}\right) \\ \hat{\mathbf{x}}_{k|N} &= \hat{\mathbf{x}}_{k|k} + \mathbf{A}_k(\hat{\mathbf{x}}_{x+1|N} - \hat{\mathbf{x}}_{k+1|k}) \\ \mathbf{P}_{k|N} &= \mathbf{P}_{k|k} + \mathbf{A}_k(\mathbf{P}_{k+1|N} - \mathbf{P}_{k+1|k})\mathbf{A}_k^T \end{aligned} \tag{5.20}$$

For more details on the Uncsented Rauch-Tung-Striebel smoother, see [3].

6

Solution for multiple models

In some situations one filter is not sufficient to describe the process or measurements with good enough accuracy. One might want to apply one process model for a maneuvering aircraft and another when it is on straight course, for example. There might also be scenarios when one would like to apply different measurement models. One solution to this problem is the hybrid solution (6.1), visualized in figure 6.1. We simply make the assumption that in each interval $t_{k-1} < t \leq t_k$ the process is in a discrete mode m_k. Each mode represents one way to describe the process we observe and also the measurements we make. For each mode we thus have one process model and one measurement model, so that we can construct one filter and one smoother for each mode.

$$\begin{aligned} \mathbf{x}_k &= f_{k-1}(\mathbf{x}_{k-1}, m_k) + \mathbf{v}_k(m_k) \\ \mathbf{z}_k &= h_k(\mathbf{x}_k, m_k) + \mathbf{w}_k(m_k) \end{aligned} \tag{6.1}$$

For transition between modes, we assume a first order markov system (6.2).

$$P\{m_k^j | m_{k-1}^i\} = \pi_{ji} \tag{6.2}$$

This system will become a decision tree with M^k terminal nodes for M modes over k measurement intervals. We define path ℓ through this tree as $\mathcal{M}_k^\ell$. The probability for each such path can be derived as (6.3).

$$\begin{aligned} P\{\mathcal{M}_k^\ell | \mathbf{Z}^k\} &\propto P\{\mathbf{z}_k | \mathcal{M}_k^\ell, \mathbf{Z}^{k-1}\} P\{\mathcal{M}_k^\ell | \mathbf{Z}^{k-1}\} \\ &= P\{\mathbf{z}_k | \mathcal{M}_k^\ell, \mathbf{Z}^{k-1}\} P\{m_k^j | \mathcal{M}_{k-1}^\ell, \mathbf{Z}^{k-1}\} P\{\mathcal{M}_{k-1}^\ell | \mathbf{Z}^{k-1}\} \end{aligned} \tag{6.3}$$

Under the markovian assumption, (6.2), we may rewrite the second factor in (6.3) as (6.4).

$$P\{m_k^j|\mathcal{M}_{k-1}^{\ell},\mathbf{Z}^{k-1}\} = P\{m_k^j|m_{k-1}^i,\mathbf{Z}^{k-1}\} = P\{m_k^j|m_{k-1}^i\} = \pi_{ji} \tag{6.4}$$

This model is refered to as a *Jump Markov System* (JMS). The problem with this model is that the decision tree has exponentially growing number of terminal nodes. To handle this exponential complexity, we may want to use a suboptimal solution to this problem. One obvious solution is to prune the tree, which means removing paths with low likelihood. Another way to go is to merge the terminal nodes. The latter is the foundation for the *Interacting Multiple Model* (IMM). In Section 6.1 we will discuss filtering with an IMM and in Section 6.2 we will discuss how we can use the same framework for smoothing.

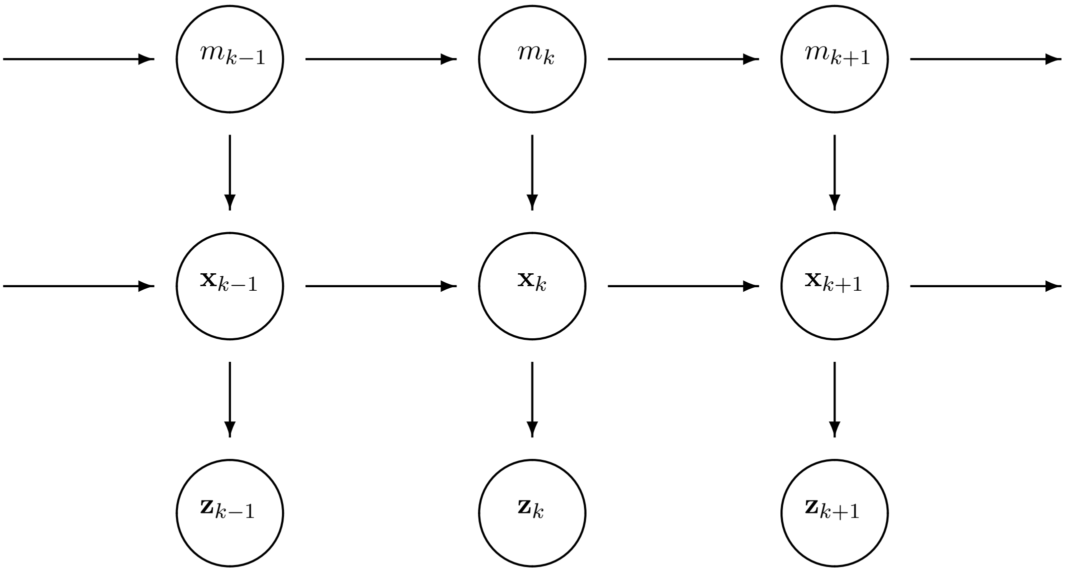


Figure 6.1: Visual schematic for the assumed model, if we disregard the possibility of multiple measurement models. The state $\mathbf{x}$ is a markov process described by the process mode m. The switching between modes is a discrete markov process. The measurement $\mathbf{z}_k$ is assumed to be dependent only on the process state $\mathbf{x}_k$ at the same instant of time.

6.1 IMM Filtering

6.1.1 Conceptual solution

In the IMM filter the posterior pdf is described by the mode probabilites, $P\{m_k^j|\mathbf{Z}^k\}$, and a Gaussian pdf, $p(\mathbf{x}_k|m_k^j,\mathbf{Z}^k)$ associated with each mode at each instant of time. This Gaussian mixture is described in equation (6.5).

$$p(\mathbf{x}_k|\mathbf{Z}^k) = \sum_j p(\mathbf{x}_k|m_k^j,\mathbf{Z}^k)P\{m_k^j|\mathbf{Z}^k\} \tag{6.5}$$

As in the conceptual solution, Section 3.1, the mode-conditioned posterior density can be divided into a prediction step and an updating step, using Bayes' formula, see (6.6).

$$p(\mathbf{x}_k|m_k^j,\mathbf{Z}^k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k, m_k^j, \mathbf{Z}^{k-1})p(\mathbf{x}_k|m_k^j,\mathbf{Z}^{k-1})}{p(\mathbf{z}_k|m_k^j,\mathbf{Z}^{k-1})} \tag{6.6}$$

The predicted pdf at time k for model m_k^j is a marginalization over the process variable $\mathbf{x}_{k-1}$, (6.7).

$$p(\mathbf{x}_k|m_k^j, \mathbf{Z}^{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, m_k^j,\mathbf{Z}^{k-1})p(\mathbf{x}_{k-1}|m_k^j, \mathbf{Z}^{k-1})d\mathbf{x}_{k-1} \tag{6.7}$$

We know the pdf $p(\mathbf{x}_{k-1}|m_{k-1}^i, \mathbf{Z}^{k-1})$, but to calculate $p(\mathbf{x}_{k-1}|m_k^j, \mathbf{Z}^{k-1})$ in (6.7) we must mix posterior densities from the prior time step, as shown in (6.8).

$$p(\mathbf{x}_{k-1}|m_k^j, \mathbf{Z}^{k-1}) = \sum_i p(\mathbf{x}_{k-1}|m_{k-1}^i, m_k^j, \mathbf{Z}^{k-1})P\{m_{k-1}^i|m_k^j, \mathbf{Z}^{k-1}\} \tag{6.8}$$

The mixing probability $P\{m_{k-1}^i|m_k^j, \mathbf{Z}^{k-1}\}$ can be derived using Bayes' rule and the markovian assumption (6.9).

$$P\{m_{k-1}^i|m_k^j, \mathbf{Z}^{k-1}\} = \frac{P\{m_k^j|m_{k-1}^i, \mathbf{Z}^{k-1}\}P\{m_{k-1}^i|\mathbf{Z}^{k-1}\}}{P\{m_k^j|\mathbf{Z}^{k-1}\}} \tag{6.9}$$

Where the predicted mode probabilities $P\{m_k^j|\mathbf{Z}^{k-1}\}$ can be decribed as the sum (6.10).

$$P\{m_k^j|\mathbf{Z}^{k-1}\} = \sum_i P\{m_k^j|m_{k-1}^i, \mathbf{Z}^{k-1}\}P\{m_{k-1}^i|\mathbf{Z}^{k-1}\} \tag{6.10}$$

The posterior mode probabilities are derived using Bayes' rule (6.11).

$$P\{m_k^j|\mathbf{Z}^k\} = P\{m_k^j|\mathbf{z}_k, \mathbf{Z}^{k-1}\} \propto p(\mathbf{z}_k|m_k^j, \mathbf{Z}^{k-1})P\{m_k^j|\mathbf{Z}^{k-1}\} \tag{6.11}$$

The factor $p(\mathbf{z}_k|m_k^j, \mathbf{Z}^{k-1})$ is simply the likelihood of measurement $\mathbf{z}_k$ given mode m_k^j which is derived by equation (6.12).

$$p(\mathbf{z}_k|m_k^j, \mathbf{Z}^{k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k, m_k^j, \mathbf{Z}^{k-1})p(\mathbf{x}_k|m_k^j, \mathbf{Z}^{k-1})d\mathbf{x}_k \tag{6.12}$$

6.1.2 Algorithm

The implementation of the algorithm goes as follows: The two moments of the gaussian mixture defined in (6.5) is calculated by (6.13) and (6.14). $w_{k|k}^i$ is the posterior mode probability for mode i at time k, defined in (6.11).

$$\hat{\mathbf{x}}_{k|k} = \sum_{j=1}^{M} w_{k|k}^j \hat{\mathbf{x}}_{k|k}^j \tag{6.13}$$

$$\mathbf{P}_{k|k} = \sum_{j=1}^{M} w^j_{k|k} \left[\mathbf{P}^j_{k|k} + (\hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}^j_{k|k})(\hat{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}^j_{k|k})^T \right] \tag{6.14}$$

The mixing probabilities $\mu^{i|j}_{k-1}$ defined in (6.9) is calculated by (6.15), where $w^j_{k|k-1}$ is the predicted mode probabilities defined in (6.10).

$$\mu^{i|j}_{k-1} = \frac{\pi_{ji} w^i_{k-1|k-1}}{w^j_{k|k-1}} \tag{6.15}$$

$$w^j_{k|k-1} = \sum_{\ell=1}^{M} \pi_{j\ell} w^\ell_{k-1|k-1} \tag{6.16}$$

The two moments of the matched gaussian (6.8) is calculated by expressions (6.17) and (6.18). This is the data that then goes into filter j as the posterior data from time $k-1$, to predict according to (6.7).

$$\hat{\mathbf{x}}^{(j)}_{k-1|k-1} = \sum_{i=1}^{M} \mu^{i|j}_{k-1} \hat{\mathbf{x}}^i_{k-1|k-1} \tag{6.17}$$

$$\mathbf{P}^{(j)}_{k-1|k-1} = \sum_{i=1}^{M} \mu^{i|j}_{k-1} \left[\mathbf{P}^i_{k-1|k-1} + (\hat{\mathbf{x}}^i_{k-1|k-1} - \hat{\mathbf{x}}^{(j)}_{k-1|k-1})(\hat{\mathbf{x}}^i_{k-1|k-1} - \hat{\mathbf{x}}^{(j)}_{k-1|k-1})^T \right] \tag{6.18}$$

The posterior mode probabilities defined in (6.11) is calculated according to (6.19), where Λ^j_k is the likelihood for measurement k according to prediction made by filter j, calculated by expression 6.20.

$$w^j_{k|k} = \frac{\Lambda^j_k w^j_{k|k-1}}{\sum_{i=1}^{M} \Lambda^i_k w^i_{k|k-1}} \tag{6.19}$$

$$\Lambda^j_k = p(\mathbf{z}_k | m^j_k, \mathbf{Z}^{k-1}) \tag{6.20}$$

For more details on IMM filtering see [4, 7].

6.2 IMM Smoothing

6.2.1 Conceptual solution

We wish to find $p(\mathbf{x}_k|\mathbf{Z}^N)$, the probability distribution of $\mathbf{x}_k$ predicated on measurements $\mathbf{Z}^N$. We naturally begin with describing $p(\mathbf{x}_k|\mathbf{Z}^N)$ as a Gaussian mixture (6.21).

$$p(\mathbf{x}_k|\mathbf{Z}^N) = \sum_j p(\mathbf{x}_k|m^j_k, \mathbf{Z}^N) P\{m^j_k|\mathbf{Z}^N\} \tag{6.21}$$

The factor $p(\mathbf{x}_k|m_k^j, \mathbf{Z}^N)$ is unknown, but we may marginalize over $\mathbf{x}_{k+1}$ according to (6.22).

$$p(\mathbf{x}_k|m_k^j,\mathbf{Z}^N) = \int p(\mathbf{x}_k,\mathbf{x}_{k+1}|m_k^j,\mathbf{Z}^N)d\mathbf{x}_{k+1} \tag{6.22}$$

The joint smoothed distribution in (6.22) can be expressed as (6.23).

$$p(\mathbf{x}_k, \mathbf{x}_{k+1}|m_k^j,\mathbf{Z}^N) = p(\mathbf{x}_k|\mathbf{x}_{k+1},m_k^j,\mathbf{Z}^N)p(\mathbf{x}_{k+1}|m_k^j,\mathbf{Z}^N) \tag{6.23}$$

Measurements $\{\mathbf{z}_i\}_{i=k+1}^N$ in $P\{\mathbf{x}_k|\mathbf{x}_{k+1}, m_{k+1}^i, \mathbf{Z}^N\}$ are redundant since we assume that we know the full state $\{\mathbf{x}_{k+1}, m_{k+1}^i\}$ at time k+1. This insight plus Bayes' formula gives us (6.24).

$$p(\mathbf{x}_k|\mathbf{x}_{k+1},m_k^j,\mathbf{Z}^N) = p(\mathbf{x}_k|\mathbf{x}_{k+1},m_k^j,\mathbf{Z}^k) = \frac{p(\mathbf{x}_{k+1}|\mathbf{x}_k,m_k^j,\mathbf{Z}^k)p(\mathbf{x}_k|m_k^j,\mathbf{Z}^k)}{p(\mathbf{x}_{k+1}|m_k^j,\mathbf{Z}^k)} \tag{6.24}$$

Using (6.23) and (6.24) we may write out the joint smoothed distribution as (6.25).

$$p(\mathbf{x}_k, \mathbf{x}_{k+1}|m_k^j,\mathbf{Z}^N) = \frac{p(\mathbf{x}_{k+1}|\mathbf{x}_k,m_k^j,\mathbf{Z}^k)p(\mathbf{x}_k|m_k^j,\mathbf{Z}^k)}{p(\mathbf{x}_{k+1}|m_k^j,\mathbf{Z}^k)}p(\mathbf{x}_{k+1}|m_k^j,\mathbf{Z}^N) \tag{6.25}$$

This is essentially the same equation as (3.9). One big difference though is that in the smoothed distribution at time $k+1$, $p(\mathbf{x}_{k+1}|m_k^j,\mathbf{Z}^N)$, $\mathbf{x}_{k+1}$ is also conditioned on m_k^j. This leads us to the backward moment matching described in (6.26).

$$p(\mathbf{x}_{k+1}|m_k^j,\mathbf{Z}^N) = \sum_i p(\mathbf{x}_{k+1}|m_{k+1}^i, m_k^j, \mathbf{Z}^N)P\{m_{k+1}^i|m_k^j, \mathbf{Z}^N\} \tag{6.26}$$

$P\{m_{k+1}^i|m_k^j, \mathbf{Z}^N\}$ is not known, and is therefore manipulated using Bayes' formula, according to (6.27).

$$P\{m_{k+1}^i|m_k^j, \mathbf{Z}^N\} = \frac{P\{m_k^j|m_{k+1}^i, \mathbf{Z}^N\}P\{m_{k+1}^i|\mathbf{Z}^N\}}{P\{m_k^j|\mathbf{Z}^N\}} \tag{6.27}$$

Measurements $\{\mathbf{z}_i\}_{i=k+1}^N$ in $P\{m_k^j|m_{k+1}^i, \mathbf{Z}^N\}$ are redundant since we assume that we know m_{k+1}^i. This insight plus Bayes' formula leads to the expression (6.28).

$$P\{m_k^j|m_{k+1}^i, \mathbf{Z}^N\} = P\{m_k^j|m_{k+1}^i, \mathbf{Z}^k\} = \frac{P\{m_{k+1}^i|m_k^j, \mathbf{Z}^k\}P\{m_k^j|\mathbf{Z}^k\}}{P\{m_{k+1}^i|\mathbf{Z}^k\}} \tag{6.28}$$

To derive an expression for the smoothed mode probability $P\{m_k^j|\mathbf{Z}^N\}$ we may represent the knowledge gained by measurements $\{\mathbf{z}_i\}_{i=k+1}^N$ by the smoothed Gaussian mixture $\mathcal{M}_{k+1|N}$, consisting of the mode conditioned distributions $p(\mathbf{x}_{k+1}|m_{k+1}^i, \mathbf{Z}^N)$, and the mode probabilites $p(m^i_{k+1}|\mathbf{Z}^N)$. Due to this approximation and Bayes' formula we get (6.29).

$$P\{m^j_k|\mathbf{Z}^N\} \triangleq P\{m^j_k|\mathcal{M}_{k+1|N},\mathbf{Z}^k\} = \frac{P\{\mathcal{M}_{k+1|N}|m^j_k,\mathbf{Z}^k\}P\{m^j_k|\mathbf{Z}^k\}}{P\{\mathcal{M}_{k+1|N}|\mathbf{Z}^k\}} \tag{6.29}$$

$P\{m^i_{k+1}|m^j_k,\mathbf{Z}^k\} = P\{m^i_{k+1}|m^j_k\}$ according to the first order markov model. We may also note that the Gaussian mixture $\mathcal{M}_k$ is reduced to the, single mode conditioned Gaussian $\mathcal{M}^j_k$, if we know the model m^j_k at time k. This leads to equation (6.30).

$$\begin{aligned} P\{\mathcal{M}_{k+1|N}|m^j_k,\mathbf{Z}^k\} &= \sum_i P\{\mathcal{M}_{k+1|N}|m^i_{k+1},m^j_k,\mathbf{Z}^k\}P\{m^i_{k+1}|m^j_k,\mathbf{Z}^k\} \\ &\triangleq \sum_i P\{\mathcal{M}_{k+1|N}|m^i_{k+1},m^j_k,\mathcal{M}_{k|k}\}P\{m^i_{k+1}|m^j_k\} \\ &= \sum_i P\{\mathcal{M}^i_{k+1|N}|\mathcal{M}^j_{k|k}\}P\{m^i_{k+1}|m^j_k\} \end{aligned} \tag{6.30}$$

6.2.2 Algorithm

The implementation of the algorithm goes as follows: Calculate $b_{ji} = P\{m^j_k|m^i_{k+1},\mathbf{Z}^N\}$ according to (6.31) as described in (6.28).

$$b_{ji} = \frac{\pi_{ij}\mu^j_{k|k}}{\sum_\ell \pi_{i\ell}\mu^\ell_{k|k}} \tag{6.31}$$

Then calculate the mixing probabilites $\mu^{i|j}_{k+1|N} = P\{m^i_{k+1}|m^j_k,\mathbf{Z}^N\}$ according to (6.32), following (6.27).

$$\mu^{i|j}_{k+1|N} = \frac{b_{ji}w^i_{k+1|N}}{\sum_\ell b_{j\ell}w^\ell_{k+1|N}} \tag{6.32}$$

The matched distribution $p(\mathbf{x}_{k+1}|m^j_k,\mathbf{Z}^N)$ for timestep $k+1$ is calculated according to (6.26), and is thus described by its two moments (6.33).

$$\begin{aligned} \hat{\mathbf{x}}^{(j)}_{k+1|N} &= \sum_i \mu^{i|j}_{k+1|N}\hat{\mathbf{x}}^i_{k+1|N} \\ \mathbf{P}^{(j)}_{k+1|N} &= \sum_i \mu^{i|j}_{k+1|N}\left[\mathbf{P}^i_{k+1|N} + \left(\hat{\mathbf{x}}^i_{k+1|N} - \hat{\mathbf{x}}^{(j)}_{k+1|N}\right)\left(\hat{\mathbf{x}}^i_{k+1|N} - \hat{\mathbf{x}}^{(j)}_{k+1|N}\right)^T\right] \end{aligned} \tag{6.33}$$

With the matched distribution we may then perform the smoothing step 6.34 for all smoothers to arrive at distributions $p(\mathbf{x}_k|m^j_k,\mathbf{Z}^N)$ [1].

[1] See section 4.2 for a derivation

$$
\begin{aligned}
\hat{\mathbf{x}}^j_{k|N} &= \hat{\mathbf{x}}^j_{k|k} + \mathbf{A}^j_k \left(\hat{\mathbf{x}}^{(j)}_{k+1|N} - \hat{\mathbf{x}}^j_{k+1|k} \right) \\
\mathbf{P}^j_{k|N} &= \mathbf{P}^j_{k|k} + \mathbf{A}^j_k \left(\mathbf{P}^{(j)}_{k+1|N} - \mathbf{P}^j_{k+1|k} \right) \mathbf{A}^{j\,T}_k
\end{aligned}
\tag{6.34}
$$

The smoothed mode probabilites $w^j_{k|N} = p(m^j_k|\mathbf{Z}^N)$ are calculated according to (6.29), following (6.35).

$$
w^j_{k|N} = \frac{\Lambda^j_{k|N} w^j_{t|t}}{\sum_\ell \Lambda^\ell_{k|N} w^\ell_{t|t}}
\tag{6.35}
$$

The likelihood $\Lambda^j_{k|N}$ for model m^j_k in (6.35) is calculated as (6.36) which is the same as (6.30).

$$
\Lambda^j_{k|N} = \sum_i \pi_{ji} \mathcal{N} \left(\hat{\mathbf{x}}^i_{k+1|N}; \hat{\mathbf{x}}^j_{k+1|k}, \mathbf{P}^j_{k+1|k} \right)
\tag{6.36}
$$

Finally the smoothed distribution (6.21) is described by its two moments (6.37).

$$
\begin{aligned}
\hat{\mathbf{x}}_{k|N} &= \sum_j w^j_{k|N} \hat{\mathbf{x}}^j_{k|N} \\
\mathbf{P}_{k|N} &= \sum_i w^j_{k|N} \left[\mathbf{P}^j_{k|N} + \left(\hat{\mathbf{x}}^j_{k|N} - \hat{\mathbf{x}}_{k|N} \right) \left(\hat{\mathbf{x}}^j_{k|N} - \hat{\mathbf{x}}_{k|N} \right)^T \right]
\end{aligned}
\tag{6.37}
$$

For more details on IMM smoothing, see [4].

7

Process models

PROCESS MODELS are easiest formulated as differential equations with a stochastic component $\mathbf{v}_d(t)$. The general form is described in (7.1), and the linear model equivalent is (7.2). In both cases, the process noise is uncorrelated over time, and is Gaussian distributed around zero with process noise covariance $\mathbf{Q}_d(t)$ as described in (7.3).

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{v}_d(t) \tag{7.1}$$

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{v}_d(t) \tag{7.2}$$

$$\mathbf{v}_d(t) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}_d(t)\right) \tag{7.3}$$

But in the filter equations that we have encountered in previous sections, the process is described as a difference equation predicting the process state vector at time $t_{k+1} = t_k + T$ given the process state vector at time t_k. In the following section 7.1 we will reformulate the linear process differential equation (7.2) to a difference equation such that it is applicable for the filter equations in chapter 4. In the subsequent section 7.2 we then do the same for the general case, so that the result is even applicable for the nonlinear filtering equations in chapter 5.

7.1 Discretization of linear differential equations

Since we assume that we know $\mathbf{x}$ at time t_k, then $\mathbf{x}(t_k)$ is our initial condition and we wish to solve equation (7.2) for time variable $t_k + \tau$ over the interval $\tau \in [0, T]$ to arrive at an explicit expression for $\mathbf{x}_{k+1} = \mathbf{x}(t_k + T)$.

$$\dot{\mathbf{x}}(t_k + \tau) = \mathbf{A}\mathbf{x}(t_k + \tau) + \mathbf{B}\mathbf{u}(t_k + \tau) + \mathbf{v}_d(t_k + \tau) \tag{7.4}$$

We then multiply $e^{-\mathbf{A}\tau}$ into (7.4) which, as we will see, will make things easier.

$$e^{-\mathbf{A}\tau}\dot{\mathbf{x}}(t_k+\tau) = e^{-\mathbf{A}\tau}\mathbf{A}\mathbf{x}(t_k+\tau) + e^{-\mathbf{A}\tau}\mathbf{B}\mathbf{u}(t_k+\tau) + e^{-\mathbf{A}\tau}\mathbf{v}_d(t_k+\tau) \tag{7.5}$$

We then subtract $e^{-\mathbf{A}\tau}\mathbf{A}\mathbf{x}(t_k+\tau)$ from (7.5) and use the fact that:

$$\frac{d}{d\tau}\left(e^{-\mathbf{A}\tau}\mathbf{x}(t_k+\tau)\right) = e^{-\mathbf{A}\tau}\dot{\mathbf{x}}(t_k+\tau) - e^{-\mathbf{A}\tau}\mathbf{A}\mathbf{x}(t_k+\tau)$$

, and we get (7.6).

$$\frac{d}{d\tau}(e^{-\mathbf{A}\tau}\mathbf{x}(t_k+\tau)) = e^{-\mathbf{A}\tau}\mathbf{B}\mathbf{u}(t_k+\tau) + e^{-\mathbf{A}\tau}\mathbf{v}_d(t_k+\tau) \tag{7.6}$$

To get $\mathbf{x}(t_k+T)$ we then integrate over the interval $\tau \in [0,T]$.

$$\int_0^T \frac{d}{d\tau}(e^{-\mathbf{A}\tau}\mathbf{x}(t_k+\tau))d\tau = \int_0^T e^{-\mathbf{A}\tau}\mathbf{B}\mathbf{u}(t_k+\tau)d\tau + \int_0^T e^{-\mathbf{A}\tau}\mathbf{v}_d(t_k+\tau)d\tau \tag{7.7}$$

$$e^{-\mathbf{A}T}\mathbf{x}(t_k+T) - \mathbf{x}(t_k) = \int_0^T e^{-\mathbf{A}\tau}\mathbf{B}\mathbf{u}(t_k+\tau)d\tau + \int_0^T e^{-\mathbf{A}\tau}\mathbf{v}_d(t_k+\tau)d\tau \tag{7.8}$$

We now have an explicit expression (7.9) for $x(t_k+T)$, albeit with some rather complicated terms still.

$$x(t_k+T) = e^{\mathbf{A}T}\mathbf{x}(t_k) + \int_0^T e^{\mathbf{A}(T-\tau)}\mathbf{B}\mathbf{u}(t_k+\tau)d\tau + \int_0^T e^{\mathbf{A}(T-\tau)}\mathbf{v}_d(t_k+\tau)d\tau \tag{7.9}$$

We make the approximation that the control input $\mathbf{u}(t_k)$ and the noise term $\mathbf{v}(t_k)$ are constant over the interval. We may do this if the interval is short enough and if they are fairly constant over the time interval. For convenience, we also make the change of variables $\nu = T-\tau$, after which may write (7.10)

$$x(t_k+T) \approx e^{\mathbf{A}T}\mathbf{x}(t_k) + \int_0^T e^{\mathbf{A}\nu}d\nu\mathbf{B}\mathbf{u}(t_k) + \int_0^T e^{\mathbf{A}\nu}d\nu\mathbf{v}(t_k) \tag{7.10}$$

We then simply use the definition for the exponential function and the assumption that the matrix $\mathbf{A}$ is nilpotent, which means that $\mathbf{A}^q = \mathbf{0}$ for some positive integer q. We may then write (7.11) and thus arrive at the solution (7.12).

$$e^{\mathbf{A}T} = \sum_{i=0}^{q-1}\frac{A^i}{i!}T^i \tag{7.11}$$

$$\mathbf{x}(t_k+T) = \mathbf{F}(T)\mathbf{x}(t_k) + \mathbf{G}(T)\mathbf{B}\mathbf{u}(t_k) + \mathbf{v}(t_k,T) \tag{7.12}$$

$$\begin{aligned} \mathbf{F}(T) &= \textstyle\sum_{i=0}^{q-1} \frac{A^i}{(i+1)!}T^{i+1} \\ \mathbf{G}(T) &= \textstyle\sum_{i=0}^{q-1} \frac{A^i}{i!}T^i \end{aligned} \tag{7.13}$$

As can be seen in equation (7.12), the discrete process noise term $\mathbf{v}(t_k,T)$ has a dependence on both t and T, as described in (7.14).

$$\mathbf{v}(t_k,T) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}(t_k,T)\right) \tag{7.14}$$

The discrete process noise covariance matrix $\mathbf{Q}(t_k,T)$ is derived in (7.15).

$$\begin{aligned} \mathbf{Q}(t_k,T) &= E\{[\mathbf{x}(t_k+T) - \hat{\mathbf{x}}(t_k+T)]\,[\mathbf{x}(t_k+T) - \hat{\mathbf{x}}(t_k+T)]^T\} \\ &= E\{\left[\int_0^T e^{\mathbf{A}\nu} d\nu \mathbf{v}_d(t_k)\right]\left[\int_0^T e^{\mathbf{A}\nu} d\nu \mathbf{v}_d(t_k)\right]^T\} \\ &= \int_0^T e^{\mathbf{A}\nu} d\nu E\{\mathbf{v}_d(t_k)\mathbf{v}_d(t_k)^T\} \int_0^T e^{\mathbf{A}^T\nu} d\nu \end{aligned} \tag{7.15}$$

Thus $\mathbf{Q}(t_k,T)$ is calculated according to (7.16), where $\mathbf{Q}_d(t_k)$ is defined in (7.3).

$$\mathbf{Q}(t_k,T) = \mathbf{G}(T)\mathbf{Q}_d(t_k)\mathbf{G}(T)^T \tag{7.16}$$

7.2 Discretized linearization

To make use of the derivations in section 7.1 we want to rewrite (7.1) so that it looks something like (7.2). What we do is to use the second order taylor expansion for $\mathbf{f}(\mathbf{x})$ at $\hat{\mathbf{x}}(t_k)$ and rearranging the terms.

$$\begin{aligned} \dot{\mathbf{x}} &\approx \mathbf{f}(\hat{\mathbf{x}}(t)) + \nabla\mathbf{f}(\hat{\mathbf{x}}(t))(\mathbf{x}(t) - \hat{\mathbf{x}}(t)) + \mathbf{v}_d(t) \\ &= \nabla\mathbf{f}(\hat{\mathbf{x}}(t))\mathbf{x}(t) + (\mathbf{f}(\hat{\mathbf{x}}(t)) - \nabla\mathbf{f}(\hat{\mathbf{x}}(t))\hat{\mathbf{x}}(t)) + \mathbf{v}_d(t) \end{aligned} \tag{7.17}$$

According to the derivation in 7.1, we may express the estimate $\hat{\mathbf{x}}(t_k+T)$ as (7.18).

$$\hat{\mathbf{x}}(t_k+T) \approx \mathbf{F}(\hat{\mathbf{x}}(t_k),T)\hat{\mathbf{x}}(t_k) + \mathbf{G}\left(\hat{\mathbf{x}}(t_k),T\right)(\mathbf{f}(\hat{\mathbf{x}}(t_k)) - \nabla\mathbf{f}(\hat{\mathbf{x}}(t_k))\,\hat{\mathbf{x}}(t_k)) \tag{7.18}$$

$$\mathbf{F}(\hat{\mathbf{x}}(t_k),T) = e^{\nabla\mathbf{f}(\hat{\mathbf{x}}(t_k))T} \tag{7.19}$$

$$\mathbf{G}(\hat{\mathbf{x}}(t_k),T) = \int_0^T e^{\nabla\mathbf{f}(\hat{\mathbf{x}}(t_k))\nu} d\nu \tag{7.20}$$

$\mathbf{G}(\hat{\mathbf{x}}(t_k),T)$ may then be expressed as (7.21).

$$\mathbf{G}(\hat{\mathbf{x}}(t_k), T) = e^{\nabla \mathbf{f}(\hat{\mathbf{x}}(t))\nu} \nabla \mathbf{f}(\hat{\mathbf{x}}(t))^{-1} \Big|_0^T = e^{\nabla \mathbf{f}(\hat{\mathbf{x}}(t))T} \nabla \mathbf{f}(\hat{\mathbf{x}}(t))^{-1} - \nabla \mathbf{f}(\hat{\mathbf{x}}(t))^{-1} \tag{7.21}$$

Using some algebra in (7.22) we then arrive at the simpler expression (7.23).

$$\begin{aligned} \hat{\mathbf{x}}(t_k + T) \approx\ & e^{\nabla \mathbf{f}(\hat{\mathbf{x}}(t_k))T} \hat{\mathbf{x}}(t_k) - e^{\nabla \mathbf{f}(\hat{\mathbf{x}}(t_k))T} \hat{\mathbf{x}}(t_k) + \hat{\mathbf{x}}(t_k) + \\ & + \left(e^{\nabla \mathbf{f}(\hat{\mathbf{x}}(t_k))T} \nabla \mathbf{f}(\hat{\mathbf{x}}(t_k))^{-1} - \nabla \mathbf{f}(\hat{\mathbf{x}}(t_k))^{-1} \right) \mathbf{f}(\hat{\mathbf{x}}(t)) \\ & = \hat{\mathbf{x}}(t_k) + \left(e^{\nabla \mathbf{f}(\hat{\mathbf{x}}(t_k))T} \nabla \mathbf{f}(\hat{\mathbf{x}}(t_k))^{-1} - \nabla \mathbf{f}(\hat{\mathbf{x}}(t_k))^{-1} \right) \mathbf{f}(\hat{\mathbf{x}}(t_k)) \end{aligned} \tag{7.22}$$

$$\hat{\mathbf{x}}(t_k + T) \approx \hat{\mathbf{x}}(t) + \int_0^T e^{\nabla \mathbf{f}(\hat{\mathbf{x}}(t))T} d\nu \mathbf{f}(\hat{\mathbf{x}}(t)) \tag{7.23}$$

We then drop the approximation notation and state (7.24) where $\mathbf{G}(\hat{\mathbf{x}}(t_k), T)$ is described as the sum (7.25), if the jacobian $\nabla \mathbf{f}(\hat{\mathbf{x}}(t_k))$ is nilpotent.

$$\mathbf{x}(t_k + T) = \mathbf{x}(t) + \mathbf{G}(\hat{\mathbf{x}}(t_k), T)\mathbf{f}(\hat{\mathbf{x}}(t_k)) + \mathbf{v}(t_k) \tag{7.24}$$

$$\mathbf{G}(\hat{\mathbf{x}}(t_k), T) = \sum_{i=0}^{q-1} \left(\nabla \mathbf{f}(\hat{\mathbf{x}}(t_k))\right)^i \frac{T^{i+1}}{(i+1)!} \tag{7.25}$$

The predicted mean value at time $t_{k+1} = t_k + T$ can thus be expressed as (7.26).

$$\hat{\mathbf{x}}(t_k + T) = \hat{\mathbf{x}}(t_k) + \left(\sum_{i=0}^{q-1} \left(\nabla \mathbf{f}(\hat{\mathbf{x}}(t_k))\right)^i \frac{T^{i+1}}{(i+1)!} \right) \mathbf{f}(\hat{\mathbf{x}}(t_k)) \tag{7.26}$$

The noise term is basically no different from (7.14), except that we have to add the dependence on $\hat{\mathbf{x}}(t_k)$ due to possible nonlinearity.

$$\mathbf{v}(\hat{\mathbf{x}}(t_k), t_k, T) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}(\hat{\mathbf{x}}(t_k), t_k, T)\right) \tag{7.27}$$

$$\mathbf{Q}(\hat{\mathbf{x}}(t_k), t_k, T) = \mathbf{G}(\hat{\mathbf{x}}(t_k), T)\mathbf{Q}_d(t_k)\mathbf{G}(\hat{\mathbf{x}}(t_k), T)^T \tag{7.28}$$

See [5] for more details on state space model discretization.

7.3 Degrees of freedom for process noise

Usually, the noise term $\mathbf{v}_d(t_k)$ is only non-zero for the highest order variables in the process state. Thus, we can describe the noise with the lowerdimensional stochastic variable $\mathbf{s}_d(t_k)$ described by (7.29).

$$\mathbf{s}_d(t_k) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Sigma}(t_k)\right) \tag{7.29}$$

To project the lowerdimensional noise into the state space, we simply use a projection matrix **B**, and get the relations (7.30) and (7.31).

$$\mathbf{v}_d(t_k) = \mathbf{B} \cdot \mathbf{s}_{\mathbf{d}}(t_k) \tag{7.30}$$

$$\mathbf{Q}_d(t_k) = \mathbf{B}\boldsymbol{\Sigma}(t_k)\mathbf{B}^T \tag{7.31}$$

8

Evaluation

The aim of this study is to be able to estimate the state of the process that we measure with as good accuracy as possible. Since tracking for air surveillence radar is the application of this study, we will use aircraft trajectories as evaluation data. More precisely, we use six benchmark scenarios among which the classes large commercial aircraft, small agile commercial aircraft, medium bomber and fighter jet are represented. Measurements are then simlulated by Monte Carlo sampling from a given measurement model, which is also used in all the filters and smoothers. The assumption being that we know the statistical properties of the sensor noise. To eliminate dependence on the direction of the aircraft, and radial its distance in relation to the radar sensor, which is an issue for radial measurements, we simply use a cartesian measurement model with gaussian distributed noise. More about the benchmark scenarios and the measurement model is given in section 8.1.

We define the absolute error, $e_k^{(i)}$ at time step k for monte carlo simulation i, out of N simulations, as (8.1). The variable $\mathbf{x}_k$ in this case meaning either position, velocity or acceleration in cartesian coordinates for the aircraft at the time for measurement k, and $\hat{\mathbf{x}}_t^{(i)}$ being the tracker estimate at the same instant of time.

$$e_k^{(i)} = |\hat{\mathbf{x}}_k^{(i)} - \mathbf{x}_k| \tag{8.1}$$

The measure that we will focus on in the evaluation part is the *Root Mean Square Error* (RMSE), at the time for measurement k, defined as (8.2).

$$RMSE(\hat{\mathbf{x}}_k) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(e_k^{(i)})^2} \tag{8.2}$$

8.1 Evaluation data

The data set used for evaluation is synthetic and was propsed by Blair and Watson [10]. The data constitutes the six benchmark scenarios 8.1(a)-8.1(f) emulating aircraft maneuvers, each scenario representing a specific kind of aircraft. Scenario one represents the class large commercial aircraft, scenario two represents agile commercial aircraft, scenario three and four represents medium bomber and scenario five and six fighter jet. The acceleration in tangential and normal directions for all six scenarios are graphically represented in figures 8.2(a)-8.2(f). The benchmark scenarios are sampled with sampling interval:

$$T = t_k - t_{k-1} = 1.0(s)$$

which generates the discrete process states:

$$\mathbf{X}^N = \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_k, ..., \mathbf{x}_N$$

The measurements

$$\mathbf{Z}^N = \mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k, ..., \mathbf{z}_N$$

are then generated by monte carlo simulation using a cartesian measurement model with

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

and gaussian noise with covariance

$$\mathbf{R} = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix},$$

where $\sigma_x = \sigma_y = 100(m)$. The trackers are evaluated over 100 monte carlo simulations for each benchmark scenario.

8.2 Process model selection

The mathematical deitals for the evaluated process models are presented in Appendix A. In words, the process models are summarized:

CA Linear model, assuming constant acceleration and gaussian distributed noise in $[J_x, J_y]^T = [\dddot{x}, \dddot{y}]^T$. Advantagous for tracking of accelerating aircraft.

CV Linear model, assuming constant velocity and gaussian distributed noise in $[\ddot{x}, \ddot{y}]^T$. Advantagous for tracking of aircraft on straight course.

CAL Nonlinear model, assuming constant acceleration and gaussian distributed noise in $[J_t, J_n]^T = [\dot{a}_t, \dot{a}_n]^T$. Advantagous for tracking of aircraft that is accelerating in tangential and/or normal direction.

CT Nonlinear model, assuming constant tangential velocity, constant angular velocity and gaussian distributed noise in $\left[a_t, \dot{\Omega}\right]^T = \left[\dot{v}, \ddot{\phi}\right]^T$. Advantagous for tracking of aircraft during constant turn.

To evaluate the advantage of combining process models using the IMM, we have selected three model combinations for evaluation:

CV+CV+CA One CV model, with small noise covariance, describes the motion during straight course. Another CV model, with larger noise covariance, describes the motion during moderate turns. The CA model describes the motion during swift turns and accelerations.

CV+CV+CAL One CV model, with small noise covariance, describes the motion during straight course. Another CV model, with larger noise covariance, describes the motion during moderate turns. The CAL model describes the motion during swift turns and tangential acceleration.

CV+CT+CAL The CV model describes the motion during straight course. The CT model describes the motion during moderate turns. The CA model describes the motion during swift turns and tangential accelerations.

8.3 Process parameter estimation

One problem that we face while comparing the different process models is parameter setting. All the trackers each have one or a set of process models, and possibly even a markov switching model. Depending on the kind of tracker, there are a number of parameters to be set, typically process noise and markov switching probabilities. The problem is thus to compare one tracker with another in an objective manner without exploring the entire parameter space. The solution used here is to optimize the paramter setting for each tracker with respect to the *mean of the root mean square error in acceleration for the smoothed estimates over all six benchmark scenarios.* In order to search the parameter space for an optimal setting, we use evolutionary optimization with a binary gene representation, mutations, crossover, tournament selection and elitism. The fitness being the invers of the acceleration mean of the root mean square error for the smoothed acceleration estimates.
For more information on evolutionary optimization, see [11].

8.4 Evaluated trackers

Each tracker have one, or a set of filters. The filters are either of the type *KF*, if the filter process model is linear, or *UKF*, if the filter process model is nonlinear. The corresponding smoothers are of the type *RTS* for linear process models and *URTS* for nonlinear models. All the filters contain the same measurement model, the same one which was used for monte carlo simulation, see Section 8.1. The evaluated trackers are described below. See appendix A for information on the evaluated process models.

CV *KF/RTS* - *constant velocity* process model.

$$\bar{\sigma}_{\ddot{x}} = \bar{\sigma}_{\ddot{y}} = 9.985(m/s^2)$$

CA *KF/RTS* - *constant acceleration* process model.

$$\bar{\sigma}_{J_x} = \bar{\sigma}_{J_y} = 4.15(m/s^3)$$

CAL *UKF/URTS* - *constant acceleration local coordinates* process model.

$$\bar{\sigma}_{J_t} = 2.83(m/s^3)$$

$$\bar{\sigma}_{J_n} = 5.16(m/s^3)$$

CV+CV+CA *IMM* filter/smoother:

1. *KF/RTS* - *constant velocity* process model.

$$\bar{\sigma}_{\ddot{x}} = \bar{\sigma}_{\ddot{y}} = 0.873(m/s^2)$$

2. *KF/RTS* - *constant velocity* process model.

$$\bar{\sigma}_{\ddot{x}} = \bar{\sigma}_{\ddot{y}} = 5.37(m/s^2)$$

3. *KF/RTS* - *constant acceleration* process model.

$$\bar{\sigma}_{J_x} = \bar{\sigma}_{J_y} = 13.39(m/s^3)$$

Mode switching probabilities:

$$\pi = \begin{pmatrix} 0.94850 & 0.07515 & 0.00010 \\ 0.01124 & 0.89514 & 0.00205 \\ 0.04026 & 0.02971 & 0.99785 \end{pmatrix}$$

CV+CV+CAL *IMM* filter/smoother:

1. *KF/RTS* - *constant velocity* process model.

$$\bar{\sigma}_{\ddot{x}} = \bar{\sigma}_{\ddot{y}} = 0.718(m/s^2)$$

2. *KF/RTS* - *constant velocity* process model.

$$\bar{\sigma}_{\ddot{x}} = \bar{\sigma}_{\ddot{y}} = 4.975(m/s^2)$$

3. *UKF/URTS* - *constant acceleration local coordinates* process model.

$$\bar{\sigma}_{J_t} = 7.000(m/s^3)$$

$$\bar{\sigma}_{J_n} = 9.981(m/s^3)$$

Mode switching probabilities:

$$\pi = \begin{pmatrix} 0.98690 & 0.00186 & 0.00108 \\ 0.01153 & 0.97430 & 0.00059 \\ 0.00157 & 0.02385 & 0.99833 \end{pmatrix}$$

CV+CT+CAL *IMM* filter/smoother:

1. *KF/RTS* - *constant velocity* process model.

$$\bar{\sigma}_{\ddot{x}} = \bar{\sigma}_{\ddot{y}} = 2.143(m/s^2)$$

2. *UKF/URTS* - *coordinated turn* process model.

$$\bar{\sigma}_{a_t} = 6.35016(m/s^2)$$

$$\bar{\sigma}_{\dot{\Omega}} = 0.059(rad/s^2)$$

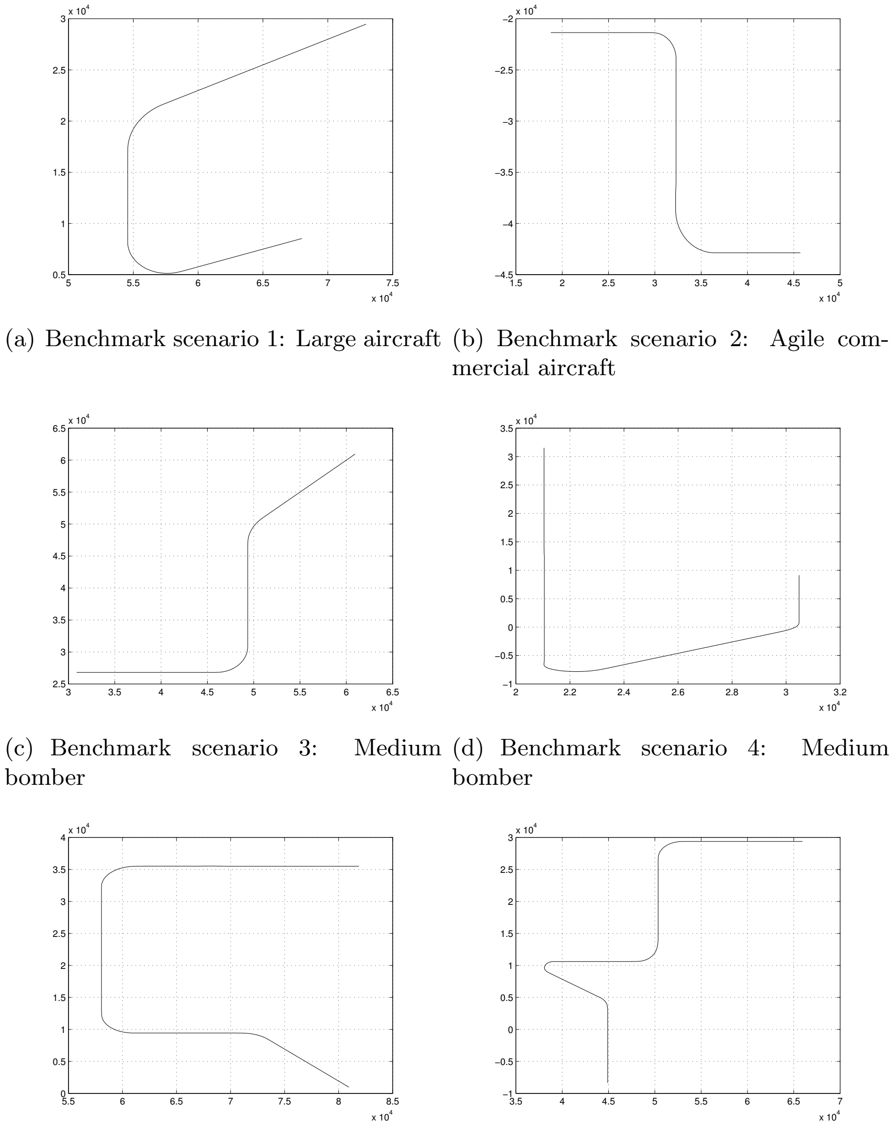
3. *UKF/URTS* - *constant acceleration local coordinates* process model.

$$\bar{\sigma}_{J_t} = 5.619(m/s^3)$$

$$\bar{\sigma}_{J_n} = 8.906(m/s^3)$$

Mode switching probabilities:

$$\pi = \begin{pmatrix} 0.91713 & 0.00224 & 0.04357 \\ 0.00568 & 0.98129 & 0.02163 \\ 0.07717 & 0.01654 & 0.93477 \end{pmatrix}$$



(a) Benchmark scenario 1: Large aircraft

(b) Benchmark scenario 2: Agile commercial aircraft

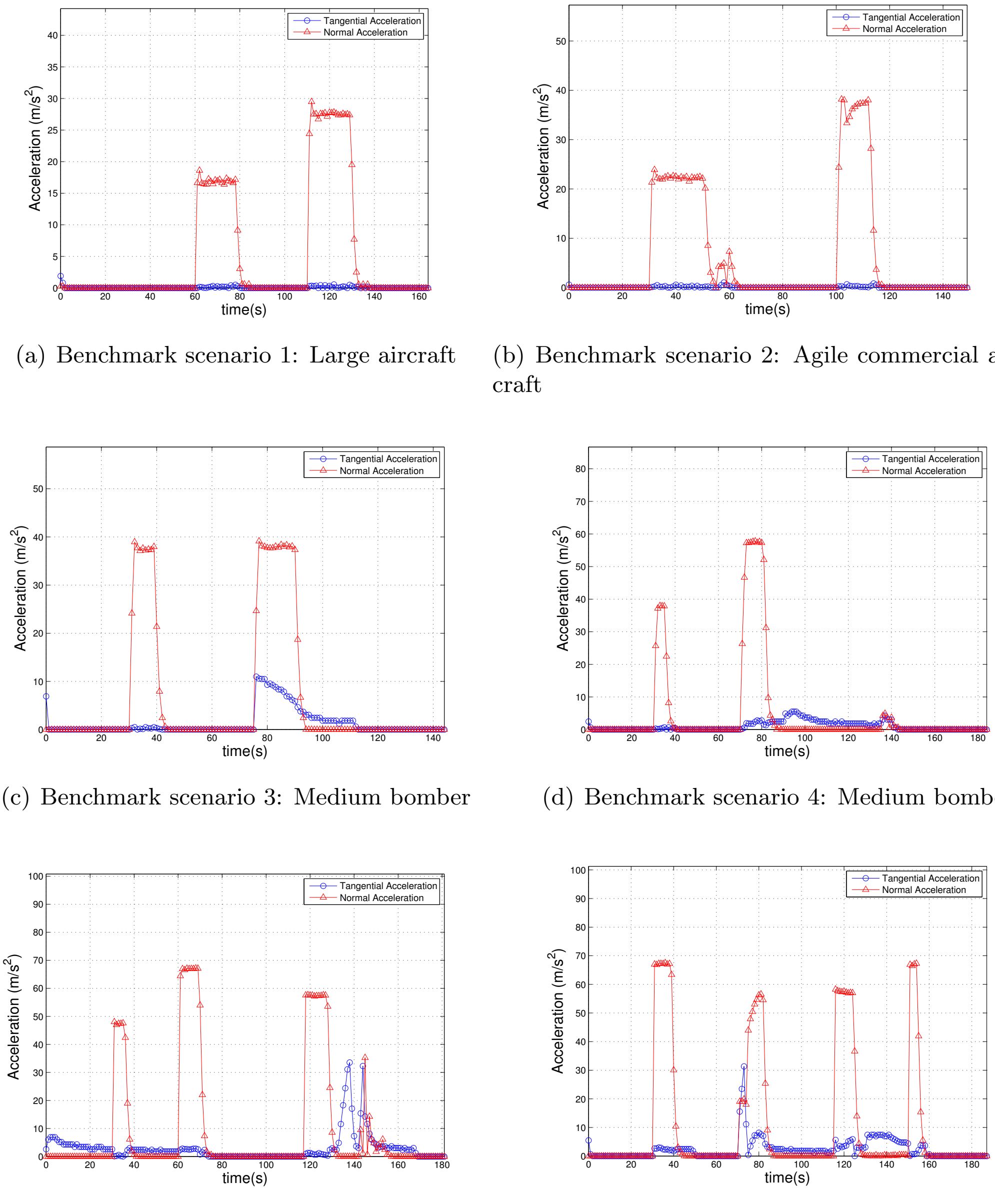
(c) Benchmark scenario 3: Medium bomber

(d) Benchmark scenario 4: Medium bomber

(e) Benchmark scenario 5: Fighter jet

(f) Benchmark scenario 6: Fighter jet

Figure 8.1: Benchmark scenarios



(a) Benchmark scenario 1: Large aircraft

(b) Benchmark scenario 2: Agile commercial aircraft

(c) Benchmark scenario 3: Medium bomber

(d) Benchmark scenario 4: Medium bomber

(e) Benchmark scenario 5: Fighter jet

(f) Benchmark scenario 6: Fighter jet

Figure 8.2: Acceleration for benchmark scenarios

9

Results

THE presented results constitutes of three tables 9.1, 9.2 and 9.3, and figure 9.1, as well as the figures in appendix B. Tables 9.1, 9.2 and 9.3 contain the mean and standard deviation for the position, velocity and acceleration root mean square error for all the evaluated filters and smoothers over each benchmark scenario separately and aggregated. Figure 9.1 contains relative computational time for the evaluated algorithms. Appendix B contains root mean square error cumulative frequency distributions for various combinations of filters and smoothers. Before constructing the frequency distributions, all measurement times t_k for the six benchmark scenarios are divided into the three sets:

Straight course $t_\ell \in \{t_k : a_n(t_k) < 0.2g\}$

Turn $t_\ell \in \{t_k : a_n(t_k) > 0.2g\}$

Sharp turn $t_\ell \in \{t_k : a_n(t_k) > 5g\}$

The $RMSE(\hat{\mathbf{x}}_\ell)$ cumulative frequency distribution for the filters and smoothers can thus be studied separately for those three cases.

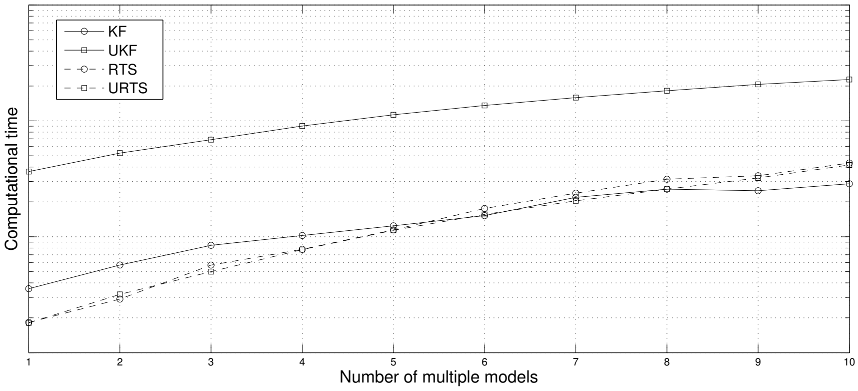


Figure 9.1: Computational time for multiple model filtering and smoothing for linear models (KF/RTS) and nonlinear models (UKF/URTS). Note: The time axis is logarithmic, and computational time for coordinate transformations in the nonlinear case is included. The computational time for a single UKF is a factor ≈ 10 longer than for KF.

	Benchmark scenario												Total	
	1		2		3		4		5		6			
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Filters														
CV	105.2	49.45	114	60.92	118.3	74.85	105.2	70.37	140.3	108.6	145.5	106.8	121.4	81.57
CA	100.3	18.83	102.7	21.95	104.9	20.16	104.3	24.82	115.8	31.62	116.8	29.62	107.5	24.95
CAL	97.28	16.48	99.79	21.33	102.4	21.94	103.3	24.44	113.5	30.5	114.4	28.97	105.1	24.41
CV+CV+CA	84.45	28.95	85.99	30.73	85.24	29.72	85.81	28.46	97.41	28.55	96.39	30.81	89.22	29.55
CV+CV+CAL	79.87	32.52	83.23	32.77	84.17	34.4	84.77	29.66	97.61	32.46	95.78	32.67	87.57	32.44
CV+CT+CAL	84.24	23.54	85.15	24.13	85.72	25.18	84.41	22.28	93.82	23.98	92.97	25.11	87.72	24.06
Smoothers														
CV	50.57	23.81	55.12	25.46	59.3	28.38	58.84	36.35	75.79	45.11	79.74	45.88	63.23	35.31
CA	49.42	21.96	52.06	24.35	54.2	22.06	56.58	27.93	65.43	29.06	69.68	32.31	57.9	26.55
CAL	48.27	19.82	50.17	24.15	52.53	23.79	54.62	26.44	64.41	28.31	66.48	28.42	56.08	25.33
CV+CV+CA	50.32	12.03	52.06	12.77	50.34	12.64	50.8	13.28	54.4	13.83	58.03	12.05	52.66	12.78
CV+CV+CAL	43.49	10.46	45.79	9.678	46.78	10.49	46.27	12.32	53.18	15.41	55.35	12.12	48.48	11.9
CV+CT+CAL	45.17	8.702	45.32	8.755	46.27	8.982	49.17	11.78	54.72	15.15	53.02	11.85	48.95	11.12

Table 9.1: RMSE mean and standard deviation for position estimation

	Benchmark scenario												Total	
	1		2		3		4		5		6			
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Filters														
CV	33.74	31.86	38.45	39.1	40.62	47.51	33	45.24	56.48	68.29	59.37	70.32	43.61	52.37
CA	39.18	9.476	42.6	15.46	45.96	19.77	45.37	26.82	60.2	37.24	63.52	37.88	49.47	26.65
CAL	36.31	7.347	39.39	12.97	43.21	17.19	43.87	23.63	58.27	34.88	61.84	34.9	47.15	24.19
CV+CV+CA	29.27	21.84	31.32	23.62	30.82	25.4	30.09	27.8	44.77	32.23	45.92	35.86	35.37	28.22
CV+CV+CAL	24.94	21.56	27.45	23.79	28.22	26.65	27.9	27.41	44.05	34.04	42.28	35.76	32.47	28.66
CV+CT+CAL	27.74	14.81	29.55	17.39	29.92	19.93	28.88	21.47	41.84	25.72	41.96	28.6	33.32	21.83
Smoothers														
CV	10.15	4.699	11.83	6.646	13.04	8.022	14.04	12.8	19.84	15.39	22.23	17.72	15.19	11.87
CA	10.49	4.435	11.71	5.627	12.72	6.334	14.23	10.85	18.47	11.92	21.31	14.61	14.82	9.702
CAL	10.12	4.225	11.03	4.714	12.12	5.729	13.14	8.962	17.89	11.56	18.97	12.67	13.88	8.635
CV+CV+CA	7.643	5.208	8.409	5.476	7.799	5.765	7.63	6.52	12.38	8.351	12.49	9.583	9.392	7.005
CV+CV+CAL	6.276	4.851	6.64	5.015	6.778	5.436	7.317	6.649	12.37	9.038	11.79	9.501	8.529	7.005
CV+CT+CAL	8.004	4.336	8.561	4.728	8.658	5.673	8.52	6.702	12.75	7.958	12.77	9.114	9.877	6.642

Table 9.2: RMSE mean and standard deviation for velocity estimation

	Benchmark scenario												Total	
	1		2		3		4		5		6			
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Filters														
CV	5.564	9.193	6.782	11.23	7.141	13.41	5.715	13.47	11.95	19.91	12.48	21.53	8.271	15.45
CA	9.016	5.194	10.58	7.691	11.93	9.658	11.65	12.66	18.23	17.51	19.83	18.6	13.54	12.86
CAL	8.133	4.424	9.579	6.967	11.07	8.937	11.36	11.66	17.51	16.83	19.67	17.47	12.89	12.06
CV+CV+CA	7.552	7.661	8.702	9.295	9.019	10.86	9.084	13.77	15.7	17.15	16.98	19.41	11.17	13.68
CV+CV+CAL	6.131	7.118	7.204	8.758	7.729	10.74	7.698	12.49	14.64	17.12	14.75	18.42	9.693	13.11
CV+CT+CAL	8.042	6.932	9.147	8.254	9.779	9.875	8.475	11.5	15.12	15.48	16	17.05	11.09	12.08
Smoothers														
CV	3.431	2.032	4.163	3.054	4.652	3.717	5.263	5.931	8.184	7.933	9.502	9.435	5.866	5.978
CA	3.471	1.893	4.086	2.704	4.551	3.348	5.318	5.503	7.941	7.375	9.51	8.819	5.813	5.546
CAL	3.472	1.902	3.958	2.439	4.444	3.095	4.9	4.609	7.789	7.107	8.645	7.889	5.535	5.052
CV+CV+CA	2.547	3.057	2.959	3.342	2.694	3.377	2.449	3.311	5.364	6.052	5.749	6.991	3.627	4.626
CV+CV+CAL	2.188	2.714	2.553	3.175	2.484	3.25	2.671	3.81	5.706	6.578	5.923	7.32	3.587	4.82
CV+CT+CAL	2.913	2.836	3.5	3.539	3.452	3.975	3.446	4.786	6.653	6.865	6.931	7.817	4.482	5.284

Table 9.3: RMSE mean and standard deviation for acceleration estimation

10

Conclusion

In section 10.2, ”Performance and parameter settings”, we will discuss the parameter setting that resulted from the evolutionary optimization and also the results from evaluation. Relevant data being the figures in Appendix B and the tables in Chapter 9. The focus will be to determine how the use of different process models, linear and nonlinear, separate and combined, affect the process state estimation error. In section 10.3, ”Future work”, we will discuss what additional work that may be done. First, though, we examine figure 9.1 and discuss the computational cost for the evaluated algorithms.

10.1 Computational complexity

From figure 9.1 we can conclude that the computational time for URTS and RTS are almost equivalent. Since they are essentially the same algorithm, once the smoothing gain has been calculated, this result is expected. In the uscented filter we construct, in this case, 21 sigma points. We then perform a coordinate transformation on these sigma points before filtering, and then transform the result back to linear coordinates. The additional factor ≈ 10 in computational time, compared to a linear filter, therefor seems reasoneble. The computational time for multiple filters is obviously linear if we disregard the IMM calculations, which supposedly is dominated by a component that has a quadratic complexity. This also corresponds to the results that we can observe. What we also may notice is that this component is more dominant in the smoothing step, which for fewer multiple models takes less time than the filtering step, but increases its computational time relative the filtering step as the number of multiple models grow larger.

10.2 Performance and parameter settings

Comparing the separate CV and CA filters/smoothers there is not one right answer to the question whether the process is best described by the assumption of constant velocity or constant acceleration. It seems that the CA process model is superior to CV in the case of position estimation, with a lower RMSE mean and standard deviation for all benchmark scenarios ,for both filtered and smoothed position estimates, see table 9.1 and figures B.1(a)-B.1(c). In the case of filtered velocity estimates, CV generally has a lower RMSE mean but a higher standard deviation. As the figures show, this is due to the fact that CV is clearly better designed for straight course, but cannot handle turns very well. For smoothed estimates, though, it seems as CA again has a lower RMSE mean and standard deviation than CV, see table 9.1 and figures B.2(a)-B.2(c). For the filtered acceleration estimates, CV again yields a lower RMSE mean but a slightly higher standard deviation than CA. For filtered acceleration estimates the RMSE mean and standard deviations for the two process models are almost equal, see table 9.3 and figures B.3(a)-B.3(c). In conclusion, we may state that the CA process model benefits from smoothing in a greater extent than does CV.

Looking at the separate CA and CAL filters/smoothers we can see the following:

$$\sqrt{\bar{\sigma}_{J_t}^2 + \bar{\sigma}_{J_n}^2} = 5.8823, \quad \sqrt{\bar{\sigma}_{J_x}^2 + \bar{\sigma}_{J_y}^2} = 5.8648$$

The optimal total acceleration variance is essentially the same for the two process models, but the nonlinear model, since it has separate variances for the tangential and normal directions may adapt itself better. This is confirmed by tables 9.1, 9.2 and 9.3 where the mean RMSE is lower for CAL than it is for CA on all the benchmark scenarios for both filtering and smoothing. In figures B.4(a)-B.6(c) it seems that the two process models yield roughly equally good results for straight course, but for turns the CAL model is slightly but consistently better for both filtering and smoothing.

Comparing the separate CV and CA process models with the markov system process model $CV + CV + CA$, one may first note that the optimal process parameters differ. In the markov system process model, the CV process noise standard deviation is lower, and thus handles straight course better. The CA process noise standard deviation is higher and thus handles turns better than does the separate CA model. As expected each model is allowed to be optmized for its respective purpose when they are combined in a markov system. Looking at the results in Tables 9.1, 9.2 and 9.3, we also see that this adaptability almost consistently yields lower RMSE mean for both filtering and smoothing. The only exception being filtered acceleration estimates where separate CV process model still yields the lowest RMSE mean. For filtered estimates, the separate CAL process model still consistently yields the lowest RMSE standard deviation. For smoothed estimates, though, the markov system process model is superior to the separate process models in close to every aspect, see figures B.13(a)-B.18(c).

Comparing the optimal parameters for the combined process models $CV + CV + CAL$ and $CV + CV + CA$, we see that the use of the CAL process model lower the process noise optimal standard deviation for the CV process models. The adaptability of the CAL model allows the CV models to improve straight course performance further without affecting performance during turns. Trading one of the CV process models for a CT model as in the $CV + CT + CAL$ markov system process model, the single CV model left will have to play an intermediate role compared to the dual CV models in the $CV + CV + CAL$ model. This results in higher RMSE on straight course, but the additional CT model seems to lower the RMSE during maneuvers, especially for filtered position- and velocity estimates, and may thus be seen as a complement to the CAL model, see figures B.13(a)- B.15(c). The results 9.1, 9.2 and 9.3 consistently show that $CV + CV + CAL$ yields the lowest RMSE mean while $CV + CT + CAL$ yields the lowest RMSE standard deviation, except for smoothed acceleration estimates.

In conclusion we may state that the use of multiple process models yields good results, especially in the case of smoothing. Also the use of nonlinear models generally have a positive effect on state estimation accuracy, if the models are choosen wisely. One additional conclusion we may draw from these results is that smoothing is necessary for acceptable accuracy in state estimation for purposes such as object classification on kinematic basis, at least for maneuvering aircrafts under the studied circumstances.

The comparison between filtering and smoothing for multiple models is not easy to make, since it turns out that the optimal process model parameter settings may differ for the two cases. In this report the we wanted to compare performance for filtering and smoothing for identical parameter settings in the two cases, and the choice to optimize on smoothed estimates was made in the belief that this would give a better process identification. It turns out that optmization on filtered estimates would most likely not have resulted in any different results for the parameters of the individual models in a multiple model filter/smoother. But the optimized transition probabilities seem to put a heavier emphesis on models with a higher process state vector dimensionality, such as CA and CAL, when optimized on smoothed estimates. These process models seem to give more room for improvement while smoothing but decreases straight course stability while filtering. There is thus room for improving the filtering performance in the case of multiple models.

10.3 Future work

Further work may involve studies on the effects of measurement uncertainty and sampling frequency on the accuracy of the state estimates. An extension of the process models into three spatial dimensions, and the impact this will have on the results is also an interesting extension. Also the effect of nonlinear measurement models involving radial velocity, due to doppler measurements, may be studied. Futher work within the area of markov system process models may be to evaluate the results of additional process models, such as the combination $CV + CV + CT + CAL$. Does additional process models come with a cost additional to heavier computations?

A proposed smoothing extension to the *Switch-Time Conditioned IMM* (STC-IMM) filter proposed by *D. Svensson* and *L. Svensson* [7] has been partially evaluated, but without promising results so far. This is an especially interesting topic for future work.

Bibliography

[1] R. E. Kalman
A New Approach to Linear Filtering and Prediction Problems
Research Institute for Advanced Study, Baltimore, Maryland
1960

[2] H. E. Rauch, F. Tung, T. Striebel
Maximum Likelihood Estimates of Linear Dynamic Systems
Lockheed Missiles and Space Company, Palo Alto, California
1965

[3] Simon Särkkä
Unscented Rauch-Tung-Striebel Smoother
Helsinki University of Technology, Helsinki, Finland
2007

[4] N. Nadarajah, R. Tharamarasa, Mike McDonald, T. Kirubarajan
IMM forward and Backward Smoothing for Maneuvering Target Tracking
Curtin University, Perth, WA, Australia
McMaster University, Hamilton, ON, Canada
Defence R & D Canada, Ottawa, ON, Canada
2011

[5] D. Svensson, L. Svensson
Discretization and linearization of time-continous state space models
Chalmers University of Technology, Gothenburg, Sweden
Unpublished

[6] X. Rong LI, V. P. Jilkov
Survey of Maneuvering Target Tracking
University of New Orleans, Lousiana
2003

[7] D. Svensson
Target Tracking in Complex Scenarios
Chalmers University of Technology, Gothenburg, Sweden
2010

[8] L. Danielsson
Tracking and radar sensor modelling for automotive safety systems
Chalmers University of Technology, Gothenburg, Sweden
2010

[9] S. J. Julier, J. K. Uhlmann
A New Extension of the Kalman Filter to Nonlinear Systems
The University of Oxford, Oxford, Oxfordshire, United Kingdom
2010

[10] W. D. Blair, G. A. Watson
Benchmark Problem for Radar Resource Allocation and Tracking Maneuvering Targets in the Presence of ECM
System Research and Technology Department
1996

[11] M. Wahde
Biologically Inspired Optimization Methods: An Introduction
WIT Press
2006

A

Application specific process models

A.1 Constant Acceleration (CA)

$$\mathbf{x} = \begin{pmatrix} x & y & \dot{x} & \dot{y} & \ddot{x} & \ddot{y} \end{pmatrix}^T \tag{A.1}$$

$$\mathbf{P} = \begin{pmatrix} \sigma_x^2 & \sigma_{xy}^2 & \sigma_{x\dot{x}}^2 & \sigma_{x\dot{y}}^2 & \sigma_{x\ddot{x}}^2 & \sigma_{x\ddot{y}}^2 \\ \sigma_{yx}^2 & \sigma_y^2 & \sigma_{y\dot{x}}^2 & \sigma_{y\dot{y}}^2 & \sigma_{y\ddot{x}}^2 & \sigma_{y\ddot{y}}^2 \\ \sigma_{\dot{x}x}^2 & \sigma_{\dot{x}y}^2 & \sigma_{\dot{x}}^2 & \sigma_{\dot{x}\dot{y}}^2 & \sigma_{\dot{x}\ddot{x}}^2 & \sigma_{\dot{x}\ddot{y}}^2 \\ \sigma_{\dot{y}x}^2 & \sigma_{\dot{y}y}^2 & \sigma_{\dot{y}\dot{x}}^2 & \sigma_{\dot{y}}^2 & \sigma_{\dot{y}\ddot{x}}^2 & \sigma_{\dot{y}\ddot{y}}^2 \\ \sigma_{\ddot{x}x}^2 & \sigma_{\ddot{x}y}^2 & \sigma_{\ddot{x}\dot{x}}^2 & \sigma_{\ddot{x}\dot{y}}^2 & \sigma_{\ddot{x}}^2 & \sigma_{\ddot{x}\ddot{y}}^2 \\ \sigma_{\ddot{y}x}^2 & \sigma_{\ddot{y}y}^2 & \sigma_{\ddot{y}\dot{x}}^2 & \sigma_{\ddot{y}\dot{y}}^2 & \sigma_{\ddot{y}\ddot{x}}^2 & \sigma_{\ddot{y}}^2 \end{pmatrix} \tag{A.2}$$

$$f(\mathbf{x}) = \begin{pmatrix} \dot{x} & \dot{y} & \ddot{x} & \ddot{y} & 0 & 0 \end{pmatrix}^T \tag{A.3}$$

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{A.4}$$

$$\mathbf{G} = \begin{pmatrix} T & 0 & \frac{T^2}{2} & 0 & \frac{T^3}{6} & 0 \\ 0 & T & 0 & \frac{T^2}{2} & 0 & \frac{T^3}{6} \\ 0 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 0 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & T \end{pmatrix} \tag{A.5}$$

$$\mathbf{\Sigma} = \begin{pmatrix} \bar{\sigma}^2_{J_x} & 0 \\ 0 & \bar{\sigma}^2_{J_y} \end{pmatrix} \tag{A.6}$$

$$\mathbf{B} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{A.7}$$

$$\mathbf{Q} = \begin{pmatrix} \frac{\bar{\sigma}^2_{J_x} T^6}{36} & 0 & \frac{\bar{\sigma}^2_{J_x} T^5}{12} & 0 & \frac{\bar{\sigma}^2_{J_x} T^4}{6} & 0 \\ 0 & \frac{\bar{\sigma}^2_{J_y} T^6}{36} & 0 & \frac{\bar{\sigma}^2_{J_y} T^5}{12} & 0 & \frac{\bar{\sigma}^2_{J_y} T^4}{6} \\ \frac{\bar{\sigma}^2_{J_x} T^5}{12} & 0 & \frac{\bar{\sigma}^2_{J_x} T^4}{4} & 0 & \frac{\bar{\sigma}^2_{J_x} T^3}{2} & 0 \\ 0 & \frac{\bar{\sigma}^2_{J_y} T^5}{12} & 0 & \frac{\bar{\sigma}^2_{J_y} T^4}{4} & 0 & \frac{\bar{\sigma}^2_{J_y} T^3}{2} \\ \frac{\bar{\sigma}^2_{J_x} T^4}{6} & 0 & \frac{\bar{\sigma}^2_{J_x} T^3}{2} & 0 & \bar{\sigma}^2_{J_x} T^2 & 0 \\ 0 & \frac{\bar{\sigma}^2_{J_y} T^4}{6} & 0 & \frac{\bar{\sigma}^2_{J_y} T^3}{2} & 0 & \bar{\sigma}^2_{J_y} T^2 \end{pmatrix} \tag{A.8}$$

A.2 Constant Velocity (CV)

$$\mathbf{x} = \begin{pmatrix} x & y & \dot{x} & \dot{y} & 0 & 0 \end{pmatrix}^T \tag{A.9}$$

$$\mathbf{P} = \begin{pmatrix} \sigma^2_x & \sigma^2_{xy} & \sigma^2_{x\dot{x}} & \sigma^2_{x\dot{y}} & 0 & 0 \\ \sigma^2_{yx} & \sigma^2_y & \sigma^2_{y\dot{x}} & \sigma^2_{y\dot{y}} & 0 & 0 \\ \sigma^2_{\dot{x}x} & \sigma^2_{\dot{x}y} & \sigma^2_{\dot{x}} & \sigma^2_{\dot{x}\dot{y}} & 0 & 0 \\ \sigma^2_{\dot{y}x} & \sigma^2_{\dot{y}y} & \sigma^2_{\dot{y}\dot{x}} & \sigma^2_{\dot{y}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \bar{\sigma}^2_{\ddot{x}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \bar{\sigma}^2_{\ddot{y}} \end{pmatrix} \tag{A.10}$$

$$f(\mathbf{x}) = \begin{pmatrix} \dot{x} & \dot{y} & 0 & 0 & 0 & 0 \end{pmatrix}^T \tag{A.11}$$

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & T & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{A.12}$$

$$\mathbf{G} = \begin{pmatrix} T & 0 & \frac{T^2}{2} & 0 & 0 & 0 \\ 0 & T & 0 & \frac{T^2}{2} & 0 & 0 \\ 0 & 0 & T & 0 & 0 & 0 \\ 0 & 0 & 0 & T & 0 & 0 \\ 0 & 0 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & T \end{pmatrix} \tag{A.13}$$

$$\mathbf{\Sigma} = \begin{pmatrix} \bar{\sigma}_{\ddot{x}}^2 & 0 \\ 0 & \bar{\sigma}_{\ddot{y}}^2 \end{pmatrix} \tag{A.14}$$

$$\mathbf{B} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{A.15}$$

$$\mathbf{Q} = \begin{pmatrix} \frac{\bar{\sigma}_{\ddot{x}}^2 T^4}{4} & 0 & \frac{\bar{\sigma}_{\ddot{x}}^2 T^3}{2} & 0 & 0 & 0 \\ 0 & \frac{\bar{\sigma}_{\ddot{y}}^2 T^4}{4} & 0 & \frac{\bar{\sigma}_{\ddot{y}}^2 T^3}{2} & 0 & 0 \\ \frac{\bar{\sigma}_{\ddot{x}}^2 T^3}{2} & 0 & \bar{\sigma}_{\ddot{x}}^2 T^2 & 0 & 0 & 0 \\ 0 & \frac{\bar{\sigma}_{\ddot{y}}^2 T^3}{2} & 0 & \bar{\sigma}_{\ddot{y}}^2 T^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{A.16}$$

A.3 Contant Acceleration Local Coordinates (CAL)

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ v \\ \phi \\ a_t \\ a_n \end{pmatrix} = \begin{pmatrix} x \\ y \\ \sqrt{\dot{x}^2 + \dot{y}^2} \\ atan2(\dot{y}, \dot{x}) \\ \ddot{x}\cos(\phi) + \ddot{y}\sin(\phi) \\ \ddot{y}\cos(\phi) - \ddot{x}\sin(\phi) \end{pmatrix} \tag{A.17}$$

$$\mathbf{P} = \begin{pmatrix} \sigma^2_x & \sigma^2_{xy} & \sigma^2_{xv} & \sigma^2_{x\phi} & \sigma^2_{xa_t} & \sigma^2_{xa_n} \\ \sigma^2_{yx} & \sigma^2_y & \sigma^2_{yv} & \sigma^2_{y\phi} & \sigma^2_{ya_t} & \sigma^2_{ya_n} \\ \sigma^2_{vx} & \sigma^2_{vy} & \sigma^2_v & \sigma^2_{v\phi} & \sigma^2_{va_t} & \sigma^2_{va_n} \\ \sigma^2_{\phi x} & \sigma^2_{\phi y} & \sigma^2_{\phi v} & \sigma^2_\phi & \sigma^2_{\phi a_t} & \sigma^2_{\phi a_n} \\ \sigma^2_{a_t x} & \sigma^2_{a_t y} & \sigma^2_{a_t v} & \sigma^2_{a_t \phi} & \sigma^2_{a_t} & \sigma^2_{a_t a_n} \\ \sigma^2_{a_n x} & \sigma^2_{a_n y} & \sigma^2_{a_n v} & \sigma^2_{a_n \phi} & \sigma^2_{a_n a_t} & \sigma^2_{a_n} \end{pmatrix} \tag{A.18}$$

$$f(\mathbf{x}) = \begin{pmatrix} v\cos(\phi) & v\sin(\phi) & a_t & \frac{a_n}{v} & 0 & 0 \end{pmatrix}^T \tag{A.19}$$

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & T\cos(\phi) + \frac{a_n T^2}{2v}\sin(\phi) & -vT\sin(\phi) & \frac{T^2}{2}\cos(\phi) + \frac{a_n T^3}{6v}\sin(\phi) & -\frac{T^2}{2}\sin(\phi) \\ 0 & 1 & T\sin(\phi) - \frac{a_n T^2}{2v}\cos(\phi) & vT\cos(\phi) & \frac{T^2}{2}\sin(\phi) - \frac{a_n T^3}{6v}\cos(\phi) & \frac{T^2}{2}\cos(\phi) \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & -\frac{a_n T}{v^2} & 1 & -\frac{a_n T^2}{2v^2} & \frac{T}{v} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{A.20}$$

$$\mathbf{G} = \begin{pmatrix} T & 0 & \frac{T^2}{2}\cos(\phi) + \frac{a_n T^3}{6v}\sin(\phi) & -v\frac{T^2}{2}\sin(\phi) & \frac{T^3}{6}\cos(\phi) + \frac{a_n T^4}{24v}\sin(\phi) & -\frac{T^3}{6}\sin(\phi) \\ 0 & T & \frac{T^2}{2}\sin(\phi) - \frac{a_n T^3}{6v}\cos(\phi) & v\frac{T^2}{2}\cos(\phi) & \frac{T^3}{6}\sin(\phi) - \frac{a_n T^4}{24v}\cos(\phi) & \frac{T^3}{6}\cos(\phi) \\ 0 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 0 & -\frac{a_n T^2}{2v^2} & T & -\frac{a_n T^3}{6v^2} & \frac{T^2}{2v} \\ 0 & 0 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & T \end{pmatrix} \tag{A.21}$$

$$\mathbf{\Sigma} = \begin{pmatrix} \bar{\sigma}^2_{J_t} & 0 \\ 0 & \bar{\sigma}^2_{J_n} \end{pmatrix} \tag{A.22}$$

$$\mathbf{B} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{A.23}$$

The noise covariance matrix $\mathbf{Q}$ was too large to fit on one paper, so I leave the computation of $\mathbf{Q}$ as an exercise. Have fun!

A.4 Coordinated Turn (CT)

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ v \\ \phi \\ 0 \\ \Omega \end{pmatrix} = \begin{pmatrix} x \\ y \\ \sqrt{\dot{x}^2 + \dot{y}^2} \\ atan2(\dot{y}, \dot{x}) \\ 0 \\ v^{-1}\left(\ddot{y}\cos(\phi) - \ddot{x}\sin(\phi)\right) \end{pmatrix} \tag{A.24}$$

$$\mathbf{P} = \begin{pmatrix} \sigma_x^2 & \sigma_{xy}^2 & \sigma_{xv}^2 & \sigma_{x\phi}^2 & 0 & \sigma_{x\Omega}^2 \\ \sigma_{yx}^2 & \sigma_y^2 & \sigma_{yv}^2 & \sigma_{y\phi}^2 & 0 & \sigma_{y\Omega}^2 \\ \sigma_{vx}^2 & \sigma_{vy}^2 & \sigma_v^2 & \sigma_{v\phi}^2 & 0 & \sigma_{v\Omega}^2 \\ \sigma_{\phi x}^2 & \sigma_{\phi y}^2 & \sigma_{\phi v}^2 & \sigma_\phi^2 & 0 & \sigma_{\phi\Omega}^2 \\ 0 & 0 & 0 & 0 & \bar{\sigma}_{a_t}^2 & 0 \\ \sigma_{\Omega x}^2 & \sigma_{\Omega y}^2 & \sigma_{\Omega v}^2 & \sigma_{\Omega\phi}^2 & 0 & \sigma_\Omega^2 \end{pmatrix} \tag{A.25}$$

$$f(\mathbf{x}) = \begin{pmatrix} v\cos(\phi) & v\sin(\phi) & 0 & \Omega & 0 & 0 \end{pmatrix}^T \tag{A.26}$$

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & T\cos(\phi) & -vT\sin(\phi) & 0 & -\frac{vT^2}{2}\sin(\phi) \\ 0 & 1 & T\sin(\phi) & vT\cos(\phi) & 0 & \frac{vT^2}{2}\cos(\phi) \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{A.27}$$

$$\mathbf{G} = \begin{pmatrix} T & 0 & \frac{T^2}{2}\cos(\phi) & -\frac{vT^2}{2}\sin(\phi) & 0 & -\frac{vT^3}{6}\sin(\phi) \\ 0 & T & \frac{T^2}{2}\sin(\phi) & \frac{vT^2}{2}\cos(\phi) & 0 & \frac{vT^3}{6}\cos(\phi) \\ 0 & 0 & T & 0 & 0 & 0 \\ 0 & 0 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 0 & 0 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & T \end{pmatrix} \tag{A.28}$$

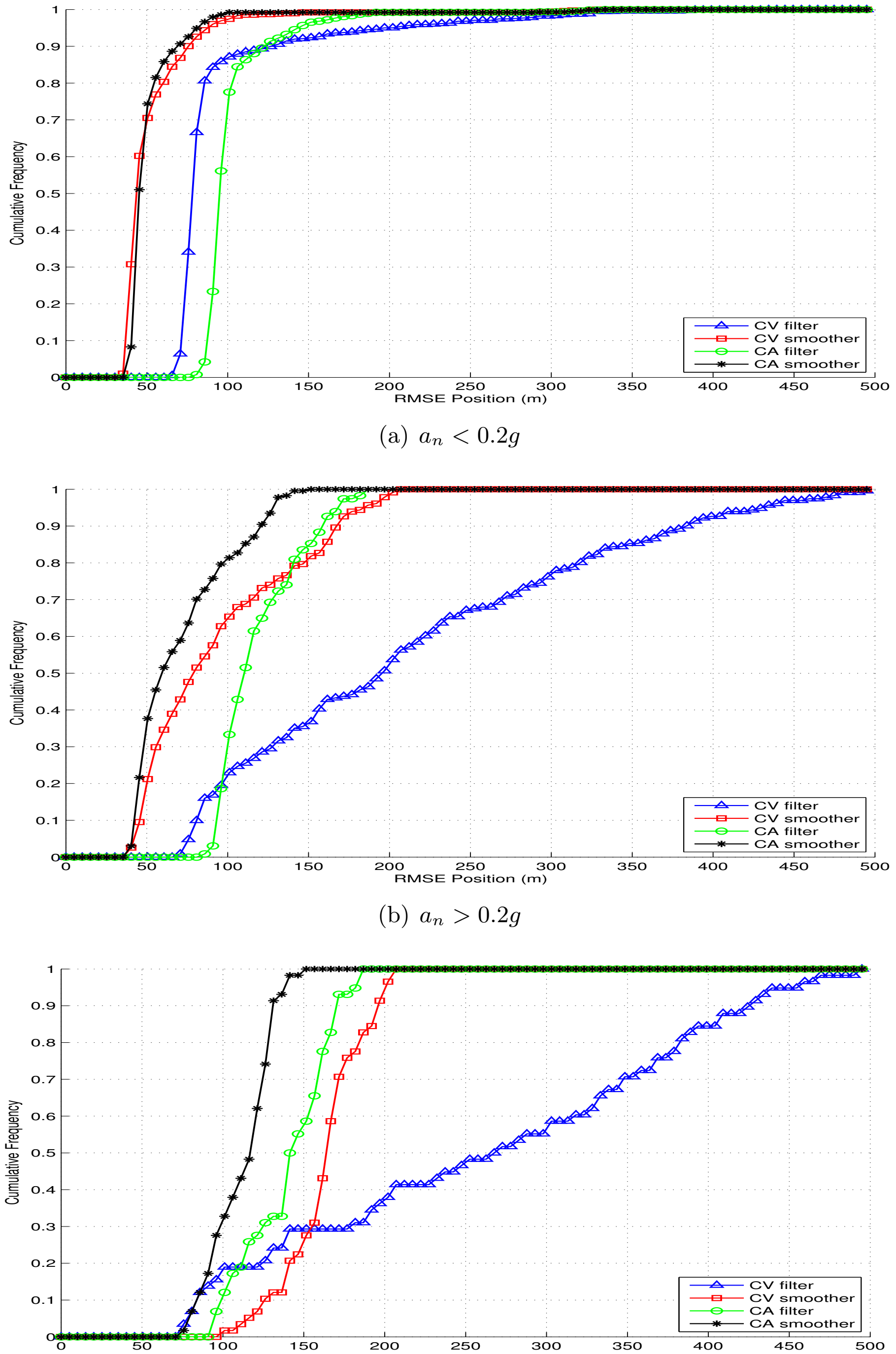
$$\mathbf{\Sigma} = \begin{pmatrix} \bar{\sigma}_{a_t}^2 & 0 \\ 0 & \bar{\sigma}_{\dot{\Omega}}^2 \end{pmatrix} \tag{A.29}$$

$$\mathbf{B} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \tag{A.30}$$

$$
\mathbf{Q} = \begin{pmatrix}
\frac{\bar{\sigma}_{a_t}^2 T^4}{4}\cos^2\phi + \frac{\bar{\sigma}_{\dot{\Omega}}^2 v^2 T^6 \sin^2\phi}{36} & \frac{\bar{\sigma}_{a_t}^2 T^4}{72}\sin(2\phi)(9\bar{\sigma}_{a_t}^2 - v^2\bar{\sigma}_{\dot{\Omega}}^2) & \frac{\bar{\sigma}_{a_t}^2 T^3}{2}\cos\phi & -\frac{\bar{\sigma}_{\dot{\Omega}}^2 v T^5}{12}\sin\phi & 0 & -\frac{\bar{\sigma}_{\dot{\Omega}}^2 v T^4}{6}\sin\phi \\
\frac{\bar{\sigma}_{a_t}^2 T^4}{72}\sin(2\phi)(9\bar{\sigma}_{a_t}^2 - v^2\bar{\sigma}_{\dot{\Omega}}^2) & \frac{\bar{\sigma}_{a_t}^2 T^4}{4}\sin^2\phi + \frac{\bar{\sigma}_{\dot{\Omega}}^2 v^2 T^6 \cos^2\phi}{36} & \frac{\bar{\sigma}_{a_t}^2 T^3}{2}\sin\phi & \frac{\bar{\sigma}_{\dot{\Omega}}^2 v T^5}{12}\cos\phi & 0 & \frac{\bar{\sigma}_{\dot{\Omega}}^2 v T^4}{6}\cos\phi \\
\frac{\bar{\sigma}_{a_t}^2 T^3}{2}\cos\phi & \frac{\bar{\sigma}_{a_t}^2 T^3}{2}\sin\phi & \bar{\sigma}_{a_t}^2 T^2 & 0 & 0 & 0 \\
-\frac{\bar{\sigma}_{\dot{\Omega}}^2 v T^5}{12}\sin\phi & \frac{\bar{\sigma}_{a_t}^2 v T^5}{12}\cos\phi & 0 & \frac{\bar{\sigma}_{\dot{\Omega}}^2 T^4}{4} & 0 & \frac{\bar{\sigma}_{\dot{\Omega}}^2 T^3}{2} \\
0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{\bar{\sigma}_{\dot{\Omega}}^2 v T^4}{6}\sin\phi & \frac{\bar{\sigma}_{a_t}^2 v T^4}{6}\cos\phi & 0 & \frac{\bar{\sigma}_{\dot{\Omega}}^2 T^3}{2} & 0 & \bar{\sigma}_{\dot{\Omega}}^2 T^2
\end{pmatrix} \tag{A.31}
$$

B

Figures

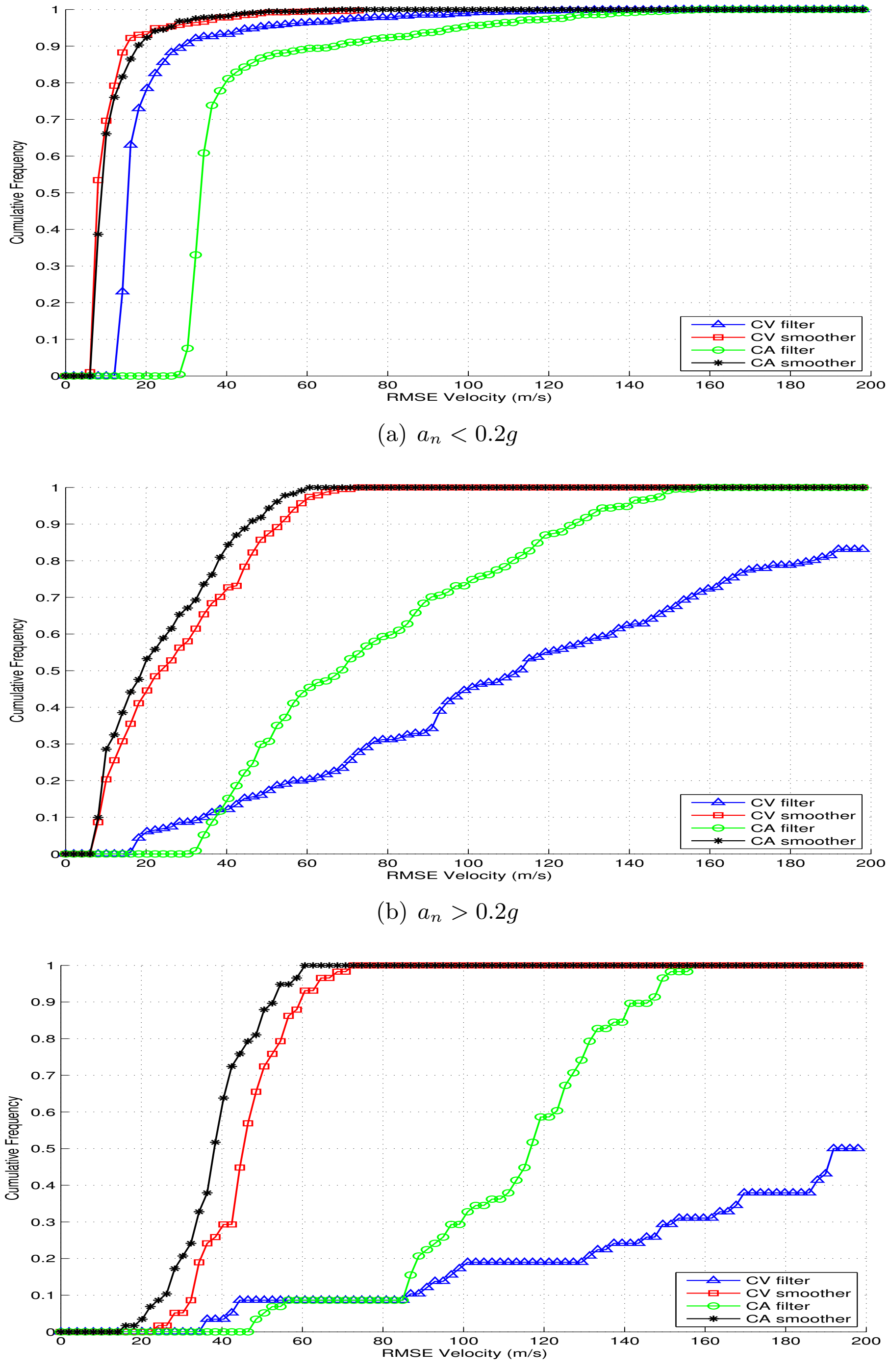


(a) $a_n < 0.2g$

(b) $a_n > 0.2g$

(c) $a_n > 5g$

Figure B.1: Position estimation RMSE cumulative frequency distributions for *CV* and *CA* filters and smoothers

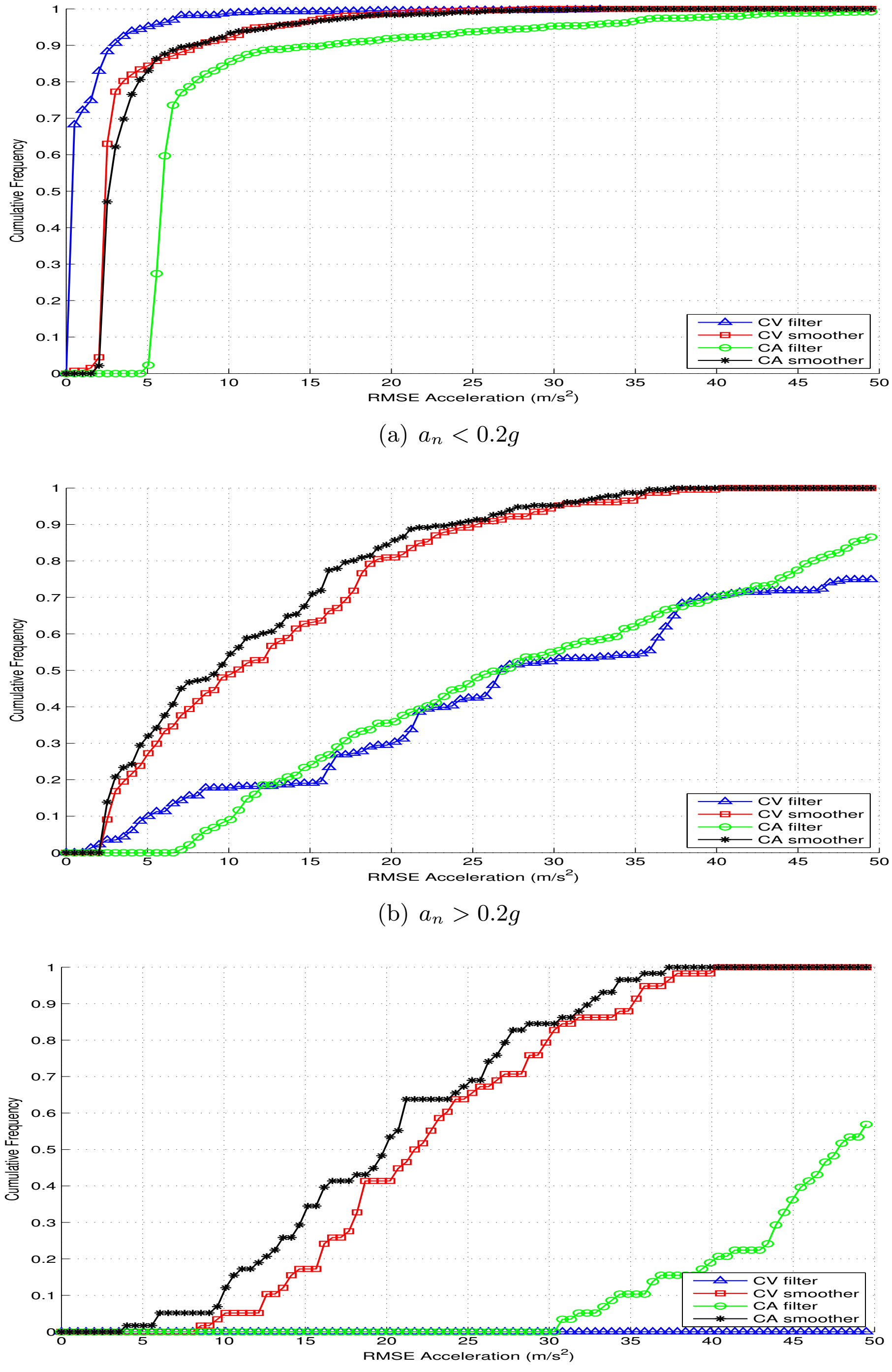


(a) $a_n < 0.2g$

(b) $a_n > 0.2g$

(c) $a_n > 5g$

Figure B.2: Velocity estimation RMSE cumulative frequency distributions for *CV* and *CA* filters and smoothers

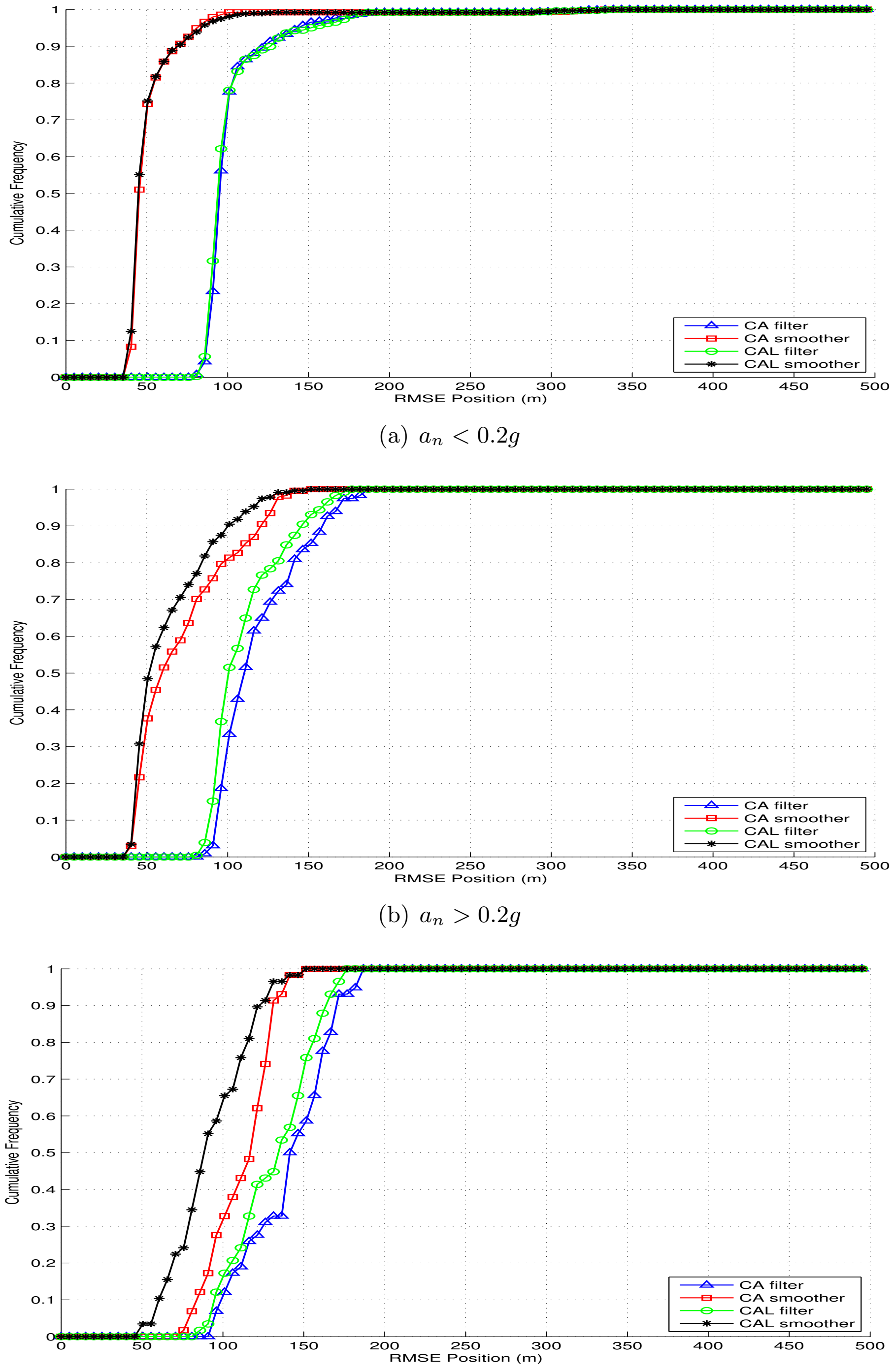


(a) $a_n < 0.2g$

(b) $a_n > 0.2g$

(c) $a_n > 5g$

Figure B.3: Acceleration estimation RMSE cumulative frequency distributions for *CV* and *CA* filters and smoothers

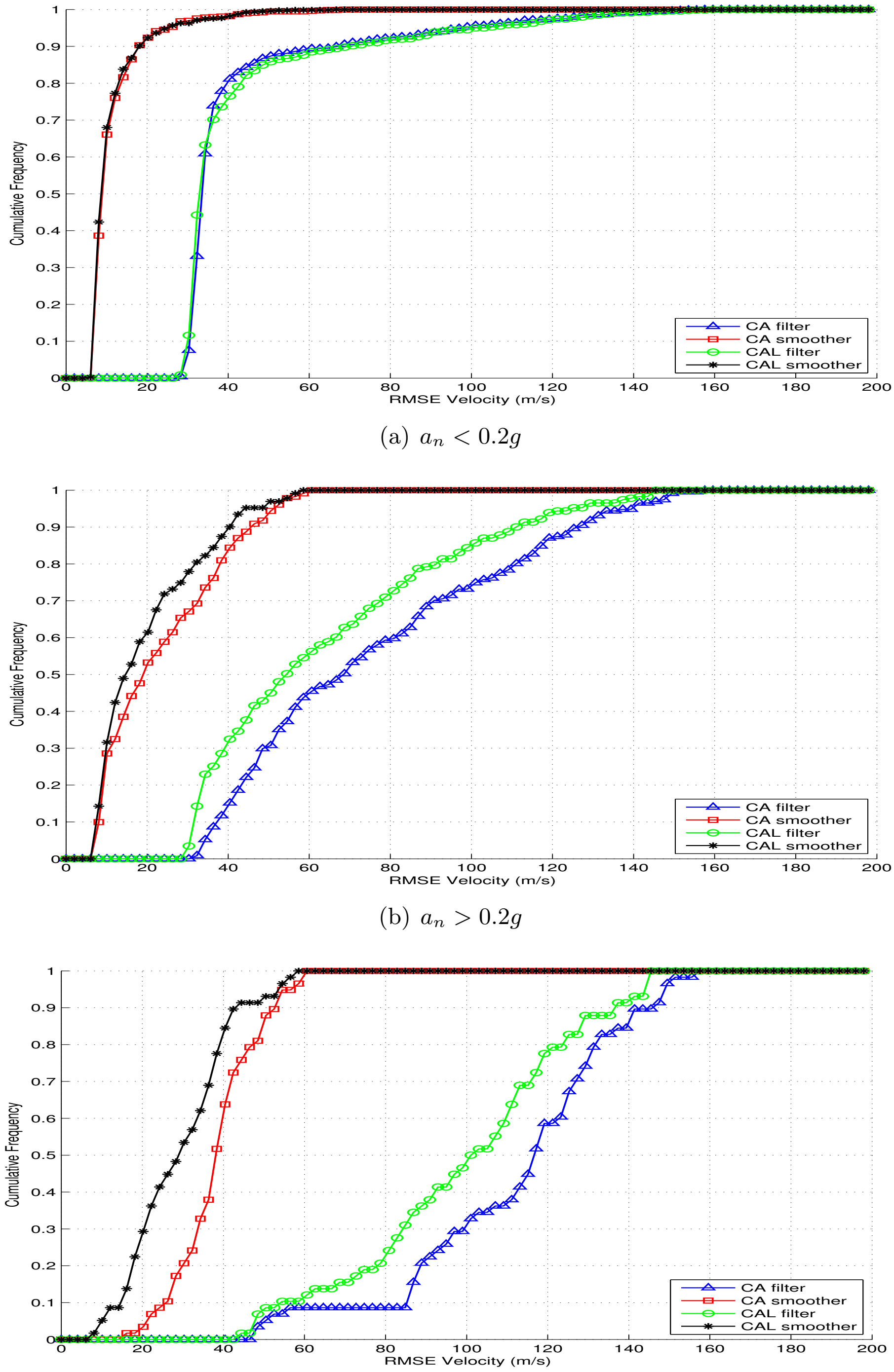


(a) $a_n < 0.2g$

(b) $a_n > 0.2g$

(c) $a_n > 5g$

Figure B.4: Position estimation RMSE cumulative frequency distributions for *CA* and *CAL* filters and smoothers

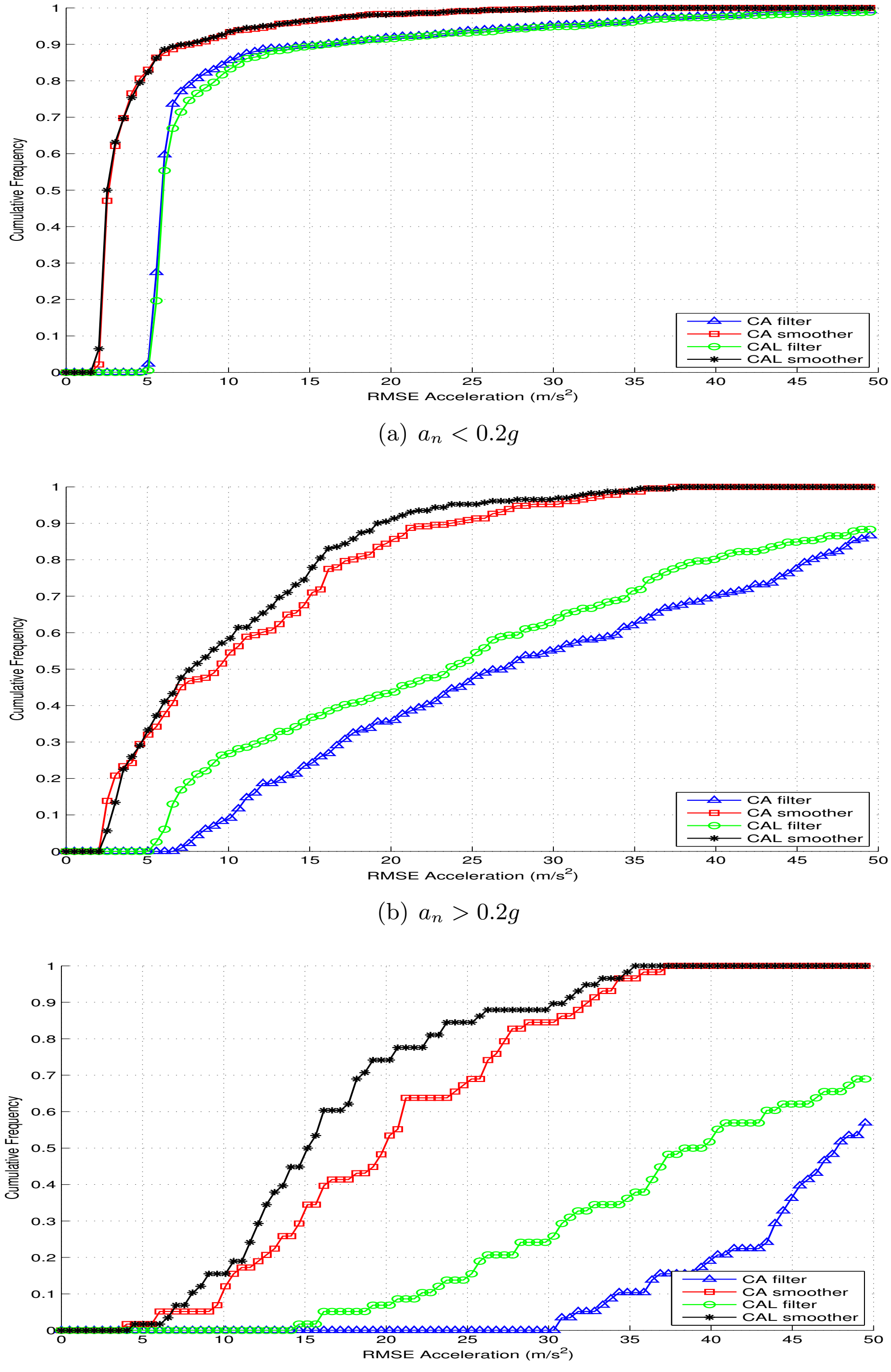


(a) $a_n < 0.2g$

(b) $a_n > 0.2g$

(c) $a_n > 5g$

Figure B.5: Velocity estimation RMSE cumulative frequency distributions for *CA* and *CAL* filters and smoothers

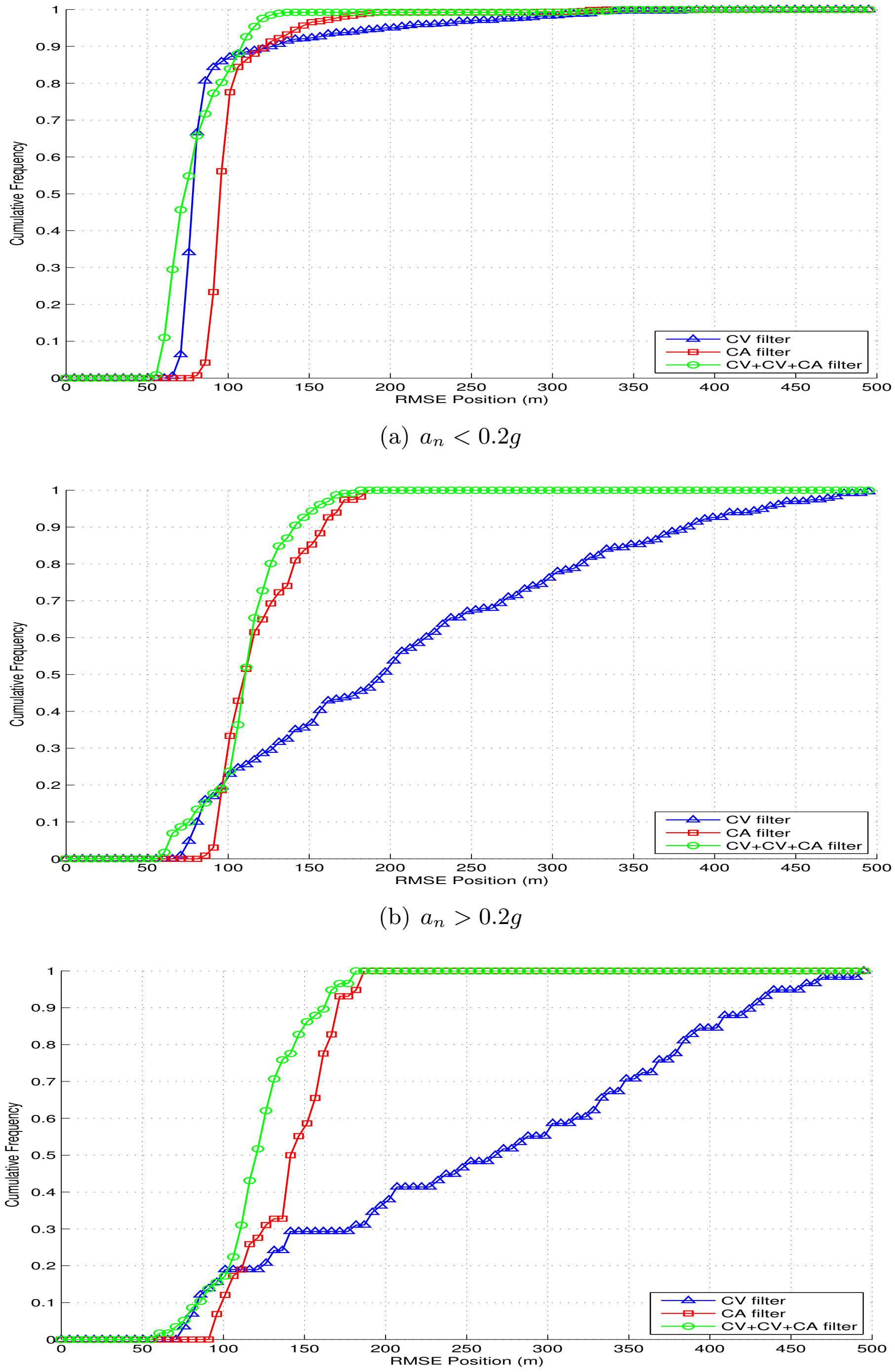


(a) $a_n < 0.2g$

(b) $a_n > 0.2g$

(c) $a_n > 5g$

Figure B.6: Acceleration estimation RMSE cumulative frequency distributions for *CA* and *CAL* filters and smoothers

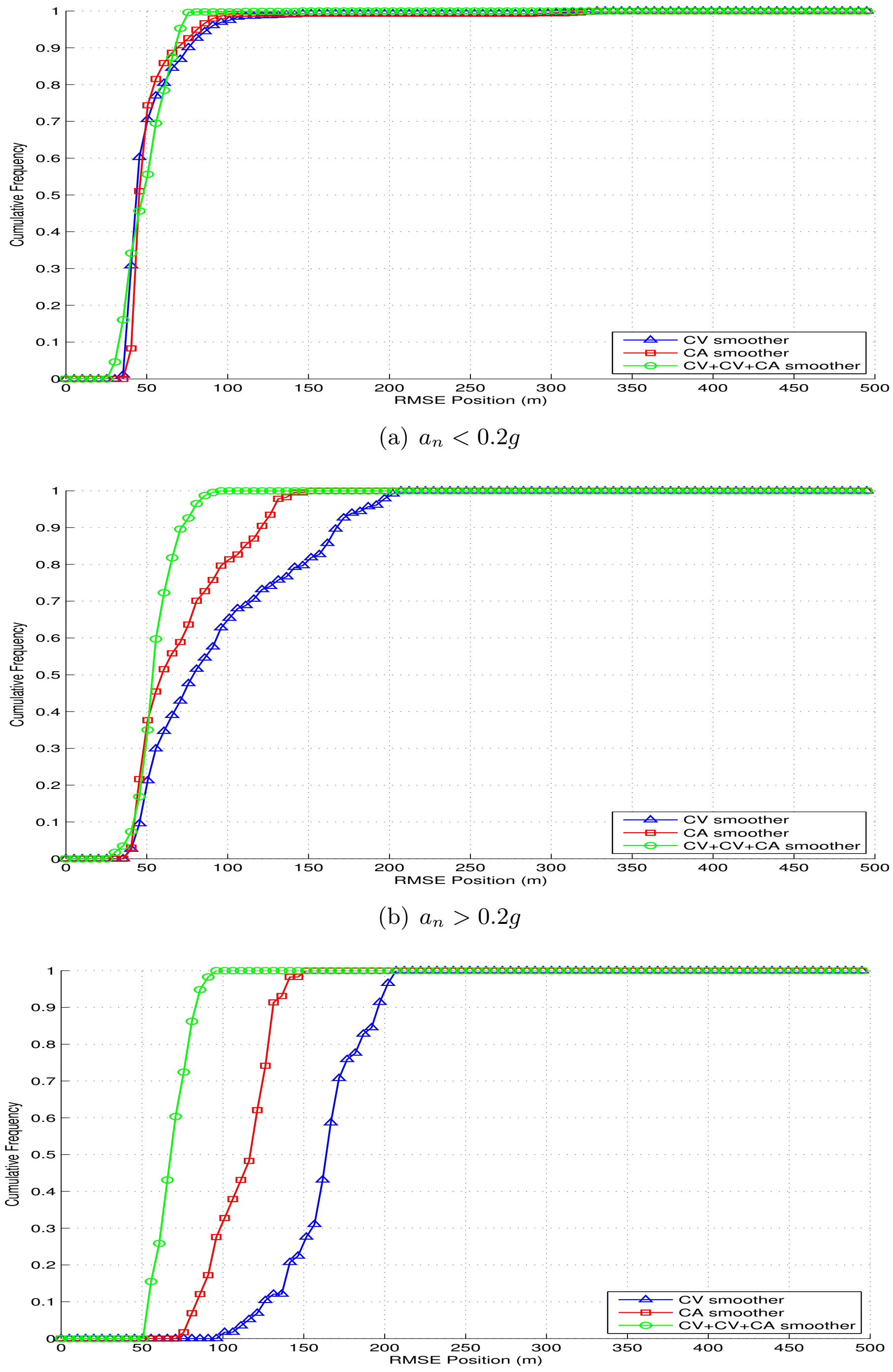


(a) $a_n < 0.2g$

(b) $a_n > 0.2g$

(c) $a_n > 5g$

Figure B.7: Position estimation RMSE cumulative frequency distributions for *CV*, *CA* and *CV+CV+CA* filters

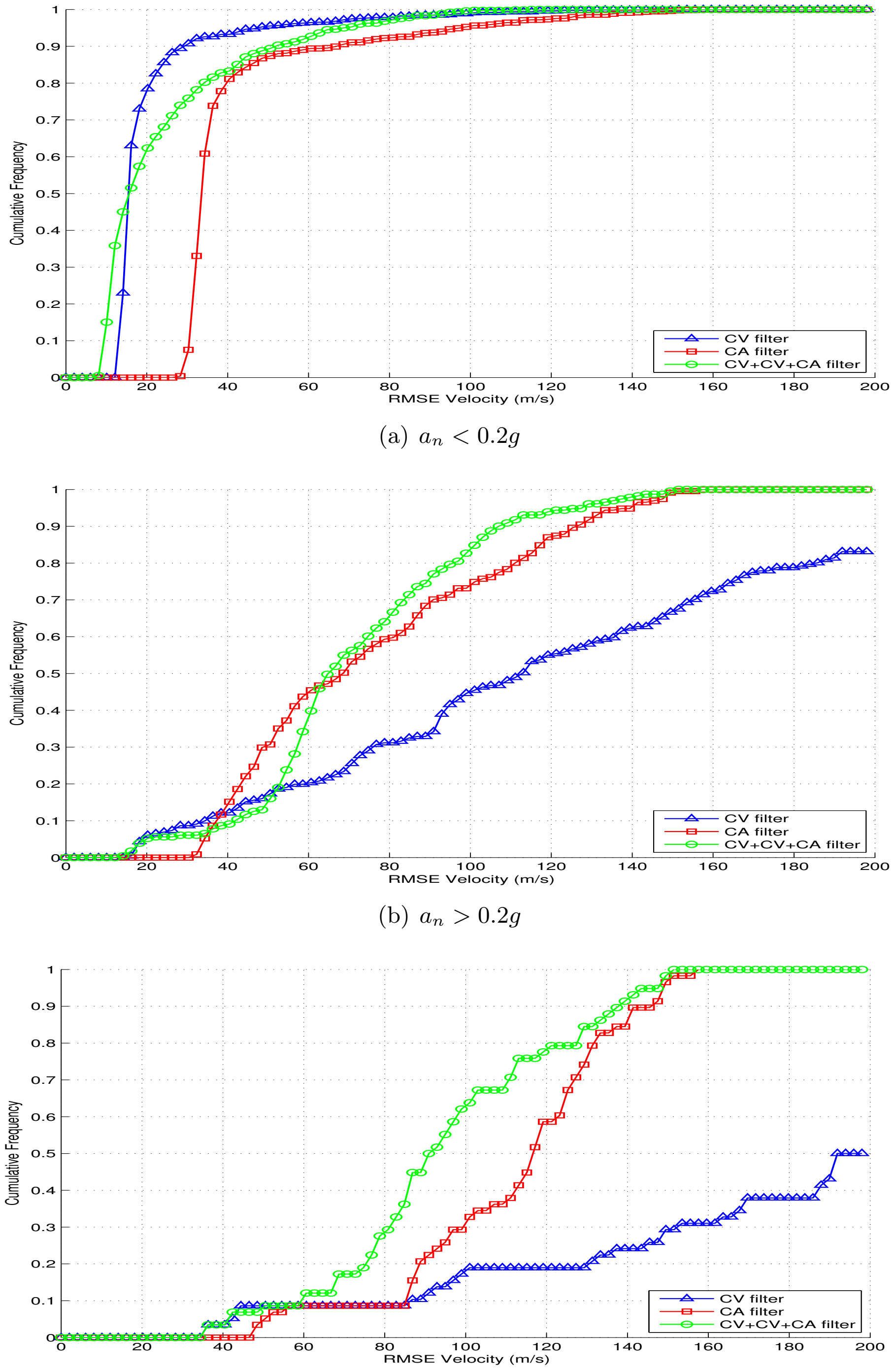


(a) $a_n < 0.2g$

(b) $a_n > 0.2g$

(c) $a_n > 5g$

Figure B.8: Position estimation RMSE cumulative frequency distributions for *CV*, *CA* and *CV+CV+CA* smoothers

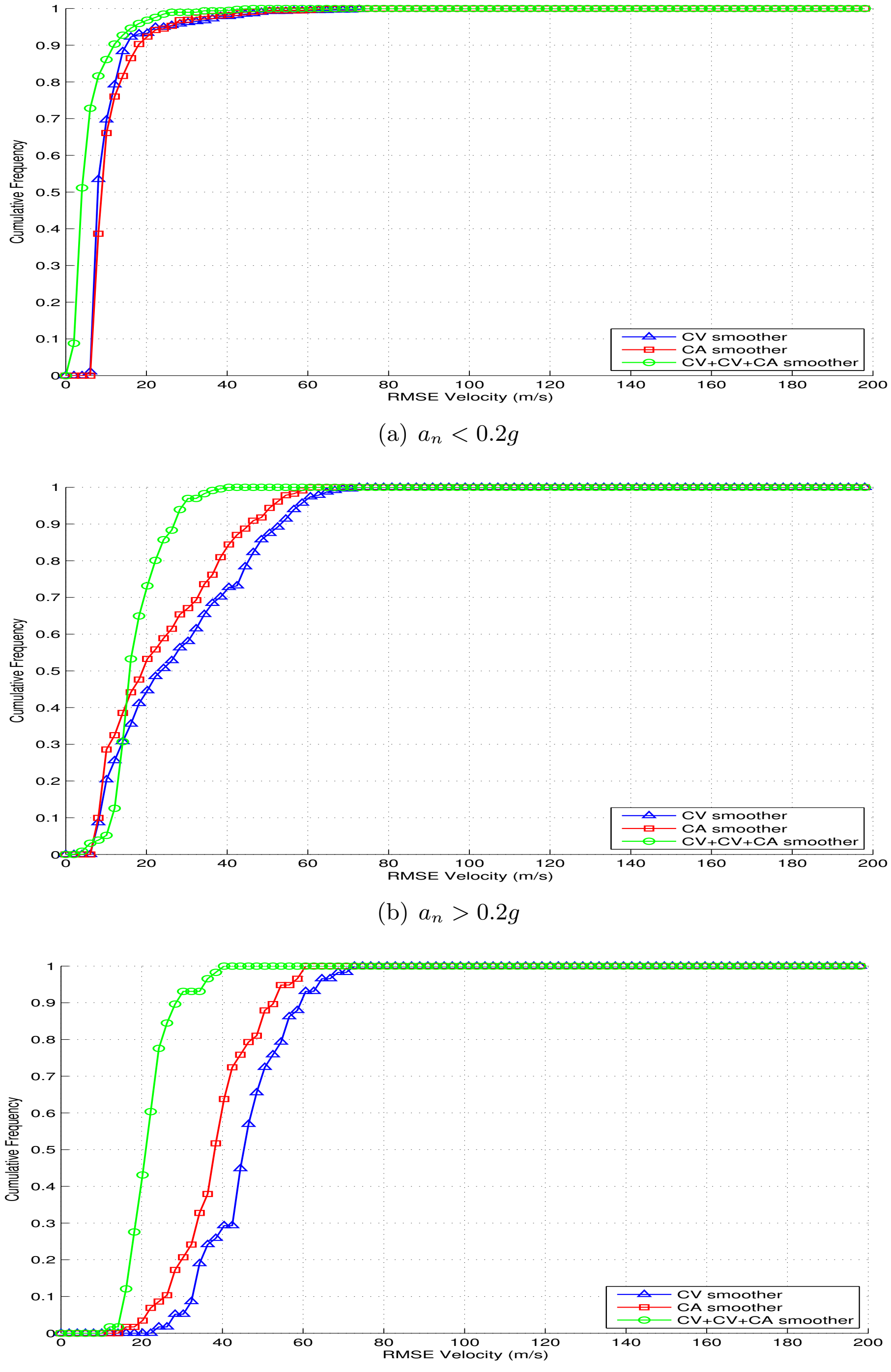


(a) $a_n < 0.2g$

(b) $a_n > 0.2g$

(c) $a_n > 5g$

Figure B.9: Velocity estimation RMSE cumulative frequency distributions for *CV*, *CA* and *CV+CV+CA* filters

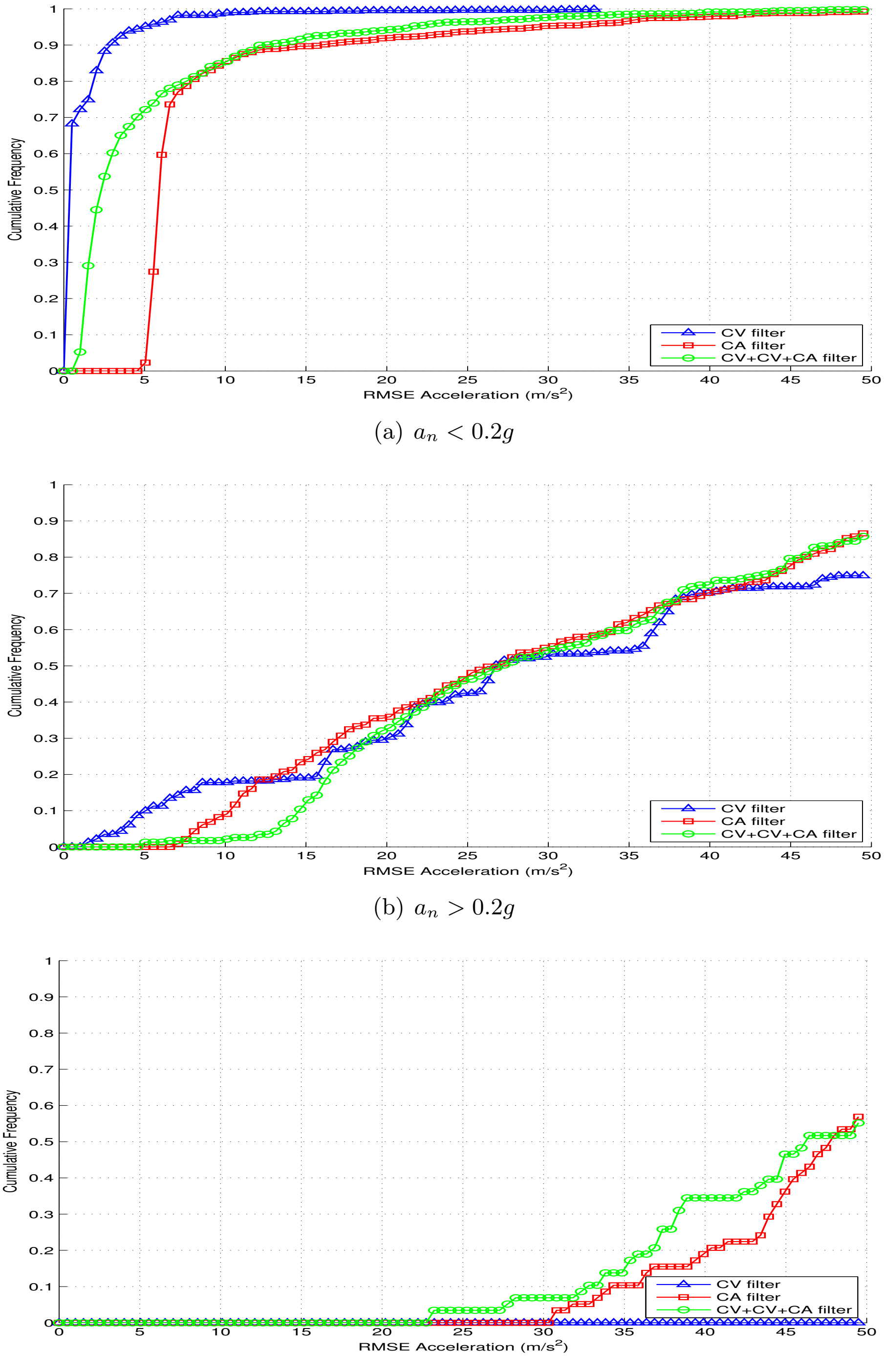


(a) $a_n < 0.2g$

(b) $a_n > 0.2g$

(c) $a_n > 5g$

Figure B.10: Velocity estimation RMSE cumulative frequency distributions for *CV*, *CA* and *CV+CV+CA* smoothers

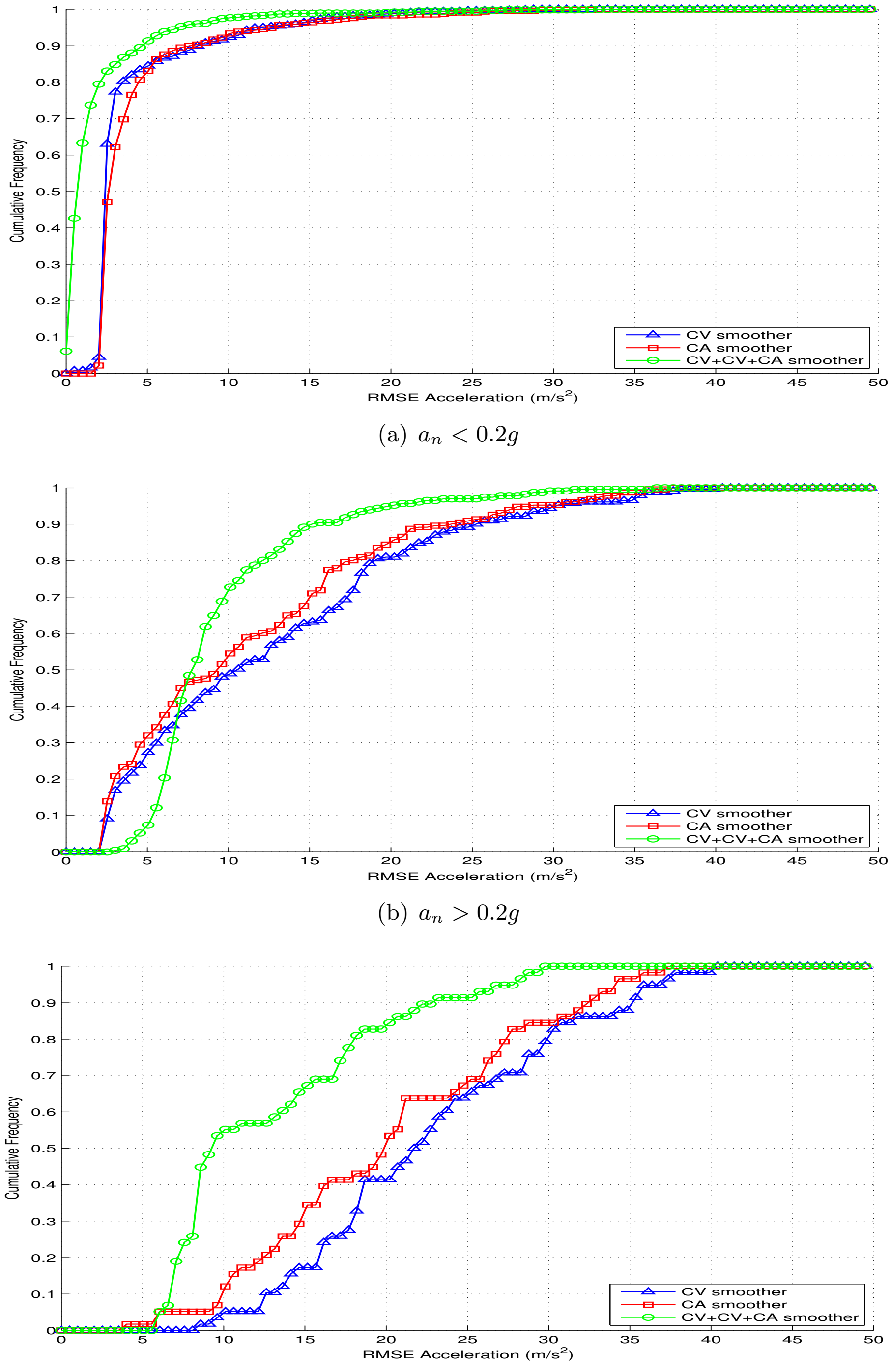


(a) $a_n < 0.2g$

(b) $a_n > 0.2g$

(c) $a_n > 5g$

Figure B.11: Acceleration estimation RMSE cumulative frequency distributions for *CV*, *CA* and *CV+CV+CA* filters

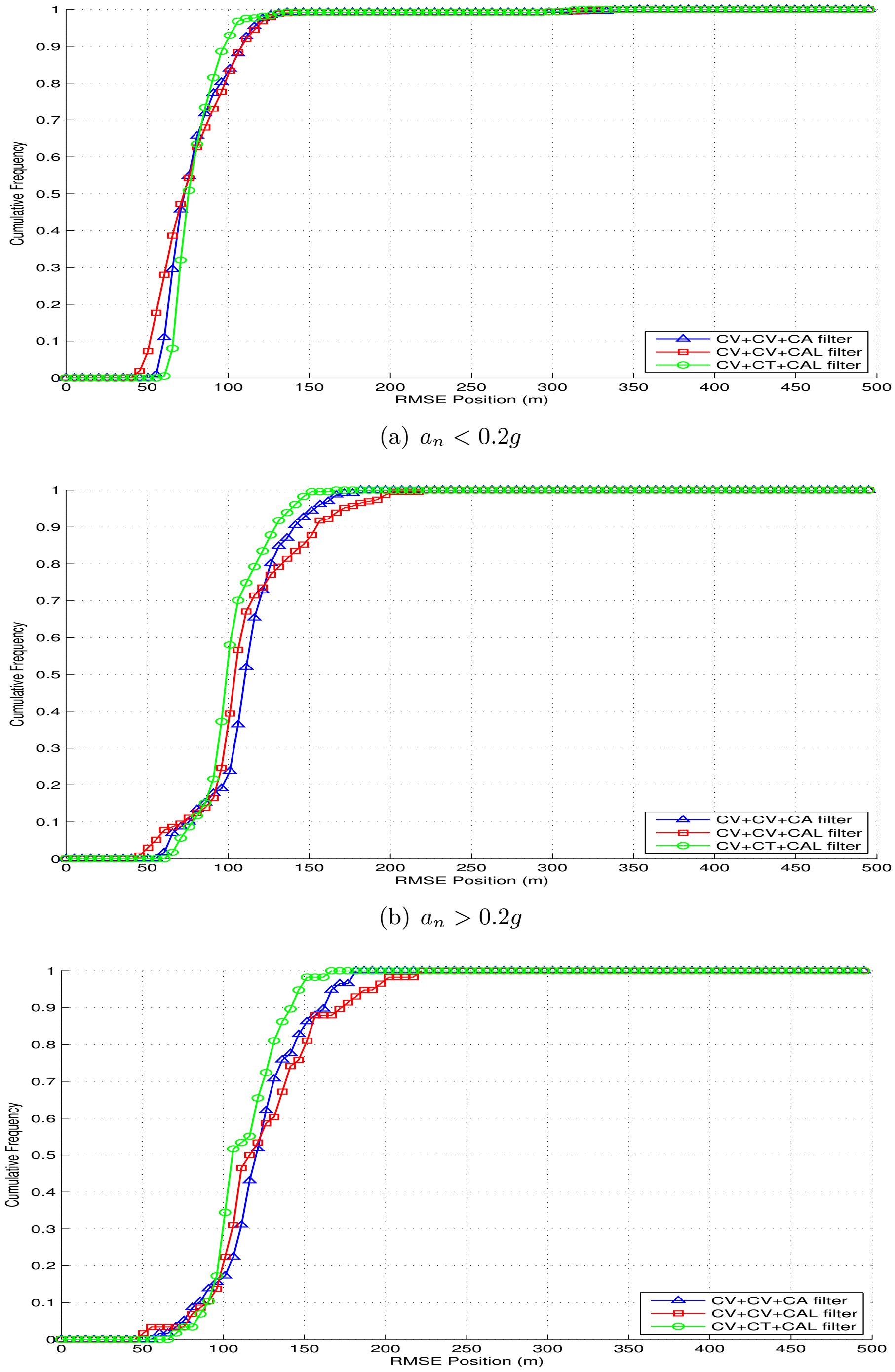


(a) $a_n < 0.2g$

(b) $a_n > 0.2g$

(c) $a_n > 5g$

Figure B.12: Acceleration estimation RMSE cumulative frequency distributions for *CV*, *CAL* and *CV+CV+CA* smoothers

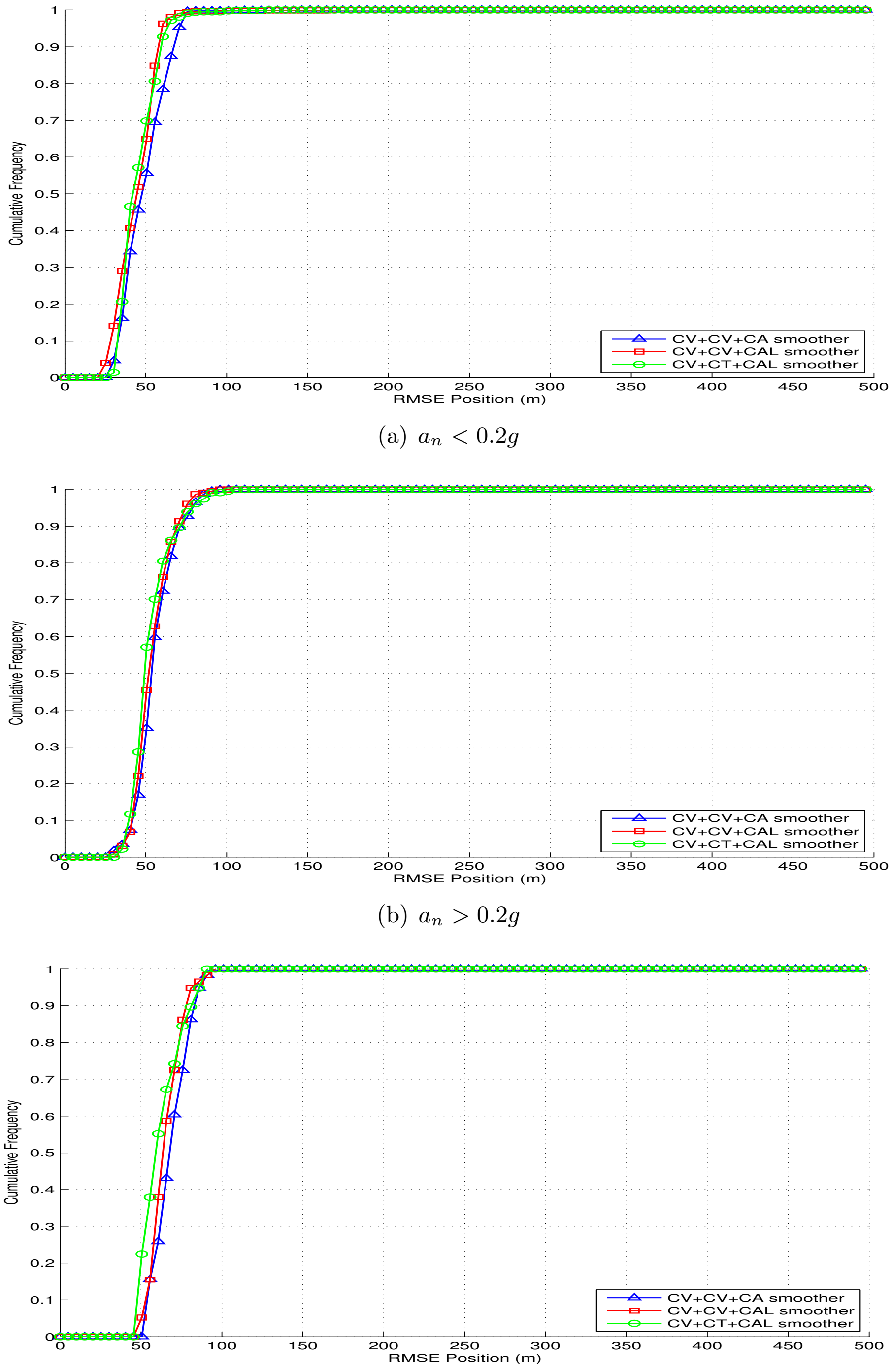


(a) $a_n < 0.2g$

(b) $a_n > 0.2g$

(c) $a_n > 5g$

Figure B.13: Position estimation RMSE cumulative frequency distributions for *CV+CV+CA*, *CV+CV+CAL* and *CV+CT+CAL* filters

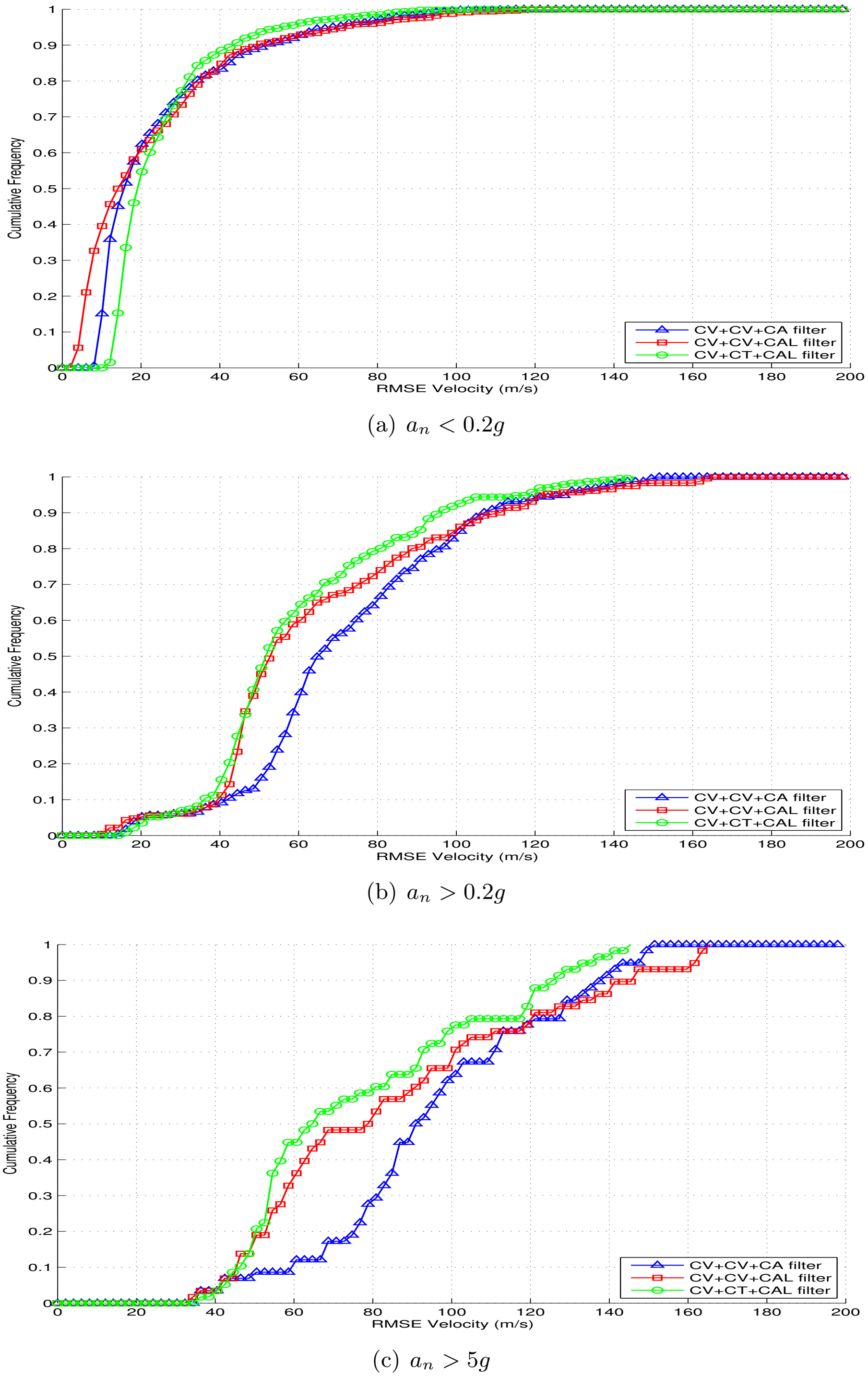


(a) $a_n < 0.2g$

(b) $a_n > 0.2g$

(c) $a_n > 5g$

Figure B.14: Position estimation RMSE cumulative frequency distributions for *CV+CV+CA*, *CV+CV+CAL* and *CV+CT+CAL* smoothers

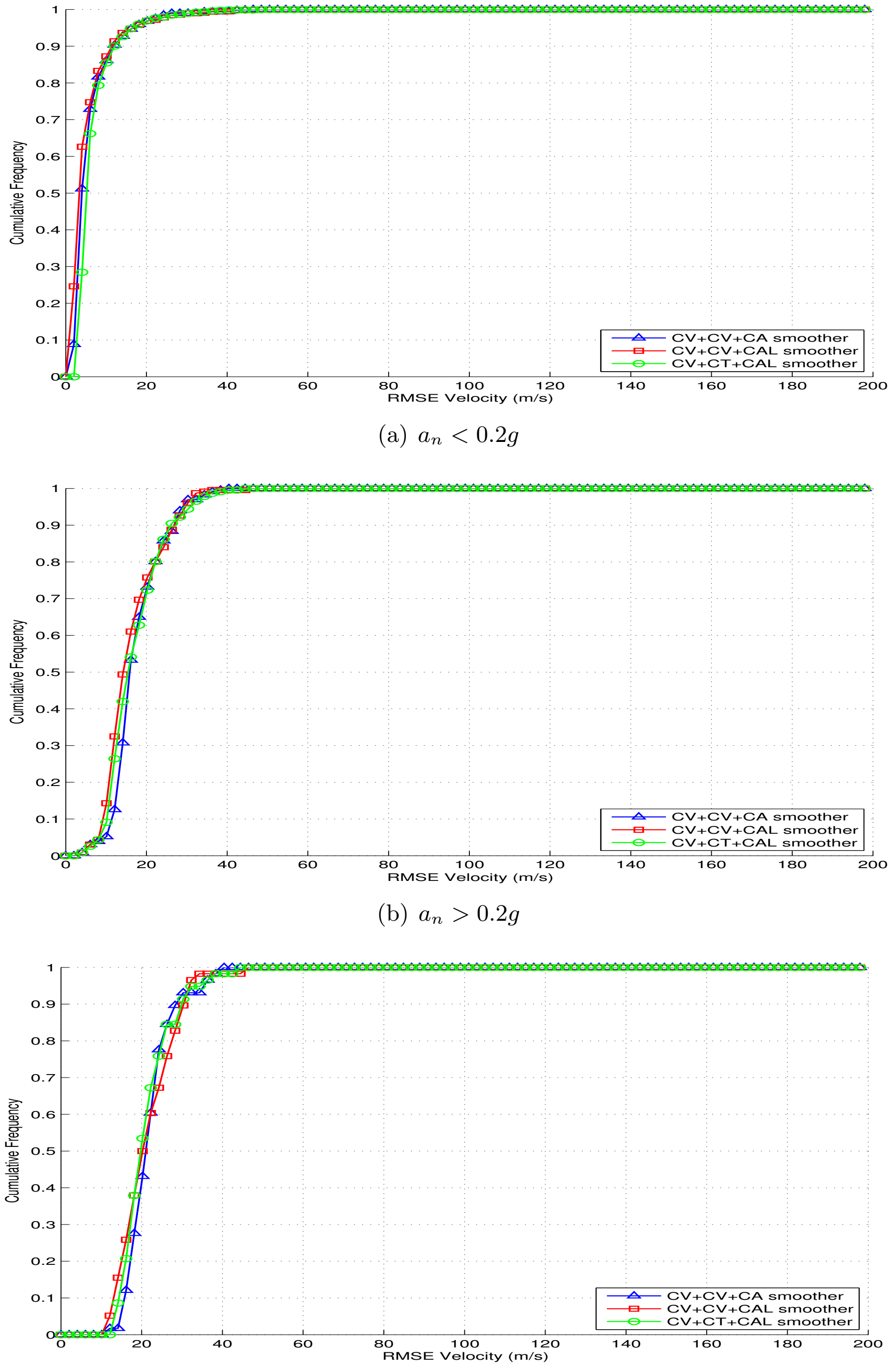


(a) $a_n < 0.2g$

(b) $a_n > 0.2g$

(c) $a_n > 5g$

Figure B.15: Velocity estimation RMSE cumulative frequency distributions for *CV+CV+CA*, *CV+CV+CAL* and *CV+CT+CAL* filters

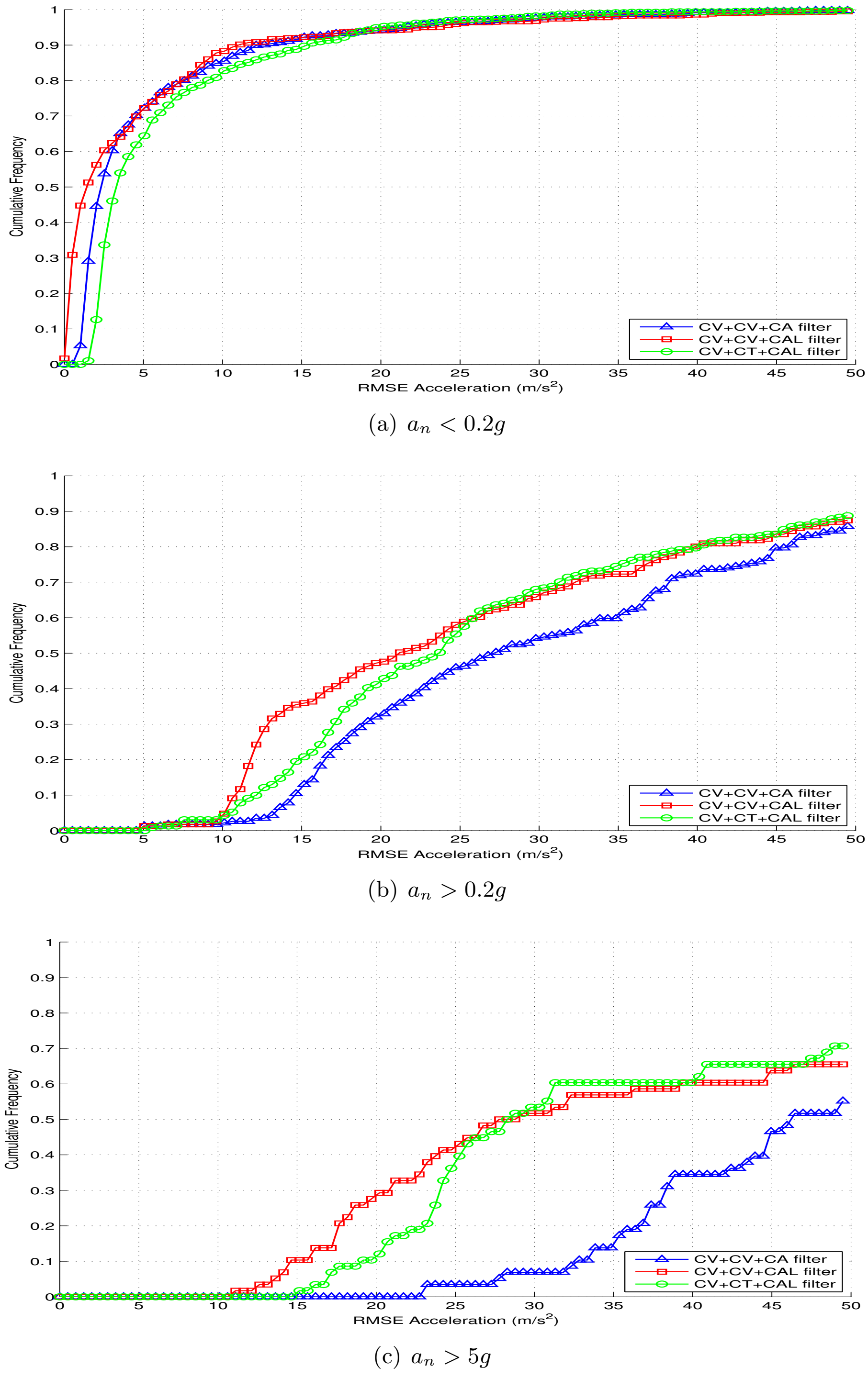


(a) $a_n < 0.2g$

(b) $a_n > 0.2g$

(c) $a_n > 5g$

Figure B.16: Velocity estimation RMSE cumulative frequency distributions for *CV+CV+CA*, *CV+CV+CAL* and *CV+CT+CAL* smoothers

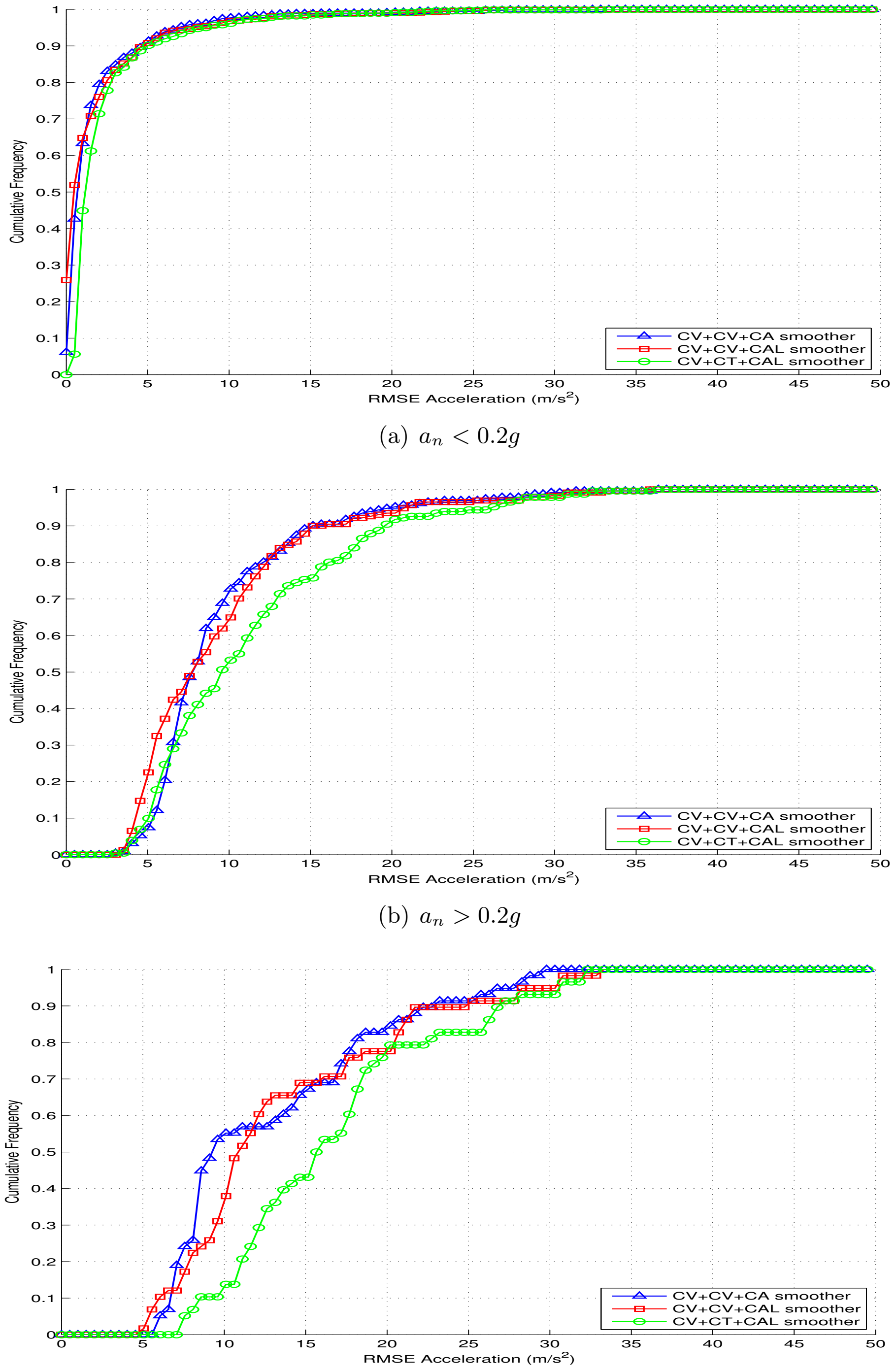


(a) $a_n < 0.2g$

(b) $a_n > 0.2g$

(c) $a_n > 5g$

Figure B.17: Acceleration estimation RMSE cumulative frequency distributions for *CV+CV+CA*, *CV+CV+CAL* and *CV+CT+CAL* filters



(a) $a_n < 0.2g$

(b) $a_n > 0.2g$

(c) $a_n > 5g$

Figure B.18: Acceleration estimation RMSE cumulative frequency distributions for *CV+CV+CA*, *CV+CV+CAL* and *CV+CT+CAL* smoothers