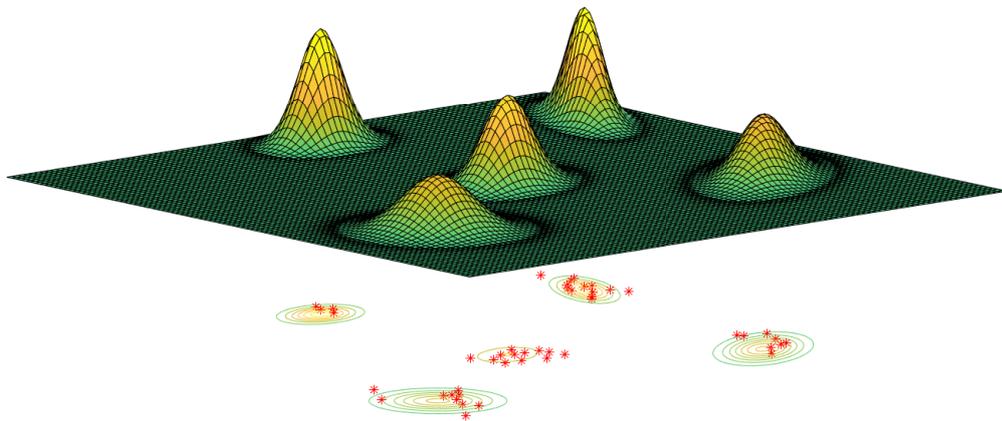




**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

---



# Trajectory Smoothing for Multiple Extended Objects

Batch estimation based on the random matrix model and expectation maximization

Master's thesis in Systems, Control and Mechatronics

JAKOB BRAMSTÅNG



MASTER'S THESIS 2018:EX042

# Trajectory Smoothing for Multiple Extended Objects

Batch estimation based on the random matrix model and  
expectation maximization

Jakob Bramstång



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering  
*Division of Signals Processing and Biomedical Engineering*  
Signal Processing Group  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden 2018

Trajectory Smoothing for Multiple Extended Objects  
Batch estimation based on the random matrix model and expectation maximization  
Jakob Bramstång

© Jakob Bramstång, 2018.

Supervisor: Karl Granström, Department of Electrical Engineering  
Examiner: Karl Granström, Department of Electrical Engineering

Master's Thesis 2018:EX042  
Department of Electrical Engineering  
Division of Signals Processing and Biomedical Engineering  
Signal Processing Group  
Chalmers University of Technology  
SE-412 96 Gothenburg  
Telephone +46 31 772 1000

Cover: Gaussian Inverse-Wishart objects constructed in Matlab showing the probability surface of five objects, their contour plots and corresponding measurements.

Typeset in L<sup>A</sup>T<sub>E</sub>X  
Gothenburg, Sweden 2018

Trajectory Smoothing for Multiple Extended Objects  
Batch estimation based on the random matrix model and expectation maximization  
Jakob Bramstång  
Department of Electrical Engineering  
Chalmers University of Technology

## Abstract

From the combination of the fact that modern sensors get better resolution, and that close range tracking applications, where the objects are close to the sensor, are becoming more and more common, the field of tracking multiple extended objects arises. The property of obtaining multiple measurements per scan violates the classical assumptions. This implies that classical tracking approaches cannot be applied directly.

Object tracking can be performed in many ways, depending on the method that is utilized. In some tracking applications, and in data annotation, Bayesian smoothing is an important tool to infer as much as possible from accumulated data.

This thesis presents two alternative smoothing methods for tracking objects and their spatial extension; the conditional random matrix model and the factorized random matrix model. These models are a natural extension to already existing work in the Bayesian forward filtering framework for extended objects using the so called random matrix model. The linear conditional model is compared to both a linear version and a nonlinear version of the factorized model. The performance of all three models are evaluated.

The models show better robustness to missed detections and few measurements, and yield better results than the existing forward filtering approaches. The linear conditional model and linear factorized model perform very similarly to one another, and outperform the nonlinear factorized model in the case when the ground truth is linear motion. In the case of non-linear ground truth motion, the results are the opposite: the nonlinear factorized model performs better. However, overall the conditional model and the linear factorized model perform better than the nonlinear factorized model, mainly due to the additional approximations needed in the nonlinear factorized model.

Keywords: extended objects, random matrix approach, Gaussian Inverse-Wishart, multiple extended objects, tracking, data association, Bayesian smoothing, Bayesian filtering, expectation maximization.



## Acknowledgements

First of all I would like to thank my supervisor Karl Granström that has guided the work of this thesis with clever ideas and perspective. Secondly my fiance Sofia Starborg who finds me when I am deeply occupied between the shelves of mathematical literature at the library. She has always been supporting me throughout the work. Without these two, this thesis would not have been possible.

Jakob Bramstång, Stockholm, June 2018



# Contents

<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xxi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Object Tracking . . . . .	1
1.2.1 Single Extended Object Tracking . . . . .	1
1.2.2 Multiple Extended Object Tracking . . . . .	2
1.2.3 Smoothing Problems . . . . .	2
1.3 Contribution . . . . .	2
1.4 Related Work . . . . .	3
1.5 Thesis Outline . . . . .	3
<b>2 Background Theory</b>	<b>5</b>
2.1 State Space Modelling . . . . .	5
2.1.1 Transition Model . . . . .	5
2.1.1.1 Constant Velocity Model . . . . .	6
2.1.1.2 Coordinated Turn Model . . . . .	6
2.1.2 Measurement Model . . . . .	7
2.2 Bayesian State Estimation . . . . .	7
2.2.1 Filtering . . . . .	8
2.2.1.1 Prediction . . . . .	8
2.2.1.2 Update . . . . .	8
2.2.2 Smoothing . . . . .	9
2.3 K-means Clustering . . . . .	9
2.4 Expectation Maximisation . . . . .	10
2.5 Kullback Leibler Divergence . . . . .	10
<b>3 Extended Object Tracking</b>	<b>11</b>
3.1 Extended Object Tracking . . . . .	11
3.1.1 Random Matrix Approach . . . . .	11
3.1.2 Model . . . . .	12
3.1.3 Conditional Model . . . . .	12
3.1.4 Factorized Model . . . . .	12
3.1.5 Assumptions . . . . .	13
3.2 Transition Model . . . . .	13

3.2.1	CM . . . . .	14
3.2.2	FM . . . . .	14
3.2.2.1	Linear FM . . . . .	14
3.2.2.2	Nonlinear FM (FMNL) . . . . .	14
3.3	Measurement Model . . . . .	15
3.3.1	Assumptions . . . . .	15
3.3.2	Measurement Likelihood . . . . .	16
3.4	Forward Filtering . . . . .	16
3.4.1	CM . . . . .	17
3.4.2	FM . . . . .	18
3.4.3	FMNL . . . . .	18
3.5	Smoothing . . . . .	18
3.6	Multiple Extended Object Tracking . . . . .	19
3.6.1	Assumptions . . . . .	20
3.6.2	Data Association . . . . .	21
3.7	Multiple Extended Object Tracking cast as an EM-problem . . . . .	21
<b>4</b>	<b>Simulation Scenarios and Performance Evaluation</b>	<b>27</b>
4.1	The Models . . . . .	27
4.2	Extended Object Tracking . . . . .	27
4.2.1	Test Perspectives . . . . .	27
4.2.1.1	Extent . . . . .	28
4.2.1.2	Probability of Detection . . . . .	28
4.2.1.3	Measurement Rate . . . . .	29
4.2.2	Test Scenarios . . . . .	29
4.2.3	Evaluation . . . . .	29
4.2.3.1	Performance Metric . . . . .	29
4.3	Multiple Extended Object Tracking . . . . .	30
4.3.1	Ground Truth . . . . .	31
4.3.2	Scenarios . . . . .	31
<b>5</b>	<b>Results</b>	<b>33</b>
5.1	Extended Object/ Single Object . . . . .	33
5.1.1	First Scenario . . . . .	33
5.1.1.1	Linear Case . . . . .	33
5.1.1.2	Nonlinear Case . . . . .	36
5.1.2	Second Scenario - Extent . . . . .	37
5.1.2.1	Linear Case . . . . .	37
5.1.2.2	Nonlinear Case . . . . .	39
5.1.3	Third Scenario - Probability of Detection . . . . .	39
5.1.3.1	Linear Case . . . . .	40
5.1.3.2	Nonlinear Case . . . . .	41
5.1.4	Fourth Scenario - Probability of Detection . . . . .	43
5.1.4.1	Linear Case . . . . .	43
5.1.4.2	Nonlinear Case . . . . .	46
5.1.5	Fifth Scenario - Measurement Rate . . . . .	49
5.1.5.1	Linear Case . . . . .	50

5.1.5.2	Nonlinear Case . . . . .	51
5.2	Multiple Extended Objects . . . . .	54
5.2.1	First Scenario - Object Distance . . . . .	55
5.2.1.1	Numerical Result . . . . .	55
5.2.1.2	Object Distances . . . . .	57
5.2.2	Second Scenario - Probability of Detection . . . . .	60
5.2.3	Third Scenario - Measurement Rate . . . . .	64
<b>6</b>	<b>Conclusion</b> . . . . .	<b>69</b>
6.1	Single Extended Object . . . . .	69
6.2	Multiple Extended Objects . . . . .	70
6.3	Summary . . . . .	71
6.4	Future Work . . . . .	71
6.4.1	Models . . . . .	71
6.4.2	Computational Tools . . . . .	72
6.4.3	Ground Truth . . . . .	73
6.4.4	Evaluation . . . . .	73
	<b>Bibliography</b> . . . . .	<b>75</b>
<b>A</b>	<b>Appendix 1</b> . . . . .	<b>I</b>
A.1	Matrix Variate Distributions . . . . .	I
A.1.1	Wishart Distribution . . . . .	I
A.1.2	Inverse-Wishart Distribution . . . . .	I
A.1.3	Generalized Beta Distribution type 2 . . . . .	II
A.2	Preliminary Results . . . . .	II
A.3	Derivation of Smoothing Equation for CM and FM . . . . .	III
A.4	Derivation of Smoothing Equation for FMNL . . . . .	IV
<b>B</b>	<b>Appendix 2</b> . . . . .	<b>V</b>
B.1	Plots from the different scenarios in the single object case . . . . .	VI
B.1.1	First scenario with nonlinear true motion . . . . .	VI
B.1.2	Second scenario with nonlinear true motion . . . . .	VII
B.1.3	Third scenario with nonlinear motion . . . . .	VIII
B.1.4	Fourth scenario with linear motion for $P_d = 0.5$ . . . . .	IX
B.1.5	Fourth scenario with linear motion and $P_d = 0.25$ . . . . .	X
B.1.6	Fourth scenario with nonlinear motion and $P_d = 0.5$ . . . . .	XI
B.1.7	Fifth scenario with linear motion and $\lambda = 3$ . . . . .	XII
B.1.8	Fifth scenario with nonlinear motion and $\lambda = 3$ . . . . .	XIII
<b>C</b>	<b>Appendix 3</b> . . . . .	<b>XV</b>
C.1	Results from the different scenarios in the multiple object case . . . . .	XV
C.2	First scenario - Only Distance . . . . .	XV
C.3	Second scenario - Probability of Detection . . . . .	XVI
C.3.1	Constant Extent . . . . .	XVI
C.3.2	Time varying Extent . . . . .	XVII
C.4	Third scenario - Measurement Rate . . . . .	XVIII

## Contents

---

C.4.1	Constant Extent . . . . .	XVIII
C.4.2	Time varying Extent . . . . .	XIX

# List of Figures

2.1	The visualization of the Bayesian forward filtering. The cyan coloured arrows are the prediction to move forward in time by utilizing the transition model. The magenta coloured arrows represents the measurement update based on the obtained measurements $Z_k$ obtained at time $k$ . $\xi_k$ denotes the object state at time $k$ . . . . .	7
4.1	An example ellipse of the extent model and its direction arrow and the definition of the axis names. . . . .	28
4.2	The shape of the true trajectories for the case when $M = 2$ and $\Delta > 10\sigma$	31
5.1	(a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is linear, $P_d = 1$ , $\lambda = 8$ and the extent is constant. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error. . . . .	34
5.2	(a) is a typical run of a randomly generated linear trajectory when $P_d = 1$ , $\lambda = 8$ and the extent is constant. Here it is almost impossible to distinguish the different estimation from each other. The blue is CM, green FM, red FMNL and black is the ground truth. (b) show a zoomed in version of (a), here the filtering and the smoothing can be seen. The filterings are the dashed lines with the same colour coding and the smoothing the solid ones. . . . .	35
5.3	(a) is a typical run of a randomly generated nonlinear trajectory when $P_d = 1$ , $\lambda = 8$ and the extent is constant. Here it is almost impossible to distinguish the different estimation from each other. The blue is CM, green FM, red FMNL and black is the ground truth. (b) show a zoomed in version of (a), here the filtering and the smoothing can be seen. The filterings are the dashed lines with the same colour coding and the smoothing the solid ones. . . . .	36

5.4	(a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is linear, $P_d = 1$ , $\lambda = 8$ and the extent time varying. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error. . . . .	37
5.5	(a) is FMNL and shows the internal error comparison between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (b) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is nonlinear, $P_d = 1$ , $\lambda = 8$ and the extent time varying. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error. . . . .	39
5.6	(a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is linear, $P_d = 0.75$ , $\lambda = 8$ and the extent time varying. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error. . . . .	40
5.7	Here is a typical run for the nonlinear true motion when $P_d = 0.75$ , $\lambda = 8$ and the extent is varying with time. It is zoomed in on a curve where there is no detection made at the third the time instances, counted from the left. The blue colored curves are CM, green color is FM and red is FMNL. The solid lines are the smoothings, the dashed lines are the filtering and the dotted lines are the predictions. The black curves are the ground truth. . . . .	42
5.8	Here are all the three smoothing result compared together for the case of nonlinear true motion and $P_d = 0.75$ , extent = varying and $\lambda = 8$ . The blue solid line is CM, and the dashed blue lines are its corresponding 5- and 95- percentiles. The same goes for the green solid and dashed lines but are for FM and the red curves are FMNL. . . . .	43
5.9	The error comparison between the filtering and smoothing for different values of $P_d$ for linear true motion. FM lies on top of CM. . . . .	44

- 
- 5.10 (a) is a typical run of a randomly generated linear trajectory when  $P_d = 0.5$ ,  $\lambda = 8$  and the extent is time varying. Here it is almost impossible to distinguish the different estimation from each other. The blue is CM, green FM, red FMNL and black is the ground truth. (b) show a zoomed in version of (a), here the filtering and the smoothing can be seen. The filterings are the dashed lines with the same colour coding and the smoothing the solid ones. The trajectories showing here have a series of 4 missed detections in a row, and thus it is possible to see how the prediction moves according to its assumed motion, and the filtering follows the prediction up until there is an detection while the smoothing has a nice estimate just next to the true trajectory. 45
- 5.11 (a) is a typical run of a randomly generated linear trajectory when  $P_d = 0.25$ ,  $\lambda = 8$  and the extent is time varying. The blue is CM, green FM, red FMNL and black is the ground truth. (b) show a zoomed in version of (a), here the forward filtering and the smoothing can be seen. The filterings are the dashed lines with the same colour coding and the smoothing the solid ones. There is a series of 11 missed detections in a row followed by an additional 6. . . . . 45
- 5.12 (a) is a comparison between the smoothings when  $P_d = 0.5$ ,  $\lambda = 8$  and the extent is time varying. (b) is a comparison between the smoothings when  $P_d = 0.25$ ,  $\lambda = 8$  and the extent is time varying. . . 46
- 5.13 The error comparison between the filtering and smoothing for different values of  $P_d$  for nonlinear true motion. . . . . 46
- 5.14 (a) is a typical run of a randomly generated nonlinear trajectory when  $P_d = 0.5$ ,  $\lambda = 8$  and the extent is time varying. The blue is CM, green FM, red FMNL and black is the ground truth. (b) show a zoomed in version of (a), here the filtering and the smoothing can be seen. The filterings are the dashed lines with the same colour coding and the smoothing the solid ones. . . . . 47
- 5.15 (a) is a comparison between GWD estimation error for the smoothing of the three models in the case of nonlinear motion,  $P_d = 0.5$ , time varying extent and  $\lambda = 8$ . (b) is a comparison between GWD estimation error for the smoothing of the three models in the case of nonlinear motion,  $P_d = 0.25$ , time varying extent and  $\lambda = 8$ . The blue curves are CM model, green FM and red FMNL. Solid line are the median error and the dashed lines are the 5- and 95-percentiles. . 48
- 5.16 (a) is a typical run of a randomly generated nonlinear trajectory when  $P_d = 0.25$ ,  $\lambda = 8$  and the extent is time varying. The blue is CM, green FM, red FMNL and black is the ground truth. (b) show a zoomed in version of (a), here the filtering and the smoothing can be seen. The filterings are the dashed lines with the same colour coding and the smoothing the solid ones. . . . . 48

5.17 (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering in magenta, and the black smoothing. Here the true motion is nonlinear,  $P_d = 0.25$ ,  $\lambda = 8$  and the extent is time varying. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error. . . . . 49

5.18 The error comparison between the filtering and smoothing for different values of  $\lambda$  for linear true motion. . . . . 50

5.19 (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering in magenta, and the black smoothing. The true motion is linear,  $P_d = 0.75$ ,  $\lambda = 1$  and the extent time varying. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error. . . 51

5.20 (a) is the comparison between GWD estimation error for the smoothing of the three models in the case of nonlinear motion,  $P_d = 0.75$ , time varying extent and  $\lambda = 3$ . (b) is the comparison between GWD estimation error for the smoothing of the three models in the case of nonlinear motion,  $P_d = 0.75$ , time varying extent and  $\lambda = 1$ . The blue curves are CM model, green FM and red FMNL. Solid line are the median error and the dashed lines are the 5- and 95-percentiles. . . 52

5.21 The error comparison between the filtering and smoothing for different values of  $\lambda$  for nonlinear true motion. . . . . 52

5.22 (a) is FMNL model for nonlinear true motion,  $P_d = 0.75$ ,  $\lambda = 3$  and the extent is varying with time. (b) is FMNL model for nonlinear true motion,  $P_d = 0.75$ ,  $\lambda = 1$  and the extent is varying with time. The black curve is the median GWD estimation error for each time instance based on 100 Monte-Carlo simulations of the smoothing estimate. The black dashed lines are the corresponding 5- and 95-percentiles of the GWD error. The magenta and cyan coloured lines have the same properties as the black ones but are based on the forward filtering estimation error and the prediction estimation error. . . 53

5.23 (a) is CM model for nonlinear true motion,  $P_d = 0.75$ ,  $\lambda = 3$  and the extent is varying with time. (b) is CM model for nonlinear true motion,  $P_d = 0.75$ ,  $\lambda = 1$  and the extent is varying with time. The black curve is the median GWD estimation error for each time instance based on 100 Monte-Carlo simulations of the smoothing estimate. The black dashed lines are the corresponding 5- and 95-percentiles of the GWD error. The magenta and cyan coloured lines have the same properties as the black ones but are based on the forward filtering estimation error and the prediction estimation error. . . . . 54

5.24 (a) is FM model for nonlinear true motion,  $P_d = 0.75$ ,  $\lambda = 3$  and the extent is varying with time. (b) is FM model for nonlinear true motion,  $P_d = 0.75$ ,  $\lambda = 1$  and the extent is varying with time. The black curve is the median GWD estimation error for each time instance based on 100 Monte-Carlo simulations of the smoothing estimate. The black dashed lines are the corresponding 5- and 95-percentiles of the GWD error. The magenta and cyan coloured lines have the same properties as the black ones but are based on the forward filtering estimation error and the prediction estimation error. . . . . 54

5.25 (a) is the comparison between GWD estimation error for the smoothing of the three models in the case of nonlinear motion,  $P_d = 0.75$ , time varying extent and  $\lambda = 3$ . (b) is the comparison between GWD estimation error for the smoothing of the three models in the case of nonlinear motion,  $P_d = 0.75$ , time varying extent and  $\lambda = 1$ . The blue curves are CM model, green FM and red FMNL. Solid line are the median error and the dashed lines are the 5- and 95-percentiles. 55

5.26 The GWD smoothing estimation error for the case of  $\Delta > 10\sigma$ ,  $P_d = 1$ , constant extent and  $\lambda = 8$ . The upper figure is for object  $m = 1$  and the lower figure is for object  $m = 2$ . This case resembles the first scenario in Section 5.1.1, and as expected the performance is good and similar to both of the objects. . . . . 56

5.27 The upper figure is a visualization of Table C.1 where the extents are constant. The lower figure is a visualization of Table C.2 where the extents are time varying. For the case of varying  $\Delta$ ,  $P_d = 1$  and  $\lambda = 8$  58

5.28 (a) is typical run with the case  $\Delta = 4\sigma$ ,  $P_d = 1$ , extent is constant and  $\lambda = 8$ . The black curves are the true trajectory and extent, the blue is CM, green FM and red FMNL. (b) is a zoomed in version of the trajectory in (a). . . . . 58

5.29 (a) is an example situation that might occur when object gets too close to each other. What happens is that they may switch measurements with each other and thus switch trajectory. This is for the case when  $P_d = 1$ ,  $\Delta = \sigma$ , the extent is time varying and  $\lambda = 8$ . In this plot are the extent estimates omitted to make it more visible how a switching of state might look like. (b) is an example when one object, here it occurred for FMNL in red, claims all the measurements and dominates. This is for the case when  $P_d = 1$ ,  $\Delta = \sigma$ , the extent is time varying and  $\lambda = 8$ . . . . . 59

5.30 (a) is the comparison between GWD estimation error for the smoothing of the three models in the case of constant extent,  $P_d = 1$ ,  $\lambda = 8$  and  $\Delta = \frac{1}{3}\sigma$ . (b) is the comparison between GWD estimation error for the smoothing of the three models in the case of time varying extent,  $P_d = 1$ ,  $\lambda = 8$  and  $\Delta = 1\sigma$ . The blue curves are CM model, green FM and red FMNL. Solid line are the median error and the dashed lines are the 5- and 95-percentiles. The error is the sum over all objects. . . . . 61

5.31 A visualization of the three Tables C.3, C.4 and C.5, for different values of the object distance  $\Delta$  and Probability of detection  $P_d$ . CM is blue, FM green and FMNL red. The plot in the upper left corner is for different values on the  $\Delta$  and  $P_d = 1$ . The upper right most figure is different  $\Delta$  and  $P_d = 0.75$ . Lower left most figure is different values on  $\Delta$  and  $P_d = 0.5$  and the lower right figure is varying  $\Delta$  and  $P_d = 0.25$ . This is for constant extent. . . . . 62

5.32 The three models smoothing performance in the case of constant extent,  $P_d = 0.75$ ,  $\lambda = 8$  and  $\Delta = \sigma$ . Blue curves are CM, green FM and red FMNL. Solid lines are the median error per time instance and the dashed lines are the corresponding 5- and 95-percentiles. . . . 63

5.33 A visualization of the three Tables C.6, C.7 and C.8 for different values of the object distance  $\Delta$  and Probability of detection  $P_d$ . CM is blue, FM green and FMNL red. The plot in the upper left corner is for different values on the  $\Delta$  and  $P_d = 1$ . The upper right most figure is different  $\Delta$  and  $P_d = 0.75$ . Lower left most figure is different values on  $\Delta$  and  $P_d = 0.5$  and the lower right figure is varying  $\Delta$  and  $P_d = 0.25$ . This is for time varying extent. . . . . 64

5.34 The three models smoothing performance when the extent is varying with time,  $P_d = 0.75$ ,  $\Delta = 4\sigma$  and  $\lambda = 8$ . The solid lines are the median error over time, and the dashed lines are the corresponding 5- and 95-percentiles. Blue is CM, green FM and red FMNL. . . . . 65

5.35 The visualization of the three Tables C.9, C.10 and C.11. With  $P_d = 1$ , constant extent and for different values of  $\Delta$  and  $\lambda$ . The upper left figure is  $\lambda = 8$ , upper right  $\lambda = 3$  and lower figure  $\lambda = 1$ . . . . . 66

5.36 The visualization of the three Tables C.12, C.13 and C.14 with  $P_d = 1$ , time varying extent and for different values of  $\Delta$  and  $\lambda$ . The upper left figure is  $\lambda = 8$ , upper right  $\lambda = 3$  and lower figure  $\lambda = 1$ . . . . . 66

5.37 The comparison of all three models in the case of constant extents at the boundary case when  $P_d = 1$ ,  $\Delta = \sigma$  and  $\lambda = 3$ . Blue curve are CM, green FM and red FMNL. The solid lines are the median error at each time instance and the dashed lines are the corresponding 5- and 95-percentiles. . . . . 67

5.38 The comparison of all three models in the case of time varying extents at the boundary case when  $P_d = 1$ ,  $\Delta = 3\sigma$  and  $\lambda = 3$ . Blue curve are CM, green FM and red FMNL. The solid lines are the median error at each time instance and the dashed lines are the corresponding 5- and 95-percentiles. . . . . 68

- 
- B.1 (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is nonlinear and  $P_d = 1$ ,  $\lambda = 8$  and the extent is constant. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error. . . . . VI
- B.2 (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is nonlinear and  $P_d = 1$ ,  $\lambda = 8$  and the time varying extent. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error. . . . . VII
- B.3 (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is nonlinear and  $P_d = .75$ ,  $\lambda = 8$  and the time varying extent. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error. . . . . VIII
- B.4 (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is linear and  $P_d = .5$ ,  $\lambda = 8$  and the time varying extent. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error. . . . . IX
- B.5 (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is linear and  $P_d = .25$ ,  $\lambda = 8$  and the time varying extent. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error. . . . . X

B.6 (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is nonlinear and  $P_d = .5$ ,  $\lambda = 8$  and the time varying extent. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error. . . . . XI

B.7 (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is linear and  $P_d = .75$ ,  $\lambda = 3$  and the time varying extent. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error. . . . . XII

# List of Tables

3.1	The Forward filtering algorithm for CM. . . . .	17
3.2	The Forward filtering algorithm for FM. . . . .	19
3.3	The prediction algorithm for FMNL. . . . .	20
3.4	The filtering algorithm for FMNL. . . . .	21
3.5	The backward smoothing algorithm for CM. . . . .	22
3.6	The backward smoothing algorithm for FM. . . . .	23
3.7	The backward smoothing algorithm for FMNL. . . . .	24
3.8	The backward smoothing algorithm for FMNL. . . . .	25
5.1	The numerical calculation based on (4.3) in the case of linear true motion, $P_d = 1$ , the extent is constant (denoted ext = const) and $\lambda = 8$ . . . . .	34
5.2	The numerical calculation based on (4.3) for the case of nonlinear true motion, $P_d = 1$ , the extent is constant and $\lambda = 8$ . . . . .	36
5.3	The numerical calculation based on (4.3) for the case of linear true motion, $P_d = 1$ , the extent is time varying and $\lambda = 8$ . . . . .	38
5.4	The numerical calculation based on (4.3) for the case of nonlinear true motion, $P_d = 1$ , the extent is changing over time and $\lambda = 8$ . . . . .	39
5.5	The numerical calculation based on (4.3) for the case of linear true motion, $P_d = 0.75$ , the extent is changing over time and $\lambda = 8$ . . . . .	41
5.6	The numerical calculation based on (4.3) for the case of nonlinear true motion, $P_d = 0.75$ , the extent is changing over time and $\lambda = 8$ . . . . .	41
5.7	The numerical results for the case of $\Delta = 10\sigma$ , $P_d = 1$ and $\lambda = 8$ and the extent is constant. . . . .	55
5.8	The numerical results for the case of $\Delta > 10\sigma$ , $P_d = 1$ , extent is constant and $\lambda = 8$ with the omitted first time instances that created a bias error. . . . .	57
5.9	The Probability of getting no measurements for different values of $P_d$ and $\lambda$ , based on $P(\text{No measurements}) = (1 - P_d)p_{n_k} + (1 - P_d)(1 - p_{n_k}) + P_d p_{n_k}$ . . . . .	65
B.1	The numerical calculation based on (4.3) for the GWD estimation error in the case of nonlinear true motion, $P_d = 0.5$ , the extent is changing over time (denoted ext = varying) and measurement rate $\lambda = 8$ . . . . .	IX

B.2 The numerical calculation based on (4.3) for the GWD estimation error in the case of nonlinear true motion,  $P_d = 0.25$ , the extent is changing over time (denoted ext = varying) and measurement rate  $\lambda = 8$ . . . . . X

B.3 The numerical calculation based on (4.3) for the case of nonlinear true motion,  $P_d = 0.5$ , the extent is changing over time and  $\lambda = 8$ . . . XI

B.4 The numerical calculation based on (4.3) for the case of nonlinear true motion,  $P_d = 0.25$ , the extent is changing over time and  $\lambda = 8$ . . XII

B.5 The numerical calculation based on (4.3) for the case of linear true motion,  $P_d = 0.75$ , the extent is changing over time and  $\lambda = 3$ . . . . XIII

B.6 The numerical calculation based on (4.3) for the case of linear true motion,  $P_d = 0.75$ , the extent is changing over time and  $\lambda = 1$ . . . . XIII

B.7 The numerical calculation based on (4.3) for the case of nonlinear true motion,  $P_d = 0.75$ , the extent is changing over time and  $\lambda = 3$ . . XIII

B.8 The numerical calculation based on (4.3) for the case of nonlinear true motion,  $P_d = 0.75$ , the extent is changing over time and  $\lambda = 1$ . . XIII

C.1 The numerical errors for the models for different distances  $\Delta$ , with  $P_d = 1$ , extent is constant and  $\lambda = 8$ . . . . . XV

C.2 The numerical errors for the models for different distances  $\Delta$ , with  $P_d = 1$ , extent is time varying and  $\lambda = 8$ . . . . . XV

C.3 The numerical errors for CM model for different probability of detection and distances  $\Delta$ , with  $P_d$ , extent is constant and  $\lambda = 8$ . . . . . XVI

C.4 The numerical errors for FM model for different probability of detection and distances  $\Delta$ , with  $P_d$ , extent is constant and  $\lambda = 8$ . . . . . XVI

C.5 The numerical errors for FMNL model for different probability of detection and distances  $\Delta$ , with  $P_d$ , extent is constant and  $\lambda = 8$ . . . XVI

C.6 The numerical errors for CM model for different probability of detection and distances  $\Delta$ , with  $P_d$ , extent is time varying and  $\lambda = 8$ . . . XVII

C.7 The numerical errors for FM model for different probability of detection and distances  $\Delta$ , with  $P_d$ , extent is time varying and  $\lambda = 8$ . . . XVII

C.8 The numerical errors for FMNL model for different probability of detection and distances  $\Delta$ , with  $P_d$ , extent is time varying and  $\lambda = 8$ . XVII

C.9 The numerical result for CM when the extent is constant and  $P_d = 1$  for different values of the object distance  $\Delta$  and the measurement rate  $\lambda$ . . . . . XVIII

C.10 The numerical result for FM when the extent is constant and  $P_d = 1$  for different values of the object distance  $\Delta$  and the measurement rate  $\lambda$ . . . . . XVIII

C.11 The numerical result for FMNL when the extent is constant and  $P_d = 1$  for different values of the object distance  $\Delta$  and the measurement rate  $\lambda$ . . . . . XVIII

C.12 The numerical result for CM when the extent is varying with time and  $P_d = 1$  for different values of the object distance  $\Delta$  and the measurement rate  $\lambda$ . . . . . XIX

C.13	The numerical result for FM when the extent is varying with time and $P_d = 1$ for different values of the object distance $\Delta$ and the measurement rate $\lambda$ . . . . .	XIX
C.14	The numerical result for FMNL when the extent is varying with time and $P_d = 1$ for different values of the object distance $\Delta$ and the measurement rate $\lambda$ . . . . .	XIX



# 1

## Introduction

This chapter gives a short background to the research field of tracking and motivates the work done in this thesis. It addresses relevant works and presents a formal problem description.

### 1.1 Motivation

Estimation and tracking have a wide application area. One common estimation application is tracking with the usage of radar. With the help of electromagnetic waves and tracking algorithms, the position and velocity of objects are being tracked.

The conventional tracking algorithms that are utilized rely on some assumptions. One of the most common is the assumption of *point object*. The point object assumption means that we do not model the spatial extent of the object that is being tracked. In other words no matter the size, all the object's attributes are concentrated to a single point, similar to the point particle assumptions made in classical Newtonian mechanics [39].

The number of measurements obtained from the sensor depends mainly on three factors; the sensor resolution, size of the object and the distance between the sensor and the object. The point object assumption implies that there is, at each time instance, at most one measurement per object. The assumption was valid at the time when the first successful tracking applications were invented, such as the *Kalman filter* developed in the middle of the 20th century by Rudolf E. Kálmán [38].

### 1.2 Object Tracking

Object tracking is in this thesis divided into two parts: tracking of a single extended object and tracking multiple objects. This is to tackle the problem from different levels of abstractions.

#### 1.2.1 Single Extended Object Tracking

Technical development has improved the quality of the sensors that typically are used for tracking, and it has also reduced the cost of the sensors. In contrast to old sensors, modern ones have better resolution and thus have the ability to get multiple data readings per object at each time instance [33]. This violates the assumption that there is only one data point at each time instance. Multiple measurements

makes it possible to model the spatial extent of the object and this means that the classical point object tracking approaches cannot be applied directly.

Some typical tracking applications today are airspace surveillance and position localization from a satellite (GPS). Both of these are usually long range tracking where the objects are far away from the sensor, and thus the point object assumption is still valid.

Situations when the point assumption is ungainly are when the extent of the objects contain valuable information that otherwise would be disregarded. For instance if the estimate is used to determine the maneuvering actions of a vehicle in a tight traffic situation, then the size is of utmost importance. The *environment perception* of vehicles, especially autonomous vehicles, are a typical example of a close range tracking application where the distances between the sensor and the objects are small in relation to the sensor resolution.

### 1.2.2 Multiple Extended Object Tracking

For the vehicle example above, the ability to avoid collisions and fatal accidents requires knowledge of both the positions of the surrounding objects and their sizes. Besides, there is seldom only a single vehicle on the road, but rather multiple ones.

If there are several objects where the object size is of interest and there is a possibility of obtaining multiple measurements per object, then the problem is a tracking of *multiple extended objects*. In other words, multiple object tracking is the problem of estimating both the kinematic attributes and the extent of several objects.

### 1.2.3 Smoothing Problems

When the application is not a real time online application, and there is an interest in extracting as much information as possible from accumulated data the Bayesian filtering technique known as *smoothing* is desirable. Smoothing applications are typical in surveillance, but can also be of great interest in data annotations. For instance when creating training data for a machine learning algorithm.

## 1.3 Contribution

This thesis makes contributions to both single extended object smoothing, and multiple extended object smoothing, under the assumption of *random matrix model* or a *Gaussian-Inverse Wishart* distribution. Firstly there is a theoretical derivation of two different smoothing models, a *conditional random matrix model* and a *factorized random matrix model*. Both models are then evaluated in a simulated environment in the case of only a single extended object and for different levels of difficulty. Thereafter both models are also evaluated for the case of tracking multiple extended objects. The contribution acts as an intuitive extension of existing work in a Bayesian forward filtering framework, where the so called random matrix approach is utilized, see, e.g. [25, 42, 9, 9, 9, 30].

## 1.4 Related Work

In the field of multiple extended object tracking there is a lot of ongoing research. For the extended object tracking field there is an elaborate overview presented in [15]. According to this article there are two main approaches to model extents. The first is to use simple geometric shapes such as straight lines [17], rectangles [11, 20], circles [23] and ellipses [18, 25, 9]. The other approach is that the extent may be modeled with more complex shapes, either as a parametric curve [2, 7, 22] or as a combination of ellipses [14, 27].

For Bayesian filtering of extended objects, the random matrix approach was first introduced in [25]. This work was developed further by several other researchers such as in [8]. An new update step for the Bayesian random matrix framework is developed in [9]. The update step was then further developed in [33]. A new prediction step was presented in [19]. In [41] the fusion of multiple sensors are introduced to the Bayesian Random matrix framework and in [40] a comparison of the existing multiple measurement approaches is done. Further in [13] the possibility of spawning and combinations of objects are introduced in the Bayesian random matrix framework.

The next step was to implement the extended object tracking work for the case of several objects. A theoretically elaborate article about the field is presented in [30]. Multiple object tracking based on the random matrix approach was developed by [42] and further related works were presented, such as [12].

When it comes to the multiple object case there are different approaches to cope with the association problem. Such as the *Probability hypothesis density filter (PHD)* in [12, 30], or the *Maximum Likelihood-Probabilistic Data Association (MLPDA)* in [5], the *Probabilistic Multi-Hypothesis Tracking (PMHT)* in [26, 42, 43]. Other filters for multiple object tracking are *cardinalized probability hypothesis density (CPHD)* in [45],  *$\delta$ -Generalized Labeled Multi Bernoulli ( $d$ -GLMB)* filter in [34] and *Poisson multi-Bernoulli mixture (PMBM)* filter in [16].

The EM-algorithm used in this thesis, is a general method for finding maximum likelihood estimates and has many applications. In some of the related works to this thesis the EM algorithm have been used for informational extraction in different tracking applications. For example in [23] where the EM is used to track the contour of an object with EM based on noisy point clouds. In [24] the EM algorithm is compared to a Gaussian Random Hyper Surface Model in a tracking situation where the extent is modeled as a star-convex shape. Other attributes, of the objects, rather than extents are estimated with the EM algorithm in [6]. EM is also used in [29] for a different aspect of tracking; to solve the association problem while estimating a map based on radar readings.

## 1.5 Thesis Outline

The rest of the thesis is structured in the following way: chapter 2 is the general tracking theory chapter where all the essential theory needed for understanding general tracking is presented. Chapter 3 gives the specific tracking theory needed to

understand the contents of the report and it assumes some background knowledge in the field of tracking in a Bayesian framework. The method of the thesis is presented in chapter 4. It describes the experiments and how the evaluations are structured. Chapter 5 presents the obtained results from the experiments together with the corresponding analyses. The sixth and last chapter contains the conclusion of the presented work and ends with a discussion about possible future work.

# 2

## Background Theory

This chapter gives a brief overview of general Bayesian filtering and smoothing. Here the transition and measurement models, that will be used in this thesis, are presented. The clustering algorithms utilized to solve the association problem and the Kullback Leibler Divergence are also presented.

### 2.1 State Space Modelling

The target *state* can be viewed as the unknown parameters that are of interest for an object. When tracking an object, one usually collects parameters, such as position and velocity, in a state vector  $x_k$ , where  $k$  denotes the time instance. In Bayesian probability the unknown parameters are modelled as random and in classical tracking approaches it is common to model the distribution of the unknown state as a Gaussian distribution,

$$p(x_k) = \mathcal{N}(x_k; m_{k|l}, P_{k|l}), \quad (2.1)$$

where  $m_{k|l}$  and  $P_{k|l}$  denotes the mean and covariance of the state at time  $k$  given information up until time  $l$ .

Since tracking consists of the problem of estimating the states of objects over time, the relationship between states from different time instances is of interest. Similarly the relation between the gathered data about the objects and the objects' states must also be considered. This is done by *models* or more specifically, *state space models*.

Models are used as an abstraction to explain relationships between different quantities. Hence the concept of models is huge. In this thesis the relationship between states over time are modeled with *transition models* and the relationship between the states and obtained data are modeled with *measurement models*.

#### 2.1.1 Transition Model

A transition model is used to model how the state evolves in time. There exists both linear and nonlinear transition models, and the choice of model depends on the type of time evolution that is desirable for the state. The two different transition models that are used in this thesis are the linear *constant velocity model (CV)* and the nonlinear *coordinated turn model (CT)*, see, e.g., [38, Chapter 4] and [36]. A common way to mathematically write a general transition model is

$$x_k = f_{k-1}(x_{k-1}) + q_{k-1}, \quad (2.2)$$

where  $f_{k-1}$  is the transition function of the state and  $p(q_{k-1}) = \mathcal{N}(\mathbf{0}, Q_{k-1})$  is the process noise and is assumed to be a Wiener Process increment, in other words white Gaussian noise.

### 2.1.1.1 Constant Velocity Model

The discrete constant velocity model is a linear kinematic motion model where the position update is based on a linear dependency of the velocity and the velocity is modeled as a *random walk*,

$$v_{k+1|k} = v_{k|k} + q_k, \quad (2.3)$$

where  $p(q_k) = \mathcal{N}(0, \sigma_v^2)$  is the process noise.

The kinematic state vector has the form  $x_k = [\mathbf{r}_k \quad \dot{\mathbf{r}}_k]$ . Where  $\mathbf{r}_k$  denotes the Cartesian position and  $\dot{\mathbf{r}}_k$  the velocity. Since the problem considered in this thesis is in 2D, the dimension of each component is  $\mathbf{r}_k, \dot{\mathbf{r}}_k \in \mathbb{R}^d$ , where  $d = 2$ . The CV model can be written on the form

$$x_{k+1|k} = F_k x_{k|k} + q_k, \quad (2.4)$$

where  $F_k$  is the transition matrix, and  $p(q_k) = \mathcal{N}(\mathbf{0}, Q_k)$  is once again the process noise and is assumed to be a Wiener Process increment with covariance  $Q_k$ . If we denote the time increment between two time steps by  $h$  and the variance of the velocity as  $\sigma_v^2$ , then the one dimensional discrete CV transition and corresponding process noise covariance matrices can be written as

$$F_k = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix}, \quad Q_k = \begin{bmatrix} 0 & 0 \\ 0 & h\sigma_v^2 \end{bmatrix}. \quad (2.5)$$

### 2.1.1.2 Coordinated Turn Model

The coordinated turn model is a nonlinear motion model. The kinematic state vector in the CT model has the form  $x_k = [r_k^x \quad r_k^y \quad v_k \quad \theta_k \quad \omega_k]$ , where  $r_k^x$  and  $r_k^y$  are the Cartesian position coordinates at time  $k$ ,  $v_k$  is the speed scalar,  $\theta_k$  is the heading and  $\omega_k$  is the turn-rate. The CT model takes the form

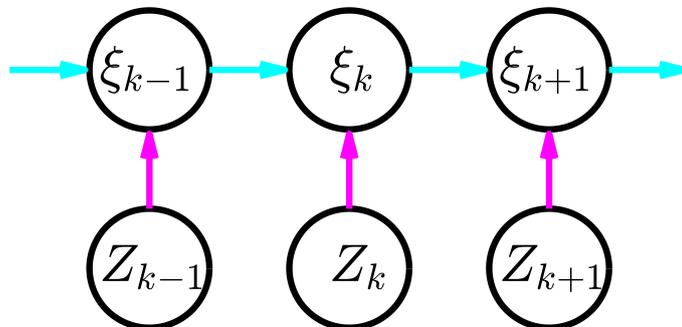
$$x_{k+1} = f(x_k) + q_k, \quad (2.6)$$

where  $p(q_k) = \mathcal{N}(\mathbf{0}, Q_k)$  is the process noise with the corresponding Wiener assumption as above. The transition function and noise matrix for the CT model are defined as

$$x_{k+1} = f_k(x_k) = \begin{bmatrix} r_{k+1}^x \\ r_{k+1}^y \\ v_{k+1} \\ \theta_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} r_k^x + hv_k \cos(\theta_k) \\ r_k^y + hv_k \sin(\theta_k) \\ v_k \\ \theta_k + h\omega_k \\ \omega_k \end{bmatrix}, \quad Q_k = \text{diag}([0 \quad 0 \quad h\sigma_v^2 \quad 0 \quad h\sigma_\omega^2]) \quad (2.7)$$

Here  $\sigma_v^2$  is the velocity variance and  $\sigma_\omega^2$  is the turn-rate variance.

## Forward filtering



**Figure 2.1:** The visualization of the Bayesian forward filtering. The cyan coloured arrows are the prediction to move forward in time by utilizing the transition model. The magenta coloured arrows represents the measurement update based on the obtained measurements  $Z_k$  obtained at time  $k$ .  $\xi_k$  denotes the object state at time  $k$ .

### 2.1.2 Measurement Model

The measurement model is used to model the relationship between the state and the measurement. A general measurement model is usually mathematically denoted as

$$z_k = h_k(x_k) + \gamma_k, \quad (2.8)$$

where  $p(\gamma_k) = \mathcal{N}(\mathbf{0}, R_k)$  is the measurement noise and is also assumed to be a Wiener Process increment with covariance  $R_k$ . A common linear measurement model where  $h_k : x_k \mapsto H_k x_k$  that only measures the position is

$$z_k = H_k x_k + \gamma_k, \quad (2.9)$$

where  $H_k$  is the measurement matrix that extracts the position form the state.

## 2.2 Bayesian State Estimation

When the Bayesian inference framework is applied to tracking problems, such as the classical Kalman filter, it consists of a *prediction* step, followed by a *filtering* step. This is referred to as *forward filtering* or *filtering*, see Figure 2.1. When forward filtering has been performed for a batch of data a *backwards smoothing* step can be performed. This is done so that each estimate is based on as much information as possible.

To be able to perform the prediction, filtering and smoothing steps there is a need for both transition and measurement models. When performing forward prediction and backward smoothing, the calculation is based on a dynamical transition model of the state and the filtering uses the measurement model to merge the new information to the old.

### 2.2.1 Filtering

Here the general Bayesian forward filtering equations are given together with the prediction and filtering steps obtained by solving the equations in the linear case. The prediction and filtering steps given here are the ones in a classical Kalman filter [38].

#### 2.2.1.1 Prediction

The prediction step works as a connection over time and utilizes a *transition* or *dynamical motion model* to make a prediction into the future based on the current calculated estimate. Definition of the transition models used in this thesis can be viewed in Section 2.1.1. A transition model can be seen as a function that maps a state estimate at one time instance to the next, i.e  $f : x_{k-1} \mapsto f(x_{k-1})$ . The prediction can mathematically be expressed as the solution to the *Chapman-Kolmogorov equation*, see, e.g., [38],

$$p(x_k|Z^{k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|Z^{k-1})dx_{k-1}, \quad (2.10)$$

where  $Z^{k-1}$  denotes all the accumulated information up until time instance  $k - 1$ . Given the linear transition model in (2.4) the solution to the Chapman-Kolmogorov equation gives the following linear prediction equations for the moments of the Gaussian distributed state,

$$m_{k|k-1} = F_k m_{k|k}, \quad (2.11)$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k, \quad (2.12)$$

where  $x_{k|k-1}$  and  $P_{k|k-1}$  are the mean vector and covariance matrix of the state vector at time  $k$  given information up until time instance  $k - 1$ . Here  $Q_k$  is the covariance matrix of the process noise. Similar update equations for the nonlinear transition model in (2.7) can be obtained with a similar approach, but then some approximation are needed, such as linearization, see, e.g., [38].

#### 2.2.1.2 Update

The filtering step combines the prediction with the latest measurements to yield a better estimate by using a measurement model and *Bayes rule*, see, e.g., [32],

$$p(x_k|Z^k) = \frac{p(z_k|x_k, Z^{k-1})p(x_k|Z^{k-1})}{p(z_k|Z^{k-1})}, \quad (2.13)$$

where  $z_k$  is the new measurement,  $Z^k$  is the accumulated measurement up until time  $k$ . With the linear measurement model given in (2.9) along with the Gaussian

distribution assumption of the state, the following linear measurement update for the moment of the state is obtained

$$m_{k|k} = m_{k|k-1} + K_k s_k, \quad (2.14)$$

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^T, \quad (2.15)$$

$$K_k = P_{k|k-1} H^T S_k^{-1}, \quad (2.16)$$

$$s_k = z_k - H_k x_{k|k-1}, \quad (2.17)$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k. \quad (2.18)$$

## 2.2.2 Smoothing

The smoothing makes a step similar to the prediction, but backwards in time. This enables the estimates at each time instance to rely on as much information as possible. The general backward smoothing is calculated according to

$$p(x_k | Z^K) = p(x_k | Z^k) \int \frac{p(x_{k+1} | x_k) p(x_{k+1} | Z^K)}{p(x_{k+1} | Z^k)} dx_{k+1} \quad (2.19)$$

By using the linear transition model in (2.4) the following smoothing update equations are obtained,

$$G_k = P_{k|k} F_k^T P_{k+1|k}^{-1} \quad (2.20)$$

$$m_{k|K} = m_{k|k} + G_k (m_{k+1|K} - m_{k+1|k}) \quad (2.21)$$

$$P_{k|K} = P_{k|k} - G_k (P_{k+1|k} - P_{k+1|K}) G_k^T \quad (2.22)$$

This equations are the ones in a *Rauch-Tung-Striebel Smoother*, see e.g., [38].

## 2.3 K-means Clustering

The *K-means algorithm* is a deterministic clustering technique that generates the clusters based on a distance metric. The whole idea boils down to a two step iterative algorithm that in its first step associates the data points to the closest cluster center, then in the second step the cluster centers are updated based on their assigned data points. This is then repeated until convergence or some termination criterion is fulfilled. We denote  $z_k^r$  as the data point  $r$  at time instance  $k$ . Binary association variables are defined as

$$a_{rm} = \begin{cases} 1, & \text{If data point } r \text{ is assigned to cluster } m \\ 0, & \text{Otherwise} \end{cases} \quad (2.23)$$

The variable  $\mu_m$  is the center of cluster  $m$  and is calculated based on its assigned data points. By using a Euclidean metric, the distortion measure, or object function, that is optimized can be written as

$$J = \sum_{r=1}^{n_k} \sum_{m=1}^M a_{rm} \|z_k^r - \mu_m\|^2 \quad (2.24)$$

In other words, the goal is to minimize  $J$  with respect to  $a_{rm}$  (the first step) and  $\mu_m$  (the second step). Here  $n_k$  is the total number of obtained data points at time  $k$ , and  $M$  is the total number of clusters (or objects).

## 2.4 Expectation Maximisation

In contrast to K-means algorithm, which is a deterministic clustering technique, the *Expectation-Maximization algorithm (EM)* is a probabilistic inference technique that can be used as a probabilistic clustering technique. The difference between the two clustering techniques is that the EM uses a probabilistic approach for the associations, in this situation it can be viewed as an probabilistic extension of the K-means algorithm.

The EM algorithm is a general technique for finding maximum likelihood solutions. In the EM framework the problem is solved by introducing the *hidden variables* or *latent variables*. The introduction of the latent variables are to create a complete likelihood that is assumed to be easier to optimize than the incomplete likelihood (without the latent variables),

$$p(\mathbf{Z}|\mathcal{X}) = \sum_A p(\mathbf{Z}, A|\mathcal{X}), \quad (2.25)$$

where  $\mathbf{Z}$  denotes all the accumulated data,  $\mathcal{X}$  are a set of parameters and  $A$  is the set of all latent variables. Here  $p(\mathbf{Z}, A|\mathcal{X})$  is the complete likelihood and  $p(\mathbf{Z}|\mathcal{X})$  denotes the incomplete likelihood that is obtained by marginalizing over the latent variables.

Just like the K-means algorithm, the EM algorithm is an iterative algorithm consisting of two different steps. The first is the *Expectation step (E-step)* where the optimization of the joint likelihood is done based on the latent variables  $A$ , and since they are unknown, the step has the structure of an expectation. The second step is the *Maximization step (M-step)* where the optimization is based on the known parameters  $\mathcal{X}$ . What the parameters and the latent variables are, depends on the applications.

## 2.5 Kullback Leibler Divergence

The derivation of the smoothing algorithms, requires distribution approximations. They are based on minimization of what is known as the *Kullback-Leibler divergence (KL)*. KL is a measure of how close two distributions are. It is mentioned here for completeness of the theory that has been utilized. For continuous densities, the KL is defined as

$$\text{KL}(p||q) = - \int p(x) \ln \left\{ \frac{q(x)}{p(x)} \right\} dx \quad (2.26)$$

It can be shown that  $\text{KL}(p||q) \geq 0$  and that equality holds if, and only if,  $p(x) = q(x)$ . Note that the Kullback-Leibler divergence is not a symmetrical quantity, i.e.  $\text{KL}(p||q) \neq \text{KL}(q||p)$ . For further properties see, e.g., [4].

# 3

## Extended Object Tracking

This chapter goes through the theory that is needed to understand the tracking research field and the contributions of this thesis. It gives a definition of extended objects and how tracking of extended objects is implemented in a Bayesian framework.

### 3.1 Extended Object Tracking

The difference between the Bayesian forward filtering and the Bayesian smoothing is that the filtering computes the estimates based on current and historical data while the smoothing estimates the states using both previous and future data. Thus, the smoothing uses more information at each estimate [38].

In the presence of difficulties, such as missed detections, smoothing should thus yield better results. The true extent might either be rigid and only change in orientation, but for example in the case of *group tracking* the shape might also vary with time. Group target tracking arises when the sensor resolution can not distinguish between individual targets that move collectively in a group. Thus it is possible to consider them as one large object with multiple measurements instead of considering the internal data association problem [15].

Since the spatial extent of the object are of interests, it needs to be estimated in some fashion. This is done using a model of the extent. As noted in Section 1.4 there are different models to choose among. The model used in this thesis is presented next.

#### 3.1.1 Random Matrix Approach

In this work the so called *Random Matrix approach* (*RM approach*) is utilized, [25, 9]. It models the extent of the object as ellipses. It might seem like a crude model to assume elliptic shapes for the extent, especially in a typical tracking example such as vehicle tracking. Elliptical vehicles are not that common, but the assumption is appropriate in other applications, such as naval surveillance and tracking of groups consisting of several smaller objects [15].

Ellipses can be represented as *symmetric positive definite* (*SPD*) matrices  $X \in \mathbb{R}^{d \times d}$ . The tracking in this thesis is in 2D, thus the matrix dimension is  $d = 2$ . There exist many different matrix distributions that could be utilized, but since the aim is to incorporate the extent into a Bayesian framework, the *Inverse Wishart* (*IW*) distribution is beneficial since it is the *conjugate prior* for a Gaussian measurement

likelihood with unknown covariance matrix [9, 4]. By using this model the extent becomes an additional state  $X_k^j$  to be estimated along with the kinematic state  $x_k^j$ . Note that in contrast to the kinematic state, the extent state matrix entries lack a direct physical interpretation and must be understood in a context.

#### 3.1.2 Model

At every time instance  $k \in 0, \dots, N$  the estimated information about an object  $j \in 1, \dots, M$  is contained in the tuple state  $\xi_k^j = \{x_k^j, X_k^j\}$ , where  $x_k^j \in \mathbb{R}^n$  is referred to as the *kinematic state* and  $X_k^j \in \mathbb{R}^{d \times d}$  the *extent state*. Here  $N$  is the total time instances in the batch of data,  $M$  is the total number of objects present in the batch,  $n$  is the size of the kinematic state and  $d$  is the dimension of the square extent state matrix. The density of the unknown states can mathematically be written as

$$p(\xi_k^j | Z^l) = p(x_k^j, X_k^j | Z^l) = p(x_k^j | X_k^j, Z^l) p(X_k^j | Z^l). \quad (3.1)$$

The state tuple is assumed to be distributed according to a *Gaussian-Inverse Wishart (GIW)* distribution. This particular choice of model is the conjugate prior of a multivariate Gaussian distribution with unknown mean and covariance matrix. Since a common assumption is that the measurement likelihood is Gaussian, the GIW model is very useful.

The great benefit of this property is that the forward filtering performed in a Bayesian framework then preserves the densities and thus reduces the amount of approximation needed. Besides this, the assumption enables the possibility to use the well known Bayesian framework of Kalman filter and Rauch-Tung-Striebel smoothing for the kinematic state [38].

The model in (3.1) gives some possibilities. By utilizing different assumptions, several different models can be obtained. Here two different models will be presented, denoted the *Conditional Model (CM)* and *Factorized Model (FM)*.

#### 3.1.3 Conditional Model

The so called conditional model was first defined in [25], for CM the following state density is used,

$$\begin{aligned} p(\xi_k^j | Z^l) &= p(x_k^j, X_k^j | Z^l) = p(x_k^j | X_k^j, Z^l) p(X_k^j | Z^l) \\ &= \mathcal{N}(x_k^j; m_{k|l}^j, P_{k|l}^j \otimes X_k^j) \times \mathcal{IW}_d(X_k^j; v_{k|l}^j, V_{k|l}^j). \end{aligned} \quad (3.2)$$

Here  $\otimes$  is the matrix Kronecker product, see, e.g., [28] and [35] for a more thorough view of the theory. This yields a linear dependency for the kinematic covariance matrix on the extent state.

#### 3.1.4 Factorized Model

The factorized model assumes that the distribution of the kinematic and extent state are independent, i.e.  $p(x_k^j | X_k^j, Z^l) = p(x_k^j | Z^l)$  and  $\Sigma_{k|l}^j = P_{k|l}^j$ . For FM the following density is used,

$$\begin{aligned}
 p(\xi_k^j | Z^l) &= p(x_k^j, X_k^j | Z^l) = p(x_k^j | X_k^j, Z^l) p(X_k^j | Z^l) \\
 &\approx p(x_k^j | Z^l) p(X_k^j | Z^l) \\
 &= \mathcal{N}(x_k^j; m_{k|l}^j, P_{k|l}^j) \times \mathcal{IW}_d(X_k^j; v_{k|l}^j, V_{k|l}^j).
 \end{aligned} \tag{3.3}$$

### 3.1.5 Assumptions

The following assumptions are made in [25]. Here the superscript  $j$  is omitted since the assumptions and following models hold for all objects.

**Assumption 1** The time evolution of the kinematic state is a first order Markov process,

$$p(x_{k+1} | X_{k+1}, x_k, X_k) = p(x_{k+1} | X_{k+1}, x_k). \tag{3.4}$$

**Assumption 2** The time evolution of the extent state is assumed to be independent of the kinematic state,

$$p(X_{k+1} | x_k, X_k) = p(X_{k+1} | X_k). \tag{3.5}$$

**Assumption 3** The extent state changes slowly with time,

$$X_{k+1} \approx X_k, \tag{3.6}$$

such that for the kinematic state, conditioned on the extent state, the following holds,

$$p(x_{k+1} | X_{k+1}) \approx p(x_{k+1} | X_k), \tag{3.7}$$

$$p(x_k | X_k) \approx p(x_k | X_{k+1}). \tag{3.8}$$

The validity of the above assumptions 2 and 3 are discussed in [25], but are mainly used for mathematical convenience. The following assumption is for FM and was mentioned previously in Section 3.1.4 but is stated here for completeness.

**Assumption 4** The time evolution of the kinematic state is independent of the extent state,

$$p(x_{k+1} | X_{k+1}, x_k) = p(x_{k+1} | x_k). \tag{3.9}$$

## 3.2 Transition Model

As stated in Section 2.1.1 the transition model is used to model the states evolution over time. Here the transition models used for the conditional and factorized model is presented.

### 3.2.1 CM

The transition density is modelled as Gaussian-Wishart,

$$\begin{aligned} & p(x_{k+1}|X_{k+1}, x_k)p(X_{k+1}|X_k) \\ &= \mathcal{N}(x_{k+1}; (F_k \otimes \mathbf{I}_d)x_k, Q_k \otimes X_{k+1}) \\ & \times \mathcal{W}_d(X_{k+1}; \delta_k, X_k/\delta_k) \end{aligned} \quad (3.10)$$

where  $Q_k$  and  $F_k$  are defined as in (2.5) and the matrix  $\mathbf{I}_d$  is the identity matrix of size  $d \times d$ .

The transition model for the kinematic state, due to the conditioning on the extent state in (3.2), must be a linear motion model [25]. The motion model used in this thesis is the CV model defined in Section 2.1.1.1.

The time evolution of the extent state is modeled as a matrix generalization to the random walk used for the velocity in (2.3). In other words, it is expected that the extent does not change much over time. The transition density has the form of a Wishart density as can be seen in (3.10). The parameter  $\delta_k$  in (3.10) is a design parameter and is analogous to the process noise statistic used in the Kalman filter, [25].

### 3.2.2 FM

Due to the additional independence assumption for FM, it is possible to utilize more complex transition models, both for the kinematic and for the extent state. In this thesis two different models will be evaluated for FM.

- Same linear motion as for CM, will be referred to as FM,
- Nonlinear motion, will be referred as the *FMNL*.

#### 3.2.2.1 Linear FM

The kinematic covariance matrix's dependency on the extent state must be removed from the model used in CM, if the same model is to be applied to FM. The motion model is a CV model as above with the same matrix definitions as in (2.5). Thus the linear transition density for FM is

$$\begin{aligned} & p(x_{k+1}|X_{k+1}, x_k)p(X_{k+1}|X_k) \\ &= \mathcal{N}(x_{k+1}; (F_k \otimes \mathbf{I}_d)x_k, Q_k \otimes \mathbf{I}_d) \\ & \times \mathcal{W}_d(X_{k+1}; \delta_k, X_k/\delta_k) \end{aligned} \quad (3.11)$$

#### 3.2.2.2 Nonlinear FM (FMNL)

Here the motion model for the kinematic state is the CT model. Due to the fact that the heading and turn-rate is included in the kinematic state, it is possible to make the time evolution of the extent state dependent on the turning. Thus the following transition density, from [19], is used for FMNL,

$$p(\xi_{k+1}|\xi_k) = \mathcal{N}(x_{k+1}; f_k(x_k), Q_k) \times \mathcal{W}_d\left(X_{k+1}; \delta_k, \frac{M(x_k)X_kM^T(x_k)}{\delta_k}\right), \quad (3.12)$$

where the matrix product  $M(x_k)X_kM^T(x_k)$  is the *similarity transformation*, see, e.g., [28]. Here the matrix valued function  $M(x_k)$  is defined as the rotational matrix,

$$M(x_k) = \begin{bmatrix} \cos(h\omega_k) & -\sin(h\omega_k) \\ \sin(h\omega_k) & \cos(h\omega_k) \end{bmatrix}, \quad (3.13)$$

where  $h$  is the sampling time as above.

### 3.3 Measurement Model

In both extended object tracking and group target tracking, there is an issue arising from multiple measurements. The problem is that in a group consisting of several point objects that are spatially close to each other, some of the objects might get blocked by other objects and do not give rise to any measurements at the current time instance. The number of measurements obtained at each time instance is unknown and needs to be modelled. Other scenarios when the number of obtained measurements are unknown are when the object, that are being tracked, moves in such a way that, depending on its orientation, its extent vary or the distance to the sensor changes. Scenarios such as these might lead to different sensor readings each time.

When dealing with multiple measurements, there are several different measurement models that can be used. One common model that is used, e.g. in [9], is a two part model. The first part is that a measurement has a spatial distribution of the point, known as *scattering point*, on the object that yield the measurement and then the second part is the pure measurement noise.

The model used in this thesis is introduced here and is similar to the one used in [25]. It assumes that the measurements are spread over the extent of the object and that the pure sensor noise is so small that the covariance is assumed to be zero. Thus with the random matrix approach, the extent estimate is similar to fitting a Gaussian distribution to a set of samples, [3].

To perform the measurement update step in the Bayesian forward filtering, a model is needed that connects the states to the measurements. The measurement model used in this thesis is assumed to be linear. If a nonlinear measurement model were to be used, then it could, e.g., be linearized. So for simplicity it is assumed that the measurement model is linear and only measures the Cartesian positions. The position of the sensor is assumed known.

#### 3.3.1 Assumptions

There are some assumptions made regarding the measurements such as the number of obtained measurements at each time instance  $k$ , denoted as  $n_k$ . The assumptions are

1. There are no clutter measurements
2.  $n_k$  is unknown and modeled as a Poisson random variable with a constant arrival rate or *measurement rate*  $\lambda$ , i.e.  $p(n_k) = Pois(\lambda)$
3. It is assumed that the number of measurements is independent of the states, i.e.  $p(n_k|\xi_k) = p(n_k)$ .

4. At each time instance  $k$  there is a *probability of detection* ( $P_d$ ) for each object
5. All objects are detected at the first time instance.

The first assumption is due to simplicity, but could be removed in a future extension of this work. The second assumption is to model the fact that sensors that have the ability to generate several measurements at each scan might give different number of measurements each time. The independence assumption between the number of measurements and the states might seem like a crude assumption, since it is likely that the larger an object is and the closer it is to the sensor, the more measurements it would yield. But a dependency would also yield a more complicated estimation task to tackle and might remove the focus from the smoothing performance.

The fourth assumption is to model the probability of successfully detecting an object. There are many likely situations that might result in a missed detection, such as occlusion of the the object. The last one is probably the weakest assumption, since in a real tracking application one can not simply assume that all the objects are being detected at the first time instance. But since there are no models of when new objects occur, *object birth*, or vanishes, *termination*, i.e. that the number of objects is assumed to be fixed during the whole batch of data, this assumption is needed to ensure convergence of the algorithm.

#### 3.3.2 Measurement Likelihood

To include the extent in a logical way in the measurement likelihood, the following approach is introduced in [25] and [42]. Since it is common that sensor readings often are located all over the extent of the object, the approach is to model the spread as Gaussian with covariance dependent of the extent state. In other words it is assumed that each measurement is a measurement of the object's centroid with a spread proportional to the extent.

$$z_k^r = H_k x_k + \gamma_k \quad \text{with} \quad H_k = \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_{d \times n-d} \end{bmatrix}, \quad (3.14)$$

where  $\gamma_k$  is a white Gaussian measurement noise with density  $p(\gamma_k) = \mathcal{N}(\mathbf{0}, X_k)$ . Note that the measurement matrix  $H$  changes in size depending on which motion model is used. For the CV the state vector dimension is  $n = 4$  and for the CT  $n = 5$ .

Since it is assumed that all the measurements are of the objects centroid it is possible to calculate two sufficient statistics, based on the obtained measurements at time  $k$ ,

$$\bar{z}_k = \frac{1}{n_k} \sum_{r=1}^{n_t} z_k^r \quad \text{and} \quad \bar{Z}_k = \sum_{r=1}^{n_t} (z_k^r - \bar{z}_k)(z_k^r - \bar{z}_k)^T. \quad (3.15)$$

The sufficient statistics are the sample centroid measurement and the sample scattering matrix.

### 3.4 Forward Filtering

Forward filtering for extended objects with the random matrix approach has been covered substantially in various literature such as [25, 9, 19, 33]. The key update

equations are summarized here for convenience. The notation  $A^{(i,j)}$  denotes the  $ij$ th element of matrix  $A$ .

### 3.4.1 CM

The forward filtering equations for CM are presented in Table 3.1.  $F_k$  and  $Q_k$  are the transition and process covariance matrix from (2.5).  $\mathbf{I}_d$  is the identity matrix of size  $d$ . The kinematic prediction equations can be found in [25] and [38].

For the extent state the prediction equations are based on a matrix distribution approximation. When evaluating the Chapman–Kolmogorov equation [38, Chapter 3] for the extent state, with the transition model given in (3.10), the distribution of the predicted extent state is a General Beta density type 2 [25], see Appendix A.1. To receive the desired Inverse Wishart distribution, the Beta density is approximated according to Corollary 1 in [19] and the following prediction equations, in Table 3.1, are obtained. Here  $v_{k+1|k}$  and  $V_{k+1|k}$  are the scalar degrees of freedom and scale matrix of the extent state at time  $k + 1$  given information up to time  $k$ . The extent process noise is denoted  $\delta_k$ .

#### CM

---

##### Prediction

Kinematic state:

$$\begin{aligned} m_{k+1|k} &= (F_k \otimes \mathbf{I}_d)m_{k|k} \\ P_{k+1|k} &= F_k P_{k|k} F_k^T + D_k \end{aligned}$$

Extent state:

$$\begin{aligned} v_{k+1|k} &= (d + 1) \frac{\zeta_1 - 2\zeta_2}{\zeta_1 - \zeta_2} \\ V_{k+1|k} &= \frac{(v_{k+1|k} - d - 1)(\delta_k - d - 1)}{v_{k|k} - d - 1} \left( \frac{V_{k|k}}{\delta_k} \right) \\ \zeta_1 &= \frac{\delta_k - d - 1}{v_{k|k} - d - 1} \\ \zeta_2 &= \frac{\delta_k}{v_{k|k} - 2d - 2} \end{aligned}$$

##### Filtering

Kinematic state:

$$\begin{aligned} m_{k|k} &= m_{k|k-1} + (W_{k|k-1} \otimes \mathbf{I}_d)(\bar{z}_k - Hm_{k|k-1}) \\ P_{k|k} &= P_{k|k-1} - W_{k|k-1} S_{k|k-1} W_{k|k-1}^T \\ S_{k|k-1} &= (e_d^1)^T P_{k|k-1} e_d^1 + \frac{1}{n_k} \\ W_{k|k-1} &= P_{k|k-1} e_d^1 S_{k|k-1}^{-1} \end{aligned}$$

Extent state:

$$\begin{aligned} V_{k|k} &= V_{k|k-1} + N_{k|k-1} + \bar{Z}_k \\ v_{k|k} &= v_{k|k-1} + n_k \\ N_{k|k-1} &= S_{k|k-1}^{-1} (\bar{z}_k - Hx_{k|k-1})(\bar{z}_k - Hx_{k|k-1})^T \end{aligned}$$


---

**Table 3.1:** The Forward filtering algorithm for CM.

In Table 3.1,  $(e_d^1)^T$  is the first identity vector of size  $d$ , e.g.  $(e_d^1)^T = [1 \ 0]$  when  $d = 2$ . Here  $W_{k|k-1}$  corresponds to the Kalman gain and  $S_{k|k-1}$  to the innovation covariance matrix in the classical Kalman filter. The kinematic state measurement update equations are from [25]. Here the synthetic measurements from (3.15) are used. The update equations for the extent state are also from [25], where  $N_{k|k-1}$  is the innovation matrix at time  $k$  given information up until time  $k - 1$ .

#### 3.4.2 FM

The forward filtering equations for FM model are presented in Table 3.2. For FM with linear motion, the prediction equations for the kinematic state are almost the same as for CM in Table 3.1 with only a slight difference. In the kinematic state update equations for FM in Table 3.2,  $\hat{Y}_k$  is a proportionality matrix that connects the precision of the synthetic measurement with the extent. The extent state update equation for FM comes from [9], where  $\hat{N}_{k|k-1}$  and  $\hat{Z}_k$  are the symmetric counterpart of the innovation and scattering matrix with the additional proportionality matrix included. Here the notation  $B^{1/2}$  of a matrix is the matrix squared root. It is defined such that a matrix  $A = B^{1/2}$  is the square root of a matrix  $B$  if  $B = A^T A$ , see, e.g., [28] and [35].

#### 3.4.3 FMNL

The forward filtering equations for FMNL model are presented in Table 3.3 and 3.4. The kinematic state prediction for FMNL is taken from the Extended Kalman filter [38], where  $f'_k$  denotes the Jacobian matrix of  $f_k$  evaluated at  $m_{k|k-1}$ , [38]. The update for both the kinematic and extent state in FMNL is exactly the same as for FM in Table 3.2 with the only difference that the size of the measurement matrix  $H$  is changed. In the kinematic state update equations for FMNL  $\hat{Y}_k$  is a proportionality matrix that connects the precision of the synthetic measurement with the extent. The extent state update equation for FMNL comes from [9], where  $\hat{N}_{k|k-1}$  and  $\hat{Z}_k$  are the symmetric counterpart of the innovation and scattering matrix with the additional proportionality matrix included.

### 3.5 Smoothing

The smoothing equations for the three models are presented here. The kinematic state equations resembles the *Rauch-Tung-Striebel smoother (RTS)*, see, e.g., [38, 25, 42]. The derivation of the smoothing equations for the extent state, that are developed in this thesis, are moved to the Appendix A.3 and A.4 for convenience. The smoothing algorithm for CM can be seen in Table 3.5.

FM smoothing equations for the kinematic state are given in Table 3.6. The matrix  $U_k$  and scalar  $u_k$  are defined in Appendix A.3. The algorithm for CM and FM resemble each other since they are based on the same models.

The smoothing algorithm for FMNL can be seen in Table 3.7 and 3.8. The matrix  $M$  is defined as the rotational matrix in (3.13). The smoothing equations for the kinematic state for FMNL are taken from the *Extended Rauch-Tung-Striebel*

**FM**


---

**Prediction**

Kinematic state:

$$\begin{aligned} m_{k+1|k} &= F_k m_{k|k} \\ P_{k+1|k} &= F_k P_{k|k} F_k^T + Q_k \end{aligned}$$

Extent state:

$$\begin{aligned} v_{k+1|k} &= (d+1) \frac{\zeta_1 - 2\zeta_2}{\zeta_1 - \zeta_2} \\ V_{k+1|k} &= \frac{(v_{k+1|k} - d - 1)(\delta_k - d - 1)}{v_{k|k} - d - 1} \left( \frac{V_{k|k}}{\delta_k} \right) \\ \zeta_1 &= \frac{\delta_k - d - 1}{v_{k|k} - d - 1} \\ \zeta_2 &= \frac{\delta_k}{v_{k|k} - 2d - 2} \end{aligned}$$

**Filtering**

Kinematic state:

$$\begin{aligned} m_{k|k} &= m_{k|k-1} + W_{k|k-1}(\bar{z}_k - H m_{k|k-1}) \\ P_{k|k} &= P_{k|k-1} - W_{k|k-1} S_{k|k-1} W_{k|k-1}^T \\ S_{k|k-1} &= H P_{k|k-1} H^T + \hat{Y}_k \frac{1}{n_k} \\ W_{k|k-1} &= P_{k|k-1} H^T S_{k|k-1}^{-1} \\ \hat{Y}_k &= X_{k|k-1} \end{aligned}$$

Extent state:

$$\begin{aligned} V_{k|k} &= V_{k|k-1} + \hat{N}_{k|k-1} + \hat{Z}_k \\ v_{k|k} &= v_{k|k-1} + n_k \\ N_{k|k-1} &= S_{k|k-1}^{-1} (\bar{z}_k - H x_{k|k-1}) (\bar{z}_k - H x_{k|k-1})^T \\ \hat{N}_{k|k-1} &= X_{k|k-1}^{1/2} S_{k|k-1}^{-1/2} N_k (S_{k|k-1}^{-1/2})^T (X_{k|k-1}^{1/2})^T \\ \hat{Z}_k &= X_{k|k-1}^{1/2} \hat{Y}_k^{-1/2} \bar{Z}_k (\hat{Y}_k^{-1/2})^T (X_{k|k-1}^{1/2})^T \end{aligned}$$


---

**Table 3.2:** The Forward filtering algorithm for FM.

(*ERTS*) smoother, see, e.g., [38]. The matrix  $L_k$  and scalar  $l_k$  are defined in Appendix A.4.

### 3.6 Multiple Extended Object Tracking

The framework for tracking extended objects, introduced above in Section 3.1, can be expanded to the case of multiple extended objects tracking, [42, 12, 30]. When introducing the possibility of multiple objects a new kind of issue arises: the *data association problem*.

**FMNL**
**Prediction**

Kinematic state:

$$m_{k|k} = f_k(m_{k|k-1})$$

$$P_{k|k} = f'_k P_{k|k-1} f_k^T + Q_k$$

Extent state:

$$v_{k+1|k} = (d+1) \frac{\zeta_1 - 2\zeta_2}{\zeta_1 - \zeta_2}$$

$$V_{k+1|k} = \frac{(v_{k+1|k} - d - 1)(s_{k|k} - d - 1)}{2b_{k+1|k}} \frac{S_{k|k}}{\gamma_{k|k}}$$

$$\zeta_1 = \frac{s_{k|k} - d - 1}{2b_{k+1|k}}$$

$$\zeta_2 = \frac{s_{k|k}}{2b_{k+1|k} - d - 1}$$

$$b_{k+1|k} = \frac{v - d - 1}{2}$$

$$S_{k|k} = \frac{1}{s_{k|k}} \mathbb{C}_{II}$$

 $s_{k|k}$  is the solution to

$$0 = d \log \left( \frac{s_{k|k}}{2} \right) - \sum_{i=1}^d \psi_0 \left( \frac{s_{k|k} - i + 1}{2} \right) + \mathbb{C}_I - \log |\mathbb{C}_{II}|$$

$$\mathbb{C}_I \approx \log |V_{k|k}|$$

$$\mathbb{C}_{II} \approx M^{-1}(m_{k|l}^{(5)} h) V_{k|k} M^{-T}(m_{k|l}^{(5)} h) + \begin{bmatrix} A_1 & A_2 \\ A_2 & A_3 \end{bmatrix} P_{k|l}^{(5,5)}$$

$$A_1 = 2h^2 (V_{k|k}^{(2,2)} - V_{k|k}^{(1,1)}) c_2 - 4V_{k|k}^{(1,2)} c_1$$

$$A_2 = -4h^2 (V_{k|k}^{(2,2)} - V_{k|k}^{(1,1)}) c_1 + V_{k|k}^{(1,2)} c_2$$

$$A_3 = 2h^2 (V_{k|k}^{(1,1)} - V_{k|k}^{(2,2)}) c_2 + 4V_{k|k}^{(1,2)} c_1$$

$$c_1 = \cos(m_{k|l}^{(5)} h) \sin(m_{k|l}^{(5)} h)$$

$$c_2 = \cos(m_{k|l}^{(5)} h)^2 - \sin(m_{k|l}^{(5)} h)^2$$

$$v = \frac{\zeta_3 - 2\zeta_4}{\zeta_3 - \zeta_4}$$

$$\zeta_3 = \frac{\delta_k - d - 1}{v_{k|k} - d - 1}$$

$$\zeta_4 = \frac{\delta_k}{v_{k|k} - 2d - 2}$$

$$\gamma_{k|k} = \delta_k \frac{v_{k|k} - d - 1}{(v_{k|k} - d - 1)(\delta_k - d - 1)}$$

**Table 3.3:** The prediction algorithm for FMNL.

**3.6.1 Assumptions**

For this thesis there are some assumptions made regarding the multiple object case. The assumptions for the single object case given in Section 3.3 are all still valid in

**FMNL**


---

**Filtering**

Kinematic state:

$$\begin{aligned}
 m_{k|k} &= m_{k|k-1} + W_{k|k-1}(\bar{z}_k - Hm_{k|k-1}) \\
 P_{k|k} &= P_{k|k-1} - W_{k|k-1}S_{k|k-1}W_{k|k-1}^T \\
 S_{k|k-1} &= HP_{k|k-1}H^T + \hat{Y}_k \frac{1}{n_k} \\
 W_{k|k-1} &= P_{k|k-1}H^T S_{k|k-1}^{-1} \\
 \hat{Y}_k &= X_{k|k-1}
 \end{aligned}$$

Extent state:

$$\begin{aligned}
 V_{k|k} &= V_{k|k-1} + \hat{N}_{k|k-1} + \hat{Z}_k \\
 v_{k|k} &= v_{k|k-1} + n_k \\
 N_{k|k-1} &= S_{k|k-1}^{-1}(\bar{z}_k - Hx_{k|k-1})(\bar{z}_k - Hx_{k|k-1})^T \\
 \hat{N}_{k|k-1} &= X_{k|k-1}^{1/2} S_{k|k-1}^{-1/2} N_k (S_{k|k-1}^{-1/2})^T (X_{k|k-1}^{1/2})^T \\
 \hat{Z}_k &= X_{k|k-1}^{1/2} \hat{Y}_k^{-1/2} \bar{Z}_k (\hat{Y}_k^{-1/2})^T (X_{k|k-1}^{1/2})^T
 \end{aligned}$$


---

**Table 3.4:** The filtering algorithm for FMNL.

the multiple object case. There is a need for some additional assumptions. They are

- the number of objects is known
- all objects are present during the whole batch, although they might not be detected at each time instance.

### 3.6.2 Data Association

The reason for the new problem is due to the fact that given a measurement scan  $\mathbf{z}_k = z_k^1, \dots, z_k^{n_k}$  at a time  $k$ , there is no telling which object yields which measurement. In the more general setting there is no deterministic way of determining how many objects that are detected during the current measurement scan.

The association problem has many aspects. In this case when the number of objects are known and deterministic, one can regard the problem as a classical machine learning clustering problem. This enables the use of several well known clustering algorithms. The ones utilized here are the K-means algorithm and Expectation-Maximization algorithm, see, e.g., [4, 37, 21, 1]. The k-means algorithm is used as an initialization to the EM algorithm that solves the association problem iteratively.

## 3.7 Multiple Extended Object Tracking cast as an EM-problem

The definition of multiple object tracking follows mainly the elaborated framework given by [42]. We denote the set of estimates for object  $j$  for the whole batch as  $\Xi^j = \{\xi_0^j, \dots, \xi_N^j\}$  and the set of all objects estimates as  $\mathcal{X} = \{\Xi^1, \dots, \Xi^M\}$ . The

**CM**
**Smoothing**

Kinematic state:

$$\begin{aligned} m_{k|l} &= m_{k|k} + (W_{k|k+1} \otimes \mathbf{I}_d)(m_{k+1|l} - m_{k+1|k}) \\ P_{k|l} &= P_{k|k} + W_{k|k+1}(P_{k+1|l} - P_{k+1|k})W_{k|k+1}^T \\ W_{k|k+1} &= P_{k|k}F_k^T P_{k+1|k}^{-1} \end{aligned}$$

Extent state:

$$\begin{aligned} V_{k|l} &= V_{k|k} + U_k \\ v_{k|l} &= v_{k|k} + u_k \\ U_k &= (u_k - d - 1)(b_k - d - 1)B_k \frac{\delta_{k|k-1}}{\delta_{k|k-1} - d - 1} \\ u_k &\text{ is the solution to} \\ 0 &= \sum_{i=1}^d \left[ \psi^{(0)}\left(\frac{u_k - d - i}{2}\right) + \psi^{(0)}\left(\frac{b_k + 1 - i}{2}\right) - \right. \\ &\quad \left. - \psi^{(0)}\left(\frac{\delta_{k|k-1} - d - i}{2}\right) \right] - d \log\left(\frac{u_k - d - 1}{2}\right) + \\ &\quad + \log \left| B_k \delta_{k|k-1} \right| + \log \left| \frac{\delta_{k|k-1} - d - 1}{b_k - d - 1} (B_k \delta_{k|k-1})^{-1} \right| \\ B_k &= V_{k+1|l} (b_k (\nu_{k+1|l} - d - 1))^{-1} \\ b_k &\text{ is the solution to} \\ 0 &= \sum_{i=1}^d \left[ \psi^{(0)}\left(\frac{b_k - i}{2}\right) + \psi^{(0)}\left(\frac{v_{k+1|l} - d - i}{2}\right) \right] + \\ &\quad + d \log\left(\frac{b_k}{2}\right) - \log \left| V_{k+1|l} \right| + d \log(2) + \log \left| \frac{V_{k+1|l}}{v_{k+1|l} - 2d - 2} \right| \end{aligned}$$

**Table 3.5:** The backward smoothing algorithm for CM.

set of all the accumulated measurement data is denoted as  $\mathbf{Z}$ . The goal of multiple object tracking, in the EM-framework given in (2.25), can be written mathematically as

$$\hat{\mathcal{X}} = \underset{\mathcal{X}}{\operatorname{argmax}} p(\mathcal{X}|\mathbf{Z}). \quad (3.16)$$

The difficulties are as mentioned above the unknown assignments. Here we denote the set of assignments as  $A = \{\mathbf{a}_k\}_{k=0}^N$  of measurements to objects,  $\mathbf{a}_k = [a_k^1 \ \dots \ a_k^{n_k}]$ , with  $a_k^r \in \{1, \dots, M\}$ . The assignments are discrete random variables that map the measurements  $r \in \{1, \dots, n_k\}$  at time  $k$  to one of the objects  $m \in \{1, \dots, M\}$ . In the EM framework the assignments are the latent variables and the parameters are the object estimates. As noted before, the hidden variables are unknown, so instead of calculating the optimal assignments  $A$ , the EM algorithm calculates the distribution over the space of  $A$ .

The EM algorithm can be interpreted as a lower-bound maximization [4, Chapter 9]. The E-step consists of calculating the optimal lower bound with respect to the posterior weights defined as  $p(A|\mathbf{Z}, \mathcal{X}^{(i)})$  for each possible value of  $A$ . Here  $i$  denotes the current EM iteration number, thus  $\mathcal{X}^{(i)}$  denotes the current guess. Thus

**FM**
**Smoothing**

Kinematic state:

$$\begin{aligned} m_{k|l} &= m_{k|k} + W_{k|k+1}(m_{k+1|l} - m_{k+1|k}) \\ P_{k|l} &= P_{k|k} + W_{k|k+1}(P_{k+1|l} - P_{k+1|k})W_{k|k+1}^T \\ W_{k|k+1} &= P_{k|k}F_k^T P_{k+1|k}^{-1} \end{aligned}$$

Extent state:

$$\begin{aligned} V_{k|l} &= V_{k|k} + U_k \\ v_{k|l} &= v_{k|k} + u_k \\ U_k &= (u_k - d - 1)(b_k - d - 1)B_k \frac{\delta_{k|k-1}}{\delta_{k|k-1} - d - 1} \\ u_k &\text{ is the solution to} \\ 0 &= \sum_{i=1}^d \left[ \psi^{(0)}\left(\frac{u_k - d - i}{2}\right) + \psi^{(0)}\left(\frac{b_k + 1 - i}{2}\right) - \right. \\ &\quad \left. - \psi^{(0)}\left(\frac{\delta_{k|k-1} - d - i}{2}\right) \right] - d \log\left(\frac{u_k - d - 1}{2}\right) + \\ &\quad + \log|B_k \delta_{k|k-1}| + \log\left|\frac{\delta_{k|k-1} - d - 1}{b_k - d - 1} (B_k \delta_{k|k-1})^{-1}\right| \\ B_k &= V_{k+1|l} (b_k (\nu_{k+1|l} - d - 1))^{-1} \\ b_k &\text{ is the solution to} \\ 0 &= \sum_{i=1}^d \left[ \psi^{(0)}\left(\frac{b_k - i}{2}\right) + \psi^{(0)}\left(\frac{v_{k+1|l} - d - i}{2}\right) \right] + \\ &\quad + d \log\left(\frac{b_k}{2}\right) - \log|V_{k+1|l}| + d \log(2) + \log\left|\frac{V_{k+1|l}}{v_{k+1|l} - 2d - 2}\right| \end{aligned}$$

**Table 3.6:** The backward smoothing algorithm for FM.

the optimal lower bound takes the following form

$$\mathcal{Q}(\mathcal{X}; \mathcal{X}^{(i)}) = \sum_A \log(p(\mathcal{X}, A, \mathbf{Z})) \times p(A|\mathbf{Z}, \mathcal{X}^{(i)}). \quad (3.17)$$

Note that (3.17) is an expectation with respect to a discrete random variable, which is the reason why the first step is called the expectation step.

During the M-step, the maximization is carried out based on the parameter  $\mathcal{X}$  given the previously calculated assignments, which then gives the next improved estimate  $\mathcal{X}^{(i+1)}$ . This estimate is then used in the following E-step to calculate the new assignments etc.

The EM algorithm continues until  $\mathcal{Q}$  converges or some termination criterion is set. The proof that each iteration of the EM indeed yields an improvement (if there is any improvement that can be done) is found in [4]. But it is important to note that the algorithm might get caught at a local maxima.

In summary, the E-step calculates the optimal assignments between measurement and object, and the assignment weights used are the probability of the assign-

**FMNL**
**Smoothing**

Kinematic state:

$$\begin{aligned} m_{k|l} &= m_{k|k} + W_{k|k+1}(m_{k+1|l} - m_{k+1|k}) \\ P_{k|l} &= P_{k|k} + W_{k|k+1}(P_{k+1|l} - P_{k+1|k})W_{k|k+1}^T \\ W_{k|k+1} &= P_{k|k}(f'_k)^T P_{k+1|k}^{-1} \end{aligned}$$

Extent state:

$$\begin{aligned} V_{k|l} &= V_{k|k} + L_k \\ v_{k|l} &= v_{k|k} + l_k \\ L_k &= (l_k - d - 1)(s_k - d - 1) \frac{g_k S_k}{h_k - d - 1} \\ l_k &\text{ is the solution to} \\ 0 &= \sum_{i=1}^d \left[ \psi^{(0)} \left( \frac{l_k - d - i}{2} \right) + \psi^{(0)} \left( \frac{s_k + 1 - i}{2} \right) - \right. \\ &\quad \left. - \psi^{(0)} \left( \frac{h_k - d - i}{2} \right) \right] - d \log \left( \frac{l_k - d - 1}{2} \right) + \\ &\quad + \log |g_k S_k| + \log \left| \frac{h_k - d - 1}{s_k - d - 1} (g_k S_k)^{-1} \right| \\ S_k &= \frac{1}{s_k} \mathbb{C}_{II} \\ s_k &\text{ is the solution to} \\ 0 &= d \log \left( \frac{s_k}{2} \right) - \sum_{i=1}^5 \psi^{(0)} \left( \frac{s_k + 1 - i}{2} \right) + \mathbb{C}_I - \log |\mathbb{C}_{II}| \\ \mathbb{C}_I &\approx \log |W_k| \\ \mathbb{C}_{II} &\approx M^{-1} (m_{k|l}^{(5)} h) W_k M^{-T} (m_{k|l}^{(5)} h) + \begin{bmatrix} A_1 & A_2 \\ A_2 & A_3 \end{bmatrix} P_{k|l}^{(5,5)} \\ A_1 &= 2h^2 (W_k^{(2,2)} - W_k^{(1,1)}) c_2 - 4W_k^{(1,2)} c_1 \\ A_2 &= -4h^2 (W_k^{(2,2)} - W_k^{(1,1)}) c_1 + W_k^{(1,2)} c_2 \\ A_3 &= 2h^2 (W_k^{(1,1)} - W_k^{(2,2)}) c_2 + 4W_k^{(1,2)} c_1 \\ c_1 &= \cos(m_{k|l}^{(5)} h) \sin(m_{k|l}^{(5)} h) \\ c_2 &= \cos(m_{k|l}^{(5)} h)^2 - \sin(m_{k|l}^{(5)} h)^2 \\ g_k &= \frac{(h_k - d - 1)(w_k - d - 1)}{\delta_{k+1|k-d-1}} \end{aligned}$$

 Continues in Table 3.8
 

---

**Table 3.7:** The backward smoothing algorithm for FMNL.

ments, i.e.

$$w_k^{rm(i)} = P(a_k^r = m | \mathbf{Z}, \xi_k^m, \pi_k^{m(i)}) \quad (3.18)$$

 where  $w_k^{rm(i)}$  is the assignment probability and  $\pi_k^{m(i)}$  is the prior assignment proba-

**FMNL**
**Smoothing**

Extent state cont.:

$$\begin{aligned}
 & h_k \text{ is the solution to} \\
 0 &= \sum_{i=1}^d \left[ \psi^{(0)} \left( \frac{h_k - d - i}{2} \right) + \psi^{(0)} \left( \frac{w_k + 1 - i}{2} \right) - \right. \\
 & \quad \left. - \psi^{(0)} \left( \frac{\delta_k - d - i}{2} \right) \right] - d \log \left( \frac{h_k - d - 1}{2} \right) + \\
 & \quad + \log \left| \delta_k M^{-1} (m_{k|l}^{(5)} h) W_k M^{-T} (m_{k|l}^{(5)} h) \right| + \\
 & \quad + \log \left| \frac{\delta_k - d - 1}{w_k - d - 1} (\delta_k M^{-1} (m_{k|l}^{(5)} h) W_k M^{-T} (m_{k|l}^{(5)} h))^{-1} \right| \\
 W_k &= \frac{V_{k+1|l} - V_{k+1|k}}{w_k (v_{k+1|l} - v_{k+1|k} - d - 1)} \\
 & w_k \text{ is the solution to} \\
 0 &= \sum_{i=1}^d \left[ \psi^{(0)} \left( \frac{w_k + 1 - i}{2} \right) + \right. \\
 & \quad \left. + \psi^{(0)} \left( \frac{v_{k+1|l} - v_{k+1|k} - d - i}{2} \right) \right] + \\
 & \quad + d \log \left( \frac{w_k}{2} \right) - \log \left| V_{k+1|l} - V_{k+1|k} \right| + d \log(2) + \\
 & \quad + \log \left| \frac{V_{k+1|l} - V_{k+1|k}}{v_{k+1|l} - v_{k+1|k} - 2d - 2} \right|
 \end{aligned}$$

**Table 3.8:** The backward smoothing algorithm for FMNL.

bility at time  $k$ , that measurement  $r$  is assigned to object  $m$ .

In the M-step the estimated trajectories and extents are optimized with respect of the current assignments. The M-step is carried out by applying the above presented forward backward smoother in Section 3.1, with the only difference that the total number of measurements  $n_k$  at time  $k$  for an object are replaced by its estimate,  $\sum_{r=1}^{n_k} w_k^{rm(i)}$ . Thus the maximization of the posterior distribution of the states is equivalent to calculating the maximum likelihood in the M-step according to [42].



# 4

## Simulation Scenarios and Performance Evaluation

This chapter gives an overview of the different scenarios the three derived models are being evaluated in. First a summary of the models is given, next it goes through the single object case, and finally the multiple object case.

### 4.1 The Models

Of the three models, CM is the model of these three that is closest to the general model in (3.1), i.e., it relies on the least number of assumptions. The drawback is that it demands a linear transition model and can not have nonlinear dependencies between the states. FM is included to enable a comparison of the conditional model and the factorized model with the same selection of transition and measurement models. FMNL is evaluated to see if the benefits of the factorized model weigh up the additional assumptions and approximations needed to obtain the model.

### 4.2 Extended Object Tracking

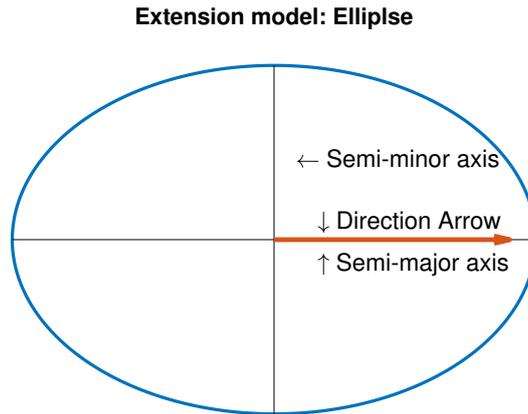
The extended object smoothing models that were derived in Section 3.1 for extended objects are first evaluated in the single object case. This initial evaluation is to examine if the smoothing calculations do indeed yield a better result than the filtering and prediction calculation from [25] and [9]. In other words, this is to see if smoothing is worth the extra effort that it costs in computational terms.

#### 4.2.1 Test Perspectives

There are two main perspectives that need to be examined, namely

1. the difference between smoothing and the forward filtering results made within the same model
2. the difference between smoothing performances for the three models.

The first perspective is to evaluate whether extra computational cost of the smoothing calculations is worth the effort in comparison to the existing forward filtering. In theory the smoothing should outperform the forward filtering since it utilizes more information at each time estimate. But since these derived smoothing algorithms are based on some additional assumptions and approximations, it can not be assumed that it is better than the filtering. The second perspective is interesting since the



**Figure 4.1:** An example ellipse of the extent model and its direction arrow and the definition of the axis names.

evaluation consists of three models: CM, FM and FMNL. If it turns out in the first perspective that the smoothing is worth the extra effort, then it is interesting to see how well they perform in relation to each other in different scenarios.

#### 4.2.1.1 Extent

The random matrix approach, see Section 3.1.1, that is used to model the extent, models the shape of the extent estimate as an ellipse. Here the true extent is also shaped as an ellipse. No matter if the extent area and shape are changing or not, the true ellipse is always orientated such that the semi-major axis always is in parallel with the heading of the object. For definitions of the axes, see Figure 4.1.

In the case of time varying extent, the length of both the semi-major axis and semi-minor axis are a function of time. Due to Assumption 3 in Section 3.1.5, the true extent is thus changing slowly over time.

#### 4.2.1.2 Probability of Detection

The probability of detection  $P_d$ , Section 3.3, is a way to model the properties of the sensor. In a real situation there is a line of sight of a sensor, and the objects might not always be detected. The reasons for this could be many, such as technical issues of the sensor that make it unable to generate any measurements at a given time instance. It could also be that an object is being covered by either an obstacle or another object. Furthermore, it could also be that the sensor has blind spots or that an object momentarily is escaping the line of sight of the sensor.

To model these situations there are at every time instance a sampling from a Bernulli distribution with the probability of success equal to  $P_d$ , this is to see if there should be any measurements at all of an object. Note that in this thesis there are no explicit obstacles.

If the sampling of the detection yields a detection at time  $k$  then the number of obtained measurements  $n_k$  at time  $k$  are generated by a Poisson distribution with

a measurement rate  $\lambda$ , see Section 3.3. An important remark is that there are two ways that there might be no measurements at a given time  $k$ , either there are no detection or the number of obtained measurements given a detection is zero. By denoting the probability of that the number of obtained measurements are zero by  $p_{n_k} = P(n_k = 0)$ , then the total probability of not obtaining any measurements can mathematically be expressed as

$$P(\text{No measurements}) = (1 - P_d)p_{n_k} + (1 - P_d)(1 - p_{n_k}) + P_dp_{n_k}. \quad (4.1)$$

#### 4.2.1.3 Measurement Rate

The measurement rate is the Poisson parameter used for the number of obtained measurements if a detection is made at a certain time instance. It is denoted  $\lambda$ . If  $\lambda$  is high, each measurement scan has a higher probability of containing much information, so the measurement rate thus gives a notion about the expected information density at each detection.

### 4.2.2 Test Scenarios

For each of the two perspectives of the evaluation, a total of five scenarios are tested. Each scenario consists of two cases, a linear case and a nonlinear case. The results from each scenario are based on 100 *Monte-Carlo simulations*, see, e.g., [3]. In the linear case the true trajectory in each Monte-Carlo iteration is randomly generated by a linear motion model, the CV model. The same holds for the nonlinear case, but with the difference that the motion model is the nonlinear CT model. The scenarios are structured as follow

1. First scenario:  $P_d = 1$ ,  $\lambda = 8$  and constant extent
2. Second scenario:  $P_d = 1$ ,  $\lambda = 8$  and time-varying extent
3. Third scenario:  $P_d = 0.75$ ,  $\lambda = 8$  and time-varying extent
4. Fourth scenario:  $P_d \in \{0.5, 0.25\}$ ,  $\lambda = 8$  and time-varying extent
5. Fifth scenario:  $P_d = 0.75$ ,  $\lambda \in \{3, 1\}$  and time-varying extent

Note that both scenario four and five actually consists of two scenarios each, where  $P_d$  and  $\lambda$  takes on two different values respectively, but are put together due to similarity.

### 4.2.3 Evaluation

The performance is measured with the *Gaussian Wasserstein Distance (GWD)*, see Section 4.2.3.1. The unit of the position is measured in meter [ $m$ ], the velocity is measured in [ $m/s$ ], heading in [ $rad$ ] and turn-rate [ $rad/s$ ]. The output from the GWD is in [ $m^2$ ], but the units are omitted in the tables below.

#### 4.2.3.1 Performance Metric

In the case of extended object tracking there are many different techniques to use as performance metrics, see the discussions in [15, 44]. To evaluate the outcome of the results in this thesis the Gaussian Wasserstein Distance is utilized. It was shown in

[44] that the GWD is a good choice for evaluating estimates of elliptically shaped objects. The squared Wasserstein distance between two multivariate Gaussians,  $\mathcal{N}_x(x; m_x, \Sigma_x)$  and  $\mathcal{N}_{\hat{x}}(\hat{x}; m_{\hat{x}}, \Sigma_{\hat{x}})$ , is defined as

$$d_{GW}(\mathcal{N}_x, \mathcal{N}_{\hat{x}})^2 = \|m_x - m_{\hat{x}}\|^2 + Tr\left(\Sigma_x + \Sigma_{\hat{x}} - 2\sqrt{\sqrt{\Sigma_x}\Sigma_{\hat{x}}\sqrt{\Sigma_x}}\right) \quad (4.2)$$

Where  $Tr$  denotes the matrix trace,  $\|\cdot\|^2$  denotes the Euclidean vector norm and  $\sqrt{\cdot}$  denotes the matrix squared root similar as in Table 3.2.

To get a numerical representation for a simulation to simplify the comparison of the three models with each other for different scenarios, the following value is calculated by taking the median of the summarized GWD estimation error over the whole time batch for each Monte-Carlo simulation. This numerical result represents the median of the cumulative error for all Monte-Carlo simulations, and thus works as an intuitive measure on the performance of the models during a 100 Monte-Carlo simulation. It can be expressed as the median of  $\mathbf{e}_A$ , where  $A \in \{CM, FM, FMNL\}$ ,  $\mathbf{e}_A \in \mathbb{R}^{MC}$  with  $MC$  is the total number of Monte-Carlo simulations and  $\mathbf{e}_A$  is defined as

$$\mathbf{e}_A = \left[ \sum_k e_{GWD}^{k,1} \quad \sum_k e_{GWD}^{k,2} \quad \dots \quad \sum_k e_{GWD}^{k,MC} \right]^T, \quad (4.3)$$

where  $e_{GWD}^{k,i}$  is the GWD error for time instance  $k \in \{1, \dots, N\}$  and Monte-Carlo simulation  $i \in \{1, \dots, MC\}$ .

### 4.3 Multiple Extended Object Tracking

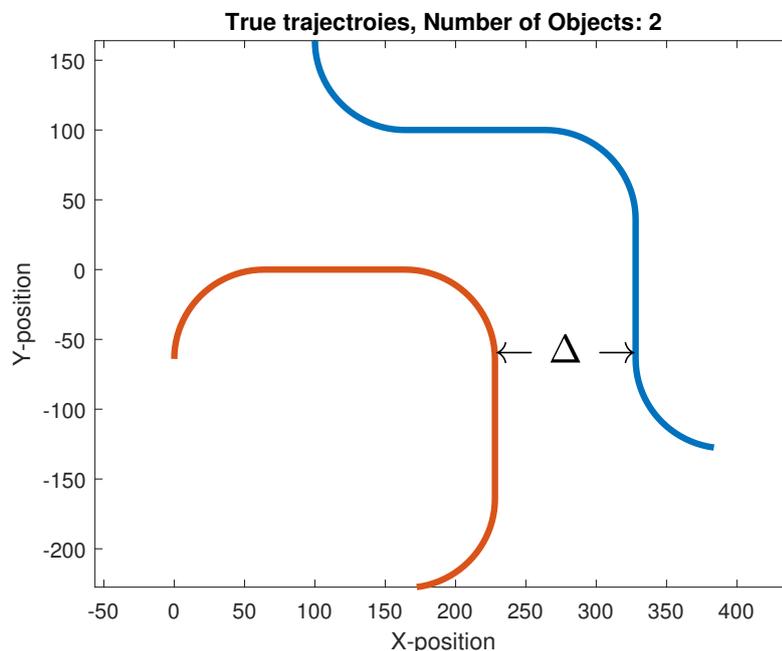
The three models introduced in Chapter 3 are also evaluated in multiple extended object scenarios. A new aspects that arises in the multiple object case, as compared to the single object case, is the data association problem or clustering problem, see Section 3.6.2. Its complexity varies, and one of the main factors that has a direct impact on the complexity is the distance between the objects. A large distance makes the clustering easier, while for narrow spaces the data points overlap and it is more difficult to distinguish the clusters from one another. If the number of objects is unknown, then this situation is extra tricky due to the fact that the obtained data might either be from one large object or several small objects that are close to each other. In this thesis the number of objects, denoted  $M$ , is assumed to be known.

As stated in Section 3.6.2 the data association problem is iteratively solved with the EM-algorithm. At each integration, EM calculates new assignment weights for each data point to each object. The weights are calculated by the previous iterations best estimate, thus the smoothing estimate. The weights are then used in both the subsequent forward filtering and smoothing in the next EM iteration and this leads to a difficulty in comparing the filtering and smoothing estimates, since the filtering is dependent on the smoothing. For this reason, the evaluation performed in the multiple extended case only concerns the different smoothing results.

### 4.3.1 Ground Truth

In the evaluation of the models when there is only one object, there are always two cases. Either the true motion is randomly generated from a linear model or it is randomly generated by a nonlinear model. In the multiple object case the true trajectories of the objects are deterministic. This is because it enables the possibility to examine the situations with different distances in a more elaborate way.

In Figure 4.2 the true trajectories are shown in the case when the number of objects is  $M = 2$ , and object distance, denoted  $\Delta$ , is larger than  $10\sigma$ . Here  $\sigma$  is the standard deviation of the true extent, and  $\Delta > 10\sigma$  is thus considered as a large distance since 99.7% of the distribution mass of a Gaussian distribution is within  $3\sigma$ .



**Figure 4.2:** The shape of the true trajectories for the case when  $M = 2$  and  $\Delta > 10\sigma$

### 4.3.2 Scenarios

The evaluation consists of the following three scenarios:

1. the distance  $\Delta$  decreases from  $> 10\sigma$  to  $\sigma$  and  $\frac{1}{3}\sigma$  for time varying extent and constant extent, respectively.
2. the impact of different values of  $P_d$  and  $\Delta$ ,
3. the impact of different values of  $\lambda$  and  $\Delta$ .

The three scenarios work as an extension to the analysis made for the single object case. The focus lies on the new difficulty and the new parameter, namely the data association problem and the object distance. In the first scenario  $P_d = 1$ ,  $\lambda = 8$  and  $\Delta$  varies. For the second scenario  $\lambda = 8$  and both  $P_d$  and  $\Delta$  varies. Finally in the third scenario  $P_d = 1$  and both  $\lambda$  and  $\Delta$  varies.



# 5

## Results

This chapter presents the results obtained based on the simulation of the models in the different scenarios. First the single object case with its five scenarios is presented, followed by the multiple object case with its corresponding three scenarios. There is a colour coding used in the following figures, in the case when the three models are compared to each other CM is always blue, FM green and FMNL red. In the case when a model is compared within it self with its prediction, filtering and smoothing results, the prediction is cyan coloured, filtering magenta and the smoothing black.

### 5.1 Extended Object/ Single Object

Some of the resulting figures are moved to Appendix B for convenience, since many of the simulated scenarios gave quite similar results. In this chapter the more interesting results are highlighted.

#### 5.1.1 First Scenario

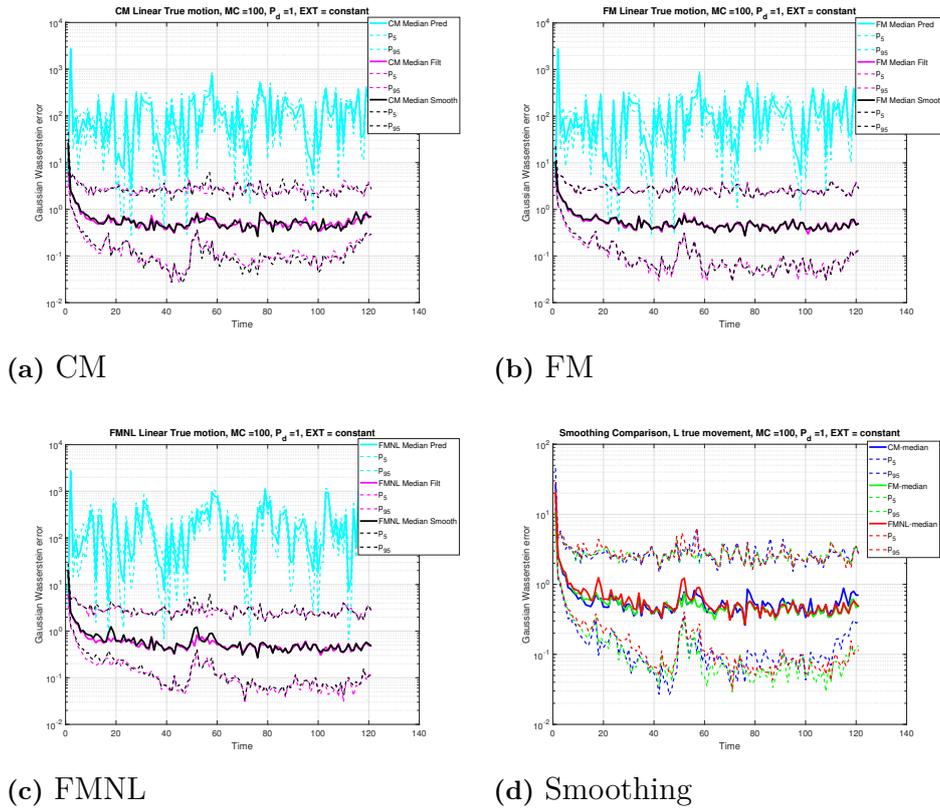
The first scenario works as a baseline. Since it is the easiest scenario it should yield the smallest error. The upcoming scenarios can then be compared to this first one.

##### 5.1.1.1 Linear Case

In contrast to the traditional assumption that there is only one measurement at each time instance, the multiple measurements case gives a lot more information about the object. The estimate can thus yield a better performance, similar to the result obtain when fusing multiple sensors readings [38].

One direct consequence of this is that in the case where the probability of detection is  $P_d = 1$ , there are measurements obtained at each time instance, except when  $n_k = 0$ . The standard forward filtering does yield a very good performance and the smoothing producing almost the same estimates.

The scenario is the same for both nonlinear and linear true motion, with  $P_d = 1$  and the extent is constant. Here the number of obtained measurements at each time instance is, as noted in Section 3.3, Poisson distributed with parameter  $\lambda = 8$ . The fact that the extent is constant enables the extent estimate, both for forward filtering and smoothing, to converge. This makes the difference between filtering and smoothing even smaller. In the case where the true motion is linear, the difference between the forward filtering and smoothing for the three models, CM, FM and FMNL can be seen in figures (a), (b) and (c) in Figure 5.1.



**Figure 5.1:** (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is linear,  $P_d = 1$ ,  $\lambda = 8$  and the extent is constant. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error.

The numerical results of the above simulations are listed in Table 5.1.

Linear $P_d = 1$ , ext = const, $\lambda = 8$	Prediction	Filtering	Smoothing
CM	17607.3	<b>113.3</b>	132.5
FM	17652.2	<b>109.6</b>	114.3
FMNL	28736.8	<b>112.5</b>	131.4

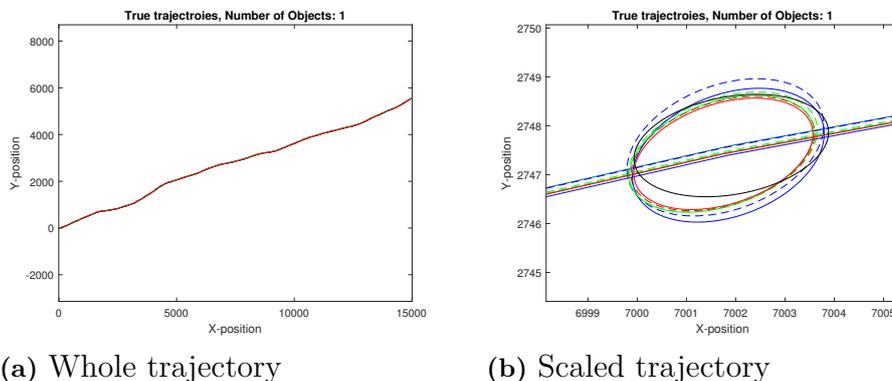
**Table 5.1:** The numerical calculation based on (4.3) in the case of linear true motion,  $P_d = 1$ , the extent is constant (denoted ext = const) and  $\lambda = 8$ .

An important remark is that in the case where the true motion is linear, FMNL has a natural disadvantage since its dynamical model is the nonlinear CT-model, that is beneficial in turning motions. Both CM and FM use the linear CV-model which is the one used to generate the true trajectories. The reason why the filtering yield a better performance, in Table 5.1, than the smoothing of all the three models

in this simple case is due to the additional approximations that are being done in all the smoothing calculations to keep the algorithms mathematically tractable.

It is also worth noting that the numerical error presented here is a median of a cumulative error over time. Thus the median difference in filtering and smoothing error at each time instance is of order of magnitude  $10^{-1}$ , which is also quite clear by looking at figures (a) and (b) in Figure 5.1. It can be seen that for both CM and FM, the smoothing error curves lie almost on top of the filtered result. But the filtering with FMNL shows a more even behaviour than the smoothing in Figure (c) 5.1. FMNL calculation is also the one of the three models that needs the most approximations.

At an example run, with these settings, all the filtering and smoothing estimates lie almost on top of each other and it is hard to distinguish the estimates from each other, as can be seen in Figure 5.2.



**Figure 5.2:** (a) is a typical run of a randomly generated linear trajectory when  $P_d = 1$ ,  $\lambda = 8$  and the extent is constant. Here it is almost impossible to distinguish the different estimation from each other. The blue is CM, green FM, red FMNL and black is the ground truth. (b) show a zoomed in version of (a), here the filtering and the smoothing can be seen. The filterings are the dashed lines with the same colour coding and the smoothing the solid ones.

A zoomed in version of the typical run presented in Figure (a) 5.2 of the three models where the prediction is omitted can be seen in Figure (b) 5.2. The solid lines are the the smoothing estimates and the dashed lines are the filtering estimate. The blue curves are CM, green FM and red FMNL.

To get a view of how the different smoothing algorithms are performing in comparison to each other, CM, FM and FMNL smoothing are plotted together in Figure (d) 5.1. As concluded from the numerical calculations in Table 5.1 that the performance are very similar to each other, and FMNL performs slightly indifferent comparing to the other two. The solid curves are the median error and the dashed lines are the 5- and 95-percentiles. Blue curve are CM, green FM and red FMNL.

### 5.1.1.2 Nonlinear Case

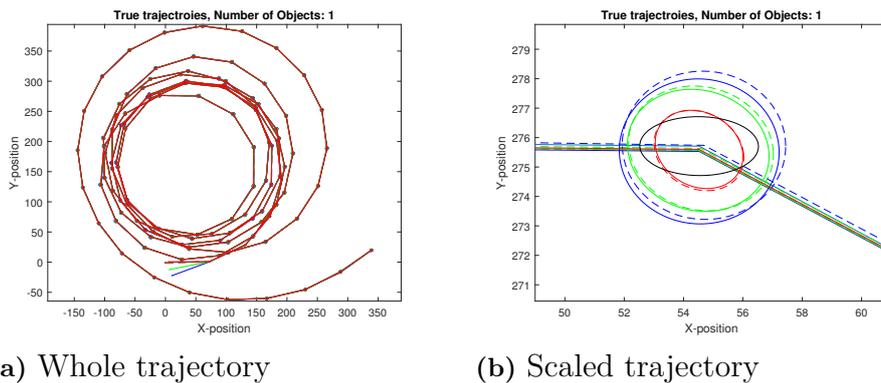
With the same setup as above with  $P_d = 1$ , constant extent and  $\lambda = 8$  but with a nonlinear true motion, a similar result as in the linear case, holds for all the models. For brevity only the numerical results are presented, which can be seen in Table 5.2 and the corresponding plots are in Appendix B.1.1 since they are almost the same as Figure 5.1.

Nonlinear $P_d = 1$ , ext = const, $\lambda = 8$	Prediction	Filtering	Smoothing
CM	9326.8	348.1	<b>124.2</b>
FM	9212.7	351.9	<b>155.5</b>
FMNL	6482.5	350.5	<b>130.1</b>

**Table 5.2:** The numerical calculation based on (4.3) for the case of nonlinear true motion,  $P_d = 1$ , the extent is constant and  $\lambda = 8$ .

Here in Table 5.2 the filtering estimates have a slightly worse performance than all the smoothing calculations, but the order of magnitude of the results are basically the same. Since there are detections at every time instance, the filtering performs similarly to the smoothing estimates. The contingent benefit of the nonlinear motion model of FMNL does not make a clear difference here when  $P_d = 1$  and  $\lambda = 8$ . The performance is thus better for FMNL in relation to the other smoothing estimates here in the nonlinear case than in the above linear case.

A typical run in the nonlinear motion case can be seen in figure (a) and (b) in Figure 5.3. As expected, the performance for this case is similar to the linear one with the same parameter settings. The solid lines are the smoothing, the dashed line the filtering. The blue is CM, green FM, red FMNL and black is the ground truth. Here the prediction is omitted.



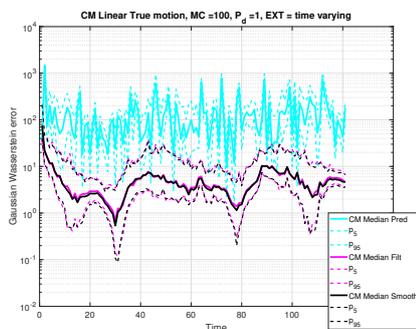
**Figure 5.3:** (a) is a typical run of a randomly generated nonlinear trajectory when  $P_d = 1$ ,  $\lambda = 8$  and the extent is constant. Here it is almost impossible to distinguish the different estimation from each other. The blue is CM, green FM, red FMNL and black is the ground truth. (b) show a zoomed in version of (a), here the filtering and the smoothing can be seen. The filterings are the dashed lines with the same colour coding and the smoothing the solid ones.

A zoomed in version of Figure (a) 5.3 can be seen in Figure (b) 5.3. Here it is easier to see the different estimates. The blue curves are CM, green FM and red FMNL. The black curve and ellipse represents the ground truth. Note here how much better the red FMNL, both its filter and smoothing, is fitting the true extent than the two linear models.

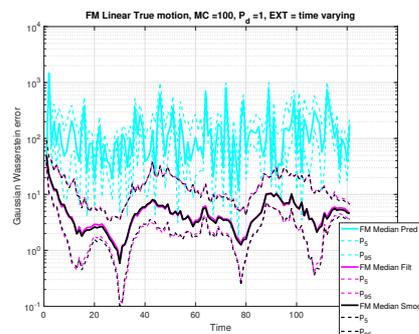
## 5.1.2 Second Scenario - Extent

As mentioned above, the extent does not necessarily need to be rigid and constant over time. Here the three models are evaluated in the case when the true ellipse changes slowly over time.

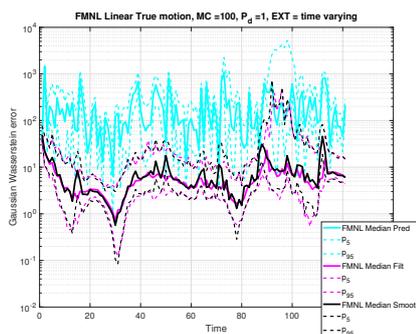
### 5.1.2.1 Linear Case



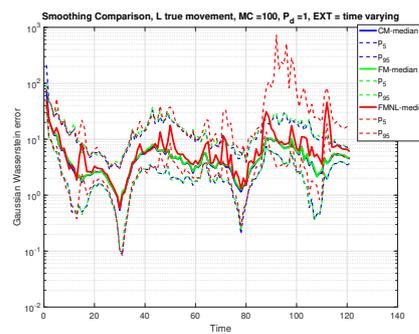
(a) CM



(b) FM



(c) FMNL



(d) Smoothing

**Figure 5.4:** (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is linear,  $P_d = 1$ ,  $\lambda = 8$  and the extent time varying. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error.

A direct advancement in difficulties from the first scenario in Section 5.1.1 is the case when the extent is changing in size. The velocity of the extent change for the true object is quite slow compared to its kinematic velocities, which are a factor  $10^2$ . For the case when the true motion is linear the numerical results for the models are summarized in Table 5.3.

Linear $P_d = 1$ , ext = varying, $\lambda = 8$	Prediction	Filtering	Smoothing
CM	19562.6	<b>760.3</b>	772.1
FM	19716.8	<b>770.5</b>	778.5
FMNL	28444.6	<b>874.6</b>	1115.3

**Table 5.3:** The numerical calculation based on (4.3) for the case of linear true motion,  $P_d = 1$ , the extent is time varying and  $\lambda = 8$ .

In comparison to the values in Table 5.1 the errors are larger here, which is expected due to the more difficult task since the extent estimate never gets the chance to converge. This can be seen in the figures that show the median of the GWD estimation error over time in figures (a) (b) (c) in Figure 5.4. Note that FMNL gives a worse performance than the other two, since here the true motion is linear. Also note that the error here is the cumulative over time and the difference is not that big at each time estimate.

When comparing the models' results from the first scenario in Section 5.1.1 in figures (a), (b) and (c) in Figure 5.1 with the ones here in figures (a), (b) and (c) in Figure 5.4 it is obvious that here there is a periodic change in the error. The true extent is changing periodically with different speeds on its semi-major axis and its semi-minor axis. But the change is deterministic and is thus the same at each time instance for each Monte-Carlo simulation. That is the reason why the error curves appear to have the same periodicity. The points, where the error is small, are at the same time instances for all three models. They occur around the times  $k = \{14, 30, 78, 108\}$ .

Since the prediction step performed for the extent state is a random walk, it assumes that the shape of the extent is similar between the time steps. Thus the yet slowly changing true extent is changing too quickly for the three models. So the models' estimates do follow the shape changing, but not fast enough. The time instances with low error all have in common that the true extent, which has a deterministic change, has either been big and is going back to be small and thus coincide with the estimates, that are still growing in size due to the previous increase of the true extent. This is true also in the opposite case where the true extent has been small and is growing again, and at the time instances with low errors, it coincides with the shrinking estimates. To compare the different smoothings with each other, their estimate errors are shown in Figure (d) 5.4.

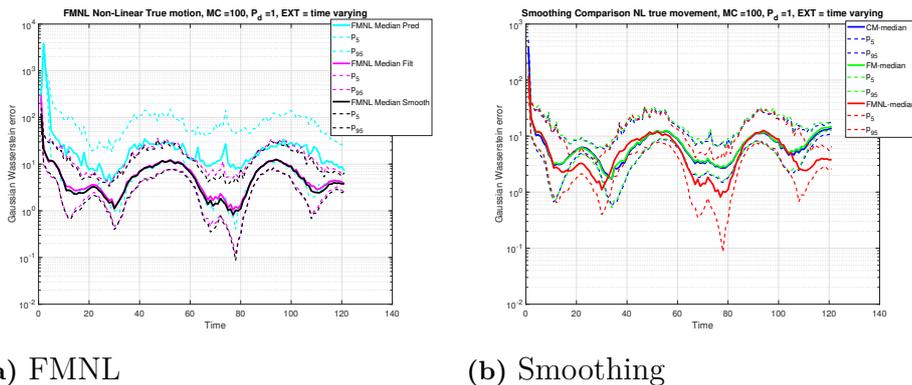
Once again FMNL has the disadvantage of using a nonlinear motion model in contrast to the other two models, and even though it uses a more advanced temporal change model that takes turning in to account for the extent estimate in its smoothing, it does not give much help since in the linear case there are not much turning going on, see for instance Figure 5.2.

### 5.1.2.2 Nonlinear Case

With the same parameter settings:  $P_d = 1$ ,  $\text{ext} = \text{varying}$  and  $\lambda = 8$  but with nonlinear true motion, FMNL does give the best performance. This can be seen by comparing the numerical results between Tables 5.3 and 5.4. This can also be noted by comparing the three smoothing models together, see Figure (b) 5.5. It might also be interesting to compare Figure (c) 5.4 and Figure (a) 5.5 to see that in the nonlinear case, the performance of FMNL is much more even than in the linear case. The corresponding model plots for CM and FM can be found in Appendix B.1.2.

Nonlinear $P_d = 1$ , $\text{ext} = \text{varying}$ , $\lambda = 8$	Prediction	Filtering	Smoothing
CM	154681.5	<b>1308.7</b>	1346.0
FM	147807.4	1321.5	<b>1119.4</b>
FMNL	8186.0	1115.1	<b>898.1</b>

**Table 5.4:** The numerical calculation based on (4.3) for the case of nonlinear true motion,  $P_d = 1$ , the extent is changing over time and  $\lambda = 8$ .



**Figure 5.5:** (a) is FMNL and shows the internal error comparison between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (b) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is nonlinear,  $P_d = 1$ ,  $\lambda = 8$  and the extent time varying. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error.

### 5.1.3 Third Scenario - Probability of Detection

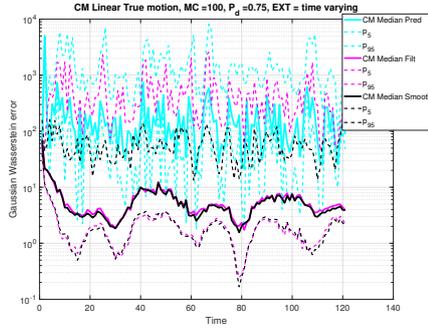
So far it seems like, based on the numerical values of Tables 5.1, 5.2, 5.3 and 5.4 that the smoothing does not yield a better performance. But so far the cases have been nice, since  $P_d = 1$  there is always available information and  $\lambda = 8$  so that the available information is comprehensive and the filtering does yield a very good performance.

## 5. Results

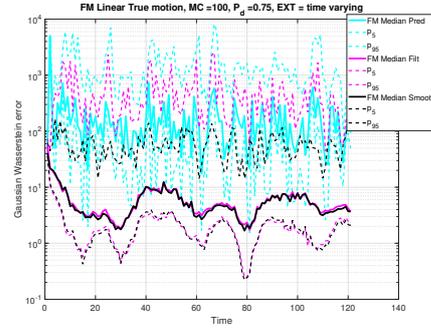
If it happens, for some reason, that the measurements at a certain time instance are absent, then the forward filtering has nothing to do rather than keep the prediction as the updated result. So the filtering result is without any actual update, since it only based its estimates on previous data up to current time. The smoothing estimates on the other hand do not suffer from this disadvantage and can, regardless of whether detections are missed or not, carry on its estimation. In the case where  $P_d < 1$  the performance difference between the forward filtering and the smoothing might be more interesting.

Since the goal is to be able to cope with as difficult situations as possible, i.e. to mimic the real world, the extent is from now on, for the single object case, always time varying with the slow periodic change. For this scenario  $P_d = 0.75$ . This is to model the chance of not getting any detection at a time instance.

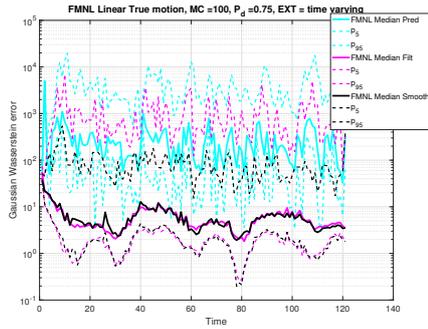
### 5.1.3.1 Linear Case



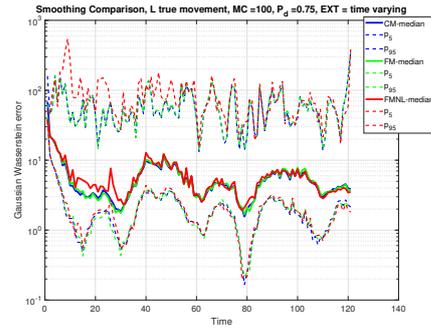
(a) CM



(b) FM



(c) FMNL



(d) Smoothing

**Figure 5.6:** (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is linear,  $P_d = 0.75$ ,  $\lambda = 8$  and the extent time varying. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error.

For the case of true linear motion,  $P_d = 0.75$ , extent is varying with time and  $\lambda = 8$ , the numerical results are presented in Table 5.5. Since it is linear true motion FMNL does not yield a performance as good as CM and FM. When comparing the difference between filtering and smoothing results, there is as expected, a clear difference in performance for all models. The disadvantage that the nonlinear motion model of FMNL yield is now obvious when comparing the numerical errors for the predictions between CM, FM versus FMNL. Since without some detection FMNL's prediction predicts a turning motion that is more correct in this situation while CM and FM predicts a more straight motion.

Linear $P_d = 0.75$ , ext = varying, $\lambda = 8$	Prediction	Filtering	Smoothing
CM	49910.8	9945.2	<b>1802.2</b>
FM	50286.0	9928.5	<b>1803.2</b>
FMNL	97315.3	18327.4	<b>2176.6</b>

**Table 5.5:** The numerical calculation based on (4.3) for the case of linear true motion,  $P_d = 0.75$ , the extent is changing over time and  $\lambda = 8$ .

The GWD estimation error plots over time can be viewed in figures (a), (b) and (c) in Figure 5.6. Although the numerical results in Table 5.5 might demonstrate a major performance difference between filtering and smoothing, it is clear by viewing the error figures in Figure 5.6 that the median difference is still not that big. Since the 95-percentiles of all the filtering lie almost a factor 10 above the smoothing 95-percentile, it implies that the smoothing is more reliable than the filtering in this case. Especially when comparing how the percentiles lies on top of each other in figures (a), (b) and (c) in Figure 5.4 and hence imply that in the previous scenario, the filtering has the same reliability as the smoothing.

In this linear case the difference between the three smoothing models are depicted in Figure (d) 5.6. Here it is easy to see that FMNL has a more difficult time, due to the nonlinear nature of FMNL.

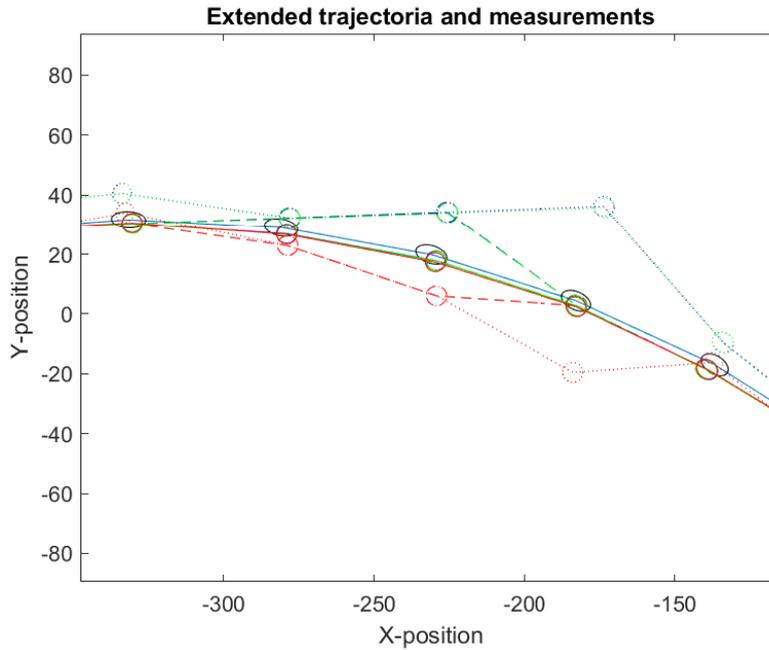
### 5.1.3.2 Nonlinear Case

The results for the same situation as in the previous linear case with parameters  $P_d = 0.75$ , extent is varying with time and  $\lambda = 8$  but now with nonlinear true motion can be seen in Table 5.6.

Nonlinear $P_d = 0.75$ , ext = varying, $\lambda = 8$	Prediction	Filtering	Smoothing
CM	408614.1	76661.5	<b>1523.1</b>
FM	402077.0	75186.6	<b>1260.2</b>
FMNL	18173.5	2357.0	<b>932.5</b>

**Table 5.6:** The numerical calculation based on (4.3) for the case of nonlinear true motion,  $P_d = 0.75$ , the extent is changing over time and  $\lambda = 8$ .

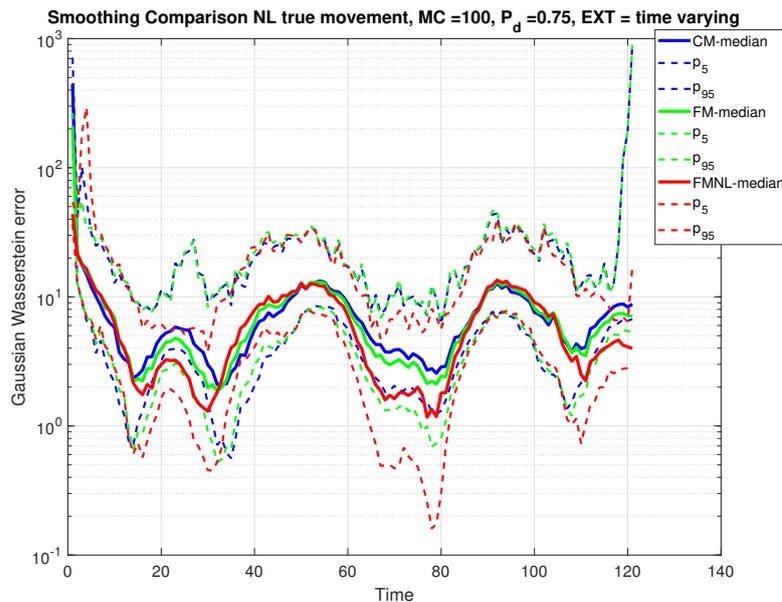
Now FMNL is the model that does gives the best performance. This is expected since the motion model of FMNL outperforms the linear one used in CM and FM. The GWD estimate error plots can be viewed in Appendix B.1.3.



**Figure 5.7:** Here is a typical run for the nonlinear true motion when  $P_d = 0.75$ ,  $\lambda = 8$  and the extent is varying with time. It is zoomed in on a curve where there is no detection made at the third the time instances, counted from the left. The blue colored curves are CM, green color is FM and red is FMNL. The solid lines are the smoothings, the dashed lines are the filtering and the dotted lines are the predictions. The black curves are the ground truth.

To get a view of a typical run for this scenario, see Figure 5.7. Here it is possible to distinguish how the different models cope with a turn where there is no detection made at the second and third time instances, counted from left. The linear models, CM and FM, predict straight lines and lie almost on top of each other, the dotted blue and green curves. FMNL prediction, red dotted curve, predicts a turning motion with a similar turn-rate as before the missed detection, and thus follows the true motion better. Here it is also possible to see how the filtering (the dashed lines) of all models first follows their corresponding prediction in the case of missed detection, but as soon as there are any measurements available at the next time instance it corrects the estimate, while the prediction carries on for one more time instance. The solid lines are the smoothing, and they use all available information both after and before the point of no detection and thus yield a good estimate.

Figure 5.8 shows the GWD estimation error for the tree models smoothing result. The blue curves are CM, green FM and red FMNL. The solid lines are the median error and the dashed are the 5- and 95-percentiles respectively. As mentioned above, the periodic behaviour of the curves are due to the periodic change of the true extent. But more importantly is to note that FMNL outperforms the other two smoothing models in the nonlinear case, which is emphasized by the numerical result in Table 5.6.



**Figure 5.8:** Here are all the three smoothing result compared together for the case of nonlinear true motion and  $P_d = 0.75$ , extent = varying and  $\lambda = 8$ . The blue solid line is CM, and the dashed blue lines are its corresponding 5- and 95- percentiles. The same goes for the green solid and dashed lines but are for FM and the red curves are FMNL.

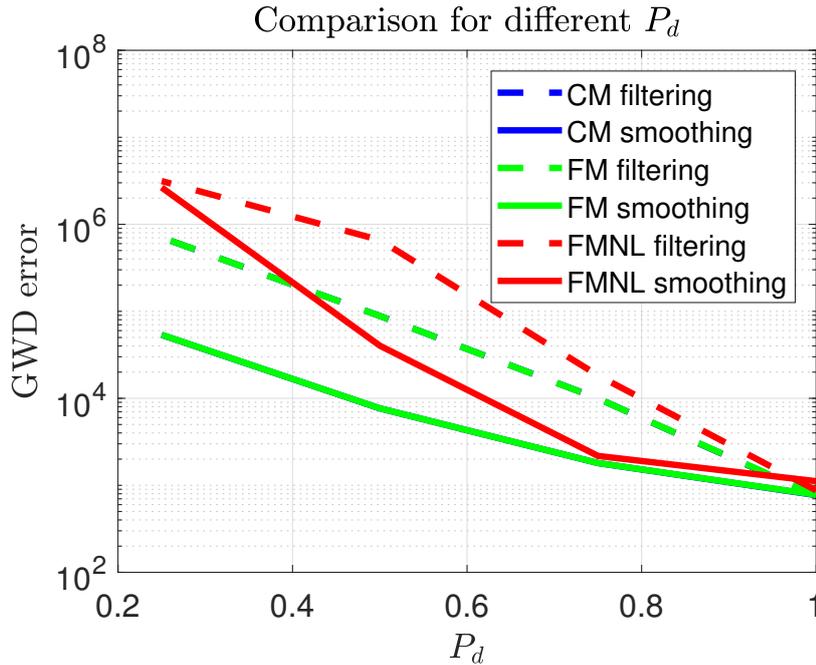
#### 5.1.4 Fourth Scenario - Probability of Detection

To stretch the difference between the filtering and smoothing even further, here the case with  $P_d = 0.5$  and  $P_d = 0.25$  will be evaluated. Since there is no clutter or any external noise sources in the measurement model, the situation when  $P_d$  gets small might be one way to see if the smoothing models can cope with the difficult situation. Here  $\lambda = 8$  to ensure that when a detection is made the information does contain a reasonable amount of information.

##### 5.1.4.1 Linear Case

For the linear true motion case when  $P_d = 0.5$  and  $P_d = 0.25$  the result can be seen in Table B.1 and B.2 in Appendix B.1.4. To compare the results from Tables 5.3, 5.5, B.1 and B.2, a visual representation of the four tables can be seen in Figure 5.9.

In Figure 5.9 the decrease of the probability of detection does yield a larger difference between the filtering and smoothing. Note that the y scale is logarithmic, blue curves are CM, green FM and red FMNL. CM and FM lies almost on top of each other and hence it is hard to distinguish the difference. But here in the linear case both CM and FM performs better than FMNL. Indeed a smaller value of  $P_d$  does not only make the forward filtering estimate error to increase, but also increases the smoothing estimation error. As can be viewed in Figure (a) 5.10 in the case of  $P_d = 0.5$  that the estimation is quite good even though, on average, every second detection are missed. In Figure (b) 5.10 a zoomed version of Figure (a) 5.10 can

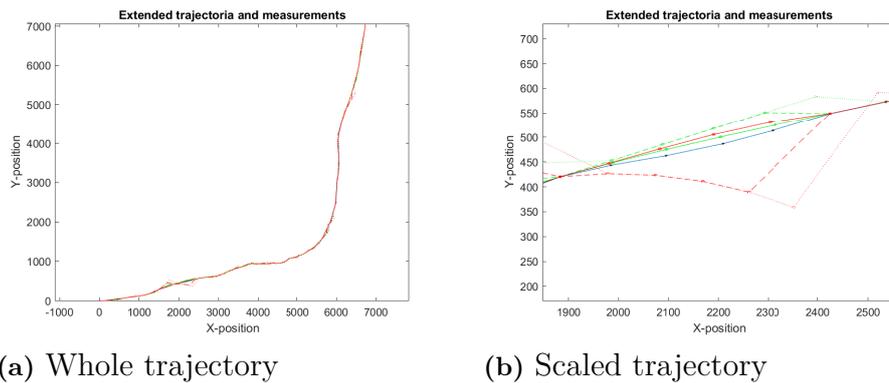


**Figure 5.9:** The error comparison between the filtering and smoothing for different values of  $P_d$  for linear true motion. FM lies on top of CM.

be seen. Here there is a sequence of four missed detection in a row. This gives the forward filtering a hard time to deal with, while the smoothing does, as expected, a better performance. Note that FMNL is predicting a turning arc and consequently gets a larger error in the linear case.

For the case when  $P_d = 0.25$ , an example trajectory can be seen in Figure (a) 5.11. FMNL is at an extreme limit, and has difficulties of following the true trajectory since its prediction estimate gives large errors. CM's and FM's predictions are still very well fitting and thus enable both the filtering and the smoothing to follow the true trajectory quite well in this extreme case. A zoomed in version of what happens can be seen in Figure (b) 5.11 where there is a sequence of 11 missed detections, then a single detection occurs and is followed by additional 6 missed detections. FMNL prediction is ill fitted and FMNL filtering is shifting rapidly and thus FMNL smoothing gets its wave-like appearance when it combines the long sequence of predictions that are turning away and the point where a single detection is obtain. For linear true motion with  $P_d = 0.25$  FMNL gives a poor estimate that is not to rely on in comparison to CM and FM models.

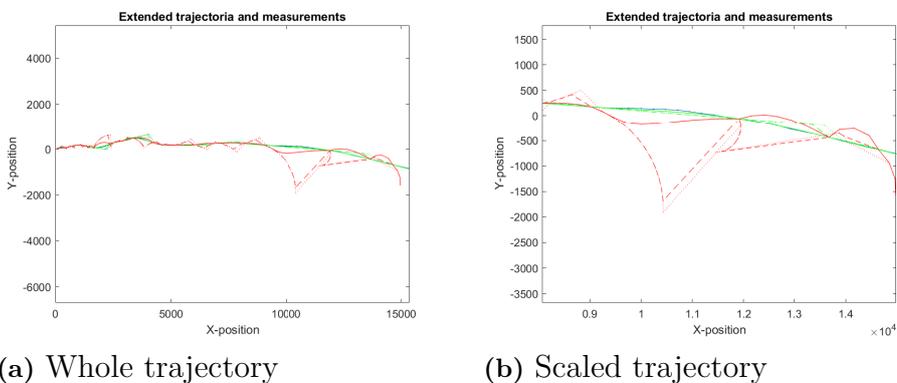
In Appendix B.1.4 the GWD estimation error figures for the three models are shown in the case of  $P_d = 0.5$  and in Appendix B.1.5 the corresponding error figures for the three models are depicted for the case of  $P_d = 0.25$ . Figure (a) 5.12 compares the three smoothing estimates when  $P_d = 0.5$  based on 100 Monte-Carlo simulations. The median of FMNL is the red solid line and emphasizes the analysis made above, that FMNL has a difficult time in the linear case in comparison to the other two, CM in blue and FM in green. It is important to remark the big difference in the 95-percentiles between the models. FMNL rises far above the other two models and implies that there are a lot of cases of the 100 Monte-Carlo simulations where FMNL



(a) Whole trajectory

(b) Scaled trajectory

**Figure 5.10:** (a) is a typical run of a randomly generated linear trajectory when  $P_d = 0.5$ ,  $\lambda = 8$  and the extent is time varying. Here it is almost impossible to distinguish the different estimation from each other. The blue is CM, green FM, red FMNL and black is the ground truth. (b) show a zoomed in version of (a), here the filtering and the smoothing can be seen. The filterings are the dashed lines with the same colour coding and the smoothing the solid ones. The trajectories showing here have a series of 4 missed detections in a row, and thus it is possible to see how the prediction moves according to its assumed motion, and the filtering follows the prediction up until there is a detection while the smoothing has a nice estimate just next to the true trajectory.



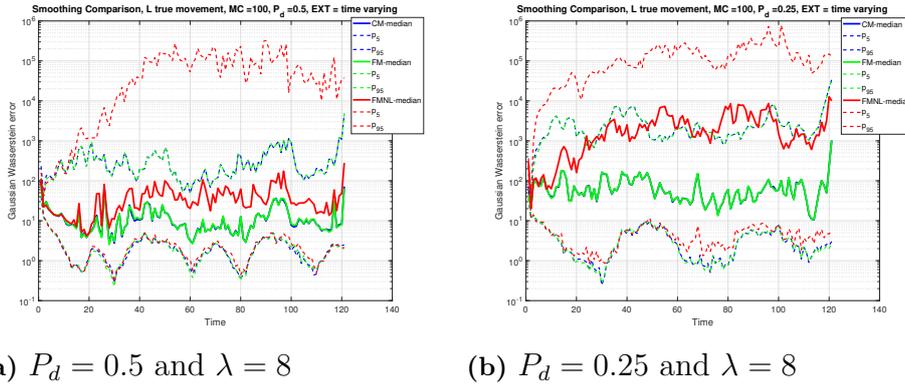
(a) Whole trajectory

(b) Scaled trajectory

**Figure 5.11:** (a) is a typical run of a randomly generated linear trajectory when  $P_d = 0.25$ ,  $\lambda = 8$  and the extent is time varying. The blue is CM, green FM, red FMNL and black is the ground truth. (b) show a zoomed in version of (a), here the forward filtering and the smoothing can be seen. The filterings are the dashed lines with the same colour coding and the smoothing the solid ones. There is a series of 11 missed detections in a row followed by an additional 6.

gives unreliable estimates in the case of  $P_d = 0.5$  with linear motion.

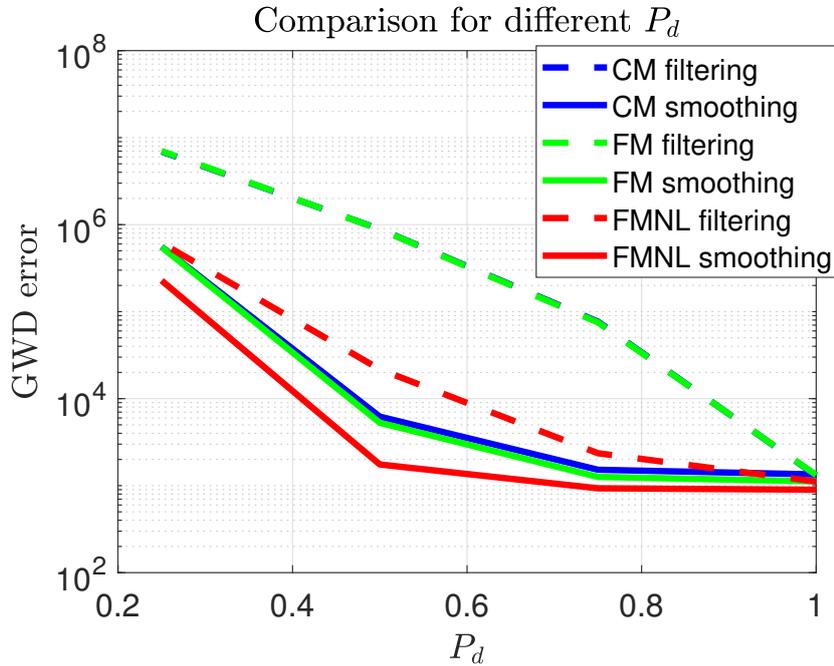
The corresponding error figure is depicted in Figure (b) 5.12 for linear motion and  $P_d = 0.25$ . For this level of detection FMNL is almost completely unreliable for the linear case. By comparing CM and FM, both in Figure (a) 5.12 and in Figure (b) 5.12 they do lie almost on top of each other, but there is a significant difference



**Figure 5.12:** (a) is a comparison between the smoothings when  $P_d = 0.5$ ,  $\lambda = 8$  and the extent is time varying. (b) is a comparison between the smoothings when  $P_d = 0.25$ ,  $\lambda = 8$  and the extent is time varying.

in error level between the situations. So although the situation of having  $P_d = 0.25$  is an extreme case, for linear true motion, CM and FM performs well enough to be reliable.

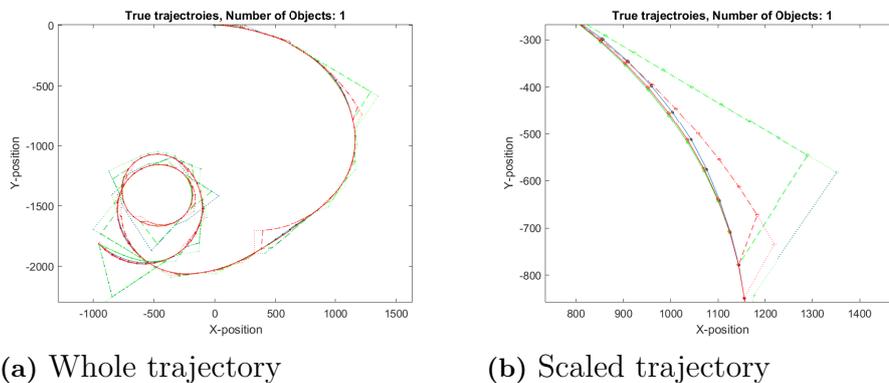
5.1.4.2 Nonlinear Case



**Figure 5.13:** The error comparison between the filtering and smoothing for different values of  $P_d$  for nonlinear true motion.

By comparing the numerical results from Tables 5.4, 5.6, B.3 and B.4, a similar conclusion as in the linear case, can be drawn regarding the relationship with an decreasing  $P_d$  and an increasing difference between the filtering and smoothing. A

visual representation of the four tables can be seen in Figure 5.13. Here in the nonlinear case, in contrast to the linear case in Figure 5.9, FMNL gives the best performance and note how much better FMNL filtering result is in comparing to CM and FM filtering. This is due to that the nonlinear model is beneficial in the nonlinear case.

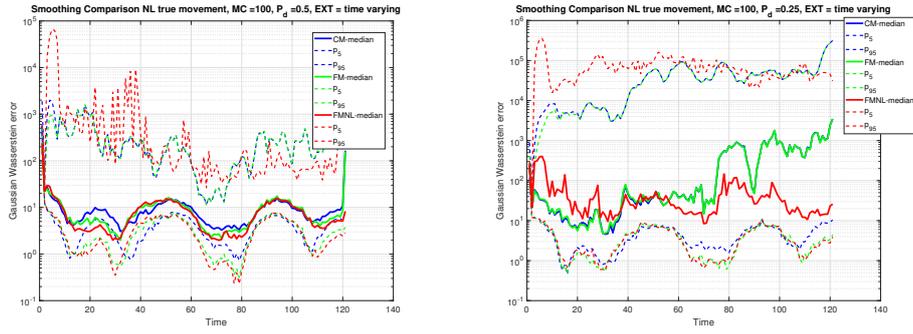


**Figure 5.14:** (a) is a typical run of a randomly generated nonlinear trajectory when  $P_d = 0.5$ ,  $\lambda = 8$  and the extent is time varying. The blue is CM, green FM, red FMNL and black is the ground truth. (b) show a zoomed in version of (a), here the filtering and the smoothing can be seen. The filterings are the dashed lines with the same colour coding and the smoothing the solid ones.

To show what happens when  $P_d$  goes from 0.5 to 0.25 in the nonlinear case, it is examined by looking at typical runs of the two cases. In Figure (a) 5.14 the trajectory estimates are shown for the nonlinear case with  $P_d = 0.5$ . In comparison to Figure 5.10 the first observation is that CM's and FM's predictions are disadvantageous, the blue and green dotted straight lines. According to previous analysis, and stressed by the result in Figure 5.13, that FMNL performs best of the filters in this case. For a better understanding, Figure (b) 5.14 shows a scaled version of the trajectory in Figure (a) 5.14 where a similar behavior of the three models can be observed as appeared in Figure 5.10.

To compare the three models in this case with a more statistical reliable foundation their GWD estimation error based on 100 Monte-Carlo simulation are shown in Figure 5.15. In figure (a) in Figure 5.15, where  $P_d = 0.5$  it is clear that FMNL performs better than the other two for this case, since its median lies below the other two. But be aware of the 95-percentiles of FMNL that for early time instances does overshoot the percentiles of both CM and FM. Thus it implies that there situations appear when the performance of FMNL is poor. For the corresponding error plots for each model see Appendix B.1.6.

When  $P_d = 0.25$ , the models have difficulties on tracking the objects since on average 75% of the detection's are missed. Based on the difference between figures (a) and (b) in Figure 5.15, the case of  $P_d = 0.25$  is a very difficult situation for the filters. For this case all three models perform similarly, but it is possible to see that while CM and FM diverge over time, FMNL converges. This is due to the nonlinear transition model. FMNL makes in this situation much better predic-



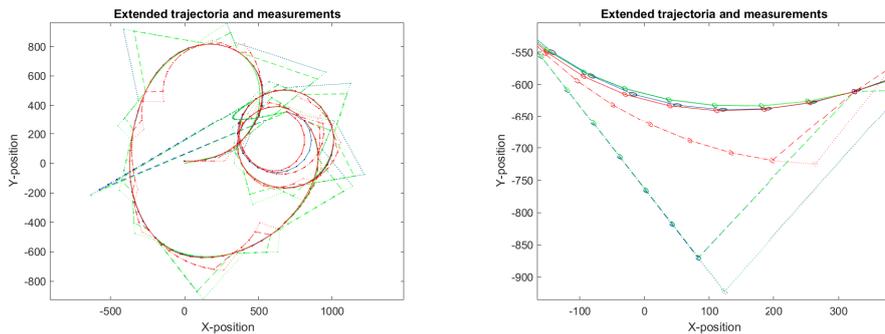
(a)  $P_d = 0.5$  and  $\lambda = 8$

(b)  $P_d = 0.25$  and  $\lambda = 8$

**Figure 5.15:** (a) is a comparison between GWD estimation error for the smoothing of the three models in the case of nonlinear motion,  $P_d = 0.5$ , time varying extent and  $\lambda = 8$ . (b) is a comparison between GWD estimation error for the smoothing of the three models in the case of nonlinear motion,  $P_d = 0.25$ , time varying extent and  $\lambda = 8$ . The blue curves are CM model, green FM and red FMNL. Solid line are the median error and the dashed lines are the 5- and 95-percentiles.

tions between the detections and thus yield a better performance than the linear predictions performed by CM and FM.

A typical run for this nonlinear case is shown in Figure (a) 5.16. It is important to remark how different FMNL performs in this case and in the linear case in Figure 5.11 with the same settings. The blue and green straight lines pointing out almost everywhere are CM and FM predictions and filterings when the detections are missed. FMNL prediction does also wander away over time, but not as drastically as the linear predictions which can be seen in the zoomed in version of the trajectory in Figure (b) 5.16.

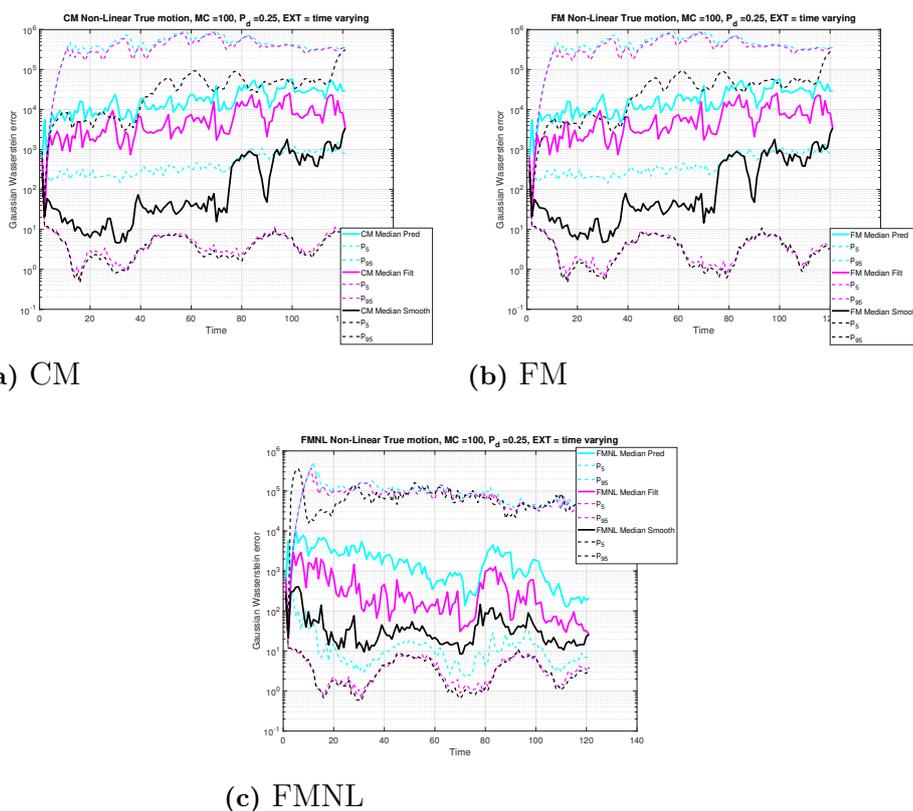


(a) Whole trajectory

(b) Scaled trajectory

**Figure 5.16:** (a) is a typical run of a randomly generated nonlinear trajectory when  $P_d = 0.25$ ,  $\lambda = 8$  and the extent is time varying. The blue is CM, green FM, red FMNL and black is the ground truth. (b) show a zoomed in version of (a), here the filtering and the smoothing can be seen. The filterings are the dashed lines with the same colour coding and the smoothing the solid ones.

In the case of small value of  $P_d$ , the smoothing is beneficial over the forward filtering estimates. The GWD estimation errors for each models are shown in figures (a), (b) and (c) in Figure 5.17. Here in all three figures, there is a large difference between the filtering errors and smoothing errors. Besides that the 95-percentiles of the filtering for all the models are almost the same as the prediction's 95-percentiles. This is because that in many time instances the filtering is equal to the prediction, since there are no detections at those times. So it is thus clear that in the case of small  $P_d$  the smoothing option is to prefer.



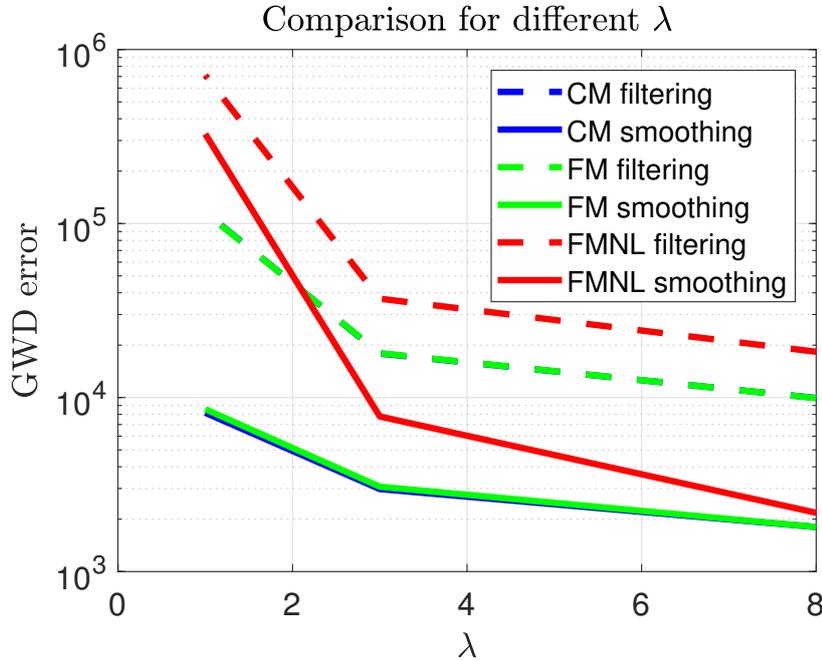
**Figure 5.17:** (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering in magenta, and the black smoothing. Here the true motion is nonlinear,  $P_d = 0.25$ ,  $\lambda = 8$  and the extent is time varying. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error.

### 5.1.5 Fifth Scenario - Measurement Rate

In scenario four above the difference between filtering and smoothing became clear when a lot of detections were missing. But once a detection is made, it did contained a lot of information, since  $\lambda = 8$ . From this, the question arises how well the models perform when each detection contains less information. Note that when  $\lambda = 1$  the situation is similar to the classic approach when there is at most one measurement

per time instance. This makes the estimation of the extent very difficult.

### 5.1.5.1 Linear Case

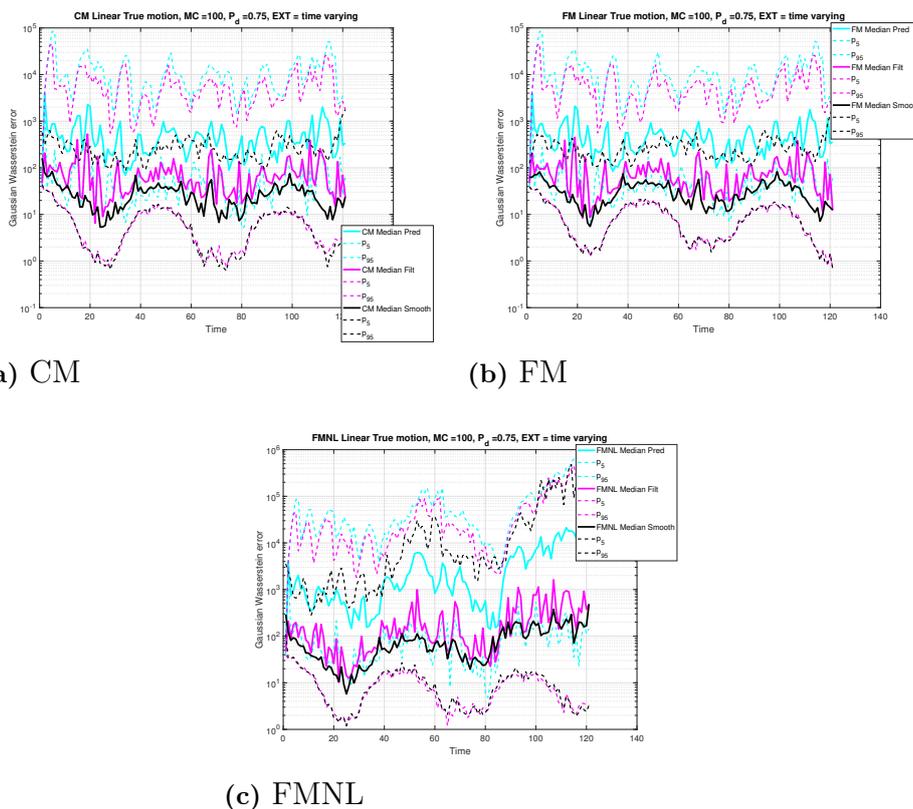


**Figure 5.18:** The error comparison between the filtering and smoothing for different values of  $\lambda$  for linear true motion.

In comparison to the result obtained in Table 5.5 where  $\lambda = 8$ , here are the results when  $\lambda = 3$  and  $\lambda = 1$ . The numerical result is summarized in Table B.5 and B.6 in Appendix B.1.7. A visual representation of the Tables 5.5, B.5 and B.6 can be seen in Figure 5.18.

Here in Figure 5.18 it is possible to see a similar trend as in the case when  $P_d$  was reduced in Figure 5.9. The less number of measurements obtained at each scan, the larger the performance difference between smoothing and filtering is. The precision of the measurements are proportional to the number of obtained measurements [9]. So a smaller number of measurements at each scan implies that the measurement update step in the forward filtering is less accurate and the prediction plays a greater part in both the filtering and smoothing estimate. The performance of FMNL is worse in the linear case than for CM and FM. This can be seen in the GWD estimate error plots in figures (a), (b) and (c) in Figure 5.19. The shape of the prediction can be seen in both the filtering and smoothing curves as a consequence of its increased importance. In comparison to the figures (a), (b) and (c) in Figure 5.6 where instead the filtering and smoothing curves share the same shape, since the prediction plays a smaller part. Note also that the 95-percentiles of the prediction and filtering are almost the same, since by decreasing the number of measurements the difference between the filtering and the prediction gets smaller.

The comparison between the smoothing estimates for the case of  $\lambda = 3$  and  $\lambda = 1$  is shown in Figures (a) 5.20 and Figure (b) 5.20. Here it can again be



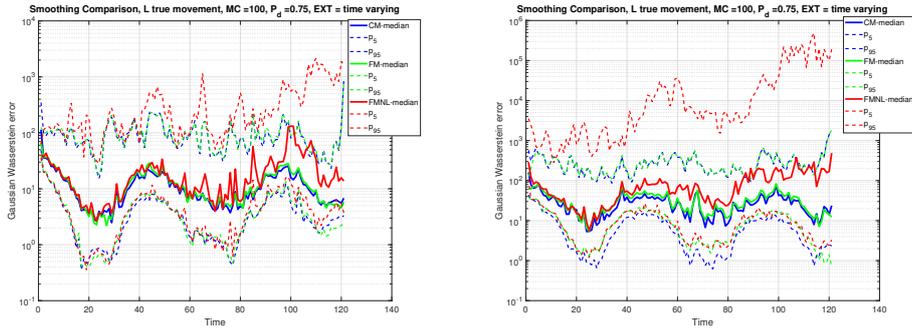
**Figure 5.19:** (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering in magenta, and the black smoothing. The true motion is linear,  $P_d = 0.75$ ,  $\lambda = 1$  and the extent time varying. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error.

seen, and emphasized by Figure 5.18, that in the linear case FMNL has problems, comparing to the other two. For the error plots for each model in the case of  $\lambda = 3$  see Appendix B.1.7

### 5.1.5.2 Nonlinear Case

For the Monte-Carlo simulations based on the nonlinear motion, the results are in Tables B.7 and B.8. They can be seen in Appendix B.1.8 and are the cases when  $\lambda = 3$  and  $\lambda = 1$  respectively. A visual representation of the Tables 5.6, B.7 and B.8 can be seen in Figure 5.21.

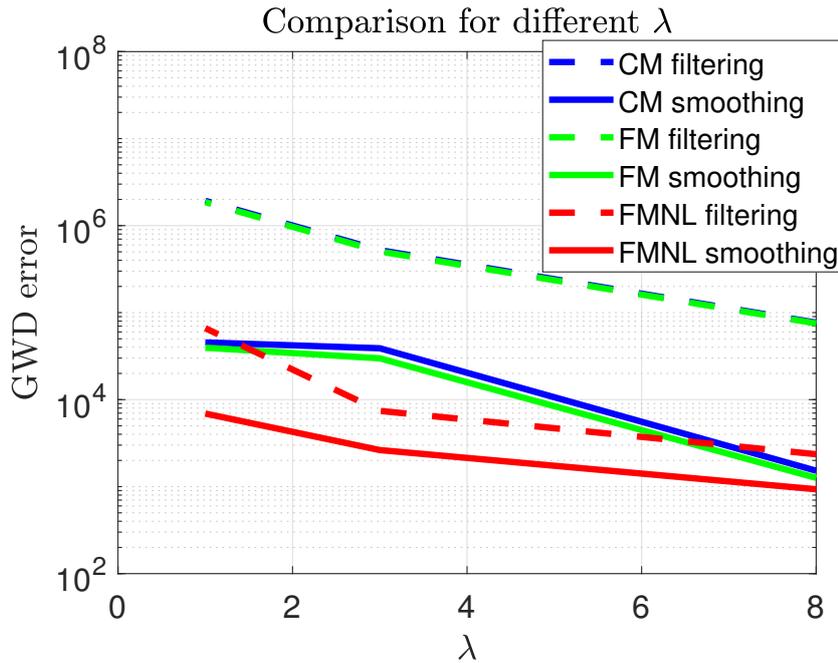
Note in Figure 5.21 that FMNL performance difference between smoothing and filtering does increase for decreasing  $\lambda$ . To see the GWD estimation error for FMNL in the case of  $\lambda = 1$  see Figure (a) 5.22. Here it is possible to see that the shape of the filtering estimation error, magenta curve, does resemble the shape of the prediction rather than that of the smoothing. As in the linear case, this is due to the higher importance of the prediction when the number of measurements gets small. This is also stressed by the fact that the filtering 95-percentiles are closer to



(a)  $P_d = 0.75$  and  $\lambda = 3$

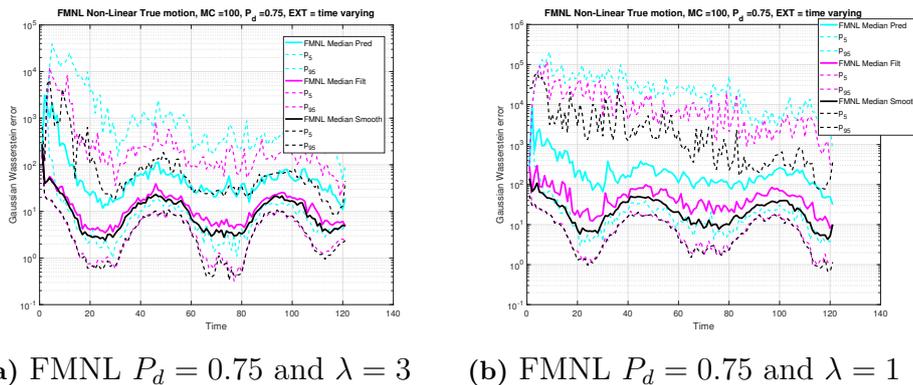
(b)  $P_d = 0.75$  and  $\lambda = 1$

**Figure 5.20:** (a) is the comparison between GWD estimation error for the smoothing of the three models in the case of nonlinear motion,  $P_d = 0.75$ , time varying extent and  $\lambda = 3$ . (b) is the comparison between GWD estimation error for the smoothing of the three models in the case of nonlinear motion,  $P_d = 0.75$ , time varying extent and  $\lambda = 1$ . The blue curves are CM model, green FM and red FMNL. Solid line are the median error and the dashed lines are the 5- and 95-percentiles.



**Figure 5.21:** The error comparison between the filtering and smoothing for different values of  $\lambda$  for nonlinear true motion.

the prediction percentiles that the smoothing and hence the performance difference between forward filtering and smoothing is large. The same properties can be found in the corresponding figure when  $\lambda = 3$ , see Figure (b) 5.22, but here it is more difficult to see the impact of the prediction to the filtering, since there are more measurements at each time instance. Which also makes the overall performance of the model better when  $\lambda = 3$  than  $\lambda = 1$ .

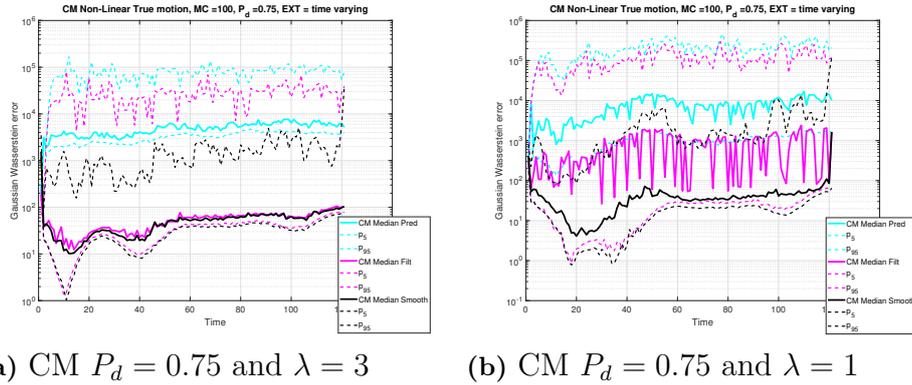


**Figure 5.22:** (a) is FMNL model for nonlinear true motion,  $P_d = 0.75$ ,  $\lambda = 3$  and the extent is varying with time. (b) is FMNL model for nonlinear true motion,  $P_d = 0.75$ ,  $\lambda = 1$  and the extent is varying with time. The black curve is the median GWD estimation error for each time instance based on 100 Monte-Carlo simulations of the smoothing estimate. The black dashed lines are the corresponding 5- and 95-percentiles of the GWD error. The magenta and cyan coloured lines have the same properties as the black ones but are based on the forward filtering estimation error and the prediction estimation error.

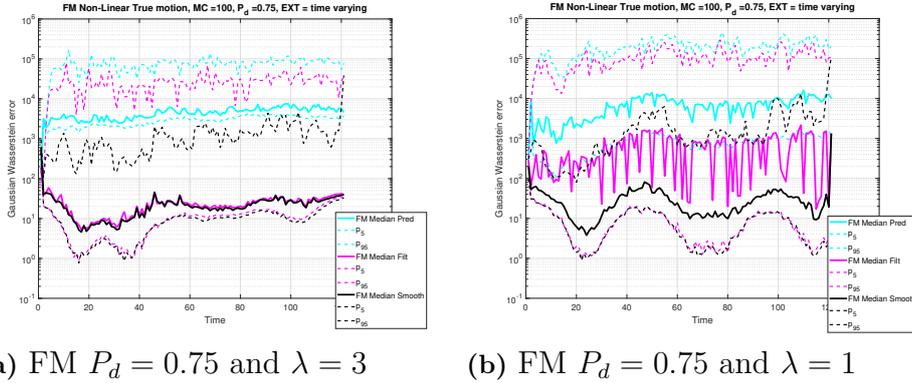
What is important from this analysis is that this shows how much the amount of information matters, i.e. the improvement that the smoothing can perform to on the filtering result is dependent on the amount of available information. Which is quite obvious, because in the extreme case where there are no detection's at all, the filtering estimate are equal to the prediction, and since there is not any future information either, the smoothing result will also be equal to the prediction and hence can not do any improvement from the filtering.

Here the benefit of the smoothing comes forth, since it is obvious that the smoothing copes with sparse information much better than the forward filtering. Just like in the case of information loss in Section 5.1.4. For both CM and FM models the filtering goes from resembling the smoothing shape in Figures (a) 5.23 and (a) 5.24 to look more like the prediction.

As expected, FMNL has a better performance in the nonlinear case when the information density decreases than CM and FM, and vice versa in the linear case. It can be seen in the comparison of the three smoothing models when  $\lambda = 3$  in Figure (a) 5.25 and when  $\lambda = 1$  in Figure (b) 5.25. One important notice is that here when the information loss is decreasing the performance of CM and FM does not resemble each other as much as they have in all previous scenarios. It might be surprising to see that FM yields a better performance than CM, although they are based on the same models. But as introduced in Section 3.1 FM and FMNL uses a different extent update formula, from [9], than the one used in CM, from [25]. So what is important is that it seems like in the nonlinear case, FM is a bit more robust to few measurements than CM.



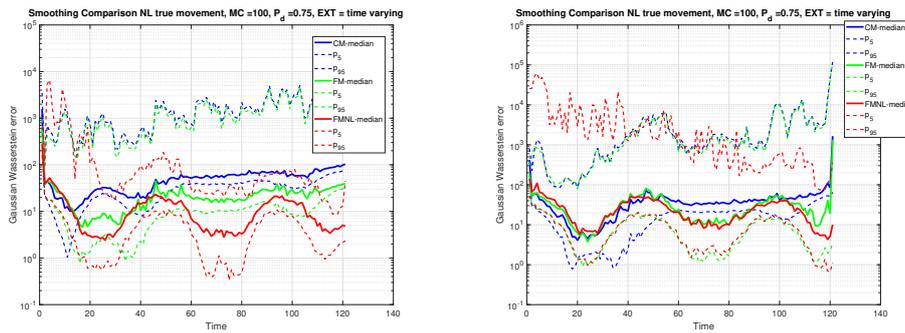
**Figure 5.23:** (a) is CM model for nonlinear true motion,  $P_d = 0.75$ ,  $\lambda = 3$  and the extent is varying with time. (b) is CM model for nonlinear true motion,  $P_d = 0.75$ ,  $\lambda = 1$  and the extent is varying with time. The black curve is the median GWD estimation error for each time instance based on 100 Monte-Carlo simulations of the smoothing estimate. The black dashed lines are the corresponding 5- and 95-percentiles of the GWD error. The magenta and cyan coloured lines have the same properties as the black ones but are based on the forward filtering estimation error and the prediction estimation error.



**Figure 5.24:** (a) is FM model for nonlinear true motion,  $P_d = 0.75$ ,  $\lambda = 3$  and the extent is varying with time. (b) is FM model for nonlinear true motion,  $P_d = 0.75$ ,  $\lambda = 1$  and the extent is varying with time. The black curve is the median GWD estimation error for each time instance based on 100 Monte-Carlo simulations of the smoothing estimate. The black dashed lines are the corresponding 5- and 95-percentiles of the GWD error. The magenta and cyan coloured lines have the same properties as the black ones but are based on the forward filtering estimation error and the prediction estimation error.

## 5.2 Multiple Extended Objects

The natural extension of tracking a single extended object, is to track several objects. As noted in Section 4.3 the number of objects are assumed to be known and fixed  $M = 2$ . The same principles hold for  $M > 2$ .

(a)  $P_d = 0.75$  and  $\lambda = 3$ (b)  $P_d = 0.75$  and  $\lambda = 1$ 

**Figure 5.25:** (a) is the comparison between GWD estimation error for the smoothing of the three models in the case of nonlinear motion,  $P_d = 0.75$ , time varying extent and  $\lambda = 3$ . (b) is the comparison between GWD estimation error for the smoothing of the three models in the case of nonlinear motion,  $P_d = 0.75$ , time varying extent and  $\lambda = 1$ . The blue curves are CM model, green FM and red FMNL. Solid line are the median error and the dashed lines are the 5- and 95-percentiles.

### 5.2.1 First Scenario - Object Distance

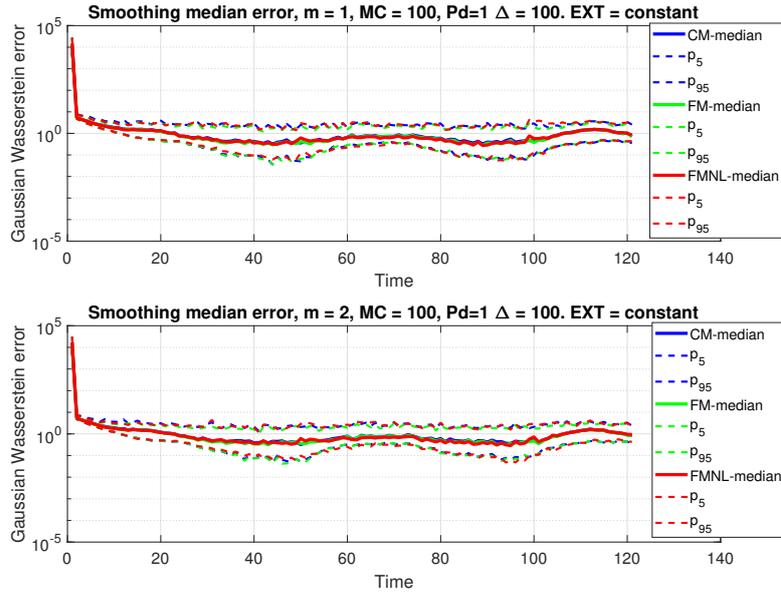
The data association problem that inevitable comes with the multiple object case give the models a more difficult situation to deal with. When the objects are separated by a large distance, the problem becomes easier and resembles the case of tracking two single objects. In Figure 5.26 the GWD smoothing estimation error is plotted for the two objects in the case of large object distance, denoted  $\Delta > 10\sigma$ ,  $P_d = 1$ , constant extent and  $\lambda = 8$ . Note that the error curves look alike and have the same order of magnitude as the ones in Figures 5.1 and B.1.

#### 5.2.1.1 Numerical Result

Model\Object	$m = 1$	$m = 2$
CM	14283.3395	16505.5526
FM	14337.0347	16849.5079
FMNL	14647.2667	16825.3441

**Table 5.7:** The numerical results for the case of  $\Delta = 10\sigma$ ,  $P_d = 1$  and  $\lambda = 8$  and the extent is constant.

The corresponding numerical results, calculated as in (4.3), for this case are shown in Table 5.7. Here the numerical errors seem to be a lot higher than for the single object case with similar settings in Tables 5.3 and 5.4. The reason for this can be seen by comparing the Figures 5.26 with 5.1 and B.1, where it can be noted that the median at time  $k = 1$ , i.e. the initial estimation error, is a lot higher for the multiple case. For the rest of the time the error's order of magnitude is similar. Hence it is not, with this performance measure, intuitively to compare the single object case with the multiple object case. The cause of the initialization errors are



**Figure 5.26:** The GWD smoothing estimation error for the case of  $\Delta > 10\sigma$ ,  $P_d = 1$ , constant extent and  $\lambda = 8$ . The upper figure is for object  $m = 1$  and the lower figure is for object  $m = 2$ . This case resembles the first scenario in Section 5.1.1, and as expected the performance is good and similar to both of the objects.

that in the single object case, the initialization is done more nicely. This is to avoid problems with the nonlinear motion model in FMNL. The CT model used in FMNL is sensitive to large initial errors for the heading angle. Depending on the turn-rate noise it takes some time to correct the error, and during that time the estimation is poor.

The initialization used for the multiple object case is done in a different way to instead avoid initial associations mistakes. The way this is done is that the objects are initialized randomly around the center of all measurement, thus in the middle of everything, and hence does the initial numerical error grow with increased true object distance. The initialization gives a bias in the numerical value.

To avoid this bias problem, the same numerical results as in (4.3) is used, but with omitted first time instance, since it is dependent of the true object distance. The new numerical results for the same case as in Table 5.7 then gets the following shape, see Table 5.8. Now the result is of the same order of magnitude as for the single object case and a comparison can be done with the single object case in Tables 5.1 and 5.2. Still, note that the initial error is included in the single object cases and hence tends to be a little larger than the numerical result for the multiple object case.

To represent the total error from a simulation, the sum of all the object errors will be used in the following scenarios. Thus if a comparison is made with respect to the single object case above, the numerical results here are approximately twice as big, since there are two objects.

Model\Object	$m = 1$	$m = 2$
CM	139.557	140.3281
FM	125.4169	127.7367
FMNL	134.7245	134.9685

**Table 5.8:** The numerical results for the case of  $\Delta > 10\sigma$ ,  $P_d = 1$ , extent is constant and  $\lambda = 8$  with the omitted first time instances that created a bias error.

### 5.2.1.2 Object Distances

When the distance gets smaller, the association becomes more tricky and there is always a risk of association mistakes. In the measurement model introduced in Section 3.3 there is no elaborated model for outliers or clutter measurements. This leads to the problem that when association mistake are made, the models consider them as fully trustworthy measurements as the other ones. The only thing that reduces the outliers' influence in the estimation is that each measurement is weighted with its association probability when creating the synthetic centroid measurement in (3.15). The further away a measurement is, the less probable is it and the less impact does it have on the estimate.

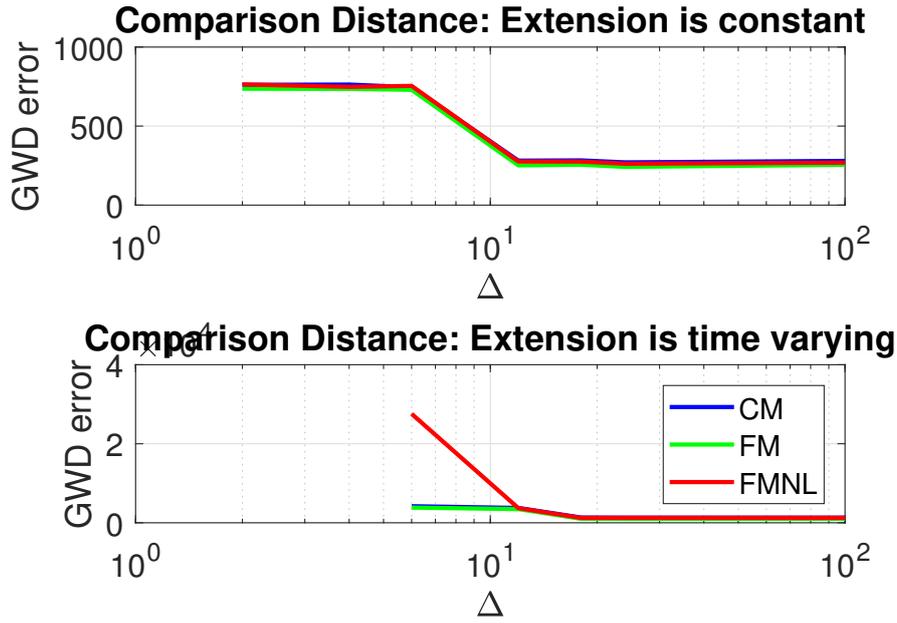
Below follows the results for different distances, namely  $\Delta \in \{> 10\sigma, 4\sigma, 3\sigma, 2\sigma, \sigma\}$  for the case where  $P_d = 1$ , extent is constant and  $\lambda = 8$ . The numerical error is summarized in Table C.1 and C.2 in Appendix C.2. A visualization of the two tables can be seen in Figure 5.27.

The distances between the objects that are used here are, as previously introduced, proportional to the standard deviation  $\sigma$ . When the true extent is varying with time, the value of the true  $\sigma$  also varies. To avoid that the true objects overlap but still are as close as possible the object distance has the constant numerical value of  $\sigma = 6$  which is the value of the standard deviation of the true extent when it is at its largest value. So all other distances are proportional to this fixed value  $\sigma$ .

In the situation when the true extents are constant over time, it is possible to narrow the true object distance down further. Thus the situation of  $\Delta = \frac{2}{3}\sigma$  and  $\Delta = \frac{1}{3}\sigma$  will be considered. Especially the situation  $\Delta = \frac{1}{3}\sigma$ , since it is the case where the constant true extents lies next to each other and is thus a similar difficult case as  $\Delta = \sigma$  for the time varying extent case.

A typical run when  $\Delta = 4\sigma$  can be seen in Figure (a) 5.28. Up in the left hand corner, the initial position for the models can be seen. They are all oriented around the mean of all the first measurements, and thus quite far away from the true trajectories. This is the step that cause the bias error before. All the estimations lie almost on top of each other when there are detections at every time. A zoomed in version can be seen in Figure (b) 5.28 where it is possible to see the three models' smoothing results, the blue CM, green FM and red FMNL. The black curves are the true trajectories and extents.

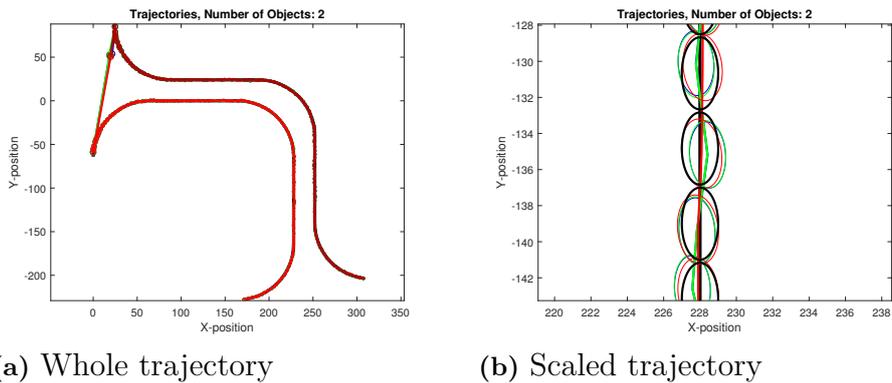
The distance  $\Delta = 4\sigma$  it is considered a large distance, both for the case of time varying extent and especially for the constant extent case, since 99.7% of the probability mass lies inside of  $3\sigma$ . In the upper figure of Figure 5.27 this is emphasized by that the numerical errors are approximately the same for the cases  $\Delta > 10\sigma$ ,  $\Delta = 4\sigma$ ,  $\Delta = 3\sigma$  and even  $\Delta = 2\sigma$ . For the time varying extent, in the



**Figure 5.27:** The upper figure is a visualization of Table C.1 where the extents are constant. The lower figure is a visualization of Table C.2 where the extents are time varying. For the case of varying  $\Delta$ ,  $P_d = 1$  and  $\lambda = 8$

lower figure of Figure 5.27 the error is approximately the same only for the cases when  $\Delta > 10\sigma$ ,  $\Delta = 4\sigma$  and  $\Delta = 3\sigma$ . The extent is generally larger in the time varying case, than in the constant case.

The errors are larger when the extent is time varying since it is a harder situation to track. The analysis is the same as for the single object case in Section 5.1.2.



(a) Whole trajectory

(b) Scaled trajectory

**Figure 5.28:** (a) is typical run with the case  $\Delta = 4\sigma$ ,  $P_d = 1$ , extent is constant and  $\lambda = 8$ . The black curves are the true trajectory and extent, the blue is CM, green FM and red FMNL. (b) is a zoomed in version of the trajectory in (a).

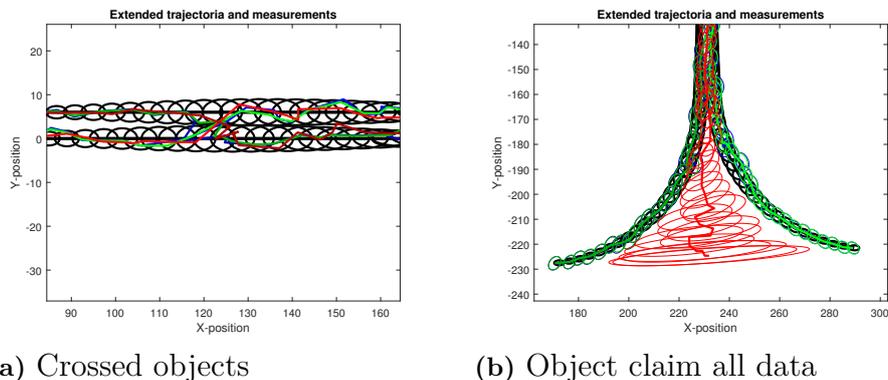
In both of the figures in Figure 5.27 the errors make a step, from  $\approx 300$  to  $\approx 750$  when  $\Delta$  goes from  $2\sigma$  to  $\sigma$  for constant extent and from  $\approx 1300$  to  $\approx 3700$

when  $\Delta$  goes from  $3\sigma$  to  $2\sigma$  for the time varying extent. This happens because when the distance is too small some unwanted situations occurs.

The main cause of the increased error when distance reduces is due to false associations. This means that one object gets another object's measurement. The association is performed with respect to both the position of the objects and the extents, in a Gaussian sense. The position of the centroid works as the mean and the extent works as the covariance of a multivariate Gaussian distribution. Thus an object with a larger extent is more likely to have more measurements. But in this thesis it is assumed that the measurement rate is constant over time and equal for all objects no matter how the extent grows. This is due to the fact that the true objects are identical.

The true track consist of both linear and nonlinear parts, the different models thus have both instances with advantage and with disadvantage due to their different motion models. The disadvantage are mainly that the prediction of the trajectory might end up close to the other object, or sometimes even at the other object. This makes the calculations of the association cumbersome and may lead to unwanted situations.

There are mainly two situations that occurs. The first situation is that the objects switch place with one another, i.e. that they take over each others measurements, see Figure (a) 5.29. The second situation is that one of the object might claim all the measurements and thus become one large object, while the other object gets small since it gets no data, see Figure (b) 5.29. The claiming situations do not occur that often when  $P_d = 1$ , even though  $\Delta = \sigma$  for the time varying extent and  $\Delta = \frac{1}{3}\sigma$  with constant extent.



**Figure 5.29:** (a) is an example situation that might occur when object gets too close to each other. What happens is that they may switch measurements with each other and thus switch trajectory. This is for the case when  $P_d = 1$ ,  $\Delta = \sigma$ , the extent is time varying and  $\lambda = 8$ . In this plot are the extent estimates omitted to make it more visible how a switching of state might look like. (b) is an example when one object, here it occurred for FMNL in red, claims all the measurements and dominates. This is for the case when  $P_d = 1$ ,  $\Delta = \sigma$ , the extent is time varying and  $\lambda = 8$ .

In figures (a) and (b) in Figure 5.30 the GWD estimation error for the smooth-

ing of all three models can be seen. In (a) the extent is constant and the distance is such that the true extents are touching,  $\Delta = \frac{1}{3}\sigma$ . In (b) the extent is varying with time and the distance is also such that the true extents are touching,  $\Delta = \sigma$ .

These figures indicate that the median of the error do not tell the full truth. What happens in Figure (a) 5.30 is the following. FMNL, red curves, has a disadvantage when the true track makes a transition from moving in a straight line to suddenly move in a curve with constant turn-rate. The turn rate makes an instant step from zero to a nonzero value. At the end of the turn the same thing happens again where the true trajectory transition from moving in a curve towards moving straight.

At the true trajectory in Figure 4.2, this transitions occurs at four places for each object. When the object distance is small the two transitions in the middle are extra tricky, around time instance 60. In Figure (a) 5.30 when  $\Delta = \frac{1}{3}\sigma$  there is a chance that an object will claim all the measurements. The other object gets stuck within an endless prediction, without any filtering updates (since it does not get any measurements), while the other object carries on as one big object, see the example in Figure (b) 5.29. What happens then is that the error for the omitted object grows linearly with time and yields the logarithmic trend that can be seen by the 95-percentiles.

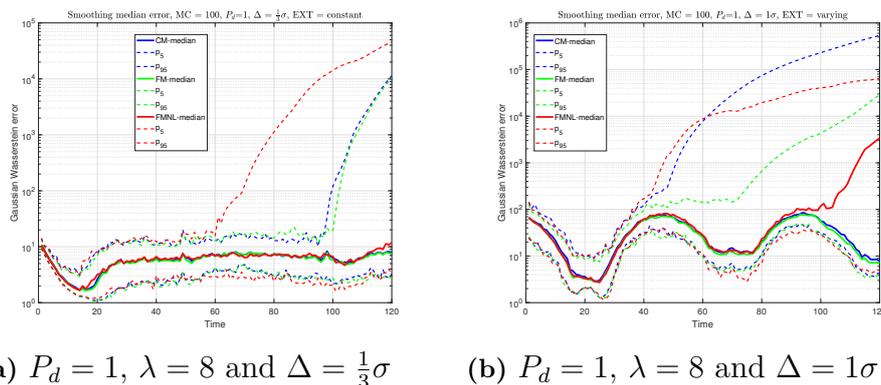
For the other two models in Figure (a) 5.30, CM in blue and FM in green, there is more common that a switch of trajectories occurs. So when the true tracks starts to move away from each other, at time instance around 100, the errors for both objects grow linearly in time and thus yield the logarithmic shape of the corresponding 95-percentiles. An example of the switching can be seen in Figure (a) 5.29.

The risk of the unwanted situations are higher in the case when the extent is varying with time. This is due to the fact that at some time instances the extents grow towards each other. In Figure (b) 5.30 for all three models, in some of the simulations, there are objects that claim all the measurements at around time instance 40. What happens at that time instance is that it is the first point where the true extents grow in to each other and touches. The reason that both the blue CM and green FM error 95-percentile wanders of to increasing values are due to its linear prediction. As time goes on, the object that did not get any measurements will only move further and further away in a straight line while the red FMNL prediction stays in a constant turn and thus stays closer to the true track and yield a smaller error.

It seems that FM is the model that deal with this situation the best since it is more common that a switching occurs rather than that an object claims all the measurements. FM seems to be better of keeping track of the two objects and does not mix them together like the other two models.

### 5.2.2 Second Scenario - Probability of Detection

In the single object case the effect of missed detections was mainly that the estimation lost track of the true object. But here in the multiple object case, losing track of the true object means that the two unwanted situations mentioned above in Section 5.2 become more common. Since each object has, at each time instance, a



**Figure 5.30:** (a) is the comparison between GWD estimation error for the smoothing of the three models in the case of constant extent,  $P_d = 1$ ,  $\lambda = 8$  and  $\Delta = \frac{1}{3}\sigma$ . (b) is the comparison between GWD estimation error for the smoothing of the three models in the case of time varying extent,  $P_d = 1$ ,  $\lambda = 8$  and  $\Delta = 1\sigma$ . The blue curves are CM model, green FM and red FMNL. Solid line are the median error and the dashed lines are the 5- and 95-percentiles. The error is the sum over all objects.

probability of detection, it might happen that either; all objects are detected, only one or non of the objects are detected.

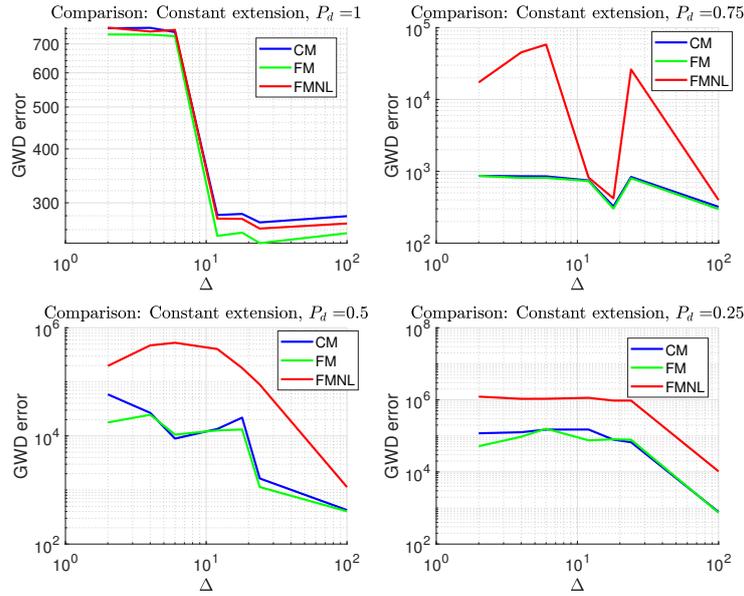
If all the objects get detected, the situation is the same as in the first scenario in Section 5.2. If non of the objects are detected there can be no association problems, since there is nothing to associate. But if only one of the objects get detected and the object distance is small enough, the unwanted situations are likely to occur.

There is no probability of detection model included in the association calculation, so at each time there is a detection only for one or both objects, the association calculations always believes that all objects are detected. In other words, each object will claim measurements as long as there are any to associate with.

The numerical result for different cases of object distances and probability of detection are listed in Tables C.3, C.4 and C.5 in Appendix C.3.1. To get a visual comparison between the three tables, see Figure 5.31. Here it is clear that CM and FM generally perform better than FMNL, for this specific true trajectory.

When  $P_d = 1$  in the upper left corner, the performance is similar between all the models. But as  $P_d$  reduces, in the other three figures in Figure 5.31, the difference between the models becomes more distinct. The bad result of FMNL resembles its performance in the single object case when the true trajectory where linear. So apparently the nonlinear transition model in FMNL is very disadvantageous for this particular true trajectory. When the distance decreases and there are absent measurements, FMNL estimation gets worse, and the unwanted situation is much more common. The reason that the error decreases again when the distance gets small, is due to that the error is proportional to the true object distance.

It is also possible to see that FM during most of the cases perform slightly better than CM. Since the scales in Figure 5.31 are logarithmic one can note that the change in  $P_d$  has a much stronger effect on the error than the change of the object distance  $\Delta$ . The impact a reduced  $P_d$  has on all the models' performances

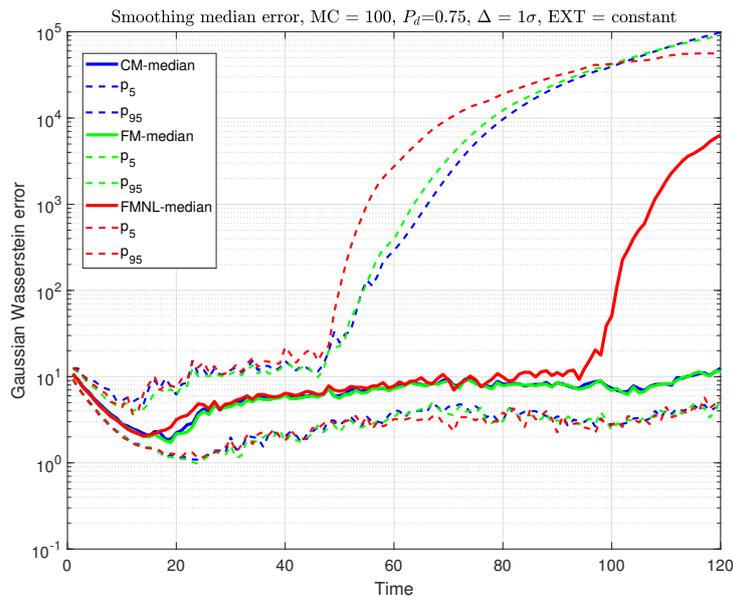


**Figure 5.31:** A visualization of the three Tables C.3, C.4 and C.5, for different values of the object distance  $\Delta$  and Probability of detection  $P_d$ . CM is blue, FM green and FMNL red. The plot in the upper left corner is for different values on the  $\Delta$  and  $P_d = 1$ . The upper right most figure is different  $\Delta$  and  $P_d = 0.75$ . Lower left most figure is different values on  $\Delta$  and  $P_d = 0.5$  and the lower right figure is varying  $\Delta$  and  $P_d = 0.25$ . This is for constant extent.

in the multiple object case is analogous to the analysis in Sections 5.1.3 and 5.1.4. That both the smoothing and filtering must rely more on the prediction and hence the risk of unwanted situations are much larger.

Note the trend of the error in the curves, that for a reduced  $P_d$  the object distance can not be too small for the models to give a reasonable performance. In Figure 5.32 the three models performance is shown in the case when  $\lambda = 8$ ,  $P_d = 0.75$  and  $\Delta = \sigma$ . When the extent is constant  $\Delta = \sigma$  corresponds to the case when the true three standard deviations touches and it is harder to see the discrepancy between the measurement clusters. The risk of association mistakes of the kind such that one object claims all the measurements are more common and thus yield the shape of the 95-percentiles in Figure 5.32. When  $P_d$  gets even smaller the prediction performance is crucial, and since the true track consists of both linear and nonlinear parts, non of he models has any large advantage or disadvantage with respect to their motion models. Thus the risk of one object will claim all measurements are more common even though the distance is quite large between the objects.

For the situation when the extent is varying with time, the corresponding numerical results can be seen in Tables C.6, C.7 and C.8 in Appendix C.3.2. To see a visual comparison between the three models for all the tables see Figure 5.33. From the Figure 5.33 one can see that a decrease in the object distance does not has a great impact on the result as the changing in  $P_d$  does. Here the difference between the models are similar to the situation in Figure 5.31, where FMNL gives the worst performance. FMNL gives a similar performance for all the cases when  $P_d = 0.75$ ,

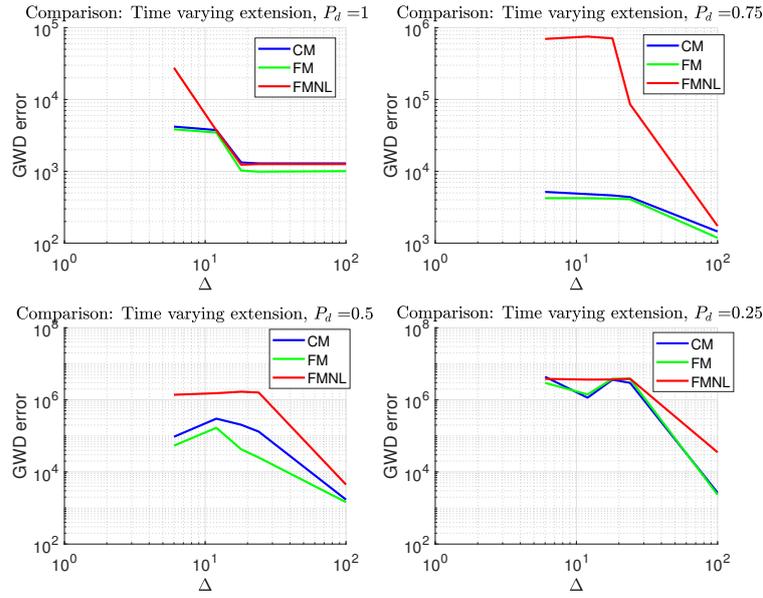


**Figure 5.32:** The three models smoothing performance in the case of constant extent,  $P_d = 0.75$ ,  $\lambda = 8$  and  $\Delta = \sigma$ . Blue curves are CM, green FM and red FMNL. Solid lines are the median error per time instance and the dashed lines are the corresponding 5- and 95-percentiles.

$P_d = 0.5$  and  $P_d = 0.25$ .

For each value of  $P_d$  there are a corresponding value of the object distance where the performance gets significantly worse. An example case, when  $P_d = 0.75$ ,  $\Delta = 4\sigma$  and  $\lambda = 8$ , can be seen in Figure 5.34. Here, although the distance is quite large, at the time around  $k = 40$ , then the extent is growing toward each other there is a high risk of an object claiming all the measurements and/or a switching of objects.

The reason why the solid red FMNL error curve increases drastic at the late time instances is due to the fact that since the track is difficult for the nonlinear motion model in FMNL the unwanted situations are more common. They are so common that they occur at the majority of the 100 Monte-Carlo simulations, especially the switching situation. Thus at the end, either the error is already large due to that one object is dominating or the error increases in the end because there have been a switch of trajectories and the trajectories are moving apart.



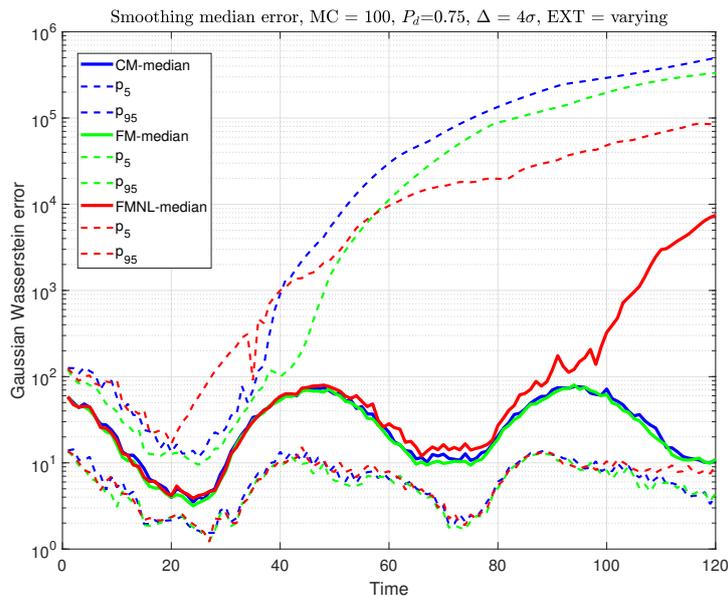
**Figure 5.33:** A visualization of the three Tables C.6, C.7 and C.8 for different values of the object distance  $\Delta$  and Probability of detection  $P_d$ . CM is blue, FM green and FMNL red. The plot in the upper left corner is for different values on the  $\Delta$  and  $P_d = 1$ . The upper right most figure is different  $\Delta$  and  $P_d = 0.75$ . Lower left most figure is different values on  $\Delta$  and  $P_d = 0.5$  and the lower right figure is varying  $\Delta$  and  $P_d = 0.25$ . This is for time varying extent.

### 5.2.3 Third Scenario - Measurement Rate

As in Section 5.1.5, the direct consequence of the measurement rate's  $\lambda$  decrease is that the expected number of obtained measurements during a detection is reduced. As noted in Section 4.2.1.2, the probability of getting no measurements at all depends on both the probability of detection but also on the sampling of  $n_k$ . When  $\lambda \in \{8, 3, 1\}$  the probability of getting no measurements, given that there is a detection are

$$\begin{aligned}
 P(\text{No measurements}) &= p_{n_k} = P(n_k = 0) = \frac{\lambda^{n_k}}{n_k!} e^{-\lambda} \Big|_{(n_k=0, \lambda=8)} = e^{-8} \approx 0.034\% \\
 P(\text{No measurements}) &= p_{n_k} = P(n_k = 0) = \frac{\lambda^{n_k}}{n_k!} e^{-\lambda} \Big|_{(n_k=0, \lambda=3)} = e^{-3} \approx 4.979\% \\
 P(\text{No measurements}) &= p_{n_k} = P(n_k = 0) = \frac{\lambda^{n_k}}{n_k!} e^{-\lambda} \Big|_{(n_k=0, \lambda=1)} = e^{-1} \approx 36.788\%
 \end{aligned} \tag{5.1}$$

The total probability of getting no measurements for different values of  $P_d$  and  $\lambda$  are shown in Table 5.9. Here it can be noted that the reduction of  $\lambda$  does not only reduces the expected information density of each detection but also has the same impact on the obtained measurements as the  $P_d$ . So to examine the impact of the measurement rate in the multiple object case, it is investigated for the case when  $P_d = 1$ . The numerical results for different values of the object distance and



**Figure 5.34:** The three models smoothing performance when the extent is varying with time,  $P_d = 0.75$ ,  $\Delta = 4\sigma$  and  $\lambda = 8$ . The solid lines are the median error over time, and the dashed lines are the corresponding 5- and 95-percentiles. Blue is CM, green FM and red FMNL.

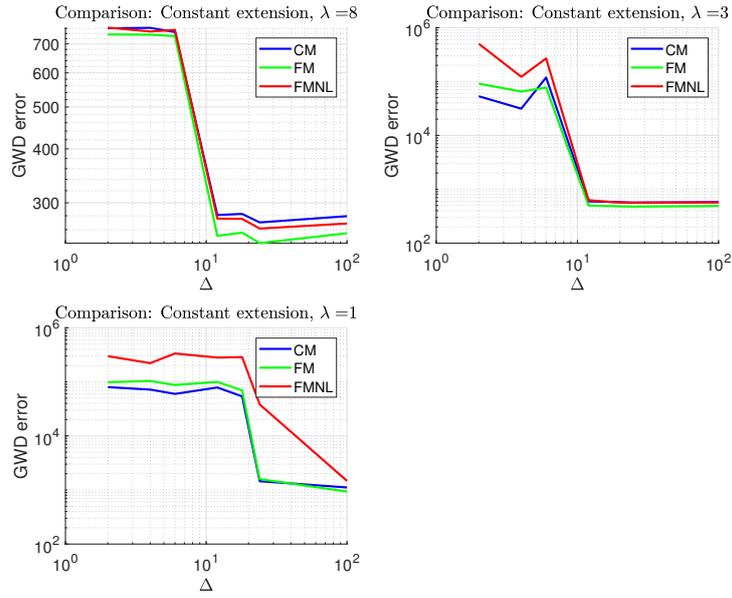
$P_d \backslash \lambda$	$\lambda = 8$	$\lambda = 3$	$\lambda = 1$
$P_d = 1$	0.033%	4.979 %	36.788 %
$P_d = 0.75$	25.025 %	28.734%	52.590 %
$P_d = 0.5$	50.017 %	52.489 %	68.394%
$P_d = 0.25$	75.008 %	76.245 %	84.197 %

**Table 5.9:** The Probability of getting no measurements for different values of  $P_d$  and  $\lambda$ , based on  $P(\text{No measurements}) = (1 - P_d)p_{n_k} + (1 - P_d)(1 - p_{n_k}) + P_dp_{n_k}$ .

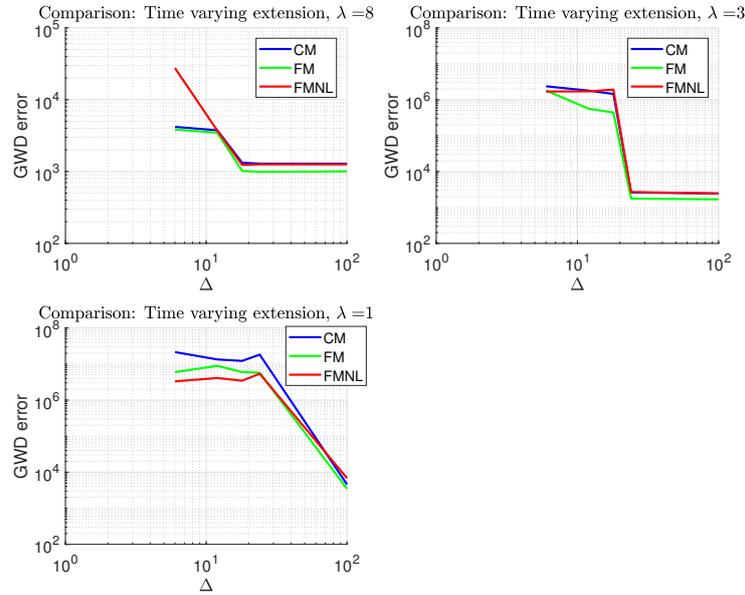
measurement rate, while the extent is constant can be viewed in Tables C.9, C.10 and C.11 in Appendix C.4.1. To get a visual comparison between the three tables see Figure 5.35. Here the blue curves are CM, green FM and FMNL red.

The corresponding tables for the same situation but when the extent is varying with time can be seen in Tables C.12, C.13 and C.14 in Appendix C.4.2. For an easier comparison between the three tables see Figure 5.36. In both the Figure 5.35 and 5.36 there is a distinct step in the curves where the error makes a significant change in the error. So clearly for a smaller value of  $\lambda$  the distance between the object can not be too narrow without any of the unwanted situation occurs.

In the case when the extent is constant, in Figure 5.35, CM yields the best performance over all setups, and FM gives a similar performance. Similar to the case in Section 5.2.2 where FMNL does not give an equally good performance as the other two. In Figure 5.35, when  $\lambda = 8$  and  $\lambda = 3$  the distance may go down to  $\Delta = 2\sigma$ . But when  $\lambda = 1$  the large error jump occurs when  $\Delta$  goes from  $4\sigma$  to  $3\sigma$ . When the number of obtained measurements decreases, the measurement clusters



**Figure 5.35:** The visualization of the three Tables C.9, C.10 and C.11. With  $P_d = 1$ , constant extent and for different values of  $\Delta$  and  $\lambda$ . The upper left figure is  $\lambda = 8$ , upper right  $\lambda = 3$  and lower figure  $\lambda = 1$ .



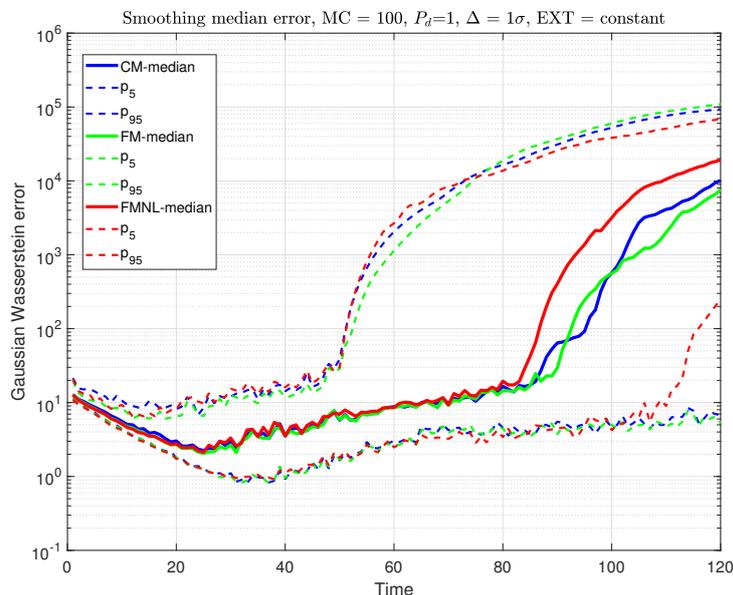
**Figure 5.36:** The visualization of the three Tables C.12, C.13 and C.14 with  $P_d = 1$ , time varying extent and for different values of  $\Delta$  and  $\lambda$ . The upper left figure is  $\lambda = 8$ , upper right  $\lambda = 3$  and lower figure  $\lambda = 1$ .

are not easily distinguished.

In Figure 5.37 the GWD errors for all three models for this boundary case is shown,  $P_d = 1$ ,  $\Delta = \sigma$  and  $\lambda = 3$ . The shape of all models curves indicate that there are often an object that claims all the measurements. When  $\lambda = 1$

the only situations when the models gives a reasonable performance are when the object distance is far away from each other, i.e.  $\Delta > 10\sigma$  and  $\Delta = 4\sigma$ . Here the performance is similar to the case of single object tracking in Section 5.1.5.

According to Table 5.9 when  $\lambda = 1$  and  $P_d = 1$  on average 36% of the time instance there are no measurements and when measurements are obtained, they are few. This means that the prediction plays a greater part and when the object gets closer, the risk of association mistakes increases. Since the average number of measurements, over the time instances when there are a detection, is equal to 1, the association is tricky. The risk of an object claim all measurements are huge since there often is only one measurement. Thus it is common that already in early time instances there is one object that dominates. This is also true for when  $\lambda = 3$  in Figure 5.37, but the trend is more obvious for  $\lambda = 1$ .



**Figure 5.37:** The comparison of all three models in the case of constant extents at the boundary case when  $P_d = 1$ ,  $\Delta = \sigma$  and  $\lambda = 3$ . Blue curve are CM, green FM and red FMNL. The solid lines are the median error at each time instance and the dashed lines are the corresponding 5- and 95-percentiles.

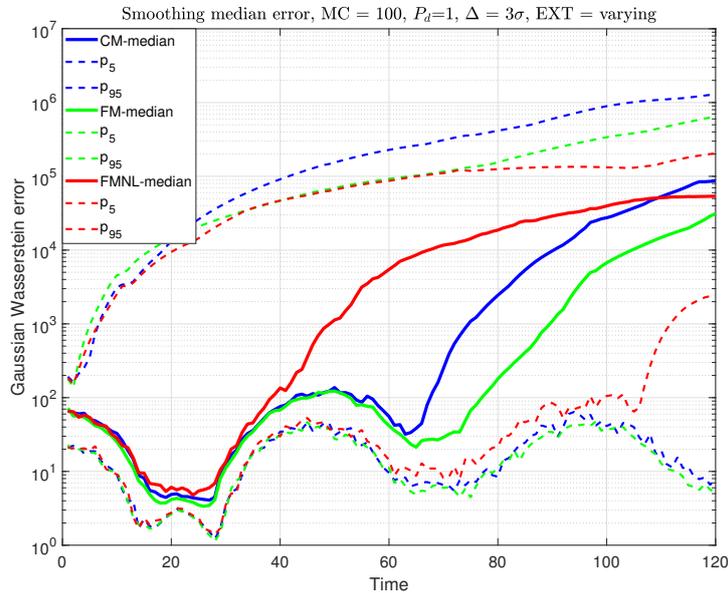
Comparing the constant extent with the case when the extent is varying with time, it is possible to note that there is a interesting difference. In the lower Figure 5.36 the performances between the models are reversed. Thus similar to the single object case for nonlinear true motion in Section 5.1.5 CM yields a worse performance. Here especially when the the object distance is small does FMNL performs best.

Overall, the error trend is similar to the constant extent case. That when  $\lambda = 8$  the object may be close to each other and the largest risk is that there is a switching of measurements. As soon as  $\lambda$  reduces, the object must be further away for the models to be able to track the objects. The case when  $\lambda = 3$  and  $\Delta = 3\sigma$  corresponds to the situation when the true 3 standard deviations touches. In Figure 5.38 the performance of the models can be seen, and it is clear that when there are

few measurements and the objects gets close, the association becomes poor. The most common situation is here the same as for the constant extent case, that one object claims all the few measurements available, and thus the other object wanders off and yields the large error.

When  $\lambda = 1$  there are so few measurements that precisely as in the constant extent case, one object claim them all as soon as the object are close to each other. So here there is only when  $\Delta > 10\sigma$  that the models gives a reliable result.

The differences in performance between the three models are smaller here in the case of decreased  $\lambda$  and fixed  $P_d = 1$  than in the case of decreased  $P_d$  and fixed  $\lambda = 8$ . The reason for this is that since there are detections at each time instance FMNL does not need to rely to much on its nonlinear transition model, that is disadvantageous for this deterministic true track. This yield a better performance than in the situation of missed detections.



**Figure 5.38:** The comparison of all three models in the case of time varying extents at the boundary case when  $P_d = 1$ ,  $\Delta = 3\sigma$  and  $\lambda = 3$ . Blue curve are CM, green FM and red FMNL. The solid lines are the median error at each time instance and the dashed lines are the corresponding 5- and 95-percentiles.

# 6

## Conclusion

Here the conclusions, based on the analysis and results in previous chapters, are presented. First for the case of only one object, then for the case of multiple objects. Afterwards follow some suggested future works.

### 6.1 Single Extended Object

In the ideal situation when there are always available measurements and they are many, the smoothing calculations does not pay off in performance, compared to the three models' corresponding forward filtering results. But as soon as the situation includes more complex setups, such as; changing extent, probability of detection smaller than one or/and measurement rate closer to one, the difference between all the three forward filterings and smoothings increases. It is then justified to spend the extra computational power on the smoothing algorithms.

The models, as they are developed in this thesis, rely on assumptions about the extent. None of the models included a model where the extent might change in shape over time, and thus it resulted in a difficult task to cope with. The nonlinear transition model used in FMNL only considers rotations of the extents and no change of the extents' shapes. But not surprisingly, all the smoothing results still performed better than the filter results.

The probability of detection makes a distinct difference between the estimations. Since it confines the measurement information needed for the filtering algorithms to perform good estimates. As mentioned in Section 4.2.1.2 the reasons why a detection could be missed are many and also quite common. Thus it is an important aspect to include in an analysis like this. The direct impact of the detection probability is that the prediction gets more important, especially for the forward filtering since it is equal to the prediction when a detection is missed. Here, the difference between smoothing and filtering becomes very distinct.

The impact of a reduced value of the measurement rate, decreases the precision of the the measurement update. In contrast to the probability of detection, where the filtering becomes equal to the prediction, here the filtering only resembles the the prediction more and more the fewer measurement that are obtained. As noted in Section 5.2.3 a low measurement rate has also a similar influence as the  $P_d$ , since there might be a detection but the number of generated measurements are equal to zero.

For the situation when the expected information density of each detection is reduced, it is more clear that the smoothing does contain more information at

each time instance and does in general yield a better performance than the forward filtering. But the sensitivity to information density is larger than the sensitivity towards probability of detection, since each scan contains more information once they occur in the latter case. The number of obtained measurements are a key element in extended object tracking, since there are more degrees of freedom for the estimate than in the case of point object tracking.

## 6.2 Multiple Extended Objects

The evaluation performed in this thesis for the multiple object case is based on a deterministic track. The reason for this is due to the fact that it enables a better control of the new situations that arises when dealing with multiple objects. As mentioned in Section 3.6.2 the data association problem that comes with the multiple objects gives new perspective to the difficulties mentioned above. The parameter that has the largest influence of the complexity of the association problem is the object distance. To be able to control the distance as well as possible, the track was chosen to be deterministic.

The track was designed in such a fashion that it was supposed to be beneficial for all three models, and in the same time enable the evaluation of what happens when objects move close to each other and then move apart.

In the case when only the distance changes, all models performed almost identical when the extent is constant. For time varying extent, the extent estimation gets more difficult, and the association as well, since it depends on the extent estimation. If the objects extent estimate are of different sizes, the difference can make a crucial association difference in the narrow case. All models perform similarly except for the closest case where FMNL's performance diverge from the other two. This is a consequence of that the true track, even though it was designed to be beneficial for all three models, it is the least beneficial for FMNL. Thus in the narrow cases CM and FM performs significantly better than FMNL.

The probability of detection makes the association difficult in the sense that the association does not have an elaborated strategy to use when there are no detection for one of the object. This leads to the drastic step change in performance of the models, in Figures 5.31 and 5.33, when the distance gets to small. Once again FMNL shows the weakest performance, and one of the main reasons are due to its disadvantageous motion model for this specific track, no matter if the extent is constant or varies in time.

When the measurement rate gets small, each obtained scan contains on average less information and thus, similarly to the case of probability of detection, the prediction plays a greater part in the estimation. So in a similar way that the probability of detection makes the association difficult, the measurement rate yield some additional difficulties. In addition to the missed detections, each detection does also contain a smaller number of measurements. This makes each measurement more important and increases the estimation's sensitivity to association mistakes. The object distance can for this reason not be too small, if the measurement rate is small. This is why it is possible to see a similar significant step in the error as for the probability of detection, in Figures 5.35 and 5.36, when the object gets too close

to each other.

## 6.3 Summary

From the analysis, it is clear how important the general information about the tracking applications is. In other words, to have at least a coarse knowledge about how the track is going to be benefits the choice of model drastically. The three models are evaluated in both linear and nonlinear environments, and as expected, the linear models outperform the nonlinear model in the linear environment, while the nonlinear model outperforms the linear models in the nonlinear environment. But it must be emphasized that nonlinear cases are more difficult to cope with. So even though the nonlinear model, FMNL, has a clear advantage when the environment is nonlinear, the calculations are based on a series of approximation to make the algorithm tractable.

So when dealing with nonlinear cases, one needs to be more careful since it is more unforeseeable and unforgivable than the linear cases. More knowledge about the application is thus needed to make sure that the result is reliable. If this is possible, then the nonlinear model will perform much better than any linear model. Another way to put it is that the linear models are more robust, but lack of expertise in the nonlinear case.

The overall performance of CM and FM are almost the same while FMNL drags behind in both the single object case and the multiple object case. It is important to emphasize that there are some situations, especially for the multiple object case, when FM outperforms CM. So in general, in the experiments performed in this thesis, FM seems to be the most reliable and robust model to work with.

## 6.4 Future Work

There are of course a lot of extensions and improvements that can be made to the work in this thesis. Here, some of the possibilities are presented.

### 6.4.1 Models

The motion model used in FMNL is the CT model with its velocity vector in polar form. It is very sensitive to initialization of the heading angle and turn-rate. To make FMNL more robust and remove the need of a good initialization, such as prior knowledge about the heading, a different nonlinear motion model should be utilized in FMNL, e.g. the coordinated turn model with Cartesian velocity vector.

The measurement model used in this thesis neglects the pure sensor noise and the impact of scattering points, that are in many real world applications a common case. Thus to make the analysis of the models performance more general, a more general measurement model would be important to consider. Right now, CM with its extent dependency on the kinematic state performs well, but that is not sure to be the case when sensor noise is included.

The assignment calculation does not consider the situation of missed detection and always tries to assign measurements to all objects. To make the performance more robust and reliable, it would be beneficial to include a model for the association calculation where the possibility of missed detection is included.

Here the number of obtained measurements, given a detection, is modeled as a Poisson random variable with a constant arrival rate. For a more general setting it would be possible to consider a model where the measurement rate is dependent on the true states, e.g., such that if the extent grows then the expected number of obtained measurements increases as well. Another possibility would be to consider if the object moves closer to the sensor then the number of expected measurements could increase. In this thesis the sensor position is omitted and is assumed to be external, such as a satellite.

The Poisson assumption about the number of obtained measurements are not always the best model when modeling a real sensor, since the Poisson distribution has a variance equal to the mean, i.e., if the mean is high, then so is the variance. So given a detection the number of measurements varies much from detection to detection. A more reasonable model would maybe be one where the variance is smaller. If there is knowledge about what kind of sensor that is being utilized in the application, than an even more specific model is beneficial.

Of course it is also possible to utilize a complete different extent model, than the random matrix approach used in this thesis. Note though that it would also require new derivation of the smoothing algorithms.

## 6.4.2 Computational Tools

To solve the data association problem, the EM algorithm is utilized in this thesis. It is a strong tool but has some drawbacks. The major issue with EM is that it easily gets stuck in local maxima. So to get an even more robust and general setup it would be beneficial to consider some of the possible extensions of the EM algorithm such as the *Expectation-Conditional Maximization algorithm (ECM)* or *Expectation-Conditional Maximization either algorithm (ECME)* from [31]. There is also the possibility of using some other *probabilistic multi-hypothesis tracking (PMHT) algorithm* or *probability hypothesis density (PHD) algorithm*, e.g. from [43], [30] or [10].

In this thesis the K-means algorithm is used to initialize the first time instance for the EM algorithm. For the initialization to work, all objects must be detected the first time instance. Hence a natural extension is to include a initialization for the assignments that does not need that all objects are detected simultaneously at the first time instance.

In the case when  $\lambda$  decreases the importance of a good prediction increases. Thus the assumption made regarding the simple prediction step for the extent in [25] is violated if  $\lambda$  gets small. So it might be worth the extra computational power to utilize a more advance dynamical extent model.

### 6.4.3 Ground Truth

In the cases dealt with in this thesis, it is assumed that the number of objects are known and fixed. Naturally the extension for this is to include a model of unknown number of objects. Thus the possibility of object spawning and termination could also be included to make the case more general, such that all the objects do not need to be present during the whole batch.

### 6.4.4 Evaluation

The Gaussian Wasserstein distance measure used in this thesis is a good performance measure, since it includes both the kinematic state and the extent. As noted in Section 5.2, the median of the error over all the Monte-Carlo simulations does not tell the full truth in the multiple object case. Thus the inclusion of a better trajectory performance metric that enables separate examination of both the location error and the switching error would be interesting in the multiple object case.

According to the analysis above the performance between the models in the multiple object case FMNL does not get a fair evaluation. Thus to consider the situation similar to the single object case, where the true trajectory is, both linearly and nonlinear, randomly generated at each Monte-Carlo iteration, would yield a more statistically reliable result.



# Bibliography

- [1] *The handbook of data mining. Human factors and ergonomics.* Lawrence Erlbaum Associates, Publishers, Mahwah, N.J., 2003.
- [2] M. Baum and U.D. Hanebeck. Shape tracking of extended objects and group targets with star-convex RHMs. *2011 Proceedings of the 14th International Conference on Information Fusion (FUSION)*, page 1, 2011.
- [3] Dimitri P Bertsekas. *Introduction to probability.* Athena Scientific, Belmont, Mass., 2002.
- [4] Christopher M. Bishop. *Pattern recognition and machine learning.* Springer, New York, NY, 2006.
- [5] W.R. Blanding, P.K. Willett, and Y. Bar-Shalom. Multiple target tracking using maximum likelihood probabilistic data association. *2007 IEEE Aerospace Conference*, page 1, 2007.
- [6] Steven Bordonaro, Peter Willett, Yaakov Bar-Shalom, Marcus Baum, and Tod Luginbuhl. Extracting speed, heading and turn-rate measurements from extended objects using the EM algorithm. *2015 IEEE Aerospace Conference*, page 1, 2015.
- [7] Xiaomeng Cao, Jian Lan, and X. Rong Li. Extension-deformation approach to extended object tracking. *2016 19th International Conference on Information Fusion (FUSION)*, page 1185, 2016.
- [8] M. Feldmann and D. Franken. Advances on tracking of extended objects and group targets using random matrices. *2009 12th International Conference on Information Fusion*, page 1029, 2009.
- [9] M. Feldmann, Dietrich Fränken, and W. Koch. Tracking of extended objects and group targets using random matrices. *IEEE Transactions on Signal Processing*, 59(4):1409–1420, 2011.
- [10] K. Granstrom, C. Lundquist, and O. Orguner. Extended target tracking using a gaussian-mixture PHD filter. *IEEE Transactions on Aerospace and Electronic Systems*, 48(4):3268–3286, 2012.
- [11] K. Granstrom, C. Lundquist, and U. Orguner. Tracking rectangular and elliptical extended targets using laser measurements. *2011 Proceedings of the 14th International Conference on Information Fusion (FUSION)*, page 1, 2011.

- [12] K. Granstrom and U. Orguner. A PHD filter for tracking multiple extended targets using random matrices. *IEEE Transactions on Signal Processing*, 60(11):5657–5671, 2012.
- [13] K. Granstrom and U. Orguner. On spawning and combination of extended/-group targets modeled with random matrices. *IEEE Transactions on Signal Processing*, 61(3):678–692, 2013.
- [14] Karl Granstrom. An extended target tracking model with multiple random matrices and unified kinematics. 2014.
- [15] Karl Granstrom, Marcus Baum, and Stephan Reuter. Extended object tracking: Introduction, overview and applications. 2016.
- [16] Karl Granstrom, Maryam Fatemi, and Lennart Svensson. Gamma gaussian inverse-wishart poisson multi-bernoulli filter for extended target tracking. *2016 19th International Conference on Information Fusion (FUSION)*, page 893, 2016.
- [17] Karl Granstrom and Christian Lundquist. On the use of multiple measurement models for extended target tracking. *Proceedings of the 16th International Conference on Information Fusion*, page 1534, 2013.
- [18] Karl Granstrom, Antonio Natale, Paolo Vraça, Giovanni Ludeno, and Francesco Serafino. PHD extended target tracking using an incoherent x-band radar: Preliminary real-world experimental results. *17th International Conference on Information Fusion (FUSION)*, page 1, 2014.
- [19] Karl Granstrom and U. Orguner. New prediction for extended targets with random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 50(2):1577–1589, 2014.
- [20] Karl Granstrom, Stephan Reuter, Daniel Meissner, and Alexander Scheel. A multiple model PHD approach to tracking of cars under an assumed rectangular shape. *17th International Conference on Information Fusion (FUSION)*, page 1, 2014.
- [21] Jiawei Han, Micheline Kamber, Books24x7, Ebook Central, and ScienceDirect. *Data mining: concepts and techniques*. Elsevier, Burlington, MA, 3rd edition, 2011.
- [22] Tobias Hirscher, Alexander Scheel, Stephan Reuter, and Klaus Dietmayer. Multiple extended object tracking using gaussian processes. *2016 19th International Conference on Information Fusion (FUSION)*, page 868, 2016.
- [23] Hauke Kaulbersch, Marcus Baum, and Peter Willett. An EM approach for contour tracking based on point clouds. IEEE, 2016.
- [24] Hauke Kaulbersch, Marcus Baum, and Peter Willett. EM approach for tracking star-convex extended objects. pages 1–7. International Society of Information Fusion (ISIF), 2017.

- 
- [25] J. W. Koch. Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1042–1059, 2008.
- [26] M. L. Krieg and D. A. Gray. Multi-sensor, probabilistic multi-hypothesis tracking. In *AUSTRALIAN DATA FUSION SYMPOSIUM*, page 6:9, 2001.
- [27] Jian Lan and X. Rong Li. Tracking of extended object or target group using random matrix: new model and approach. *IEEE Transactions on Aerospace and Electronic Systems*, 52(6):2973 – 2989, 2016.
- [28] Alan J. Laub. *Matrix analysis for scientists and engineers*. Society for Industrial and Applied Mathematics, Philadelphia, 2005.
- [29] M. Lundgren, L. Svensson, and L. Hammarstrand. Variational bayesian expectation maximization for radar map estimation. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, (6):1391, 2016.
- [30] R. P. S. Mahler. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, 2003.
- [31] Geoffrey J. McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*. Wiley, Hoboken, N. J, 2. edition, 2008.
- [32] J. S. Milton and Jesse C. Arnold. *Introduction to probability and statistics: principles and applications for engineering and the computing sciences*. McGraw-Hill, Boston, 4. edition, 2003.
- [33] U. Orguner. A variational measurement update for extended target tracking with random matrices. *IEEE Transactions on Signal Processing*, 60(7):3827–3834, 2012.
- [34] Francesco Papi. Multi-sensor d-GLMB filter for multi-target tracking using doppler only measurements. *2015 European Intelligence and Security Informatics Conference*, page 83, 2015.
- [35] Kaare Brandt Petersen and Michael Syskind Pedersen. *The Matrix Cookbook*. Technical University of Denmark, 2012.
- [36] Michael Roth, Gustaf Hendeby, and Fredrik Gustafsson. EKF/UKF maneuvering target tracking using coordinated turn models with polar/cartesian velocity. *17th International Conference on Information Fusion (FUSION)*, page 1, 2014.
- [37] Stuart Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Pearson Education Limited, Harlow, 3., global edition, 2016.
- [38] Simo Särkkä. *Bayesian filtering and smoothing*. Cambridge university press, Cambridge, 2013.
- [39] Firdaus E. Udwardia and Robert E. Kalaba. *Analytical dynamics: new approach*. Cambridge University Press, Cambridge, 1996.

- [40] G. Vivone, Karl Granström, P. Braca, and P. Willett. Multiple sensor measurement updates for the extended target tracking random matrix model. *IEEE Transactions on Aerospace and Electronic Systems*, (5):2544, 2017.
- [41] Gemine Vivone, Karl Granstrom, Paolo Braca, and Peter Willett. Multiple sensor bayesian extended target tracking fusion approaches using random matrices. *2016 19th International Conference on Information Fusion (FUSION)*, page 886, 2016.
- [42] M. Wieneke and W. Koch. Probabilistic tracking of multiple extended targets using random matrices. *PROCEEDINGS- SPIE THE INTERNATIONAL SOCIETY FOR OPTICAL ENGINEERING*, page 7698 12, 2010.
- [43] M. Wieneke and W. Koch. A PMHT approach for extended objects and object groups. *IEEE Transactions on Aerospace and Electronic Systems*, 48(3):2349–2370, 2012.
- [44] Shishan Yang, Marcus Baum, and Karl Granstrom. Metrics for performance evaluation of elliptic extended object tracking methods. pages 523–528. IEEE, September 2016.
- [45] Wang Yun, Hu Guo-ping, and Zhou Hao. Group targets tracking using multiple models GGIW-CPHD based on best-fitting gaussian approximation and strong tracking filter. *Journal of Sensors, Vol 2016 (2016)*, 2016.

# A

## Appendix 1

### A.1 Matrix Variate Distributions

#### A.1.1 Wishart Distribution

Let  $\mathbb{S}_{++}^d$  be the set of symmetric positive definite  $d \times d$  matrices. The random matrix  $X \in \mathbb{S}_{++}^d$  is Wishart distributed with degrees of freedom  $n > d - 1$  and  $d \times d$  scale matrix  $N \in \mathbb{S}_{++}^d$  if it has *probability density function (pdf)*

$$p(X) = \mathcal{W}_d(X; n, N) = \frac{|X|^{(n-d-1)/2}}{2^{nd/2} \Gamma_d(\frac{n}{2}) |N|^{n/2}} \text{etr} \left\{ -\frac{1}{2} N^{-1} X \right\}. \quad (\text{A.1})$$

Where  $|\cdot|$  denotes the matrix determinant and  $\text{etr}(\cdot)$  is an abbreviation for  $\exp(\text{tr}(\cdot))$ , where  $\text{tr}(\cdot)$  denotes the trace of a matrix. For  $a > \frac{d-1}{2}$ ,  $\Gamma_d(a)$  is the multivariate gamma function. It can be expressed in terms of the ordinary gamma function as

$$\Gamma_d(a) = \pi^{d(d-1)} \prod_{i=1}^d \Gamma(a - (i-1)/2) \quad (\text{A.2})$$

Let  $A_{ij}$  denote the  $i, j$ :th element of a matrix  $A$ . The expected value and covariance of the Wishart distributed matrix  $X$  are defined as

$$\mathbb{E}[X_{ij}] = nN_{ij} \quad (\text{A.3})$$

$$\mathbb{C}(X_{ij}, X_{kl}) = n(N_{ik}N_{jl} + N_{il}N_{jk}). \quad (\text{A.4})$$

#### A.1.2 Inverse-Wishart Distribution

The random matrix  $X \in \mathbb{S}_{++}^d$  is inverse Wishart distributed with degrees of freedom  $v > 2d$  and inverse scale matrix  $V \in \mathbb{S}_{++}^d$  if it has pdf

$$p(X) = \mathcal{IW}_d(X; v, V) = \frac{2^{-(v-d-1)/2} |V|^{(v-d-1)/2}}{\Gamma_d((v-d-1)/2) |X|^{v/2}} \text{etr} \left\{ -\frac{1}{2} X^{-1} V \right\}. \quad (\text{A.5})$$

The expected value and covariance of  $X$  are

$$\mathbb{E}[X_{ij}] = \frac{V_{ij}}{v - 2d - 2}, \quad v - 2d - 2 > 0 \quad (\text{A.6})$$

$$\mathbb{C}(X_{ij}, X_{kl}) = \frac{2(v - 2d - 2)^{-1} V_{ij} V_{kl} + V_{ik} V_{jl} + V_{il} V_{jk}}{(v - 2d - 1)(v - 2d - 2)(v - 2d - 4)}, \quad v - 2d - 4 > 0 \quad (\text{A.7})$$

The Wishart distributions main appearance in a Bayesian framework is that it is the conjugate prior of the precision matrix, i.e. Inverse of the covariance matrix, of a multivariate Gaussian random variable [4]. If a random matrix  $X$  is Wishart distributed then the inverse  $X^{-1}$  is Inverse Wishart distributed.

### A.1.3 Generalized Beta Distribution type 2

Let  $\mathbb{S}_+^d$  be the set of symmetric semi-positive definite  $d \times d$  matrices. The random matrix  $X \in \mathbb{S}_{++}^d$  is generalized matrix variate Beta type II distributed with matrix parameters  $\Psi \in \mathbb{S}_+^d$ ,  $\Omega > \Psi$ , and scalar parameters  $a$  and  $b$ , if it has pdf

$$p(X) = \mathcal{GB}_d^{II}(X; a, b, \Omega, \Psi) = \frac{|X - \Psi|^{(2a-d-1)/2} |\Omega + X|^{-(a+b)}}{\beta_d(a, b) |\Omega + \Psi|^{-b}}, \quad X > \Psi. \quad (\text{A.8})$$

Where, for  $a > (d-1)/2$  and  $b > (d-1)/2$ , the multivariate beta function is as

$$\beta_d(a, b) = \frac{\Gamma_d(a)\Gamma_d(b)}{\Gamma_d(a+b)}. \quad (\text{A.9})$$

Let  $\mathbf{0}_d$  be a  $d \times d$  all zero matrix. If  $\Psi = \mathbf{0}_d$ , the first and second moment of  $X$  are

$$\mathbb{E}[X_{ij}] = \frac{2a}{2b-d-1} \Omega_{ij} \quad (\text{A.10})$$

$$\begin{aligned} \mathbb{E}[X_{ij}X_{kl}] &= \frac{2a}{(2b-d)(2b-d-1)(2b-d-3)} \times \\ &\left( (2a(2b-d-2) + 2)\Omega_{ij}\Omega_{kl} + (2a+2b-d-1)(\Omega_{jl}\Omega_{ik} + \Omega_{il}\Omega_{kj}) \right), \\ &2b-d-3 > 0. \end{aligned} \quad (\text{A.11})$$

## A.2 Preliminary Results

**Lemma 1:** The product of two inverse Wishart pdfs is proportional to an inverse Wishart pdf,

$$\mathcal{IW}_d(X; a, A)\mathcal{IW}_d(X; b, B) \propto \mathcal{IW}_d(X; a+b, A+B) \quad (\text{A.12})$$

**Lemma 2:** The fraction of two inverse Wishart pdfs is proportional to an inverse Wishart pdf,

$$\frac{\mathcal{IW}_d(X; a, A)}{\mathcal{IW}_d(X; b, B)} \propto \mathcal{IW}_d(X; a-b, A-B) \quad (\text{A.13})$$

**Lemma 3:** For Wishart and Inverse Wishart pdfs, the following holds,

$$\mathcal{W}_d(Y; n, \frac{MYM^T}{n}) \propto \mathcal{IW}_d(X; n, nM^{-1}Y(M^{-1})^T) \quad (\text{A.14})$$

**Lemma 4:**

$$\int \mathcal{W}_d(Y; n, \frac{MXM^T}{n}) \mathcal{W}_d(Y; w, W) dY \propto \mathcal{GB}_d^{II} \left( X; \frac{w}{2}, \frac{n-d-1}{2}, nM^{-1}Y(M^{-1})^T, \Psi = \mathbf{0} \right) \quad (\text{A.15})$$

**Lemma 5:** For two random variables  $x$  and  $y$ , with joint density  $p(x,y)$ , the factorized density  $q(x)q(y)$  that minimizes the Kullback-Leibler divergence to  $p(x,y)$

$$\operatorname{argmin} KL(p(x,y) || q(x)q(y)) \quad (\text{A.16})$$

is given by the marginals,

$$\begin{aligned} q(x) &= \int p(x,y) dy \\ q(y) &= \int p(x,y) dx \end{aligned} \quad (\text{A.17})$$

### A.3 Derivation of Smoothing Equation for CM and FM

$$p(X_k | Z^K) = \mathcal{IW}_d(X_k; v_{k|k}, V_{k|k}) \int \frac{\mathcal{W}_d(X_{k+1}; \delta_k, \frac{X_k}{\delta_k}) \mathcal{IW}_d(X_{k+1}; v_{k+1|K}, V_{k+1|K})}{\mathcal{IW}_d(X_{k+1}; v_{k+1|k}, V_{k+1|k})} dX_{k+1}$$

Lemma 2  
 $\propto$

$$\propto \mathcal{IW}_d(X_k; v_{k|k}, V_{k|k}) \int \mathcal{W}_d(X_{k+1}; \delta_k, \frac{X_k}{\delta_k})$$

$$\mathcal{IW}_d(X_{k+1}; v_{k+1|K} - v_{k+1|k}, V_{k+1|K} - V_{k+1|k}) dX_{k+1}$$

Thm 3 in [13]  
 $\approx$

$$\approx \mathcal{IW}_d(X_k; v_{k|k}, V_{k|k}) \int \mathcal{W}_d(X_{k+1}; \delta_k, \frac{X_k}{\delta_k}) \mathcal{W}_d(X_{k+1}; w_{k+1}, W_{k+1}) dX_{k+1}$$

Lemma 4  
 $\underline{\underline{=}}$

$$= \mathcal{IW}_d(X_k; v_{k|k}, V_{k|k}) \mathcal{GB}_d^{II} \left( X_k; \frac{w_{k+1}}{2}, \frac{\delta_k - d - 1}{2}, \delta_k W_{k+1}, \Psi = \mathbf{0} \right)$$

Thm 1 in [19]  
 $\approx$

$$\approx \mathcal{IW}_d(X_k; v_{k|k}, V_{k|k}) \mathcal{IW}_d(X_k; u_k, U_k)$$

Lemma 1  
 $\propto$

$$\propto \mathcal{IW}_d(X_k; v_{k|k} + u_k, V_{k|k} + U_k)$$

(A.18)

## A.4 Derivation of Smoothing Equation for FMNL

$$\begin{aligned}
p(X_k|Z^K) &= p(X_k|Z^k) \int \int \frac{p(X_{k+1}|x_k, X_k)p(X_{k+1}|Z^K)}{p(X_{k+1}|Z^k)} p(x_k|Z^K) dX_{k+1} dx_k \\
&= p(X_k|Z^k) \int \int \frac{\mathcal{W}_d(X_{k+1}; n_{k+1|k}, \frac{M(x_k)X_k M^T(x_k)}{n_{k+1|k}}) \mathcal{I}\mathcal{W}_d(X_{k+1}; \nu_{k+1|K}, V_{k+1|K})}{\mathcal{I}\mathcal{W}_d(X_{k+1}; \nu_{k+1|k}, V_{k+1|k})} \\
&\quad dX_{k+1} p(x_k|Z^K) dx_k \\
&\stackrel{\text{Lemma 2}}{\propto} \\
&\propto p(X_k|Z^k) \int \int \mathcal{W}_d(X_{k+1}; n_{k+1|k}, \frac{M(x_k)X_k M^T(x_k)}{n_{k+1|k}}) \\
&\quad \mathcal{I}\mathcal{W}_d(X_{k+1}; \nu_{k+1|K} - \nu_{k+1|k}, V_{k+1|K} - V_{k+1|k}) dX_{k+1} p(x_k|Z^K) dx_k \\
&\stackrel{\text{Thm 3 in [13]}}{\approx} \\
&\approx p(X_k|Z^k) \int \int \mathcal{W}_d(X_{k+1}; n_{k+1|k}, \frac{M(x_k)X_k M^T(x_k)}{n_{k+1|k}}) \\
&\quad \mathcal{W}_d(X_{k+1}; w_{k+1|K}, W_{k+1|K}) dX_{k+1} p(x_k|Z^K) dx_k \\
&\stackrel{\text{Lemma 4}}{\propto} \\
&\propto p(X_k|Z^k) \int \mathcal{G}\mathcal{B}_d^{II}(X_k; \frac{w_{k+1|K}}{2}, \frac{n_{k+1|k} - d - 1}{2}, n_{k+1|k} M^{-1}(x_k) W_{k+1|K} M^{-T}(x_k), \Psi = \mathbf{0}) \\
&\quad p(x_k|Z^K) dx_k \\
&\stackrel{\text{Thm 1 in [19]}}{\approx} \\
&\approx p(X_k|Z^k) \int \mathcal{I}\mathcal{W}_d(X_k; u_{k|K}, g_{k|K} \underbrace{M^{-1}(x_k) W_{k+1|K} M^{-T}(x_k)}_{=\mathbb{V}_{x_k}}) \mathcal{N}(x_k; m_{k|K}, P_{k|K}) dx_k \\
&\stackrel{\text{Thm 2 in [19]}}{\approx} \\
&\approx p(X_k|Z^k) \int \mathcal{I}\mathcal{W}_d(X_k; u_{k|K}, g_{k|K} \mathbb{V}_{x_k}) \mathcal{W}_d(\mathbb{V}_{x_k}; s_{k|K}, S_{k|K}) d\mathbb{V}_{x_k} \\
&\stackrel{\text{Thm 3 in [19]}}{=} \\
&= \mathcal{I}\mathcal{W}_d(X_k; \nu_{k|K}, V_{k|k}) \mathcal{G}\mathcal{B}_d^{II}(X_k; \frac{s_{k|K}}{2}, \frac{u_{k|K} - d - 1}{2}, g_{k|K} S_{k|K}, \Psi = \mathbf{0}) \\
&\stackrel{\text{Thm 1 in [19]}}{\approx} \\
&\approx \mathcal{I}\mathcal{W}_d(X_k; \nu_{k|K}, V_{k|k}) \mathcal{I}\mathcal{W}_d(X_k; l_{k|K}, L_{k|K}) \\
&\stackrel{\text{Lemma 1}}{\propto} \\
&\approx \mathcal{I}\mathcal{W}_d(X_k; \nu_{k|K} + l_{k|K}, V_{k|k} + L_{k|K})
\end{aligned} \tag{A.19}$$

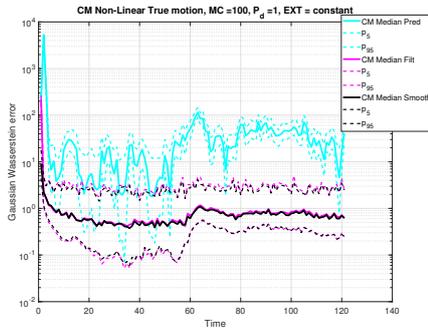


# B

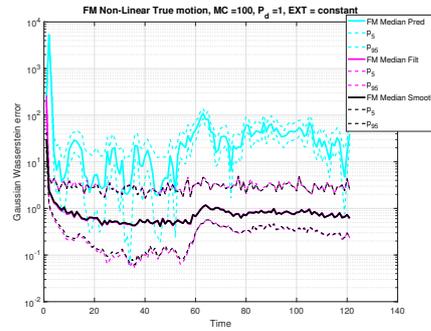
## Appendix 2

### B.1 Plots from the different scenarios in the single object case

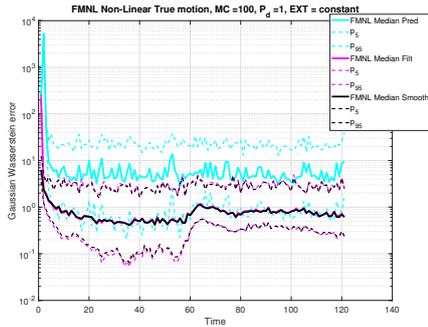
#### B.1.1 First scenario with nonlinear true motion



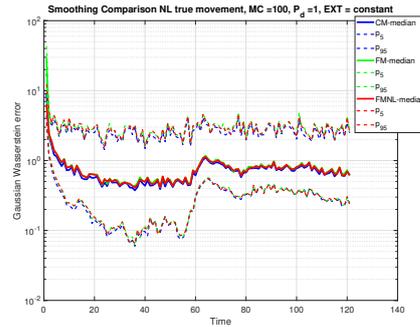
(a) CM



(b) FM



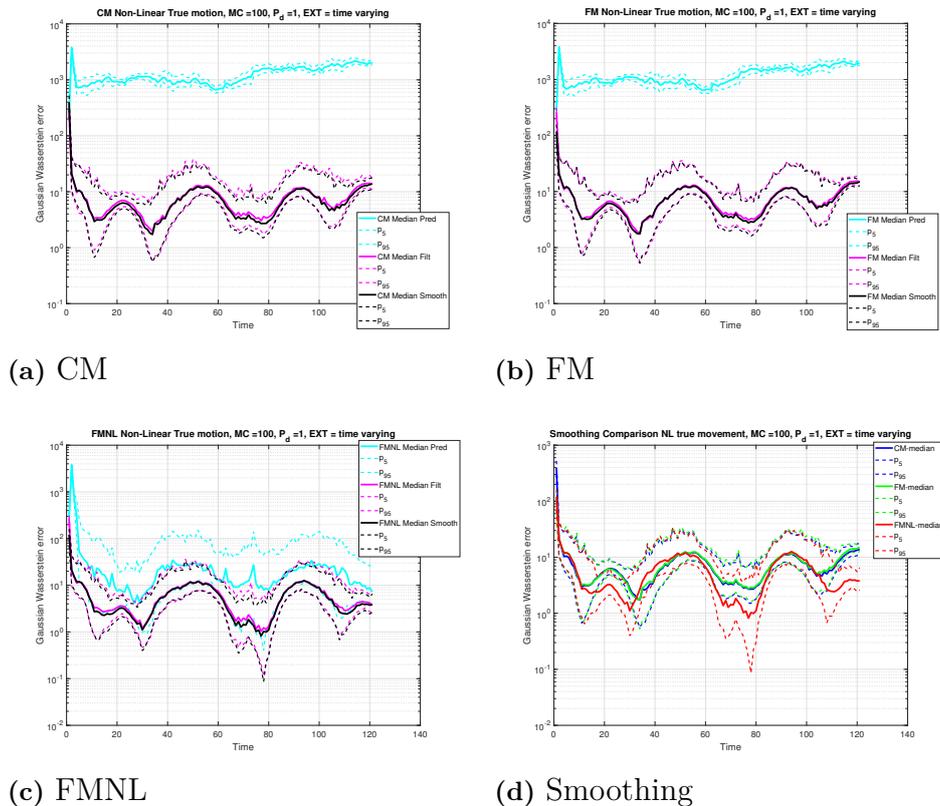
(c) FMNL



(d) Smoothing

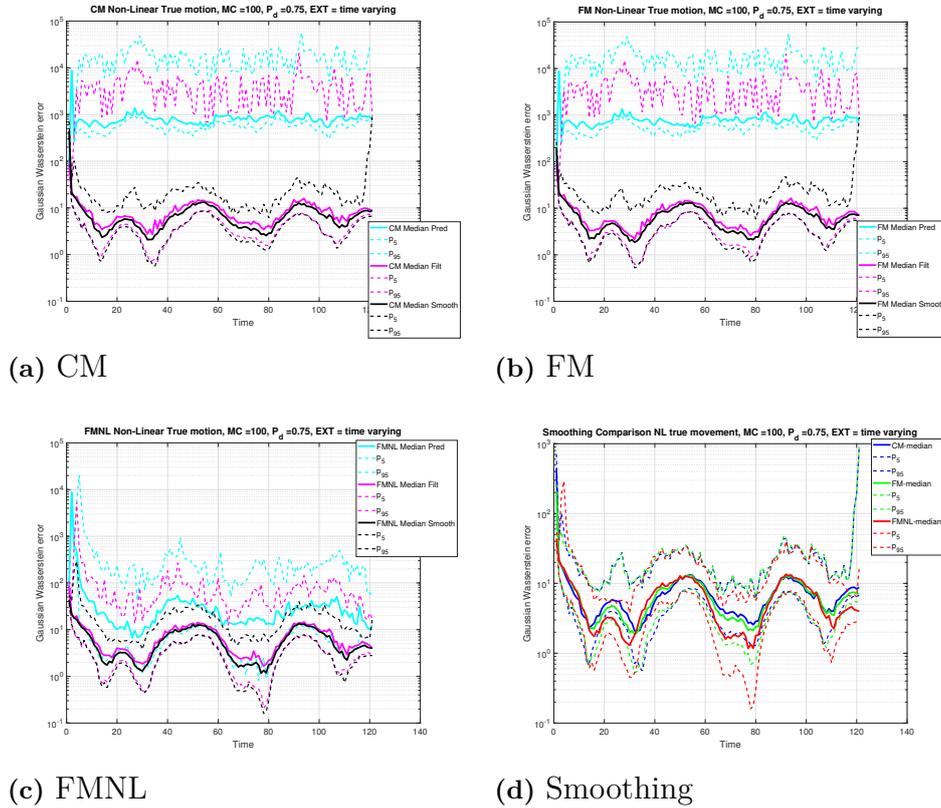
**Figure B.1:** (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is nonlinear and  $P_d = 1$ ,  $\lambda = 8$  and the extent is constant. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error.

## B.1.2 Second scenario with nonlinear true motion



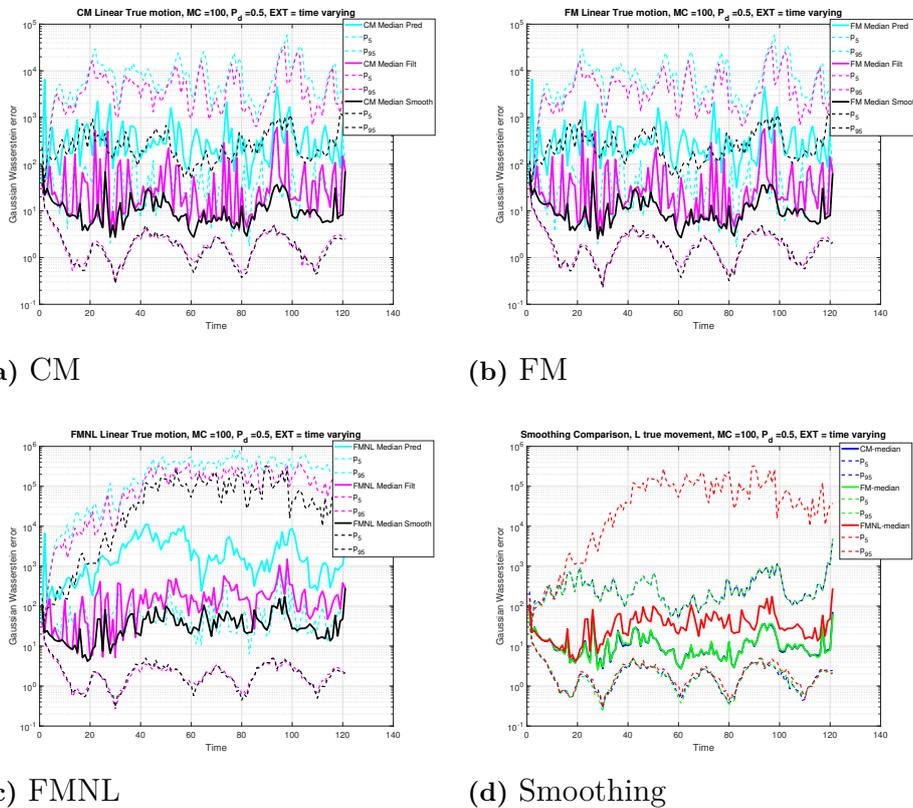
**Figure B.2:** (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is nonlinear and  $P_d = 1$ ,  $\lambda = 8$  and the time varying extent. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error.

### B.1.3 Third scenario with nonlinear motion



**Figure B.3:** (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is nonlinear and  $P_d = .75$ ,  $\lambda = 8$  and the time varying extent. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error.

### B.1.4 Fourth scenario with linear motion for $P_d = 0.5$



**Figure B.4:** (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is linear and  $P_d = .5$ ,  $\lambda = 8$  and the time varying extent. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error.

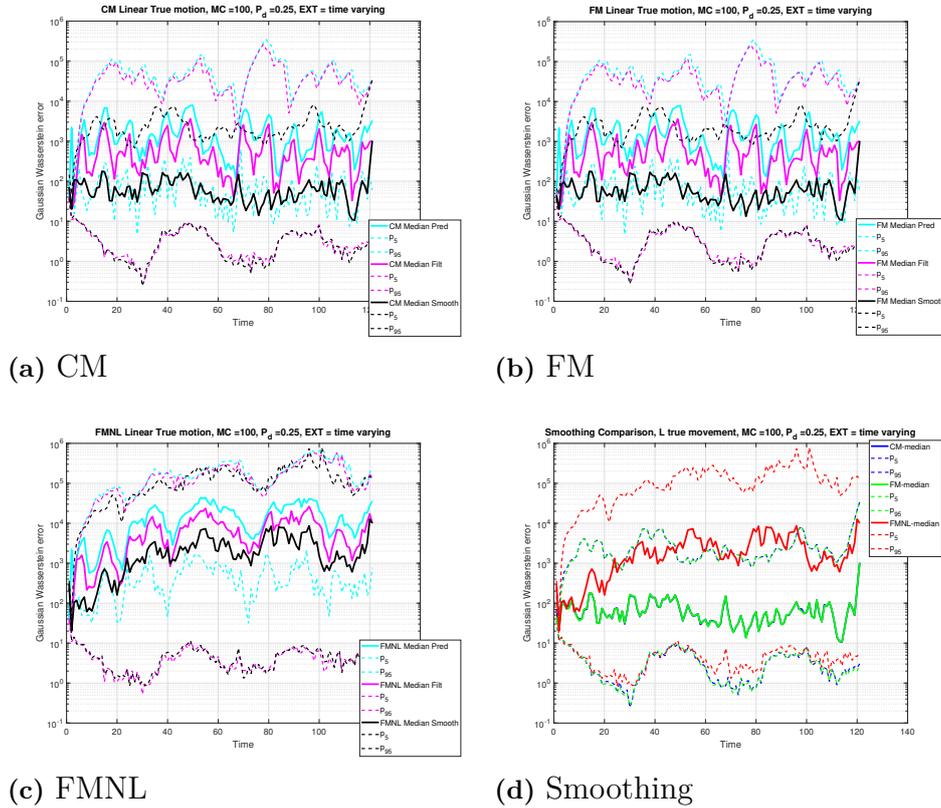
Linear $P_d = 0.5$ , ext = varying, $\lambda = 8$	Prediction	Filtering	Smoothing
CM	195264.9	88511.6	<b>7660.7</b>
FM	195939.9	88412.1	<b>7670.7</b>
FMNL	1192663.1	659188.0	<b>40280.8</b>

**Table B.1:** The numerical calculation based on (4.3) for the GWD estimation error in the case of nonlinear true motion,  $P_d = 0.5$ , the extent is changing over time (denoted ext = varying) and measurement rate  $\lambda = 8$ .

Linear $P_d = 0.25$ , ext = varying, $\lambda = 8$	Prediction	Filtering	Smoothing
CM	1017708.2	705560.2	53590.5
FM	1023482.6	705595.8	53709.8
FMNL	4413152.2	3143278.1	2651117.6

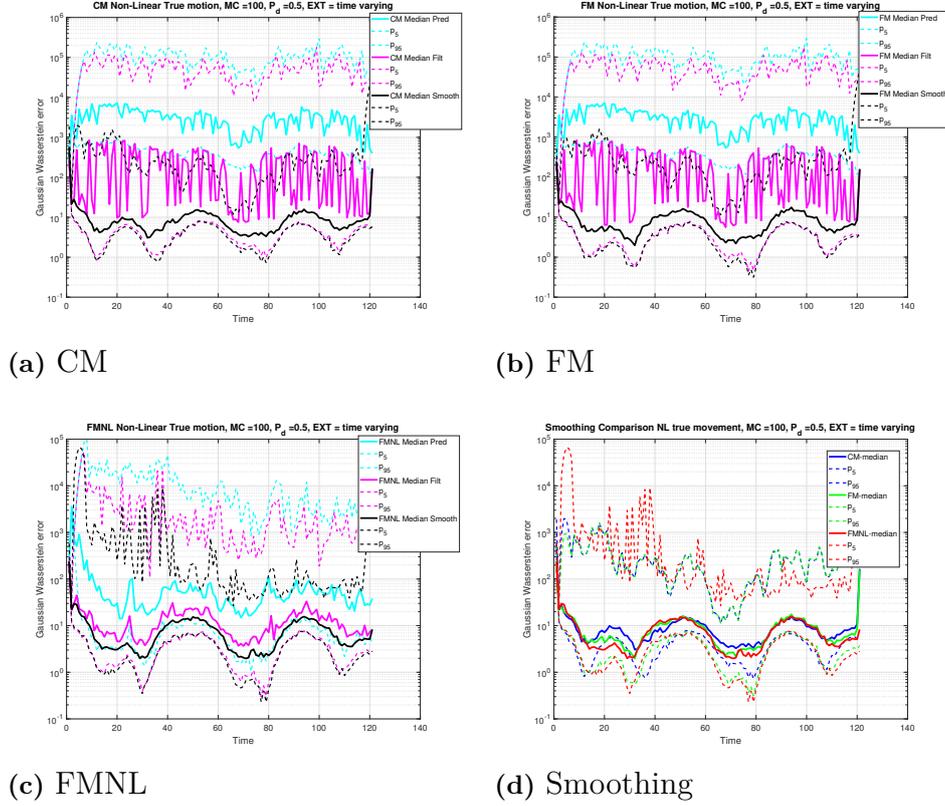
**Table B.2:** The numerical calculation based on (4.3) for the GWD estimation error in the case of nonlinear true motion,  $P_d = 0.25$ , the extent is changing over time (denoted ext = varying) and measurement rate  $\lambda = 8$ .

### B.1.5 Four scenario with linear motion and $P_d = 0.25$



**Figure B.5:** (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is linear and  $P_d = .25$ ,  $\lambda = 8$  and the time varying extent. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error.

### B.1.6 Fourth scenario with nonlinear motion and $P_d = 0.5$



**Figure B.6:** (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is nonlinear and  $P_d = .5$ ,  $\lambda = 8$  and the time varying extent. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error.

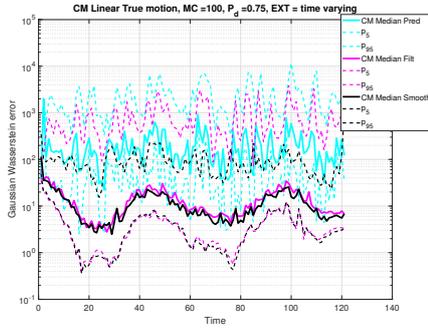
Nonlinear $P_d = 0.5$ , ext = varying, $\lambda = 8$	Prediction	Filtering	Smoothing
CM	2025576.6	887291.1	<b>6210.8</b>
FM	2017601.0	883095.0	<b>5277.6</b>
FMNL	52467.7	21713.2	<b>1751.5</b>

**Table B.3:** The numerical calculation based on (4.3) for the case of nonlinear true motion,  $P_d = 0.5$ , the extent is changing over time and  $\lambda = 8$ .

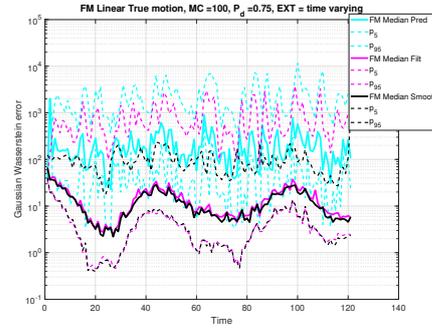
Nonlinear $P_d = 0.25$ , ext = varying, $\lambda = 8$	Prediction	Filtering	Smoothing
CM	925459.2	615497.4	227530.6
FM	10098216.1	7009829.8	552678.2
FMNL	10141541.0	6920408.8	555675.4

**Table B.4:** The numerical calculation based on (4.3) for the case of nonlinear true motion,  $P_d = 0.25$ , the extent is changing over time and  $\lambda = 8$ .

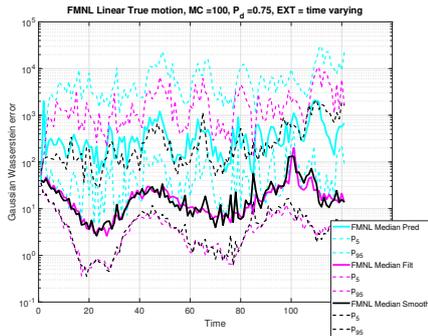
### B.1.7 Fifth scenario with linear motion and $\lambda = 3$



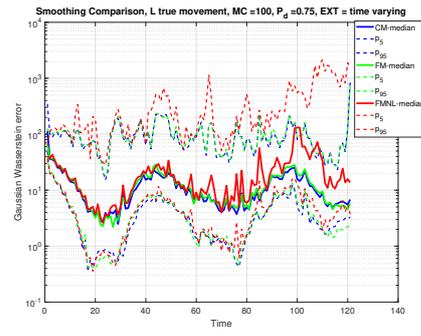
(a) CM



(b) FM



(c) FMNL



(d) Smoothing

**Figure B.7:** (a) is CM, (b) FM and (c) FMNL. (a), (b) and (c) shows the internal error comparison for each model between its prediction, in cyan coloured curves, filtering, the magenta coloured curves, and the black smoothing curves. (d) compares only the three smoothings where blue is CM, green FM and red FMNL. Here the true motion is linear and  $P_d = .75$ ,  $\lambda = 3$  and the time varying extent. The solid curves of all figures are the median GWD error for each time instance based on 100 Monte-Carlo simulations. The dashed lines are the corresponding 5- and 95-percentiles of the GWD error.

Linear $P_d = 0.75$ , ext = varying, $\lambda = 3$	Prediction	Filtering	Smoothing
CM	67936.8	17903.5	<b>2978.4</b>
FM	68172.6	17986.8	<b>3073.4</b>
FMNL	158036.9	36980.9	<b>7782.8</b>

**Table B.5:** The numerical calculation based on (4.3) for the case of linear true motion,  $P_d = 0.75$ , the extent is changing over time and  $\lambda = 3$ .

Linear $P_d = 0.75$ , ext = varying, $\lambda = 1$	Prediction	Filtering	Smoothing
CM	262393.9	116079.7	<b>8176.5</b>
FM	261823.7	115512.9	<b>8574.2</b>
FMNL	1713518.5	710902.7	<b>326815.2</b>

**Table B.6:** The numerical calculation based on (4.3) for the case of linear true motion,  $P_d = 0.75$ , the extent is changing over time and  $\lambda = 1$ .

### B.1.8 Fifth scenario with nonlinear motion and $\lambda = 3$

Nonlinear $P_d = 0.75$ , ext = varying, $\lambda = 3$	Prediction	Filtering	Smoothing
CM	2030412.3	522449.3	<b>38933.0</b>
FM	1959646.3	504364.9	<b>29826.6</b>
FMNL	30618.6	7419.7	<b>2630.9</b>

**Table B.7:** The numerical calculation based on (4.3) for the case of nonlinear true motion,  $P_d = 0.75$ , the extent is changing over time and  $\lambda = 3$ .

Nonlinear $P_d = 0.75$ , ext = varying, $\lambda = 1$	Prediction	Filtering	Smoothing
CM	4065746.7	1926297.4	<b>45483.5</b>
FM	3944307.6	1861930.5	<b>39466.2</b>
FMNL	149125.0	66254.6	<b>6912.8</b>

**Table B.8:** The numerical calculation based on (4.3) for the case of nonlinear true motion,  $P_d = 0.75$ , the extent is changing over time and  $\lambda = 1$ .



# C

## Appendix 3

### C.1 Results from the different scenarios in the multiple object case

### C.2 First scenario - Only Distance

Object\case	$\Delta > 10\sigma$	$\Delta = 4\sigma$	$\Delta = 3\sigma$	$\Delta = 2\sigma$	$\Delta = \sigma$	$\Delta = \frac{2}{3}\sigma$	$\Delta = \frac{1}{3}\sigma$
CM	279.4	270.2	282.8	281.2	745.7	762.6	760.7
FM	255.2	242.0	255.9	251.5	729.5	735.4	736.3
FMNL	268.7	261.5	275.4	275.8	754.1	748.0	763.5

**Table C.1:** The numerical errors for the models for different distances  $\Delta$ , with  $P_d = 1$ , extent is constant and  $\lambda = 8$ .

Object\case	$\Delta > 10\sigma$	$\Delta = 4\sigma$	$\Delta = 3\sigma$	$\Delta = 2\sigma$	$\Delta = \sigma$
CM	1287.9	1288.9	1328.5	3750.7	4189.6
FM	1004.9	988.8	1023.0	3465.9	3843.1
FMNL	1259.2	1259.2	1234.3	3706.7	27569.4

**Table C.2:** The numerical errors for the models for different distances  $\Delta$ , with  $P_d = 1$ , extent is time varying and  $\lambda = 8$ .

## C.3 Second scenario - Probability of Detection

### C.3.1 Constant Extent

Case \ $P_d$	$P_d = 1$	$P_d = 0.75$	$P_d = 0.5$	$P_d = 0.25$
CM: $\Delta > 10\sigma$	279.4	320.1	425.0	776.2
CM: $\Delta = 4\sigma$	270.2	327.5	1634.9	66803.0
CM: $\Delta = 3\sigma$	282.8	324.5	21775.1	78661.4
CM: $\Delta = 2\sigma$	281.2	746.0	13455.8	150418.4
CM: $\Delta = \sigma$	745.7	850.7	8910.1	149782.2
CM: $\Delta = \frac{2}{3}\sigma$	762.6	853.6	26685.2	126421.0
CM: $\Delta = \frac{1}{3}\sigma$	760.7	863.9	58675.3	118124.8

**Table C.3:** The numerical errors for CM model for different probability of detection and distances  $\Delta$ , with  $P_d$ , extent is constant and  $\lambda = 8$ .

Case \ $P_d$	$P_d = 1$	$P_d = 0.75$	$P_d = 0.5$	$P_d = 0.25$
FM: $\Delta > 10\sigma$	255.2	297.9	402.8	739.9
FM: $\Delta = 4\sigma$	242.0	305.9	1134.2	78452.0
FM: $\Delta = 3\sigma$	255.9	302.8	13159.9	80600.3
FM: $\Delta = 2\sigma$	251.5	731.4	12619.0	75261.7
FM: $\Delta = \sigma$	729.5	809.8	10584.3	158218.5
FM: $\Delta = \frac{2}{3}\sigma$	735.4	809.8	24598.1	94849.0
FM: $\Delta = \frac{1}{3}\sigma$	736.3	863.2	17701.0	51794.7

**Table C.4:** The numerical errors for FM model for different probability of detection and distances  $\Delta$ , with  $P_d$ , extent is constant and  $\lambda = 8$ .

Case \ $P_d$	$P_d = 1$	$P_d = 0.75$	$P_d = 0.5$	$P_d = 0.25$
FMNL: $\Delta > 10\sigma$	268.7	398.3	1131.8	10364.0
FMNL: $\Delta = 4\sigma$	261.5	364.4	89151.4	955118.7
FMNL: $\Delta = 3\sigma$	275.4	420.2	180004.8	954103.0
FMNL: $\Delta = 2\sigma$	275.8	818.7	404232.2	1139133.1
FMNL: $\Delta = \sigma$	754.1	58146.6	527425.3	1073118.3
FMNL: $\Delta = \frac{2}{3}\sigma$	748.0	45278.8	471982.0	1068376.0
FMNL: $\Delta = \frac{1}{3}\sigma$	763.5	17293.4	196699.0	1227619.9

**Table C.5:** The numerical errors for FMNL model for different probability of detection and distances  $\Delta$ , with  $P_d$ , extent is constant and  $\lambda = 8$ .

### C.3.2 Time varying Extent

Case \ $P_d$	$P_d = 1$	$P_d = 0.75$	$P_d = 0.5$	$P_d = 0.25$
CM: $\Delta > 10\sigma$	1287.9	1453.0	1707.4	2649.0
CM: $\Delta = 4\sigma$	1288.9	4377.5	131461.3	2966307.9
CM: $\Delta = 3\sigma$	1328.5	4614.9	204141.4	3617944.1
CM: $\Delta = 2\sigma$	3750.7	4810.5	301042.6	1146592.7
CM: $\Delta = \sigma$	4189.6	5165.0	94687.7	4362873.4

**Table C.6:** The numerical errors for CM model for different probability of detection and distances  $\Delta$ , with  $P_d$ , extent is time varying and  $\lambda = 8$ .

Case \ $P_d$	$P_d = 1$	$P_d = 0.75$	$P_d = 0.5$	$P_d = 0.25$
FM: $\Delta > 10\sigma$	1004.9	1184.6	1451.5	2339.1
FM: $\Delta = 4\sigma$	988.8	4078.2	25133.0	3807862.0
FM: $\Delta = 3\sigma$	1023.0	4162.8	42655.4	3827197.4
FM: $\Delta = 2\sigma$	3465.9	4218.4	167987.8	1413322.1
FM: $\Delta = \sigma$	3843.1	4228.8	53690.2	2961858.0

**Table C.7:** The numerical errors for FM model for different probability of detection and distances  $\Delta$ , with  $P_d$ , extent is time varying and  $\lambda = 8$ .

Case \ $P_d$	$P_d = 1$	$P_d = 0.75$	$P_d = 0.5$	$P_d = 0.25$
FMNL: $\Delta > 10\sigma$	1259.2	1735.0	4467.0	35024.6
FMNL: $\Delta = 4\sigma$	1259.2	85734.1	1597880.6	3862260.2
FMNL: $\Delta = 3\sigma$	1234.3	709450.2	1682435.0	3668876.3
FMNL: $\Delta = 2\sigma$	3706.7	750672.2	1528366.1	3645698.2
FMNL: $\Delta = \sigma$	27569.4	696204.7	1385337.0	3816754.8

**Table C.8:** The numerical errors for FMNL model for different probability of detection and distances  $\Delta$ , with  $P_d$ , extent is time varying and  $\lambda = 8$ .

## C.4 Third scenario - Measurement Rate

### C.4.1 Constant Extent

$\Delta \backslash \lambda$	$\lambda = 8$	$\lambda = 3$	$\lambda = 1$
CM $\Delta > 10\sigma$	279.4	577.9	1113.6
CM $\Delta = 4\sigma$	270.2	567.0	1451.5
CM $\Delta = 3\sigma$	282.8	580.8	53813.6
CM $\Delta = 2\sigma$	281.2	594.2	78589.0
CM $\Delta = \sigma$	745.7	118791.2	59828.3
CM $\Delta = \frac{2}{3}\sigma$	762.6	31317.0	71879.5
CM $\Delta = \frac{1}{3}\sigma$	760.7	53267.3	79815.2

**Table C.9:** The numerical result for CM when the extent is constant and  $P_d = 1$  for different values of the object distance  $\Delta$  and the measurement rate  $\lambda$ .

$\Delta \backslash \lambda$	$\lambda = 8$	$\lambda = 3$	$\lambda = 1$
FM $\Delta > 10\sigma$	255.2	488.0	935.4
FM $\Delta = 4\sigma$	242.0	475.3	1587.2
FM $\Delta = 3\sigma$	255.9	484.9	69308.4
FM $\Delta = 2\sigma$	251.5	498.6	99775.5
FM $\Delta = \sigma$	729.5	77200.4	87411.0
FM $\Delta = \frac{2}{3}\sigma$	735.4	65065.7	103660.8
FM $\Delta = \frac{1}{3}\sigma$	736.3	91236.1	98366.2

**Table C.10:** The numerical result for FM when the extent is constant and  $P_d = 1$  for different values of the object distance  $\Delta$  and the measurement rate  $\lambda$ .

$\Delta \backslash \lambda$	$\lambda = 8$	$\lambda = 3$	$\lambda = 1$
FMNL $\Delta > 10\sigma$	268.7	571.4	1474.1
FMNL $\Delta = 4\sigma$	261.5	561.6	38144.3
FMNL $\Delta = 3\sigma$	275.4	576.8	286066.5
FMNL $\Delta = 2\sigma$	275.8	627.2	280652.3
FMNL $\Delta = \sigma$	754.1	268400.0	334280.5
FMNL $\Delta = \frac{2}{3}\sigma$	748.0	122177.2	221571.8
FMNL $\Delta = \frac{1}{3}\sigma$	763.5	502512.9	298335.8

**Table C.11:** The numerical result for FMNL when the extent is constant and  $P_d = 1$  for different values of the object distance  $\Delta$  and the measurement rate  $\lambda$ .

### C.4.2 Time varying Extent

$\Delta \backslash \lambda$	$\lambda = 8$	$\lambda = 3$	$\lambda = 1$
CM $\Delta > 10\sigma$	1287.9	2454.3	4590.6
CM $\Delta = 4\sigma$	1288.9	2607.5	18252654.7
CM $\Delta = 3\sigma$	1328.5	1444222.1	12230553.1
CM $\Delta = 2\sigma$	3750.7	1787940.1	13418794.3
CM $\Delta = \sigma$	4189.6	2360928.4	21476642.1

**Table C.12:** The numerical result for CM when the extent is varying with time and  $P_d = 1$  for different values of the object distance  $\Delta$  and the measurement rate  $\lambda$ .

$\Delta \backslash \lambda$	$\lambda = 8$	$\lambda = 3$	$\lambda = 1$
FM $\Delta > 10\sigma$	1004.9	1679.0	3475.0
FM $\Delta = 4\sigma$	988.8	1770.6	5736411.0
FM $\Delta = 3\sigma$	1023.0	439391.1	6006137.0
FM $\Delta = 2\sigma$	3465.9	562858.0	8901773.4
FM $\Delta = \sigma$	3843.1	1800690.4	5990356.9

**Table C.13:** The numerical result for FM when the extent is varying with time and  $P_d = 1$  for different values of the object distance  $\Delta$  and the measurement rate  $\lambda$ .

$\Delta \backslash \lambda$	$\lambda = 8$	$\lambda = 3$	$\lambda = 1$
FMNL $\Delta > 10\sigma$	1259.2	2452.2	6848.1
FMNL $\Delta = 4\sigma$	1259.2	2695.1	5394262.1
FMNL $\Delta = 3\sigma$	1234.3	1907358.3	3454064.5
FMNL $\Delta = 2\sigma$	3706.7	1708219.3	4096796.2
FMNL $\Delta = \sigma$	27569.4	1690706.0	3310880.4

**Table C.14:** The numerical result for FMNL when the extent is varying with time and  $P_d = 1$  for different values of the object distance  $\Delta$  and the measurement rate  $\lambda$ .