



DEGREE PROJECT IN MATHEMATICS,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2016

Evaluation of Decentralized Information Matrix Fusion for Advanced Driver-Assistance Systems in Heavy-Duty Vehicles

VIKTOR ERIKSSON



KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ENGINEERING SCIENCES

Scania CV AB
Driver Assistance Controls
Vehicle Control Systems



Evaluation of Decentralized Information Matrix Fusion for Advanced Driver-Assistance Systems in Heavy-Duty Vehicles

Utvärdering av Decentraliserad Informationsmatris-Fusion för Avancerande Försärsystem i Lastbilar

V I K T O R E R I K S S O N

Master's Thesis in Optimization and Systems Theory (30 ECTS credits)
Master Programme in Applied and Computational Mathematics (120 credits)
Royal Institute of Technology year 2016
Supervisor at Scania: Dr. Christian Larsson
Supervisor at KTH: Xiaoming Hu
Examiner: Xiaoming Hu

TRITA-MAT-E 2016:53
ISRN-KTH/MAT/E--16/53--SE

Royal Institute of Technology
SCI School of Engineering Sciences
KTH SCI
SE-100 44 Stockholm, Sweden

URL: www.kth.se/sci

Abstract

Advanced driver-assistance systems (ADAS) is one of the fastest growing areas of automotive electronics and are becoming increasingly important for heavy-duty vehicles. ADAS aims to give the driver the option of handing over all driving decisions and driving tasks to the vehicle, allowing the vehicle to make fully automatic maneuvers. In order to perform such maneuvers target tracking of surrounding traffic is important in order to know where other objects are. Target tracking is the art of fusing data from different sensors into one final value with the goal to create an as accurate as possible estimate of the reality.

Two decentralized information matrix fusion algorithms and a weighted least-squares fusion algorithm for target tracking have been evaluated on two simulated overtaking maneuvers performed by a single target. The first algorithm is the optimal decentralized algorithm (ODA), which is an optimal IMF filter, the second algorithm is the decentralized-minimum-information algorithm (DMIA), which approximates the error covariance of received estimates, and the third algorithm is the naïve algorithm (NA), which uses weighted-least-squares estimation for data fusion. In addition, DMIA and NA are evaluated using real sensor data from a test vehicle.

The results are generated from 100 Monte Carlo runs of the simulations. The error of position and velocity as well as the their corresponding root-mean-squared-error (RMSE) are smallest for ODA followed by NA and DMIA. ODA gives consistent estimators for the first simulated overtaking but not the second. DMIA and NA are not statistically significant on a 95 % level. The robustness against sensor failures shows that ODA is robust and yields similar results to the simulations without sensor failures. DMIA and NA are sensitive to sensor failures and yield unstable results. ODA is clearly the best option to use for sensor fusion in target tracking.

Sammanfattning

Avancerade förarsystem (ADAS) är en av de snabbast växande områdena inom fordonselektronik och blir mer och mer viktigt även för lastbilar. ADAS riktar sig till att ge föraren möjligheten att låta fordonet ta beslut om köningen och utföra autonoma manövrar. För att kunna utföra sådana manövrar krävs objektföljning av omkringvarande fordon. Sensorfusion inom objektföljning är tekniken att kombinera data från olika sensorer till ett värde med målet att skapa en så precis skattning av verkligheten som möjligt.

Två decentraliseraade informationsmatris-fusions algoritmer och en viktad minsta-kvadrat fusions algoritm för objektskattning har blivit utvärderade utifrån två simulerade omkörningar utförda av ett enskilt objekt. Den första algoritmen är *optimal decentralized algorithm* (ODA), som är ett optimalt informationsmatris-fusions filter, den andra algoritmen är *decentralized-minimum-information algorithm* (DMIA), som approximerar kovariansmatrisen av residualerna från mottagna skattningar, samt den tredje algoritmen är *naïve algorithm* (NA), som kombinerar data från sensorerna med hjälp av viktad minsta-kvadrat fusion. Utöver detta är DMIA och NA även utvärderade på riktig sensordata från ett testfordon.

Resultaten är genererade från 100 Monte Carlo körningar av simuleringarna. Residualerna för position och hastighet samt minsta-kvadrat felet är minst för ODA följt av NA och DMIA. ODA ger konsistenta skattningar under den första simulerade omkörningen men inte under den andra omkörningen. DMIA och NA är inte konsistenta på en 95 % signifikansnivå under någon av omkörningarna. ODA är robust och ger liknande resultat i simuleringarna med och utan sensorfel. DMIA och NA är känsliga mot sensorfel och ger instabila resultat. ODA är det klart bästa alternativet för sensorfusion inom objektföljning.

Acknowledgements

The work described in this master's thesis has been conducted at the Vehicle Control Systems Department, REV, at Scania CV AB in Södertälje, Sweden. It was supervised by the Department of Mathematics, at the Royal Institute of Technology (KTH), in Stockholm. I would first and foremost like to thank my supervisor Christian Larsson at Scania, for all his help and guidance during this Master's Thesis. His input have been invaluable and inspiring. Along with Christian Larsson, other people at Scania namely Per Sahlholm, Assad Alam, Jonny Andersson and Hjalmar Lundin have been most helpful and supportive. I would therefore like to extend my gratitude towards them as well. Finally I would like to thank my supervisor Xiaoming Hu at KTH for his guidance, input, time and support.

Viktor Eriksson, Stockholm, August 2016

Glossary

ADAS Advanced Driver-Assistance Systems. 1

DMIA Decentralized-Minimum-Information Algorithm. 5

IMF Information Matrix Fusion. 4

MFF Master Fusion Filter. 5

NA Naïve Algorithm. 5

NEES Normalized Estimation Error Squared. 25

ODA Optimal Decentralized Algorithm. 5

RMSE Root Mean Squared Error. 6

Contents

Glossary	x
1 Introduction	1
2 Background	3
2.1 Sensor fusion	3
2.1.1 Premise	3
2.1.2 Delimitations	4
2.1.3 Implementation	5
2.2 Related work	5
2.3 Thesis outline & objective	6
3 Sensor fusion	7
3.1 Target tracking	9
3.1.1 Sensor properties	11
3.2 Decentralized fusion architectures	14
3.2.1 Information matrix fusion	15
3.2.2 Decentralized-minimum-information algorithm	17
3.2.3 Naïve algorithm	21
4 Analysis & simulation	25
4.1 Simulation setup	25
4.1.1 Singular covariance matrices	27
4.1.2 Real scenario	28
4.2 Robustness	29
4.3 Consistency of estimators	30
5 Results	33
5.1 Simulated scenarios	33
5.1.1 Scenario 1	33
5.1.2 Scenario 2	36
5.2 Robustness to sensor failures	39
5.2.1 Scenario 1	40
5.2.2 Scenario 2	43
5.3 Test vehicle evaluation	46

6 Discussion	49
7 Conclusion	53
8 Future work & extensions	55
Bibliography	55
A Appendix - The Kalman filter	I
B Appendix - The information filter	V

Introduction

Scania CV AB is one of the leading manufacturers of heavy-duty trucks and busses as well as engines for industry and marine applications. Advanced driver-assistance systems (ADAS) is one of the fastest growing areas of automotive electronics and are becoming increasingly important for heavy-duty vehicles. Two successfully implemented ADAS systems developed by Scania CV AB are Adaptive Cruise Control (AiCC) and Advanced Emergency Braking (AEB). Many other systems are also being developed. Common to all these systems is that they rely on measurements from, e.g. radars, cameras, GPS, or vehicle-to-vehicle communication to get information about the surrounding traffic.

The aim of ADAS is to give the driver the option of handing over all driving decisions and driving tasks to the vehicle, allowing the vehicle to make fully automatic maneuvers. Although autonomous, the systems needs to be monitored so that the driver can take control during critical situations that the systems may not be able to handle.

In the future, autonomous vehicles will be used more extensively and will have to satisfy world-wide standards. It is thus important that ADAS systems are robust and secure in order to ensure the safety of the driver and the surrounding traffic. In order for future ADAS systems to reach expectations, smart, secure, efficient and innovative engineering solutions and implementations must be made. One step towards the realization of the next generation of ADAS systems is the field of sensor fusion since single-sensor perception systems will not provide the necessary reliability and robustness.

Background

2.1 Sensor fusion

When multiple sensors measure the same physical entity of a target, the obtained information can be combined to obtain a better estimate, which is known as sensor fusion. The sensor fusion problem can be divided into two subproblems: data association where measurements are assigned to a specific target, and state estimation and target tracking where the target properties are estimated and tracked in time and space. The state estimation problem is often performed in a decentralized manner, where initial estimates are made directly in the sensors. These estimates are then sent to and fused by a central processor node. Alternatively, the raw data can be sent directly to a central node. At this point, it is difficult to say which alternative is most suitable for coming sensor fusion algorithms in heavy-duty vehicles. ADAS systems are today more developed in automobiles as a results of higher production volumes. ADAS systems for heavy-duty vehicles needs to be more robust, from both a hardware and software point of view, since the life expectancy is longer for a heavy-duty vehicle compared to an automobile. In addition, the process for cleaning heavy-duty vehicles is more strenuous which puts constraints on the sensors being used.

A *centralized* architecture has access to, and processes all the measurement data from the sensors using a single sensor fusion node which is directly connected to all sensors. In a *decentralized* architecture, each sensor processes raw measurement data prior to communication with the central sensor fusion node.

2.1.1 Premise

The sensors used are assumed to be a camera and a radar oriented in a *forward-looking* direction, i.e. the longitudinal direction of the vehicle. The sensors detect targets in a span of $\pm 45^\circ$ from the longitudinal axis.

The radar is most accurate when a target is at a certain distance from the host vehicle, the radar is thus working at a *short range* or a *long range* depending on the distance between the host vehicle and the target. The difference between short

range and long range is the measurement covariance matrix which is discussed later on in the thesis.

It is assumed that all connections between sensors and the central processor are wired and thus no communication errors due to wireless signaling occur.

The vehicular camera is good at detecting the width of a target, i.e. has a low variance in lateral distance, but is less accurate in detecting the distance to the target. The vehicular radar is good at detecting how far away a target is, i.e. has a low variance in longitudinal distance, but is less accurate in detecting the width of the target.

The simulations are restricted to observations of a single target vehicle in order to better analyse the performance of the fusion algorithms without additional errors occurring due to false associations.

2.1.2 Delimitations

Sensors used in the automotive industry, e.g. on heavy-duty vehicles, are produced by automotive suppliers and usually in such a way that the end user, in this case the heavy-duty vehicle, only has access to processed and tracked object lists [1]. The information sent from a sensor to the central processor is thus limited. For ADAS systems, it is most likely that the data fusion will be the fusion of already locally tracked sensor objects, i.e. the sensors have a local filter tracking an object and the central processor fuses the locally tracked estimates.

As heavy-duty vehicles spend most of their transport time on highway-driving the simulations and results are focused on this environment. If the vehicular radar and camera are accurate, global estimation of longitudinal distance, lateral distance and their relative velocities will be proper. However, there are practical limitations that might influence the validity of the results; delays in the processing subsystems, unknown data processing in the sensors, limited knowledge about sensor measurement noise among others make way for significant possibility of errors. Delays in the subsystems are ignored and the data processing in the sensors are assumed to be local Kalman filters where the measurement noise is assumed to be known. Furthermore, a simulation carried out on real logged data shows the proper extent of these assumptions.

There are several approaches and algorithms available for multiple-sensor data fusion but the algorithms in this thesis are based on information matrix fusion (IMF) which uses the knowledge of two types of filters, the Kalman filter and the information filter, sometimes referred to as inverse Kalman filter or information form Kalman filter. Investigating more architectures and sensor fusion algorithms would require more time than what is allowed for this thesis.

2.1.3 Implementation

The first part of the implementation is to compare and evaluate a *decentralized-minimum-information algorithm* (DMIA) against an *optimal decentralized algorithm* (ODA) and a simpler algorithm, referred to as the *naïve algorithm* (NA), on two simulated overtaking maneuvers. The target trajectory is constructed from a constant acceleration motion model where the process noise from the model and the measurement noise from the sensors are known. NA fuses sensor data using weighted least-squares estimation. The measurement noise is constructed to imitate the vehicular camera and radar to the highest degree possible. The overtaking scenarios are performed on a highway with three lanes, see Fig. 2.1. The data processing in the sensors are modelled to use a Kalman filter and the central processor, referred to as the *master fusion filter* (MFF), use an information filter for processing sensor data. Additionally, DMIA and NA are implemented on real logged data. The absence of a true state value, due to unknown position of target vehicles, makes the simulation on real logged data impossible to evaluate from a error and consistency point of view. Thus the fused target trajectory of DMIA and NA is compared.

In order to evaluate the robustness of the algorithms, sensor failures where one or both sensors fail to communicate with the MFF are implemented into the simulations. The results are then compared to the previous simulations where both sensors are working.

All results from the simulations are based on Monte Carlo simulations in order to obtain proper statistical results.

2.2 Related work

A lot of research has been done in the field of sensor fusion since the introduction of the Kalman filter in the 1960's and target tracking has been extensively studied for military, aeronautics, robotics and automation applications for quite some time. For systems that are able to detect all surrounding traffic, the motivation for a decentralized architecture has been addressed in [2], to name one, for its modularity, scalability and robustness.

The use of the information filter in the central processor for a decentralized architecture has been motivated in [2], [3], [4] due to its ability to retain consistency in the state estimate with correlated data and its simpler estimation update equations.

Work has been done on tracking of the time-dependent cross-correlation coefficient between two signals in [5]. The results could be applied to a centralized architecture where the estimation errors from different sensors are correlated.

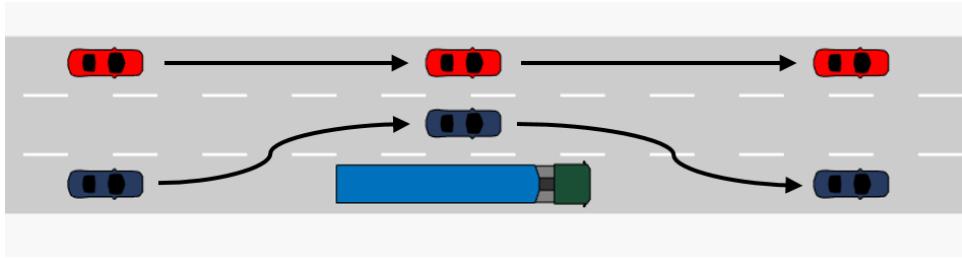


Fig. 2.1. Two cars passing by a heavy-duty vehicle on a highway with three lanes. The car in the far left lane is driving in a straight line while the other car has to perform some maneuvers during the overtaking. The overtaking maneuvers are similar to the simulated overtakes studied in the thesis.

2.3 Thesis outline & objective

The objective of this master thesis is to evaluate the performance of two decentralized IMF algorithms, ODA and DMIA, and a third decentralized fusion algorithm, NA, on vehicular camera and radar data with focus on state estimation and target tracking. Furthermore, this master thesis investigates if the novel decentralized algorithm DMIA is suitable for the next generation of ADAS systems at Scania CV AB. The DMIA is based on the same algorithm as ODA but receives less information. It shall thus be investigated how DMIA performs compared to ODA to conclude if DMIA yields acceptable results.

The report starts with the basics and general concepts of sensor fusion in Chapter 3. The first half of the chapter explains concepts of target tracking and state estimation of an object moving with constant acceleration and what assumptions need to be made. Then the sensor properties are examined. The second half of the chapter focuses on decentralized fusion architectures. IMF is explained and the construction of DMIA and NA is explained in detail. The non-linear properties of the radar are linearized in order to approximate the estimation error covariance used in DMIA. In Chapter 4 are the simulated scenarios explained including simulation setup, robustness to sensor failures and consistency of estimators. In addition, an optimization algorithm for handling singular estimation error covariances used in DMIA is introduced. The solution to the optimization algorithm is built on the knowledge of eigenvalue decomposition. In Chapter 5 are the results and sensitivity analysis from the simulations presented from 100 Monte Carlo runs. The algorithms are evaluated from a root mean squared error (RMSE) point of view. Additionally the consistency of the estimators are analyzed since in order for a filter to be optimal the estimates of the filter need to be consistent. The results and performance of the fusion algorithms are discussed in Chapter 6. Lastly, conclusions about limitations and strengths of the algorithms are presented in Chapter 5 and future work and extensions of the topic is discussed in Chapter 8.

Sensor fusion

Sensor fusion is the art of combining data from different sensors into one final value with the goal to create an as accurate as possible estimate of the reality. In order to combine data from sensors for target tracking, a fusion architecture and fusion algorithms are required. Sensor fusion architectures can be divided into three general architectures: centralized, decentralized and distributed, see Fig. 3.1. In the centralized architecture a central processor has access to the raw measurement data sent from the sensors. In a decentralized architecture the central processor has access to preprocessed data from local sensors. In a fully distributed architecture there is no central processor and no superior/subordinate relationship between the nodes. All nodes can communicate with each other, subject to hardware connectivity constraints. For ADAS systems the most likely fusion to take place is the fusion of preprocessed tracked sensor objects which motivates the choice of a decentralized architecture [3]. A decentralized architecture is natural for many applications with different kinds of sensors, e.g. radar and camera. When more and more sensors are used the computational problem becomes bigger and having a decentralized architecture where multiple fusion nodes process sensor data and communicate with each other will improve upon local results while maintaining a limited bandwidth. Additionally, a decentralized architecture is more robust to sensor failures or module losses which increase the safety of the system. However, a possible problem is combining the results from two fusion nodes and approaches like distributed Bayes' rule, IMF and a hierarchical fusion algorithm were discussed in [6]. A problem when combining sensor data is how to handle the possible correlated inputs from the sensors. IMF algorithms have been presented in [2] [3] [4], to name a few, and shown to be very useful in that they decorrelates sensor inputs before they are fused into the global estimate.

Sensor fusion for ADAS systems is today mainly used to detect objects around the vehicle and associate them correctly, i.e. identify which sensor observation belong to which target [7]. The field of state estimation and target tracking is a step in the direction of making heavy-duty vehicles more autonomous. The ability to predict and estimate where a target is moving around the vehicle will be necessary for the next generation of ADAS to improve safety, comfort and driving efficiency [3].

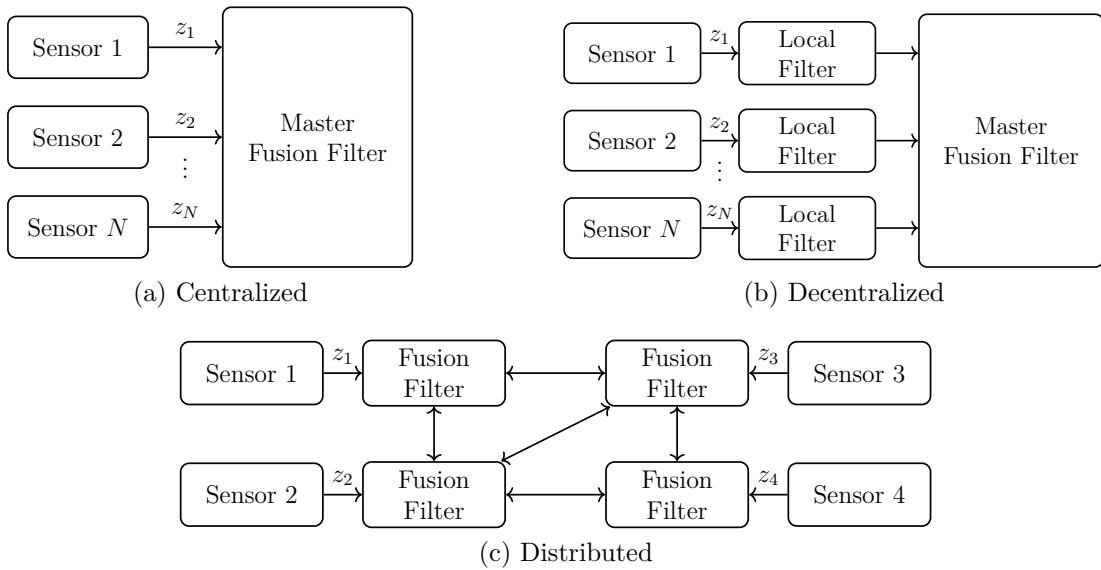


Fig. 3.1. General sensor fusion architectures. In a centralized architecture (a) the MFF receives raw measurement data from the sensors which are to be fused. In a decentralized architecture (b) each sensor has its local filter that process the measurement data before it is sent to the MFF. The MFF then fuses the estimates from the local filters. In a distributed architecture (c) there is no superior/subordinate relationship between the fusion nodes. There is no MFF that computes globally fused estimates, instead there are several fusion filters communicating with each other and perform their own fusion of raw measurement data and transmitted estimates from other fusion filters.

Consider the time-discrete linear dynamic system of a state vector x_k with a measurement given by a vector z_k

$$x_{k+1} = F_k x_k + \Gamma_k v_k, \quad (3.1a)$$

$$z_k = H_k x_k + w_k, \quad (3.1b)$$

where v_k is the process noise and Γ_k its corresponding noise gain; w_k is the measurement noise. The matrices F , Γ and H are assumed to be known.

The process noise and measurement noise are assumed to be zero-mean white Gaussian noise with the properties

$$\mathbb{E} \left[\begin{pmatrix} v_k \\ w_k \end{pmatrix} \begin{pmatrix} v_k^\top & w_k^\top \end{pmatrix} \right] = \begin{pmatrix} Q_k & 0 \\ 0 & R_k \end{pmatrix}, \quad \forall k, \quad (3.2)$$

When deriving the sensor fusion architectures an assumption that the measurement noise is cross-uncorrelated has been made. The reason for this is because the method for handling cross-correlated measurement noise is not straight forward and

cross-correlation is sometimes difficult to detect [8]. The algorithms used in the architectures derive from the Kalman filter and the information filter. Both filters are described more thoroughly in Appendix A and Appendix B. The system (3.1) is expressed on a linear form due to the simplicity when deriving the fusion algorithms. In the case of a non-linear system, Taylor approximation is applied in order to linearize the system. The case of a non-linear system is handled later on in the chapter.

3.1 Target tracking

Target tracking is the state estimation of a moving object based on measurements of the object by remote sensors. The sensors are either at a fixed location or on a moving platform [9]. The targets are modelled according to a dynamic model that is derived from assumptions of the surrounding traffic environment, e.g. highway roads, urban roads or country roads. The general concept of target tracking for ADAS on a highway road is illustrated in Fig. 3.2. The target, here a car, can be assumed to move according to a dynamic state equation with a certain process noise. In the case when the motion model is non-linear, Taylor approximation can be applied in order to linearize the system. The sensors mounted on the truck measures state properties, e.g. relative position and velocity, of the target with a certain measurement noise. The measurement equation may not be linear, depending on the variables that the sensor is measuring, and again Taylor approximation may be applied in order to linearize the equation. The sensors detect surrounding objects, see Fig. 3.3, and the detection points can be combined into one point representing the target. Hence, the target tracking can be simplified into tracking one point per target instead of several points.

The target tracking is focused on objects driving on highway roads, i.e. roads with two lanes, or more, with vehicles traveling in the same direction. The dynamic model of a target is assumed to be a constant acceleration model with the state vector

$$\xi = \begin{pmatrix} x & y & \dot{x} & \dot{y} & \ddot{x} & \ddot{y} \end{pmatrix}^\top. \quad (3.3)$$

The constant acceleration dynamic model is a time-discrete state space system on the form (3.1a) where the transition matrix and the process noise gain are

$$F_k = \begin{pmatrix} 1 & 0 & \Delta t_k & 0 & \frac{\Delta t_k^2}{2} & 0 \\ 0 & 1 & 0 & \Delta t_k & 0 & \frac{\Delta t_k^2}{2} \\ 0 & 0 & 1 & 0 & \Delta t_k & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t_k \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \Gamma_k = \begin{pmatrix} \frac{\Delta t_k^3}{6} & 0 \\ 0 & \frac{\Delta t_k^3}{6} \\ \frac{\Delta t_k^2}{2} & 0 \\ 0 & \frac{\Delta t_k^2}{2} \\ \Delta t_k & 0 \\ 0 & \Delta t_k \end{pmatrix},$$

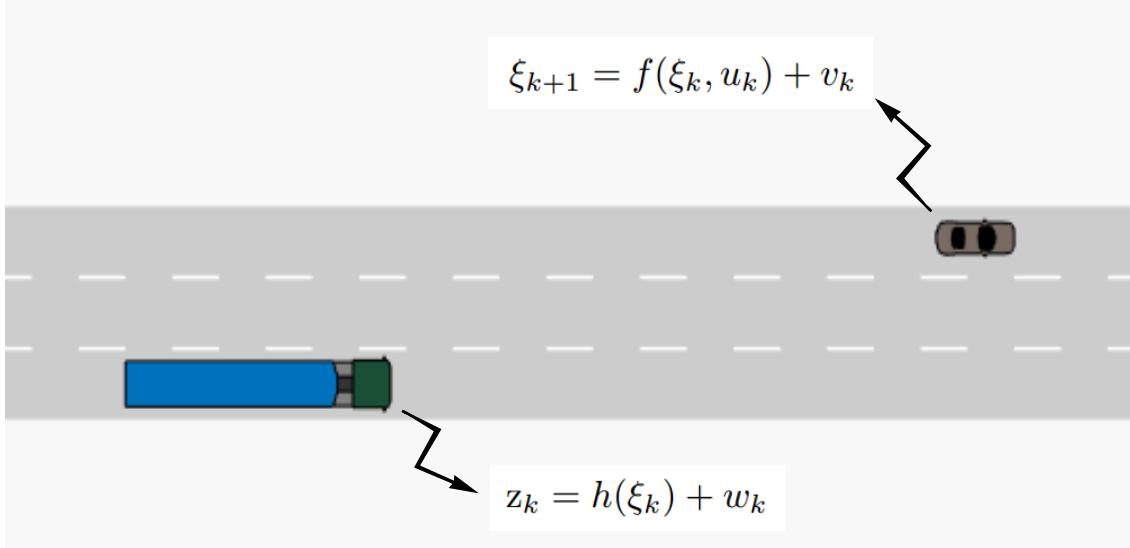


Fig. 3.2. Dynamic concept of the motion of a target and measurement from sensors. The motion of a target, depicted as a car in the figure, can be modelled to move according to a dynamic equation $\xi_{k+1} = f(\xi_k, u_k) + v_k$ where $f(\xi_k, u_k)$ is the motion dynamics and v_k the corresponding process noise. The sensors mounted on the heavy-duty vehicle observes the target and measures the properties, $z_k = h(\xi_k) + w_k$ where $h(\xi_k)$ is the measurement equation and w_k its corresponding measurement noise.

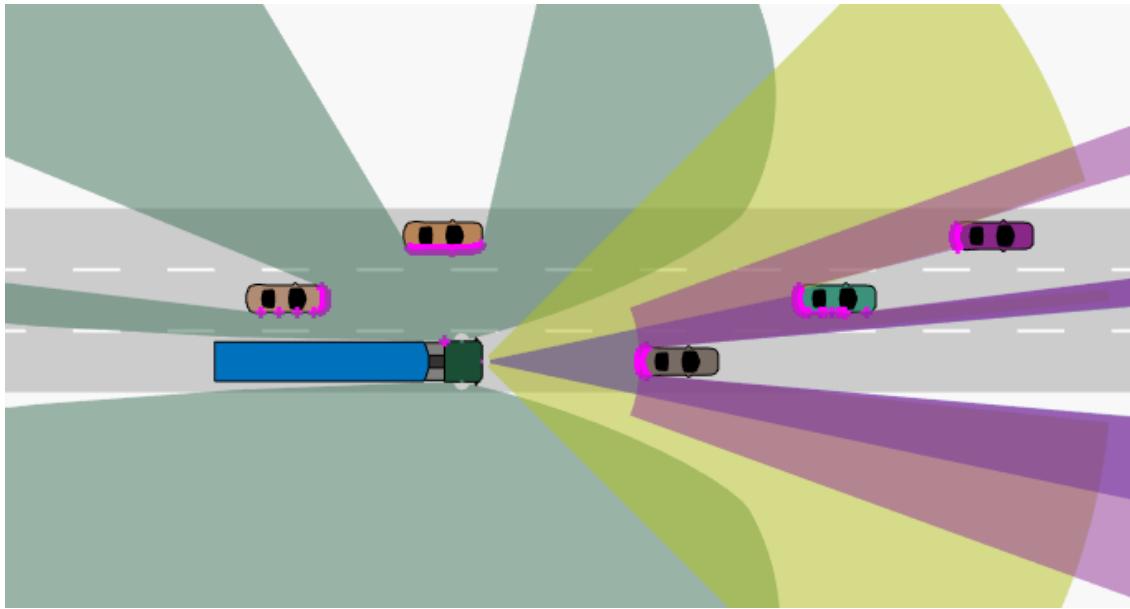


Fig. 3.3. Sensor detections on surrounding highway traffic. The purple points on the targets, depicted as cars in the figure, are detection points measured by the sensors. It can be seen how the field-of-view of the sensors are compromised when a target enters the sensor field, i.e. an object behind a target will be harder to detect due to the cancellation of the field-of-view of the sensor. The sensor configuration of the heavy-duty vehicle is vehicular camera and radar as well as radars mounted on the sides of the vehicle. In this configuration the host vehicle have almost a 360° field-of-view.

with $\Delta t_k = t_{k+1} - t_k$.

Together with the state space equation and the measurements from all sensors the dynamic system is given by

$$\xi_{k+1} = F_k \xi_k + \Gamma_k v_k, \quad (3.4a)$$

$$z_{i,k} = H_{i,k} \xi_k + w_{i,k}, \quad \forall i = 1, \dots, N, \quad (3.4b)$$

where $z_{i,k}$ is the measurement from sensor i and $w_{i,k}$ its corresponding measurement noise. The properties of the process noise and measurement noise are

$$E \left[\begin{pmatrix} v_k \\ w_{i,k} \end{pmatrix} \begin{pmatrix} v_k^\top & w_{i,k}^\top \end{pmatrix} \right] = \begin{pmatrix} Q_k & 0 \\ 0 & R_{i,k} \end{pmatrix}, \quad \forall k. \quad (3.5)$$

Furthermore, we assume that the measurement noise between the sensors are cross-uncorrelated, i.e. $w_{i,k}$ and $w_{j,k}$ are independent for $i \neq j$ at time instant k .

3.1.1 Sensor properties

The measurement equation for each sensor, often referred to as the sensor output, can either be linear or non-linear depending on the sensor properties. The measurements from camera and radar have varying degrees of accuracy, which if combined can obtain a better estimate than the local sensor estimates [6]. In the setup the radar has high accuracy in longitudinal direction and low accuracy in lateral direction. In addition, the camera has high accuracy in lateral direction and low accuracy in longitudinal direction. Hence, by combining the strengths from both sensors, a better estimate can be obtained compared to if only one of the sensors is used.

Camera

The camera is assumed to measure position and velocity of a target, hence the output equation for the camera is given by the linear equation

$$z_{c,k} = H_{c,k} \xi_k + w_{c,k}, \quad (3.6)$$

where

$$H_{c,k} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \quad (3.7)$$

and the index c denotes the camera.

The variances for the camera variables used in the simulations are illustrated in Fig. 3.4. The variance of the variables is constructed in such a way that the variance increase when the distance to the target increases. The variances in longitudinal distance, x -direction, and longitudinal velocity, \dot{x} -direction, does first decrease in magnitude before they start to increase with the distance. The covariance matrix for the measurement noise is given by

$$R_{c,k} = \begin{pmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_{\dot{x}}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{y}}^2 \end{pmatrix}. \quad (3.8)$$

Radar

The radar is assumed to measure angle, radial distance and radial velocity to a target, see Fig. 3.5. As a consequence, the output equation from the radar is a non-linear equation given by

$$z_{r,k} = h_{r,k}(\xi_k) + w_{r,k}, \quad (3.9)$$

where the angle, radial velocity and radial distance are computed as

$$\begin{pmatrix} \theta \\ \dot{r} \\ r \end{pmatrix} = \begin{pmatrix} \tan^{-1}(y/x) \\ \sqrt{\dot{x}^2 + \dot{y}^2} \\ \sqrt{x^2 + y^2} \end{pmatrix}. \quad (3.10)$$

The variances of the radar variables used in the simulations are illustrated in Fig. 3.6. The radar is assumed to be very good at measuring radial distance and radial velocity to a target, hence the variances for those variables are constant no matter the distance to a target. In the case of the variance of the measured angular to a target, the radar has two working modes depending on the distance between host vehicle and target. The first mode of the radar measuring the angle works up until a certain distance between the host vehicle and the target being tracked, exceeding that distance the other mode takes over measuring the angle to the target and vice versa. The different modes are referred to as ranges of the radar, i.e. short range and long range. The covariance for the radar measurement noise is given by

$$R_{r,k} = \begin{pmatrix} \sigma_\theta^2 & 0 & 0 \\ 0 & \sigma_{\dot{r}}^2 & 0 \\ 0 & 0 & \sigma_r^2 \end{pmatrix}. \quad (3.11)$$

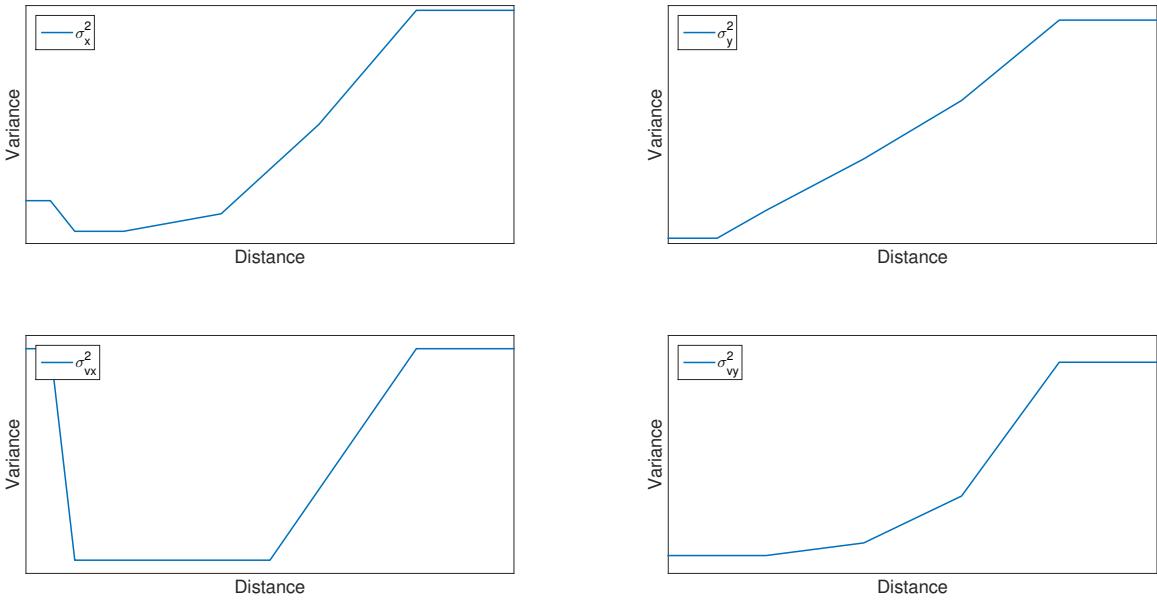


Fig. 3.4. Variances for camera variables x , y , \dot{x} and \dot{y} . All variances increase as the distance increase. What is special about the camera is that the variance for x and \dot{x} first decrease before they start to increase as the distance increase. In addition, the variance for x and \dot{x} increases faster than the variance for y and \dot{y} . As a consequence the camera is more accurate in measuring y and \dot{y} due to the lower variance.

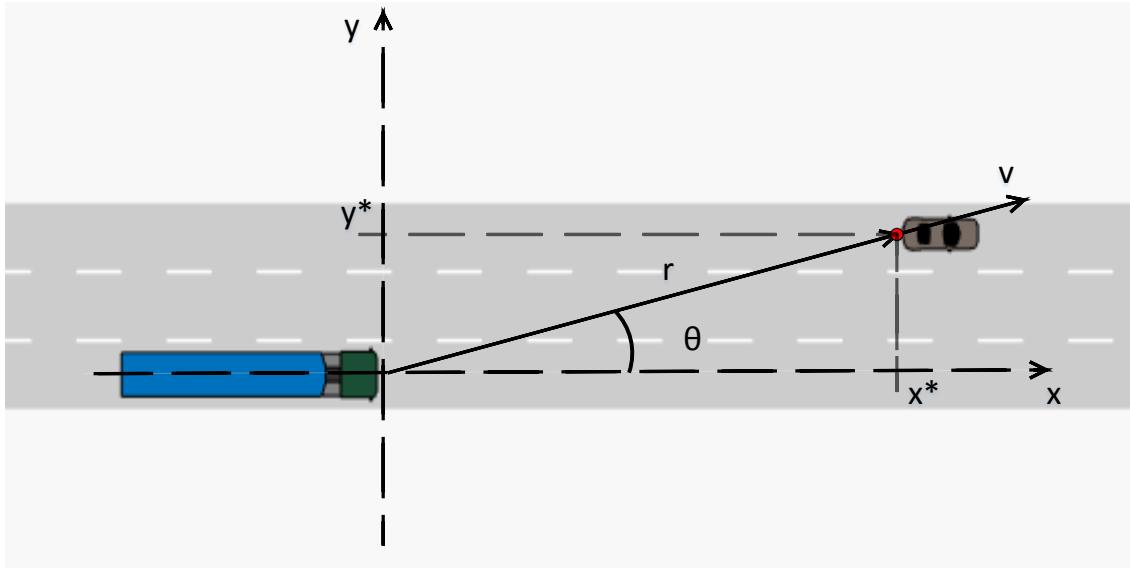


Fig. 3.5. Variables measured by the radar. The radar measures angle θ , radial distance r and radial velocity \dot{r} to a target. The figure depict the origin of the host vehicle where the radar is mounted. Both sensor are mounted in such a way that they work in the same coordinate system.

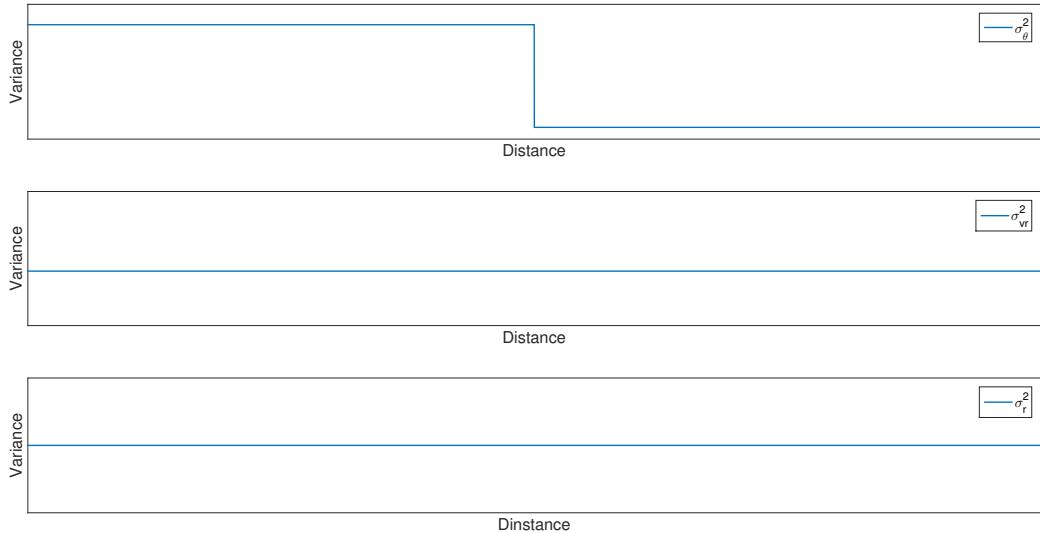


Fig. 3.6. Variances for radar variables θ , \dot{r} and r . The short range and long range mode of the radar can be seen in the figure depicting the variance for θ . When the target exceeds a certain distance from the host vehicle the radar switch from short range to long range. As a consequence, the radar is more accurate when the target is further away from the host vehicle than when it is close.

3.2 Decentralized fusion architectures

Theoretically, a centralized architecture is globally optimal in the mean squared error (MSE) sense but will be computationally heavy if the dimension of the measurement vector is larger than the dimension of the state [10]. This is often the case when measurements from several sensors are to be fused. As earlier mentioned, when fusing preprocessed data from different kinds of sensors, a decentralized architecture is favorable in the sense that it is modular, practical and scalable. The problem within this architecture is to find a fusion algorithm that is able to retain consistency and maintain a steady global tracking over time as targets move through sensors field of view. In addition, the signals from sensors may be asynchronous and out-of-sequence which needs to be handled by the algorithm. In [8] and [11] it was presented that a decentralized Kalman filter is equal to its corresponding centralized Kalman filter, which further motivates the choice of a decentralized architecture. Many applications with a decentralized architecture have used a system of cascaded Kalman filters, i.e. treating processed estimates from sensor filters as measurements in a second Kalman filter [1]. However, applying a Kalman filter to such data neglects the correlation that arises between tracks due to the common process noise and motion model used. Ignoring that correlation leads to inaccurate estimates since the state estimation error covariance becomes smaller than its true value. Several approaches to handle the correlation have been investigated when fusing sensor data such as adaptive Kalman filter, cross-covariance, covariance intersection and covariance union [3], [1].

3.2.1 Information matrix fusion

IMF is based on the information filter, i.e. the information form of the estimate error covariance and its corresponding information filter state, see Appendix B. First introduced in 1997, the IMF algorithm was compared against other algorithms that calculated the cross covariance between two track estimates [12]. The architecture, however, lets the sensors send feedback to one another in order to achieve optimal fused estimates.

Let the state estimate at time t_j be denoted by the conditional mean

$$\hat{\xi}(j|k) = \hat{\xi}_{j|k} \triangleq \mathbb{E}[\xi(j), \{Z\}_0^{t_k}], \quad (3.12)$$

where $\{Z\}_0^{t_k}$ is the sequence of observations, or the sequence of information, available at time t_k . The estimation error, or estimation residual, is defined by

$$\tilde{\xi}_{j|k} \triangleq \xi(j) - \hat{\xi}_{j|k}. \quad (3.13)$$

The estimation error is assumed to be zero-mean white Gaussian noise and its corresponding covariance is given by

$$P(j|k) = P_{j|k} \triangleq \mathbb{E} [\tilde{\xi}_{j|k} \tilde{\xi}_{j|k}^\top | \{Z\}_0^{t_k}]. \quad (3.14)$$

The main idea of the IMF algorithm is to decorrelate the input tracks before they are fused into the global track. The information graph for the IMF algorithm is illustrated in Fig. 3.7. At time t_{k-2} sensor i measures $z_{i,k-2}$ and the measurement is added to the sequence of observations $\{Z_i\}_0^{t_{k-2}}$ for sensor i . Let $\{Z_i, Z_j\}_0^{t_{k-2}}$ be the global fused sequence of observations available up until time t_{k-2} . When a new measurement is recorded at t_{k-1} , by both sensors, the new information is fused into the global sequence which becomes $\{Z_i, Z_j\}_0^{t_{k-1}}$. The new information between time t_{k-2} and t_{k-1} for sensor i is denoted as $\{Z_i\}_{t_{k-2}}^{t_{k-1}}$, the new measurement at time t_{k-1} is added to the global information by decorrelating the information between t_{k-2} and t_{k-1} and simply fusing the new information $\{Z_i\}_{t_{k-2}}^{t_{k-1}}$ into the global information.

How the new information is fused into the global information depends on the architecture. Consider an architecture with N sensors observing a target and sending their information to the MFF. Let the globally fused state estimate and its corresponding estimation error covariance be denoted by (3.12) and (3.14) respectively. The state estimate and its corresponding estimation error covariance for sensor i is given by

$$\hat{\xi}_i(j|k) = \hat{\xi}_{i,j|k} \triangleq \mathbb{E}[\xi_i(j), \{Z_i\}_0^{t_k}], \quad \forall i = 1, \dots, N, \quad (3.15)$$

$$P_i(j|k) = P_{i,j|k} \triangleq \mathbb{E} [\tilde{\xi}_{i,j|k} \tilde{\xi}_{i,j|k}^\top | \{Z_i\}_0^{t_k}], \quad \forall i = 1, \dots, N, \quad (3.16)$$

where the statistical properties are discussed more thoroughly in Appendix A.

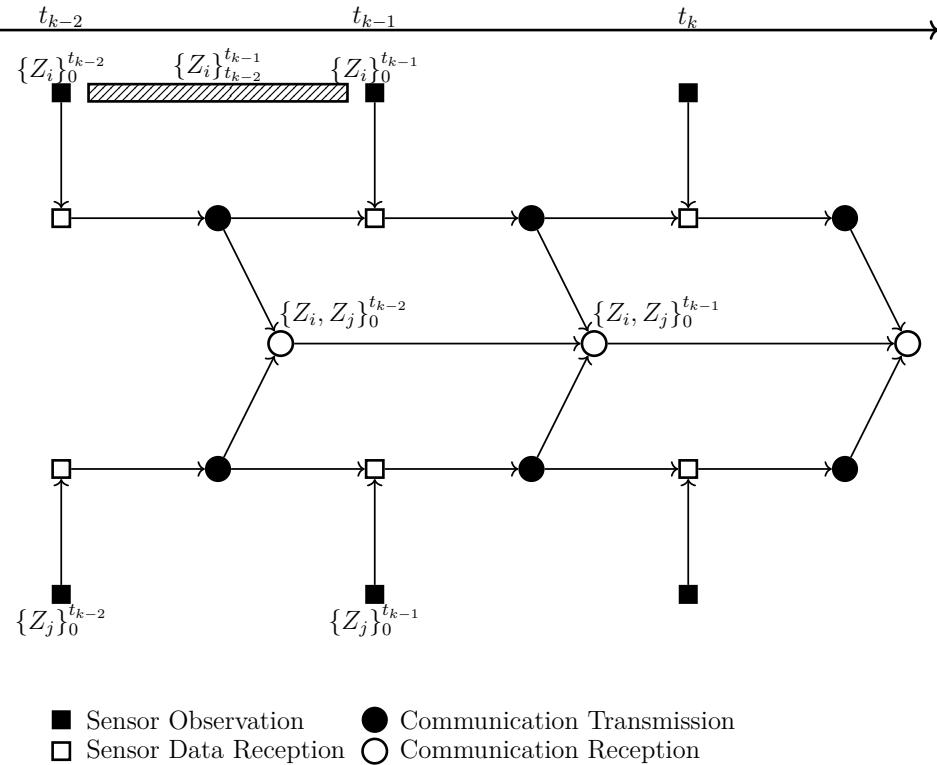


Fig. 3.7. Information graph of the IMF algorithm. Two sensors, i and j , receive measurements at time t_{k-2} , t_{k-1} and t_k and between time t_{k-2} and t_{k-1} is the new information for sensor i depicted. The new information from sensor i , $\{Z_i\}_{t_{k-2}}^{t_{k-1}}$, is fused into the global track $\{Z_i, Z_j\}_{t_{k-2}}^{t_{k-1}}$, the corresponding process is done for sensor j . Additionally, in the figure is the potential time-delay of a decentralized architecture depicted. The central processor, communication receptor, does always receive the new information with a delay.

In decentralized fusion the MFF receives estimates from the sensors local filters. The decentralized architecture can choose to send feedback to the local sensor filters in order to reduce the covariance of each local tracking error [8]. But as mentioned earlier, presented in [8] and [11], a decentralized architecture with or without feedback from MFF to local sensor filters is equivalent to its corresponding centralized IMF filter. The advantage with decentralized IMF is that it is more robust to fault detections, sensor failures and is modular. A disadvantage is that it can be less accurate, e.g. due to time-delays from local sensor filters. Another disadvantage with the decentralized IMF for multiple sensor fusion is that, for each sensor, the previously fused sensor-level track must be saved in order to carry out the decorrelation process [3]. In Fig. 3.8a and Fig. 3.8b the two decentralized architectures are presented.

The MFF update equation in a decentralized architecture without feedback, see Fig. 3.8a, needs to carry out the decorrelation process. This is the architecture that ODA uses. The updated global information matrix and its corresponding information filter state where only the new information is added is given by

$$P_{k|k}^{-1} = P_{k|k-1}^{-1} + \sum_{i=1}^N \left(P_{i,k|k}^{-1} - P_{i,k|k-1}^{-1} \right), \quad (3.17)$$

$$P_{k|k}^{-1} \hat{\xi}_{k|k} = P_{k|k-1}^{-1} \hat{\xi}_{k|k-1} + \sum_{i=1}^N \left(P_{i,k|k}^{-1} \hat{\xi}_{i,k|k} - P_{i,k|k-1}^{-1} \hat{\xi}_{i,k|k-1} \right). \quad (3.18)$$

When there is feedback from the MFF to local sensor the local predicted information matrix of sensor i is replaced by the global predicted information matrix, i.e. $P_{i,k|k}^{-1} = P_{k|k}^{-1}$, and the same is true for the locally predicted information filter state, see Fig. 3.8b. The MFF update equations for the decentralized architecture with feedback is thus

$$P_{k|k}^{-1} = \sum_{i=1}^N P_{i,k|k}^{-1} - (N-1)P_{k|k-1}^{-1}, \quad (3.19)$$

$$P_{k|k}^{-1} \hat{\xi}_{k|k} = \sum_{i=1}^N P_{i,k|k}^{-1} \hat{\xi}_{i,k|k} - (N-1)P_{k|k-1}^{-1} \hat{\xi}_{k|k-1}. \quad (3.20)$$

3.2.2 Decentralized-minimum-information algorithm

Consider a decentralized sensor fusion architecture where the MFF only has access to the updated state estimates but not their corresponding error covariances from the local sensor filters, see Fig. 3.8c. Let the architecture be referred to as the DMIA architecture. To perform a IMF, approximations of the updated and predicted error covariance must be made. In addition, the predicted state estimate does also need to be estimated.

In order to make an approximation of the error covariance of an estimate, properties of the sensor is needed. To understand the process of estimating the covariance some theory about Taylor expansion and its statistical properties is needed.

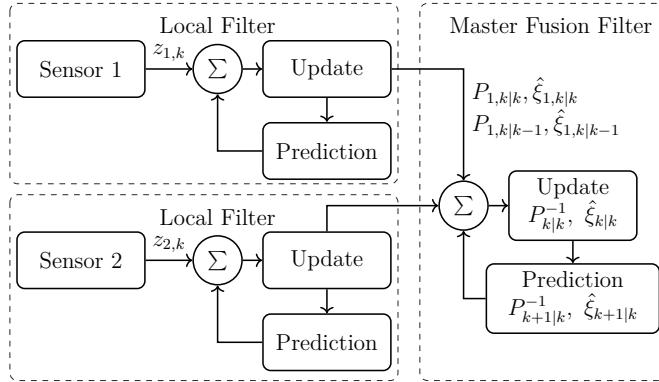
Consider the non-linear equation system

$$Z_k = f(X_k) = \begin{pmatrix} f_1(X_k) \\ \vdots \\ f_N(X_k) \end{pmatrix},$$

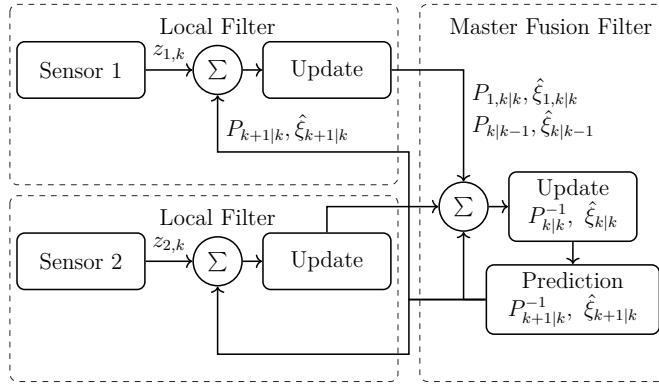
where $X_k = (x_1 \ \cdots \ x_N)^\top$.

The statistical properties of X_k is assumed to be known and given by

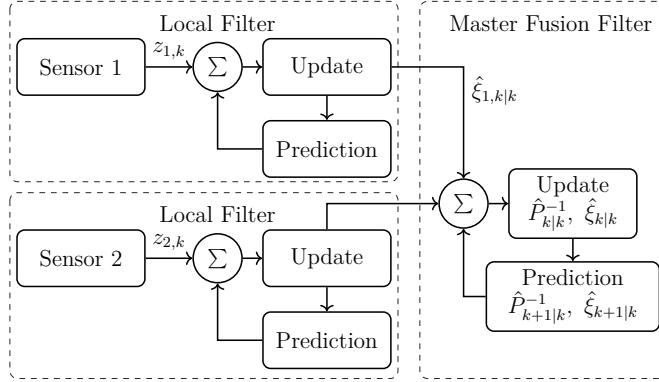
$$\mathbb{E}[X_k] = \mu_x, \quad \text{Var}(X_k) = \Omega.$$



(a) Decentralized architecture without feedback, ODA



(b) Decentralized architecture with feedback



(c) DMIA architecture

Fig. 3.8. Decentralized IMF architectures. The sensors use their own local filter and sends state estimates and the corresponding error covariances to the MFF. In (a) is the architecture without feedback depicted, referred to as ODA architecture, and in (b) is the architecture with feedback depicted. The difference between (a) and (b) is that (a) uses local predictions, $\hat{\xi}_{i,k|k-1}, P_{i,k|k-1}$, $\forall i = 1, \dots, N$, and (b) uses global predictions, $\hat{\xi}_{k|k-1}, P_{k|k-1}$, in the local sensor filters respectively. In (c) is the DMIA architecture presented. The MFF only have access to updated state estimates from the local sensor filters and needs to approximate the updated error covariances $P_{r,k|k}$ and $P_{c,k|k}$ and their predictions in order to carry out the IMF.

Z_k can be linearized using Taylor expansion

$$Z_k = f(X_k) = f(\mu_x) + \nabla f(\mu_x)^\top (X_k - \mu_x) + \text{H.O.T.} \approx f(\mu_x) + \nabla f(\mu_x)^\top (X_k - \mu_x),$$

where

$$\nabla f(X_k)^\top = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix},$$

is the Jacobian of $f(X_k)$ and H.O.T. stands for *higher-order-terms*.

It is now possible to estimate the statistical properties of Z_k as

$$\mathbb{E}[Z_k] = \mathbb{E}[f(X_k)] \approx \mathbb{E}[f(\mu_x) + \nabla f(\mu_x)^\top (X_k - \mu_x)] = f(\mu_x) = \mu_z, \quad (3.21)$$

$$\text{Var}(Z_k) = \mathbb{E}[(Z_k - \mu_z)(Z_k - \mu_z)^\top] = \nabla f(\mu_x)^\top \Omega \nabla f(\mu_x). \quad (3.22)$$

For the case when Z_k is the estimator of a variable that can be described by its estimate and error term

$$z_k = Z_k + \tilde{z}_k, \quad (3.23)$$

the property (3.22) of Z_k is also true for the error term \tilde{z}_k .

Consider the camera properties; the camera measures position and velocity, thus an approximation of the acceleration needs to be made since the state vector (3.3) also contains longitudinal and lateral acceleration part. Using central difference approximation for the acceleration the terms can be estimated as

$$\ddot{x}_k \approx \frac{1}{\Delta t_k^2} (x_{k+1} - 2x_k + x_{k-1}), \quad (3.24)$$

$$\ddot{y}_k \approx \frac{1}{\Delta t_k^2} (y_{k+1} - 2y_k + y_{k-1}). \quad (3.25)$$

The estimate approximation from the camera output (3.6) without measurement noise can thus be expressed as

$$\xi_k \approx f_{c,k}(x_k, y_k, \dot{x}_k, \dot{y}_k) = \begin{pmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \\ \frac{1}{\Delta t_k^2} (x_{k+1} - 2x_k + x_{k-1}) \\ \frac{1}{\Delta t_k^2} (y_{k+1} - 2y_k + y_{k-1}) \end{pmatrix}, \quad (3.26)$$

where index c stands for camera and which has the Jacobian

$$\nabla f_{c,k}(\mathbb{E}[x_k, y_k, \dot{x}_k, \dot{y}_k])^\top = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{2}{\Delta t_k^2} & 0 & 0 & 0 \\ 0 & -\frac{2}{\Delta t_k^2} & 0 & 0 \end{pmatrix}. \quad (3.27)$$

Hence, the error covariance from the camera is approximated using the Jacobian (3.27) and the covariance matrix for the camera variables (3.8) as

$$\hat{P}_{c,k|k} = \nabla f_{c,k}(\mathbb{E}[x_k, y_k, \dot{x}_k, \dot{y}_k])^\top R_{c,k} \nabla f_{c,k}(\mathbb{E}[x_k, y_k, \dot{x}_k, \dot{y}_k]). \quad (3.28)$$

Consider the radar properties; the radar measures angle, radial distance and radial velocity of a target. The state estimate, $\hat{\xi}$, of (3.3) at a time t_k can thus be expressed by the non-linear system given by

$$\hat{\xi}_k = f_{r,k}(\theta_k, \dot{r}_k, r_k) = \begin{pmatrix} r_k \cos \theta_k \\ r_k \sin \theta_k \\ \dot{r}_k \cos \theta_k - r \dot{\theta}_k \sin \theta_k \\ \dot{r}_k \sin \theta_k + r \dot{\theta}_k \cos \theta_k \\ (\ddot{r}_k - r_k \dot{\theta}_k^2) \cos \theta_k - (2\dot{r}_k + r_k \ddot{\theta}_k) \sin \theta_k \\ (\ddot{r}_k - r_k \dot{\theta}_k^2) \sin \theta_k + (2\dot{r}_k + r_k \ddot{\theta}_k) \cos \theta_k \end{pmatrix}. \quad (3.29)$$

The mean of θ_{k+1} , r_{k+1} and \dot{r}_{k+1} is assumed to be as follows

$$\mathbb{E}[\theta_{k+1}] = \theta_k + \Delta t_k \dot{\theta}_k + \frac{\Delta t_k^2}{2} \ddot{\theta}_k, \quad (3.30)$$

$$\mathbb{E}[r_{k+1}] = r_k + \Delta t_k \dot{r}_k + \frac{\Delta t_k^2}{2} \ddot{r}_k, \quad (3.31)$$

$$\mathbb{E}[\dot{r}_{k+1}] = \dot{r}_k + \Delta t_k \ddot{r}_k. \quad (3.32)$$

The updated error covariance from the radar can thus be approximated as

$$\hat{P}_{r,k|k} = \nabla f_{r,k}(\mathbb{E}[\theta_k, \dot{r}_k, r_k])^\top R_{r,k} \nabla f_{r,k}(\mathbb{E}[\theta_k, \dot{r}_k, r_k]). \quad (3.33)$$

The update equations for the MFF are then given by

$$P_{k|k}^{-1} = P_{k|k-1}^{-1} + \sum_{i=1}^N \left(\hat{P}_{i,k|k}^{-1} - \bar{P}_{i,k|k-1}^{-1} \right), \quad (3.34)$$

$$P_{k|k}^{-1} \hat{\xi}_{k|k} = P_{k|k-1}^{-1} \hat{\xi}_{k|k-1} + \sum_{i=1}^N \left(\hat{P}_{i,k|k}^{-1} \hat{\xi}_{i,k|k} - \bar{P}_{i,k|k-1}^{-1} \bar{\xi}_{i,k|k-1} \right). \quad (3.35)$$

The predicted state estimate from sensor i , $\bar{\xi}_{i,k|k-1}$, and its corresponding approximated error covariance matrix, $\bar{P}_{i,k|k-1}$, in (3.34) and (3.35) are not sent to the MFF from the sensors, seen in Fig. 3.8c, but are easily obtained using the Kalman equations

$$\bar{\xi}_{i,k+1|k} = F_k \hat{\xi}_{i,k|k}, \quad \forall i = 1, \dots, N, \quad (3.36)$$

$$\bar{P}_{i,k+1|k} = F_k \hat{P}_{i,k|k} F_k^\top + \Gamma_k Q_k \Gamma_k^\top, \quad \forall i = 1, \dots, N. \quad (3.37)$$

3.2.3 Naïve algorithm

Consider the linear equation system for a sensor model with two sensors

$$z = H\xi + \varepsilon = \begin{pmatrix} I \\ I \end{pmatrix} \xi + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \end{pmatrix}, \quad (3.38)$$

where I is the identity matrix. When the system measures real data the equation becomes

$$z = H\hat{\xi} + e = \begin{pmatrix} I \\ I \end{pmatrix} \hat{\xi} + \begin{pmatrix} e_1 \\ e_2 \end{pmatrix}, \quad (3.39)$$

which can be seen as the sensors target estimation. A least-squares solution, $\hat{\xi}^{LS}$, for the system is obtained by minimizing the squared Euclidean norm for the error, $e^\top e$, with respect to $\hat{\xi}$, [13]

$$\frac{\partial e^\top e}{\partial \hat{\xi}} = 0 \implies \hat{\xi}^{LS} = (H^\top H)^{-1} H^\top z. \quad (3.40)$$

Furthermore, ε is assumed to be Gaussian with the properties

$$\varepsilon \sim N(0, R), \quad R = \begin{pmatrix} R_1 & 0 \\ 0 & R_2 \end{pmatrix}, \quad (3.41)$$

where R_1 and R_2 are the error covariance matrices for sensor 1 and sensor 2 respectively.

The weighted least-squares approach is more appropriate for many applications and can be seen as an extension of the least-squares estimate (3.40). According to the Gauss-Markov theorem, an optimal unbiased linear estimator of ξ given the above properties of ε is

$$\hat{\xi}^{WLS} = (H^\top R^{-1} H)^{-1} H^\top R^{-1} z, \quad (3.42)$$

where using the information matrix R^{-1} gives the expression for the minimum variance unbiased estimator [13].

Obtaining the minimum variance unbiased estimator for the sensor model (3.39), the estimate is given by

$$\hat{\xi}^{WLS} = (R_1^{-1} + R_2^{-2})^{-1}(R_1^{-1}z_1 + R_2^{-1}z_2), \quad (3.43)$$

which is equivalent to being the fused estimate from the two sensors.

The weighted least-squares estimate is, as mentioned earlier, unbiased with the properties

$$E[\hat{\xi}^{WLS}] = \xi, \quad E[\hat{\xi}^{WLS}\hat{\xi}^{WLS\top}] = (R_1^{-1} + R_2^{-2})^{-1}. \quad (3.44)$$

Suppose know that the sensor model (3.38) and its corresponding measurement equation (3.39) represents the vehicular radar and camera. The minimum variance unbiased estimator (3.47) will be the fused estimate from the sensor measurements.

NA use the same measurement covariance matrix (3.8) from the camera presented in Section 3.1.1. The algorithm assumes that the radar measurement covariance matrix is on the same form as the camera measurement covariance matrix, i.e. a diagonal matrix with the variance for position and velocity. The algorithm uses a simpler variant of the approximated error covariance matrix (3.33) where the only elements used are the diagonal elements of longitudinal and lateral position and their corresponding velocities. As a consequence, the radar covariance matrix used by NA is on the form

$$R_{r,k} = \begin{pmatrix} \sigma_{x,r}^2 & 0 & 0 & 0 \\ 0 & \sigma_{y,r}^2 & 0 & 0 \\ 0 & 0 & \sigma_{\dot{x},r}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{y},r}^2 \end{pmatrix}, \quad (3.45)$$

where index r denotes the radar.

Hence, the updated state estimate and its corresponding estimation error covariance from NA is given by

$$P_{k|k} = (R_{r,k}^{-1} + R_{c,k}^{-1})^{-1}, \quad (3.46)$$

$$\hat{\xi}_{k|k} = P_{k|k}(R_{r,k}^{-1}\hat{z}_{r,k} + R_{c,k}^{-1}\hat{z}_{c,k}). \quad (3.47)$$

where $\hat{z}_{i,k}$ is the estimated measurement from sensor i at time k .

NA only provides updated state estimates and their corresponding estimation error covariances. No prediction about future states are made by the MFF which can be interpreted as a memoryless property of the algorithm. The architecture for NA is illustrated in Fig. 3.9.

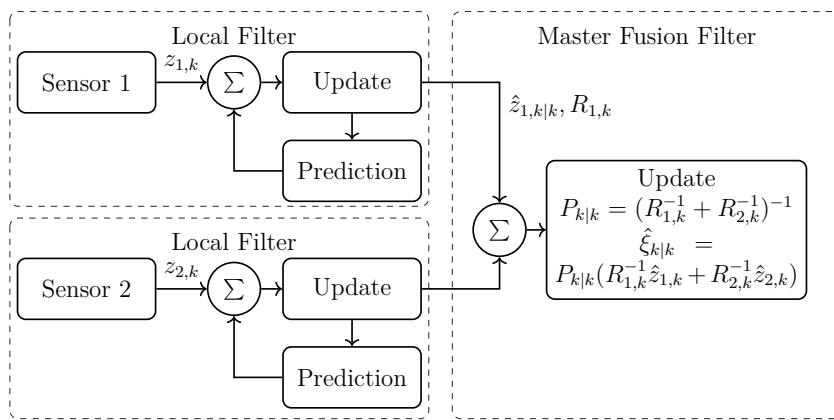


Fig. 3.9. Architecture of NA. The local sensor filters send estimated measurements to the MFF, assumed measurement covariance matrices of the sensors are also sent to the MFF. The estimated measurements and the measurement covariance matrices are then used to compute global estimates using weighted least-squares fusion. The MFF does not compute any prediction about future states which can be interpreted as a memoryless property of the algorithm. As a consequence of the memoryless property the algorithm is sensitive to sensor failures when all sensors fail. If all sensor fail the MFF is not able to compute any global estimates.

Analysis & simulation

The performance of DMIA, ODA and NA is analysed in the simulations. DMIA and ODA use a MFF without feedback, see Fig. 3.8a and Fig. 3.8c, and hence the update equations for the IMF are given by (3.17) and (3.18). The reason architectures with feedback is not simulated is because of the fact that it is more likely that an architecture without feedback is used in ADAS [1], [2], [3]. The performance of the fusion approaches is evaluated by looking at the error for position and velocity as well as the RMSE of a tracked target. In addition, the consistency of the estimators is evaluated looking at the normalized (state) estimation error squared (NEES). To test the accuracy and consistency over time the simulations are evaluated from 100 Monte Carlo runs.

4.1 Simulation setup

Two scenarios are simulated in order to evaluate the fusion algorithms and fusion strategies. The scenarios are two variants of overtaking maneuvers that could occur while driving on a highway, the target trajectories are illustrated in Fig. 4.1. The highway road has three lanes; right, middle and left. The heavy-duty vehicle is driving in the right lane and the vehicle performing the overtaking maneuver starts in the left lane. The first scenario is when the target vehicle drives in a straight line in the left lane. The car travels with constant velocity and no additional maneuvers are performed by the target. In the second scenario the target vehicle performs a lane changing maneuver during the overtaking.

The heavy-duty vehicle in the simulation is mounted with vehicular camera and radar with field-of-view directed in the longitudinal direction of the vehicle, see Fig. 4.2. The radar is working at different range depending on the distance to the object being measured, the two ranges are referred to as short range and long range. The difference between short range and long range is the variance of the measured angle to the target. The variance of the measured angle of the long range radar is smaller than the variance of the short range radar, i.e. the radar is more accurate at long range.

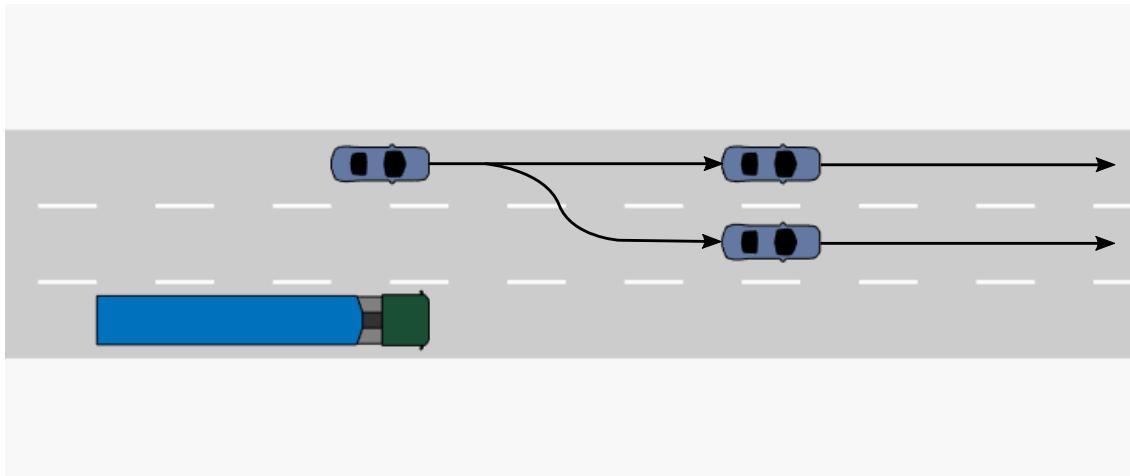


Fig. 4.1. Trajectories for simulated overtaking scenarios. During the first scenario the target vehicle drives in the left lane during the overtaking and does not perform any additional maneuvers. The target vehicle drives with a constant velocity. In the second scenario the target vehicle maneuvers from the left lane to the middle lane during the overtaking. In order for the target vehicle to change lane additional accelerations are initiated. After the the lange changing maneuver the target vehicle drives in the middle lane.

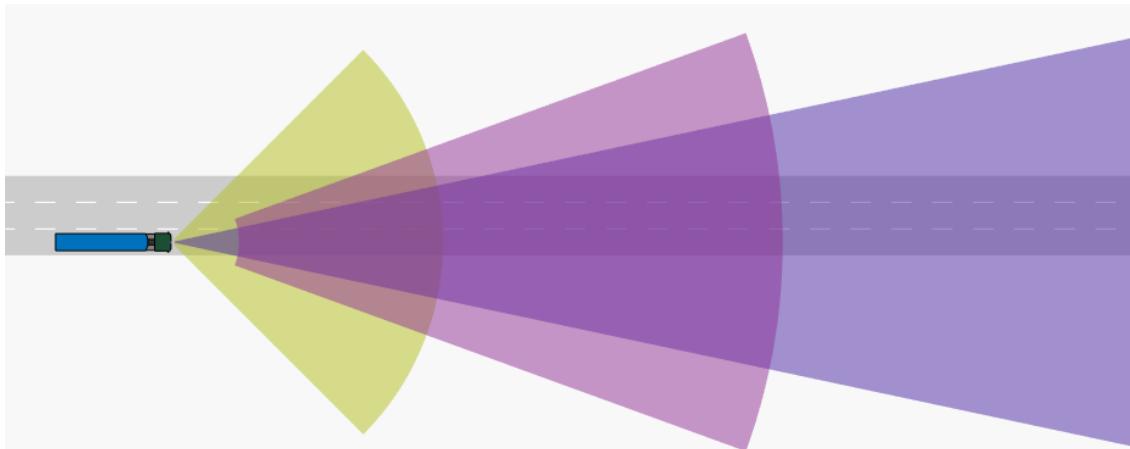


Fig. 4.2. Sensor configuration for test vehicle. The vehicle is mounted with radar and camera directed in the longitudinal direction of the vehicle. The radar has a short range, depicted as the yellow sensor field in the figure, and a long range, depicted as the dark purple sensor field in the figure. In the figure the field-of-view of the camera is depicted with a shorter range than it has in reality.

The initial state vector (3.3) of the target in both scenarios is

$$\xi_0 = (8 \ 8 \ 7 \ 0 \ 0 \ 0)^\top, \quad (4.1)$$

where the target starts to be observed at 8 m longitudinal distance and 8 m lateral distance from the heavy-duty vehicle. The target has a initial speed of 7 m/s in longitudinal direction and 0 m/s in lateral direction. Additionally, the target has

no acceleration in longitudinal or lateral direction.

As presented in Appendix B and [14], IMF is easily initialized without knowing a priori information of the state of the system and as a consequence the updated information matrix and its corresponding information filter state in the MFF is put to zero

$$Y_{f,0|0} = 0, \quad \hat{y}_{f,0|0} = 0, \quad (4.2)$$

where $Y_{f,k|k} \in \mathbb{R}^{6 \times 6}$ and $\hat{y}_{f,k|k} \in \mathbb{R}^{6 \times 1}$.

Scenario 1

In scenario 1 the target vehicle is performing an overtaking in the left lane. The target vehicle is traveling at a constant speed and does not do any lane changing maneuvers. In order for the target vehicle to stay in the lane constraints are implemented on the lateral velocity; if the target vehicle should start to gain lateral velocity an lateral acceleration is induced to counteract the lateral velocity as

$$\begin{cases} \dot{y}_k > 0.1 & \Rightarrow \ddot{y}_k = -10^{-3}, \\ \dot{y}_k < -0.1 & \Rightarrow \ddot{y}_k = 10^{-3}, \end{cases} \quad (4.3)$$

where the lateral velocity, \dot{y} , is in [m/s] and the lateral acceleration, \ddot{y} , in [m/s²]. As a result, the target vehicle drive in a more or less straight line.

Scenario 2

In the second scenario the target vehicle performs an overtaking maneuver and additionally changes lane from the left to the middle lane. The lane changing maneuver takes place between 1-11 s of the 20 s simulation, the acceleration curves can be seen in Fig. 4.3. Additionally, after the change of lane is made the same constraints in scenario 1 is applied in order for the target vehicle to stay in the middle lane.

4.1.1 Singular covariance matrices

It might happen that the approximated error covariance matrices used in DMIA are singular due to numerical effects such as; choice of difference approximations and the order of the Taylor approximation. The DMIA cannot handle singular covariance matrices due to the use of information matrices, i.e. inverted covariance matrices. In the case of singular matrix a optimization algorithm is applied in order to make the matrix non-singular while minimizing the norm of the new matrix and the old one.

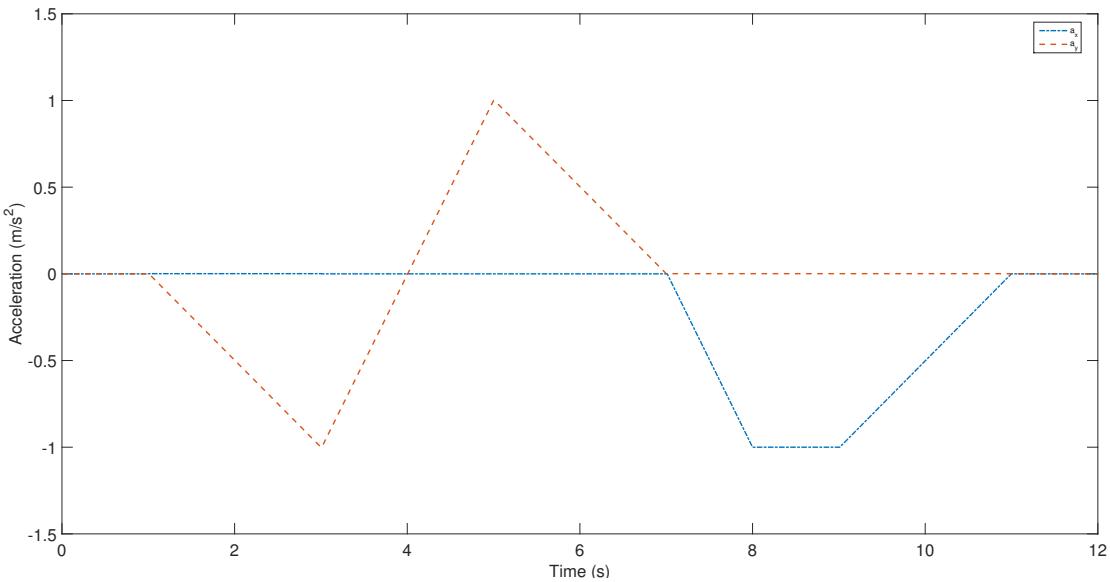


Fig. 4.3. Accelerations of the target vehicle during the second overtaking scenario. The target vehicle performs a change of lane according to the longitudinal and lateral acceleration. The red dotted line depicts the lateral acceleration during the lane changing maneuver. Right after the lane changing maneuver the target vehicle lowers the longitudinal velocity which is depicted as the blue dotted line in the figure.

Consider a covariance matrix Q that is singular, the optimization problem is thus given by

$$\begin{aligned} \min \quad & \|Q - \hat{Q}\|, \\ \text{s.t. } & \hat{Q} \succ 0, \end{aligned} \tag{4.4}$$

where \hat{Q} is the non-singular matrix closest to Q .

The solution \hat{Q} to the optimization problem (4.4) obtained by applying eigenvalue decomposition to Q and modify the diagonal matrix containing all the eigenvalues such that all eigenvalues are positive. The optimization algorithm is illustrated in Fig. 4.4.

4.1.2 Real scenario

DMIA and NA is simulated on real logged data from the vehicular camera and radar. During the simulation target vehicles performs similar overtakes as in scenario 1 and scenario 2. The fused estimates from DMIA and NA are evaluated against measurements from the sensors.

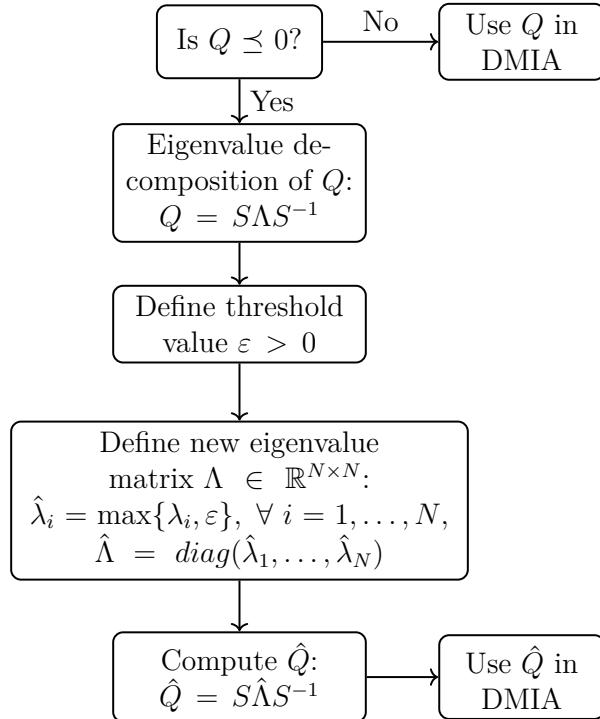


Fig. 4.4. Algorithm for approximating a singular matrix Q . If Q is singular a new matrix \hat{Q} is computed that is non-singular while minimizing the norm of Q and \hat{Q} .

4.2 Robustness

To test the robustness of the architectures, sensor failures are implemented into the simulations. The sensor failure can be seen as the MFF not receiving any information from the local sensor, the sensor is not necessarily broken but the signaling between sensor and MFF fails. The sensor failures are simulated to occur randomly with a given intensity. A continuous uniform random variable $\phi_k \sim U(0, 2)$ is generated for each sensor and if the variable exceeded a threshold value it is registered as a sensor failure. If a sensor failure occurs, it occurs in ODA, DMIA and NA which is necessary to secure the validity of the results from the simulations.

If the random variable for the radar satisfies

$$\phi_{r,k} \in [0.05, 1.95], \quad (4.5)$$

the signaling between radar and MFF is not failing.

If the random variable for the camera satisfies

$$\phi_{c,k} \in [0.1, 1.9], \quad (4.6)$$

the signaling between camera and MFF is not failing.

The failure rate of the sensors is chosen to be approximately 5 % for the radar and approximately 10 % for the camera. The sensors are considered to be of a high quality and thus the failure rate should not be too high.

The structure of ODA and DMIA allows them to compute both predicted and updated estimates whether or not the sensors fail. The IMF approach allows both algorithms to compute predicted or updated estimates even if no new information is received which makes them more robust to sensor failures [3] [2]. The NA do only compute updated state estimates which makes it vulnerable to sensor failures. If both sensors fail at the same time NA will not be able to compute any state estimate.

4.3 Consistency of estimators

The consistency of the ODA, DMIA and NA estimators is evaluated with the normalized (state) estimation error squared (NEES) [15],[16]. An estimator is consistent if it converges in probability to the real value and consistency is necessary for filter optimality. By evaluating the consistency of an estimator, a filter design can be verified.

Assuming that the estimation error, $\tilde{x}_{k|k} \triangleq x(k) - \hat{x}_{k|k} \in \mathbb{R}^{n_x \times 1}$, is approximately Gaussian distributed with the properties

$$\mathbb{E}[\tilde{x}_{k|k}] = 0, \quad (4.7)$$

$$\mathbb{E}[\tilde{x}_{k|k}\tilde{x}_{k|k}^\top] = P_{k|k}, \quad (4.8)$$

the NEES is defined as

$$\epsilon_k = \tilde{x}_{k|k}^\top P_{k|k}^{-1} \tilde{x}_{k|k}, \quad (4.9)$$

where $\epsilon_k \sim \chi^2_{n_x}$.

The NEES is used in order to check whether the estimator is efficient or not, i.e. if the error covariance match the minimal covariance given by the Cramer-Rao lower bound.

Monte Carlo Simulations

If the same test runs for several times the results can be combined to obtain statistical properties of the estimation error. These simulations are often referred to as Monte Carlo simulations, or Monte Carlo runs. In order to evaluate the consistency of an estimator a statistical test based on the results from N independent Monte Carlo runs is carried out.

The sample mean of the independent samples, $\epsilon_{i,k}$, $i = 1, \dots, N$, is given by

$$\bar{\epsilon}_k = \frac{1}{N} \sum_{i=1}^N \epsilon_{i,k}. \quad (4.10)$$

Then under the hypothesis, H_0 , that the filter is consistent and that (4.9) is chi-square distributed with n_x degrees of freedom, the Monte Carlo NEES have the distribution [16]

$$\epsilon_k^{MC} \triangleq N\bar{\epsilon} \sim \chi_{Nn_x}^2. \quad (4.11)$$

The two-sided 95% confidence interval, $[\chi_{Nn_x}^2(0.025), \chi_{Nn_x}^2(0.975)]$, is then computed. If $Nn_x > 100$ the following equation is used to approximate the points on the chi-square distribution [9]

$$\chi_n^2(1 - Q) = \frac{1}{2} \left[\mathcal{G}(1 - Q) + \sqrt{2n - 1} \right]^2, \quad (4.12)$$

where \mathcal{G} for the 95% two-sided probability region is

$$[\mathcal{G}(0.025), \mathcal{G}(0.975)] = [-1.96, 1.96]. \quad (4.13)$$

In the simulations are $n_x = 6$ and $N = 100$, hence, the 95% two-sided confidence interval is

$$[\chi_{600}^2(0.025), \chi_{600}^2(0.975)] = [534, 669], \quad (4.14)$$

and dividing the interval with $N = 100$ the 96% confidence interval for (4.10) is

$$\bar{\epsilon}_k \in [5.34, 6.69]. \quad (4.15)$$

If the estimation error, $\tilde{x}_{k|k}$, is biased it will result in an increasing NEES that will yield unacceptably large values for (4.10). If the statistic is not in the interval (4.15) a bias test can be carried out on the sample mean of the estimation error to identify the problem.

Results

The results of the fusion algorithms are based on the simulations described in Chapter 4 for two overtaking maneuvers, with and without sensor failures, with a sensor configuration of forward-looking camera and radar. The results are based on 100 Monte Carlo runs for each simulated scenario. In addition, the results from the simulation on real logged sensor data for DMIA and NA are presented at the end of the chapter.

5.1 Simulated scenarios

5.1.1 Scenario 1

In the first overtaking scenario the target vehicle performs an overtaking maneuver in the left lane. The error for position and velocity for the first overtaking scenario is depicted in Fig. 5.1. During the first seconds of the overtaking all algorithms perform very similar. For the position error ODA and NA performs very similar but ODA is better. The size of the position error for ODA and NA are very small which can be seen in Fig. 5.1. For DMIA the position error is larger than both ODA and NA, almost five times as large, and has a more volatile behaviour than ODA and NA. The position errors for all algorithms are varying around the value zero, i.e. the error terms are both positive and negative during the overtaking, and there does not seem to be any bias in the error terms.

The velocity error for all algorithms are smaller than their corresponding position error. ODA has the lowest velocity error and NA performs very similar to ODA. DMIA has the largest velocity error which is mostly negative during the overtaking. Towards the end of the overtaking the DMIA performs similar to NA. Both ODA and NA have velocity errors varying around zero, i.e. the velocity error is both negative and positive during the overtaking, and there does not seem to be any bias in the error terms. The negative velocity error tendencies of DMIA could imply that there is a bias when computing the velocity.

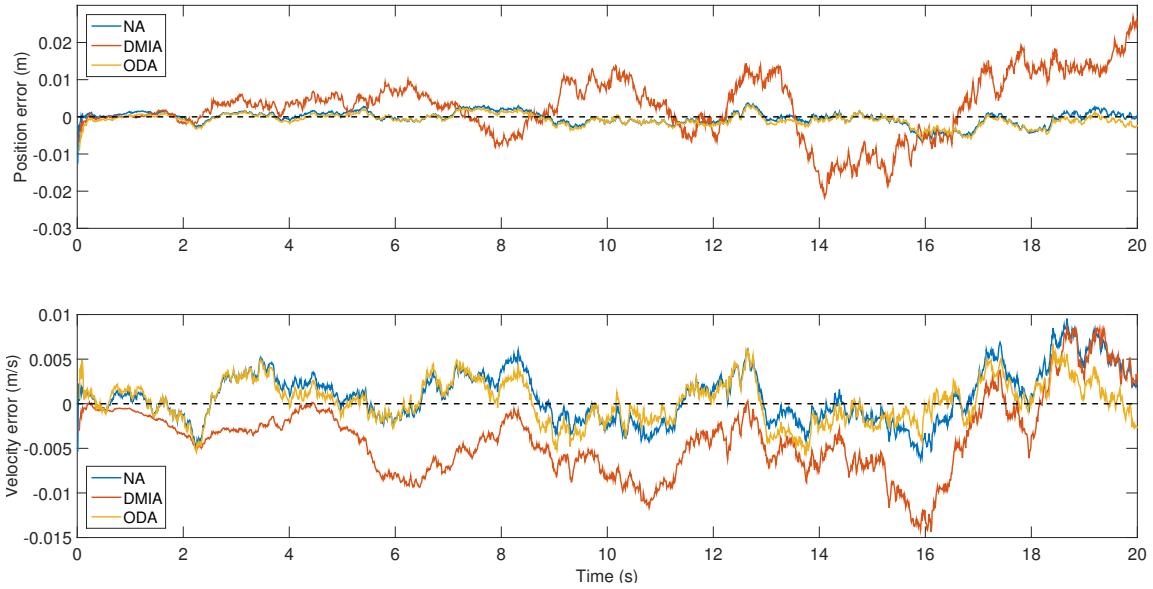


Fig. 5.1. Error for position and velocity for all algorithms during the first overtaking scenario from 100 Monte Carlo runs. ODA has the smallest error for both position and velocity followed by NA and DMIA. The error terms of ODA and NA is very similar for both position and velocity during the overtaking. The error terms of DMIA shows a much more volatile behaviour and the velocity error is mostly negative during the overtaking. The negative behaviour of the velocity error of DMIA could imply that there is bias when computing the velocity.

The RMSE's for position and velocity for the first overtaking scenario is illustrated in Fig. 5.2. In this sensor configuration, ODA performs best of the algorithms in that it makes the most precise estimate of the target with an average position RMSE of 0.0243 m and an average velocity RMSE of 0.0332 m/s. NA performs almost as well as the ODA in terms of RMSE for both position and velocity and has an average position RMSE of 0.0251 m and an average velocity RMSE of 0.0347 m/s. ODA and NA seems to have a more or less constant RMSE for position and velocity during the simulation. The velocity RMSE is slightly more noisy than the position RMSE for ODA and NA. DMIA performs worse than both ODA and NA and shows diverging tendencies in position RMSE. In addition, DMIA shows a much more noisy behaviour than ODA and NA for the position RMSE. The velocity RMSE of DMIA diverge in the beginning of the overtaking but seems to converge towards the values of ODA and NA at the end of the simulation. DMIA has an average position RMSE of 0.1101 m and an average velocity RMSE of 0.0778 m/s. It is thus clear that DMIA has a higher average RMSE for position and velocity compared to ODA and NA. In Table. 5.1 is the average RMSE's of the algorithms are presented.

The results of NEES for ODA, DMIA and NA from the simulation are illustrated in Fig. 5.3. The lower of the two figures is a zoom in of the first where the black dotted lines show the 95 % confidence interval. ODA is consistent over the bigger part of the simulation and converges fast into the confidence region. DMIA has some high extreme values in the beginning of the simulation that could be the result of

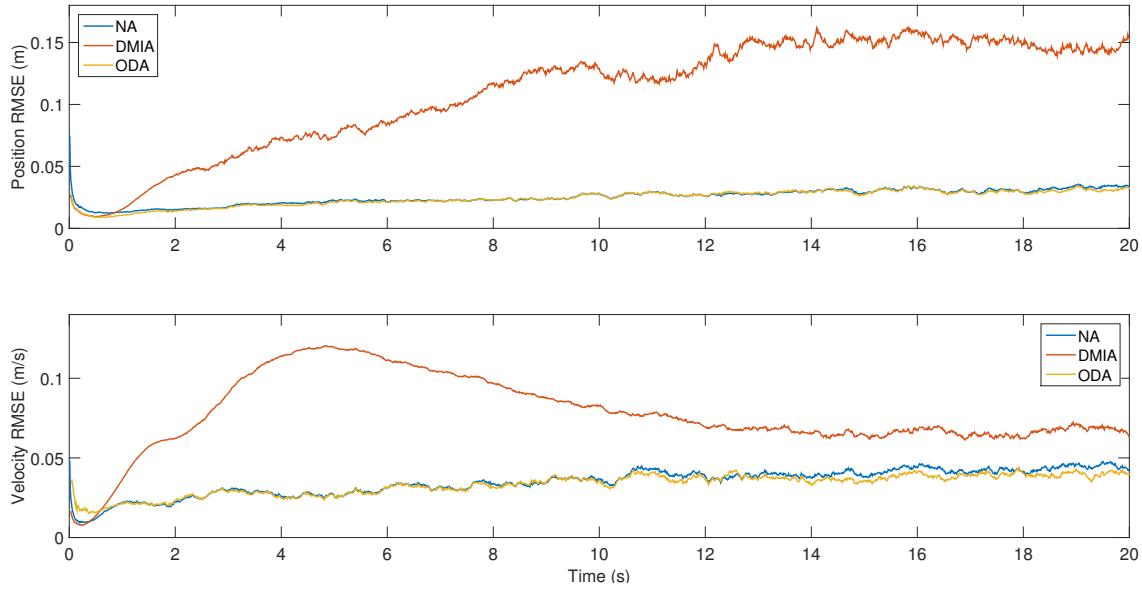


Fig. 5.2. RMSE for position and velocity from 100 Monte Carlo runs from the first simulated overtaking scenario. ODA performs best in that it has the lowest position and velocity RMSE during the simulation. NA performs almost as well as ODA. DMIA shows diverging tendencies in position RMSE and is noisy during the simulation. For velocity RMSE, DMIA diverge in the beginning of the simulation but shows signs of converging towards the RMSE's of ODA and NA during the end of the simulation.

Table 5.1. Average RMSE for position and velocity of the first overtaking scenario. ODA have the lowest RMSE for both position and velocity. NA performs almost as well as ODA. DMIA has the highest RMSE for both position and velocity.

Algorithm	Position (m)	Velocity (m/s)
ODA	0.0243	0.0332
DMIA	0.1101	0.0778
NA	0.0251	0.0347

numerical effects due to the covariance approximations. DMIA shows consistency during the first 8 seconds of the simulation and after that the DMIA estimate is no longer consistent on a 95 % level. Furthermore, the approximated covariance matrices used in DMIA are almost always singular which means that they need to pass through the optimization algorithm described in Chapter 4. The process of turning singular matrices into non-singular could affect the results of the fusion algorithm. It can be seen that the DMIA estimate have some bias affecting the consistency of the filter since it yield unacceptably large NEES values during the later half of the simulation. The NA estimate shows no consistency at all during the simulation and stays more or less at a constant value close to zero. This could be a result of the time-invariant structure of the covariance matrices used in NA and clearly shows that they do not yield an optimal filter.

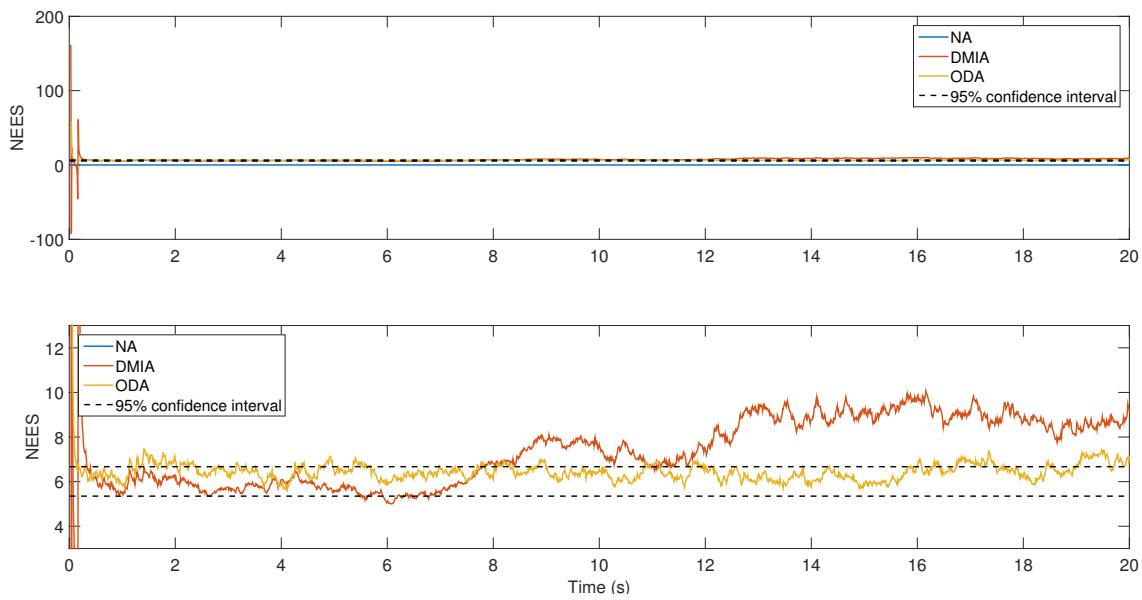


Fig. 5.3. NEES from 100 Monte Carlo runs from the first simulated overtaking scenario. ODA is consistent over the whole simulation. DMIA is consistent during the first half of the simulation and then diverge and yield unacceptably high values. The reason for the high values of DMIA is most probably a result of a bias in the approximated covariance matrices. NEES of NA is not consistent at all and stays more or less constant during the simulation. As a consequence, NA is not an optimal filter. A reason for the inconsistency of NA could be the time-invariant structure of the covariance matrices used in the algorithm.

5.1.2 Scenario 2

In the second overtaking scenario the target vehicle performs an overtaking maneuver starting in the left lane and additionally make a change from the left lane to the middle lane during the overtaking. The target vehicle must make some longitudinal and lateral acceleration in order to perform the lane changing maneuver, which can be seen in Fig. 4.3, and as a consequence the dynamic assumption of constant acceleration is violated. The error for the position and velocity for the second overtaking scenario is illustrated in Fig. 5.4. The lane changing maneuver is clearly visible in as all error terms have some large deviations due to the sudden accelerations during the overtaking. ODA has the lowest deviation in position error and respond to the changes in acceleration quite well. DMIA respond almost as well as ODA in the beginning of the simulation and then during the second half performs not as well compared to ODA. DMIA shows a noisy behaviour during the bigger part of the simulation. NA has large deviations during the first half of the simulation and then during the second half responds better and performs almost as well as ODA. For the velocity error all algorithms shows similar tendencies in deviation due to the accelerations of the lane change maneuver. ODA is the algorithm that has the lowest deviations followed by NA and DMIA. NA performs similar to ODA with slightly bigger deviations. DMIA has bigger deviations in velocity error than both ODA and NA and shows signs of delay. During the second half of the simulation DMIA converges towards the values of ODA and NA.

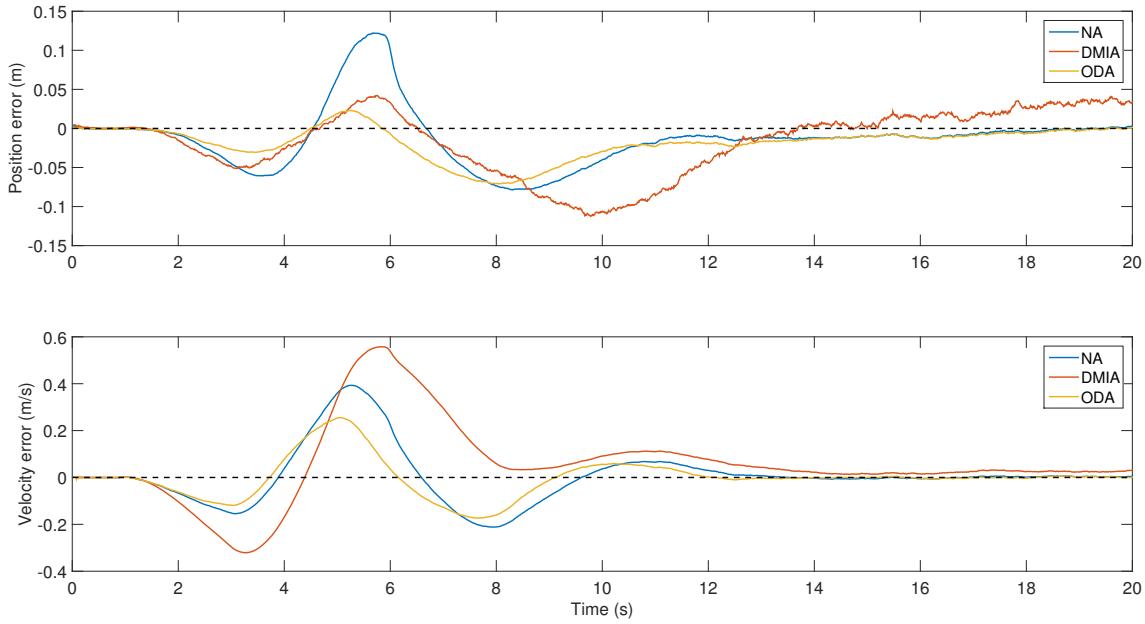


Fig. 5.4. Error for position and velocity for all algorithms during the second overtaking scenario from 100 Monte Carlo runs. The additional accelerations due to the lane changing maneuver is visible as fluctuations in the error terms. ODA has the smallest error for both position and velocity followed by NA and DMIA. DMIA shows inertia in the position error in that it respond both slower and different than ODA and NA to the maneuvers. In the velocity error DMIA seems to have a delay compared to ODA and NA.

The RMSE's for position and velocity for the second overtakning scenario are illustrated in Fig. 5.5. It is clearly visible in both position RMSE and velocity RMSE that the assumptions on the model dynaimcs are violated during the lane chaning maneuver. ODA performs well and converges fast back to the same RMSE's as in scenario 1 after the lane changing maneuver. NA respond with higher deviations to the maneuver and converges towards the same values as ODA after the lane changing maneuver. DMIA shows inertial tendencies for both RMSE's and does not converge in position RMSE after the lane changing maneuver. DMIA decreases in velocity RMSE after the lange changing maneuver but at a much slower rate than ODA and NA. The average position RMSE of ODA, DMIA and NA is 0.0415 m, 0.1237 m and 0.0469 m respectively. The average velocity RMSE of ODA, DMIA and NA is 0.0797 m/s, 0.2648 m/s and 0.0972 m/s respecitely. The RMSE's of ODA and NA is approximately twice as big in the second scenario compared to the first scenario. DMIA has about the same position RMSE in both scenarios while the velocity RMSE is about thrice as big in scenario 2 compared to scenario 1. It is clear from the average RMSE's that ODA performs best during the simulation followed by NA and DMIA. The averages RMSE's from the sencond overtakning scenario is presented in Table. 5.2.

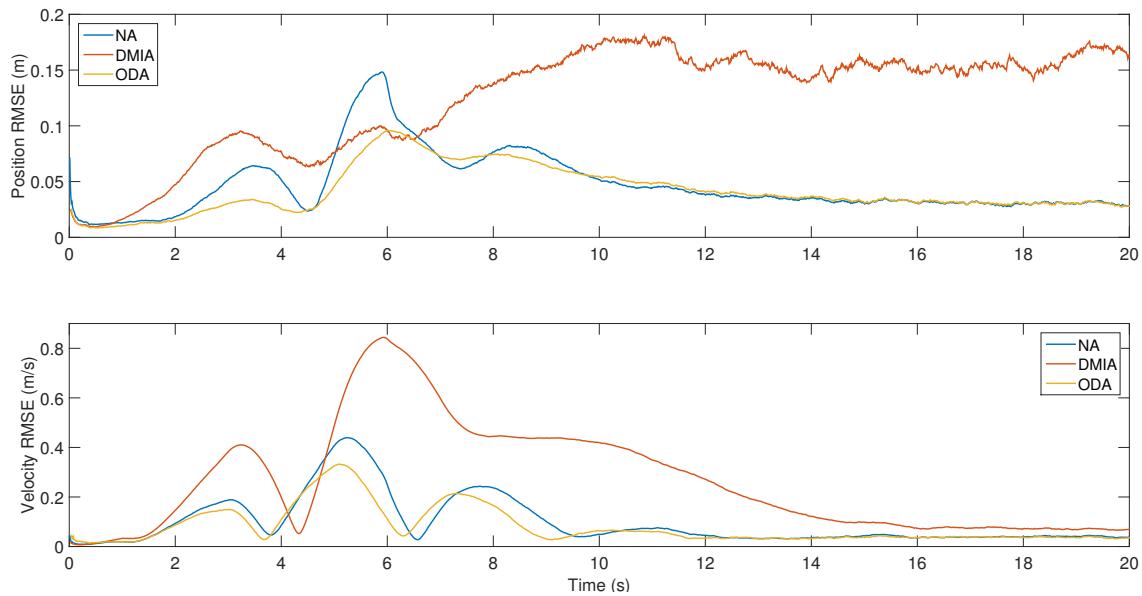


Fig. 5.5. RMSE for position and velocity from 100 Monte Carlo runs from the second overtaking scenario. The lane changing maneuver is visible in the RMSE's of the algorithms as the wave-like deviations. ODA has the lowest average RMSE's followed by NA and DMIA. DMIA have diverging tendencies after the lane changing maneuver while ODA and NA converges towards the RMSE values of scenario 1. In the upper figure, a bias can be assumed to affect DMIA since it does not converge towards the values of ODA and NA. Additionally, DMIA shows sign of inertia in the lower figure depicting velocity RMSE since it converges at a much slower rate than ODA and NA.

Table 5.2. Average RMSE for position and velocity in the second overtaking scenario. ODA have the lowest RMSE for both position and velocity. NA performs almost as well as ODA. DMIA has the highest RMSE for both position and velocity. Compared to the results of the first overtaking scenario, ODA and NA approximately doubles their RMSE's. The position RMSE of DMIA is almost the same in both scenarios while the velocity RMSE is thrice as big.

Algorithm	Position (m)	Velocity (m/s)
ODA	0.0415	0.0797
DMIA	0.1237	0.2648
NA	0.0469	0.0972

The NEES's of the algorithms for the second overtaking scenario are illustrated in Fig. 5.6. ODA and DMIA are inconsistent during the bigger part of the simulation due to the sudden occurring accelerations during the overtaking. Both algorithms are consistent in the beginning before the lane changing maneuver initiates. ODA and DMIA shows similar behaviour in inconsistency and converges towards the confidence region when the lane changing maneuver is finished. ODA becomes consistent during the last seconds of the simulation while DMIA does not enter the confidence region and yields too high values in order to be an optimal filter.

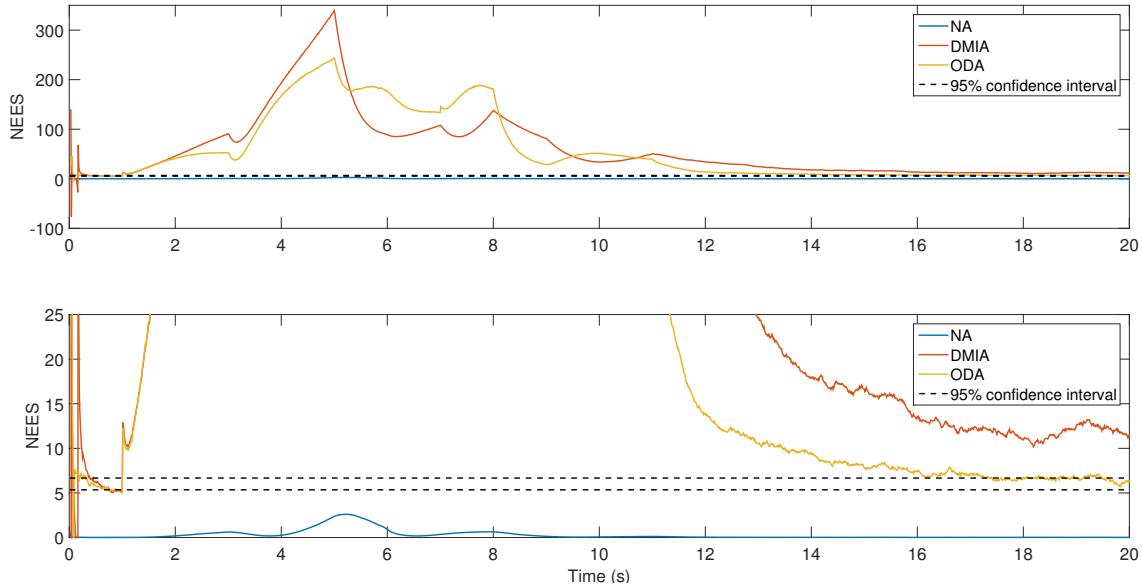


Fig. 5.6. NEES from 100 Monte Carlo runs from the second overtaking scenario. ODA and DMIA are consistent in the beginning of the simulation but as the target vehicle starts to accelerate in order to perform the change of lane the assumptions on the motion dynamics are violated which results in an inconsistent behaviour of ODA and DMIA. Towards the end of the simulation ODA becomes consistent while DMIA seems to have a bias and do not converge into the confidence region. NA is inconsistent during the whole simulation.

The bias in DMIA from scenario 1 is also present in the second scenario which results in a NEES of DMIA not entering the confidence region during the later part of the simulation. NA is inconsistent during the whole simulation and respond very little to the maneuvers performed during the overtaking. Since the NEES of NA is not within the 95 % confidence interval the algorithm cannot be said to be an optimal filter.

5.2 Robustness to sensor failures

In the simulated scenarios with sensor failures, the ODA performs better than DMIA and NA. The radar fails in approximately 5 % of the simulated time and the camera fails in approximately 10 % of the simulated time. NA is not able to handle cases when both sensors are failing simultaneously and gives thus incomplete results. DMIA shows signs of numerical effects such as singularities, giving high values in the position and velocity RMSE sense. The ODA shows robustness and is still consistent on a 95% level during the first scenario with sensor failures. During the second scenario with sensor failures no algorithm shows consistency.

5.2.1 Scenario 1

The same overtaking scenario as in Section 5.1.1 is performed with implemented sensor failures. The radar has a failure rate of 4.96 % and the camera has a failure rate of 10 % during the simulation. The position and velocity error for ODA, DMIA and NA are presented in Fig. 5.7. ODA is most robust against sensor failures and yield similar results compared to the results of ODA in Section 5.1.1. NA is sensitive to sensor failures and gives incomplete and much more noisy results compared to the results of NA in Section 5.1.1. What hard to see in Fig. 5.7 is that when the sensors fail at the same time, NA cannot compute any estimate. DMIA is extremely sensitive to sensor failures and has a lot of singularities occurring during the bigger part of the simulation. The errors of DMIA is hard to compare against the results of DMIA in Section 5.1.1 due to all the singularities.

The RMSE's from the simulation are illustrated in Fig. 5.8. ODA shows robustness and yield stable results similar to the results of ODA in Section 5.1.1. NA yields incomplete and noisy RMSE's due to its incapability to handle simultaneous sensor failures. DMIA yield a much higher RMSE for position and velocity compared to the RMSE's of DMIA in Section 5.1.1. It is visible that DMIA has a lot of singularities during the simulation which is depicted as outliers/spikes in the figure. If the singularities of DMIA are disregarded, a positive trend is visible both position RMSE and velocity RMSE. The trend is a lot bigger than the trends DMIA showed in Section 5.1.1. In a RMSE sense, ODA is the only algorithm that shows to be robust against sensor failures. The average position RMSE of ODA and DMIA is 0.0250 m and 0.2721 m respectively. The average velocity RMSE of ODA and DMIA is 0.0343 m/s and 3.7407 m/s respectively. Since NA yields incomplete results during the simulation average RMSE's cannot be computed. The average RMSE's of the algorithms and failure rates of the sensors presented is presented in Table. 5.3.

The NEES's from the simulation are illustrated in Fig. 5.9. ODA is the only algorithm that shows consistency and yields almost as good results as in Section 5.1.1. The results of ODA shows that the algorithm is not sensitive to sensor failures. DMIA is consistent during the first seconds of the simulation and after that it is inconsistent. During the bigger part of the simulation it is hard to distinguish the NEES of DMIA due to the many singularities occurring. The many singularities makes it also hard to detect if there is any bias present in the NEES. NA is inconsistent during the whole simulation and yield incomplete results when both sensors fail simultaneously. The NEES's of DMIA and NA shows that the filters are not optimal.

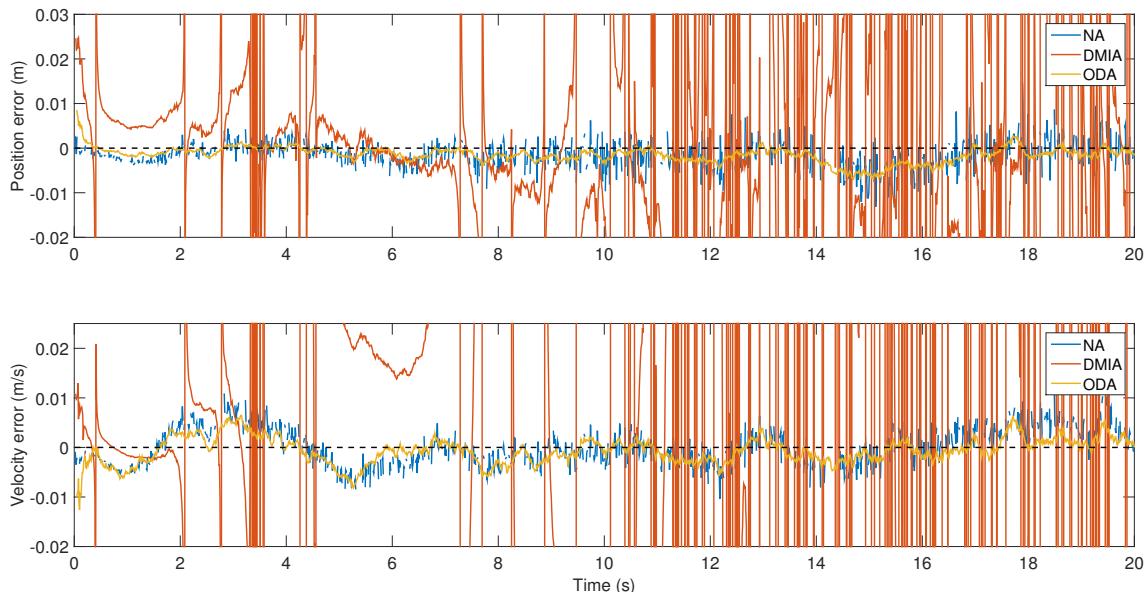


Fig. 5.7. Error for position and velocity for all algorithms during the first overtaking scenario from 100 Monte Carlo runs with sensor failures. ODA is robust to the sensor failures while DMIA and NA are not. NA cannot compute any estimate when both sensors fails simultaneously and thus yield incomplete results. DMIA has many singularities and is as a consequence hard to evaluate from a position and velocity error point of view. The structure of DMIA lets it compute estimates even though both sensors fail simultaneously. The radar has a failure rate of 4.96 % and the camera has a failure rate of 10 % during the simulation.

Table 5.3. Average RMSE for position and velocity in the first overtaking scenario with sensor failures including the average failure rate of the sensors. ODA has the lowest average RMSE for both position and velocity. The high velocity RMSE of DMIA is a result of the many singularities that occur during the simulation. NA cannot compute any estimate if both sensors fail simultaneously and hence no average RMSE's can be computed for NA.

Algorithm	Position (m)	Velocity (m/s)
ODA	0.0250	0.0343
DMIA	0.2721	3.7407
NA	-	-
Radar failure rate	4.96 %	
Camera failure rate	10.00 %	

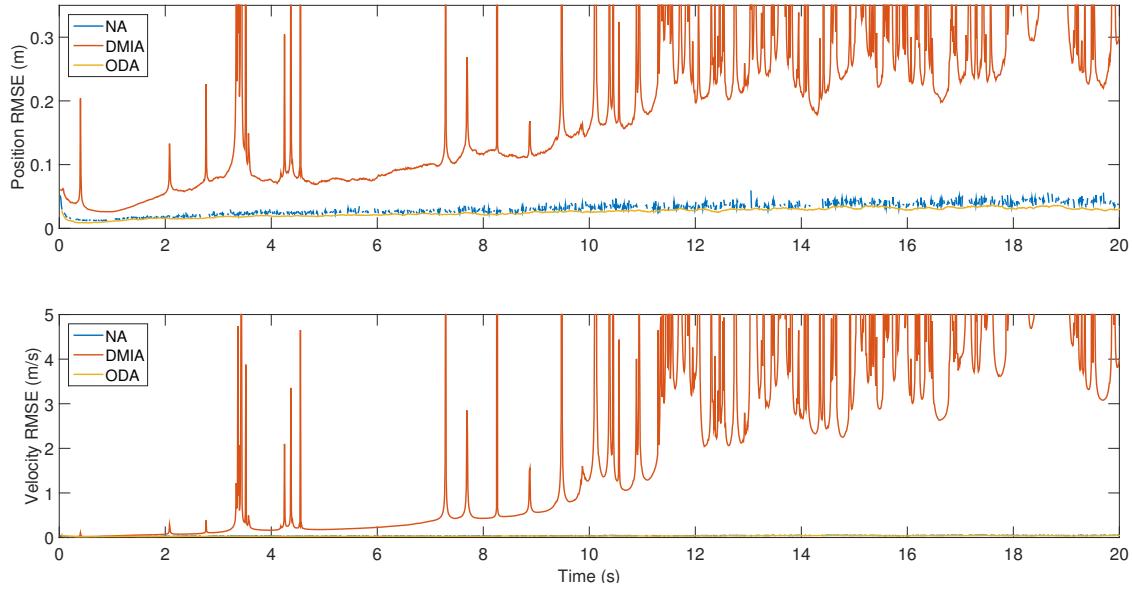


Fig. 5.8. RMSE for position and velocity from 100 Monte Carlo runs from the first simulated overtaking scenario with sensor failures. ODA performs similar to the results from the first simulation without sensor failures. DMIA shows a lot of singularities which is depicted as extremepoints/spikes in the figure. NA gives noisy results and cannot compute any estimate when both sensors fails at the same time. The radar has a failure rate of 4.96 % and the camera has a failure rate of 10 % during the simulation.

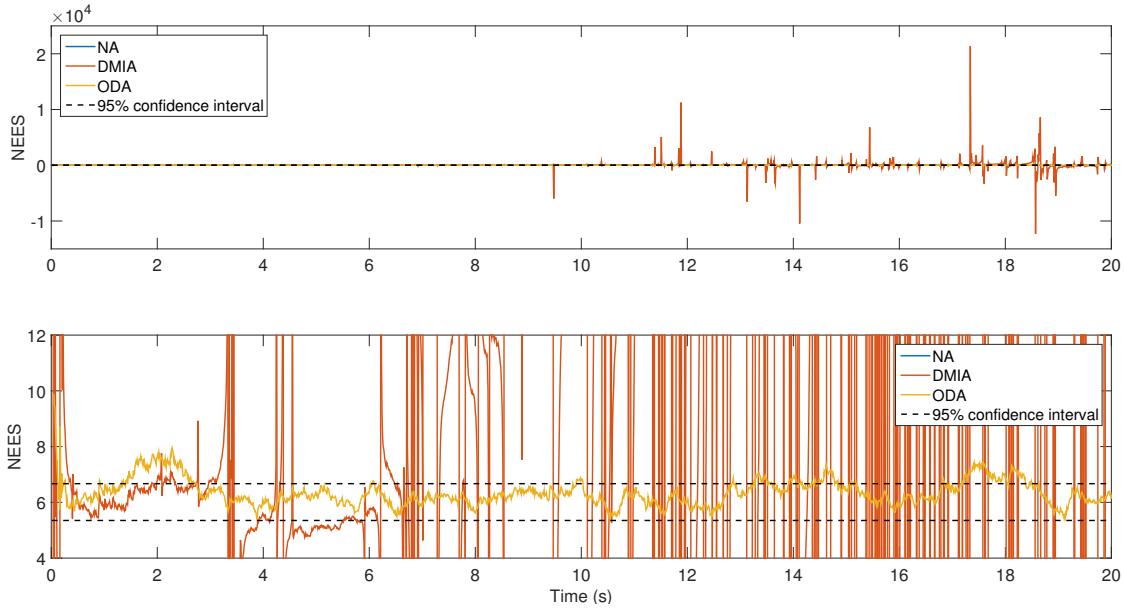


Fig. 5.9. NEES from 100 Monte Carlo runs from the first simulated overtaking scenario with sensor failures. ODA is consistent during the major part of the simulation and shows robustness to sensor failures. DMIA is consistent only in the beginning of the simulation. NA is inconsistent during the whole simulation. Both DMIA and NA yields results that unacceptable for an optimal filter. The radar has a failure rate of 4.96 % and the camera has a failure rate of 10 % during the simulation.

5.2.2 Scenario 2

The same overtaking scenario as in Section 5.1.2 is performed with implemented sensor failures. The radar has a failure rate of 4.93 % and the camera has a failure rate of 10.05 % during the simulation. The position and velocity error for ODA, DMIA and NA are presented in Fig. 5.10. ODA is most robust against sensor failures and yield similar results compared to the results of ODA in Section 5.1.2. In Fig. 5.10 is it visible when both sensors fail simultaneously in that NA yield incomplete results. If the incompleteness of the results of NA is disregarded, the algorithm show a similar behaviour compared to the results of NA in Section 5.1.2. DMIA is extremely sensitive to sensor failures and has a lot of singularities and show a similar behaviour as the results of DMIA in Section 5.2.1. The errors of DMIA is hard to compare against the results of DMIA in Section 5.1.2 due to all the singularities.

The position and velocity RMSE for the second scenario with sensor failures are illustrated in Fig. 5.11. ODA yield similar results compared to the results of ODA in Section 5.1.2 which shows the robustness of the algorithm. NA yields incomplete and noisy RMSE's due to the algorithms incapability of handling simultaneous sensor failures. The incomplete results of NA is visible in Fig. 5.11. In the incomplete results and the noisy behaviour of NA is disregarded the results from the simulation are similar to the results of NA in Section 5.1.2. The many singularities of DMIA is visible in Fig. 5.11 as outliers or spikes. If the singularities are disregarded the results of position RMSE of DMIA have a similar behaviour the the results of DMIA in Section 5.1.2. If the singularities and disregarded for the velocity RMSE of DMIA an positive trend can be identified that is not there for the results of DMIA in Section 5.1.2. ODA is the only algorithm that yield good results. The average RMSE for position and velocity is 0.0433 m and 0.0835 m/s for ODA and 0.3668 m and 7.2266 m/s for DMIA. Due to the incomplete results from NA average RMSE's cannot be computed. The average RMSE's of the algorithms and failure rates of the sensors presented is presented in Table. 5.4.

The NEES's from the simulation are illustrated in Fig. 5.12. ODA and DMIA are consistent during the first second of the simulation and then none of the algorithms are consistent during the rest of the simulation. As a consequence, none of the filters prove to be optimal during the simulated overtaking. ODA converges towards the 95 % confidence region during the end of the simulation but does not enter the region. The lane changing maneuver starts 1 second into the simulation and finishes 11 seconds into the simulation, which explain the inconsistency during this time. DMIA has a lot of singularities which makes it hard to distinguish if there is any bias affecting the algorithm. NA is also inconsistent during the whole simulation and yield incomplete results.

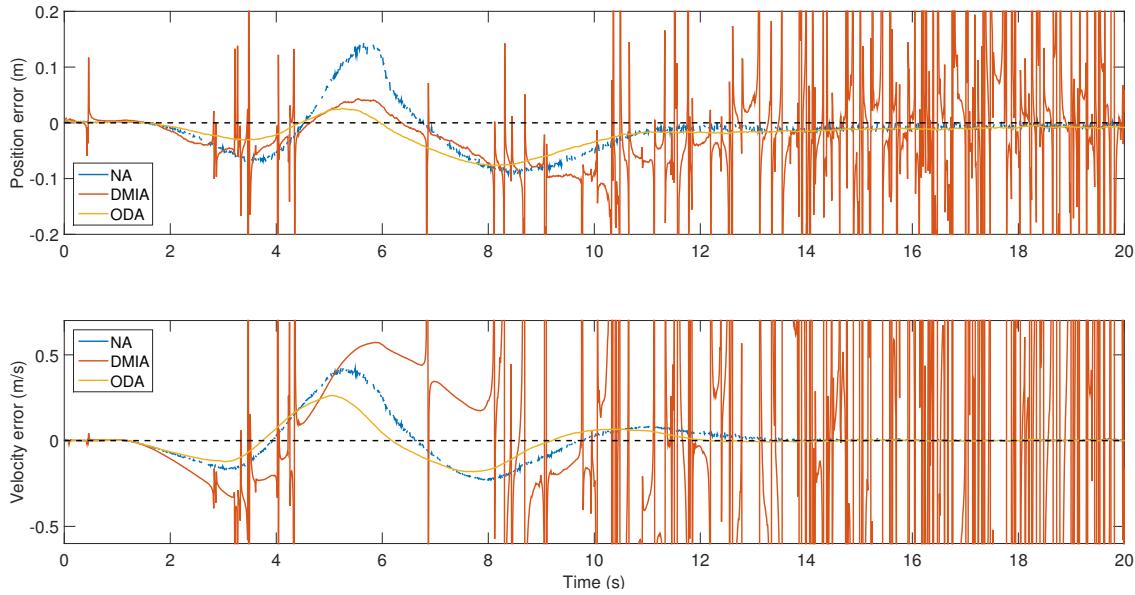


Fig. 5.10. Error for position and velocity for all algorithms during the second overtaking scenario from 100 Monte Carlo runs with sensor failures. ODA shows robustness to sensor failures while DMIA and NA does not. NA cannot handle simultaneous sensor failures and yield incomplete results while DMIA has a lot of singularities. The radar has a failure rate of 4.93 % and the camera has a failure rate of 10.05 % during the simulation.

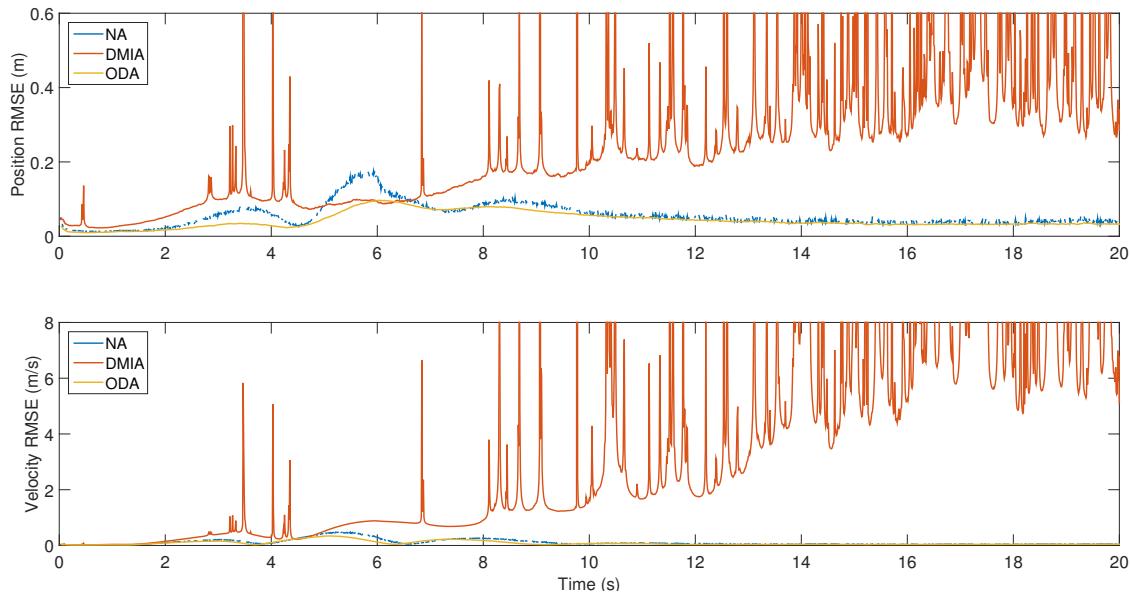


Fig. 5.11. RMSE for position and velocity from 100 Monte Carlo runs from the second simulated overtaking scenario with sensor failures. The RMSE's of ODA shows that the algorithm is robust to sensor failures. NA and DMIA is not robust to sensor failures and NA yield incomplete results which are seen in the figure. DMIA has a lot of singularities and have a positive increasing trend for both position and velocity RMSE. The radar has a failure rate of 4.93 % and the camera has a failure rate of 10.05 % during the simulation.

Table 5.4. Average RMSE for position and velocity in the second overtaking scenario with sensor failures including the average failure rate of the sensors. ODA has the lowest average RMSE's. DMIA has high average RMSE's as a result of the many singularities due to the sensor failures. NA yields incomplete results since the algorithm cannot compute estimates when both sensor fail simultaneously. As a consequence, average RMSE's for NA cannot be computed.

Algorithm	Position (m)	Velocity (m/s)
ODA	0.0433	0.0835
DMIA	0.3668	7.2266
NA	-	-
Radar failure rate	4.93 %	
Camera failure rate	10.05 %	

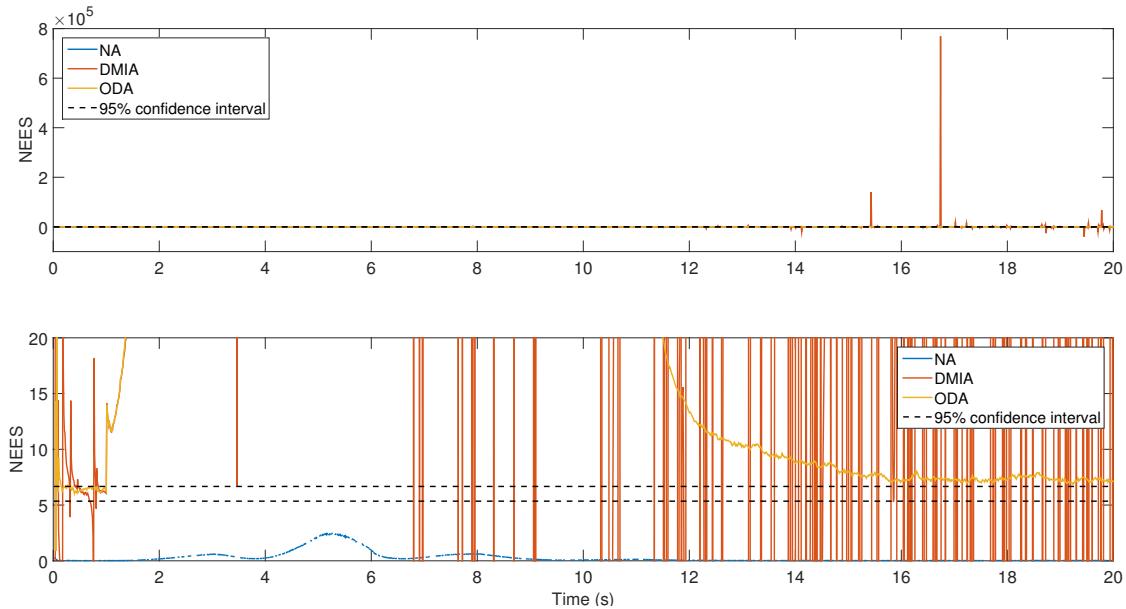


Fig. 5.12. NEES from 100 Monte Carlo runs from the second simulated overtaking scenario with sensor failures. The lower of the two figures are a zoom in on the 95 % confidence region. ODA and DMIA are consistent during the first second of the simulation. The during the rest of the simulation none of the algorithms proves to be consistent. The lane changing maneuver takes place during seconds 1-11 of the 20 seconds long simulation. ODA converges towards the confidence region after the lane changing maneuver but never enters the region. DMIA has a lot of singularities which makes it hard to distinguish any trends. The singularities are a result of the sensor failures and shows that DMIA is not robust to sensor failures. The many singularities of DMIA are probably a result of the approximated estimation error covariance matrices used in the algorithm. NA is inconsistent during the whole simulation and yields incomplete results which are depicted in the figure. The radar has a failure rate of 4.93 % and the camera has a failure rate of 10.05 % during the simulation.

5.3 Test vehicle evaluation

The novel IMF algorithm DMIA and the weighted least-squares algorithm NA are simulated on real logged sensor data on a test vehicle. The sensor configuration used is seen in Fig. 4.2 in Section 4.1 which is the same sensor configuration that is used in the other simulated scenarios. During the simulation four target vehicles perform overtakes similar to the overtakes of scenario 1 and scenario 2. The overtakes are performed consecutively and no sensor failures are present during the simulation.

The first target vehicle enters the sensors field-of-view 7 seconds into the simulation with a relative velocity of 9 m/s in longitudinal direction and 0 m/s in lateral direction. The target vehicle performs a lane changing maneuver during the overtaking. The target vehicle drives with a constant velocity during the whole overtaking.

The second target vehicle enters the sensors field-of-view 10 seconds into the simulation with a relative velocity of 10 m/s in longitudinal direction and 0 m/s in lateral direction. The target vehicle drives in a straight line in the left lane during the whole overtaking with a constant velocity.

The third target vehicle enters the sensors field-of-view 21 seconds into the simulation with a relative velocity of 7.5 m/s in longitudinal direction and -0.5 m/s in lateral direction. The target vehicle does a similar overtaking as the first target vehicle and performs a lane changing maneuver. The target vehicle drives with a constant velocity during the whole overtaking.

The fourth target vehicle enters the sensors field-of-view 25 seconds into the simulation with a relative velocity of 10.1 m/s in longitudinal direction and -0.8 m/s in lateral direction. The target vehicle drives in a straight line in the left lane during the overtaking with a constant velocity.

DMIA and NA performs in such a way that the fused estimates from the algorithms are hard to separate and yields similar results. The fused estimate from both algorithms follows the radar measurement more than the camera measurement which can be seen in Fig. 5.13. This is probably a result from the radar having lower variance in longitudinal direction compared to the camera variance and hence the fusion algorithms follow the radar measurements more than the camera measurements.

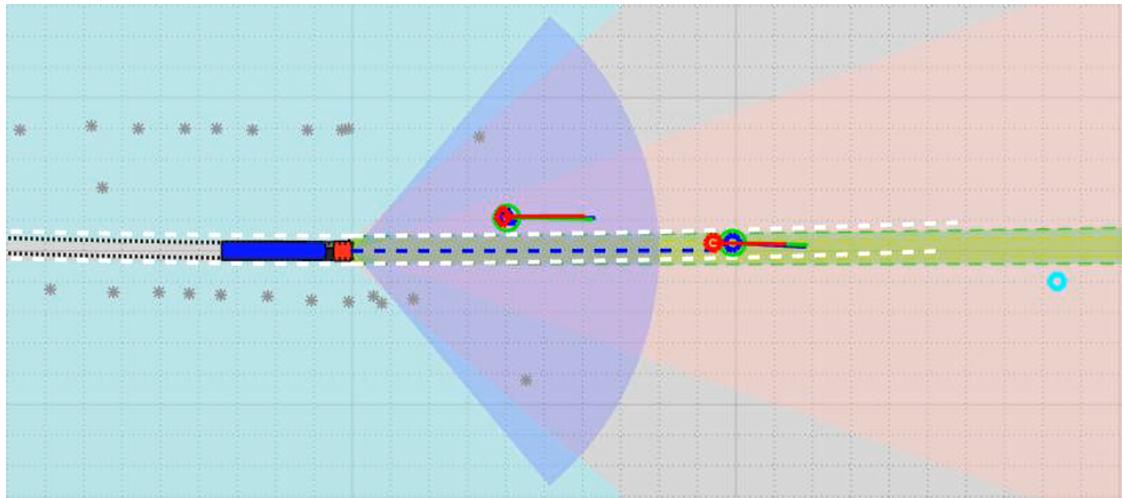


Fig. 5.13. Snapshot from the simulation on real logged data where the first two targets are tracked by vehicular camera and radar. The camera measurements are depicted as red rings and the radar measurements are depicted as blue rings. Additionally, the measurements from radar and camera also depicts a relative velocity vector which can be seen as a red and blue line in front of their respective sensor measurement. The position of the fused estimate from DMIA is seen as a light green ring and the relative velocity vector as a light green line in front of the ring. The fused estimate from NA yield the same results as DMIA and therefore is only DMIA depicted in the figure. The fused estimate from DMIA is closer to the radar measurement compared to the camera measurement due to the lower longitudinal variance of the radar compared to the longitudinal variance of the camera.

Discussion

The results shows that ODA performs better than DMIA and NA in both average RMSE's and consistency which confirms the findings from the literature. Furthermore, the results shows that ODA is robust to sensor failures while both DMIA and NA are more sensitive and not able to handle sensor failures.

The novel IMF algorithm DMIA performs well during the first scenario without sensor failures from both a RMSE and NEES point of view. Even though the algorithm has the largest RMSE's compared to ODA and NA, DMIA are consistent during the first half of the scenario. During the second half of the first scenario DMIA is inconsistent as a result of some bias affecting the algorithm. The inconsistent behavior of DMIA is present in all simulations and especially during the simulations with sensor failures. The inconsistency and many singularities present during the scenarios with sensor failures could be a result of the numerical approximation of the estimation error covariance matrices used in the algorithm. The approximated covariance matrices in DMIA tends to always have one eigenvalue being close to zero which results in the covariance matrix being singular. The singular matrix is then handled by the optimization algorithm described in Section 4.1.1 and modified to be non-singular. Thus, the estimation error covariance matrix is first linearized and then passed through the optimization algorithm before it is used in the IMF. Another possible error that could affect the DMIA estimate is the choice of difference approximations for the derivatives in the Jacobian matrices used in the approximation of the estimation error covariance matrices. For first-order derivatives the implicit Euler approximation was used because of its numerical stability. For second order derivatives a central difference approximation was used which has a second order of accuracy.

The weighted least-squares algorithm NA performs almost as well as ODA in the simulations without sensor failures from an average RMSE point of view. The algorithm has small position and velocity errors and is simple to understand and implement. However, a drawback of the algorithm is the inability of handling simultaneous sensor failures due to the none existing influence of a predicted state estimate when computing the updated state estimate. Another drawback of NA is the inconsistency of the estimates, i.e. the estimates of NA does not converge in

probability to the true value of the state. NA is inconsistent in all simulations, with and without sensor failures, which would indicate that the algorithm is not optimal as a filter. The inconsistency of NA could be a result of the assumption that the radar measures position and velocity of an object the same way that the camera is measuring an object.

ODA performs well during all simulations and the robustness of the algorithm is visible especially during the simulations with sensor failures. ODA is built on the same architecture as DMIA but receives the true estimation error covariance matrices of the local sensor filters. ODA is not sensitive to sensor failures since the updated state estimates are based on the MFF's predicted estimates and the predicted and updated estimates of the sensors local filters. The MFF continues to predict and update estimates even though the sensors fail to send new information to the MFF. The inaccuracy of the MFF is proportional to the time the filter does not receive new information, i.e. if the sensors fail to send new information to the MFF for a long time the global estimate will be less accurate. However, ODA is built on the fact that the processes in the local sensor filters are known and available to the MFF. But as mentioned in Chapter 2, the sensors used for ADAS systems are developed and produced by automotive suppliers and not by the end user, in this case Scania CV AB. In reality this means that the technology in the sensors are not available to the end user because it is core business competence of the suppliers which they are not willing to share.

Filter optimality is the art of tuning the filter in such a way that the estimates of the filter are consistent. The difference in consistency of ODA and DMIA between scenario 1 and scenario 2, without sensor failures, is a result of how the two filters are tuned. A filter can be tuned by increasing, or decreasing, the process noise and the measurement noise used in the filter, see Appendix A and Appendix B. This means that the results of ODA and DMIA for scenario 2 could be improved if the two filters are tuned in such a way that they anticipate a more *volatile* behavior of the target vehicle. ODA and DMIA are tuned in such a way that the algorithms perform best when the target vehicle does not perform any lane changing maneuvers, i.e. the algorithms anticipate that the target vehicle stays in the same lane during the overtaking and thus the process noise is assigned to be small. By increasing the value of the process noise used in the algorithms the filters could respond better to the lane changing maneuver. As a consequence the results from the first overtaking scenario would deteriorate if the process noise is increased due to the anticipation that the target vehicle will not perform any lane changing maneuvers.

The results show that the ODA is precise, consistent and robust to sensor failures and is thus the most suitable algorithm for sensor fusion of preprocessed sensor data. A decentralized IMF algorithm for future ADAS systems will be advantageous when more sensors are added due to the modularity of the algorithm. Even though DMIA did not perform better than NA in the average RMSE sense, it showed more consistency. Other aspects of DMIA compared to NA is that DMIA is computationally heavier but as a consequence also more robust to sensor failures in that

the algorithm does not yield incomplete results. In literature, authors tend to value consistency higher than minimal error performance. Hence, this must be considered when choosing which sensor fusion algorithm to use for the next generation of ADAS systems.

Conclusion

A new approach to IMF has been presented with the DMIA where only the updated state estimates are available to the MFF. The decentralized IMF approach ODA has been shown to perform similar as would a centralized IMF architecture, which is considered to be the most accurate method when fusing sensor data.

Conclusions drawn from the discussion about the novel IMF approach DMIA are:

- There is some bias affecting the DMIA estimate in such a way that the filter is not optimal. If the bias is identified the algorithm could be modified against the bias and yield results that would be in line for an optimal filter.
- The approximated error covariance matrices used in DMIA may have propagated errors that substantially differ from the true error covariance matrices. The linearization and optimization algorithm may individually yield errors that are accumulated into the error covariance matrices used in the end in DMIA.
- The choice of implicit difference approximations seems to be good since the problems tends to be stiff when looking at the eigenvalues of the error covariance matrices, i.e. the size of the eigenvalues range from very small to big.

In order for DMIA to be suitable for the next generation of ADAS systems at Scania CV AB the following needs to be done; investigate and remove the bias, improve the numerical stability of the approximated error covariance matrices used in the algorithm.

Conclusions drawn from the discussion about the weighted least-squares approach NA are:

- Disregarding the inconsistency of the algorithm, NA is computationally simpler than both ODA and DMIA with good performance in the average RMSE sense

- The algorithm is vulnerable to sensor failures, i.e. not robust, and the inconsistency proves that the filter is not optimal. Therefore the algorithm is not favorable for an ADAS system.
- From the sensor properties explained in Section 3.1.1 the assumption that the radar measures position and velocity the same way as the camera is clearly false.

Conclusions drawn from the discussion about the optimal IMF approach ODA are:

- The robustness of the algorithm is profound during all simulations and the filter clearly yields optimal results for the first overtaking scenario.
- The available information from the local sensor filters may be limited, by the suppliers, in such a way that ODA cannot be used even though it is the best alternative of the three algorithms.

To conclude from the comparison of NA and DMIA, NA is better from a minimal error performance point of view and DMIA is better from a consistency point of view. Therefore, this must be considered when choosing which sensor fusion algorithm to use. The next generation of ADAS systems for Scania CV AB should strive to use a decentralized IMF algorithm for sensor fusion where the first option should be ODA, the second option DMIA and the third option NA.

Future work & extensions

The results of the IMF approaches ODA and DMIA from the two simulated scenarios differ because the filters are not tuned in a way that is optimal for both scenarios. In order to improve the results, the dynamic motion model of the target needs to be improved. The model could be improved if vehicle-to-vehicle communication is established between the target vehicle and the host vehicle. The target vehicle could then communicate with the host vehicle about what maneuvers are going to be performed which could be used as input in the dynamic equation.

Another natural extension of this thesis is to investigate the performance of the IMF approach to other environments. There are many heavy-duty vehicles distributing goods and managing waste in urban areas and these vehicles are exposed to other types of objects than a vehicle driving on a highway. In an urban environment there are cars, busses, bicycles and pedestrians to take into consideration. Even if vehicle-to-vehicle communication is established between the target vehicles and the host vehicle in order to improve the motion dynamics of the target vehicle, the problem of modeling pedestrians and bicycles in an urban environment still remains. In order to extend the IMF approach to urban environments, assumptions on the motion of bicycles and pedestrians need to be made. Additionally, ADAS systems need to be able to determine if a target is a car, buss, bicycle or pedestrian before applying a IMF approach for target tracking.

A future improvement that would add further robustness to the decentralized IMF approaches is to establish feedback from the MFF to the local sensor filters. By doing so the measurement noise of the two sensors could be reduced.

Bibliography

- [1] S. Matzka and R. Altendorfer, “A comparison of track-to-track fusion algorithms for automotive sensor fusion,” in *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*, pp. 189–194, Aug 2008.
- [2] S. Nilsson and A. Klekamp, “A comparison of architectures for track fusion,” in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 517–522, Sept 2015.
- [3] M. Aeberhard, S. Schlichtharle, N. Kaempchen, and T. Bertram, “Track-to-track fusion with asynchronous sensors using information matrix fusion for surround environment perception,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, pp. 1717–1726, Dec 2012.
- [4] D.-J. Lee, *Unscented Information Filtering for Distributed Estimation and Multiple Sensor Fusion*. Guidance, Navigation, and Control and Co-located Conferences, American Institute of Aeronautics and Astronautics, Aug 2008. 0.
- [5] R. M. Aarts, R. Irwan, and A. J. E. M. Janssen, “Efficient tracking of the cross-correlation coefficient,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, pp. 391–402, Sep 2002.
- [6] M. E. Liggins, C.-Y. Chong, I. Kadar, M. G. Alford, V. Vannicola, and S. Thomopoulos, “Distributed fusion architectures and algorithms for target tracking,” *Proceedings of the IEEE*, vol. 85, pp. 95–107, Jan 1997.
- [7] F. Gustafsson, *Statistical sensor fusion*. Lund : Studentlitteratur, 2010.
- [8] X. Shen, Y. Luo, Y. Zhu, and E. Song, “Globally optimal distributed kalman filtering fusion,” *Science China Information Sciences*, vol. 55, no. 3, pp. 512–529, 2012.
- [9] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Introduction*, pp. 1–88. John Wiley & Sons, Inc., 2002.
- [10] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Adaptive Estimation and Maneuvering Targets*, pp. 421–490. John Wiley & Sons, Inc., 2002.

- [11] E. Song, J. Xu, and Y. Zhu, “Optimal distributed kalman filtering fusion with singular covariances of filtering errors and measurement noises,” *IEEE Transactions on Automatic Control*, vol. 59, pp. 1271–1282, May 2014.
- [12] K. C. Chang, R. K. Saha, and Y. Bar-Shalom, “On optimal track-to-track fusion,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, pp. 1271–1276, Oct 1997.
- [13] T. Kailath, *Linear estimation*. Upper Saddle River, NJ : Prentice Hall, 2000.
- [14] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Computational Aspects of Estimation*, pp. 301–318. John Wiley & Sons, Inc., 2002.
- [15] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Linear Estimation in Static Systems*, pp. 121–177. John Wiley & Sons, Inc., 2002.
- [16] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *State Estimation in Discrete-Time Linear Dynamic Systems*, pp. 199–266. John Wiley & Sons, Inc., 2002.

Appendix - The Kalman filter

The Kalman filter is a method for linear estimation in dynamic systems. The Kalman filter minimizes the estimation error covariance and because of its recursive nature it has proven to be very useful. The Kalman filter has found a wide range of applications, e.g navigation and target tracking, and has been extensively used in both centralized and decentralized architectures for processing multiple sensor data.

Consider the general description of a time-discrete dynamic linear system

$$x(k+1) = F(k)x(k) + G(k)u(k) + \Gamma(k)v(k), \quad (\text{A.1a})$$

$$z(k) = H(k)x(k) + w(k), \quad (\text{A.1b})$$

where $v(k)$ and $w(k)$ are zero-mean white Gaussian noises with the properties

$$\mathbb{E} \left[\begin{pmatrix} v(k) \\ w(k) \end{pmatrix} \begin{pmatrix} v(k)^\top & w(k)^\top \end{pmatrix} \right] = \begin{pmatrix} Q(k) & 0 \\ 0 & R(k) \end{pmatrix}, \quad \forall k. \quad (\text{A.2})$$

Let

$$\hat{x}(j|k) \triangleq \mathbb{E}[x(j)|Z^k], \quad (\text{A.3})$$

denote the conditional mean where

$$Z^k \triangleq \{z(i), i \leq k\},$$

is the sequence of observations available at time k [16].

The state space description enables the filter to be either a filter, smoother or predictor

$$\hat{x}(j|k) = \begin{cases} \text{estimate (filter) of the state if } j = k, \\ \text{smoothed value of the state if } j < k, \\ \text{predicted value of the state if } j > k. \end{cases}$$

The estimation error is defined as

$$\tilde{x}(j|k) \triangleq x(j) - \hat{x}(j|k), \quad (\text{A.4})$$

and its corresponding covariance

$$P(j|k) \triangleq \mathbb{E}[\tilde{x}(j|k)\tilde{x}(j|k)^\top | Z^k] = \mathbb{E}[(x(j) - \hat{x}(j|k))(x(j) - \hat{x}(j|k))^\top | Z^k], \quad (\text{A.5})$$

sometimes referred to as the conditional covariance matrix of $x(j)$ given Z^k or the covariance associated with the estimate (A.3). The covariance matrix of the error (A.4) is equal to the covariance of the state since the estimate is the conditional mean (A.3) [16].

The Estimation Algorithm

The algorithm can be divided into a prediction part and an update part [16].

Prediction:

$$\hat{x}(k+1|k) = F(k)\hat{x}(k|k) + G(k)u(k), \quad (\text{A.6})$$

$$P(k+1|k) = F(k)P(k|k)F(k)^\top + \Gamma(k)Q(k)\Gamma(k)^\top, \quad (\text{A.7})$$

$$\hat{z}(k+1|k) = H(k+1)\hat{x}(k+1|k). \quad (\text{A.8})$$

Update:

$$\nu(k+1) = z(k+1) - \hat{z}(k+1|k), \quad (\text{A.9})$$

$$S(k+1) = R(k+1) + H(k+1)P(k+1|k)H(k+1)^\top, \quad (\text{A.10})$$

$$W(k+1) = P(k+1|k)H(k+1)^\top S(k+1)^{-1}, \quad (\text{A.11})$$

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1)\nu(k+1), \quad (\text{A.12})$$

$$P(k+1|k+1) = P(k+1|k) - W(k+1)S(k+1)W(k+1)^\top. \quad (\text{A.13})$$

In Fig. A.1 one cycle of the Kalman filter is illustrated as a flow chart. The figure shows the prediction part and the update part of the state estimate and the state covariance and how they are related to one-another.

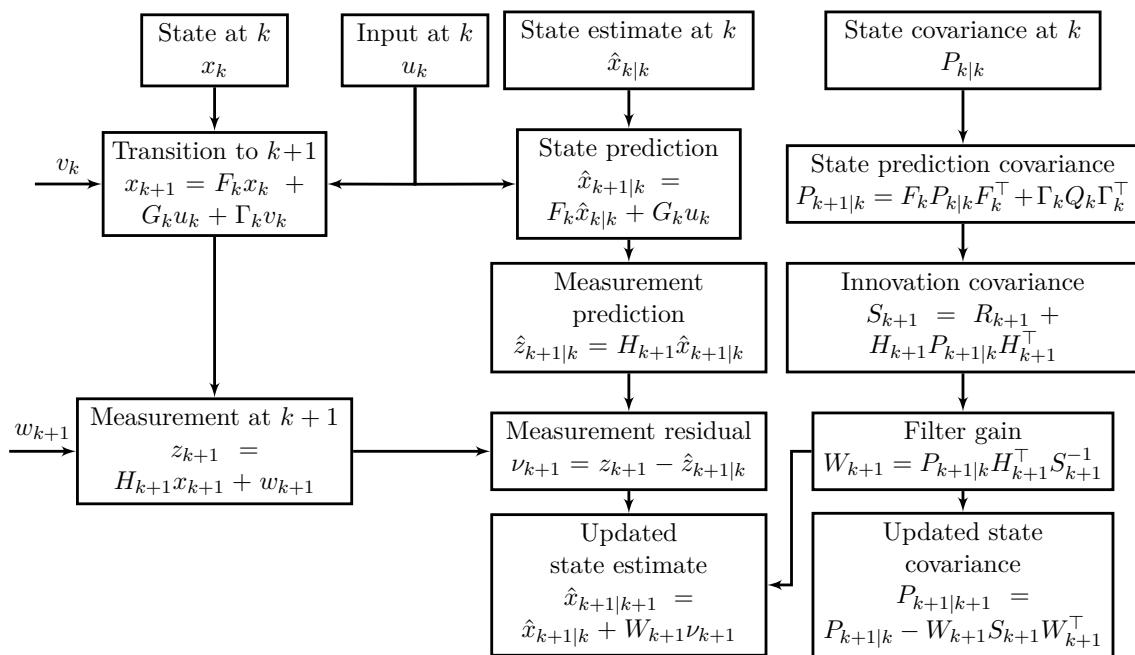


Fig. A.1. One cycle of the Kalman filter. In the filter three columns corresponding to the state, the state estimate and the state covariance is visible. In the columns of the state estimate and the state covariance the filter goes through the prediction part and the update part of the filter. It can be seen how the filter gain, often referred to as the Kalman gain, influences the updated state estimate. It is also depicted how the measurement is added to the state estimate column and influences the updated state estimate.

Appendix - The information filter

The information filter is built on the knowledge of the Kalman filter but uses the inverse of the state covariance matrix as the information of the different states. This filter is usually a better alternative when multiple sensors are used. The information filter can be divided into a prediction part and an update part [14].

Consider the general description of a time-discrete dynamic linear system

$$\begin{aligned} x(k+1) &= F(k)x(k) + G(k)u(k) + \Gamma(k)v(k), \\ z(k) &= H(k)x(k) + w(k), \end{aligned} \quad (\text{B.1})$$

where $v(k)$ and $w(k)$ are zero-mean white Gaussian noises with the properties

$$\mathbb{E} \left[\begin{pmatrix} v(k) \\ w(k) \end{pmatrix} \begin{pmatrix} v(k)^\top & w(k)^\top \end{pmatrix} \right] = \begin{pmatrix} Q(k) & 0 \\ 0 & R(k) \end{pmatrix}, \quad \forall k. \quad (\text{B.2})$$

The information filter is obtained by inverting the estimation error covariance to obtain the information matrix (B.3) and multiplying it with the state estimate to obtain its corresponding information filter state (B.4) [14]

$$Y(i|j) = P(i|j)^{-1}, \quad (\text{B.3})$$

$$\hat{y}(i|j) = Y(i|j)\hat{x}(i|j). \quad (\text{B.4})$$

The Estimation Algorithm

Prediction:

$$A(k) = (F(k)^{-1})^\top Y(k|k) F(k)^{-1}, \quad (\text{B.5})$$

$$Y(k+1|k) = A(k) - A(k)\Gamma(k) \left(\Gamma(k)^\top A(k)\Gamma(k) + Q(k)^{-1} \right)^{-1} \Gamma(k)^\top A(k), \quad (\text{B.6})$$

$$\begin{aligned} \hat{y}(k+1|k) &= (F(k)^{-1})^\top \left(I - Y(k|k)F(k)^{-1}\Gamma(k) \left(\Gamma(k)^\top A(k)\Gamma(k) + \right. \right. \\ &\quad \left. \left. + Q(k)^{-1} \right)^{-1} \Gamma(k)^\top (F(k)^{-1})^\top \right) \hat{y}(k|k). \end{aligned} \quad (\text{B.7})$$

Update:

$$\mathcal{I}(k+1) = H(k+1)^\top R(k+1)^{-1} H(k+1), \quad (\text{B.8})$$

$$\mathbf{i}(k+1) = H(k+1)^\top R(k+1)^{-1} z(k+1), \quad (\text{B.9})$$

$$Y(k+1|k+1) = Y(k+1|k) + \mathcal{I}(k+1), \quad (\text{B.10})$$

$$\hat{y}(k+1|k+1) = \hat{y}(k+1|k) + \mathbf{i}(k+1). \quad (\text{B.11})$$

The information filter is advantageous compared to the Kalman filter in that it does not need any initial guess in order to start. The filter is initialized simply by putting $Y_{k|k} = 0$ and similarly $\hat{y}_{k|k} = 0$. When $Y_{i|j}$ is invertible, the state estimate can simply be calculated as $\hat{x}_{i|j} = Y_{i|j}^{-1} \hat{y}_{i|j}$.

In Fig. B.1 one cycle of the information filter is illustrated as a flow chart. The figure shows how the update of the state information matrix and information filter is built on the new information entering the system.

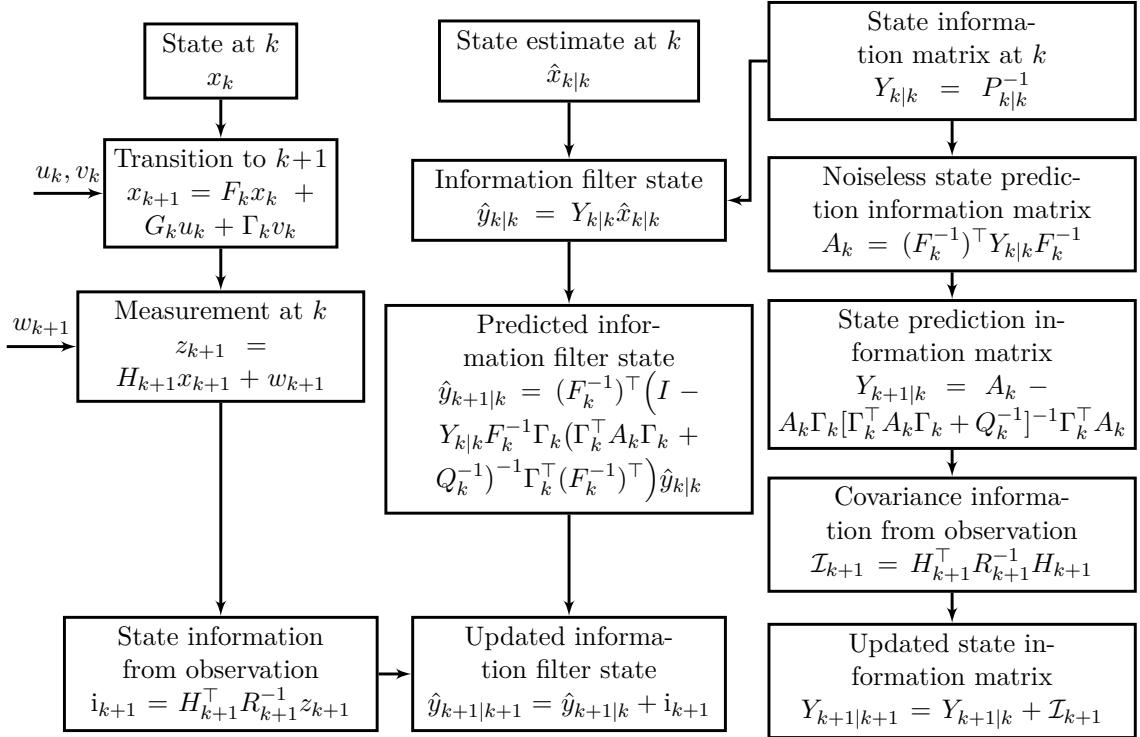


Fig. B.1. One cycle of the information filter. In the filter three columns corresponding to the state, the information filter state and the information matrix is visible. In the columns of the information filter state and the information matrix goes through the prediction part and the update part of the filter. It is depicted how the information matrix is used to compute the information filter state by adding the new information. At the end of the cycle the new information from the measurement is used to compute the updated information filter state.

TRITA -MAT-E 2016:53
ISRN -KTH/MAT/E--16/53--SE