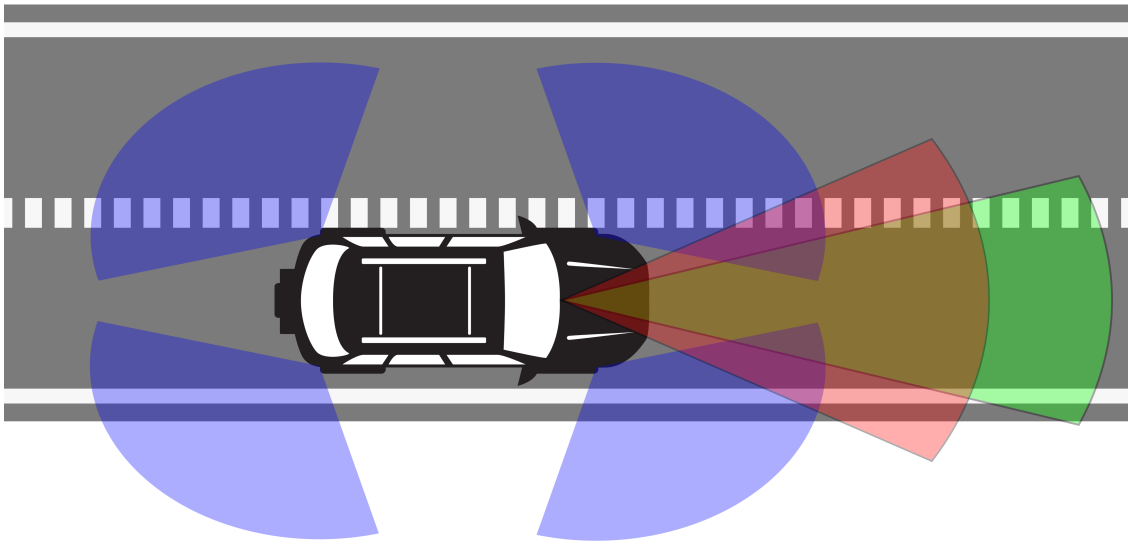
CHALMERS
UNIVERSITY OF TECHNOLOGY



Sensor fusion of radar and stereo-vision for tracking moving vehicles as extended objects

Master's thesis in Systems, Control and Mechatronics

Ludvig Ekström & Jonathan Risberg

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

Master's thesis 2018:EX037

Sensor fusion of radar and stereo-vision for tracking moving vehicles as extended objects

Ludvig Ekström & Jonathan Risberg

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2018

Sensor fusion of radar and stereo-vision for tracking moving vehicles as extended objects
Ludvig Ekström & Jonathan Risberg

Examiner: Lars Hammarstrand, Department of Electrical Engineering

Master's Thesis 2018:EX037
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Visualisation of a vehicle and its six sensors covering the surroundings. The vehicle and the road is from Vecteezy.com.

Typeset in LaTeX
Gothenburg, Sweden 2018

Sensor fusion of radar and stereo-vision for tracking moving vehicles as extended objects
Ludvig Ekström & Jonathan Risberg
Department of Electrical Engineering
Chalmers University of Technology

Abstract

This thesis studies the problem of using radar and stereo-camera sensors on a car to track other vehicles in its proximity. The system is viewed as a distributed tracking system where independent tracks are generated from radar detections and stereo-camera respectively. This thesis presents solutions on how the information from the individual tracks can be combined using track to track fusion.

All results are based on simulations of both the vehicles and the sensor data. For this we have developed a simulation tool which models vehicle trajectories, the tracks from the camera and the radar detections. The radar detections are used in a radar tracker to generate tracks. It uses a method based on the global nearest neighbour to handle the measurement association problem, where the measurements then are used in an extended Kalman filter. The filter uses a modified bicycle model and a discrete number of reflector points on a rectangular vehicle to model the target vehicles.

To combine the information of the radar and the stereo-camera tracks the LMMSE fuser method is used. This method was selected since does not require the same state space representations for the different tracks. The method is tested in two configurations, with and without feedback to the radar tracker. Both configurations improve the extended target tracking in most cases. However, the radar tracker uses a measurement association algorithm that causes the tracks to be overconfident in its state estimation. This leads to a unstable behaviour and in some cases this causes the radar track to diverge. When the radar track diverges it leads to large errors for the fusion since the uncertainty of the radar tracker does not reflect the error. By using feedback to the radar tracker the information from the camera prevents the radar track to diverge. Thus reducing the implications from overconfidence. Overall the track to track fusion with feedback shows the best performance and is most reliable of the two configurations and the individual trackers.

Keywords: Track-to-track fusion, sensor fusion, radar, stereo-camera, extended target tracking, LMMSE

Acknowledgements

We want to thank our supervisor Lars Hammarstrand for guiding us through the thesis. We also want to thank Karl Granström for answering our questions on stuff we did not understand.

Ludvig Ekström & Jonathan Risberg, Gothenburg, June 2018

Contents

1

Introduction

In recent years transportation has begun to be revolutionised by autonomous vehicles. This could affect traffic safety, the environment and how transportation is used. In order for driverless vehicles to succeed, a number of tasks that normally are the responsibility of a human driver needs to be solved through engineering. One such task is perception of the environment and more specifically the ability to keep track of other cars on the road and predict where they will be in the future. This ability is important to avoid collisions, especially during complex manoeuvres like overtaking other vehicles or lane changing.

Perception of the surroundings comes from the use of external sensors of which there are many in use today. For tracking applications the main focus lies on the use of radar, LiDAR and camera sensors. These three have different strengths and weaknesses which makes the combination of them important for technologies like autonomous driving.

Tracking in the context of vehicles is about estimating an object's position and its motion. This is usually done by combining measurements from one or many sensors together with models of the object's dynamics. An object which is tracked in this way is called a target. Target tracking can be divided into different subproblems depending on how the target appears in the sensors, due to both sensor properties and distance to the target. If the target appear smaller than the resolution of the sensors, then the spatial attributes of the target are insignificant. This is called a point target tracking problem, where the target will at most generate one measurement per sensor. If instead the target appear larger than the resolution of the sensors, such that one can get measurements from different parts of the target, then it is an extended target tracking problem. This enables the estimation of the spatial attributes which can be used to better predict the state of the target.

When you want to track multiple targets, there is an added difficulty due to the fact that which target is observed is unknown. This problem of multiple target tracking has been researched for over 50 years [1], [2] and has resulted in a multitude of different methods and applications [3], [4].

1.1 Problem formulation and limitations

In this thesis the aim is to implement an algorithm for tracking a single simulated vehicle using data from a simulated stereo-vision system and Doppler radar sensors. These two different sensor systems are used to create two different tracks, where a track is a state estimation of a target over time.

The stereo-vision system is assumed to use prefiltered data such that the output from the system are complete tracks of the targets. A way of simulating the behaviour such a system would have is needed to be able to draw conclusions regarding real world performance.

The radar sensors generates measurements which are used to create the radar tracks for the targets. This can be divided into two sub problems, measurement to track association and filtering. Measurement to track association is the problem of relating measurements to the track of a target. Measurements of a target which is tracked are called detections and should be used to estimate the state of that target. When sensors have a sufficient resolution, multiple detections can be made on a target. This complicates the problem since the number of detections are unknown and because detections from different parts of a target can have different attributes. This gives rise to the extended target tracking problem, where the extent of a target is an important attribute. Measurements can also be clutter, which are unwanted detections of other objects, false detections or echoes. These measurements need to be distinguished such that they do not affect the next steps of the tracking. A data association algorithm is needed to determine which measurements are detections of the target and also to determine which part of the target the detections originate from. These measurements are then used to perform the tracking, for which a filter is needed that creates and updates the tracks.

Finally with the two different tracks, from the radar tracker and the stereo-camera tracking system, a combined estimation of the target can be made. For this an algorithm for track to track fusion is needed. Since there is only one target, the tracks are assumed to be of the same target.

1.2 Proposed solution

We propose a solution based on heterogeneous track to track fusion between individual tracks generated from a radar tracker and a stereo-camera tracking system. Heterogeneous track to track fusion is used to allow different state space representations for the two trackers.

The radar tracker uses an extended Kalman filter to track and predict a targets kinematics and extent. The filter uses a motion model which is based on movements along curves and a measurement model which assumes that detections comes from a

set of points on a rectangular target. A data association algorithm is used to remove clutter and determine from which points the detections originate.

The stereo-camera tracking system gives complete simulated tracks. The performance of the tracks depends on how much of the target is detectable by the sensor and how long it has tracked the object. Larger targets are more accurately tracked and the accuracy of the tracks increases for targets tracked over time.

The two individual tracks are combined using the LMMSE fuser which is a method for heterogeneous track to track fusion. Two configurations are tested using this method for track to track fusion, one uses the trackers in parallel and outputs the fusion between them and the other one uses feedback into the radar tracker.

1.3 Related works and contributions

Target tracking is a subject with many different approaches [3, 4] . Target tracking in distributed systems where all measurements can't be processed in a centralised way, e.g. because of communication constraints, track to track fusion can be applied [5]. In some cases the same state representation isn't available in every part of the system, this means the sensors are heterogeneous and the fusion methods needs to be designed around it. Heterogeneous track to track fusion has been studied in [6] where two such methods are tested and compared, namely the LMMSE fuser and the ML fuser.

This thesis studies the application of previously studied methods, to the case where a decentralised tracking system composed of radar and stereo-camera is used for vehicle tracking. Different track to track fusion approaches are tested against each other to determine the effects it has on the system.

1.4 Thesis outline

In this chapter an introduction to the thesis and its contributions in relation to other related works. The problem was also introduced with the proposed solution given in this thesis. In Chapter 2 the theory of the used methods is introduced. In Chapter 3 the methods used are described with information about the simulation and test scenarios used. In Chapter 4 the results of the thesis and a discussion of these are given. Finally in Chapter 5 and 6, our conclusions are presented with suggested future work.

2

Theory

In this chapter the theory behind the methods used in this thesis are presented. In Section 2.1 theory of using the information in measurements to estimate a targets state is presented. In Section 2.2 an introduction to the target tracking problem is given. In Section 2.3 the theory of determining which measurements to use in cases relevant to the subject of the thesis is presented. Since the subject of the thesis is a distributed sensor system, Section 2.4 will present the theory of fusing individual tracks.

2.1 Filtering

Filtering is the process of removing unwanted features of data to extract information which can be used. There are many kinds of filters, both analogue and digital. Examples of analogue filters are both active and passive electrical component, however these are not of interest for this thesis. Instead digital filters are used, more precisely Kalman filters. In the following sections a introduction to the theory of Bayesian filtering and Kalman filters will be given. More information about these subjects can be found in [7] and [8]. The following sections largely follows the notation in [7].

2.1.1 Bayesian filtering

Bayesian filtering is the application of Bayesian statistics to estimate an unknown probability density function using prior knowledge and observations of the system.

The notation used to describe the filtering system is:

$$\begin{aligned} \text{Discrete time index - } & k \in \mathbb{N} \\ \text{State - } & \mathbf{x}_k \in \mathbb{R}^n \\ \text{Measurement - } & \mathbf{y}_k \in \mathbb{R}^m \\ \text{Motion model - } & p(\mathbf{x}_k|\mathbf{x}_{k-1}) \\ \text{Measurement model - } & p(\mathbf{y}_k|\mathbf{x}_k) \end{aligned}$$

If the system can be described by a Markovian process, as seen in Figure 2.1, the following properties hold

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1}) &= p(\mathbf{x}_k|\mathbf{x}_{k-1}) \\ p(\mathbf{x}_{k-1}|\mathbf{x}_{k:T}, \mathbf{y}_{k:T}) &= p(\mathbf{x}_{k-1}|\mathbf{x}_k) \\ p(\mathbf{y}_k|\mathbf{x}_{1:k}, \mathbf{y}_{1:k-1}) &= p(\mathbf{y}_k|\mathbf{x}_k) \end{aligned}$$

where $T > k$. The goal of the filtering process is to calculate

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \tag{2.1}$$

To accomplish this we first calculate the predictive distribution, given the motion model, by the Chapman Kolmogorov equation

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})\mathrm{d}\mathbf{x}_{k-1} \tag{2.2}$$

and then using Bayes' rule we get the update

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})/Z_k, \tag{2.3}$$

where

$$Z_k = \int p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})d\mathbf{x}_k \tag{2.4}$$

is a normalisation.

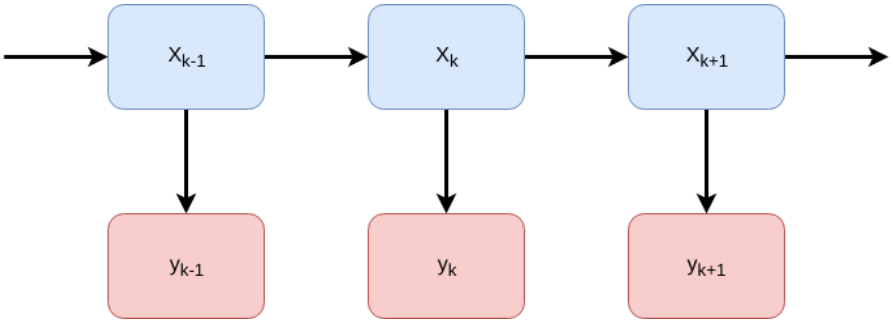


Figure 2.1: Illustration of the Markov property of the state and the measurements.

2.1.2 Kalman filter

The Kalman filter is a closed form solution to the Bayesian filtering problem in equations (2.1), (2.2), (2.3) and (2.4). It assumes linear Gaussian motion and measurement models and the prior distribution

$$\mathbf{x}_0 \sim N(\hat{\mathbf{x}}_0, \mathbf{P}_0), \tag{2.5}$$

where $\hat{\mathbf{x}}_0$ and $\mathbf{P}_0$ are the initial state and covariance. With the previous assumptions we get the following

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{q}_{k-1} \tag{2.6}$$
$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{r}_k \tag{2.7}$$

where $\mathbf{A}_{k-1}$ is a transition matrix of the motion model, $\mathbf{H}_k$ is the measurement model matrix and

$$\mathbf{q}_{k-1} \sim N(0, \mathbf{Q}_{k-1})$$
$$\mathbf{r}_k \sim N(0, \mathbf{R}_k)$$

where $\mathbf{Q}_{k-1}$ is the process noise covariance and $\mathbf{R}_k$ is the measurement noise covariance.

Each iteration of the filter is performed in two steps, prediction and update. In the first step the Gaussian predicted distribution

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = N(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}),$$

is calculated using the motion model in the equations

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} \tag{2.8}$$
$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1}. \tag{2.9}$$

In the second step measurements and the measurement model is used to calculate

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = N(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}),$$

from the equations

$$\nu_k = \mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1} \tag{2.10}$$
$$\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k \tag{2.11}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{S}_k^{-1} \tag{2.12}$$
$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\nu_k \tag{2.13}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T \tag{2.14}$$

2.1.3 Nonlinear Kalman filter

If the motion and measurement models are nonlinear, the Kalman filter can't be used. There are however extensions of the Kalman filter to nonlinear applications. One such extension is the extended Kalman filter which is described in this section. There are other filters for nonlinear filtering problems which could also be used, more information can be found in [7].

The idea of the extended Kalman filter is to assume a Gaussian approximation of the probability density. The filter linearise about the current estimation using Taylor series expansions. The filtering process is similar to the Kalman filter as each iteration involves a prediction and a update step. The prediction is calculated as

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{\mathbf{k-1|k-1}}) \tag{2.15}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_{k-1} \tag{2.16}$$

$$\tag{2.17}$$

and the update is calculated as

$$\nu_k = \mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{\mathbf{k|k-1}}) \tag{2.18}$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \tag{2.19}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \tag{2.20}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \nu_k \tag{2.21}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T \tag{2.22}$$

where

$$\mathbf{F}_k = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}_k} \right|_{\mathbf{x}_k = \hat{\mathbf{x}}_{k-1|k-1}} \tag{2.23}$$

and

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}_k} \right|_{\mathbf{x}_k = \hat{\mathbf{x}}_{k|k-1}} \tag{2.24}$$

are the Jacobians of the nonlinear motion and measurement model functions respectively.

2.2 Target tracking

Target tracking is the ability to find and estimate a moving object's state over time using sensor measurements. The target tracking problem contains many sub-problems. Measurements have errors due to noise, disturbances and bias which needs to be filtered. In a lot of applications it is not clear whether a measurement comes from a target or whether it is clutter, which for example in the case of radar can represent unwanted echoes. To solve this a method is needed to associate the correct data to the correct target.

A generalisation of target tracking problem is the extended target tracking problem. This is when the size and shape of the target is such that we can get multiple detections from the same target. This can be solved in a number of ways by either clustering measurements into a single measurement, modelling the targets as extended targets with shape and size and modelling where the measurements can come from, or allowing multiple measurements from the same point source through expectation-maximisation. An overview to the subject can be found in [9].

2.3 Data association

The problem of associating data to some object of origin is a problem with many methods and applications. One such application is that of radar detections in a cluttered environment. The radar detections can come from any target or be clutter. A data association algorithm is thus needed to associate detections to the correct targets and reject measurements which are clutter.

One simple data association algorithm is called the nearest neighbour method [4]. It uses a measure of distance between all targets and all detections to assign each measurement to its closest target. This results in that multiple detections can be assigned to a single target. If this is an unwanted result, the global nearest neighbour (GNN) method [4] can be used instead. The GNN method minimizes the total distance of all made associations while only allowing one associated detection per target. This is done by calculating a measure of distance between all targets and detections and constructing a cost matrix C from these. Where $C(i, j)$ is the cost for assigning the j:th detection to the i:th target. Then one element from each column is chosen as a assignment. One way to choose these are to use the greedy approach of first choosing the lowest distance and removing that target from being selected again then continue selecting the lowest remaining association until all measurements has been assigned. It is, however, easy to show that this yields results which are not optimal. However, there are other methods one can use to select the optimal assignment, one of them is Munkres algorithm [10] which is used in this thesis.

An extension of the GNN is to use multiple hypotheses in parallel, meaning that more than one assignment hypothesis is kept. This is useful since the best assignment hypothesis given by the GNN might contain incorrect assignments due to clutter or errors. Thus by using multiple assignment hypotheses the risk of discarding the correct assignment decreases. This extension of the GNN is further discussed in [4] and is used in this thesis.

2.3.1 Measure of distance

One of the simplest ways of measuring how valid a association of a detection is, is to use the euclidean distance. Meaning that the closer a detection and a target is

in an euclidean way the more probable it is that the association is correct. The 2D-euclidean distance d_e is calculated as

$$d_e^2(p,q) = (q_x - p_x)^2 + (q_y - p_y)^2 \tag{2.25}$$

where p and q are two points with x- and y-coordinates. When the measure is used for association one points is the estimated position and the other one is the measured position. The distance d_e is then the cost used in the costmatrix C for associating the measurement to the estimated position.

Another measure of distance is the Mahalanobis distance. It takes into account the difference between all the attributes of a detection. In the case of Doppler radar the detections give range to detection from sensor, angle between sensor and detection and the radial velocity relative the sensor. It also takes into account the uncertainty of the state and measurement noise of the detection in the form of the innovation covariance as calculated in the Kalman filter, equation 2.19. The Mahalanobis distance d_m is calculated as

$$d_m^2 = (\mathbf{y} - \mathbf{h}(\hat{\mathbf{x}}))^T \mathbf{S}^{-1} (\mathbf{y} - \mathbf{h}(\hat{\mathbf{x}})) \tag{2.26}$$

where $\mathbf{h}(\hat{\mathbf{x}})$ is the expected measurement measurement from the estimated position $\hat{\mathbf{x}}$, $\mathbf{y}$ is the measurement, $\mathbf{S}^{-1}$ is the innovation covariance and d_m is the cost used in the costmatrix C for associating y to $\hat{x}$.

2.3.2 Gating

Gating is a method which can both decrease the number of calculations needed in the measurement association and to reduce the effects of clutter. It works by considering assignment distances which exceed some threshold to be forbidden. When using euclidean distances, the threshold is simply a distance in meters, while when using Mahalanobis distance the threshold is given by the cumulative chi-squared distribution given the degrees of freedom of the measurement and the probability of detection. This is because d_m^2 is chi-squared distributed with the degrees of freedom of the measurement.

2.4 Track to track fusion

In this thesis the tracks that are to be fused together have different state representations and therefore a heterogeneous track to track fusion method must be used. A method to do this is the LMMSE fuser which is studied in [6] and is based on the linear minimum mean square error estimation as talked about in [11]. This method will be presented in this section as presented in [6].

2.4.1 LMMSE fuser

We have two different tracks with different state representations $\mathbf{x}^i$ and $\mathbf{x}^j$. Between these we have the nonlinear relationship

$$\mathbf{x}^j \triangleq \mathbf{g}(\mathbf{x}^i). \tag{2.27}$$

It is assumed that they are synchronised and the time index is omitted. These two tracks have the estimates $\hat{\mathbf{x}}^i$ and $\hat{\mathbf{x}}^j$ with the corresponding covariance matrices

$$\mathbf{P}^i = E[(\mathbf{x}^i - \hat{\mathbf{x}}^i)(\mathbf{x}^i - \hat{\mathbf{x}}^i)^T] \tag{2.28}$$

and

$$\mathbf{P}^j = E[(\mathbf{x}^j - \hat{\mathbf{x}}^j)(\mathbf{x}^j - \hat{\mathbf{x}}^j)^T]. \tag{2.29}$$

The fusion with the LMMSE fuser is based on LMMSE estimation where $\hat{\mathbf{x}}^i$ is seen as the state prediction and $\hat{\mathbf{x}}^j$ is seen as the measurement. The fused state estimation is then calculated as

$$\hat{\mathbf{x}}^i_{LMMSE} = \hat{\mathbf{x}}^i + \mathbf{P}^X(\mathbf{P}^Z)^{-1}[\hat{\mathbf{x}}^j - \mathbf{g}(\hat{\mathbf{x}}^i)] \tag{2.30}$$

with the covariance matrix

$$\mathbf{P}^i_{LMMSE} = \mathbf{P}^i - \mathbf{P}^X(\mathbf{P}^Z)^{-1}\mathbf{P}^X, \tag{2.31}$$

where

$$\begin{aligned}
\mathbf{P}^X &\triangleq E[(\mathbf{x}^i - \hat{\mathbf{x}}^i)(\hat{\mathbf{x}}^j - \mathbf{g}(\hat{\mathbf{x}}^i))^T] && (2.32)\\
&\approx \mathbf{P}^i(\mathbf{G}^i)^T - \mathbf{P}^{ij} && (2.33)\\
\mathbf{P}^Z &\triangleq E[(\hat{\mathbf{x}}^j - \mathbf{g}(\hat{\mathbf{x}}^i))(\hat{\mathbf{x}}^j - \mathbf{g}(\hat{\mathbf{x}}^i))^T] && (2.34)\\
&\approx \mathbf{P}^j - \mathbf{G}^i\mathbf{P}^{ij} - \mathbf{P}^{ji}(\mathbf{G}^i)^T + \mathbf{G}^i\mathbf{P}^i(\mathbf{G}^i)^T && (2.35)
\end{aligned}$$

where

$$\mathbf{G}^i \triangleq [\nabla_{\mathbf{x}^i}\mathbf{g}(\mathbf{x}^i)^T]^T\Big|_{\mathbf{x}^i=\hat{\mathbf{x}}^i} \tag{2.36}$$

and

$$\mathbf{P}^{ij} \triangleq E[(\mathbf{x}^i - \hat{\mathbf{x}}^i)(\mathbf{x}^j - \hat{\mathbf{x}}^j)^T]. \tag{2.37}$$

There is no clear way of calculating the crosscorrelation $\mathbf{P}^{ij}$ in the heterogeneous problem. One solution to make the method work is to assume that the crosscorrelation $\mathbf{P}^{ij} = \mathbf{P}^{ji} = 0$. This simplifies the equations (2.30) and (2.31), using the approximations of equations (2.33) and (2.35) to

$$\hat{\mathbf{x}}^i_{LMMSE} = \hat{\mathbf{x}}^i + \mathbf{P}^i(\mathbf{G}^i)^T(\mathbf{P}^j + \mathbf{G}^i\mathbf{P}^i(\mathbf{G}^i)^T)^{-1}[\hat{\mathbf{x}}^j - \mathbf{g}(\hat{\mathbf{x}}^i)] \tag{2.38}$$

and

$$\mathbf{P}^i_{LMMSE} = \mathbf{P}^i - \mathbf{P}^i(\mathbf{G}^i)^T(\mathbf{P}^j + \mathbf{G}^i\mathbf{P}^i(\mathbf{G}^i)^T)^{-1}\mathbf{P}^i(\mathbf{G}^i)^T \tag{2.39}$$

If the assumption is incorrect, estimation errors will be introduced in the calculations.

2.4.2 Sample crosscorrelation

The crosscorrelation between tracks can be difficult to calculate, but using Monte Carlo simulations the crosscorrelation coefficient can be estimated [6].

The sample crosscorrelation coefficient between the l:th component in $\mathbf{x}^i$ and the h:th component in $\mathbf{x}^j$ can be calculated as

$$\hat{\rho}^M_{\mathbf{x}^i_l \mathbf{x}^j_h} \triangleq \frac{\sum_{m=1}^{M}(\hat{\mathbf{x}}^i_{l,m} - \mathbf{x}^i_l)(\hat{\mathbf{x}}^j_{h,m} - \mathbf{x}^i_h)}{\sqrt{[\sum_{m=1}^{M}(\hat{\mathbf{x}}^i_{l,m} - \mathbf{x}^i_l)^2][\sum_{m=1}^{M}(\hat{\mathbf{x}}^j_{h,m} - \mathbf{x}^j_h)^2]}}. \tag{2.40}$$

The coefficient takes values between -1 to 1, which means that the components are negatively correlated or correlated respectively. If the coefficient is 0 the components are uncorrelated.

3

Method and simulation

This chapter handles and describes the methods and simulation models used in the thesis.

3.1 Vehicle representation

The movement of the vehicles in the simulation is designed to mimic the movement behaviour of a real vehicle. The key features of real vehicle motion is that lateral movement is impossible and that turning without moving is also not possible. Therefore a modified bicycle model which uses curves to describe movement is used to simulate the motion of the vehicles. The selected model uses the state representation

$$\mathbf{x} = [x, y, v, a, \varphi, \lambda, w, l]^T$$

for each vehicle.

The states are x and y, the global position of the vehicle in global Cartesian coordinates; φ, the heading angle; v, the velocity in the heading direction; a, the acceleration; λ, which represents the curvature of the vehicle path; w and l, width and length of the vehicle, respectively. Using this model the next time state is described by

$$x_{k+1} = x_k + \lambda^{-1}(\cos(\frac{\pi}{2} + \varphi_k - v_k\lambda_k\Delta T) - \cos(\frac{\pi}{2} + \varphi_k)) \tag{3.1}$$

$$y_{k+1} = y_k + \lambda_k^{-1}(\sin(\frac{\pi}{2} + \varphi_k - v_k\lambda_k\Delta T) - \sin(\frac{\pi}{2} + \varphi_k)) \tag{3.2}$$

$$v_{k+1} = v_k + a_k\Delta T \tag{3.3}$$

$$a_{k+1} = a_k + q_k^a \tag{3.4}$$

$$\varphi_{k+1} = \varphi_k + v_k\lambda_k\Delta T \tag{3.5}$$

$$\lambda_{k+1} = \lambda_k + q_k^\lambda \tag{3.6}$$

$$w_{k+1} = w_k \tag{3.7}$$

$$l_{k+1} = l_k \tag{3.8}$$

where ΔT is the sampling time used in the system, $q_k^a \sim \mathcal{N}(0, \sigma_a^2)$ and $q_k^\lambda \sim \mathcal{N}(0, \sigma_\lambda^2)$ are zero-mean noise processes with standard deviation σ_a and σ_λ, respectively. This motion model is illustrated in Figure 3.1.

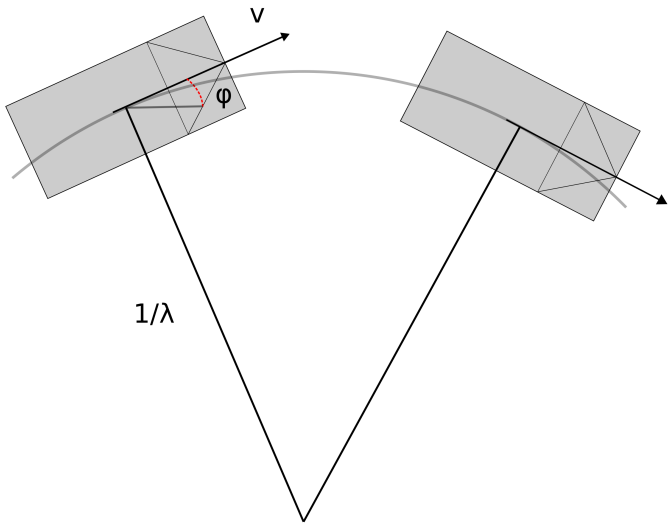


Figure 3.1: Illustration of the curve model used to describe the motions of a vehicle. The vehicle can be seen to follow a curved path based on the turning radius $1/\lambda$.

Since this is a nonlinear motion model a linearization is needed for the implementation of the EKF. This is done by calculating the Jacobian $\mathbf{F}_k$ of the motion model. However, for simplicity and due to the low sampling time, higher order time terms are ignored.

3.1.1 Sensors

The ego vehicle in this thesis is equipped with five Doppler radars, four short range radars are mounted on the corners facing outwards and one long range radar is facing forwards, and one stereo-vision camera that faces forward. The radars generates detections of targets within their fields of vision and the stereo-camera generates a track of the path taken by the vehicle while the target is in its field of vision. The position and orientation of the sensors and their fields of vision is schematically shown in figure 3.2.

3.2 Stereo-camera simulation

The stereo-camera system considered, generates vehicle tracks based on a vision based tracking algorithm. However, no real data is used in this thesis, instead the

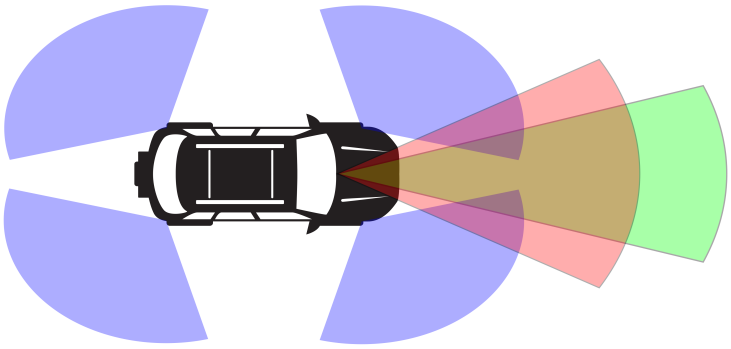


Figure 3.2: Schematics over the location of the sensors on the ego vehicle. The stereo-camera is shown in red, the radars are shown in blue and green.

vehicle tracks are simulated based on how a real system would perform.

The stereo-camera tracks gives the estimated position (x, y), heading (φ) and extent (w, l) of the detected targets. The accuracy of the tracks is designed to be similar to an actual stereo-camera tracking system. This means that there are multiple factors which affects the accuracy of the tracker. The ones accounted for in this thesis are:

- the distance between sensor and target
- the number of consecutive detections of the target
- the angular span the target occupies in the sensor
- the structural mismatch of features on the target

3.2.1 Camera measurements

Stereo-camera based measurements are done by matching the position of a feature (point) in the two images and then using it in combination with knowledge of the positions of the camera to calculate the position of the feature. If the matching is done incorrectly the estimated position of the feature will be wrong. This is what is shown in Figure 3.3, where the error in position is shown when a feature's position in the two cameras are off by one resolution cell. The resulting error is dependant on the distance between sensor and feature, and the error is greater in the radial direction than the tangential one. These types of feature mismatches can occur randomly due to temporary effects, however they can also systematically occur if two different features are incorrectly identified as the same in the camera system.

The random mismatches will be averaged out over time whilst the systematic error will not, since it is dependant on the physical attributes of the target.

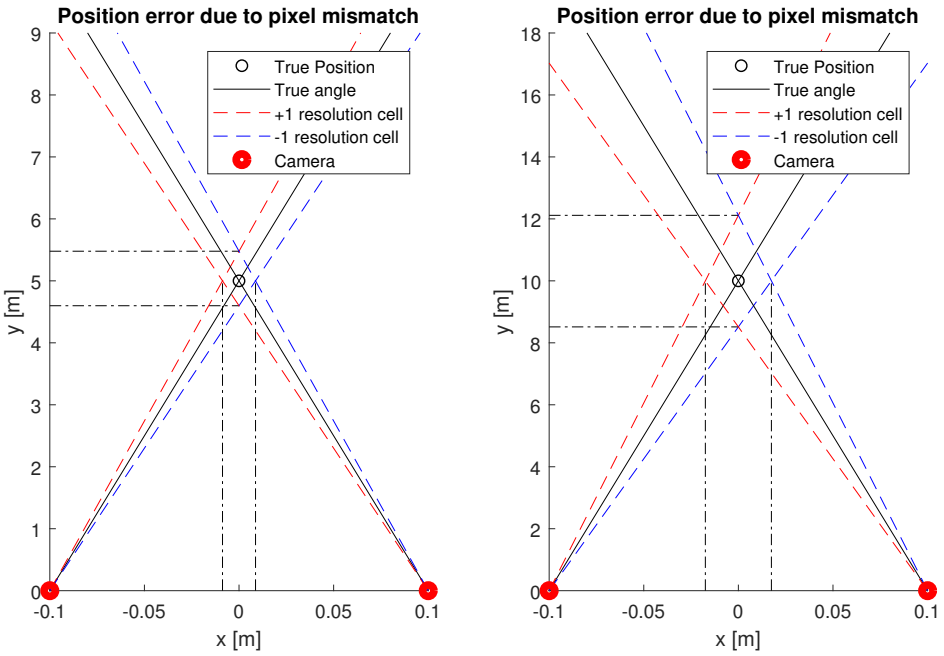


Figure 3.3: The graphs shows a theoretical position error when the pixel matching between images is incorrect. The measured position of the target is given by the intersection between the lines corresponding to the matched pixels from each camera. The left graph shows the position error which occurs if the camera mismatches a feature with one resolution cell at a distance of 5 m away, whilst the right shows the same thing when the feature is 10 m away.

3.2.2 Camera track error

The error in the simulated camera tracks are designed to behave as a real implementation, however it is hard to estimate the performance of a stereo based camera tracker. Since the impact of measurement errors will be reduced by filtering and using multiple features on the targets to estimate the state. Therefore a simple and highly modifiable error model was used for the tracks, which adds errors to the estimated position, heading and extent of the target. These errors were tuned to generate what we found as reasonable errors.

The error of the position in the radial direction is modelled as a function of r, the radial distance between sensor and target, and k, the number of consecutive detections. The used function is

$$e_r(k, r) = br + s_{1,r} r \eta_{1,r}(k) + s_{2,r} r \eta_{2,r}(k) e^{-\xi_r k} \tag{3.9}$$

where $s_{1,r}$, $_{2,r}$ and ξ_r are scaling constants which are used as design parameters, b is a bias constant and $\eta_{1,r}(k)$ and $\eta_{2,r}(k)$ are random variables. The bias constant b

is used to account for the possibility of systematic feature mismatches in a camera triangulation system. Therefore it is unique and constant for each target and taken from $\mathcal{N}(0, \sigma_b^2)$ when the target is first detected. The random variables are given by

$$\eta(k) = \begin{cases} q(k) & \text{if } k = 0 \\ (1-\alpha)\eta(k-1) + \alpha q(k) & \text{if } k \geq 1 \end{cases} \tag{3.10}$$

where α is a design parameter and $q(k) \sim \mathcal{N}(0, 1)$. These terms are used to account for the random nature of any tracking system. By using this model the variance of the error can be calculated and it should match the variance estimation that a real tracking system would estimate the variance to be.

The error in the tangential position is modelled in a similar way, however the bias term is excluded and the design constants have lower values to increase the accuracy of the estimation. The model is

$$e_t(k, r) = s_{1,t} r \eta_{1,t}(k) + s_{2,t} r \eta_{2,t}(k) e^{-\xi_t k}. \tag{3.11}$$

These two errors are transformed to the Cartesian representation which is used in the state representation of the positions. This is done according to

$$e_x(k, r) = e_r \cos(\theta) - e_t \sin(\theta) \tag{3.12}$$
$$e_y(k, r) = e_r \sin(\theta) + e_t \cos(\theta) \tag{3.13}$$

where θ is the angle to the target from the sensor.

The track estimation of the heading of the target is estimated in the same way as the tangential error,

$$e_\varphi(k, r) = s_{1,\varphi} r \eta_{1,\varphi}(k) + s_{2,\varphi} r \eta_{2,\varphi}(k) e^{-\xi_\varphi r} \tag{3.14}$$

where the constants have been scaled to match the angle tracking.

Also the camera systems estimation of the targets extent contains one of these "standard" error representations, however it is combined with an additional scaling term which depends on the number of resolution cells the target occupies in the sensor. The full expression for the error of the extent estimation is thus

$$e_l = \left(s_{1,l}\eta_{1,l}(k) + s_{2,l}\eta_{2,l}(k)e^{-\xi_l k}\right) r e^{-\zeta \theta_{span}} \tag{3.15}$$
$$e_w = \left(s_{1,w}\eta_{1,w}(k) + s_{2,w}\eta_{2,w}(k)e^{-\xi_w k}\right) r e^{-\zeta \theta_{span}} \tag{3.16}$$

where e_l is the error of the length and e_w is the error of the width. θ_{span} is the angular span which the target occupies in the sensor and ζ is a constant used as a design parameter.

3.3 Radar measurements

The radar measurements used in this thesis are generated by simulating radar sensors. The model of the vehicles used to generate these radar measurements is described in Section 3.3.1 and the implementation of the simulation in Section 3.3.2.

3.3.1 Vehicle detection model

The vehicle detection model used in this thesis is the same as in [12], where radar detections are only generated around certain points on the vehicle called reflection points. These reflection points are placed at the corners and wheelhouses of the vehicles and they have a direction and a field of reflection which determines from where the reflection points are visible and thus able to generate detections, see Figure 3.4.

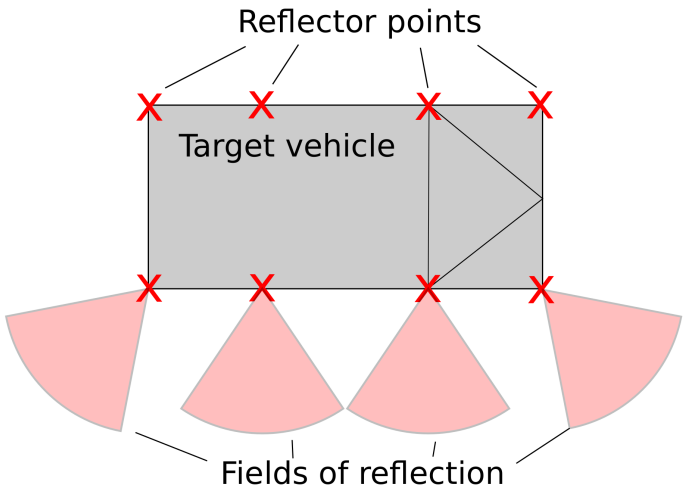


Figure 3.4: Illustration of the reflector points with their fields of reflections on a target vehicle.

3.3.2 Radar simulation

The actual simulation code of the radar sensors is based on the code used in [12]. The radar simulation uses the reflection point system to generate the detections of vehicles within its field of view, and the yielded measurements contain clutter and measurement noise. The spread of measurements taken of a vehicle driving across in front of the ego vehicle (scenario 2) is shown in Figure 3.5. This result is similar to the actual radar measurements gathered from a similar scenario in [13].

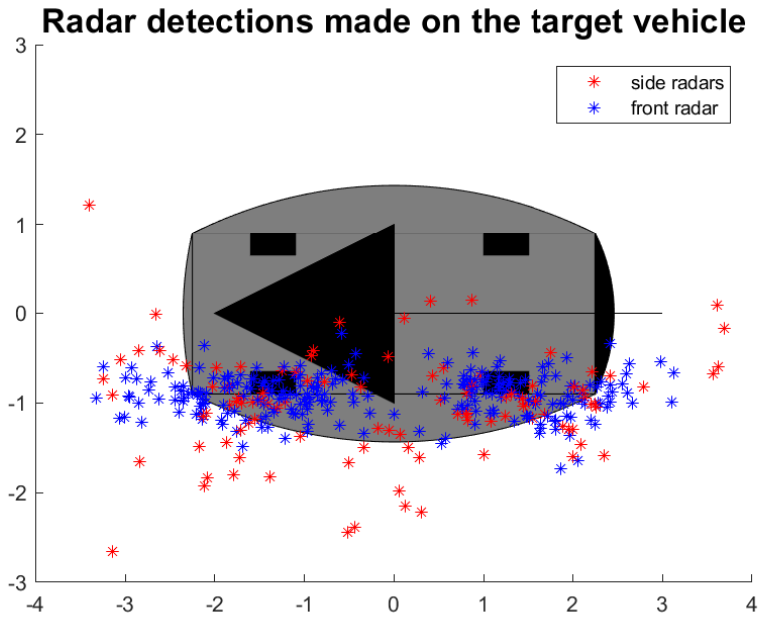


Figure 3.5: The figure shows all radar detections over several time steps, drawn relative the target vehicle, during scenario 2.

In order for a radar measurement of a reflection point to occur in the simulation the sensor must be able to "see" the reflection point. This means that the sensor must have the reflector point within its field of vision and the sensor must be within the radar point's field of reflection for a measurement to occur, this is visualised in Figure 3.6. However, if two or more reflection points appear close together, when seen from the radar sensor, there is a chance of the sensor not being able to differentiate between these detections due to resolution limitations of the sensor. When this occurs the detections are clustered together and the received measurement will be of the cluster and not the individual reflector points.

3.3.3 Radar measurement model

The radar measurements are presumed to originate from a reflection point i on a target vehicle and if so the expected radar measurement would depend on the target vehicle's current state. The radial distance r between the sensor and the target is given by

$$r = \sqrt{\chi^2 + \zeta^2} \tag{3.17}$$

where

$$\begin{bmatrix} \chi \\ \zeta \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} - \begin{bmatrix} x_{sensor} \\ y_{sensor} \end{bmatrix} \tag{3.18}$$

where $[x, y]^T$ is the global position of the target vehicle centre, φ is the heading of the target vehicle, $[x_i, y_i]^T$ is the local position of the reflection point relative to the

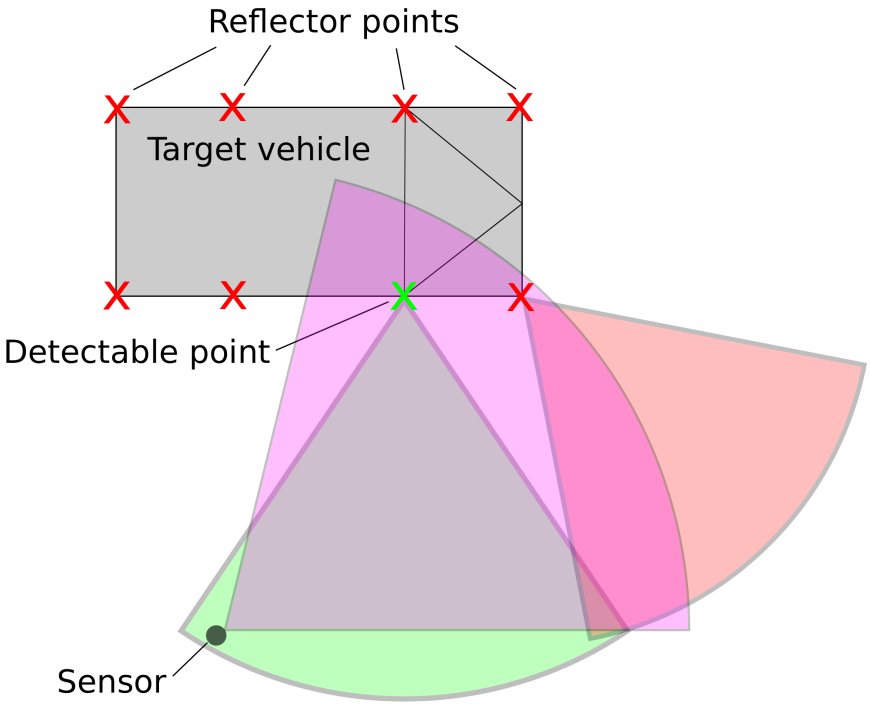


Figure 3.6: Visualisation of how detections are simulated. A target vehicle is shown with its 8 reflector points, shown as red crosses, around the corners and the wheels. One of these is, shown as a green cross, a detectable point since it is inside the sensor's field of vision and the sensor is within the points field of reflection.

target vehicle centre and $[x_{sensor}, y_{sensor}]^T$ is the global position of the sensor. The detection angle θ is then given by

$$\theta = atan2(\zeta, \chi) \tag{3.19}$$

and the radial velocity v_r is given by

$$v_r = v\cos(\psi) - vr_i\lambda\cos(\psi_r) \tag{3.20}$$

where $r_i = \sqrt{x_i^2 + y_i^2}$ is the distance to the reflector point from the vehicle centre, λ is the inverse turn radius of the target and

$$\psi = \theta - \varphi$$

$$\psi_r = \theta - \varphi - \phi_r - \pi/2$$

$$\phi_r = atan2(y_i, x_i).$$

These together form the radar detection

$$r_d = [r, \theta, v_r]$$

which depends on from which reflector point the detection originated from.

3.4 Radar target tracking

The target tracking idea used in this thesis builds on that at each time step multiple measurements are gathered from each radar sensor. The measurements from the sensors may contain clutter and detections of the target, where the radar measurements are assumed to appear around reflector points on the target. By predicting the position of these reflector points one can estimate which measurements originated from the target. These measurements can then be used to estimate the targets state through filtering.

An extension to this tracking method is to use multiple track hypotheses, state estimations of the target, and generating multiple data assignment hypotheses, ways of how the measurements should be assigned, for each track hypothesis. When this is used the number of track hypotheses increases exponentially over time and therefore a method which reduces the number of hypotheses is required.

The target tracking algorithm used in this thesis can be described as four steps which are iterated in each time step and where the three last steps are iterated for each sensor.

1. Track prediction. The track hypotheses from the previous time step are predicted forward using the motion model of the target. The motion model is described in section 3.1 and the track prediction is described in section 3.4.1.

2. Data association. Create multiple possible measurement assignments for each current state estimation. This is described in section 3.4.2.

3. Track update. Update the current track estimations using the measurement assignments generated in the data association step. How this is done is described in section 3.4.3.

4. Track management. Reduce the number of tracks by removing the updated tracks which correspond poorly with the observed radar data. The track management is described in section 3.4.4.

The updated track that corresponds best with the observed data after all the sensors have been iterated through, is used as the systems state estimation for this time step. The method is also illustrated in Figure 3.7.

3.4.1 Track prediction

The track prediction step is only done once in each time step. It takes the track hypotheses from the previous time step and calculates the predicted state for each hypothesis. This is done by using the extended Kalman filter equations 2.15, 2.16 and the motion model described in Section 3.1. This results in a set of state pre-

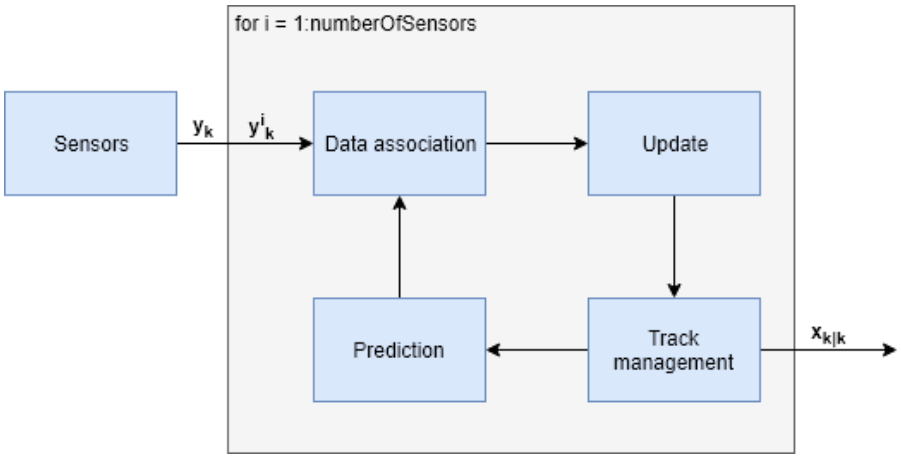


Figure 3.7: Illustration of the algorithm, for one time sample, in the radar tracker. The algorithm runs sequentially for the number of sensors used, before the posterior is output.

dictions of the target's state, one state prediction for each track hypothesis used. The set of predicted states is what is output from the track prediction system and is used for the data association between the radar measurement and the reflector points.

3.4.2 Data association

The data association system uses the current set of state estimations, generated either by the track prediction or the track management, to assign the radar measurements to the reflection points on the target. The Global Nearest Neighbour (GNN) method in combination with Munkres algorithm is used to generate the associations. Multiple ways of assigning the measurements to the reflection points are generated for each predicted state. The reason for this is described in Section 2.3.

The data association is done for one state estimation at the time. From the state estimation the expected measurements from each reflector point is calculated using the model described in Section 3.3.3. These expected measurements can be written as $\hat{\mathbf{y}}_{j,r} = \mathbf{h}(\hat{\mathbf{x}}_{j,r})$ where h is the measurement model for the sensor and $\hat{\mathbf{x}}_{j,r}$ is the estimated state of the r:th reflection point in the j:th state estimation.

The cost for associating a radar measurement to a certain estimated reflector point is calculated using the Mahalanobis distance, defined in Equation (2.26). This gives that the cost for assigning, $\mathbf{y}_n$, the n:th measurement from the sensor, to, $\hat{y}_{j,r}$, the r:th reflection point in the j:th state estimation to be calculated as

$$c_{j,n,r} = \sqrt{(\mathbf{y}_n - \hat{\mathbf{y}}_{j,r})^T S_j^{-1} (\mathbf{y}_n - \hat{\mathbf{y}}_{j,r})} \tag{3.21}$$

where S_j^{-1} is the innovation covariance of the j:th state estimation. This is done for all combinations of measurements and reflector points in each state estimation, and

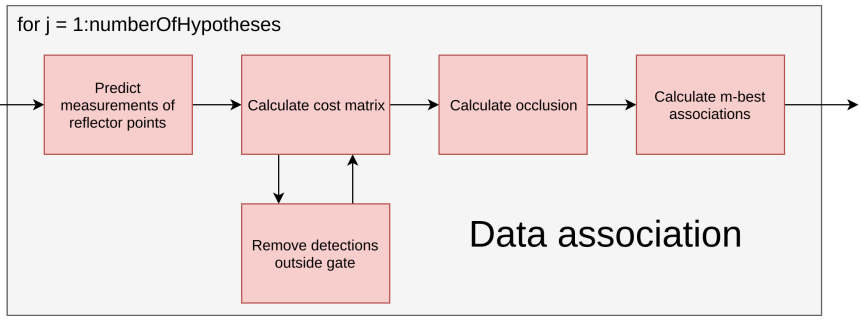


Figure 3.8: Illustration of the data association.

the costs are arranged into a cost matrix C_j. Where each column represents a measurement and each row a reflector point. Such that $C_j(n, r)$ is the cost for assigning the n:th measurement to the r:th reflection point in the j:th state estimation.

Gating is then used to remove the measurements which do not correspond to any reflection point. This is done by comparing the costs in the cost matrix to a threshold. Any measurements that does not have one or more assignment costs lower than the threshold associated to it is removed. The threshold is set using the chi-square distribution for three degrees of freedom, the theory behind this is given in section 2.3.2. By doing this the cost matrix only consists of the measurements which at least have one valid association that can be made.

To enable the possibility of assigning a measurement as clutter, additional rows in the cost matrix are added. Where each row corresponds to one measurement being assigned to clutter. This is accomplished by introducing as many extra rows as measurements, and the costs for all but one measurement association is set very high, and the last cost is set to the same as the threshold used for the gating.

Munkres algorithm is then applied to the resulting cost matrix and the data association with the lowest cost associated to it is generated. This gives the first assignment hypothesis of the measurements. The costs in the cost matrix for the associations used in the assignment hypothesis are then replaced individually one at the time with a large cost. Munkres algorithm is then run on this updated cost matrix. This replacing of weights prevents the same set of matches to be selected again, thus different association hypotheses will be generated. This is then iterated until sufficiently many assignment hypotheses are generated.

3.4.3 Track update

The track update system takes the current state estimations and updates them using the measurement associations generated in the data association system. The update is done using one measurement at the time using the extended Kalman filter described in Section 2.1.3. This results in a multitude of updated state estimations, one for each assignment hypothesis in each state estimation.

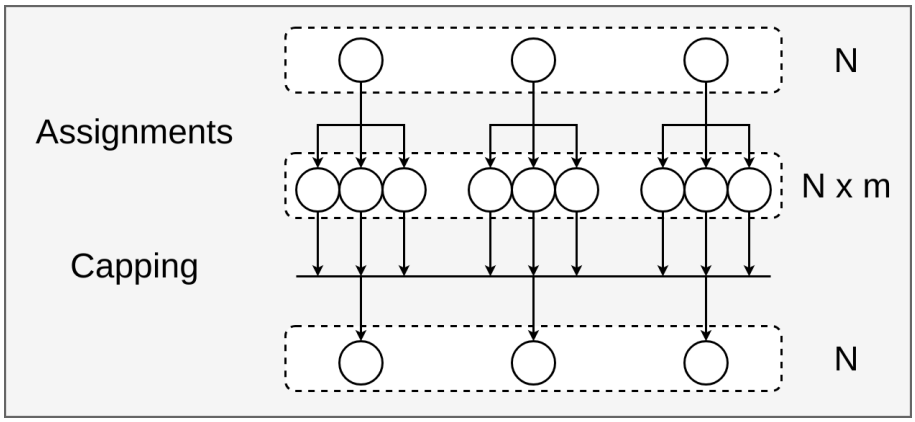


Figure 3.9: Description of how the number of hypotheses changes in the tracking algorithm given that N is the number of used hypotheses and m is the number of different assignments that are tested. Assignments is the step where each hypotheses is updated in m different ways. Capping is the step where the $N \times M$ number of hypotheses are reduced to N by choosing the best ones.

3.4.4 Track management

Since the track output system generates multiple state estimations for each one of the inputted ones, a way to remove redundant tracks needs to be implemented. This is done by sorting all the updated state hypotheses based on the total cost of the used measurement assignment for it, this is calculated by summing the costs for each measurement. The measurements which are used have their Mahalanobis distance as their costs, while the measurements that are either gated out or assigned to nothing are said to cost as much as the threshold used for the data association. The updated state estimations with the lowest costs associated to them are then saved and used in the next step, either as the set of current state estimations for the next sensor or as the previous time steps state estimations. The updated state estimation when all sensors has been used is used as the state estimation for that time step.

3.5 Track to track fusion

The fusion of the information from the radar tracker and the stereo-camera is done by track to track fusion using the LMMSE fuser described in section 2.4.1. Two configurations of fusion is tested and the setup of these can be seen in figure 3.10.

The right configuration fuses the tracks from the radar tracker and the stereo-camera each time step, this fused track is the state estimation. The other configuration fuses the tracks each time step in the same way as the first configuration. The fused state estimation is however used as feedback into the radar tracker such that the radar tracker now uses the fused track in its next iteration. In this configuration however,

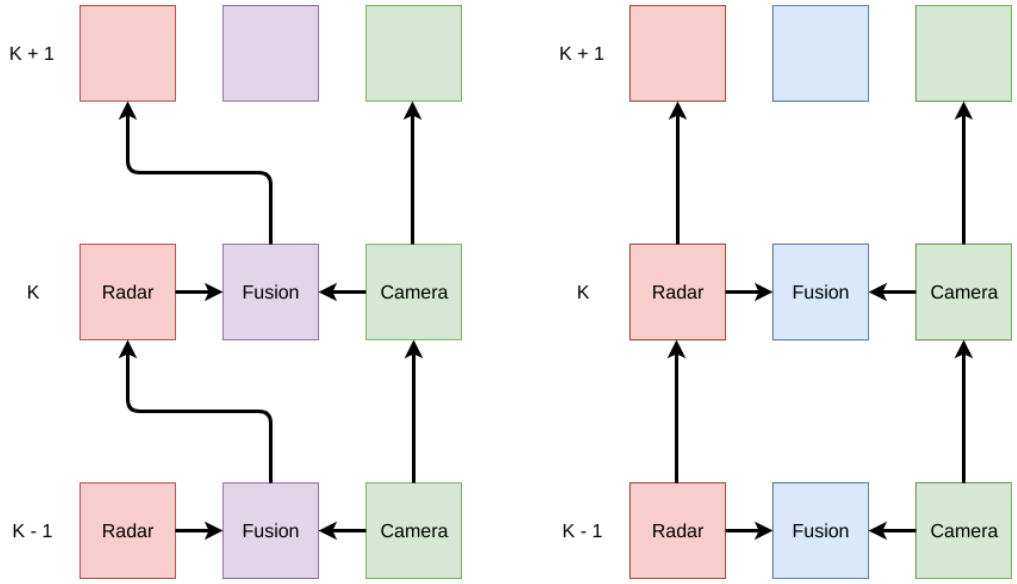


Figure 3.10: Framework of two configurations of track to track fusion. The left image shows the track to track fusion with feedback into the radar tracker. The right image shows the track to track fusion without feedback.

the stereo-camera tracks are used multiple times which could result in estimation errors since the fusion method used assumes that the tracks from the radar tracker and the stereo-camera system is uncorrelated.

4

Results and Discussion

In this chapter we will present the results from our extended target tracking implementation and discuss its performance. The design parameters which are used to adjust the implementation are:

- Number of hypotheses
- Uncertainty scaling
- Gating

The effects of changing these are shown and discussed. In Section 4.1 the different test scenarios used to test these design parameters are introduced. In Sections 4.2 to 4.7 the results of the algorithms are presented starting with the chosen combination of design parameteres. Finally, in Section 4.8 the crosscorrelation between the tracks are estimated and discussed.

The performance of the trackers were evaluated using the RMSE (root mean square error) for each state calculated for each time frame from 1000 Monte Carlo runs of the same scenario. This meant that the same true states were used in all the runs, but the radar detections and the camera track used were unique for each run. By using this approach we can comfortably say that the results are reliable. Note that this is not a guarantee of the performance.

4.1 Evaluation scenarios

The tracking algorithms used in this thesis is evaluated in two different driving scenarios and in different intensities of clutter. The two driving scenarios can be seen in Figure 4.1. Scenario 1 simulates the ego vehicle being overtaken by another vehicle on the right hand side. In the beginning of this scenario the target vehicle is only visible in the right facing radar sensors and after a few time samples it enters the field of vision of the front facing radar and stereo-camera. Thus the target is initially only being tacked by the radars. In scenario 2 the target vehicle is instead passing in front of the ego vehicle, so the target is in the stereo-cameras field of view

for the entire simulation. The paths of the target vehicle in these two scenarios are represented in Figure 4.1 along with the fields of vision of the sensors. The intensity of clutter is also varied in the scenarios to investigate the performance changes due to the gating.

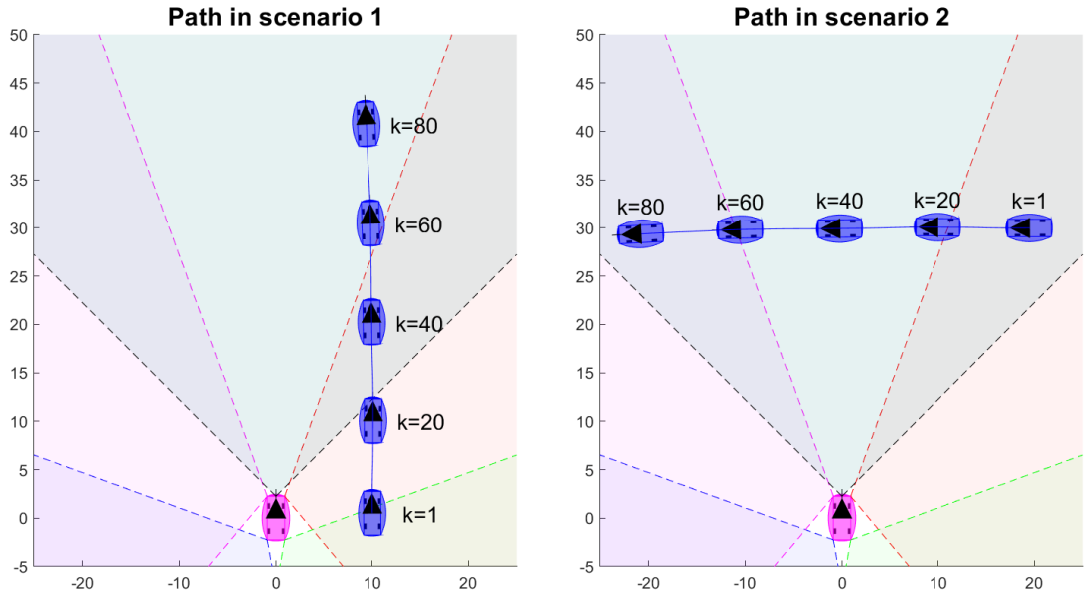


Figure 4.1: Visualisation of the two scenarios which are used for testing. In scenario 1, in the left image, the target vehicle is passing by the ego-vehicle. In scenario 2, in the right image, the target vehicle is going across the ego-vehicles path. In both of the scenarios the ego-vehicle is standing still and the time in between samples is $t = 0.05\,\text{s}$. In both images the field of view of all of the sensors are also seen, of which one is the stereo-camera with direction along the y-axis.

4.2 Base case performance

This section is used to present the performance of the final design of the tracker implementation. This filter implementation was designed by comparing how the performance was affected by the design parameters and the different scenarios, some of the choices made are discussed later in this chapter. The selected base case filter uses 10 hypotheses, the gating parameter 0.9 and covariance scaling 2 for the states x, y and φ. How the performance is affected by the number of hypotheses is discussed in section 4.3, by covariance scaling in section 4.4 and by gate size in section 4.6.

This base case filter implementation of the T2T and T2TWFB fusion trackers are better than the individual radar and camera tracks in the two different scenarios. This can be seen by comparing the different trackers mean RMSE presented in Table 4.1, where the T2T and the T2TWFB both have lower errors than the initial tracks. Which is to be expected since the T2T and T2TWFB implementations both

Table 4.1: The mean over time of the RMSE for 1000 simulations of the final tracking implementation for the two scenarios.

	Scenario 1				Scenario 2			
	Radar	Camera	T2T	**T2TWFB**	Radar	Camera	T2T	**T2TWFB**
x	0.5710	1.1742	0.4278	**0.3335**	0.7166	1.2654	0.5373	**0.4539**
y	1.0491	2.2336	1.1746	**0.8999**	0.3736	2.5989	0.5186	**0.1956**
v	0.9451	-	0.9813	**0.5658**	1.3381	-	1.5226	**0.8938**
a	0.5161	-	0.5362	**0.4250**	0.6451	-	0.8923	**0.6358**
φ	0.1703	0.0331	0.0638	**0.0614**	0.1220	0.0378	0.0374	**0.0275**
λ	0.0636	-	0.0926	**0.0457**	0.0786	-	0.1098	**0.0426**
w	0.0648	0.0251	0.0296	**0.0262**	0.1004	0.0019	0.0018	**0.0018**
l	0.0793	0.0355	0.0644	**0.0588**	0.1743	0.0046	0.0045	**0.0045**

combines the information from each initial track, and thus have more information to base the estimations on. However the T2TWFB is also better than the T2T. This can also be seen in Table 4.1, and in Figure 4.2 where the RMSE of the estimated x and y position from the different trackers are presented for scenario 1. From this image one can determine that the T2TWFB algorithm gives the best results. The T2TWFB was also the best at tracking in scenario 2, see Figure 4.3. The main advantage the T2TWFB implementation has over the T2T implementation is that it reduces the chance of the radar tracks diverging. Thus, in the cases where the radar tracker alone diverges the T2T estimations will be made using these incorrect radar estimations and thus contain errors, whilst the T2TWFB will not end up in this situation. The implications of this is further discussed in section 4.7.

The T2TWFB implementation does, however, struggle with overconfidence in its position. This can be seen in Figure 4.4 where the 3σ ellipses of the position estimation does not encircle the correct state for the T2TWFB algorithm. The figure shows the estimated position of the different trackers and their 3σ ellipses from one tracking done in scenario 2. This over confidence issue comes from the radar tracker implementation, which also can be seen has the same issue in the figure.

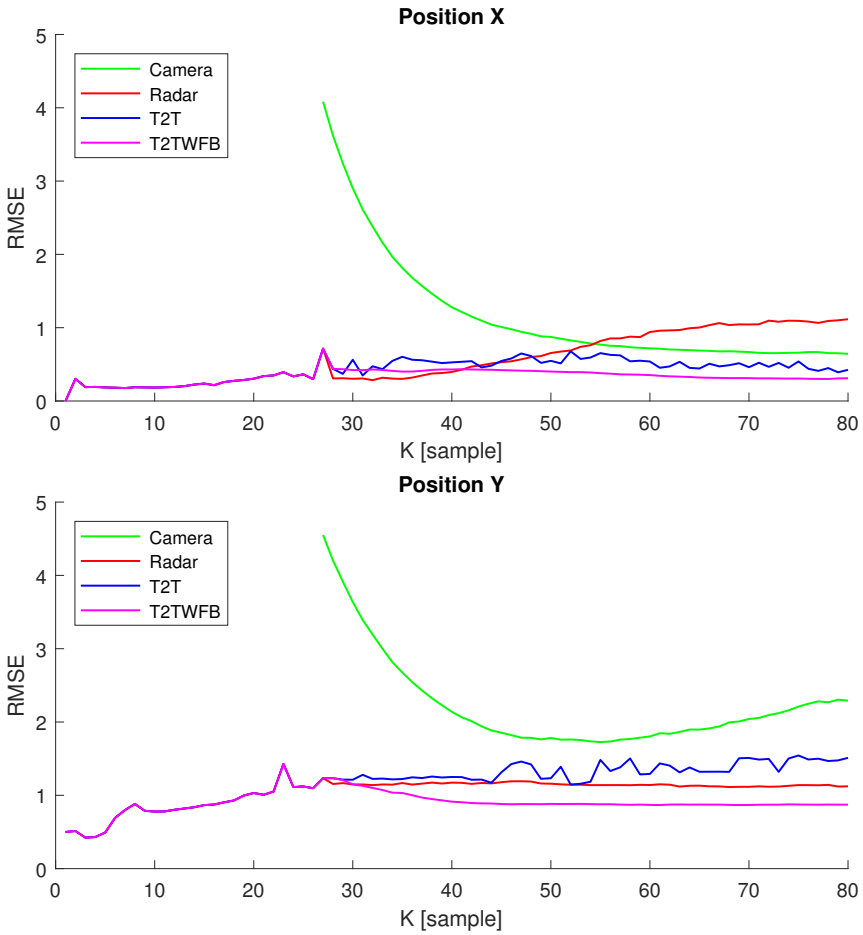


Figure 4.2: The graphs shows the RMSE of the position estimates for the different used trackers when the target vehicle passes the ego vehicle on the right hand side, scenario 1.

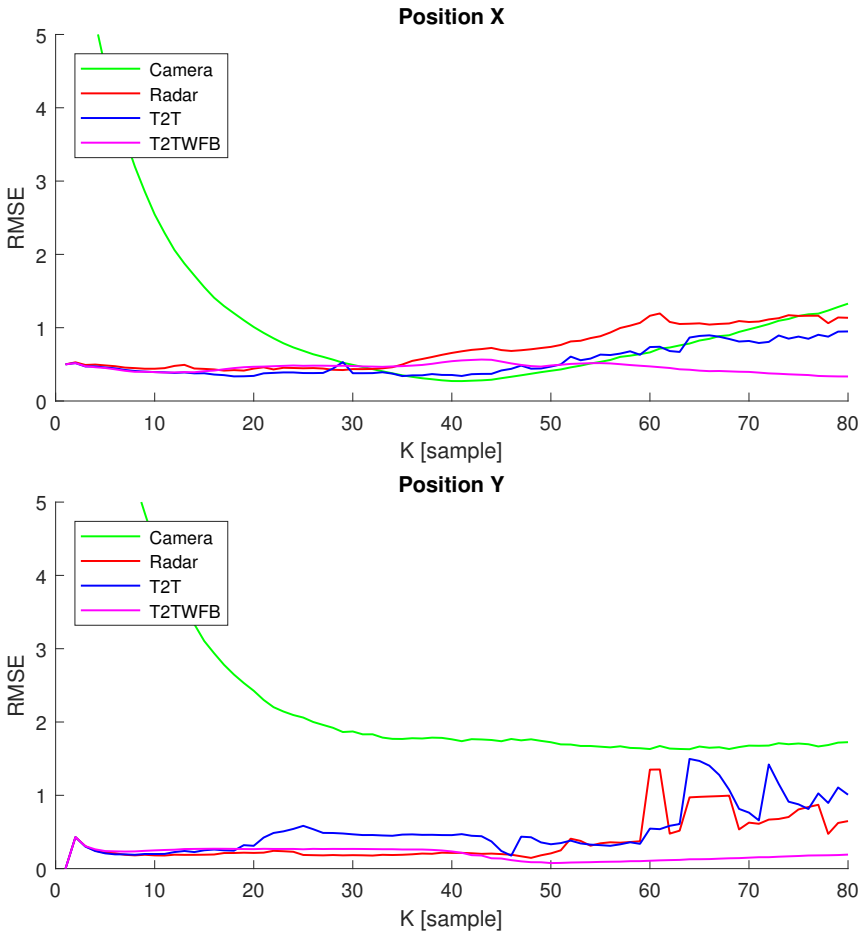


Figure 4.3: The graphs shows the RMSE of the position estimates for the different used trackers when the target vehicle passes in front of the ego vehicle, scenario 2.

4.3 Varied number of hypotheses

This section is used to present the gathered results for how the number of used hypotheses in the radar tracker affects the performance of the T2T fusion and the T2T fusion with feedback trackers.

The performance of the radar tracker and the T2T was greatly increased when the number of used hypotheses were increased, this can be seen in Figure 4.5 where the RMSE for the radar tracker and T2T decreases with the number of hypotheses used. This corresponds well with the predicted behaviour of the radar tracker, since the probability of diverging should decrease when more hypotheses are used. The reason for the large and jumping behaviour of the T2T graph for 1 and 5 used hypotheses is due to that when the radar tracker diverges from the true state. The estimated variance of the radar state might decrease when the predicted state matches a clutter measurement making the radar tracker overconfident in its estimation and, thus,

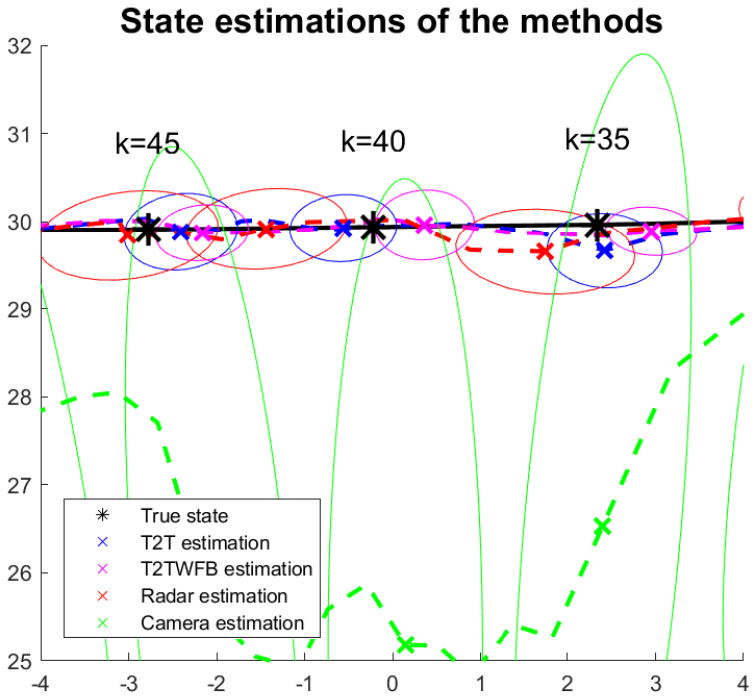


Figure 4.4: The figure shows the estimated positions from the different trackers for three time steps along with their corresponding 3σ ellipses which describes the uncertainty of the estimate.

causing the T2T estimate to be incorrect. This jumping behaviour can also be seen to occur in all states in T2T, when 1 or 5 hypotheses are used, by noting that the mean RMSE (shown in table 4.2) is much higher than the median RMSE (shown in table 4.3) for these cases.

The results also shows that the performance of the T2TWFB tracker is only slightly increased when more hypotheses are used. This is most likely due to that the state feedback from the camera limits the error of the state estimation in the radar tracker to the performance of the camera. Thus removing the possibility to diverge completely and therefore the T2TWFB tracker is able to use only one hypothesis and still perform well. This is an interesting result since the number of hypotheses increases the computational resources needed to use the tracking algorithm. And since its performance is comparable to the ones with more used hypotheses it might actually be preferable over the other.

Table 4.2: The mean over time of the RMSE for 1000 simulations of scenario 2 for different number of used hypotheses in the radar tracker.

	T2T			T2TWFB		
	1 Hyp	5 Hyp	**10 Hyp**	1 Hyp	5 Hyp	**10 Hyp**
x	13.2777	16.2619	**0.5373**	0.4973	0.4640	**0.4539**
y	28.6427	8.0035	**0.5186**	0.3981	0.2743	**0.1956**
v	25.1971	14.3679	**1.5226**	1.1381	0.9260	**0.8938**
a	11.8808	7.3367	**0.8923**	0.7790	0.6998	**0.6358**
φ	0.1415	0.0928	**0.0374**	0.0339	0.0305	**0.0275**
λ	4.5689	2.4646	**0.1098**	0.1478	0.0460	**0.0426**
w	0.0018	0.0021	**0.0018**	0.0018	0.0017	**0.0018**
l	0.0046	0.0052	**0.0045**	0.0046	0.0043	**0.0045**

Table 4.3: The median over time of the RMSE for 1000 simulations of scenario 2 for different number of used hypotheses in the radar tracker.

	T2T			T2TWFB		
	1 Hyp	5 Hyp	**10 Hyp**	1 Hyp	5 Hyp	**10 Hyp**
x	2.8190	2.5457	**0.4430**	0.4954	0.4761	**0.4694**
y	4.0029	2.1294	**0.4502**	0.4427	0.2989	**0.2038**
v	11.3726	4.9923	**1.6477**	1.2506	1.0138	**0.9574**
a	7.0979	3.3317	**0.9706**	0.7465	0.6658	**0.6162**
φ	0.1387	0.0943	**0.0324**	0.0188	0.0182	**0.0190**
λ	1.9627	1.1254	**0.1080**	0.1571	0.0432	**0.0419**
w	0.0008	0.0009	**0.0007**	0.0007	0.0007	**0.0007**
l	0.0019	0.0023	**0.0019**	0.0018	0.0018	**0.0019**

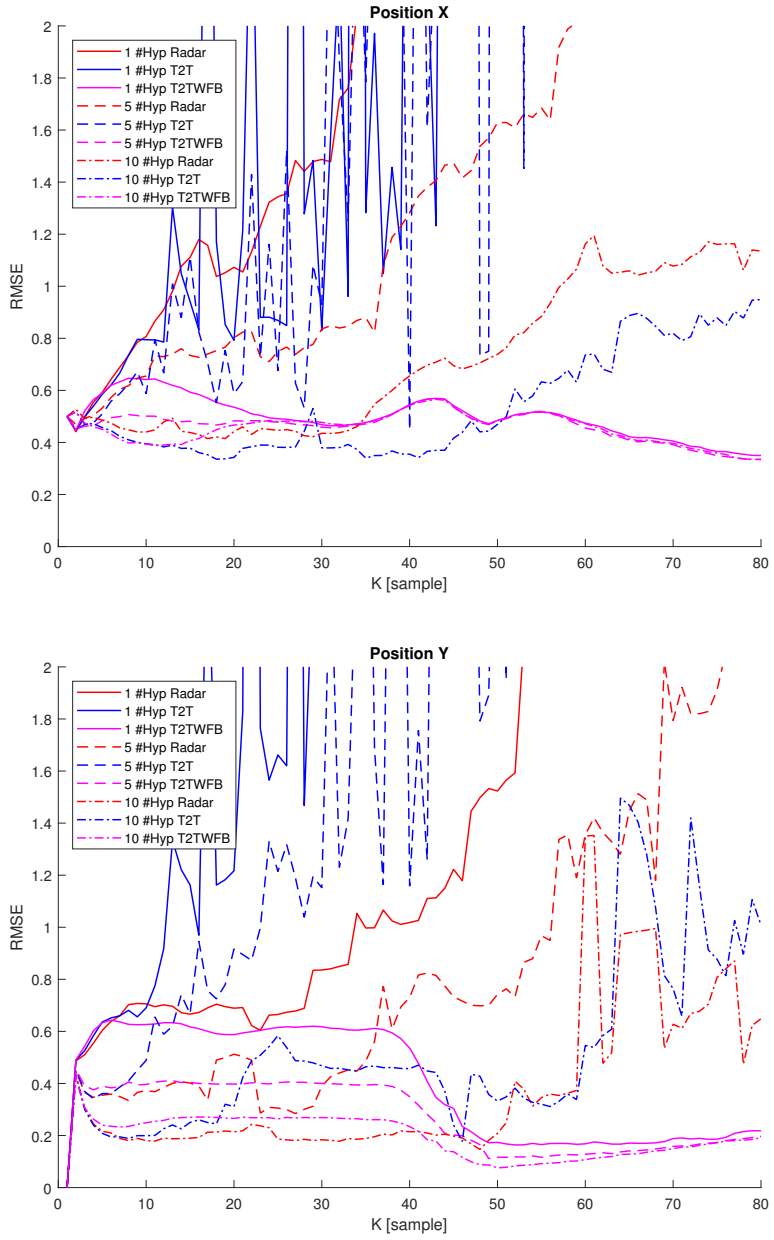


Figure 4.5: The graphs shows a comparison of how the performance of the trackers are affected by the number of used hypotheses for the estimation of the position along the cartesian x axis in the top image and the y axis in the bottom one. Note that the performance of the radar and the T2T implementations are greatly improved when the number of hypotheses increases, whilst the T2TWFB implementation is only slightly improved.

4.4 Varied state covariance scaling

Due to that the radar tracker presumes the correct measurement assignment was used in the update step, it tends to underestimate the uncertainty of the state. Therefore is the estimated covariance matrix from the radar tracker rescaled to mitigate this tendency. This is, however, only done for the position and heading states, since these are the states which are over confidently estimated. By using covariance scaling the performance for both the T2T and T2TWFB tracks are improved, this can be seen both in table 4.4 and in Figure 4.6. The performance of the T2T tracker is greatly increased when covariance scaling is used. This is most likely due to that the over confidence in the radar tracker is reduced and therefore lowers the probability of diverging. The T2T algorithm seems to perform best when the covariance scaling is 5, while the T2TWFB is performs best when it is set to 2.

Table 4.4: The time mean RMSE for 1000 simulations of different covariance scaling settings.

	T2T			T2TWFB		
	CovScale 1	**CovScale 2**	CovScale 5	CovScale 1	**CovScale 2**	CovScale 5
x	2.3693	**0.5373**	0.4832	0.4675	**0.4539**	0.5361
y	3.7420	**0.5186**	0.2271	0.3011	**0.1956**	0.2142
v	10.5793	**1.5226**	1.5088	0.6759	**0.8938**	1.1205
a	5.1620	**0.8923**	0.7825	0.6600	**0.6358**	0.7157
φ	0.0834	**0.0374**	0.0252	0.0311	**0.0275**	0.0273
λ	1.3776	**0.1098**	0.0663	0.0434	**0.0426**	0.0501
w	0.0019	**0.0018**	0.0018	0.0018	**0.0018**	0.0018
l	0.0048	**0.0045**	0.0045	0.0045	**0.0045**	0.0045

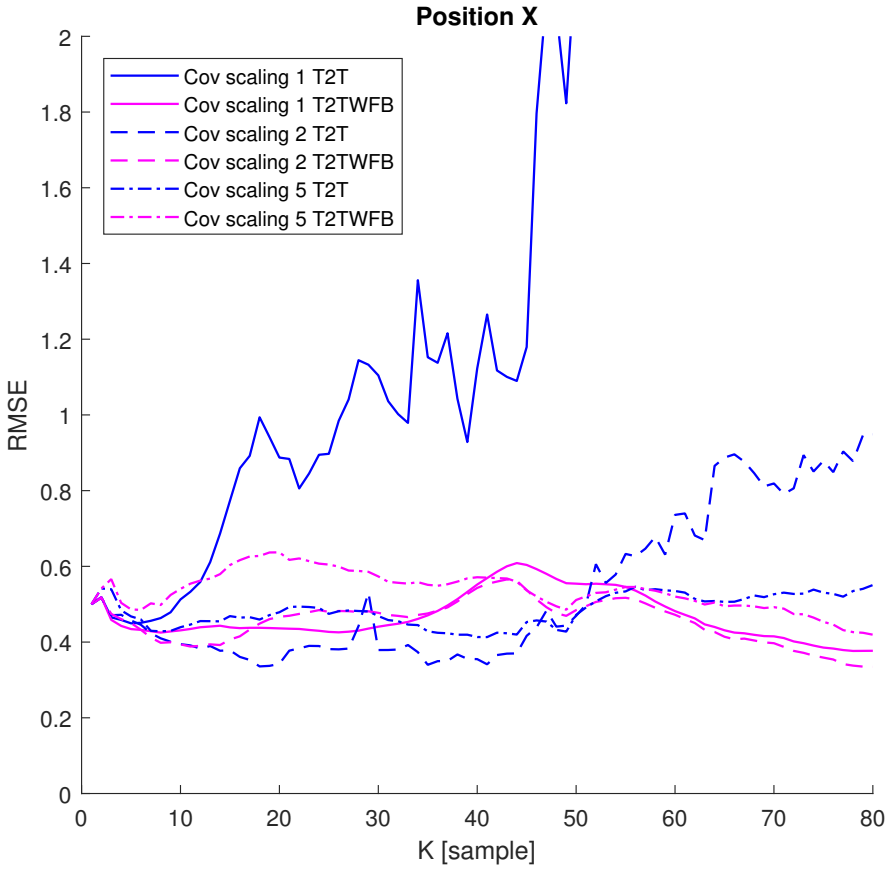


Figure 4.6: The graph shows a comparison of how the performance of the tracker is affected by covariance scaling.

4.5 Varied clutter intensity

In this section the performance of the T2T and T2TWFB trackers are evaluated when there are different levels of clutter present in the radar measurements. The predicted behaviour from theory is that the performance of the T2T tracker should decrease when higher levels of clutter are present since the radar estimation should have a higher chance of diverging since there are more clutter which may interfere with the data assignment. This is also what the results show, see table 4.5 and Figure 4.7, that the performance of the T2T algorithm decreases for the high levels of clutter. However one interesting thing to note is that the performance of the T2T tracker in the case of low and medium clutter levels are similar. This is rather unexpected since the performance of the low clutter simulation in theory should be better, however this is probably due to that the variance of the radar tracks is incorrectly estimated. These simulations uses the covariance scaling 2 which is the best performing one for the T2TWFB tracker when medium clutter is present, thus the variance is most likely not correctly corrected for the radar tracker.

The T2TWFB tracker however is barely affected at all by the clutter which indicates that it is a stable implementation.

Table 4.5: The mean RMSE over time for varied clutter intensities.

	T2T			T2TWFB		
	Low	**Medium**	High	Low	**Medium**	High
x	0.5620	**0.5373**	0.9866	0.4378	**0.4539**	0.4622
y	0.6104	**0.5186**	0.8782	0.1576	**0.1956**	0.1896
v	1.7023	**1.5226**	2.5653	0.8465	**0.8938**	0.8808
a	1.0543	**0.8923**	1.7145	0.6133	**0.6358**	0.6536
φ	0.0511	**0.0374**	0.0691	0.0270	**0.0275**	0.0281
λ	0.1366	**0.1098**	0.2732	0.0413	**0.0426**	0.0452
w	0.0017	**0.0018**	0.0017	0.0017	**0.0018**	0.0018
l	0.0043	**0.0045**	0.0044	0.0044	**0.0045**	0.0044

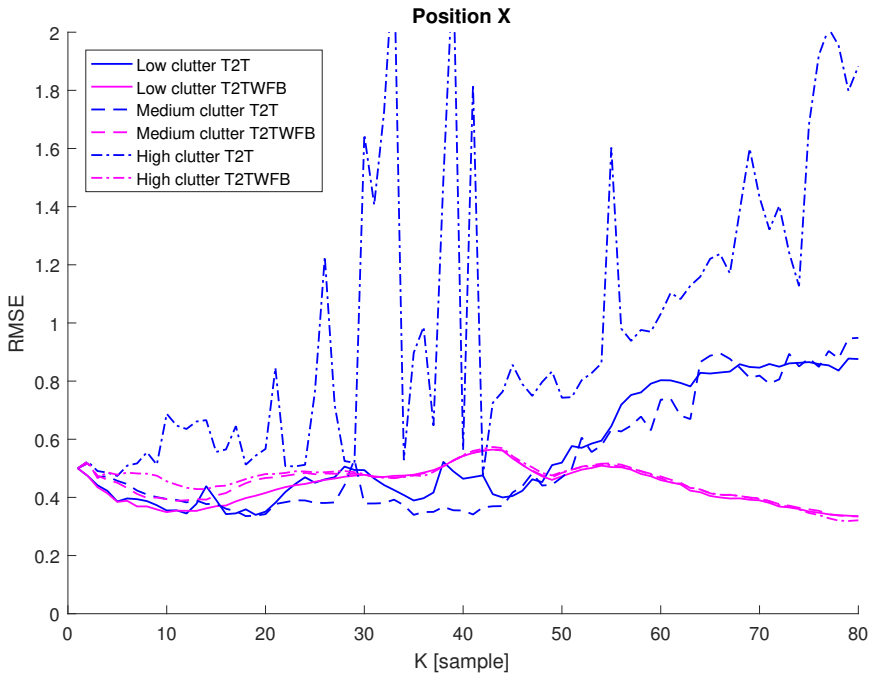


Figure 4.7: The graph shows a comparison of how the performance of the tracker is affected by the clutter intensity.

4.6 Varied gate size

The gate size is used to determine how well the measurements need to correspond to the predicted ones in order to be used in the update algorithm. Therefore if one uses a to low gate, some of the actual measurements will be incorrectly discarded

as clutter. Whilst if the gate is to high, clutter might be incorrectly included and used as measurements. Thus if the gate is incorrectly set the performance of the tracker will be decreased. This can be seen in the performance of the T2T algorithm when a lower gate is used, see table 4.6 and figure 4.8, where the performance of the tracker is reduced when the lower gate is used. This is most likely due to that the algorithm then ignores actual measurements and thus diverges from the correct state estimation. However the higher gate size does not seem to affect the performance in a major way, for neither the T2T or the T2TWFB algorithms. This indicates that the issue of including clutter as measurements is not a major issue, at least not for the used clutter intensity. One may also note that the T2TWFB implementation is barely affected by the changes in gate size. This is most likely due to that it can compensate for a incorrect assignment made in the radar tracker with the camera. Whilst the same error would cause the radar to diverge from the correct state when feedback is not used.

Table 4.6: The time mean RMSE for 1000 simulations of scenario 2 for varied gate sizes.

	T2T			T2TWFB		
	Gate 0.8	**Gate 0.9**	Gate 0.99	Gate 0.8	**Gate 0.9**	Gate 0.99
x	2.3070	**0.5373**	0.6018	0.5000	**0.4539**	0.4020
y	1.5565	**0.5186**	0.5300	0.2417	**0.1956**	0.1581
v	4.0031	**1.5226**	2.0378	0.9009	**0.8938**	0.9022
a	2.2092	**0.8923**	1.1992	0.6150	**0.6358**	0.6824
φ	0.0650	**0.0374**	0.1508	0.0570	**0.0275**	0.0279
λ	0.4720	**0.1098**	0.9973	0.2378	**0.0426**	0.0491
w	0.0018	**0.0018**	0.0023	0.0018	**0.0018**	0.0018
l	0.0045	**0.0045**	0.0045	0.0045	**0.0045**	0.0045

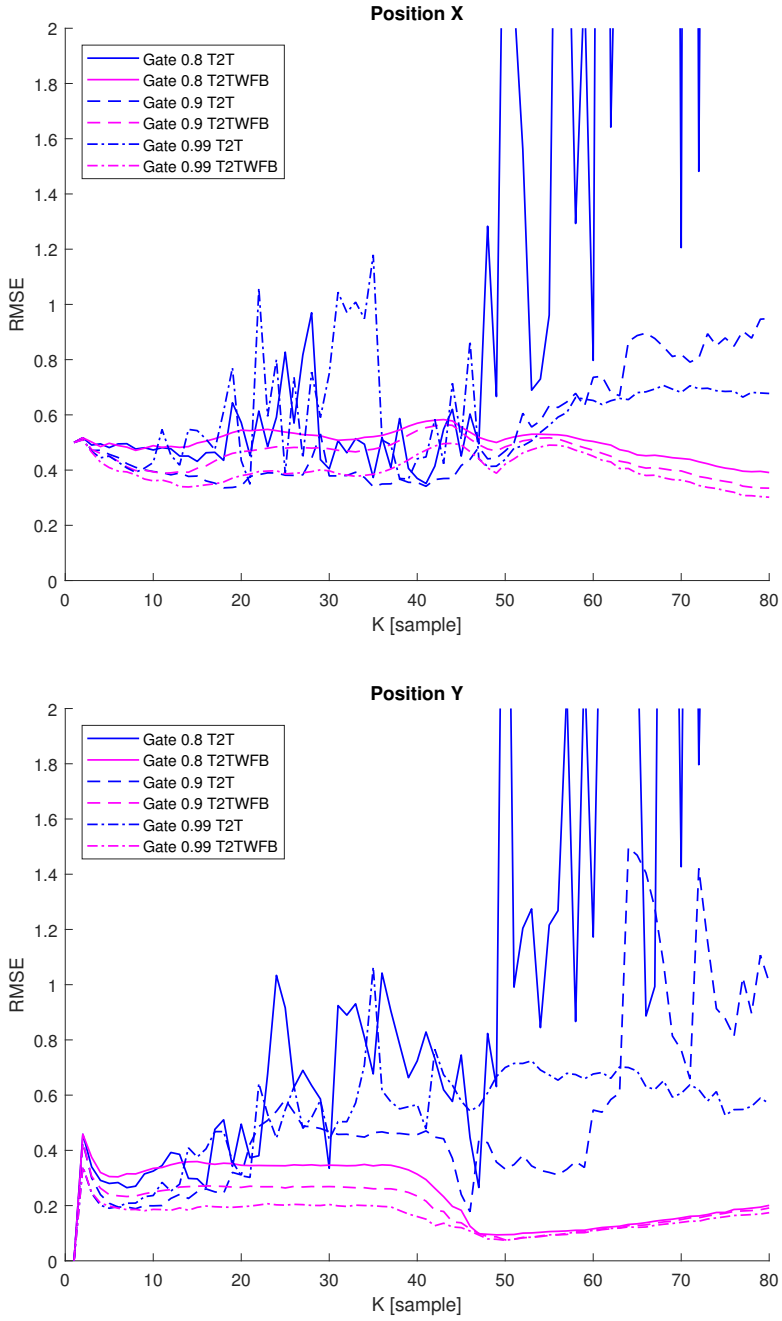


Figure 4.8: The graphs shows a comparison of how the performance of the tracker is affected by the gating size.

4.6.1 Varied gate size with high clutter

When the clutter intensity is increased the probability of incorrectly using a clutter measurement to update the state estimation of the target increases. Thus increasing the probability of the radar tracker diverging. The effects of this can be mitigated by reducing the gate size used in the tracker. That this works is shown in table 4.7 where

the mean RMSE is reduced for the T2T algorithm when a lower gating threshold is used. However the performance of the T2TWFB algorithm is only slightly affected by the changed gate size. This indicates that the large gate in combination with the high clutter intensity increases the chance of causing the radar tracker to diverge.

Table 4.7: The time mean RMSE for 1000 simulations of scenario 2 for varied gate sizes with high levels of clutter.

	T2T		T2TWFB	
	Gate 0.9	Gate 0.99	**Gate 0.9**	Gate 0.99
x	**0.9866**	4.9524	**0.4622**	0.4268
y	**0.8782**	3.0801	**0.1896**	0.1692
v	**2.5653**	7.9673	**0.8808**	1.1248
a	**1.7145**	5.0992	**0.6536**	0.7686
φ	**0.0691**	0.1508	**0.0281**	0.0320
λ	**0.2732**	0.9973	**0.0452**	0.2122
w	**0.0017**	0.0023	**0.0018**	0.0017
l	**0.0044**	0.005	**0.0044**	0.0044

4.7 Removed diverged radar tracks

The used radar tracker implementation does not have a way to identify when it has diverged from the target. This causes major errors from some of the runs since it continues to use the radar estimation in the T2T fusion algorithm. However if the diverged tracks are removed and ignored. Then the performance of the radar and T2T implementations for the remaining tracks is greatly improved, see Figure 4.9 and Table 4.8. This means that a major part of the RMSE for the radar and T2T implementations are caused by a small part of the simulation runs.

Table 4.8: The time mean RMSE for 1000 simulations of scenario 2 calculated using only the 90%, 95% and 100% best radar tracks.

	Radar			T2T			T2TEFB		
	100 %	95%	90 %	100 %	95 %	90 %	100 %	95 %	90 %
x	0.7166	0.5012	0.4106	0.5373	0.3908	0.3501	0.4539	0.4496	0.4484
y	0.3736	0.1531	0.1282	0.5186	0.1539	0.1314	0.1956	0.1813	0.1754
v	1.3381	1.0248	0.9115	1.5226	1.0532	0.9397	0.8938	0.8730	0.8669
a	0.6451	0.5732	0.5463	0.8923	0.6534	0.6313	0.6358	0.6144	0.6111
φ	0.1220	0.0674	0.0570	0.0374	0.0239	0.0233	0.0275	0.0267	0.0266
λ	0.0786	0.0641	0.0568	0.1098	0.0557	0.0498	0.0426	0.0416	0.0415
w	0.1004	0.0921	0.0841	0.0018	0.0018	0.0018	0.0018	0.0018	0.0018
l	0.1743	0.1742	0.1741	0.0045	0.0045	0.0045	0.0045	0.0045	0.0045

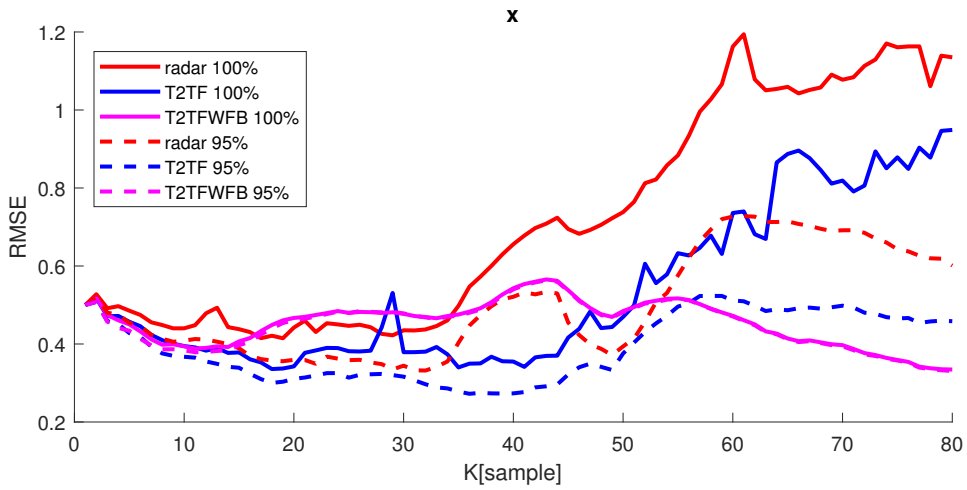


Figure 4.9: The graph shows a comparison of when the RMSE is calculated using all Montecarlo runs with the RMSE where the worst performing 5% have been excluded.

4.8 Sample crosscorrelation coefficient

The sample crosscorrelation coefficient is a way of determining whether two states have a correlation. If two states correlate then the assumption made in section 3.5 for the LMMSE fuser is not correct, which will introduce errors. In Figure 4.10 and 4.11 the results from calculating the coefficient as in equation (2.40) for scenario 2 over 1000 Monte Carlo simulation runs is shown.

In Figure 4.10 the sample coefficient between states in the radar track and the stereo-camera track is shown. It can be seen that coefficient takes on low values for all states and time samples. In Figure 4.11 the sample coefficient between states in the fused track, with feedback, and the stereo-camera track is shown. It can be seen that for one state the crosscorrelation coefficient is high, which means that they are correlated. The values in the first case compares well to the results in [6]. In the article, the authors claim that approximating the crosscorrelation didn't improve their results considerably. This indicates that our assumption is valid and that the LMMSE fuser can be used as presented in equations (2.38) and (2.39). However, in the second case the tracks are not uncorrelated which means that estimation errors are introduced in this method.

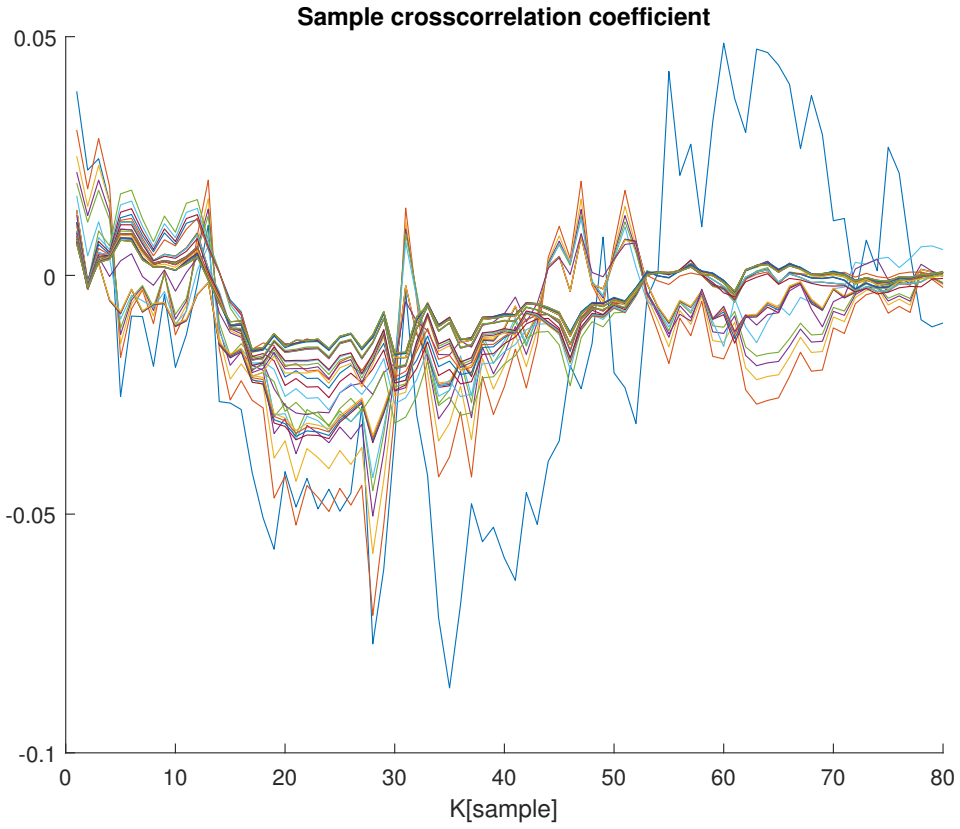


Figure 4.10: The sample crosscorrelation coefficient between the radar track and the stereo-camera track over 1000 Monte Carlo simulation runs for scenario 2. The lines show every combination of the 8 radar states and the 5 camera states.

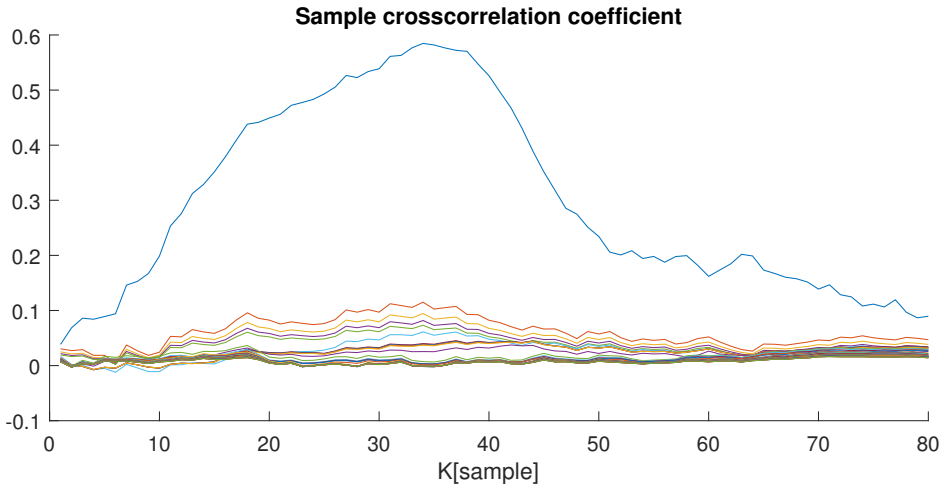


Figure 4.11: The sample crosscorrelation coefficient between the fused state with feedback and the stereo-camera over 1000 Monte Carlo simulation runs for scenario 2. The lines show every combination of the 8 fused states and the 5 camera states. The blue line which reaches above 0.5, represents the crosscorrelation coefficient between the positions on the x-axis, in both states.

5

Conclusion

The track to track fusion shows very promising results. Both the implementation with and the one without feedback performs better than the pure radar or camera tracks. However, the track to track fusion with feedback to the radar tracker is to be preferred if the system can handle it.

The radar tracker used in this thesis is generally unstable, and techniques like covariance scaling and spawning can be used to mitigate this. This instability of the radar tracker is the largest issue in this thesis. The track to track fusion with feedback implementation handles the instability, giving it a reliable performance. While the track to track fusion without feedback implementation does not handle the instability and thus it too expresses a unstable behaviour. The performance of the track to track fusion without feedback might be better than the one with feedback given a more stable radar tracker.

6

Future works

The error in the stereo-camera is based on a simple model with many tunable parameters. By comparing the performance of the simulation with that of some real world stereo-camera setup, it could improve the validity of the simulation.

The radar tracker should be able to use the same algorithms to track multiple targets. This would be close to a real world application and the results even more interesting.

A major weakness with the radar tracker is that it does not initiate new tracks. This means that if a track diverges it might not be able to recover. If instead detections could initiate new tracks, the diverged one could be removed. This feature is something which is incorporated in all modern tracking systems.

The radar detections have a signal amplitude which isn't used. There are examples in research where it is incorporated in the tracker. This would be an interesting aspect to study.

The track to track fusion is based on the LMMSE fuser, but in research a fuser based on the maximum likelihood has been studied. Implementing this and comparing with the current performance would also be interesting.

The track to track fusion algorithm with feedback is shown to have correlated states between the fused track and stereo-camera track. Investigating how to estimate this correlation or how much the state estimations are affected by this would be interesting and could improve the results.

Bibliography

[1] R. W. Sittler. An optimal data association problem in surveillance theory. *IEEE Transactions on Military Electronics*, 8(2):125–139, April 1964.

[2] Y. Bar-Shalom. Tracking methods in a multitarget environment. *IEEE Transactions on Automatic Control*, 23(4):618–626, Aug 1978.

[3] Yaakov Bar-Shalom and William D. Blair. *Multitarget-multisensor tracking: applications and advances : Vol. 3*. Artech House, Norwood, MA, 2000.

[4] Ronald P. S. Mahler. *Statistical multisource-multitarget information fusion*. Artech House, Boston, 2007.

[5] Yaakov Bar-Shalom, Peter K Willett, and Xin Tian. *Tracking and data fusion*. YBS publishing Storrs, CT, USA:, 2011.

[6] T. Yuan, Y. Bar-Shalom, and X. Tian. Heterogeneous track-to-track fusion. In *14th International Conference on Information Fusion*, pages 1–8, July 2011.

[7] Simo Särkkä. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, 2013.

[8] Fredrik Gustafsson. *Statistical sensor fusion*. Studentlitteratur, Lund, 2. edition, 2012.

[9] Karl Granstrom, Marcus Baum, and Stephan Reuter. Extended object tracking: Introduction, overview and applications. *arXiv preprint arXiv:1604.00970*, 2016.

[10] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

[11] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[12] L. Hammarstrand, M. Lundgren, and L. Svensson. Adaptive radar sensor model for tracking structured extended objects. *IEEE Transactions on Aerospace and*

Electronic Systems, 48(3):1975–1995, JULY 2012.

[13] Lars Hammarstrand, Lennart Svensson, Fredrik Sandblom, and Joakim Sorstedt. Extended object tracking using a radar resolution model. *IEEE Transactions on Aerospace and Electronic Systems*, 48(3):2371–2386, 2012.