Linköping studies in science and technology. Dissertations.
No. 1409

Sensor Fusion for Automotive Applications

Christian Lundquist

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

Linköping 2011

Cover illustration: The intensity map describes the density of stationary targets along the road edges. A photo of the the driver's view in the green car is shown in Figure 5a on Page 216.

Linköping studies in science and technology. Dissertations.
No. 1409

Sensor Fusion for Automotive Applications

Christian Lundquist

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

ISBN 978-91-7393-023-9 ISSN 0345-7524

Printed by LiU-Tryck, Linköping, Sweden 2011

To Nadia

Abstract

Mapping stationary objects and tracking moving targets are essential for many autonomous functions in vehicles. In order to compute the map and track estimates, sensor measurements from radar, laser and camera are used together with the standard proprioceptive sensors present in a car. By fusing information from different types of sensors, the accuracy and robustness of the estimates can be increased.

Different types of maps are discussed and compared in the thesis. In particular, road maps make use of the fact that roads are highly structured, which allows relatively simple and powerful models to be employed. It is shown how the information of the lane markings, obtained by a front looking camera, can be fused with inertial measurement of the vehicle motion and radar measurements of vehicles ahead to compute a more accurate and robust road geometry estimate. Further, it is shown how radar measurements of stationary targets can be used to estimate the road edges, modeled as polynomials and tracked as extended targets.

Recent advances in the field of multiple target tracking lead to the use of finite set statistics (FISST) in a set theoretic approach, where the targets and the measurements are treated as random finite sets (RFS). The first order moment of a RFS is called probability hypothesis density (PHD), and it is propagated in time with a PHD filter. In this thesis, the PHD filter is applied to radar data for constructing a parsimonious representation of the map of the stationary objects around the vehicle. Two original contributions, which exploit the inherent structure in the map, are proposed. A data clustering algorithm is suggested to structure the description of the prior and considerably improving the update in the PHD filter. Improvements in the merging step further simplify the map representation.

When it comes to tracking moving targets, the focus of this thesis is on extended targets, i.e., targets which potentially may give rise to more than one measurement per time step. An implementation of the PHD filter, which was proposed to handle data obtained from extended targets, is presented. An approximation is proposed in order to limit the number of hypotheses. Further, a framework to track the size and shape of a target is introduced. The method is based on measurement generating points on the surface of the target, which are modeled by an RFS.

Finally, an efficient and novel Bayesian method is proposed for approximating the tire radii of a vehicle based on particle filters and the marginalization concept. This is done under the assumption that a change in the tire radius is caused by a change in tire pressure, thus obtaining an indirect tire pressure monitoring system.

The approaches presented in this thesis have all been evaluated on real data from both freeways and rural roads in Sweden.

Populärvetenskaplig sammanfattning

Dagens bilar blir säkrare för varje år. Tidigare var det den passiva säkerheten med bilbälten, krockkuddar och krockzoner som förbättrades. Nu sker de snabbaste förändringarna inom aktiv säkerhet, med förarstödsystem såsom antisladd och nödbromssystem och förarvarningssystem för farliga filbyten och framförvarande hinder.

Förarstödsystem kräver omvärldsuppfattning, och de sensorer för detta som finns i bruk idag är kamera och radar. Kameran ser t.ex. filmarkeringar och kan användas för att detektera fordon och fotgängare. Radarn är mycket bra på att detektera rörliga objekt och på avståndsbedömning, och används idag för att följa bilen framför. Radarn ger en mängd information som idag är outnyttjad, och ett syfte med avhandlingen är att undersöka hur radarinformationen kan användas fullt ut i framtida säkerhetssystem.

Ett bidrag i avhandlingen är att använda radarns mätningar av stillastående objekt på och jämte vägen för att bygga upp en lokal karta. Kartan visar områden som är körbara och hinder som ska undvikas. Ett annat bidrag är att sortera alla de träffar från rörliga objekt som radarn registrerar och ge en noggrann karta över hur många andra fordon det finns, samt deras positioner, hastigheter och storlekar. Tillsammans bildar dessa två kartor en lägesbild som kan användas i nästa generations kollisionsundvikande system som kan kombinera broms och styringrepp för att på ett intelligent, säkert och kontrollerat sätt väja undan i kritiska nödsituationer.

En annan typ av säkerhetssystem är varningssystem till föraren på förändringar i fordonsdynamiska parametrar som kan försämra köregenskaperna. Exempel på sådana parametrar är däcktryck, friktion och fellastning. Ett bidrag i avhandlingen är ett nytt sätt att skatta en sänkning i däcktryck, genom att kombinera sensorer i fordonet med satellitnavigering.

De metoder som presenteras i avhandlingen har utvärderats på verkliga data från bland annat motorvägar och landsvägar i Sverige.

Acknowledgments

It all started in the end of the 80s, when my teacher asked me what I wanted to become when I grow up. I had only one wish, to study at Chalmers; followed by my answer my teacher laughed and told me “you will never make it, you are too stupid". This incentivized me, and encourage by other teachers I managed to finish school and be admitted at Chalmers. Several years later I’m now here and I have finished my thesis, something I would never have managed alone, without the support of many wonderful persons.

I would like to thank my supervisor professor Fredrik Gustafsson for his positive, inspiring, encouraging and relaxed attitude and my co-supervisor Dr. Thomas B. Schön for giving me a speedy start into academic research. Whenever I run into problems Dr. Umut Orguner, who possesses an enormous amount of knowledge about everything beyond least squares, pushed me in the right direction. Thank you for your help and for always having time. Further, I would like to acknowledge professor Svante Gunnarsson, who is a skillful head of the group, and his predecessor professor Lennart Ljung. They have created a wonderful research atmosphere which makes enjoyable going to office.

Part of the work has been performed with colleagues. I would like to mention Lic. Karl Granstöm, since one can’t find a better partner to collaborate with. Further I like to mention Dr. Lars Hammarstrand, at Chalmers, with whom I had very useful and interesting discussions. Finally, I would like to thank Dr. Emre Özkan, who opened the Gaussian window and let me see other distributions.

Andreas Andersson at Nira Dynamics AB and Dr. Andreas Eidehall at Volvo Personvagnar AB have rescued me from the simulation swamp, by supporting me with measurement data from prototype vehicles. Dr. Gustaf Hendeby and Dr. Henrik Tidefelt helped me with latex issues, without their support this thesis would not be in this professional shape. Ulla Salaneck, Åsa Karmelind and Ninna Stensgård have helped with many practical things during the years in the group.

An important part of the PhD studies is the time spent outside the office bunker, i.e., in the *fika* room and in pubs. I would like to thank Lic. Jonas Callmer, Lic. Karl Granström and Lic. Martin Skoglund for sharing *vagnen* with me. Further I would like to thank Lic. Christian Lyzell, Dr. Ragnar Wallin, Dr. Henrik Ohlsson Lic. Zoran Sjanic and Sina Khoshfetrat Pakazad for being generous and good friends.

Dr. Wolfgang Reinelt became a mentor for me, he supported and encouraged me during the time at ZF Lenksysteme GmbH. With him I wrote my first publications. My former boss Gerd Reimann taught me about vehicle dynamics and the importance of good experiments; I will always remember *Sinuslenken*.

One of the most inspiring persons I met is Dr. Peter Bunus. Together with professor Fredrik Gustafsson, Dr. David Törnqvist, Lic. Per Skoglar and Lic. Jonas Callmer we started SenionLab AB. I’m looking forward to keeping on working with all of them in the near future.

I would like to acknowledge the supported from the SEnsor fusion for Safety (SEFS) project within the Intelligent Vehicle Safety Systems (IVSS) program and the support from the Swedish Research Council under the frame project grant Extended Target Tracking.

I would never have been able to fulfill my wish to study without the support from my family, for this I am endlessly thankful. Finally, I would like to thank *amore della mia vita* Nadia, who has brought so much love and joy into my life.

Linköping, October 2011
Christian Lundquist

Contents

II Publications

Notation

Operators

Notation	Meaning
min	minimize
max	maximize
$\arg\min_{x} A$	the value x that minimizes A
$\arg\max_{x} A$	the value x that maximizes A
E	expectation
Cov	covariance
Var	variance
$\mathrm{diag}(a, b)$	diagonal matrix with elements a and b
$\mathrm{blkdiag}(A, B)$	Block diagonal matrix with matrices A and B
$\Pr(A)$	probability of event A
A^{T}	transpose of matrix A
Std	standard deviation
$\mathrm{Tr}(A)$	trace of matrix A
$\lvert x \rvert$	absolute value of x (or cardinality if x is a set)
$\hat{x}$	estimate of the stochastic variable x
$\triangleq$	equal by definition
$\sim$	is distributed according to
$\propto$	proportional to
$\in$	belongs to
$\dot{x}$	time derivative of x

Probability theory and Distributions

Notation	Meaning
iW	inverse Wishart distribution
D	intesity or PHD function
$\mathcal{N}$	normal or Gaussian distribution
NiW	normal inverse Wishart distribution
Pois	Poisson ditribution
St	student-t distribution
$\mathcal{U}$	uniform distribution
$p(x)$	density function of x
$p(x, y)$	joint density function of x and y
$p(x\|y)$	conditional density function of x given y

Sets

Notation	Meaning
$\mathbb{N}$	set of natural numbers
$\mathbf{M}$	set of map variables or features
$\mathbb{R}$	set of real numbers
$\mathbf{S}$	set of measurement generating points
$\mathbf{X}$	set of state variables
$\mathbf{Z}$	set of measurements
$N_{\mathbf{x}}$	number of elements in the set $\mathbf{X}$
$\emptyset$	empty set

Random Variables

Notation	Meaning
$\mathbf{x}$	state
$\mathbf{y}$	measurement (no data association needed)
$\mathbf{z}$	measurement (data association needed)
$\mathbf{u}$	input
P	state covariance
$\mathbf{w}$	process noise
Q	process noise covariance
$\mathbf{e}$	measurement noise
R	measurement noise covariance
$\mathbf{m}$	map state
$\mathbf{p}$	point feature
$\mathbf{s}$	state of a measurement generating point
$\boldsymbol{\theta}$	parameter
n_x	dimension of the vector $\mathbf{x}$
$\hat{\mathbf{x}}_{k\|\kappa}$	estimate of $\mathbf{x}$ at time given measurements $\mathbf{y}_{1:\kappa}$

Geometry and dynamics

Notation	Meaning
c	Cartesian representation
p	polar representation
x	Cartesian position x-coordinate (longitudinal)
y	Cartesian position y-coordinate (lateral)
z	Cartesian position z-coordinate (vertical)
w	width
l	length
a	acceleration
v	velocity
u, v	pixel coordinates
$\boldsymbol{d}$	displacement vector
r	range (between sensor and target)
ψ	heading angle, yaw angle or bearing (angle around z-axis)

Miscellaneous

Notation	Meaning
k	discrete time variable
T	sample time
l_b	wheel base (distance between front and rear axle)
l_t	wheel track (distance between left and right wheel)
c_o	road curvature
c_1	road curvature derivative

Abbreviations

Abbreviation	Meaning
ABS	antilock braking system
ACC	adaptive cruise control
C2C	car to car communication
CAN	controller area network
EIV	errors in variables
EIO	errors in output
EKF	extended Kalman filter
ELA	emergency lane assist
ET-GM-PHD	extended target Gaussian mixture probability hypothesis density
FISST	finite set statistics
FMCW	frequency modulated continuous-wave (a radar system)
GM-PHD	gaussian mixture probability hypothesis density
GNN	global nearest neighbor (data association)
GPS	global positioning system
IMU	inertial measurement unit
KF	Kalman filter
LKA	lane keeping assistance
LS	least Squares
MAE	mean absolute error
MAP	maximum a posteriori
MC	Monte Carlo (experiments based on repeated random sampling)
MGP	measurement generating point (reflection point on target)
ML	maximum likelihood
MPF	marginalized particle filter
NN	nearest neighbor (data association)
OGM	occupancy grid map
OLR	optical lane recognition
OSPA	optimal subpattern assignment (multitarget evaluation measure)
PDF	probability density function
PHD	probability hypothesis density
PF	particle Filter
RFS	random finite set
RMSE	root mean square error
SLAM	simultaneous localization and mapping
s.t.	subject to
UKF	unscented Kalman filter
virt	virtual sensor measurement
WLS	weighted least squares

Part I

Background

1 Introduction

This thesis is concerned with the problem of estimating the motion of a vehicle and the characteristics of its surroundings. More specifically, the description of the ego vehicle's surroundings consists of other vehicles and stationary objects as well as the geometry of the road. The signals from several different sensors, including camera, radar and inertial sensor, must be combined and analyzed to compute estimates of various quantities and to detect and classify many objects simultaneously. Sensor fusion allows the system to obtain information that is better than if it was obtained by individual sensors.

Situation awareness is the perception of environmental features, the comprehension of their meaning and the prediction of their status in the near future. It involves being aware of what is happening in and around the vehicle to understand how the subsystems impact on each other.

Sensor fusion is introduced in Section 1.1 and its application within the automotive community is briefly discussed in Section 1.2. The study presented in this thesis was conducted within two Swedish research project, briefly described in Section 1.3 and 1.4. The sensor fusion framework and its components, such as infrastructure, estimation algorithms and various mathematical models, are all introduced in Section 1.5. Finally, the chapter is concluded with an overview of the author's publications in Section 1.6, a statement of the contributions in Section 1.7 and the outline of this thesis given in Section 1.8.

1.1 Sensor Fusion

Sensor fusion is the process of using information from several different sensors to compute an estimate of the state of a dynamic system. The resulting estimate is

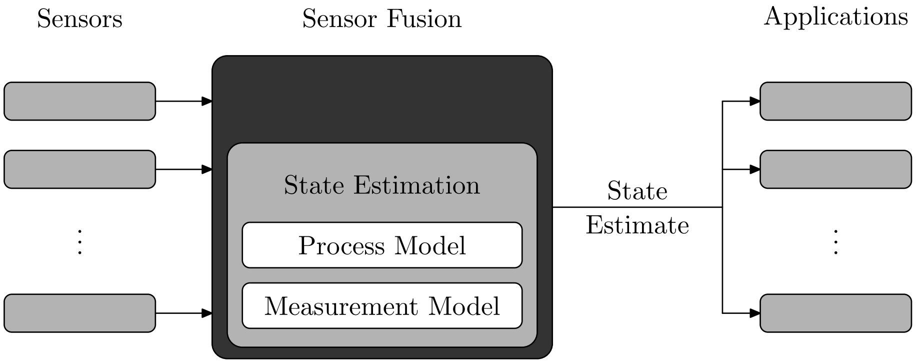


Figure 1.1: The main components of the sensor fusion framework are shown in the middle box. The framework receives measurements from several sensors, fuses them and produces one state estimate, which can be used by several applications.

in some sense better than it would be if the sensors were used individually. The term better can in this case mean more accurate, more reliable, more available and of higher safety integrity. Furthermore, the resulting estimate may in some cases only be possible to obtain by using data from different types of sensors. Figure 1.1 illustrates the basic concept of the sensor fusion framework. Many systems have traditionally been stand alone systems with one or several sensors transmitting information to only one single application. Using a sensor fusion approach it might be possible to remove one sensor and still perform the same tasks, or add new applications without the need to add new sensors.

Sensor fusion is required to reduce cost, system complexity and the number of components involved and to increase accuracy and confidence of sensing.

1.2 Automotive Sensor Fusion

Within the automotive industry there is currently a huge interest in active safety systems. External sensors are increasingly important and typical examples used in this work are radar sensors and camera systems. Today, a sensor is usually connected to a single function. However, all active safety functions need information about the state of the ego vehicle and its surroundings, such as the lane geometry and the position of other vehicles. The use of signal processing and sensor fusion to replace redundant and costly sensors with software attracted recent attention in the IEEE Signal Processing Magazine (Gustafsson, 2009).

The sensors in a modern passenger car can be divided into a number of subgroups; there are internal sensors measuring the motion of the vehicle, external sensors measuring the objects surrounding the vehicle and there are sensors communicating with other vehicles and with the infrastructure. The communication between sensors, fusion framework, actuators and controllers is made possible by the controller area network (CAN). It is a serial bus communication protocol developed by Bosch in the early 1980s and presented by Kiencke et al. (1986) at the SAE international congress in Detroit. An overview of the CAN bus, which has become the *de facto* standard for automotive communication, is given in Johansson et al. (2005).

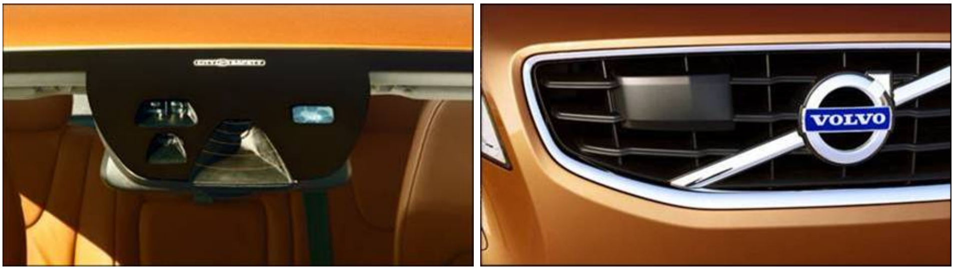
Internal sensors are often referred to as proprioceptive sensors in the literature. Typical examples are gyroscopes, primarily measuring the yaw rate about the vehicle's vertical axis, and accelerometers, measuring the longitudinal and lateral acceleration of the vehicle. The velocity of the vehicle is measured using inductive wheel speed sensors and the steering wheel position is measured using an angle sensor. External sensors are referred to as exteroceptive sensors in the literature, typical examples are radar (RAdio Detection And Ranging), lidar (LIght Detection And Ranging) and cameras.

An example of how a radar and a camera may be mounted in a passenger car is illustrated in Figure 1.2. These two sensors complement each other very well, since the advantage of the radar is the disadvantage of the camera and vice versa. A summary of the two sensors' properties is presented in Table 1.1 and in e.g., Jansson (2005).

As already mentioned, the topic of this thesis is how to estimate the state variables describing the ego vehicle's motion and the characteristics of its surroundings. The ego vehicle is one subsystem, labeled (E) in this work. The use of data from the vehicle's actuators, e.g., the transmission and the steering wheel, to estimate a change in position over time is referred to as odometry. The ego vehicle's surroundings consists of other vehicles, referred to as targets (T), and stationary objects as well as the shape and the geometry of the road (R). Mapping is the problem of integrating the information obtained by the sensors into a given representation, denoted (M), see Adams et al. (2007) for a recent overview and Thrun (2002) for an older survey. Simultaneous localization and mapping (SLAM) is an approach used by autonomous vehicles to build a map while at the

Table 1.1: *Properties of radar and camera for object detection*

	Camera	**Radar**
Detects	other vehicles, lane markings, pedestrians	other vehicles, stationary objects
Classifies objects	yes	no
Azimuth angle	high accuracy	medium accuracy
Range	low accuracy	very high accuracy
Range rate	not	very high accuracy
Field of View	wide	narrow
Weather Conditions	sensitive to bad visibility	less sensitive



(a) *(b)*

Figure 1.2: *Figure (a) shows the camera and Figure (b) the front looking radar in a Volvo S60 production car. Courtesy of Volvo Car Corporation.*

same time keeping track of their current locations, see e.g., Durrant-Whyte and Bailey (2006); Bailey and Durrant-Whyte (2006). This approach is not treated in this thesis.

1.3 Sensor Fusion for Safety

Parts of the work in this thesis have been performed within the research project *Sensor Fusion for Safety* (SEFS), which is funded by the Swedish *Intelligent Vehicle Safety Systems* (IVSS) program. The project is a collaboration between Volvo Technology, Volvo Cars, Volvo Trucks, Mecel, Chalmers University of Technology and Linköping University.

The overall objective of this project is to obtain sensor fusion competence for automotive safety applications in Sweden by doing research within relevant areas. This goal is achieved by developing a sensor fusion platform, algorithms, modeling tools and a simulation platform. More specifically, the aim is to develop general methods and algorithms for a sensor fusion system utilizing information from all available sensors in a modern passenger car. The sensor fusion will provide a refined description of the vehicle's environment that can be used by a number of different safety functions. The integration of the data flow requires new specifications with respect to sensor signals, hardware, processing, architectures and reliability.

The SEFS work scope is divided into a number of work packages. These include at a top level, fusion structure, key scenarios and the development of requirement methods. The next level consists in work packages such as pre-processing and modeling, the implementation of a fusion platform and research done on fusion algorithms, into which this thesis can be classified. The use-case work package consists of implementation of software and design of prototypes and demonstrators. Finally, there is an evaluation and validation work package.

During the runtime of the SEFS project, i.e., from 2005 until 2009, three PhD theses (Schön, 2006; Gunnarsson, 2007; Danielsson, 2010) and three licentiate theses (Bengtsson, 2008; Danielsson, 2008; Lundquist, 2009) have been produced. An overview of the main results in the project is given in Ahrholdt et al. (2009) and the sensor fusion framework is well described in Bengtsson and Danielsson (2008). Furthermore it is worth mentioning some of the publications produced by the project partners. Motion models for tracked vehicles are covered in Svensson and Gunnarsson (2006); Gunnarsson et al. (2006); Sörstedt et al. (2011). A better sensor model of the tracked vehicle is presented in Gunnarsson et al. (2007). Detection of lane departures and lane changes of leading vehicles is studied in Schön et al. (2006), with the goal to increase the accuracy of the road geometry estimate. Computational complexity for systems obtaining data from sensors with different sampling rates and different noise distributions is studied in Schön et al. (2007).

Paper D in this thesis describes a method to estimate and represent stationary objects along the road edges. This publication is based on data collected from the SEFS prototype car and the work was carried out within the project.

1.4 Extended Target Tracking

The final parts of the work in the thesis have been performed within the frame project grant *Extended Target Tracking* (621-2010-4301), founded by the Swedish Research Council. This research project started in 2011 and lasts until 2014. The overall goal is to study new ways of representing targets and their posterior distribution. The work is performed for different types of imagery sensors, such as camera, radar and laser. More specifically the purpose of the project is to study different options to extend the state vector of the target with specific and unusual features, such as physical size and shape, physical properties, colors etc.

So far only a few articles, which relates to the work presented in this thesis, have been published in the project. It is worth mentioning the following publications. An approach to track bicycles from imagery sensor data is proposed by Ardeshiri et al. (2011). It is based on detecting ellipsoids in the images, and treat these pairwise using a dynamic bicycle model. Wahlström et al. (2011) shows theoretically and experimentally that the position and heading of a moving metallic target can be tracked using two magnetometers placed on the ground, for surveillance and traffic monitoring applications.

The Papers E and F in this thesis are produced within the extended target tracking project. In these papers new approaches are proposed to represent and estimate the position and size of targets which, due to their size, might generate more than one measurement per time step.

1.5 Components of the Sensor Fusion Framework

A systematic approach to handle sensor fusion problems is provided by nonlinear state estimation theory. Estimation problems are handled using discrete-time model based methods. The systems discussed in this thesis are primarily dynamic and they are modeled using stochastic difference equations. More specifically, the systems are modeled using the *discrete-time nonlinear state space model*

$$\mathbf{x}_{k+1} = f_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k, \boldsymbol{\theta}_k), \tag{1.1a}$$

$$\mathbf{y}_k = h_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{e}_k, \boldsymbol{\theta}_k), \tag{1.1b}$$

where (1.1a) describes the evolution of the state variable $\mathbf{x}$ over time and (1.1b) explains how the *state variable* $\mathbf{x}$ relates to the *measurement* $\mathbf{y}$[1]. The state vector at time k is denoted by $\mathbf{x}_k \in \mathbb{R}^{n_x}$, with elements $x_1, \dots, x_{n_x}$ being real numbers. Sensor observations collected at time k are denoted by $\mathbf{y}_k \in \mathbb{R}^{n_y}$, with elements $y_1, \dots, y_{n_y}$ being real numbers. The model f_k in (1.1a) is referred to as the *process model*, the *motion model*, the dynamic model or the system model, and it describes how the state propagates in time. The model h_k in (1.1b) is referred to as the *measurement model* or the *sensor model* and it describes how the state is propagated into the measurement space. The random vector $\mathbf{w}_k$ describes the process noise, which models the fact that the actual state dynamics is usually unknown. The random vector $\mathbf{e}_k$ describes the sensor noise. Furthermore, $\mathbf{u}_k$ denotes the deterministic input signals and $\boldsymbol{\theta}_k$ denotes the possibly unknown parameter vector of the model.

The ego vehicle constitutes an important dynamic system in this thesis. The yaw and lateral dynamics are modeled using the so called single track model. This model will be used as an example throughout the thesis. Some of the variables and parameters in the model are introduced in Example 1.1.

1.1 Example: Single Track Ego Vehicle Model

A so called bicycle model is obtained if the wheels at the front and the rear axle of a passenger car are modeled as single wheels. This type of model is also referred to as a single track model and a schematic drawing is given in Figure 1.3. Some examples of typical variables and parameters are:

State variables x: the yaw rate $\dot{\psi}_E$ and the body side slip angle β, i.e.,

$$\mathbf{x} = \begin{bmatrix} \dot{\psi}_E & \beta \end{bmatrix}^{\mathrm{T}}. \tag{1.2}$$

Measurements y: the yaw rate $\dot{\psi}_E$ and the lateral acceleration $\mathsf{a_y}$, i.e.,

$$\mathbf{y} = \begin{bmatrix} \dot{\psi}_E & \mathsf{a_y} \end{bmatrix}^{\mathrm{T}}, \tag{1.3}$$

which both are measured by an inertial measurement unit (IMU).

Input signals u: the steering wheel angle δ_s, which is measured with an angular sensor at the steering column and the velocity $\dot{\mathsf{v}}$, which is measured by

[1] Note that the measurement vector is denoted $\mathbf{y}$ when the sensor measures one or several specific quantities, e.g., acceleration and yaw rate from an IMU sensor. However, when the number of sensor measurements (observations) depends on the environmental circumstances the measurement vector is denoted $\mathbf{z}$, e.g., for a radar which observes an unknown number of targets. This usually leads to a data association problem.

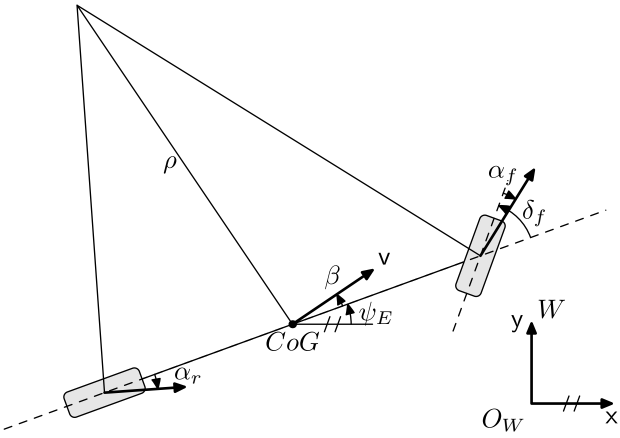


***Figure 1.3:** Illustration of the geometry for the single track model, describing the motion of the ego vehicle. The ego vehicle velocity vector* v *is defined from the center of gravity (CoG) and its angle to the longitudinal axis of the vehicle is denoted by β, referred to as the body side slip angle. Furthermore, the slip angles are referred to as α_f and α_r. The front wheel angle is denoted by δ_f and the current driven radius is denoted by ρ.*

wheel speed sensors, i.e.,

$$\mathbf{u} = \begin{bmatrix} \delta_s & \mathsf{v} \end{bmatrix}^{\mathrm{T}}. \tag{1.4}$$

Parameters $\boldsymbol{\theta}$: the vehicle mass m, which is weighed before the tests, the steering ratio i_s between the steering wheel angle and the front wheels, which has to be estimated in advance, and the tire parameter C_α, which is estimated on-line, since the parameter value changes due to different road and weather conditions.

The nonlinear models f and h are specified in Chapter 2.

The model (1.1) must describe the essential properties of the system, but it must also be simple enough to be efficiently used within a state estimation algorithm. Chapter 3 describes algorithms that are used to compute estimates of the state $\mathbf{x}_k$ and the parameter $\boldsymbol{\theta}_k$ in (1.1).

Before describing the individual steps of the sensor fusion framework another important example is presented in Example 1.2.

1.2 Example: Object Tracking

Other objects, such as vehicles or stationary objects on and along the road, are tracked using measurements from a radar mounted in the ego vehicle. A simple model for one such tracked object is given by using the following variables:

State variables x: Cartesian position of tracked targets $i = 1, \ldots, N_{\mathbf{x}}$ in a world fixed coordinate frame W, i.e., $\mathbf{x}^{(i)} = \begin{bmatrix} \mathrm{x}^W & \mathrm{y}^W \end{bmatrix}^{\mathrm{T}}$.

Measurements z: Range and azimuth angle to objects $m = 1, \ldots, N_{\mathbf{z}}$ measured by the radar in the ego vehicle fixed coordinate frame E, i.e., $\mathbf{z}^{(m)} = \begin{bmatrix} r^E & \psi \end{bmatrix}^{\mathrm{T}}$.

At every time step k, $N_{\mathbf{z}}$ observations are obtained by the radar. Hence, the radar delivers $N_{\mathbf{z}}$ range and azimuth measurements in a multi-sensor set $\mathbf{Z} = \left\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(N_{\mathbf{z}})}\right\}$ to the sensor fusion framework. The sensor fusion framework currently also tracks $N_{\mathbf{x}}$ targets. The multi-target state is given by the set $\mathbf{X} = \left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N_{\mathbf{x}})}\right\}$ where $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N_{\mathbf{x}})}$ are the individual states.

Obviously, the total number of state variables in the present example is $2N_{\mathbf{x}}$ and the total number of measurements is $2N_{\mathbf{z}}$. This issue may be compared to Example 1.1, where the size of the $\mathbf{y}$-vector corresponds to the total number of measurements at time k. Typically, the radar also observes false detections, referred to as clutter, or receives several measurements from the same target, i.e., $N_{\mathbf{z}}$ is seldom equal to $N_{\mathbf{x}}$ for radar sensors.

The different steps of a typical sensor fusion algorithm, as the central part of the larger framework, are shown in Figure 1.4. The algorithm is initiated using a prior guess of the state $\mathbf{x}_0$ or, if it is not the first iteration, the state estimate $\hat{\mathbf{x}}_{k-1|k-1}$ from the previous time step $k-1$ is used. New measurements $\mathbf{Z}_k$ are collected from the sensors and *preprocessed* at time k. Model (1.1) is used to *predict* the state estimate $\hat{\mathbf{x}}_{k|k-1}$ and the measurement $\hat{\mathbf{z}}_{k|k-1}$. For Example 1.2 it is necessary to *associate* the radar observations $\mathbf{Z}_k$ with the predicted measurements $\widehat{\mathbf{Z}}_{k|k-1}$ of the existing state estimates and to *manage the tracks*, i.e., initiate new states and remove old, invalid states. The data association and track management are further discussed in Section 4.2. Finally, the new measurement $\mathbf{z}_k$ is used to calculate the state estimate $\hat{\mathbf{x}}_{k|k}$ at time k in the so called *measurement update* step. The prediction and measurement update are described in Section 3.2. This algorithm is iterated, $\hat{\mathbf{x}}_{k|k}$ is used to predict $\hat{\mathbf{x}}_{k+1|k}$, new measurements $\mathbf{Z}_{k+1}$ are collected at time $k+1$ and so on. The state estimation theory, as part of the sensor fusion framework, is discussed further in Chapter 3. Note that in Example 1.1, the data association and track management are obviously not needed, since there the data association is assumed fixed. In the example the measurements $\mathbf{y}$ from the pre-processing are fed directly to the measurement update.

1.6 Publications

Published work of the author that are of relevance to this thesis are listed below. The publications are clustered in groups preambled by a short summary. The publications are principally listed in chronological order.

The author has been involved in the development of an active steering system prior to starting his PhD-studies. The active steering system superimposes an

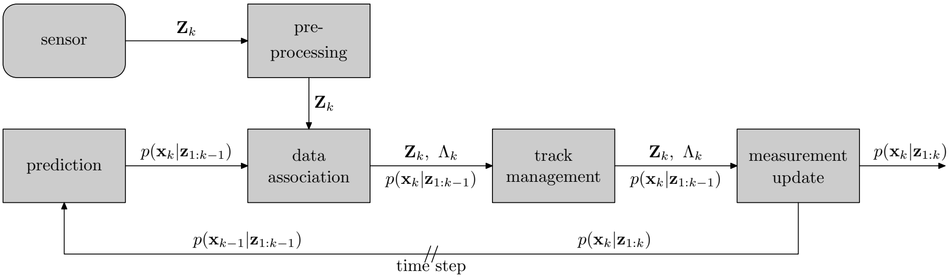

sensor
$\mathbf{Z}_k$
pre-processing
$\mathbf{Z}_k$
prediction
$p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$
data association
$\mathbf{Z}_k,\ \Lambda_k$
$p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$
track management
$\mathbf{Z}_k,\ \Lambda_k$
$p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$
measurement update
$p(\mathbf{x}_k|\mathbf{z}_{1:k})$
$p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$
$p(\mathbf{x}_k|\mathbf{z}_{1:k})$
time step

Figure 1.4: *The new measurements* $\mathbf{Z}_k$ *contain new information and are* associated *to the* predicted *states* $\widehat{\mathbf{X}}_{k|k-1}$ *and thereafter used to* update *them to obtain the improved state estimates* $\widehat{\mathbf{X}}_{k|k}$*.*

electronically controlled angle to the driver's steering input. The aim of the system is to reduce the driver's steering efforts at low velocities, to reduce the vehicle's sensibility at very high velocities and to intervene when the vehicle tends to become unstable. The system is thoroughly described in

> W. Reinelt and C. Lundquist. Mechatronische Lenksysteme: Modellbildung und Funktionalität des Active Front Steering. In R. Isermann, editor, *Fahrdynamik Regelung - Modellbildung, Fahrassistenzsysteme, Mechatronik*, pages 213–236. Vieweg Verlag, September 2006a.
>
> W. Reinelt, W. Klier, G. Reimann, C. Lundquist, W. Schuster, and R. Großheim. Active front steering for passenger cars: System modelling and functions. In *Proceedings of the IFAC Symposium on Advances in Automotive Control*, Salerno, Italy, April 2004.

Sensor fusion based monitoring systems are described in

> W. Reinelt, C. Lundquist, and H. Johansson. On-line sensor monitoring in an active front steering system using extended Kalman filtering. In *Proceedings of the SAE World Congress*, SAE paper 2005-01-1271, Detroit, MI, USA, April 2005.
>
> W. Reinelt and C. Lundquist. Observer based sensor monitoring in an active front steering system using explicit sensor failure modeling. In *Proceedings of the IFAC World Congress*, Prague, Czech Republic, July 2005.
>
> S. Malinen, C. Lundquist, and W. Reinelt. Fault detection of a steering wheel sensor signal in an active front steering system. In *Preprints of the IFAC Symposium on SAFEPROCESS*, pages 547–552, Beijing, China, August 2006.
>
> C. Lundquist and W. Reinelt. Electric motor rotor position monitoring method for electrically aided steering system e.g. steer by wire,

for motor vehicle, involves outputting alarm when difference between measurement value and estimated value of motor exceeds threshold. German Patent Application DE 102005016514 October 12, 2006, Priority date April 8, 2006.

W. Reinelt, C. Lundquist, and S. Malinen. Automatic generation of a computer program for monitoring a main program to provide operational safety. German Patent Application DE 102005049657 April 19, 2007, Priority date October 18, 2005.

W. Reinelt, W. Schuster, R. Großheim, and C. Lundquist. Operating method for electronic servo steering system of vehicle, involves presetting steering wheel angle by steering mechanism as measure for desired wheel turning angle for steering wheel of vehicle. German Patent Application DE 102006052092 May 8, 2008, Priority date November 4, 2006.

W. Reinelt, W. Schuster, R. Großheim, and C. Lundquist. Electronic servo steering system operating method for motor vehicle, involves recognizing track misalignment of vehicle when forces differentiate around preset value from each other at preset period of time in magnitude and/or direction. German Patent DE 102006043069 December 3, 2009, Priority date September 14, 2006.

W. Reinelt, W. Schuster, R. Großheim, and C. Lundquist. Motor vehicle's electronically regulated servo steering system operating method, involves comparing actual value of measured value with stored practical value of corresponding measured value. German Patent DE 10 2006 040 443 January 27, 2011, Priority date August 29, 2006.

Sensor fusion based utility functions to improve driving comfort, safety or agility are described in

W. Reinelt, W. Schuster, R. Großheim, and C. Lundquist. Operating method for electronic power-assisted steering system of vehicle, involves overlapping additional angle, which is disabled after re-start of utility function. German Patent Application DE 102006041236 Mars 6, 2008, Priority date September 2, 2006a.

W. Reinelt, W. Schuster, R. Großheim, and C. Lundquist. Operating method for electronic power-assisted steering system of vehicle, involves re-starting utility function, and after re-start of utility function superimposition of additional angle is unlatched. German Patent DE 102006041237 December 3, 2009, Priority date September 2, 2006b.

G. Reimann and C. Lundquist. Method for operating electronically controlled servo steering system of motor vehicle, involves determining steering wheel angle as measure for desired steering handle angle by steering handle for steering wheels of motor vehicle. German

> Patent Application DE 102006053029 May 15, 2008, Priority date November 10, 2006.
>
> C. Lundquist and R. Großheim. Method and device for determining steering angle information. German Patent Application DE 10 2007 000 958 Mai 14, 2009, International Patent Application WO 2009 047 020 April 16, 2009 and European Patent Application EP 2205478 April 16, 2009, Priority date October 2, 2007.

A reverse driving assistant for passenger cars with trailers based on sensor data of the angle between car and trailer is presented in

> C. Lundquist and W. Reinelt. Rückwärtsfahrassistent für PKW mit Aktive Front Steering. In *Proceedings of the AUTOREG (Steuerung und Regelung von Fahrzeugen und Motoren*, VDI Bericht 1931, pages 45–54, Wiesloch, Germany, March 2006b.
>
> C. Lundquist and W. Reinelt. Back driving assistant for passenger cars with trailer. In *Proceedings of the SAE World Congress*, SAE paper 2006-01-0940, Detroit, MI, USA, April 2006a.
>
> W. Reinelt and C. Lundquist. Method for assisting the driver of a motor vehicle with a trailer when reversing. German Patent DE 10 2006 002 294 February 24, 2011, European Patent Application EP 1810913 July 25, 2007 and Japanese Patent Application JP 2007191143 August 2, 2007, Priority date January 18, 2006.

Furthermore, a control system which stabilizes a vehicle trailer combination, based on the same sensor information as the publications listed above, is described in

> C. Lundquist. Method for stabilizing a vehicle combination. U.S. Patent US 8010253 August 30, 2011 and German Patent Application DE 102007008342 August 21, 2008, Priority date February 20, 2007.

The functional safety concept for the active steering system, among others including a sensor fusion based monitoring system, is described in

> W. Reinelt and C. Lundquist. Controllability of active steering system hazards: From standards to driving tests. In Juan R. Pimintel, editor, *Safety Critical Automotive Systems*, pages 173–178. SAE International, 400 Commonwealth Drive, Warrendale, PA, USA, August 2006b.

During his time as a PhD student the author has been involved in the following publications. Starting with road geometry estimation, i.e., describing the curvature of the lane, which is covered in

> C. Lundquist and T. B. Schön. Road geometry estimation and vehicle tracking using a single track model. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 144–149, Eindhoven, The Netherlands, June 2008.

> C. Lundquist and T. B. Schön. Joint ego-motion and road geometry estimation. *Information Fusion*, 12:253–263, October 2011.

Road edge estimation primarily aims to estimate the position and shape of stationary objects next to the road. Two approaches have been studied; road edges modeled as extended targets are described in

> C. Lundquist and T. B. Schön. Estimation of the free space in front of a moving vehicle. In *Proceedings of the SAE World Congress*, SAE paper 2009-01-1288, Detroit, MI, USA, April 2009a.

> C. Lundquist, U. Orguner, and T. B. Schön. Tracking stationary extended objects for road mapping using radar measurements. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 405–410, Xi'an, China, June 2009.

> C. Lundquist, U. Orguner, and F Gustafsson. Estimating polynomial structures from radar data. In *Proceedings of the International Conference on Information Fusion*, Edinburgh, UK, July 2010b.

> C. Lundquist, U. Orguner, and F. Gustafsson. Extended target tracking using polynomials with applications to road-map estimation. *IEEE Transactions on Signal Processing*, 59(1):15–26, January 2011c.

and an approach where the stationary objects along the road are represented by an intensity map is presented in

> C. Lundquist, L. Danielsson, and F. Gustafsson. Random set based road mapping using radar measurements. In *Proceedings of the European Signal Processing Conference*, pages 219–223, Aalborg, Denmark, August 2010a.

> C. Lundquist, L. Hammarstrand, and F. Gustafsson. Road intensity based mapping using radar measurements with a probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 59(4):1397–1408, April 2011b.

A method to on-line estimate the cornering stiffness parameters of the tires is presented in

> C. Lundquist and T. B. Schön. Recursive identification of cornering stiffness parameters for an enhanced single track model. In *Proceedings of the IFAC Symposium on System Identification*, pages 1726–1731, Saint-Malo, France, July 2009b.

An overview of the results from the SEFS project, and specifically an overview about mapping techniques, is given in

> M. Ahrholdt, F. Bengtsson, L. Danielsson, and C. Lundquist. SEFS – results on sensor data fusion system development. In *Proceedings of the World Congress on Intelligent Transportation Systems and Services*, Stockholm, Sweden, September 2009.

C. Lundquist. *Automotive Sensor Fusion for Situation Awareness*. Licentiate Thesis No 1422, Department of Electrical Engineering, Linköping University, Sweden, 2009.

C. Lundquist, T. B. Schön, and F. Gustafsson. Situational awareness and road prediction for trajectory control applications. In A. Eskandarian, editor, *Handbook of Intelligent Vehicles*, chapter 24. Springer, November 2011e.

Extended target tracking with a PHD filter is described in

K. Granström, C. Lundquist, and U Orguner. A Gaussian mixture PHD filter for extended target tracking. In *Proceedings of the International Conference on Information Fusion*, Edinburgh, UK, July 2010.

C. Lundquist, K. Granström, and U. Orguner. Estimating the shape of targets with a PHD filter. In *Proceedings of the International Conference on Information Fusion*, Chicago, IL, USA, July 2011a.

U. Orguner, C. Lundquist, and K. Granström. Extended target tracking with a cardinalized probability hypothesis density filter. In *Proceedings of the International Conference on Information Fusion*, pages 65–72, Chicago, IL, USA, July 2011.

K. Granström, C. Lundquist, and U Orguner. Tracking rectangular and elliptical extended targets using laser measurements. In *Proceedings of the International Conference on Information Fusion*, Chicago, IL, USA, July 2011b.

K. Granström, C. Lundquist, and U. Orguner. Extended target tracking using a Gaussian-mixture PHD filter. *IEEE Transactions on Aerospace and Electronic Systems*, 2011a. Under review.

An approach to use camera data to improve the position estimate of a vehicle is presented in

E. Nilsson, C. Lundquist, T. B. Schön, D. Forslund, and J. Roll. Vehicle motion estimation using an infrared camera. In *Proceedings of the World Congress of the International Federation of Automatic Control*, Milan, Italy, August 2011.

A marginalized particle filter based method to estimate the tire radii, and thereby being able to detect pressure losses is presented in

E. Özkan, C. Lundquist, and F. Gustafsson. A Bayesian approach to jointly estimate tire radii and vehicle trajectory. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, Washington DC, USA, October 2011.

> C. Lundquist, E. Özkan, and F. Gustafsson. Tire radii estimation using a marginalized particle filter. *IEEE Transactions on Intelligent Transportation Systems*, 2011d. Submitted.

Finally, the pedagogic contribution

> C. Lundquist, M. Skoglund, K. Granström, and T. Glad. How peer-review affects student learning. In *Utvecklingskonferens för Sveriges ingenjörsutbildingar*, Norrköping, Sweden, November 2011f.

describes how peer-review can improve student's learning. This work is implemented in a course in sensor fusion.

1.7 Contributions

The main contributions of this thesis are briefly summarized and presented below:

Mapping: An overview of different methods to map stationary objects in the surroundings of the vehicle are summarized in Paper A.

Road curvature estimation: The estimation from camera and radar is improved by using information about the ego vehicle motion. The results are presented in Paper B.

Road edge estimation: Errors in variables methods are applied in order to track the road edges using point measurements, and the results are given in Paper C. The edges are modeled as extended targets.

Intensity based map: An intensity function is used to represent stationary objects along the road. Structures in the road are used to improve the update and representation of the map as described in Paper D.

Extended target tracking: A modification of the GM-PHD filter, which is able to handle extended targets, is presented in Paper E.

Target size estimation: An approach to estimate the size and the shape is introduced in Paper F. The tracking framework contains a hybrid state space where measurement generating points and the measurements are modeled by random finite sets and target states by random vectors.

Tire radii estimation: The marginalized particle filter is applied to the problem of estimating the tire radii of a vehicle. The Bayesian method, presented in Paper G, is efficient and the estimate is more accurate than estimates from comparable methods.

1.8 Thesis Outline

The thesis is divided into two parts. The first part contains an overview of the research area and background theory for the second part, which contains edited versions of published papers.

1.8.1 Outline of Part I

An overview of vehicle, road and target models is given in Chapter 2, as well as some methods to simplify and make the models applicable for automotive applications. The chapter is concerned with the inner part of the model based estimation process i.e., the *process model* and the *measurement model* illustrated by the two white rectangles in Figure 1.1.

Chapter 3 describes estimation and filtering algorithms, which have been applied to derive the results in Part II. The *estimation process* is illustrated by the gray rectangle in Figure 1.1, and the chapter has a tutorial character.

Target tracking is the subject of Chapter 4, and it describes the framework which takes care of incoming measurements. The framework is illustrated by the black rectangle in Figure 1.1.

Chapter 5 describes a filtering framework which propagates the so called PHD, being an approximation of a finite random set of targets. The PHD-filter unifies the track management and the estimation process, i.e., the black and the gray box in Figure 1.1.

Finally, the work is summarized and the next steps for future work are given in Chapter 6.

1.8.2 Outline of Part II

Part II presents a collections of publications where the author of this thesis has been involved. The aim of this section is to give a very brief abstract to the publications, describe how they relate to each other and emphasize the authors contribution.

The first four papers are concerned about estimating and mapping the road and the surroundings of the vehicle.

Paper A,

> C. Lundquist, T. B. Schön, and F. Gustafsson. Situational awareness and road prediction for trajectory control applications. In A. Eskandarian, editor, *Handbook of Intelligent Vehicles*, chapter 24. Springer, November 2011e.

gives an overview of mapping methods for automotive applications. The paper summarizes both standard approaches, of which some have been implemented by the author, and methods developed by the author in other publications. There is a slight overlap of the material presented in the paper with other papers in this thesis. The author has written the major part of this paper, besides the section covering lane tracking using a camera.

Paper B,

> C. Lundquist and T. B. Schön. Joint ego-motion and road geometry estimation. *Information Fusion*, 12:253–263, October 2011.

presents early work of the author. It is shown how the road curvature estimate can be improved and made more robust by fusing lane marking information with the position of other vehicles on the road and knowledge about the ego vehicles motion. The lane information is obtained by a camera, the other vehicles are tracked by a radar and the ego vehicle's motion is measured with e.g., an IMU. The authors contribution is the model which connects the ego vehicle's motion to the estimate of the curvature.

Paper C,

> C. Lundquist, U. Orguner, and F. Gustafsson. Extended target tracking using polynomials with applications to road-map estimation. *IEEE Transactions on Signal Processing*, 59(1):15–26, January 2011c.

describes a method to track the road edges as extended targets. This is the first article in a series where the author is dealing with extended targets. In this paper the targets are represented by polynomials, which represent e.g., guardrails along the road. The presented ideas are primarily the author's contribution, however, Dr. Umut Orguner has assisted with his knowledge in the area of target tracking to make it possible to implement the ideas.

Paper D,

> C. Lundquist, L. Hammarstrand, and F. Gustafsson. Road intensity based mapping using radar measurements with a probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 59(4):1397–1408, April 2011b.

is the last of the four papers describing models and algorithms to represent and estimate the road. In this paper objects along road edges are represented by an intensity function. The intensity in each position of the map describes the density of targets in that position. Dr. Lars Hammarstrand came up with the initial ideas to this paper, the realization of the idea was formed in discussion with the author. The modifications to the standard PHD filter were implemented by the author, who also wrote the major part of the paper.

The next two papers treat methods to track moving targets, e.g., vehicles, which have an extent.

Paper E,

> K. Granström, C. Lundquist, and U. Orguner. Extended target tracking using a Gaussian-mixture PHD filter. *IEEE Transactions on Aerospace and Electronic Systems*, 2011a. Under review.

introduces a PHD filter based method to track targets which potentially might give rise to more than one measurement. The authors came up with the idea to this paper at a conference where the mathematical framework was presented.

The contribution of this paper is the implementation, including the necessary assumptions, which must be made to make the theory computationally tractable. All authors of this paper have contributed to the presented approach, however the major part of the implementation and the writing was made by Karl Granström.

Paper F,

> C. Lundquist, K. Granström, and U. Orguner. Estimating the shape of targets with a PHD filter. In *Proceedings of the International Conference on Information Fusion*, Chicago, IL, USA, July 2011a.

presents a new approach to track the size and the shape of a target. Since the output of many off-the-shelf sensors are point observations, a model must be found to relate point measurements to a shape. In this paper a random finite set describes points on the target's surface, which are assumed to have generated the measurements. The idea is mainly developed by the author, also the implementation and writing were primarily performed by the author. However, the results were improved through the valuable advice regarding target tracking given by Dr. Umut Orguner.

The last paper is concerned about the ego vehicle only.

Paper G,

> C. Lundquist, E. Özkan, and F. Gustafsson. Tire radii estimation using a marginalized particle filter. *IEEE Transactions on Intelligent Transportation Systems*, 2011d. Submitted.

is based on theories recently developed by among others the two co-authors Dr. Emre Özkan and Professor Fredrik Gustafsson. In this paper these theories are applied to the problem of estimating the tire radii of a vehicle. The primary aim is to detect tire pressure losses. The author has contributed by implementing these theories, by applying them on real data and by writing major part of the paper. Dr. Emre Özkan has helped to implement the method based on his knowledge in the area.

2 Models of Dynamic Systems

Given measurements from several sensors the objective is to estimate one or several state variables, either by means of improving a measured signal or by means of estimating a signal which is not, or can not, be directly measured. In either case the relationship between the measured signals and the state variable must be described, and the equations describing this relationship is referred to as the *measurement model*. When dealing with dynamic systems, as is commonly the case in automotive applications, the objective might be to predict the value of the state variable at the next time step. The prediction equation is referred to as the *process model*. This section deals with these two types of models.

As mentioned in the introduction in Section 1.5, a general model of dynamic systems is provided by the nonlinear state space model

$$\mathbf{x}_{k+1} = f_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k, \boldsymbol{\theta}_k), \tag{2.1a}$$

$$\mathbf{y}_k = h_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{e}_k, \boldsymbol{\theta}_k). \tag{2.1b}$$

Most mechanical and physical laws are provided in continuous-time, but computer implementations are made in discrete-time, i.e., the process and measurement models are derived in continuous-time t according to

$$\dot{\mathbf{x}}(t) = a(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), \boldsymbol{\theta}(t), t), \tag{2.2a}$$

$$\mathbf{y}(t) = c(\mathbf{x}(t), \mathbf{u}(t), \mathbf{e}(t), \boldsymbol{\theta}(t), t), \tag{2.2b}$$

and are then discretized. Discretization is the topic of Section 2.2. Special cases of the general state space model (2.1), such as the state space model with additive noise and the linear state space model, are discussed in Section 2.3 and 2.4, respectively.

Several models for various applications are given in the papers in Part II. This chapter therefore begins with an introduction to some of these models in Section 2.1.

2.1 Overview of the Models Used in the Thesis

The single track model was introduced in Example 1.1 and it used as an example throughout the first three chapters of this thesis. Furthermore, the model is a vital part for deriving the results in Paper B. The process and measurement models of the single track model are given in Example 2.1.

2.1 Example: Single Track Model

The state variables **x**, the input signals **u** and the measurement signals **y** of the ego vehicle model were defined in Example 1.1, and are repeated here for convenience

$$\mathbf{x} = \begin{bmatrix} \dot{\psi}_E & \beta \end{bmatrix}^{\mathrm{T}}, \tag{2.3a}$$

$$\mathbf{u} = \begin{bmatrix} \delta_f & \mathsf{v} \end{bmatrix}^{\mathrm{T}}, \tag{2.3b}$$

$$\mathbf{y} = \begin{bmatrix} \dot{\psi}_E & \mathsf{a}_\mathsf{y} \end{bmatrix}^{\mathrm{T}}. \tag{2.3c}$$

Note that the front wheel angle δ_f is used directly as an input signal to simplify the example. The continuous-time single track process and measurement models are given by

$$\dot{\mathbf{x}} = \begin{bmatrix} -\frac{C_{\alpha f} l_f^2 \cos\delta_f + C_{\alpha r} l_r^2}{I_{zz}\mathsf{v}} \dot{\psi}_E + \frac{-C_{\alpha f} l_f \cos\delta_f + C_{\alpha r} l_r}{I_{zz}} \beta + \frac{C_{\alpha f} l_f \tan\delta_f}{I_{zz}} \\ -\left(1 + \frac{C_{\alpha f} l_f \cos\delta_f - C_{\alpha r} l_r}{\mathsf{v}^2 m}\right) \dot{\psi}_E - \frac{C_{\alpha f}\cos\delta_f + C_{\alpha r}}{m\mathsf{v}} \beta + \frac{C_{\alpha f}\sin\delta_f}{m\mathsf{v}} \end{bmatrix}, \tag{2.4a}$$

$$\mathbf{y} = \begin{bmatrix} \dot{\psi}_E \\ \frac{-C_{\alpha f} l_f \cos\delta_f + C_{\alpha r} l_r}{m\mathsf{v}} \dot{\psi}_E - \frac{C_{\alpha f}\cos\delta_f + C_{\alpha r}}{m} \beta + \frac{C_{\alpha f}\sin\delta_f}{m} \end{bmatrix}, \tag{2.4b}$$

with parameter vector

$$\boldsymbol{\theta} = \begin{bmatrix} l_f & l_r & I_{zz} & m & C_{\alpha f} & C_{\alpha r} \end{bmatrix}, \tag{2.5}$$

where l_f and l_r denotes the distances between the center of gravity of the vehicle and the front and rear axles, respectively. Furthermore, m denotes the mass of the vehicle and I_{zz} denotes the moment of inertia of the vehicle about its vertical axis in the center of gravity. The parameters $C_{\alpha f}$ and $C_{\alpha f}$ are called cornering stiffness and describe the road tire interaction. Typical values for the parameters are given in Table 2.1. The model is derived in many vehicle dynamics books, e.g., Mitschke and Wallentowitz (2004); Wong (2001); Gillespie (1992); Rajamani (2006).

Road models are treated in Paper A to D. A good overview of available and commonly used road models is given in Paper A, and therefore no introduction to road modeling in general is given in this chapter. However, road models are often combined with ego vehicle models and described in one single state space

***Table 2.1:** Typical ranges for the vehicle parameters used in the single track model.*

m kg	I_{zz} kgm^2	C_α N/rad	$l_f + l_r$ m
1000 – 2500	850 – 5000	45000 – 75000	2.5 – 3.0

model. One approach is presented in Paper B and compared with two other vehicle models with road interaction, which are described as follows.

The first model is described in Example 2.2 and it is commonly used for autonomous driving and lane keeping. This model is well described by e.g., Dickmanns (2007) and Behringer (1997). Note that the ego vehicle's motion is modeled with respect to a road fixed coordinate frame, unlike the single track model in Example 2.1, which is modeled in a Cartesian world coordinate frame.

2.2 Example: Vehicle Model with Road Interaction

The relative angle between the vehicle's longitudinal axis and the tangent of the road is denoted ψ_{RE}. The notation is illustrated in the Figures 1 to 4 in Paper B (Pages 140-144). Ackermann's steering geometry is used to obtain the relation

$$\dot{\psi}_{RE} = \frac{\mathrm{v}}{l_b}\delta_f - \mathrm{v} \cdot c_0, \tag{2.6}$$

where the current curvature of the road c_0 is the inverse of the road's radius. The lateral displacement of the vehicle in the lane is given by

$$\dot{l}_R = \mathrm{v}(\psi_{RE} + \beta). \tag{2.7}$$

A process model for the body side slip angle was given in (2.4a), but since the yaw rate $\dot{\psi}$ is not part of the model in this section, equation (2.4a) has to be rewritten according to

$$\dot{\beta} = -\frac{C_{\alpha f}\cos\delta_f + C_{\alpha r}}{m\mathrm{v}}\beta - \left(1 + \frac{C_{\alpha f} l_f \cos\delta_f - C_{\alpha r} l_r}{\mathrm{v}^2 m}\right)\frac{v_x}{l_b}\tan\delta_f + \frac{C_{\alpha f}}{m\mathrm{v}}\sin\delta_f, \tag{2.8}$$

which is further simplified by assuming small angles, to obtain a linear model according to

$$\dot{\beta} = -\frac{C_{\alpha f} + C_{\alpha r}}{m\mathrm{v}}\beta + \left(\frac{C_{\alpha f}}{m\mathrm{v}} - \frac{\mathrm{v}}{l_b}\right)\delta_f. \tag{2.9}$$

The steering wheel angle might have a bias, for example if the sensor is not calibrated, which leads to an accumulation of the side slip angle β in (2.9). Other reasons for a steering wheel angle bias is track torsion or strong side wind, which the driver compensates for with the steering wheel. The problem is solved by introducing an offset to the front wheel angel as a state variable according to

$$\delta_f^m = \delta_f + \delta_f^{\text{offs}}. \tag{2.10}$$

Note that the curvature c_0 is included in (2.6). The curvature c_0 is usually treated as a state variable and in this example the vehicle motion model is augmented with the simple clothoid road model, see Paper A.

To summarize, the state variable vector is defined as

$$\mathbf{x} = \begin{bmatrix} \psi_{RE} \\ l_R \\ \beta \\ \delta_f \\ \delta_f^{\text{offs}} \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} \text{relative angle between vehicle and road} \\ \text{lateral displacement of vehicle in lane} \\ \text{vehicle body side slip angle} \\ \text{front wheel angle} \\ \text{front wheel angle bias offset} \\ \text{road curvature at the ego vehicle} \\ \text{curvature derivative} \end{bmatrix} \tag{2.11}$$

and the process model is given by

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\psi}_{RE} \\ \dot{l}_R \\ \dot{\beta} \\ \dot{\delta}_f \\ \dot{\delta}_f^{\text{offs}} \end{bmatrix} = \begin{bmatrix} \frac{v_x}{l_b}\delta_f - \mathsf{v}c_0 \\ \mathsf{v}(\psi_{RE} + \beta) \\ -\frac{C_{\alpha f}+C_{\alpha r}}{m\mathsf{v}}\beta + \left(\frac{C_{\alpha f}}{m\mathsf{v}} - \frac{\mathsf{v}}{l_b}\right)\delta_f \\ w_{\delta_f} \\ 0 \\ \mathsf{v}c_0 \\ w_{c_1} \end{bmatrix}. \tag{2.12}$$

With the camera it is possible mo measure ψ_{RE}, l_R and c_0. The front wheel angle δ_f can be derived from the steering wheel angle δ_s if the steering geometry is known; i.e., the measurement vector is

$$\mathbf{y} = \begin{bmatrix} \psi_{RE} & l_R & c_0 & \delta_s \end{bmatrix}^{\mathrm{T}} \tag{2.13}$$

and the measurement model is trivial. This model is used in the approach referred to as “fusion 3” in Paper B.

Another and simpler vehicle model is obtained if the side slip angle is omitted and the ego vehicle’s yaw rate $\dot{\psi}_E$ is used instead of the steering wheel angle. The model is summarized in Example 2.3, and thoroughly described together with results in Eidehall (2007); Eidehall et al. (2007); Eidehall and Gustafsson (2006); Gern et al. (2000, 2001); Zomotor and Franke (1997).

More advanced vehicle models with more degrees of freedom, including the two track model, are described by Schofield (2008).

2.3 Example: Simplified Vehicle Model with Road Interaction

The state variable vector is here defined as

$$\mathbf{x} = \begin{bmatrix} \psi_{RE} & l_R & \dot{\psi}_E & c_0 & c_1 \end{bmatrix}^{\mathrm{T}}, \tag{2.14}$$

and the process model is simply given by

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\psi}_{RE} \\ \dot{l}_E \\ \dot{c}_0 \\ \dot{c}_1 \end{bmatrix} = \begin{bmatrix} v_x c_0 + \dot{\psi}_E \\ v_x \psi_{RE} \\ w_{\dot{\psi}_{RE}} \\ \mathrm{v} c_0 \\ w_{c_1} \end{bmatrix}. \tag{2.15}$$

The measurement vector is

$$\mathbf{y} = \begin{bmatrix} \psi_{RE} & l_R & c_0 & \dot{\psi}_E \end{bmatrix}^{\mathrm{T}} \tag{2.16}$$

where the yaw rate is measured by the vehicle's IMU. This model is used in the approach referred to as "fusion 2" in Paper B.

In this work, only measurements from the ego vehicle's sensors are available; that is the target's motion is measured using the ego vehicle's radar and camera. This is the reason for why the target model is simpler than the ego vehicle model. The targets play an important role in the sensor fusion framework presented in this work, but little effort has been spent modeling their motion. Instead standard models from target tracking literature are used. A survey of different process models and measurement models are given by Rong Li and Jilkov (2003) and Rong Li and Jilkov (2001), respectively. The subject is also covered in the books by Blackman and Popoli (1999) and Bar-Shalom et al. (2001). One typical target model is given in Example 2.4. This model is used in the Papers F and G, and a somewhat similar model, called constant velocity model, is used in Paper E.

2.4 Example: Coordinated Turn Model

The coordinated turn model is commonly used to model moving targets. The ego vehicle's radar and camera measures the range $r_{T_iE_s}$ and the azimuth angle $\psi_{T_iE_s}$ to target number i as described in the introduction in Example 1.2. The states of the coordinated turn model in polar velocity are given by

$$\mathbf{x} = \begin{bmatrix} \mathrm{x}^W_{T_iW} \\ \mathrm{y}^W_{T_iW} \\ \mathrm{v}^{T_i} \\ \psi_{T_i} \\ \dot{\psi}_{T_i} \end{bmatrix} = \begin{bmatrix} \text{x-position in } W\text{-frame} \\ \text{y-position in } W\text{-frame} \\ \text{longitudinal velocity} \\ \text{heading angle} \\ \text{yaw rate} \end{bmatrix}. \tag{2.17}$$

The process and measurement models are given by

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathrm{x}}_{T_iW}^{W} \\ \dot{\mathrm{y}}_{T_iW}^{W} \\ \dot{\mathrm{v}}^{T_i} \\ \dot{\psi}_{T_i} \\ \ddot{\psi}_{T_i} \end{bmatrix} = \begin{bmatrix} \mathrm{v}^{T_i} \cos \psi_{T_i} \\ \mathrm{v}^{T_i} \sin \psi_{T_i} \\ 0 \\ \dot{\psi}_{T_i} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ w_{\dot{\mathrm{v}}^{T_i}} \\ 0 \\ w_{\ddot{\psi}_{T_i}} \end{bmatrix} \tag{2.18a}$$

$$\mathbf{z} = \begin{bmatrix} r_{T_iE_s} \\ \psi_{T_iE_s} \end{bmatrix} = \begin{bmatrix} \sqrt{\left(\mathrm{x}_{T_iW}^{W} - \mathrm{x}_{EW}^{W} - \mathrm{x}_{E_sE}^{E}\right)^2 + \left(\mathrm{y}_{T_iW}^{W} - \mathrm{y}_{EW}^{W} - \mathrm{y}_{E_sE}^{E}\right)^2} \\ \arctan \frac{\mathrm{y}_{T_iW}^{W}}{\mathrm{x}_{T_iW}^{W}} - \psi_E - \psi_{E_sE} \end{bmatrix} + \mathbf{e} \tag{2.18b}$$

where $(\mathrm{x}_{E_sE}^{E}, \mathrm{y}_{E_sE}^{E}, \psi_{E_sE})$ represents the sensor mounting position and orientation in the ego vehicle coordinate frame E. The ego vehicle's position in the world frame $(\mathrm{x}_{EW}^{W}, \mathrm{y}_{EW}^{W})$ is included in the measurement model of the target. To be able to estimate the ego vehicles position the single track ego vehicle state variable vector and state space model (2.4) has to be augmented with the ego vehicle's position. Note that the ego vehicle's heading angle is denoted ψ_E here to differentiate it from the heading angle of the target.

2.2 Discretizing Continuous-Time Models

The measurements dealt with in this work are sampled and handled as discrete-time variables in computers and electronic control units (ECU). All sensor signals are transferred in sampled form from different sensors to the log-computer using the so called CAN-Bus (Controller Area Network). Hence, the systems discussed in this thesis must also be described using discrete-time models according to the state space model in (2.1). Nevertheless, since physical relations commonly are given in continuous-time, the various systems presented in this thesis, such as the single track model in Example 2.1, are derived and represented using continuous-time state space models in the form (2.2). Thus, all continuous-time models in this thesis have to be discretized in order to describe the measurements. Only a few of the motion models can be discretized exactly by solving the sampling formula

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \int_{k}^{k+T} a(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), \boldsymbol{\theta}(t))dt, \tag{2.19}$$

analytically, where T denotes the sampling time.

2.5 Example: Exact Sampling of the Coordinated Turn Model

Consider the continuous-time coordinate turn model in Example 2.4. The analytic solution of (2.19) is

$$
\begin{aligned}
\mathsf{x}^W_{T_iW,k+1} &= \mathsf{x}^W_{T_iW,k} + \frac{2\mathsf{v}^{T_i}_k}{\dot{\psi}_{T_i,k}} \sin\frac{\dot{\psi}_{T_i,k}T}{2} \cos\left(\psi_{T_i,k} + \frac{\dot{\psi}_{T_i,k}T}{2}\right) && \text{(2.20a)}\\
\mathsf{y}^W_{T_iW,k+1} &= \mathsf{y}^W_{T_iW,k} + \frac{2\mathsf{v}^{T_i}_k}{\dot{\psi}_{T_i,k}} \sin\frac{\dot{\psi}_{T_i,k}T}{2} \sin\left(\psi_{T_i,k} + \frac{\dot{\psi}_{T_i,k}T}{2}\right) && \text{(2.20b)}\\
\mathsf{v}^{T_i}_{k+1} &= \mathsf{v}^{T_i}_k && \text{(2.20c)}\\
\psi_{T_i,k+1} &= \psi_{T_i,k} + T\dot{\psi}_{T_i,k} && \text{(2.20d)}\\
\dot{\psi}_{T_i,k+1} &= \dot{\psi}_{T_i,k}. && \text{(2.20e)}
\end{aligned}
$$

Simpler than using the exact sampling formula (2.19) is it to make use of the standard forward Euler method, which approximates (2.2a) according to

$$\mathbf{x}_{k+1} \approx \mathbf{x}_k + Ta(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k, \boldsymbol{\theta}_k) \triangleq f_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k, \boldsymbol{\theta}_k). \tag{2.21}$$

This is a very rough approximation with many disadvantages, but it is frequently used due to its simplicity. Example 2.6 shows the Euler approximation of the coordinated turn model.

2.6 Example: Euler Sampling of the Coordinated Turn Model

Consider the continuous-time coordinate turn model in Example 2.4. The solution of (2.21) is

$$
\begin{aligned}
\mathsf{x}^W_{T_iW,k+1} &= \mathsf{x}^W_{T_iW,k} + T\mathsf{v}^{T_i}_k \cos\psi_{T_i,k} && \text{(2.22a)}\\
\mathsf{y}^W_{T_iW,k+1} &= \mathsf{y}^W_{T_iW,k} + T\mathsf{v}^{T_i}_k \sin\psi_{T_i,k} && \text{(2.22b)}\\
\mathsf{v}^{T_i}_{k+1} &= \mathsf{v}^{T_i}_k && \text{(2.22c)}\\
\psi_{T_i,k+1} &= \psi_{T_i,k} + T\dot{\psi}_{T_i,k} && \text{(2.22d)}\\
\dot{\psi}_{T_i,k+1} &= \dot{\psi}_{T_i,k}. && \text{(2.22e)}
\end{aligned}
$$

Sampling of linear systems is thoroughly described by Rugh (1996). Moreover, different options to sample and linearize non-linear continuous-time systems are described by Gustafsson (2000). The linearization problem is treated in Chapter 3, in a discussion of approximative model based filters such as the extended Kalman filter.

2.3 Linear State Space Model

An important special case of the general state space model (2.1) is the linear Gaussian state space model, where f and h are linear functions and the noise is Gaussian,

$$\mathbf{x}_{k+1} = F_k(\boldsymbol{\theta})\mathbf{x}_k + G_k^{\mathbf{u}}(\boldsymbol{\theta})\mathbf{u}_k + G_k^{\mathbf{w}}\mathbf{w}_k(\boldsymbol{\theta}), \tag{2.23a}$$

$$\mathbf{y}_k = H_k(\boldsymbol{\theta})\mathbf{x}_k + H_k^{\mathbf{u}}(\boldsymbol{\theta})\mathbf{u}_k + \mathbf{e}_k(\boldsymbol{\theta}), \tag{2.23b}$$

where $\mathbf{w}_k \sim \mathcal{N}(0, Q_k)$ and $\mathbf{e}_k \sim \mathcal{N}(0, R_k)$. Note that the single track model (2.4) is linear in the state variables, as shown in Example 2.7.

2.7 Example: Linearized Single Track Model

The front wheel angle is usually quite small at higher velocities and the assumptions $\cos\delta_f \approx 1$, $\tan\delta_f \approx \sin\delta_f \approx \delta_f$ therefore applies. The continuous-time single track model (2.4) may first be discretized using Euler sampling and can then be written on the linear form (2.23) according to

$$\dot{\mathbf{x}}_{k+1} = \begin{bmatrix} 1 - T\frac{C_{\alpha f}l_f^2 + C_{\alpha r}l_r^2}{I_{zz}\mathrm{v}_t} & T\frac{-C_{\alpha f}l_f + C_{\alpha r}l_r}{I_{zz}} \\ -T - T\frac{C_{\alpha f}l_f - C_{\alpha r}l_r}{\mathrm{v}_k^2 m} & 1 - T\frac{C_{\alpha f} + C_{\alpha r}}{m\mathrm{v}_k} \end{bmatrix}\mathbf{x}_k + \begin{bmatrix} \frac{C_{\alpha f}l_f}{I_{zz}} \\ \frac{C_{\alpha f}}{m\mathrm{v}} \end{bmatrix}\delta_{f,k} + \mathbf{w}_k, \tag{2.24a}$$

$$\mathbf{y}_k = \begin{bmatrix} 1 & 0 \\ \frac{-C_{\alpha f}l_f + C_{\alpha r}l_r}{m\mathrm{v}_k} & -\frac{C_{\alpha f} + C_{\alpha r}}{m} \end{bmatrix}\mathbf{x}_k + \begin{bmatrix} 0 \\ \frac{C_{\alpha f}}{m} \end{bmatrix}\delta_{f,k} + \mathbf{e}_k. \tag{2.24b}$$

The model is linear in the input $\delta_{f,k}$. However, the input v_k is implicitly modeled in the matrices $F_k(\mathrm{v}_k, \boldsymbol{\theta}_k)$, $G_k^{\mathbf{u}}(\mathrm{v}_k, \boldsymbol{\theta}_k)$ and $H_k(\mathrm{v}_k, \boldsymbol{\theta}_k)$.

Linear state space models and linear system theory in general are thoroughly described by Rugh (1996) and Kailath (1980).

2.4 Nonlinear State Space Model with Additive Noise

A special case of the general state space model (2.1) is given by assuming that the noise enters additively and the input signals are subsumed in the time-varying dynamics, which leads to the form

$$\mathbf{x}_{k+1} = f_k(\mathbf{x}_k, \boldsymbol{\theta}_k) + \mathbf{w}_k, \tag{2.25a}$$

$$\mathbf{y}_t = h_k(\mathbf{x}_k, \boldsymbol{\theta}_k) + \mathbf{e}_k. \tag{2.25b}$$

In Example 1.1 an ego vehicle model was introduced, where the steering wheel angle and the vehicle velocity were modeled as deterministic input signals. This consideration can be motivated by claiming that the driver controls the vehicle's lateral movement with the steering wheel and the longitudinal movement with the throttle and brake pedals. Furthermore, the steering wheel angle and the velocity are measured with less noise than the other measurement signals, and they are often pre-processed to improve the accuracy and remove bias. With these arguments the resulting model, given in Example 2.1, may be employed. The model is in some sense simpler than the case when these two signals are assumed to be stochastic measurements, as shown in Example 2.8.

2.8 Example: Single Track Model without Deterministic Input Signals

In classical signal processing it is uncommon to allow deterministic input signals, at least not if these are measured by sensors. The input signals in Example 1.1 should instead be modeled as stochastic measurements. Hence, the measurement vector and the state vector are augmented and the system is remodeled. One example is given by the state space model

$$\begin{bmatrix} \dot{\psi}_{E,k+1} \\ \beta_{k+1} \\ \delta_{f,k+1} \\ \mathsf{v}_{k+1} \end{bmatrix} = \begin{bmatrix} \dot{\psi}_{E,k} + T\left(-\frac{C_{\alpha f} l_f^2 + C_{\alpha r} l_r^2}{I_{zz} \mathsf{v}_k} \dot{\psi}_{E,k} + \frac{-C_{\alpha f} l_f + C_{\alpha r} l_r}{I_{zz}} \beta_k + \frac{C_{\alpha f} l_f \delta_{f,k}}{I_{zz}}\right) \\ \beta_k + T\left(-\left(1 + \frac{C_{\alpha f} l_f - C_{\alpha r} l_r}{\mathsf{v}_k^2 m}\right) \dot{\psi}_{E,k} - \frac{C_{\alpha f} + C_{\alpha r}}{m \mathsf{v}_k} \beta_k + \frac{C_{\alpha f} \delta_{f,k}}{m \mathsf{v}_k}\right) \\ \delta_{f,k} \\ \mathsf{v}_k \end{bmatrix} + \mathbf{w}_k, \tag{2.26a}$$

$$\begin{bmatrix} \dot{\psi}_t \\ \mathsf{a}_{\mathsf{y},k} \\ \delta_{s,k} \\ \mathsf{v}_k \end{bmatrix} = \begin{bmatrix} \dot{\psi}_{E,k} + e_{\dot{\psi},k} \\ \frac{-C_{\alpha f} l_f + C_{\alpha r} l_r}{m \mathsf{v}_k} \dot{\psi}_{E,k} - \frac{C_{\alpha f} + C_{\alpha r}}{m} \beta_k \\ h_{\delta_f}(\dot{\psi}_k, \beta_k, \delta_{f,k}, \boldsymbol{\theta}) \\ \mathsf{v}_{x,t} \end{bmatrix} + \mathbf{e}_k, \tag{2.26b}$$

where T is the sample time. The first two rows of the process and measurement models are the discretized versions of the model given in (2.4). In the second row in the measurement model the simplification $\cos \delta_f \approx 1$ and $\sin \delta_f \approx 0$ is made to not let the measurement model be dependent on the measurement δ_f. The third measurement signal is the steering wheel angle δ_s, but the third state is the front wheel angle δ_f. A possible measurement model h_{δ_f} will be discussed in Example 3.1. Finally, a random walk is assumed for the front wheel angle δ_f and the velocity v in the process model.

Another way to represent the state space model is given by considering the probability density function (PDF) of different signals or state variables of a system. The *transition density* $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ models the dynamics of the system and if the process noise is assumed additive, the transition model is given by

$$p(\mathbf{x}_{k+1}|\mathbf{x}_k) = p_{\mathbf{w}}(\mathbf{x}_{k+1} - f(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\theta}_k)), \tag{2.27}$$

where $p_{\mathbf{w}}$ denotes the PDF of the process noise $\mathbf{w}$. The density describes the state at time $k+1$ given that information about the state at the previous time step k is known. A fundamental property of the process model is the Markov property,

$$p(\mathbf{x}_{k+1}|\mathbf{x}_1, \ldots, \mathbf{x}_k) = p(\mathbf{x}_{k+1}|\mathbf{x}_k). \tag{2.28}$$

This means that the state of the system at time k contains all necessary information about the past, which is needed to predict the future behavior of the system.

Furthermore, if the measurement noise is assumed additive then the *likelihood function*, which describes the measurement model, is given by

$$p(\mathbf{y}_k|\mathbf{x}_k) = p_{\mathbf{e}}(\mathbf{y}_k - h(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\theta}_k)), \tag{2.29}$$

where $p_{\mathbf{e}}$ denotes the PDF of the sensor noise $\mathbf{e}$. The measurement $\mathbf{z}_k$ received by the sensor are described by the likelihood function given that information about the state $\mathbf{x}_k$ at the same time step k is known. The two density functions in (2.27) and (2.29) are often referred to as a hidden Markov model (HMM) according to

$$\mathbf{x}_{k+1} \sim p(\mathbf{x}_{k+1}|\mathbf{x}_k), \tag{2.30a}$$

$$\mathbf{y}_k \sim p(\mathbf{y}_k|\mathbf{x}_k), \tag{2.30b}$$

since $\mathbf{x}_k$ is not directly visible in $\mathbf{y}_k$. It is a statistical model where one Markov process, that represents the system, is observed through another stochastic process, the measurement model.

3 Estimation Theory

This thesis is concerned with estimation problems, i.e. given measurements **y** the aim is to estimate the parameter $\boldsymbol{\theta}$ or the state **x** in (1.1). Both problems rely on the same theoretical basis and the same algorithms can be used. The parameter estimation problem is a part of the system identification process, which also includes the derivation of the model structure, discussed in the previous chapter. The state estimation problem utilizes the model and its parameters to solve for the states. When estimating **x** it is assumed that $\boldsymbol{\theta}$ is known and vice versa. The parameter is estimated in advance if $\boldsymbol{\theta}$ is time invariant or in parallel with the state estimation problem if $\boldsymbol{\theta}$ is assumed to be time varying. Example 3.1 illustrates how the states and parameters may be estimated.

3.1 Example: Parameter and State Estimation

Consider the single track model, which was introduced in Example 1.1 and the equations were given in Example 2.1. The front wheel angle δ_f is considered to be a state variable in Example 2.8 and the steering wheel angle δ_s is treated as a measurement. The measurement equation is in its simplest form a constant ratio given by

$$\delta_{s,k} = h(\delta_{f,k}, \theta) = i_s \delta_{f,k}. \tag{3.1}$$

The parameter $\theta = i_s$ is assumed to be time invariant. The state δ_f must be known in order to identify the parameter θ. Usually the parameter is estimated off-line in advance using a test rig, where the front wheel angle is measured using highly accurate external sensors. The parameter is then used within the model in order to estimate the states on-line while driving.

The tire parameter C_α is assumed to change with weather and road conditions, hence it is a time varying parameter. It has to be identified on-line at time k using the state estimates from the previous time step $k-1$, which in turn were estimated using the parameter estimate from time step $k-1$.

For various reasons some systems are only modeled by a likelihood function. Often these systems are static and there exists no Markov transition density. However, most systems in this thesis are modeled using both a prediction and a likelihood function. In system identification, the model parameter is estimated without physically describing the parameter's time dependency, hence static estimation theory is used. The state can be estimated in more or less the same way. However, the process model (1.1a) is often given and its time transition information is exploited to further improve the state estimate.

The origins of the estimation research field can be traced back to the work by Gauss in 1795 on least squares (Abdulle and Wanner, 2002) and Bayes (1763) on conditional probabilities. Bayes introduced an important theorem which has come to be referred to as *Bayes' theorem*,

$$p(\mathbf{x},\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})p(\mathbf{x},\boldsymbol{\theta})}{p(\mathbf{y})}, \tag{3.2}$$

with which it is possible to calculate the *posterior probability* $p(\mathbf{x},\boldsymbol{\theta}|\mathbf{y})$ given a *prior probability* $p(\mathbf{x},\boldsymbol{\theta})$ and the *likelihood function* $p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})$. Note that both the measurement and the state or parameter are treated as random variables. Another view of the estimation problem was introduced by Fisher (1922), who claimed that the probability of an estimate should be seen as a relative frequency of the state or parameter, given data from long-run experiments. Fisher also treats the measurement as a random variable. The main difference to Bayes' approach is that in Fisher's approach there is a true state or parameter which is treated as deterministic, but unknown. To accentuate the different views, the likelihood is often written using $\ell(\mathbf{x},\boldsymbol{\theta})$ to emphasize that the likelihood is regarded as a function of the state $\mathbf{x}$ and the parameter $\boldsymbol{\theta}$.

After this brief historical background, the remainder of this chapter is outlined as follows. In Section 3.1, static estimation methods based on both Fishers and Bayes theories, are discussed. These methods can be used for both state and parameter estimation. In Section 3.2, dynamic estimation methods are discussed. These methods are, within the scope of this thesis, only used for state estimation and are based solely on Bayes' theories.

3.1 Static Estimation Theory

The general estimation problem consists of finding the estimates $\hat{\mathbf{x}}$ and $\hat{\boldsymbol{\theta}}$ that minimize a given loss function $V(\mathbf{x},\boldsymbol{\theta};\mathbf{y})$. This problem is separated into a parameter estimation problem and a state estimation problem according to

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \; V(\boldsymbol{\theta};\mathbf{x},\mathbf{y}), \tag{3.3a}$$

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \; V(\mathbf{x};\boldsymbol{\theta},\mathbf{y}). \tag{3.3b}$$

How to separate a typical estimation problem into these two parts is shown Example 3.2.

General estimation techniques are covered by most textbooks on this topic, e.g., Kay (1993); Kailath et al. (2000); Ljung (1999). There are many estimation methods available, however, in this section the focus is on the methods used in Part II of this thesis.

3.2 Example: Parameter and State Estimation

Consider the linear single track model in Example 2.7. Suppose that the state variables are measured with external and highly accurate sensors. The yaw rate is measured with an extra IMU and the body side slip angle β is measured with a so called Correvit® sensor, which uses correlation technology to compute the optical flow. The vehicle in Figure 2 on Page 297 is equipped with this type of sensor at the front. Now, the parameter $\boldsymbol{\theta}$ can be estimated, according to (3.3a). This approach has been used to find some of the vehicle parameters in Papers B and G.

Conversely, if $\boldsymbol{\theta}$ is known and $\mathbf{y}$ is measured, the state variables $\mathbf{x}$ can be estimated using (3.3b).

This section covers estimation problems without any process model $f(\,\cdot\,)$, where a set of measurements is related to a parameter only via the measurement model $h(\,\cdot\,)$. Furthermore, only an important and special case where the measurement model is linear in $\mathbf{x}$ is considered. The linear measurement model was given in (2.23b) and is repeated here for convenience

$$\mathbf{y}_k = H_k(\boldsymbol{\theta})\mathbf{x}_k + \mathbf{e}_k. \tag{3.4}$$

3.1.1 Least Squares Estimator

The *least squares* (LS) estimate is defined as the solution to the optimization problem, where the squared errors between the predicted measurements and the actual measurements are minimized according to,

$$\hat{\mathbf{x}}_k^{LS} = \arg\min_{\mathbf{x}} \sum_{\tau=1}^{k} \|\mathbf{y}_\tau - H_\tau(\boldsymbol{\theta})\mathbf{x}_\tau\|_2^2. \tag{3.5}$$

The solution for the linear case is given in Algorithm 1.

If the measurement covariance $R = \operatorname{Cov}(\mathbf{e})$ is known, or in practice at least assumed to be known, then the *weighted least squares* (WLS) estimate is given by the optimization problem

$$\hat{\mathbf{x}}_k^{WLS} = \arg\min_{\mathbf{x}} \sum_{\tau=1}^{k} (\mathbf{y}_\tau - H_\tau(\boldsymbol{\theta})\mathbf{x}_\tau)^{\mathrm{T}} R_\tau^{-1} (\mathbf{y}_\tau - H_\tau(\boldsymbol{\theta})\mathbf{x}_\tau). \tag{3.6}$$

The solution for the linear case is given in Algorithm 2, and Example 3.3 illustrates how the single track vehicle model can be reformulated to estimate the

Algorithm 1 Least Squares

The least squares estimate and its covariance are given by

$$\hat{\mathbf{x}}_k^{LS} = \left(\sum_{\tau=1}^{k} H_\tau^{\mathrm{T}} H_\tau \right)^{-1} \sum_{\tau=1}^{k} H_\tau^{\mathrm{T}} \mathbf{y}_\tau = \left(\boldsymbol{H}^{\mathrm{T}} \boldsymbol{H} \right)^{-1} \boldsymbol{H}^{\mathrm{T}} \boldsymbol{Y}, \tag{3.7a}$$

$$\mathrm{Cov}\,(\hat{\mathbf{x}}^{LS}) = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{H})(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1} \triangleq P^{LS}. \tag{3.7b}$$

The last equality is the batch solution, where $\boldsymbol{H}$ and $\boldsymbol{Y}$ are augmented measurement models and measurement vectors, respectively. Furthermore, the measurement noises $R_\tau = \mathrm{Cov}\,(\mathbf{e}_\tau)$ form the main diagonal of $\boldsymbol{R}$ according to $\boldsymbol{R} = \mathrm{diag}(R_1, \ldots, R_k)$.

Algorithm 2 Weighted Least Squares

The weighted least squares estimator and its covariance matrix are given by

$$\hat{\mathbf{x}}_k^{WLS} = \left(\sum_{\tau=1}^{t} H_\tau^{\mathrm{T}} R_\tau^{-1} H_\tau \right)^{-1} \sum_{\tau=1}^{k} H_\tau^{\mathrm{T}} R_\tau^{-1} \mathbf{y}_\tau = \left(\boldsymbol{H}^{\mathrm{T}} \boldsymbol{R}^{-1} \boldsymbol{H} \right)^{-1} \boldsymbol{H}^{\mathrm{T}} \boldsymbol{R}^{-1} \boldsymbol{Y}, \tag{3.8a}$$

$$\mathrm{Cov}\,(\hat{\mathbf{x}}^{WLS}) = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{H})^{-1} \triangleq P^{WLS}, \tag{3.8b}$$

where the weighting matrix is the noise covariance $\boldsymbol{R}$.

parameters using the WLS.

Another example, where both the LS and the WLS estimators are applied, is given in Paper C. The left and right borders of a road are modeled by polynomials and the coefficients are the parameters which are estimated given a batch of measurements from a radar.

3.3 Example: Parameter and State Estimation

Consider the linear single track model in Example 2.7 and the separation of the parameter and the state estimation problems in Example 3.2. Suppose that the vehicle's mass m and the dimensions l_f and l_r are known. Furthermore, suppose that the state variable $\mathbf{x}$ may be measured as described in Example 3.2. Consider the measurement equation (2.24b); the parameter estimation problem can now be formulated in the form (3.4) with

$$\mathbf{y} = H(\mathbf{x}, \mathbf{u}, l_f, l_r, m) \begin{bmatrix} C_{\alpha f} \\ C_{\alpha r} \end{bmatrix} + \mathbf{e}, \tag{3.9}$$

and the parameters $C_{\alpha f}, C_{\alpha r}$ can be solved for using e.g., WLS in (3.6). Furthermore, the inverse of the moment of inertia $1/I_{zz}$ may be estimated off-line by writing the process model (2.24a) in the form (3.4) according to

$$\mathbf{x}_{t+1} = H(\mathbf{x}_t, \mathbf{u}, l_v, l_f, m, C_{\alpha f}, C_{\alpha r}) \cdot \frac{1}{I_{zz}} + \mathbf{w}. \tag{3.10}$$

3.1.2 Probabilistic Point Estimates

The *maximum likelihood* (ML) estimate, first introduced by Fisher (1912, 1922), is defined by

$$\hat{\mathbf{x}}_k^{ML} = \arg\max_{\mathbf{x}_k} \; p(\mathbf{y}_{1:k}|\mathbf{x}_k). \tag{3.11}$$

Put into words, the estimate is chosen to be the parameter most likely to produce the obtained measurements.

The posterior $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, when $\mathbf{x}_k$ is random, contains all known information about the state of the target at time k. The *maximum a posteriori* (MAP) estimator is defined by

$$\hat{\mathbf{x}}_k^{MAP} = \arg\max_{\mathbf{x}_k} \; p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \arg\max_{\mathbf{x}_k} \; p(\mathbf{y}_{1:k}|\mathbf{x}_k)p(\mathbf{x}_k), \tag{3.12}$$

or put in words, find the most likely estimate of the parameter given the measurements $\mathbf{y}_{1:k}$. Bayes' theorem (3.2) and the fact that the maximization is performed over $\mathbf{x}_k$ is used in the second equality of (3.12). The ML estimate is for instance used in Paper E.

3.2 Filter Theory

The topic of this section is recursive state estimation based on dynamic models. The iteration process of the state space estimation was briefly described in words in Section 1.5. The state estimation theory is influenced by the Bayesian view, which implies that the solution to the estimation problem is provided by the filtering PDF $p(\mathbf{x}_k|\mathbf{y}_{1:k})$. The introduction to this section will be rather general using the model defined in (2.30). Bayes' theorem was introduced in (3.2) and is used to derive the recursive *Bayes filter equations*

$$p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k}) = \int p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k})d\mathbf{x}_k, \tag{3.13a}$$

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})}, \tag{3.13b}$$

with the denominator

$$p(\mathbf{y}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})d\mathbf{x}_k. \tag{3.13c}$$

These equations describe the time evolution

$$\cdots \rightarrow \mathbf{x}_{k|k} \rightarrow \mathbf{x}_{k+1|k} \rightarrow \mathbf{x}_{k+1|k+1} \rightarrow \cdots \tag{3.14}$$

of the random state vector $\mathbf{x}$. The Bayes posterior density function $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ conditioned on the time sequence $\mathbf{y}_{1:k} = \{\mathbf{y}_1, \ldots, \mathbf{y}_k\}$ of measurements accumulated at time k is the probability density function of $\mathbf{x}_k$ given measurements $\mathbf{y}_{1:k}$. The probability density function $p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k})$ is the time prediction of the posterior $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ to the time step of the next measurement $\mathbf{y}_{k+1}$. Note that the Bayes normalization factor given by (3.13c) is independent of $\mathbf{x}$. In practice the numerator of (3.13b) is calculated and then simply normalized, since the integral of the posterior density function must be equal to one. The single target Bayes filter recursion is illustrated with a flow chart as follows

$$\begin{tikzcd}
\cdots \arrow[r] & p_{k|k}(\mathbf{x}_k|\mathbf{z}_{1:k}) \arrow[r, "\text{prediction}"] & p_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{z}_{1:k}) \arrow[r, "\text{update}"] & p_{k+1|k+1}(\mathbf{x}_{k+1}|\mathbf{z}_{1:k+1}) \arrow[r] & \cdots \\
 & & \begin{array}{c}\text{Transition density}\\ p(\mathbf{x}_{k+1}|\mathbf{x}_k)\end{array} \arrow[u] & \begin{array}{c}\text{Likelihood function}\\ p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})\end{array} \arrow[u] & \\
 & & \begin{array}{c}\text{Motion model}\\ \mathbf{x}_{k+1} = f(\mathbf{x}_k) + \mathbf{w}_k\end{array} \arrow[u] & \begin{array}{c}\text{Measurement model}\\ \mathbf{z}_{k+1} = h(\mathbf{x}_{k+1}) + \mathbf{e}_{k+1}\end{array} \arrow[u] &
\end{tikzcd}$$

After the update step is made the algorithm continues recursively, i.e., a new prediction is performed and thereafter a update or *correction* with the likelihood of the new measurements.

If $p(\mathbf{y}_k|\mathbf{x}_k)$, $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and $p(\mathbf{x}_k)$ are Gaussian and their corresponding process and senor models are linear, as in (2.23), then (3.13a) and (3.13b) reduce to the Kalman filter prediction and measurement update, respectively. The Kalman filter is treated in Section 3.2.1. In contrast, if $p(\mathbf{y}_k|\mathbf{x}_k)$, $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and $p(\mathbf{x}_k)$ can be approximated by a Gaussian and their corresponding process and sensor models are nonlinear (2.1), several approximations of (3.13a) and (3.13b) exist. The two most common filters are the extended Kalman Filter and the unscented Kalman filter, which are outlined in Sections 3.2.2 and 3.2.3, respectively. Other methods, including methods that approximate other density functions than Gaussian, are neatly covered by Hendeby (2008) and Schön (2006). The most popular approaches are the particle filter, which is covered in Section 3.2.4, and the marginalized particle filter, see e.g., Arulampalam et al. (2002); Cappe et al. (2007); Djuric et al. (2003); Karlsson (2005); Schön et al. (2005).

3.2.1 The Kalman Filter

The linear state space representation subject to Gaussian noise, which was given in (2.23), is the simplest special case when it comes to state estimation. The model is repeated here for convenience;

$$\mathbf{x}_{k+1} = F_k(\boldsymbol{\theta})\mathbf{x}_k + G_k^{\mathbf{u}}(\boldsymbol{\theta})\mathbf{u}_k + G_k^{\mathbf{w}}\mathbf{w}_k, \qquad \mathbf{w}_k \sim \mathcal{N}(0, Q_k), \tag{3.15a}$$

$$\mathbf{y}_k = H_k(\boldsymbol{\theta})\mathbf{x}_k + H_k^{\mathbf{u}}(\boldsymbol{\theta})\mathbf{u}_k + \mathbf{e}_k, \qquad \mathbf{e}_k \sim \mathcal{N}(0, R_k). \tag{3.15b}$$

The linear model (3.15) has two important properties. All density functions involved in the model and state estimation are Gaussian and a Gaussian density function is completely parametrized by the mean and the covariance, i.e. the first and second order moment. Hence, the Bayesian recursion (3.13) is simplified into only propagating the mean and covariance of the involved probability den-

Algorithm 3 Kalman Filter

Consider the linear state space model (3.15). The Kalman filter is given by the two following steps.

Prediction

$$\hat{\mathbf{x}}_{k|k-1} = F_{k-1}\hat{\mathbf{x}}_{k-1|k-1} + G^{\mathbf{u}}_{k-1}\mathbf{u}_{k-1} \tag{3.16a}$$

$$P_{k|k-1} = F_{k-1}P_{k-1|k-1}F^{\mathrm{T}}_{k-1} + G^{\mathbf{w}}_{k-1}Q_{k-1}G^{\mathbf{w}\mathrm{T}}_{k-1} \tag{3.16b}$$

Measurement Update

$$K_k = P_{k|k-1}H^{\mathrm{T}}_k(H_kP_{k|k-1}H^{\mathrm{T}}_k + R_k)^{-1} \tag{3.17a}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{y}_k - H_k\hat{\mathbf{x}}_{k|k-1} - H^u_k\mathbf{u}_k) \tag{3.17b}$$

$$P_{k|k} = (I - K_kH_k)P_{k|k-1} \tag{3.17c}$$

sity functions, as illustrated in the following flow chart

$$\begin{tikzcd}
\cdots \arrow[r] & \mathcal{N}(\mathbf{x}_{k|k}, P_{k|k}) \arrow[r, "\text{prediction}"] & \mathcal{N}(\mathbf{x}_{k+1|k}, P_{k+1|k}) \arrow[r, "\text{update}"] & \mathcal{N}(\mathbf{x}_{k+1|k+1}, P_{k+1|k+1}) \arrow[r] & \cdots \\
 & & \begin{array}{c}\text{Transition density}\\ \mathcal{N}(\mathbf{x}_{k+1}; F\mathbf{x}_k, Q_k)\end{array} \arrow[u] & \begin{array}{c}\text{Likelihood function}\\ \mathcal{N}(\mathbf{z}_{k+1}; H\mathbf{x}_{k+1}, R_{k+1})\end{array} \arrow[u] & \\
 & & \begin{array}{c}\text{Motion model}\\ \mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{w}_k\end{array} \arrow[u] & \begin{array}{c}\text{Measurement model}\\ \mathbf{z}_{k+1} = H\mathbf{x}_{k+1} + \mathbf{e}_{k+1}\end{array} \arrow[u] &
\end{tikzcd}$$

The most well known estimation algorithm is the Kalman Filter (KF), derived by Kalman (1960), and shown in Algorithm 3. Example 3.4 shows how the single track vehicle model, introduced in Example 1.1, may be rewritten to be used with the Kalman filter, which in turn is used to estimate the states.

3.4 Example: Linearized Single Track Model

The single track vehicle model was introduced in Example 1.1 and the model equations were posed in Example 2.1. The process model (2.4a) and the measurement model (2.4b) are linear in the state variables and can be written in the form

$$\begin{bmatrix}\dot{\psi}_{k+1}\\ \beta_{k+1}\end{bmatrix} = F_k(\mathrm{v_x}, \boldsymbol{\theta})\begin{bmatrix}\dot{\psi}_k\\ \beta_k\end{bmatrix} + G^{\mathbf{u}}_k(\mathrm{v_x}, \boldsymbol{\theta})\delta_f + \mathbf{w}, \qquad \mathbf{w} \sim \mathcal{N}(0, Q), \tag{3.18a}$$

$$\begin{bmatrix}\dot{\psi}^m_k\\ a_{\mathrm{y},k}\end{bmatrix} = H_k(\mathrm{v_x}, \boldsymbol{\theta})\begin{bmatrix}\dot{\psi}_k\\ \beta_k\end{bmatrix} + H^u_k(\boldsymbol{\theta})\delta_f + \mathbf{e}, \qquad \mathbf{e} \sim \mathcal{N}(0, R), \tag{3.18b}$$

as shown in Example 2.7. Since the input $\mathrm{v_x}$ is present in F_k, $G^{\mathbf{u}}_k$ and H_k, these matrices must be recalculated at each time step before being used in the Kalman filter (Algorithm 3) to estimate the states.

3.2.2 The Extended Kalman Filter

In general, most complex automotive systems tend to be nonlinear. When it comes to solving state estimation problems in a sensor fusion framework, nonlinear models are commonly applied. This holds also for the work presented in this thesis, but the problems are restricted by the assumption that the process and measurement noise are Gaussian. The most common representation of nonlinear systems is the state space model given in (1.1), repeated here for convenience;

$$\begin{aligned} \mathbf{x}_{k+1} &= f_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k, \boldsymbol{\theta}), & \mathbf{w}_k &\sim \mathcal{N}(0, Q_k), \quad (3.19a)\\ \mathbf{y}_k &= h_k(\mathbf{x}_k, \mathbf{u}_k, \mathbf{e}_k, \boldsymbol{\theta}), & \mathbf{e}_k &\sim \mathcal{N}(0, R_k). \quad (3.19b) \end{aligned}$$

The basic idea underlying the extended Kalman filter (EKF) is to approximate the nonlinear model (3.19) by a local linear model and apply the Kalman filter to this approximation. This local linear approximation is obtained by computing a first order Taylor expansion around the current estimate. The result is the extended Kalman filter, which is given in Algorithm 4. Early practical applications and examples of the EKF are described in the works by Smith et al. (1962); Schmidt (1966). An early reference where the EKF is treated is Jazwinski (1970), other standard references are Anderson and Moore (1979); Kailath et al. (2000) .

The linearization used in the EKF assumes that all second and higher order terms in the Taylor expansion are negligible. This is certainly true for many systems, but for some systems this assumption can significantly degrade the estimation performance. Higher order EKF are discussed by Roth and Gustafsson (2011); Bar-Shalom and Fortmann (1988); Gustafsson (2000). This problem will be revisited in the next section.

Algorithm 4 Extended Kalman Filter

Consider the state space model (3.19). The extended Kalman filter is given by the two following steps.

Prediction

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= f_{k-1}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}, 0, \boldsymbol{\theta}) \quad (3.20a)\\ P_{k|k-1} &= F_{k-1} P_{k-1|k-1} F_{k-1}^{\mathrm{T}} + G_{k-1} Q_{k-1} G_{k-1}^{\mathrm{T}} \quad (3.20b) \end{aligned}$$

where

$$F_k = \left.\frac{\partial f_k(\mathbf{x}_k, \mathbf{u}_k, 0, \boldsymbol{\theta})}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k = \hat{\mathbf{x}}_{k|k}} \qquad G_k = \left.\frac{\partial f_k(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k, \mathbf{w}_k, \boldsymbol{\theta})}{\partial \mathbf{w}_t}\right|_{\mathbf{w}_k = 0} \quad (3.20c)$$

Measurement Update

$$\begin{aligned} K_k &= P_{k|k-1} H_k^{\mathrm{T}} (H_k P_{k|k-1} H_k^{\mathrm{T}} + R_k)^{-1} \quad (3.21a)\\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{y}_k - h_k(\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_t, 0, \boldsymbol{\theta})) \quad (3.21b)\\ P_{k|k} &= (I - K_k H_k) P_{k|k-1} \quad (3.21c) \end{aligned}$$

where

$$H_k = \left.\frac{\partial h_k(\mathbf{x}_k, \mathbf{u}_k, 0, \boldsymbol{\theta})}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k = \hat{\mathbf{x}}_{k|k-1}} \quad (3.21d)$$

3.2.3 The Unscented Kalman Filter

The EKF is sufficient for many applications. However, to use an EKF the gradients of $f_k(\cdot)$ and $h_k(\cdot)$ must be calculated, which in some cases is either hard to do analytically or computationally expensive to do numerically. An alternative approach, called the unscented Kalman filter (UKF) was proposed by Julier et al. (1995); Julier and Uhlmann (1997) and further refined by e.g., Julier and Uhlmann (2002, 2004); Julier (2002). Instead of linearizing $f_k(\cdot)$ and $h_k(\cdot)$, the unscented transform (UT) is used to approximate the moments of the predicted and updated states. Thereby the UKF to some extent also considers the second order terms of the models, which is not done by the EKF.

The principle of the unscented transform is to carefully and deterministically select a set of points, called sigma points, of the initial stochastic variable $\mathbf{x}$, such that their mean and covariance are equal to those of $\mathbf{x}$. Then the sigma points are passed through the non-linear function and based on the output the resulting mean and covariance are derived. In case the process noise and measurement noise are not additive, sigma points are selected from an augmented state space, which includes the state $\mathbf{x}$, the process noise $\mathbf{w}$ and the measurement noise $\mathbf{e}$ in one augmented state vector

$$\hat{\mathbf{x}}^a_{k|k} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \mathrm{E}\left(\mathbf{w}_k\right) \\ \mathrm{E}\left(\mathbf{e}_{k+1}\right) \end{bmatrix}, \tag{3.22}$$

with dimension $n_a = n_\mathbf{x} + n_\mathbf{w} + n_\mathbf{e}$ and the corresponding covariance matrix

$$P^a_{k|k} = \begin{bmatrix} P_{k|k} & 0 & 0 \\ 0 & Q_k & 0 \\ 0 & 0 & R_{k+1} \end{bmatrix}. \tag{3.23}$$

If the noise is additive, then the noise covariances can be added directly to the estimated covariances of the non-augmented sigma points.

There exist many possibilities to choose the sigma points, a thorough discussion about different alternatives is presented by Julier and Uhlmann (2004). In the present work only the standard form is reproduced. The basic principle is to choose one sigma point in the mean of $\mathbf{x}^a$ and $2n_a$ points symmetrically on a given contour, described by the state covariance P^a. The sigma points χ^i and the associated weights $w^{(i)}$ are chosen as

$$\chi^{(0)} = \hat{\mathbf{x}}^a \qquad w^{(0)} = w^{(0)} \tag{3.24a}$$

$$\chi^{(i)} = \chi^{(0)} + \left(\sqrt{\frac{n_a}{1-w^{(0)}}P^a}\right)_i \qquad w^{(i)} = \frac{1-w^{(0)}}{2n_a} \tag{3.24b}$$

$$\chi^{(i+n_a)} = \chi^{(0)} - \left(\sqrt{\frac{n_a}{1-w^{(0)}}P^a}\right)_i \qquad w^{(i+n_a)} = \frac{1-w^{(0)}}{2n_a} \tag{3.24c}$$

for $i = 1, \dots, n_a$, where $(\sqrt{A})_i$ is the i^{th} column of any matrix B, such that $A = BB^\mathrm{T}$. The augmented state vector makes it possible to propagate and estimate nonlin-

Algorithm 5 Unscented Kalman Filter

Consider the state space model (3.19). The unscented Kalman filter is given by the following steps, which are iterated in the filter.

Choose sigma points according to (3.24)

Prediction

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2n_a} w^{(i)} \chi^{\mathbf{x},(i)}_{k|k-1} \tag{3.25a}$$

$$P_{k|k-1} = \sum_{i=0}^{2n_a} w^{(i)} \left(\chi^{\mathbf{x},(i)}_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right)\left(\chi^{\mathbf{x},(i)}_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right)^{\mathrm{T}} \tag{3.25b}$$

where

$$\chi^{\mathbf{x},(i)}_{k|k-1} = f_{k-1}\left(\chi^{\mathbf{x},(i)}_{k-1|k-1}, \mathbf{u}_{k-1}, \chi^{\mathbf{w},(i)}_{k-1|k-1}, \boldsymbol{\theta}\right) \tag{3.25c}$$

Measurement Update

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + P_{\mathbf{xy}} P_{\mathbf{yy}}^{-1} (\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}) \tag{3.26a}$$

$$P_{k|k} = P_{k|k-1} - P_{\mathbf{xy}} P_{\mathbf{yy}}^{-1} P_{\mathbf{xy}}^{\mathrm{T}} \tag{3.26b}$$

where

$$\mathbf{y}^{(i)}_{k|k-1} = h_k\left(\chi^{\mathbf{x},(i)}_{k|k-1}, \mathbf{u}_k, \chi^{\mathbf{e},(i)}_{k|k-1}, \boldsymbol{\theta}\right) \tag{3.26c}$$

$$\hat{\mathbf{y}}_{k|k-1} = \sum_{i=0}^{2n_a} w^{(i)} \mathbf{y}^{(i)}_{k|k-1} \tag{3.26d}$$

$$P_{\mathbf{yy}} = \sum_{i=0}^{2n_a} w^{(i)} \left(\mathbf{y}^{(i)}_{k|k-1} - \hat{\mathbf{y}}_{k|k-1}\right)\left(\mathbf{y}^{(i)}_{k|k-1} - \hat{\mathbf{y}}_{k|k-1}\right)^{\mathrm{T}} \tag{3.26e}$$

$$P_{\mathbf{xy}} = \sum_{i=0}^{2n_a} w^{(i)} \left(\chi^{\mathbf{x},(i)}_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}\right)\left(\mathbf{y}^{(i)}_{k|k-1} - \hat{\mathbf{y}}_{k|k-1}\right)^{\mathrm{T}} \tag{3.26f}$$

ear influences that the process noise and the measurement noise have on the state vector and the measurement vector, respectively. The weight on the mean $w^{(0)}$ is used for tuning and according to Julier and Uhlmann (2004) preferable properties for Gaussian density functions are obtained by choosing $w^{(0)} = 1 - \frac{n_a}{3}$. After the sigma points have been acquired, the augmented state vector can be partitioned according to

$$\chi^a_{k|k} = \begin{bmatrix} \chi^{\mathbf{x}}_{k|k} \\ \chi^{\mathbf{w}}_{k} \\ \chi^{\mathbf{e}}_{k+1} \end{bmatrix}. \tag{3.24d}$$

The rest of the UKF is summarized in Algorithm 5.

An advantage of the UKF, compared to the EKF, is that the second order bias correction term is implicitly incorporated in the mean estimate. Example 3.5 shows an important problem where the second order term should not be neglected.

3.5 Example: Tracked Radar Object

The radar target tracking problem was introduced in Example 1.2 and the model was defined in Example 2.4. The sensor model converts the Cartesian state variables to polar measurements. This is one of the most important and commonly used transformations for sensors measuring range and azimuth angle. Usually the azimuth angle error of these type of sensors is significantly larger than the range error. This also holds for the sensors used in this thesis.

Let the sensor be located at the origin and the target at $(\mathrm{x}, \mathrm{y}) = (0, 1)$ in this simple, and commonly used example (Julier and Uhlmann, 2004). Measurements may be simulated by adding Gaussian noise to the actual polar value $(r, \psi) = (1, \pi/2)$ of the target localization. A plot of several hundred state estimates, produced in a Monte Carlo simulation, forms a banana shaped arc around the true value $(\mathrm{x}, \mathrm{y}) = (0, 1)$, as shown in Figure 3.1. The azimuth error causes this band of Cartesian points to be stretched around the circumference of a circle, with the result that the mean of these points lies somewhat closer to the origin than the point $(0, 1)$. In the figure it is clearly shown that that the UT estimate ($\times$) lies close to the mean of the measurements ($\circ$). Furthermore, it is shown that the linearized state estimate ($+$) produced by the EKF is biased and the variance in the y component is underestimated.

As a result of the linearization in the EKF, the second order terms are neglected, which produces a bias error in the mean as shown in Example 3.5. In Julier and Uhlmann (2004) it is shown how the UT in some cases calculates the projected mean and covariance correctly to the second order terms.

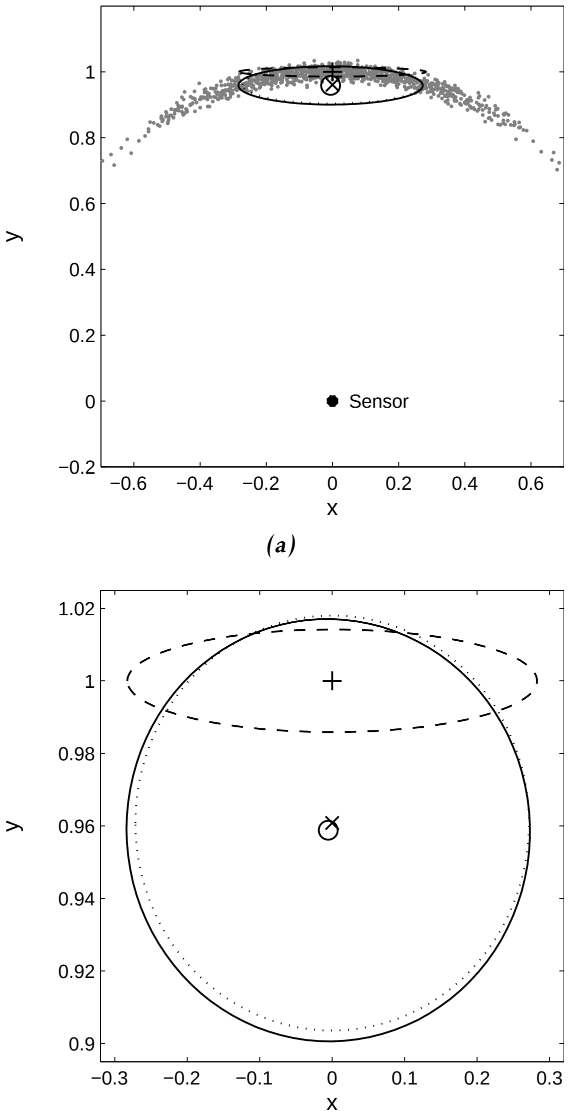
The unscented Kalman filter is applied in Papers C and D.

3.2.4 The Particle Filter

Numerical approximations of distribution function used in the Bayes equations (3.13) are necessary since analytical solutions to the filtering problems do not exist in most cases. A stochastic method called sequential Monte Carlo (SMC) or particle filter (PF), which is based on Monte Carlo integration, has become popular during the last years. It was introduced by Gordon et al. (1993) and theory is well described by e.g., Doucet et al. (2001, 2000); Ristic et al. (2004); Gustafsson (2010). The idea is to approximate the PDF with a number of samples, called particles, $\{\mathbf{x}_k^{(i)}\}_{i=1}^N$ with associated weights $\{w_{k|k-1}^{(i)}\}_{i=1}^N$ such that

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \approx \sum_{i=1}^{N} w_{k|k-1}^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}). \tag{3.27}$$

The particles are selected to be identically and independently distributed samples from a so called proposal distribution $q(\mathbf{x}_{k+1}|\mathbf{x}_k^{(i)}, \mathbf{y}_{1:k+1})$ and the weights are called importance weights. Using Monte Carlo integration the mean value of the



(a)

(b)

Figure 3.1: *A Monte Carlo simulation of the problem in Example 3.5 is shown in Figure (a). The sensor, for example a radar, is in the position* $(0, 0)$ *and the true position of the target is in the position* $(0, 1)$*. The mean of the measurements is at* $\circ$ *and the uncertainty ellipse is solid. The linearized mean is at* $+$ *and its ellipse is dashed. The UT mean is at* $\times$ *and its uncertainty ellipse is dotted. Figure (b) is a zoom. Note that the scaling in the* x *and the* y *axis are different.*

Algorithm 6 Particle Filter

```
 1: Initiate $\{\mathbf{x}_0^{(i)}\}_{i=1}^N \sim p_{\mathbf{x}_0}$ and $\{w_0^{(i)}\}_{i=1}^N = \frac{1}{N}$ for $i = 1, \ldots, N$
 2: $k = 0$
 3: loop
 4:   for $i = 1$ to $N$ do
 5:     Time Update: generate the sample, $\mathbf{x}_{k|k-1}^{(i)} \sim q(\mathbf{x}_k|\mathbf{x}_{k-1|k-1}^{(i)}, \mathbf{y}_k)$
 6:     Update importance weights
            $w_{k|k-1}^{(i)} = w_{t-1|t-1}^{(i)} \frac{p(\mathbf{x}_{k|k-1}^{(i)}|\mathbf{x}_{k-1|k-1}^{(i)})}{q(\mathbf{x}_{k|k-1}^{(i)}|\mathbf{x}_{k-1|k-1}^{(i)}, \mathbf{y}_k)}$
        {Note, if $q(\mathbf{x}_k|\mathbf{x}_{k-1|k-1}^{(i)}, \mathbf{y}_{1:k}) = p(\mathbf{x}_k|\mathbf{x}_{k-1|k-1}^{(i)})$ this simplifies to
        $w_{k|k-1}^{(i)} = w_{t-1|t-1}^{(i)}$}
 7:     Measurement update: $\bar{w}_{k|k}^{(i)} = p(\mathbf{y}_k|\mathbf{x}_{k|k}^{(i)})w_{k|k-1}^{(i)}$
 8:   end for
 9:   Normalize weights $w_{k|k}^{(i)} = \frac{\bar{w}_{k|k}^{(i)}}{\sum_j \bar{w}_{k|k}^{(j)}}$ for $i = 1, \ldots, N$
10:   Resample
11: end loop
```

distribution can easily be calculated according to

$$\hat{\mathbf{x}}_{k|k-1} = \mathrm{E}(\mathbf{x}_k) \approx \sum_{i=1}^{N} w_{k|k-1}^{(i)} \mathbf{x}_k^{(i)}. \tag{3.28}$$

The PF is described in Algorithm 6.

The resampling in Line 10 rejuvenates the particles used to represent the PDF. There exist different methods to resample, and it will not be discussed further here; the reader is referred to Hendeby (2008); Schön (2006) instead.

The PF is used in Paper C to track extended targets. In Paper G, the PF is used since the mean and covariance of the process noise are unknown and must be estimated together with the states. In that case the particles are drawn from a Student-t distribution.

4

Target Tracking

The process of estimating over time the location and characteristics of one or more objects of interest, denoted *targets*, using one or several sensors is referred to as target *tracking*. The aim of the tracking algorithm is to detect the true targets, collect observations that originate from them and to estimate quantities of interest, such as target position, velocity and other object characteristics. Typical sensors for these applications, such as radar, laser and vision, report noisy measurements. Besides the fact that the measurements are noisy another difficulty for the algorithm is that the sensor measurements not only originate from the targets, but also from clutter or spurious detections. The target tracking area is well covered in the literature and recommended books are Blackman and Popoli (1999); Bar-Shalom and Fortmann (1988); Bar-Shalom et al. (2001); Ristic et al. (2004); Liggins et al. (2009).

This chapter begins by putting the filter, introduced in the last chapter about estimation theory, into the target tracking framework. In Section 4.1 the focus is on the filter and therefore only the single target case is considered. An automotive target tracking system, must obviously be able to handle multiple targets, because potentially more than one vehicle are surrounding the own vehicle. Therefore the chapter is continued by discussing the extension to *multi target tracking* in Section 4.2. When reading the first two sections, compare with the block diagram of the target tracking framework illustrated in the introduction in Figure 1.4. Finally, in Section 4.3 the system is further enlarged to also be able to track the shape and size of the the targets. This research area is called *extended target tracking*.

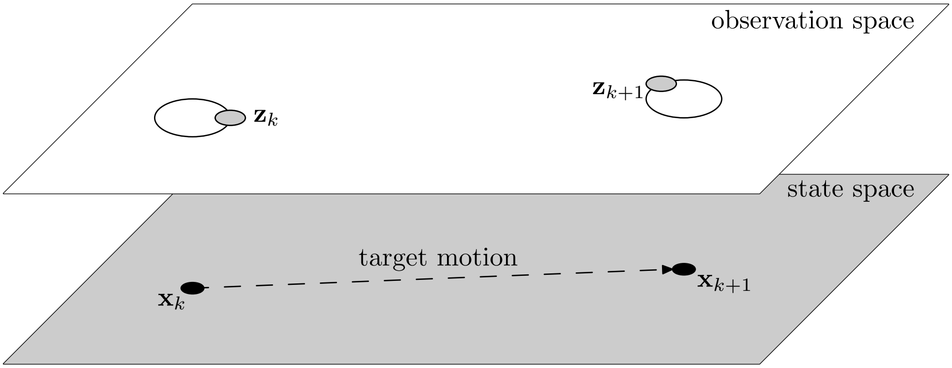


Figure 4.1: *Single target tracking problem. The upper layer is the observation space with the measurements and the lower layer is the state space with the state variables.*

4.1 Single Target Tracking

The complete information required to describe the system at time k is summarized in the state vector $\mathbf{x}_k \in \mathbb{R}^{n_x}$. At each time step the tracking algorithm is supplied with a measurement vector $\mathbf{z} \in \mathbb{R}^{n_z}$. The aim of the filter, as part of a target tracking algorithm, is to find the *posterior density* $p(\mathbf{x}_k|\mathbf{z}_{1:k})$, given a sequence of noisy measurement vectors up to and including time k, collected in

$$\mathbf{z}_{1:k} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots \mathbf{z}_k\}. \tag{4.1}$$

From the posterior it is possible to calculate an estimate $\hat{\mathbf{x}}_k$ of the state $\mathbf{x}_k$. Typically, the targets or the sensors are moving and the system changes over time. The motion of the targets in time is modeled with a Markov density, given in (2.27), and the PDF is specified with a motion model (2.1a). The measurement $\mathbf{z}_k$ received by the sensor is modeled by a likelihood function (2.29), which is specified by a measurement model (2.1b). Both the process and measurement noise variables are assumed to be white and independent. Figure 4.1 illustrates the filtering problem. The upper layer represents the *observation space*, where one measurement $\mathbf{z}_k$ is observed at each time step k and another measurement is observed at time step $k + 1$. The lower layer represents the *state space*, with the values of the state vector $\mathbf{x}_k$ at each time step.

The single target Bayes filter recursion consists of two steps, the *prediction* step and the *correction* step. Since all variables are assumed to be unknown stochastic variables in the Bayesian framework it is necessary to describe *a priori* information about the state in the form of a prior density $p_0(\mathbf{x}_0)$. The recursion was illustrated with a flow chart in Section 3.2. A special case is the Kalman filter, where the densities are assumed to be Gaussian.

4.2 Extension to Multitarget Tracking

In this chapter until now only single targets have been considered. This scenario is quite unusual, in most target tracking applications more than one target may appear and must be handled by the target tracking system. A multi target tracking system must not only provide estimates of the state variables

$$\left\{\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}, \ldots, \mathbf{x}_k^{(N_\mathbf{x})}\right\} \tag{4.2}$$

it must also estimate the number of targets $N_\mathbf{x}$, and find likely associations of the measurements to the tracks. The latter uncertainty is known as the data association problem.

A typical multi target system is shown in Figure 4.2, where several targets exist at each time step. Aspects to consider in a multitarget tracking filter are listed below:

- The number of targets $N_\mathbf{x}$ changes over time, for instance there are probably more vehicles on the roads during rush hour than in the night.
- New targets appear and old targets disappear, as they enter or leave the field of view of the sensor. Compare with Figure 4.2, where target $\mathbf{x}^{(4)}$ appears at time $k + 1$.
- The system fails to detect targets because they are temporarily occluded or because the sensor data is very noisy.
- The sensors receive a set of spurious measurements, also denoted *clutter*, stemming from false reflections or from other type of objects in the field of view, etc. In Figure 4.2 there are more observations than targets at each time step.

In this section first the data association problem is treated in Section 4.2.1 and thereafter an overview of the track management, which takes care of the problems in the list above is given in Section 4.2.2.

4.2.1 Data Association

This section would not be needed if only the state variables of the ego vehicle, introduced in Example 1.1 are estimated, because in that case it is obvious how the measurements are associated with the state variables. In the object tracking problem, introduced in Example 1.2, it is no longer obvious which measurement should update which track. There are many methods available for finding likely measurement-to-track associations, i.e., for solving the data association problem, see e.g., Bar-Shalom and Fortmann (1988); Blackman and Popoli (1999). However, the task is seldom easy, due to noisy measurements, multiple reflections on each target and erroneous detections caused by spurious reflections.

The first step in the data association process is called *gating*. Gates are constructed around the predicted measurement $\hat{\mathbf{z}}_{k|k-1}^{(i)}$ of each track i to eliminate

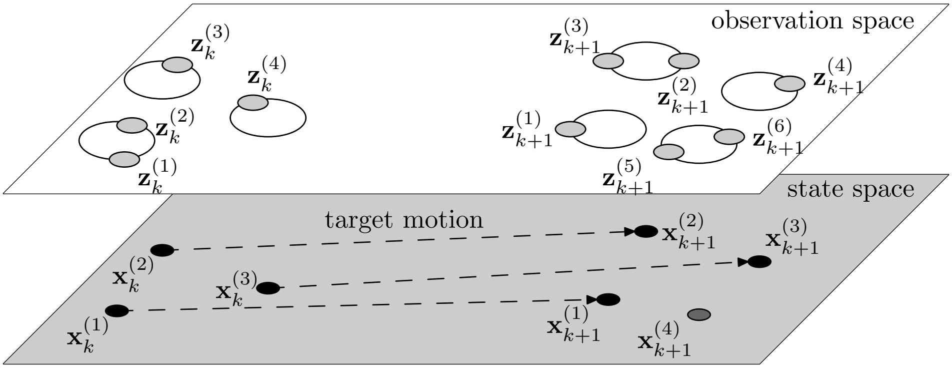


Figure 4.2: *Multi target tracking system, several state vectors are present at each time step and producing more than one observation. Note that there are more observations than targets. A new target $\mathbf{x}^{(4)}$ appears at time $k+1$.*

unlikely pairings and thereby to limit the number of measurement-to-track associations. This reduces the number of measurements that are examined by the data association algorithm and reduces the computational load. The residual, which is also called innovation, between a measurement $\mathbf{z}_k^{(j)}$ and a predicted measurement $\hat{\mathbf{z}}_{k|k-1}^{(i)}$ is

$$\tilde{\mathbf{z}}_{k|k-1}^{(i,j)} = \mathbf{z}_k^{(j)} - \hat{\mathbf{z}}_{k|k-1}^{(i)}, \tag{4.3}$$

and it is assumed Gaussian distributed according to

$$\tilde{\mathbf{z}}_{k|k-1}^{(i,j)} \sim \mathcal{N}(0, S_k^{(i)}), \tag{4.4}$$

where $S_k^{(i)}$ is the innovation covariance. Further, under this assumption, the statistical distance between the measurement and the predicted measurement, given by the norm of the residuals according to

$$d_{i,j}^2 = (\tilde{\mathbf{z}}_{k|k-1}^{(i,j)})^{\mathrm{T}} S_{i,k}^{-1} (\tilde{\mathbf{z}}_{k|k-1}^{(i,j)}), \tag{4.5}$$

is a $\chi^2_{n_y}$ random variable. The elliptical gate $\mathcal{G}_i$ is defined as the region

$$\mathcal{G}_i \triangleq \left\{ \mathbf{z} \middle| (\tilde{\mathbf{z}}_{k|k-1}^{(i,j)})^{\mathrm{T}} S_{i,k}^{-1} (\tilde{\mathbf{z}}_{k|k-1}^{(i,j)}) \leq \gamma_{\mathcal{G}} \right\}, \tag{4.6}$$

where $\gamma_{\mathcal{G}}$ is the gating threshold or gate size. Given a certain probability that a true measurement produced by a target i will fall inside its gate, and that the assumption $d_{i,j}^2 \sim \chi^2_{n_y}$ holds, a suitable gate threshold can be determined. The measurements $\mathbf{z}_k^{(j)} \in \mathcal{G}_i$ are considered as candidates for updating the track $\mathbf{x}_k^{(i)}$ in the data association algorithm. In other words the gating is a hard decision about which measurements are feasible measurements for a target.

Now, different conflicts occur. There are several measurements falling within the same gate and there are also measurements falling within more than one gate. There exist many techniques to solve these conflicts, which are considered to be the main part of the data association process. The simplest association algorithm is called *nearest neighbor* (NN). This approach searches for a unique pairing, i.e., one track $\mathbf{x}_k^{(i)}$ is only updated by at most one observation $\mathbf{z}_k^{(j)}$. There are some possibilities to decide which measurement actually is the nearest. Common approaches are to choose the measurement with the smallest error $\tilde{\mathbf{z}}_{k|k-1}^{(i,j)}$ or the smallest statistical distance $d^2(\tilde{\mathbf{z}}_{k|k-1}^{(i,j)})$, defined in (4.5), which is also known as the Mahalanobis distance, see e.g., Bar-Shalom et al. (2001). The association distance for each individual track is locally minimized separately with the NN approach, which could lead to that two tracks are associated with the same measurement. To avoid this problem it would be more beneficial to find the global minimum distance considering all tracks simultaneously, and restrict that a measurement can only be associated with one track. This approach is referred to as global nearest neighbor (GNN), and its aim is to solve

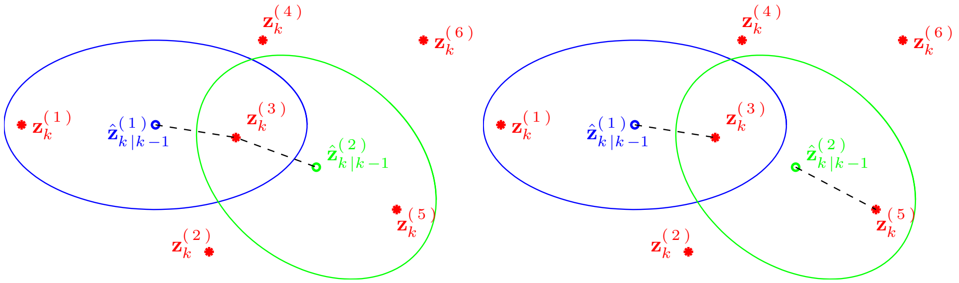
$$\min_{\lambda} = \sum_{i}^{N_k} d_{i,\lambda_i}^2 \tag{4.7}$$

where $\lambda_i \in \{0, 1, \ldots, N\}$ indicates which measurement has been assigned to track i. Note that the value $\lambda_i = 0$ means that track i has not been associated with any measurement. The so called auction algorithm is the most well known method to solve this problem, see Bertsekas (1990). Example 4.1 illustrated gating and a comparison between NN and GNN.

4.1 Example: Gating and Nearest Neighbor Data Association

Consider the target tracking example introduced in Example 1.2. Assume that two vehicles are currently tracked. Their predicted measurements $\hat{\mathbf{z}}_{k|k-1}^{(1)}$ and $\hat{\mathbf{z}}_{k|k-1}^{(2)}$ are shown in blue and green in Figure 4.3. Six measurements $\mathbf{z}_k^{(j)}$, $j = 1, \ldots, 6$ are collected at time k and shown as red stars in the figure. The measurements 2, 4 and 6 fall outside the gates, which are illustrated with ellipses. Measurement 1 and 3 may be obtained from target 1 since they are both in its gate. However, measurement 3 is also in the gate of target 2. The NN data association considers each track individually, and since measurement 3 is the closes measurements of both targets it is used to update both targets as shown in Figure 4.3a. The GNN data association avoids this problem and considers all tracks simultaneously. The result is shown in Figure 4.3b, where target 2 is associated with measurement 5 instated.

There exist a further group of association methods, which uses all measurements that fall inside the gate. In these methods each measurement j is weighted in accordance with the probability that it originates from track i. The two most well known algorithms are probabilistic data association (PDA) (Bar-Shalom and Tse, 1975) and joint probabilistic data association (JPDA) (Fortmann et al., 1983),



(a) *NN*

(b) *GNN*

Figure 4.3: *An example with two targets (blue and green) and six measurements (in red). The ellipses are the gates and the two figures show the difference between the association methods NN and GNN.*

where the difference lies in that PDA considers each track separately, whereas JPDA forms global hypotheses to calculate the probabilities. Note that the number of targets is assumed fixed in the above mentioned association methods. These methods are not further discussed in this thesis, hence a detailed description is omitted at this place.

4.2.2 Track Management

The main task of the track management is to decide on how many true targets are observed. Only tracks of sufficient quality should be considered as valid tracks and used in the data association algorithms described above. According to their different life stages, tracks can be classified into three cases.

Tentative track: A track that is in the track initiation process. It is not sure that there is sufficient evidence that it is actually a target or not. A tentative track is started as soon as it is likely that a measurement originates from a new target. The tentative tracks can be associated to further measurements.

Confirmed track: A track that was decided to belong to a valid target in the surveillance area. If several sequential arriving observations indicate that the tentative track resembles a true target rather than noise or clutter it is decided to be a confirmed track.

Deleted track: At the other end of the initiation process, this is a track that is decided to come from all random false alarm. It can also be a track that is no longer visible to any sensor when the uncertainty of the target increases above a given threshold. All of its information should be deleted.

When starting a new track some initial guess about the properties of the track must be made. Put into a Bayesian wording, a *prior density* $p(\mathbf{x}_0)$, that contains the information of the state before any observation is made, must be designed. This can sometimes be a simple task, for instance if you expect targets to appear from a certain place, but in most cases it is not easy to know beforehand. A practical and straightforward method is to use the first measurement as the mean value of the prior distribution. Other statistics of the prior distribution are still considered as design variables. A better approach may be considered in the Kalman filter, where the prior distribution is Gaussian and may be chosen as $\mathbf{x}_0 \sim \mathcal{N}(0, P_0)$. If P_0 is chosen very large in comparison with the measurement uncertainty, the first updated state estimate will largely be based on the associated measurement.

In order to know if the first measurement stems from a true target or from clutter, the track management system must observe the tentative track for a number of samples. There exist different methods to decide if a tentative track shall be transformed into a confirmed track. One of the simplest and most well known is the so called M/N logic, where a track is confirmed if M measurements out of N possible are associated with the track. Other common methods are based on scoring measurements, with for example *sequential probability ratio test* (SPRT), which was first proposed by Wald (1945). The method is based on comparing the ratio between the likelihood of the hypothesis that a track describes a true target and the likelihood of the hypothesis that it is a false alarm. Furthermore, there exist combined validation and association methods. Integrated probabilistic data association (IPDA) expresses both the probability of target existence and data association based on PDA.

The track deletion methods are similar to the validation methods, both the M/N logic and the score based approaches, mentioned above, can be applied. More information and details about track management can be found in various books about target tracking, see e.g., Blackman and Popoli (1999).

4.3 Extended Target Tracking

In classical target tracking problems the objects are modeled as point sources and it is assumed that only one measurement is received from each target at each time step. In automotive applications, the targets are at a close distance and of such a large size that individual features can be resolved by the sensor. A target is denoted extended whenever the target extent is larger than the sensor resolution, and it is large enough to occupy multiple resolution cells of the sensor. Put in other words, if a target should be classified as extended does not only depend on its physical size, but rather on the physical size relative to the sensor resolution.

The relevant target characteristics that are to be estimated form the target's state vector $\mathbf{x}$. Generally, beside the kinematic variables as position, velocity and orientation, the state vector may also contain information about the target's spatial extension. However, when the target's state does not contain any variables related to the target extent, though the estimation is done as if the target was a point, the algorithms should still take care of the multiple measurements that originate from a target. In Paper E a generalized definition of an extended target is presented, and it is repeated below. This definition does not depend on whether the target extent is estimated or not.

4.2 Definition (Extended Target). A target which potentially gives rise to more than one measurement per time step irrespective of whether the target's extent is explicitly modeled (and/or estimated) or not is referred to as an extended target.

The methods used to track extended target are very similar to the ones used for tracking a group of targets moving in formation. Extended target tracking and group tracking are thoroughly described in e.g., Ristic et al. (2004). The bibliography of Waxman and Drummond (2004) provides a comprehensive overview of existing literature in the area of group and cluster tracking. There exist some different approaches to represent, i.e., to model, the extended targets, of which five methods are described in this section.

4.3.1 Point Features

The first and most traditional method is to model the target as a set of point features in a target reference frame, each of which may contribute at most one sensor measurement. The exact location of a feature in the target reference frame is often assumed uncertain. However, if the appearance of the target is known and especially if typical radar reflection points are known, then the location of the features in the target reference frame can be assumed known. The motion of an extended target is modeled through the process model in terms of the translation and rotation of the target reference frame relative to a world coordinate frame, see e.g., Dezert (1998).

For an application in two dimensions, the point features are defined as

$$\mathbf{P}^T = \left\{\mathbf{p}^{(i)}\right\}_{i=1}^{N_p} \quad \text{with} \quad \mathbf{p}^{(i)} = \begin{bmatrix} \mathrm{x}^{(i)} & \mathrm{y}^{(i)} \end{bmatrix}^{\mathrm{T}} \tag{4.8}$$

and these are usually expressed in a target fixed coordinate frame T. The position $\boldsymbol{d}_{TW}^{W} = \begin{bmatrix} \mathrm{x}_{TW}^{W} & \mathrm{y}_{TW}^{W} \end{bmatrix}^{\mathrm{T}}$ of the target's origin and the orientation ψ_T of the target's frame is tracked relative to the world coordinate frame. The state vector may be defined as

$$\mathbf{x} = \begin{bmatrix} \boldsymbol{d}_{TW}^{W} & \psi_T & \mathbf{P}^T \end{bmatrix}^{\mathrm{T}}. \tag{4.9}$$

The point features in the target's coordinate frame can be mapped into a point in the world frame through the transform

$$\mathbf{p}^{(i),W} = R^{WT}\mathbf{p}^{(i),T} + \boldsymbol{d}_{TW}^{W}. \tag{4.10}$$

The equation above constitutes the measurement model if the point measurements are expressed in the world coordinate frame W. The rotation matrix is given by

$$R^{WT} = \begin{bmatrix} \cos\psi_T & -\sin\psi_T \\ \sin\psi_T & \cos\psi_T \end{bmatrix}. \tag{4.11}$$

The uncertainty about the exact position of the point feature is modeled according to

$$p(\mathbf{P}^W|\boldsymbol{d}_{TW}^W, \psi_T) = \prod_{i=1}^{N_p} \mathcal{N}(p^{(i),W}|R^{WT}(\psi_T)p^{(i),T} + \boldsymbol{d}_{TW}^W, w_p I_2), \tag{4.12}$$

which means that the uncertainty is assumed isotropic around the mean location of the point and with known variance w_p.

At each time step a set of $N_\mathbf{z}$ measurements $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^{N_\mathbf{z}}$ is received and has to be associated to the states. Not all measurements arise from a point feature, some are due to false detections (clutter). The association hypotheses are derived through some data association algorithm. In Vermaak et al. (2005) a method is proposed where the association hypotheses are included in the state vector and the output of the tracking filter is a joint posterior density function of the state vector and the association hypotheses. Furthermore, a multi-hypothesis likelihood is obtained by marginalizing over all the association hypotheses. An alternative solution is also proposed using a particle filter, where the unknown hypotheses are sampled from a well designed proposal density function.

An automotive radar sensor model developed for simulation purposes is proposed in Bühren and Yang (2006), where it is assumed that radar sensors often receive measurements from specific reflection centers on a vehicle. These reflection centers can be tracked in a filter and valuable information regarding the vehicle's orientation can be extracted as shown by Gunnarsson et al. (2007). A difficulty in solving the data association problem is the large number of association hypotheses available. To reduce the complexity Gunnarsson et al. (2007) proposes an approach where detections are associated with reflector groups. The spatial Poisson distribution, discussed in the subsequent section, is considered to be inappropriate, since the number of vehicle detections is assumed essentially known and not adequately modeled by a Poisson process.

In paper F the point features are denoted measurement generating points (MGP), and they are positioned on the surface of the target. In the referred publication, the MGPs are not considered having a fixed position on the surface, but they are instead defined as a random finite set of points on the one-dimensional surface of the target. The positions of the points on the surface are estimated in the target tracking filter, but they are of less importance, since they only serve as a means to estimate the position, shape and size of the entire target.

4.3.2 Spatial Distribution

Instead of modeling the target as a number of point features, which are assumed to be explicit measurement sources, the target may also be represented by a spatial probability distribution. It is more likely that a measurement comes from a region of high spatial density than from a sparse region. In Gilholm and Salmond (2005); Gilholm et al. (2005) it is assumed that the number of received target and clutter measurements are Poisson distributed, hence several measurements may originate from the same target. Each target related measurement is an independent sample from the spatial distribution. The spatial model could be a bounded distribution, such as a uniform PDF or an unbounded distribution, such as a Gaussian. The Poisson assumption allows the problem, or more specifically the evaluation of the likelihood, to be solved without association hypotheses. The spatial distribution is preferable where the point source models are poor representations of reality, that is in cases where the measurement generation is diffuse.

In Gilholm and Salmond (2005) two simple examples are given. One where the principle axis of the extended target is aligned with the velocity vector, i.e., the target is represented by a one dimensional uniform stick model. In the other example, a Gaussian mixture model is assumed for the target. A Kalman filter implementation with explicit constructions of assignment hypotheses is derived from the likelihood in Gilholm and Salmond (2005), whereas in Gilholm et al. (2005), a particle filter is applied directly given the likelihood which is represented by the Poisson spatial model of the stick. Hence, the need to construct explicit measurement-target assignment hypotheses is avoided in Gilholm et al. (2005). Swain and Clark (2010) proposes instead a standard measurement model but represents instead the extended targets as a spacial cluster process.

Boers et al. (2006) presents a similar approach, but since raw data is considered, no data association hypotheses are needed. The method to use raw data, i.e., consider the measurements without applying a threshold, is referred to as track before detect. A one dimensional stick target is assumed also by Boers et al. (2006), but unlike Gilholm and Salmond (2005), the target extent is assumed unknown. The state vector is given by the stick's center position and velocity as well as the stick's extension according to

$$\mathbf{x} = \begin{bmatrix} \mathsf{x} & \mathsf{y} & \dot{\mathsf{x}} & \dot{\mathsf{y}} & \mathsf{l} \end{bmatrix}^{\mathrm{T}}. \tag{4.13}$$

The process model is a simple constant velocity model and the length l is modeled as a random walk. The likelihood function is given by the probability distribution

$$p(\mathbf{z}|\mathbf{x}) = \int p(\mathbf{z}|\tilde{\mathbf{x}})p(\tilde{\mathbf{x}}|\mathbf{x})d\tilde{\mathbf{x}}, \tag{4.14}$$

where the spatial extension is modeled by the PDF $p(\tilde{\mathbf{x}}|\mathbf{x})$ and $\tilde{\mathbf{x}}$ is assumed to be a point source from an extended target with center given by the state vector $\mathbf{x}$. Hence, a measurement is received from a source $\tilde{\mathbf{x}}$ with likelihood $p(\mathbf{z}|\tilde{\mathbf{x}})$.

4.3.3 Elliptical Shaped Target

In many papers dealing with the shape of a target it is assumed that the sensor, e.g., radar, is also able to measure one or more dimensions of the target's extent. A high-resolution radar sensor may provide measurements of a targets down-range extent, i.e., the extension of the objects along the line-of-sight. The information of the target's extent is incorporated in the tracking filter and aids the tracking process to maintain track on the target when it is close to other objects.

An elliptical target model, to represent an extended target or a group of targets, is proposed in Drummond et al. (1990). The idea was improved by Salmond and

Parr (2003), where the sensor not only provides measurements of point observations, but rather range, bearing and down-range extent. The prime motivation of the study is to aid track retention for closely spaced moving targets. Furthermore, the state vector includes the position, velocity and the size of the ellipse. An EKF is used in Salmond and Parr (2003), but it is concluded that the filter may diverge under certain conditions, since the relation between the down-range extent measurement of the target and the position and velocity coordinates in the state vector is highly nonlinear. The same problem is studied in Ristic and Salmond (2004), where a UKF is implemented and tested. Even though the UKF shows better performance it is concluded that neither the EKF nor the UKF are suitable for this problem. The problem is further studied by Angelova and Mihaylova (2008), where other filter techniques, based on Monte Carlo algorithms, are proposed. In this paper the size of the ellipse takes values from a set of standard values, i.e., the algorithm estimates the type of object from a list, under the assumption that typical target sizes are known.

A group of objects moving collectively may also be modeled as an extended target. The ellipse model is used to model a formation of aircraft in Koch (2008). The object extension is represented by a symmetric positive definite random matrix, however, the measurement error is not considered in this publication. Improvements of this approach, including the consideration of sensor error, has been published by Feldmann et al. (2011). An alternative measurement model and an extesion using principal components is proposed by Degerman et al. (2011). Wieneke and Davey (2011) show how the random matrix approach can be used to track multiple extended targets directly from image data.

The concept of random hypersurface model, introduced by Baum and Hanebeck (2009), assumes that each measurement source is an element of a randomly generated hypersurface. In this publication an elliptical target shape is used to exemplify the approach. Improvements of the approach has been published in Baum et al. (2010), and a generalization which allows for tracking detail shapes, based on star convex shaped extended targets, is presented in Baum and Hanebeck (2011).

4.3.4 Curved Target

In Paper C the road borders are modeled as extended targets in the form of curved lines. A curved line is expressed as a third order polynomial in its coordinate frame. Since the road borders are assumed to be stationary, the frames are not included in the state vector. Furthermore, stationary points such as delineators and lamp posts are also modeled in Paper C. The nearest neighbor algorithm is used to associate measurements from stationary observations $\mathbf{z}^{(m)}$ to the targets. Here it is assumed that an extended line target $\mathrm{L}^{(j)}$ can give rise to several measurements, but a point target $\mathrm{P}^{(i)}$ can only contribute to one measurement. Since the likelihood of a line $\ell_{m,j}$ is a one dimensional spatial density function, but the likelihood of a point $\ell_{m,i}$ is given by a two dimensional density function, a likelihood ratio test is applied to determine the measurement-to-track association problem.

The likelihood ratio for a measurement $\mathbf{z}^{(m)}$ is given by

$$\Lambda(\mathbf{z}^{(m)}) \triangleq \frac{\sqrt{\ell_{m,i}}}{\ell_{m,j}}. \tag{4.15}$$

where the square root of the point likelihood is taken for unit matching, since the unit of the point likelihood is (distance)$^{-2}$ and the unit of the line likelihood is (distance)$^{-1}$. The corresponding *likelihood ratio test* is

$$\Lambda(\mathbf{z}^{(m)}) \underset{H_1}{\overset{H_0}{\gtrless}} \eta, \tag{4.16}$$

where H_0 and H_1 correspond to hypotheses that the measurement $\mathbf{z}^{(m)}$ is associated to the point P_i and to the line L_j, respectively. The threshold is selected experimentally. More theory about likelihood ratio test is given by e.g., van Trees (1968).

4.3.5 Extended Target Tracking and PHD filter

In the recent work Mahler (2009a) an extension of the (PHD) filter to also handle extended targets of the type presented in Gilholm et al. (2005) is given. The PHD filter is the topic of next section and therefore it will only be briefly summarized here. In Paper E a Gaussian-mixture implementation of the PHD-filter for extended targets called the extended target GM-PHD-filter (ET-GM-PHD) is presented. It is shown that this filter works well in most situation, but that it is sensitive in estimating the number of targets in a few situation e.g., when occlusion occurs. Therefore, generalization of Mahler's work has been made to derive the cardinalized PHD (CPHD) filter for extended targets, presented by Orguner et al. (2011). In addition to the derivation, a Gaussian mixture implementation for the derived CPHD filter is presented. This filter has less sensitive estimates of the number of targets. Early results on laser data are shown which illustrates robust characteristics of the CPHD filter compared to its PHD version.

Initial step have also been taken towards including estimation of target extent in the GM-PHD-filter, see Granström et al. (2011b). The state vector is augmented with a discrete state describing the type of object e.g., rectangle, ellipse etc. In Paper F a hybrid state space is introduced, where MGPs and the measurements are modeled by random finite sets and target states by random vectors. For each realization of the state vector, a PHD filter is utilized for estimating the conditional set of MGPs given the target states.

5

Probability Hypothesis Density Filter and Its Implementation

In the previous chapter it was shown how single target tracking filters can be extended and used to handle multiple targets by enclosing the filters with a target management framework. In this way measurements are associated to targets, which are tracked as isolated entities. To be able to perform the data association it is assumed that the number of present targets is known. This is rarely true, and therefore a track management logic estimated the number of valid and confirmed tracks. These methods explicitly separate the estimation of the number of targets and estimation of target states, which is suboptimal. *Random finite set* (RFS) formalism introduces a more rigorous approach to the multi target tracking problem. A set contains the random number of single target tracks, of which each is a random vector. The state vectors are elements of the RFS and their change in time is described with a motion model. The overall tracking problem is to compute the posterior density of the set-valued quantity, which also enables the approach to describe the uncertainty in the number of objects. The ordering of each single target state is not of importance in the the set notation; the estimation problem is reduced to capture the most essential part of the multi target tracking problem, estimating the number of objects and their individual states.

This chapter is split into three sections, which takes the estimation problem from a pure theoretical view into a practical implementation. The chapter begins in Section 5.1 with an introduction to the Bayes formulation of the RFS filter, which propagates the complete density of the RFS. This approach is not practically implementable, and therefore a solution is to only propagate the first order moment, called the *probability hypothesis density* (PHD), as described in Section 5.2. There exist a few practical representations of the PHD, of which the Gaussian mixture has turned out to be the most well-used during the last years. The *Gaussian mixture PHD* filter (GM-PHD) is the topic of Section 5.3. This chapter aims at sum-

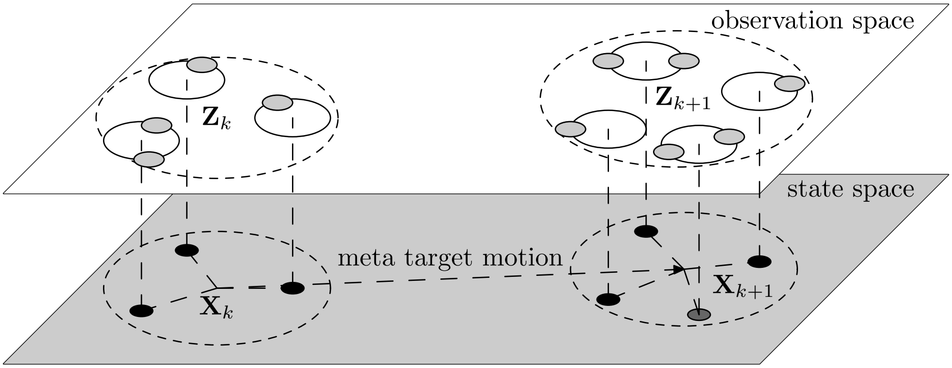


Figure 5.1: *Illustration of the RFS of states and measurements at time k and $k+1$. Note that this is the same setup as previously shown for the standard multitarget case in Figure 4.2.*

marizing the basic ideas and implementation methods, more details can be found in the textbook by Mahler (2007a), and other overviews are given by Mahler (2009b); Challa et al. (2011). Point process theory is described by e.g., Daley and Vere-Jones (2003); Streit (2010).

5.1 Introduction to Finite Set Statistics

The RFS estimation problem is in general solved in the same way as the single target tracking problem is approached, i.e., using Bayes' theorem. However, with the difference that the density is defined on a set rather than on a vector. *Finite set statistics* (FISST), introduced by Goodman et al. (1997); Mahler (2004), is a multisource-multitarget differential and integral calculus based on the so called belief-mass function, and the fact that it is the multisensor-multitarget counterpart of the probability-mass function for single target.

The conceptual idea of FISST is to redefine the target set as a single target, a so called meta target, with multitarget state

$$\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N_\mathrm{x})}\}, \tag{5.1}$$

a so called meta state; and similarly to redefine the observation set

$$\mathbf{Z} = \{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots, \mathbf{z}^{(N_\mathrm{z})}\}, \tag{5.2}$$

as a single measurement, a meta measurement, of the observed meta targets. The concept is illustrated in Figure 5.1, where meta targets are shown in the state space in the lower layer and meta observations in the observation space in the upper layer. Furthermore, the multitarget multisensor data can be modeled using a multisensor-multitarget measurement model

$$\mathbf{Z}_k = H(\mathbf{X}_k) \cup \mathbf{C}_k, \tag{5.3}$$

where $H(\mathbf{X}_k)$ is the RFS of measurements originating from the true target and $\mathbf{C}$ is the clutter RFS. To see the similarity, compare this multitarget model with the single target measurement model (2.1a). The motion of the meta target can be modeled using a multitarget motion model

$$\mathbf{X}_{k+1} = F(\mathbf{X}_k) \cup \mathbf{B}_k \tag{5.4}$$

where $F(\cdot)$ models the change of a RFS from time k to $k+1$, and $\mathbf{B}$ is the RFS of born targets. Compare, also here, with the single target motion model (2.1b). Given this the multisensor, multitarget estimation problem can be reformulated into a single-sensor, single-target problem. The RFS is formally defined in Section 5.1.1, followed by a definition of the belief-mass function and multitarget density function in Section 5.1.2. The multisource multitarget Bayes filter is described in Section 5.1.3.

5.1.1 Random Finite Set

Before continuing describing the FISST and PHD a formal definition of a RFS is given below.

5.1 Definition (random finite set). A *random finite set* (RFS) Ψ is a random variable that has realizations $\Psi = Y \in \mathfrak{Y}$ where the hyperspace $\mathfrak{Y}$ is the set of all finite subsets of some underlying space $\mathfrak{Y}_0$.

Since the definition may seem a bit abstract it is illustrated with a simple example, related to target tracking.

5.2 Example: Target Tracking

Consider the target tracking example, which was introduced in Example 1.2. In the example the state vector represents the Cartesian position of a target, i.e., $n_x = 2$. The state space is defined as an Euclidean vector space, i.e., $\mathbf{x} \in \mathbb{R}^2$. This state space is the underlying space $\mathfrak{Y}_0$ in the definition above. The hyperspace $\mathfrak{Y}$ includes all finite subsets of $\mathfrak{Y}_0 = \mathbb{R}^{n_x}$. Let the state variable vectors be $\mathbf{x}^{(i)} \in \mathfrak{Y}_0 = \mathbb{R}^2$ for $i = 1, \ldots, \infty$, then some realizations X of the random set $\mathbf{X}$ can be $X = \emptyset$ (no targets), $X = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\}$ (two targets with states $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$), or $X = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N_\mathbf{x})}\}$ ($N_\mathbf{x}$ targets with states $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N_\mathbf{x})}$).

5.1.2 Belief-Mass and Multitarget Density Function

The reformulation of the multisensor multitarget estimation problem as a single-metasensor, single-metatarget problem is based on the concept of belief-mass. Belief-mass functions are nonadditive generalizations of probability-mass function. In order to let the content be more easily digested, the multitarget notation is accompanied with the single target analogous notation.

Starting with some simple notation taken from single target tracking (cf. Section 4.1). For a random vector $\mathbf{y}$ the *probability mass function* $p_\mathbf{y}(\mathbb{S})$ gives the probability of $\mathbf{y}$ being in some subset of the region $\mathbb{S} \subseteq \mathfrak{Y}_0$, i.e.,

$$p_\mathbf{y} = \Pr(\mathbf{y} \in \mathbb{S}). \tag{5.5}$$

The probability density function $p_{\mathbf{y}}(y)$ describes the likelihood of $\mathbf{y}$ to occur at a given point y. The relation between the mass function and the density function is given by

$$p_{\mathbf{y}}(y) = \frac{dp_{\mathbf{y}}}{dy}. \tag{5.6}$$

The probability mass function for a random vector can be generalized to the belief-mass function for RFS. The belief-mass function is denoted $\beta_{\Psi}(\mathbb{S})$, and is defined as the probability that the random finite set Ψ on $\mathfrak{Y}_0$ is within some region $\mathbb{S}$, $\beta_{\Psi}(\mathbb{S}) = \Pr(\Psi \subseteq \mathbb{S})$. Similarly, the probability density function for a random vector can be generalized to the probability density function $p_{\Psi}(Y)$ of a random finite set Ψ. The relation between the probability density function $\Psi p_{\Psi}(Y)$ of a random finite set and the belief-mass function $\beta_{\Psi}(S)$ is given by

$$\beta_{\Psi}(S) \int\limits_{\mathbb{S}} p_{\Psi}\,(Y)\,\delta Y = \Pr\,(\Psi \subseteq \mathbb{S}) \tag{5.7}$$

and

$$p_{\Psi}(Y) = \frac{\delta \beta_{\Psi}}{\delta Y}(\emptyset). \tag{5.8}$$

Here, $\int \cdot\, \delta Y$ denotes the *set integral*, and $\frac{\delta\, \cdot}{\delta Y}$ denotes the *set derivative*. The definitions of the set integral and the set derivative are too complicated to present here, the interested reader can consult Mahler (2007a).

These definitions have the following practical use when applied to the likelihood function and the Markov transition density

- In the single target case the probabilistic-mass function of the sensor model $p(\mathbb{S}|\mathbf{x}) = \Pr(\mathbf{z} \in S|\mathbf{x})$ is the probability that the random observation $\mathbf{z}$ will be found in a given region $\mathbb{S}$ if the target state is $\mathbf{x}$. Analogously, in the multitarget case the belief-mass function $\beta(\mathbb{S}|\mathbf{X}) = \Pr(\mathbf{Z} \subseteq \mathbb{S}|\mathbf{X})$ is the total probability that all observations will be found in any given region $\mathbb{S}$, if the multitarget state is $\mathbf{X}$. In the single target case the likelihood function is obtained by differentiation of the probabilistic-mass function. The same concept holds for the multitarget case, i.e., the true multitarget likelihood function $p(\mathbf{Z}|\mathbf{X})$ can, using FISST, be derived from $\beta(\mathbb{S}|\mathbf{X})$ according to

$$p_{k|k}(\mathbf{Z}_k|\mathbf{X}_k) = \frac{\delta \beta_{k|k}(\emptyset|\mathbf{X}_k)}{\delta \mathbf{Z}_k} \tag{5.9}$$

  where $\frac{\delta}{\delta \mathbf{Z}_k}$ is the set derivative.

- Similarly, the probability-mass $p_{k+1|k}(\mathbb{S}|\mathbf{x}) = \Pr(\mathbf{x}_{k+1} \in \mathbb{S}|\mathbf{x}_k)$ of a single target motion model is the probability that the target will be found in region $\mathbb{S}$ at time step $k+1$, given that it had state $\mathbf{x}_k$ at time step k. In the multitarget case the belief-mass function $\beta_{k+1|k}(\mathbb{S}|\mathbf{X}) = \Pr(\mathbf{X}_{k+1} \subseteq \mathbb{S}|\mathbf{X}_k)$ of a multitarget motion model is the total probability of finding all targets in region $\mathbb{S}$ at time $k+1$, if at time k they had a multi object state $\mathbf{X}_k$. By differentiating $p(\mathbb{S}|\mathbf{x}_k)$ the Markov transition density $p_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k)$ can be derived in the single target case. Using FISST the multitarget Markov density can be derived by taking the set derivative of the belief-mass function

$$p_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{X}_k) = \frac{\delta\beta_{k+1|k}(\emptyset|\mathbf{X}_k)}{\delta\mathbf{X}_{k+1}} \tag{5.10}$$

5.1.3 The Multitarget Bayes Filter

Consider the single target state propagation summarized in Section 4.1. The filter recursion is extended to the multitarget case under the FISST assumptions in this section. The single target random state variables $\mathbf{x}$ are substituted with the RFS of a set of targets. Before giving the filter recursion, the interpretation of a set probability function is discussed. Consider the multitarget state propagation

$$\cdots \rightarrow \underline{p_{k|k}(\mathbf{X}_k|\mathbf{Z}_{1:k})} \overset{\text{predictor}}{\rightarrow} p_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{Z}_{1:k}) \overset{\text{corrector}}{\rightarrow} p_{k+1|k+1}(\mathbf{X}_{k+1}|\mathbf{Z}_{1:k+1}) \rightarrow \cdots$$

The meaning of the underlined prior is exemplified below,

$p(\emptyset|\mathbf{Z}_{1:k})$	probability that there are no targets present
$p(\{\mathbf{x}_k^{(1)}\}|\mathbf{Z}_{1:k})$	likelihood of one target with state $\mathbf{x}_k^{(1)}$
$p(\{\mathbf{x}_k^{(1)},\mathbf{x}_k^{(2)}\}|\mathbf{Z}_{1:k})$	likelihood of two targets with state $\mathbf{x}_k^{(1)},\mathbf{x}_k^{(2)}$
$\vdots$	
$p(\{\mathbf{x}_k^{(1)},\ldots,\mathbf{x}_k^{(N_\mathbf{x})}\}|\mathbf{Z}_{1:k})$	likelihood of $N_\mathbf{x}$ targets with state $\mathbf{x}_k^{(1)},\ldots,\mathbf{x}_k^{(N_\mathbf{x})}$

i.e., the set density is not only a function of the state variable values, but also a function of the number of targets $N_\mathbf{x}$ in the set.

The single target Bayes filter equations were stated in (3.13), the multitarget Bayes filter counterpart has the form

$$p_{k|k}\left(\mathbf{X}_k|\mathbf{Z}_{1:k}\right) = \frac{p_k\left(\mathbf{Z}_{1:k}|\mathbf{X}_k\right)\cdot p_{k|k-1}\left(\mathbf{X}_k|\mathbf{Z}_{1:k-1}\right)}{\int p_k\left(\mathbf{Z}_{1:k}|\mathbf{X}_k\right)\cdot p_{k|k-1}\left(\mathbf{X}_k|\mathbf{Z}_{1:k-1}\right)\delta\mathbf{X}_k} \tag{5.11a}$$

$$p_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{Z}_{1:k}) = \int p_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{X}_k)\cdot p_{k|k}(\mathbf{X}_k|\mathbf{Z}_{1:k})\delta\mathbf{X}_k \tag{5.11b}$$

where $p_k\left(\mathbf{Z}_k|\mathbf{X}_k\right)$ is the multisource likelihood function, $p_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{X}_k)$ is the multitarget Markov transition function, $p_{k|k-1}\left(\mathbf{X}_k|\mathbf{Z}_{k-1}\right)$ is the multitarget prior, $p_{k|k}(\mathbf{X}_k|\mathbf{Z}_k)$ is the multitarget posterior and $p_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{Z}_k)$ is the multitarget prediction. The filter recursion is given by,

$$\begin{tikzcd}
\cdots \arrow[r] & p_{k|k}(\mathbf{X}_k|\mathbf{Z}_{1:k}) \arrow[r, "\text{predictor}"] & p_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{Z}_{1:k}) \arrow[r, "\text{corrector}"] & p_{k+1|k+1}(\mathbf{X}_{k+1}|\mathbf{Z}_{1:k+1}) \arrow[r] & \cdots \\
& & \begin{array}{c}\text{Markov density}\\ p(\mathbf{X}_{k+1}|\mathbf{X}_k)\end{array} \arrow[u] & \begin{array}{c}\text{Likelihood function}\\ p(\mathbf{Z}_{k+1}|\mathbf{X}_{k+1})\end{array} \arrow[u] & \\
& & \begin{array}{c}\text{Motion Model}\\ \mathbf{X}_{k+1} = \underbrace{F(\mathbf{X}_k)}_{\text{persisting}} \bigcup \underbrace{\mathbf{B}_k}_{\text{new}}\end{array} \arrow[u] & \begin{array}{c}\text{Measurement Model}\\ \mathbf{Z}_{k+1} = \underbrace{H(\mathbf{X}_{k+1})}_{\substack{\text{generated}\\ \text{observations}}} \bigcup \underbrace{\mathbf{C}_{k+1}}_{\text{clutter}}\end{array} \arrow[u] &
\end{tikzcd}$$

compare with the single target filter recursions given on the same form in Section 3.2. Note that the motion model describes a union of the set of persisting targets and the set of new born, or detected, targets. In the same manner, the measurement model describes a union of the set of target generated observations and the set of clutter measurement. Here, the measurement set **Z** is the set of all observations received by the sensor.

The conclusions of this section can be summarized in a few items, which will be furher addressed in the next section:

- The Bayesian filter on sets is very similar to standard Bayesian density recursion.
- Related single target densities and operations are replaced with their set equivalents, compare (5.11) with (3.13).
- A set integral $\int \cdot \, \delta\mathbf{X}$ is necessary to derive the posterior

$$p(\mathbf{X}_k|\mathbf{Z}_{1:k}) \propto p(\mathbf{Z}_k|\mathbf{X}_k) \int p(\mathbf{X}_k|\mathbf{X}_{k-1}) p(\mathbf{X}_{k-1}|\mathbf{Z}_{1:k-1}) \delta\mathbf{X}_{k-1}$$

- Therefore, the filter is computationally prohibitive.

5.2 Introduction to the PHD filter

The multisource multitarget filter as described in the last section is computationally demanding to implement. This section describes a new approach, first proposed by Mahler (2003), for approximate multitarget nonlinear filters called the *probability hypothesis density* (PHD) filter. The PHD is a first order moment approximation of the RFS. To describe this approximation the single target case will first be repeated in Section 5.2.1 and then compared with the multitarget case in Section 5.2.2, where also the PHD is defined. The PHD filter recursion is explained in Section 5.2.3. The section is concluded in Section 5.2.4 with a short literature survey of generalizations and extensions of the PHD filter.

5.2.1 Approximations in Single-Target Tracking

The single target density propagation is illustrated below

$$\begin{tikzcd}
\cdots \arrow[r] & p_{k|k}(\mathbf{x}_k|\mathbf{z}_{1:k}) \arrow[r, "\text{predictor}"] \arrow[d] & p_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{z}_{1:k}) \arrow[r, "\text{corrector}"] \arrow[d] & p_{k+1|k+1}(\mathbf{x}_{k+1}|\mathbf{z}_{1:k+1}) \arrow[r] \arrow[d] & \cdots \\
 & \hat{\mathbf{x}}_{k|k} \arrow[r, "\text{predictor}"] & \hat{\mathbf{x}}_{k+1|k} \arrow[r, "\text{corrector}"] & \hat{\mathbf{x}}_{k+1|k+1} &
\end{tikzcd}$$

At each time step the first order moment, i.e., the expected value may be collapsed from the density function. In the so called constant-gain filter, of which the alpha-beta filter is the most well known, only the expected value is propagated over time. This makes the filter computationally faster. Both the first and the second order moments are propagated in the standard Kalman filter, compare with Section 3.2.1. If the density functions are Gaussian, the first and second order moments are the sufficient statistics, hence propagating these variables describes the density functions sufficiently. However, if the density functions are not Gaussian, propagating the moments is only an approximation of the density.

5.2.2 Approximations in Multitarget Tracking

Based on the same analogy as with the single target tracking the density of the RFS my be compressed to the first order moments, and these may be propagated as follows:

$$\begin{tikzcd}
\cdots \arrow[r] & p_{k|k}(\mathbf{X}_k|\mathbf{Z}_{1:k}) \arrow[r, "\text{predictor}"] \arrow[d] & p_{k+1|k}(\mathbf{X}_{k+1}|\mathbf{Z}_{1:k}) \arrow[r, "\text{corrector}"] \arrow[d] & p_{k+1|k+1}(\mathbf{X}_{k+1}|\mathbf{Z}_{1:k+1}) \arrow[r] \arrow[d] & \cdots \\
 & D_{k|k}(\mathbf{x}_k|\mathbf{Z}_{1:k}) \arrow[r, "\text{predictor}"] & D_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{Z}_{1:k}) \arrow[r, "\text{corrector}"] & D_{k+1|k+1}(\mathbf{x}_{k+1}|\mathbf{Z}_{1:k+1}) &
\end{tikzcd}$$

The first order moment is called a PHD and denoted D. However, it is not obvious what the multitarget counterpart of an expected value is; a naïve defininition would be

$$\mathrm{E}(\Psi) = \int X \cdot p_\Psi(X)\delta X, \tag{5.12}$$

where $p_\Psi(X)$ is the multitarget probability density of a RFS Ψ. However, this integral is not mathematically defined, since addition of finite subsets is not usefully defined. A strategy is to transform subsets X into vectors T_X in some vector space. It is important that this transformation $X \to T_X$ preserves set-theoretic structures, i.e., transforming unions into sums $T_{X \cup X'} = T_X + T_{X'}$, whenever $X \cap X' = \emptyset$. An indirect expected value can now be defined according to

$$\mathrm{E}(\Psi) = \int T_X \cdot p_\Psi(X)\delta X. \tag{5.13}$$

The vector is commonly chosen as $T_X = \delta_X(\mathbf{x})$ with

$$T_X = \begin{cases} 0 & \text{if } X = \emptyset \\ \sum_{\mathbf{y} \in \mathbf{X}} \delta_{\mathbf{y}}(\mathbf{x}) & \text{otherwise} \end{cases} \tag{5.14}$$

where $\delta_{\mathbf{y}}(\mathbf{x})$ is the Dirac delta function with peak at $\mathbf{y}$. Given these assumptions, the multitarget analog of the expected value is given by

$$D_{\Psi}(\mathbf{x}) = \int \delta_X(\mathbf{x}) \cdot p_{\Psi}(X)\delta X. \tag{5.15}$$

To exemplify this rather theoretical discussion, consider a realization X_k of the RFS $\mathbf{X}_k = \{\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}, \dots, \mathbf{x}_k^{(N_{\mathbf{x}})}\}$. Define a scalar valued function of X_k according to

$$\delta_{X_k} = \sum_{i=1}^{N} \delta_{\mathbf{x}_k^{(i)}}(\mathbf{x}), \tag{5.16}$$

as illustrated in Figure 5.2a. Then, the probability hypothesis density (PHD) is the expectation of δ_{X_k} with respect to X_k

$$D_{k|k}(\mathbf{x}) = \mathrm{E}(\delta_{X_k}) = \int \delta_{X_k} \cdot p_{\Psi}(X)\delta X. \tag{5.17}$$

The expected value is illustrated in Figure 5.2b. Note also that the expected number of targets in the volume $\mathbb{S}$ is

$$\widehat{N}_{\mathbf{x}} = \int_{\mathbb{S}} D_{k|k}(\mathbf{x})d\mathbf{x} \tag{5.18}$$

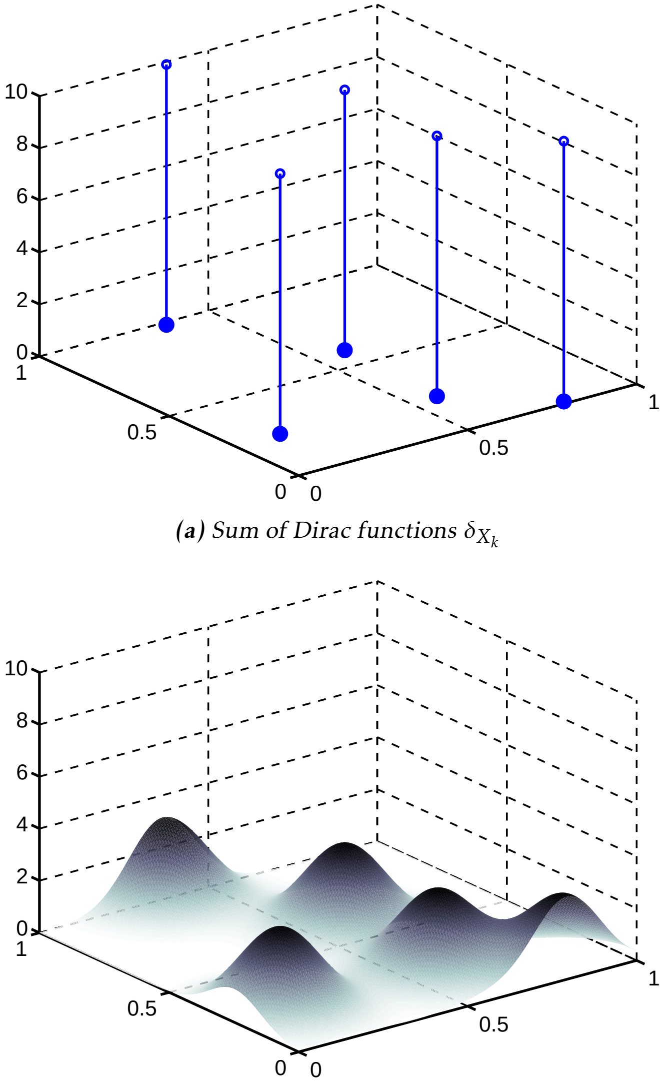
In fact one can define the PHD using the expected number of targets as follows: Given any region $\mathbb{S}$ of single target state space $\mathfrak{X}_0$ the integral $\int_{\mathbb{S}} D_{k|k}(\mathbf{x})d\mathbf{x}$ is the expected number of targets in $\mathbb{S}$.

The PHD is an *intensity function*, which models the joint density over the states. The PHD is multi-modal, with peaks near the actual targets, see Figure 5.3. This figure shows the PHD approximation of the RFS in Figure 5.1. The intensity function is used in Paper D to represent the density of stationary targets along the edges of a road. An autonomous driving vehicle should avoid areas with high intensity and the objective should be to keep the vehicle in the valleys of the intensity function.

5.2.3 The PHD Filter

The PHD filter consists of two equations, the predictor equation, which extrapolates the current PHD to the predicted PHD at the time of the new observation, and the corrector equation, which updates the predicted PHD with the new observations.

The PHD predictor equations takes care of the motion of existing targets, birth and spawn of new targets and the death of existing targets. The components of the prediction model are summarized below:



(a) Sum of Dirac functions δ_{X_k}

(b) Expectation $\mathrm{E}(\delta_{X_k})$

Figure 5.2: *Definition and example of the PHD, the Dirac functions represents the RFS and the surface the first order moment, i.e., the PHD.*

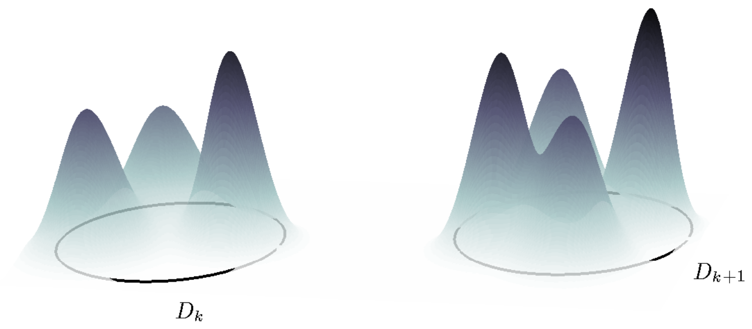


Figure 5.3: *Illustration of the PHD for the time steps k and $k+1$. This is the PDF of the RFS in Figure 5.1.*

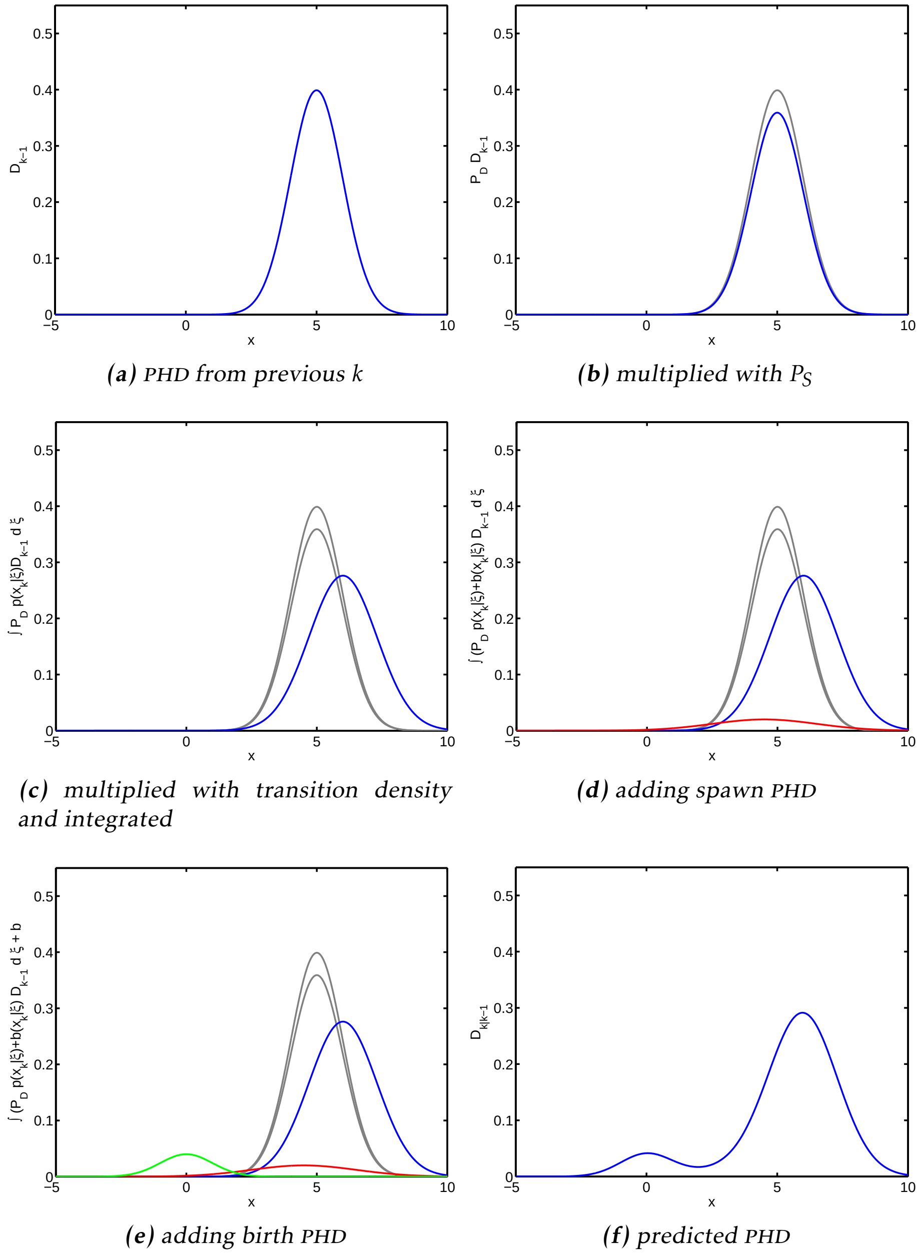
	model	**time step**	**description**
motion:	$p(\mathbf{x}_{k+1}\|\mathbf{x}_k)$	$\mathbf{x}_k \rightarrow \mathbf{x}_{k+1}$	likelihood that target will have state $\mathbf{x}_{k+1}$ if it had state $\mathbf{x}_k$
death:	$1 - P_S(\mathbf{x}_k)$	$\mathbf{x}_k \rightarrow \emptyset$	probability that target will vanish if it had state $\mathbf{x}_k$
spawn:	$b(\mathbf{X}_{k+1}\|\mathbf{x}_k)$	$\mathbf{x}_k \rightarrow \mathbf{X}_{k+1}$	likelihood that a target will spawn target set $\mathbf{X}_{k+1}$ if it had state $\mathbf{x}_k$
birth:	$b(\mathbf{X}_{k+1})$	$\emptyset \rightarrow \mathbf{X}_{k+1}$	likelihood that a target set $\mathbf{X}_{k+1}$ will appear in scene

In the table above $P_S(\mathbf{x})$ denotes the probability that a target with state $\mathbf{x}$ at time k will survive at time $k+1$, $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ is the single target Markov transition density, $b(\mathbf{X}_{k+1}|\mathbf{x}_k)$ is the PHD of targets spawned by other targets with state $\mathbf{x}$, and $b(\mathbf{X}_{k+1})$ is the PHD of new appearing targets. The PHD predictor equation is given by

$$\underbrace{D_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})}_{\text{time updated PHD}} =$$

$$= \underbrace{b_{k+1|k}(\mathbf{x}_{k+1})}_{\text{target birth PHD}} + \int \Big[\underbrace{P_S(\xi_k)}_{\substack{\text{probability of}\\ \text{survival}}} \underbrace{p(\mathbf{x}_{k+1}|\xi_k)}_{\substack{\text{Markov transition}\\ \text{density}}} + \underbrace{b(\mathbf{x}_{k+1}|\xi_k)}_{\substack{\text{target spawn by}\\ \text{existing targets}}} \Big] \underbrace{D_{k|k}(\xi_k|\mathbf{z}_{1:k})}_{\text{PHD from previous } k} \; d\xi_k. \tag{5.19}$$

The parts of the model are illustrated with a number of plots in Figure 5.4. One example of a PHD $D_{k|k}(\xi|\mathbf{z}_{1:k})$ from the previous time step k is illustrated in Figure 5.4a. The magnitude of the PHD is reduced a bit when multiplied with the probability of survival $P_S(\xi)$ in Figure 5.4b. Thereafter, multiplying with the Markov transition density and then integrating changes the mean value and the covariance, see Figure 5.4c. Note, that all target motions are assumed statistically independent. In Figure 5.4d, the spawn PHD $b(\mathbf{x}_{k+1}|\xi_k)$ is added in red, and in Figure 5.4e the birth PHD $b_{k+1|k}(\mathbf{x}_{k+1})$ in green. Finally, the total predicted PHD $D_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})$, after time update, is shown in Figure 5.5a.



(a) *PHD from previous k*

(b) *multiplied with P_S*

(c) *multiplied with transition density and integrated*

(d) *adding spawn PHD*

(e) *adding birth PHD*

(f) *predicted PHD*

Figure 5.4: *An example showing the components of the predictor equation* (5.19) *of PHD filter.*

The PHD corrector equations takes care of the likelihood of the existing targets, misdetection of targets and clutter. The components of the prediction model are summarized below:

	model	**update**	**description**
likelihood:	$p(\mathbf{z}_k\vert\mathbf{x}_k)$	$\mathbf{x}_k \to \mathbf{z}_k$	likelihood that target will generate observation $\mathbf{z}_k$ if it has state $\mathbf{x}_k$
misdetection:	$(1 - P_D(\mathbf{x}_k))$	$\mathbf{x}_k \to \emptyset$	state dependent probability that target will not generate an observation
clutter:	$c(\mathbf{Z}_k)$	$\emptyset \to \mathbf{Z}_k$	likelihood that a set $\mathbf{Z}_k = \{\mathbf{z}_k^{(1)}, \dots, \mathbf{z}_k^{(M)}\}$ of clutter observations will be generated

In the table above $P_D(\mathbf{x}_k)$ is the probability of detection of a target with the state $\mathbf{x}$ at time step k, $p(\mathbf{z}_k|\mathbf{x}_k)$ is the single target likelihood function and $c(\mathbf{Z}_k)$ is the density of Poisson false alarms, due to clutter. The modeling is done under the assumption that the observations and the clutter are statistically independent. Furthermore it is assumed that the multitarget prediction is approximately Poisson. Given a new scan of data $\mathbf{Z}_k = \{\mathbf{z}_k^{(1)}, \dots, \mathbf{z}_k^{(M)}\}$ the corrector equation is given by

$$\underbrace{D_{k|k}(\mathbf{x}|\mathbf{Z}_{1:k})}_{\text{updated PHD}} \approx \underbrace{\sum_{\mathbf{z}\in\mathbf{Z}_k} \frac{\Lambda_{k|k}(\mathbf{x}|\mathbf{z})}{\lambda_k c_k(\mathbf{z}) + \int \Lambda_{k|k}(\mathbf{x}|\mathbf{z}) d\mathbf{x}} + (1 - P_D(\mathbf{x}))}_{\text{pseudo-likelihood}} \underbrace{D_{k|k-1}(\mathbf{x}|\mathbf{Z}_{1:k-1})}_{\text{predicted PHD}} \tag{5.20}$$

where λ_k is the average number of false alarms. Furthermore

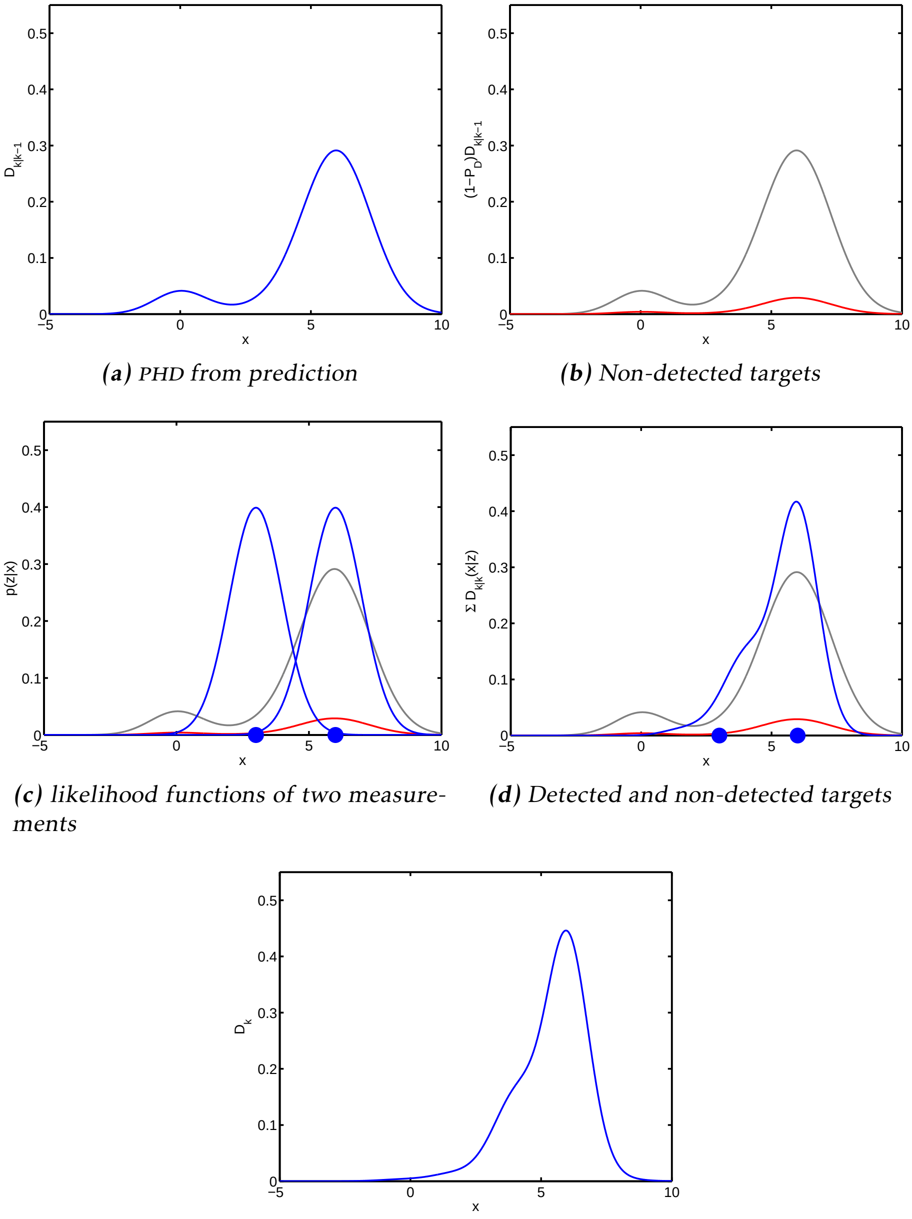
$$\Lambda_{k|k}(\mathbf{x}|\mathbf{z}) = P_D(\mathbf{x})p(\mathbf{z}|\mathbf{x})D_{k|k-1}(\mathbf{x}|\mathbf{z}_{1:k}). \tag{5.21}$$

The corrector equation is again exemplified by a series of plots in Figure 5.5. Figure 5.5a shows the predicted PHD. The non-detected targets are modeled by multiplying the predicted PHD with $(1 - P_D(\mathbf{x}))$, as illustrated in red in Figure 5.5b. Assume that two detections are made; the likelihood functions $p(\mathbf{z}|\mathbf{x})$ of these individual observations are shown in Figure 5.5c. The pseudo-likelihood function of the detections multiplied with the predicted PHD is shown in blue in Figure 5.5d. Finally, the updated posterior PHD is illustrated in Figure 5.5e.

In Paper E, the pseudo-likelihood function is modified to handle extended targets. This modification is one of the major contributions of that paper.

5.2.4 Generalizations of the PHD filter

There exists an improvement of the PHD filter, called the *cardinalized probability hypothesis density* (CPHD) filter, which also propagates the probability distribution on the target number. This filter generally performs better, but with increased computational load. The CPHD filter is too complex to be described in



(a) PHD *from prediction*

(b) Non-detected targets

(c) likelihood functions of two measurements

(d) Detected and non-detected targets

(e) Updated PHD

Figure 5.5: *Steps in the* PHD *update and corrector equations* (5.20).

the present chapter, and it is also not used in any of the publications in Part II. Details about the CPHD filter can be found in Mahler (2007b,a), and a Gaussian mixture implementation is described in Vo et al. (2007).

The standard PHD filter as described in this section does not maintain track labels from time step to time step. A so called peak to track association technique has been proposed by Lin et al. (2004); Panta et al. (2004) in order to maintain track labels.

Furthermore, PHD smoother has been studied in Mahler et al. (2010); Clark (2010), and a Gaussian mixture implementation is presented by Vo et al. (2010, 2011). Again, since the smoother is not further used in Part II, it is also not described here.

The so called *intensity filter*, developed by Streit, is another generalization of the PHD filter. The difference is that in the PHD filter certain prior assumptions regarding target birth and measurement clutter are made. More specifically the intensity filter uses an augmented single target state space, while the PHD filter uses only the standard single target state space. The augmented state space represents the absent target hypothesis, and it therefore allows the filter to on-line estimate the intensities of the target birth and measurement clutter Poisson point processes. Note that the PHD is an intensity function, and in the PHD filter the birth and clutter intensities are known *a priori*. A thorough description of the intensity filter is given in the textbook by Streit (2010) and a summary is given in the publications Streit and Stone (2008); Streit (2008).

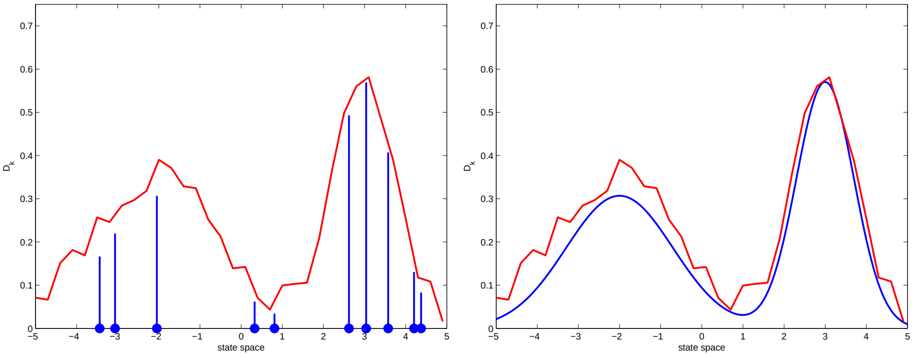
5.3 Gaussian Mixture Implementation

There exist primarily two different implementations of the PHD filter, the SMC approximation, see Vo et al. (2003); Sidenbladh (2003); Zajic and Mahler (2003), and the Gaussian mixture (GM) approximation, introduced in Vo and Ma (2006), and hence called GM-PHD filter. A physical interpretation of the PHD filter is proposed by Erdinc et al. (2009). Both approximations of the PHD are exemplified in Figure 5.6.

The advantage of the SMC approximation is that it can accommodate highly nonlinear motion and measurement models. The disadvantages are that it is computational demanding. The SMC based approximation is not further used in this thesis and it is therefore not described in further detail here, the interested reader is referred to Vo et al. (2003) or the textbook Mahler (2007a). The GM implementation is discussed in more detail in this section.

The GM-PHD filter has the following properties,

- It is less computationally demanding than the SMC solution.
- It is exact, i.e., it provides a true closed-form algebraic solution.
- It is easy to implement,



***(a)** Sequential Monte Carlo PHD approximations (SMC-PHD)*

***(b)** Gaussian mixture PHD approximation (GM-PHD)*

***Figure 5.6:** Practical implementation methods of the PHD.*

- however, measurement and motion models must be linear-Gaussian.

This section is outlined as follows. The approximations made in the GM-PHD filter are summarized in Section 5.3.1 and the algorithm is given in Section 5.3.2. Some practical details regarding the implementation, e.g., target extraction are described in Section 5.3.3.

5.3.1 Gaussian Mixture PHD Approximation

Similarly, as first shown for the theoretical Bayesian multitarget filter in Section 5.1.3 and then for the more practical standard PHD filter in Section 5.2.2, the easily implementable filter recursion of the GM-PHD filter is summarized below:

$$\begin{tikzcd}
\cdots \arrow[r] & D_{k|k}(\mathbf{x}|\mathbf{Z}_{1:k}) \arrow[r, "\text{predictor}"] \arrow[d] & D_{k+1|k}(\mathbf{x}|\mathbf{Z}_{1:k}) \arrow[r, "\text{corrector}"] \arrow[d] & D_{k+1|k+1}(\mathbf{x}|\mathbf{Z}_{1:k+1}) \arrow[d] \\
\cdots \arrow[r] & \sum_{i=1}^{J_{k|k}} w^{(i)}_{k|k}\mathcal{N}\left(\mathbf{x};\mu^{(i)}_{k|k},P^{(i)}_{k|k}\right) \arrow[r] \arrow[d] & \sum_{i=1}^{J_{k+1|k}} w^{(i)}_{k+1|k}\mathcal{N}\left(\mathbf{x};\mu^{(i)}_{k+1|k},P^{(i)}_{k+1|k}\right) \arrow[r] \arrow[d] & \sum_{i=1}^{J_{k+1|k+1}} w^{(i)}_{k+1|k+1}\mathcal{N}\left(\mathbf{x};\mu^{(i)}_{k+1|k+1},P^{(i)}_{k+1|k+1}\right) \arrow[d] \\
& \mu^{(1)}_{k|k},\ldots,\mu^{(J_{k|k})}_{k|k} \arrow[r] & \mu^{(1)}_{k+1|k},\ldots,\mu^{(J_{k+1|k})}_{k+1|k} \arrow[r] & \mu^{(1)}_{k+1|k+1},\ldots,\mu^{(J_{k+1|k+1})}_{k+1|k+1} \\
& P^{(1)}_{k|k},\ldots,P^{(J_{k|k})}_{k|k} \arrow[r] & P^{(1)}_{k+1|k},\ldots,P^{(J_{k+1|k})}_{k+1|k} \arrow[r] & P^{(1)}_{k+1|k+1},\ldots,P^{(J_{k+1|k+1})}_{k+1|k+1} \\
& w^{(1)}_{k|k},\ldots,w^{(J_{k|k})}_{k|k} \arrow[r] & w^{(1)}_{k+1|k},\ldots,w^{(J_{k+1|k})}_{k+1|k} \arrow[r] & w^{(1)}_{k+1|k+1},\ldots,w^{(J_{k+1|k+1})}_{k+1|k+1}
\end{tikzcd}$$

The PHD is represented by a Gaussian mixture and the summary statistics of the Gaussian components, i.e., the mean value μ and the covariance P are propagated together with a weight w for each Gaussian component. The number of Gaussian components at each time step is denoted J. The prior, predicted and posterior GM-PHD are shown above.

When deriving the GM-PHD filter some assumptions are made, see Vo and Ma (2006). Since these assumptions are formally repeated in Paper E, on Page 230, these are only summarized here. It is assumed that each target follows a linear Gaussian motion model,

$$p_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_{k+1}; F_k\mathbf{x}_k, Q_k), \tag{5.22}$$

where F_k is the state transition matrix and Q_k is the process noise covariance, and it is assumed that all target motions are statistically independent. Observations and clutter are assumed statistically independent and the sensor is assumed to have a linear Gaussian measurement model, i.e.,

$$p_{k|k}(\mathbf{z}_k|\mathbf{x}_k) = \mathcal{N}(\mathbf{z}_k; H_k\mathbf{x}_k, R_k), \tag{5.23}$$

where H_k is the measurement model matrix and R_k is the observation noise covariance. Furthermore it is assumed that the survival and detection probabilities are state independent, i.e., $P_S(\mathbf{x}) = P_S$ and $P_D(\mathbf{x}) = P_D$. The PHD of the birth and the spawn are Gaussian mixtures

$$b(\mathbf{X}_{k+1}) = \sum_{i=1}^{J_{b,k}} w_{b,k}^{(i)} \mathcal{N}(\mathbf{x}; \mu_{b,k}^{(i)}, P_{b,k}^{(i)}), \tag{5.24}$$

$$b(\mathbf{X}_{k+1}|\mathbf{x}_k^{(i)}) = \sum_{\ell=1}^{J_{\beta,k+1}} w_{\beta,k+1}^{(\ell)} \mathcal{N}(\mathbf{x}; \mu_{\beta,k+1|k}^{(i,\ell)}, P_{\beta,k+1|k}^{(i,\ell)}), \tag{5.25}$$

where $w_{b,k}^{(i)}$, $\mu_{b,k}^{(i)}$ and $P_{b,k}^{(i)}$ are the weight, the mean and the covariance of the ith born component, $i = 1, \ldots, J_{b,k}$, and $J_{b,k}$ is the number of components. Further, $w_{\beta,k+1}^{(\ell)}$, $\mu_{\beta,k+1|k}^{(i,\ell)}$ and $P_{\beta,k+1|k}^{(i,\ell)}$ are the weight, mean and covariance of the ℓth component $\ell = 1, \ldots, J_{\beta,k+1}$ at time $k+1$ spawned from the ith component at time k, and $J_{\beta,k+1}$ is the total number of components spawned from component i. It is also assumed that the predicted multitarget RFS are Poisson.

5.3.2 GM-PHD Filter Algorithm

In this section the GM-PHD filter algorithm including the predictor and the corrector equations, are given. The prediction equation is an approximation of the general PHD prediction equation (5.19), with the difference that the PHD components are substituted with Gaussian mixtures according to

$$\begin{aligned} D_{k|k-1}(\mathbf{x}|\mathbf{Z}_{1:k-1}) = &\sum_{j=1}^{J_{b,k}} w_{b,k}^{(j)} \mathcal{N}(\mathbf{x}; \mu_{b,k}^{(j)}, P_{b,k}^{(j)}) + P_S \sum_{j=1}^{J_{k-1}} w_{k-1}^{(j)} \mathcal{N}(\mathbf{x}; \mu_{k|k-1}^{(j)}, P_{k|k-1}^{(j)}) \\ &+ \sum_{j=1}^{J_{k-1}} \sum_{\ell=1}^{J_{\beta,k}} w_{\beta,k|k-1}^{(j,\ell)} \mathcal{N}(\mathbf{x}; \mu_{\beta,k|k-1}^{(j,\ell)}, P_{\beta,k|k-1}^{(j,\ell)}). \end{aligned} \tag{5.26}$$

In the case when the motion and sensor models are linear, the Gaussian components can be predicted using the prediction step of the Kalman filter (3.16). The illustration in Figure 5.4 still holds for this Gaussian case, and the discussion is therefore not repeated again. The corrector equation is an approximation of the general PHD corrector equation (5.20), with the difference that the PHD components are substituted with Gaussian mixtures according to

$$D_{k|k}(\mathbf{x}|\mathbf{Z}_{1:k}) \approx \underbrace{\sum_{i=1}^{J_{k|k-1}} w_{k|k}^{(i)} \mathcal{N}\left(\mu_{k|k}^{(i)}, P_{k|k}^{(i)}\right)}_{\text{not detected targets}} + \underbrace{\sum_{j=1}^{N_{\mathbf{z}}} \sum_{i=1}^{J_{k|k-1}} w_{k|k}^{(i,j)} \mathcal{N}\left(\mu_{k|k}^{(i,j)}, P_{k|k}^{(i,j)}\right)}_{\text{detected targets}} \tag{5.27}$$

The complete filter recursion, as it is presented by Vo and Ma (2006), is given in Algorithm 7.

Algorithm 7 Gaussian Mixture PHD filter

```
Require: the Gaussians $\{w_{k-1}^{(i)}, \mu_{k-1}^{(i)}, P_{k-1}^{(i)}\}_{i=1}^{J_{k-1}}$ and the measurement set $\mathbf{Z}_k$
    Prediction for birth and spawn targets
    i = 0
    for $j = 1$ to $J_{b,k}$ do
        $i = i + 1$
        $w_{k|k-1}^{(i)} = w_{b,k}^{(j)}, \quad \mu_{k|k-1}^{(i)} = \mu_{b,k}^{(j)}, \quad P_{k|k-1}^{(i)} = P_{b,k}^{(j)}$
    end for
    for $j = 1$ to $J_{\beta,k}$ do
        for $\ell = 1$ to $J_{k-1}$ do
            $i = i + 1$
            $w_{k|k-1}^{(i)} = w_{k-1}^{(\ell)} w_{\beta,k}^{(j)}$
            $\mu_{k|k-1}^{(i)} = F_{\beta,k-1}^{(j)} \mu_{k-1}^{(\ell)} + d_{\beta,k-1}^{(j)}, \quad P_{k|k-1}^{(i)} = Q_{\beta,k-1}^{(j)} + F_{\beta,k-1}^{(j)} P_{\beta,k-1}^{(\ell)} (F_{\beta,k-1}^{(j)})^{\mathrm{T}}$
        end for
    end for
    Prediction for existing targets
    for $j = 1$ to $J_{k-1}$ do
        $i = i + 1$
        $w_{k|k-1}^{(i)} = P_S w_{k-1}^{(j)}$
        $\mu_{k|k-1}^{(i)} = F_{k-1} \mu_{k-1}^{(j)}, \quad P_{k|k-1}^{(i)} = F_{k-1} P_{k-1}^{(j)} F_{k-1}^{\mathrm{T}} + Q_{k-1}$
    end for
    $J_{k|k-1} = i$
    Construction of PHD update components
    for $j = 1$ to $J_{k|k-1}$ do
        $\eta_{k|k-1}^{(j)} = H_k \mu_{k|k-1}^{(j)}, \quad S_k^{(j)} = R_k + H_k P_{k|k-1}^{(j)} H_k^{\mathrm{T}}$
        $K_k^{(j)} = P_{k|k-1}^{(j)} H_k^{\mathrm{T}} (S_k^{(j)})^{-1}, \quad P_{k|k}^{(j)} = (I - K_k^{(j)} H_k) P_{k|k-1}^{(j)}$
    end for
    Update non-detected targets
    for $j = 1$ to $J_{k|k-1}$ do
        $w_k^{(j)} = (1 - P_D) w_{k|k-1}^{(j)}$
        $\mu_k^{(j)} = \mu_{k|k-1}^{(j)}, P_k^{(j)} = P_{k|k-1}^{(j)}$
    end for
```

```
**Update detected targets**
$\ell = 0$
**for** each $\mathbf{z} \in \mathbf{Z}_k$ **do**
    $\ell = \ell + 1$
    **for** $j = 1$ **to** $J_{k|k-1}$ **do**
        $w_k^{(\ell J_{k|k-1}+j)} = P_D w_{k|k-1} \mathcal{N}(\mathbf{z}; \eta_{k|k-1}^{(j)}, S_k^{(j)})$
        $\mu_k^{(\ell J_{k|k-1}+j)} = \mu_{k|k-1}^{(j)} + K_k^{(j)}(\mathbf{z}_k - \eta_{k|k-1}^{(j)})$
        $P_k^{(\ell J_{k|k-1}+j)} = P_{k|k}^{(j)}$
    **end for**
    **for** $j = 1$ **to** $J_{k|k-1}$ **do**
        $w_k^{(\ell J_{k|k-1}+j)} = \frac{w_k^{(\ell J_{k|k-1}+j)}}{\lambda c(\mathbf{z})+\sum_{i=1}^{J_{k|k-1}} w_k^{(\ell J_{k|k-1}+i)}}$
    **end for**
**end for**
$J_k = \ell J_{k|k-1} + J_{k|k-1}$
**return** $\{w_k^{(i)}, \mu_k^{(i)}, P_k^{(i)}\}_{i=1}^{J_k}$
```

To allow for nonlinear motion and sensor models, the EKF or UKF can be utilized, in that case the prediction and update of the single Gaussian components are instead performed as described in Algorithm 4 and 5, respectively. The UKF version of the GM-PHD filter is used in Paper D to track the edges of a road and thereby to create an intensity map of the environment. An extension of the KF based GM-PHD used to track extended targets, is presented Paper E. In that paper modifications are made to the update step to account for the extended targets, which may give rise to more than one measurement per target and time step. In Paper F the GM-PHD filter is used as an inner core in a tracking filter which aims at estimating the size and shape of targets. The outer shell of the approach represents the position and motion of the target, whereas the PHD is defined on the surface of the target as a means to estimate its shape.

5.3.3 Merge, Prune and Extract Targets

A few approximations and modifications to the GM-PHD filter must be made in order to obtain a computationally realizable filter. The number of Gaussian components grows rapidly, since each predicted Gaussian component is updated with each measurement. To handle this situation components with low weights are pruned and similar components are merged. The procedure is described in Vo and Ma (2006) and in Algorithm 8. A model based approach to merge components is presented in Paper D.

The extraction of multiple-target state estimates is straightforward from the posterior PHD. It can be assumed that closely spaced components are reasonable well separated after the prune and merge step of the algorithm. The means of the remaining Gaussian components are the local maxima of the PHD. The states are extracted by selecting the means of the Gaussians that have weights greater

Algorithm 8 Merging and Pruning for GM-PHD filter

```
Require: the Gaussian components $\{w_k^{(i)}, \mu_k^{(i)}, P_k^{(i)}\}_{i=1}^{J_{k|k}}$, the truncation threshold $\delta_\mathrm{t}$ and the merging threshold $\delta_\mathrm{m}$
initiate: by setting $\ell = 0$, $I = \{i = 1, \dots, J_{k|k} | w_k^{(i)} > \delta_\mathrm{t}\}$ and then
repeat
    $\ell := \ell + 1$
    $j := \arg\max_{i \in I} w_k^{(i)}$
    $L := \left\{ i \in I \middle| \left(\mu_k^{(i)} - \mu_k^{(j)}\right)^\mathrm{T} \left(P_k^{(i)}\right)^{-1} \left(\mu_k^{(i)} - \mu_k^{(j)}\right) \leq \delta_\mathrm{m} \right\}$,
    $\tilde{w}_k^{(\ell)} = \sum_{i \in L} w_k^{(i)}$
    $\tilde{\mu}_k^{(\ell)} = \frac{1}{\tilde{w}_k^{(\ell)}} \sum_{i \in L} w_k^{(i)} \mu_k^{(i)}$
    $\tilde{P}_k^{(\ell)} = \frac{1}{\tilde{w}_k^{(\ell)}} \sum_{i \in L} w_k^{(i)} \left( P_k^{(i)} + \left(\tilde{\mu}_k^{(\ell)} - \mu_k^{(i)}\right)\left(\tilde{\mu}_k^{(\ell)} - \mu_k^{(i)}\right)^\mathrm{T} \right)$
    $I := I \backslash L$
until $I = \emptyset$
if $\ell > J_{\max}$, the then chose the Gaussians $\{\tilde{w}_k^{(i)}, \tilde{\mu}_k^{(i)}, \tilde{P}_k^{(i)}\}_{i=1}^{\ell}$ with the largest weights.
return the Gaussian components $\{\tilde{w}_k^{(i)}, \tilde{\mu}_k^{(i)}, \tilde{P}_k^{(i)}\}_{i=1}^{\ell}$
```

than some threshold, see Algorithm 9. An advantage with the PHD filter is that a hard decision, i.e., extraction of targets, is only performed for obtaining an output from the filter at each time step. The extracted targets are not used within the filter; the information about the intensity of targets remains in the filter without the need to make hard decisions, which would destroy information.

Algorithm 9 Multitarget State Extraction

```
Require: $\{w_k^{(i)}, \mu_k^{(i)}, P_k^{(i)}\}_{i=1}^{J_k}$ and a threshold $\delta_\mathrm{e}$
Set $\widehat{\mathbf{X}}_k = \emptyset$
for $i = 1$ to $J_k$ do
    if $w_k^{(i)} > \delta_\mathrm{e}$ then
        for $j = 1$ to round$(w_k^{(i)})$ do
            update $\widehat{\mathbf{X}}_k = \begin{bmatrix}\widehat{\mathbf{X}}_k & \mu_k^{(i)}\end{bmatrix}$
        end for
    end if
end for
return the multitarget state estimate $\widehat{\mathbf{X}}_k$
```

6 Concluding Remarks

In the first part an overview of the basics behind the research reported in this thesis has been presented. This part also aims at explaining how the papers in Part II relate to each other and to the existing theory. A conclusion of the results presented in this thesis is given in Section 6.1 and ideas for future work are discussed in Section 6.2.

6.1 Conclusion

The work presented in this thesis has dealt with the problem of estimating the motion and parameters of a vehicle, as well as representing and estimating its surroundings. In this context the surroundings consist of other vehicles, stationary objects, and the road. Here, a major part of the work is not only the estimation problem itself, but also the way in which to represent the environment, i.e., building maps of stationary objects and describing the size of other vehicles.

The second part of the thesis begins in Paper A with an overview of mapping techniques used to describe a road and its closer environment. Four different types of mapping philosophies are described; beginning with so called feature based maps, where each element of the map both described the properties and location of the specific map-element. In location based maps each map-element already corresponds to a certain location and the value of the element describes only the properties in that position. Hence, feature based maps are usually rather sparse. Furthermore, road maps and intensity based maps are also summarized. The intensity based map is the main topic of paper D.

Paper B is concerned with estimating the lane geometry. The lane markings are described by a polynomial and the coefficients are the states to estimate. This problem can be solved with a camera and computer vision, but by fusing the data obtained from the image processing with information about the ego vehicle's motion and the other vehicles' movement on the road, the road geometry estimate can be improved. The other vehicles are tracked primarily by using measurements from a radar. The motion of the ego vehicle is estimated by combining measurements from the vehicle's IMU, steering wheel angle sensor and wheel velocity sensors in a model based filter. The model is in this case the so called single track model or bicycle model.

Paper C and D both deal with the problems of estimating and representing stationary objects along the edges of a road. Paper C, associates the radar measurements to extended stationary objects in the form of curved lines and tracks these lines as extended targets. The lines are modeled as polynomials and one novelty in this paper is that the approach leads to a so called errors in variables problem when noisy point measurements are associated to polynomials. The standard approach is to assume that the measurements are only noisy in one direction when associated to a line, here however noise is assumed to be present in all directions. This assumption is of course more realistic and the results are more accurate.

Paper D describes a new type of map, the intensity based map. In this paper the environment, i.e., the stationary objects, is represented by an intensity function or probability hypothesis density (PHD) as it is also called. The intensity function is described by a Gaussian mixture. This makes the representation very sparse, since only a number of sufficient statistics must be tracked, but also very rich since the extension of the Gaussian components can cover and model larger areas. It is shown how prior knowledge of the road construction can be used to both improve the update of the filter and to simplify the map representation.

Paper E and F deal with the tracking of extended moving targets. In Paper E the target is still modeled as a point target, however, it allows for obtaining more than one measurement per target. A modification of the pseudo-likelihood function used in the update step of the so called PHD filter is presented. In the standard filter one measurement can only be obtained from each target per time step. The novelty of the paper is the practical implementation of the pseudo-likelihood function and partitioning of the measurement data. In a number of simulations and experiments the advantages of the modifications are shown. A drawback is that the wrong number of targets is estimated in some rare cases, this is further discussed in the next section outlining some possible directions of future work.

In Paper F, the size and the shape of an extended target are also estimated. It is still assumed that the sensor obtains a number of point measurements. In order to associate the point measurements to the shaped target it is assumed that the measurements have been produced by a number of measurement generating points on the surface of the target. The measurement generating points are represented by a random finite set and these are only used as a means to estimate the size and the shape of the target. Only one simple simulation example is shown and also here future work will cover more situations.

The last paper, Paper G, aims at estimating the wheel radii of a vehicle. This is made under the assumption that a change in the wheel radii is related to a change in tire pressure, and that tire pressure losses can be detected using this approach. The proposed method is only based on measurements from the wheel speed sensors and the GPS position information. The wheel radii are represented with noise parameters in the state space model. The novelty lies in a Bayesian approach to estimate these time-varying noise parameters on-line using a marginalized particle filter. Experimental results of the proposed approach show many advantages, both regarding the accuracy and the reduced computational complexity, when it is compared with other methods.

The approaches in Papers A-D, and G have been evaluated on real data from both freeways and rural roads in Sweden. In Paper E data of moving pedestrians has been used. Of the seven publications in Part II six are journal papers and one is a peer-reviewed conference paper. Five of the papers are published and two are still under review.

6.2 Future Research

The radar and camera data used in this thesis is generally preprocessed. However, the preprocessing is not covered in this thesis. Specifically, more effort can be spent on the image processing to increase the information content. For example within the area of odometry the estimate could be more accurate if the camera information is used in addition to the IMU measurements. This is called visual odometry and it would probably improve the estimate of the body side slip angles, especially during extreme maneuvers where the tire road interaction is strongly nonlinear. Since only one camera is used, the inverse depth parametrization introduced by Civera et al. (2008) is an interesting approach, see e.g., Schön and Roll (2009); Nilsson et al. (2011) for automotive examples on visual odometry. To verify the state estimates, more accurate reference values are needed as well. Furthermore, the methods presented in Paper G could possibly be used as a basis to more accurately describe the road-tire interaction and thereby improve the estimate of the slip of the vehicle.

Different aspects to estimate extended objects have been discussed in this thesis, however an efficient solution on how to estimate the shape of any possible objects has not been found. Paper F, is closest to this goal, but more experimental studies must be performed and the computational time must be reduced by finding more efficient forms of representation. Also, as concluded in Paper E, the number of targets is falsely estimated in some rare cases, e.g., in the case of occlusion or not well separated targets. In that specific case, the so called cardinalized PHD filter (CPHD) is assumed to improve the results drastically, see Orguner et al. (2011). However, more experimental results must be obtained before conclusions can be drawn.

Currently there is a lot of activity within the computer vision community to enable non-planar road models, making use of parametric road models similar to the ones used in this thesis. A very interesting avenue for future work is to combine the ideas presented in this thesis with information from a camera about the height differences on the road side within a sensor fusion framework. This would probably improve the estimates, especially in situations when there are too few radar measurements available.

Bibliography

A. Abdulle and G. Wanner. 200 years of least squares method. *Elemente der Mathematik*, 57:45–60, May 2002.

M. Adams, W. S. Wijesoma, and A. Shacklock. Autonomous navigation: Achievements in complex environments. *IEEE Instrumentation & Measurement Magazine*, 10(3):15–21, June 2007.

M. Ahrholdt, F. Bengtsson, L. Danielsson, and C. Lundquist. SEFS – results on sensor data fusion system development. In *Proceedings of the World Congress on Intelligent Transportation Systems and Services*, Stockholm, Sweden, September 2009.

B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Information and system science series. Prentice Hall, Englewood Cliffs, NJ, USA, 1979.

D. Angelova and L. Mihaylova. Extended object tracking using Monte Carlo methods. *IEEE Transactions on Signal Processing*, 56(2):825–832, February 2008.

T. Ardeshiri, F. Larsson, F. Gustafsson, T. B. Schön, and M. Felsberg. Bicycle tracking using ellipse extraction. In *Proceedings of the International Conference on Information Fusion*, pages 1–8, Chicago, IL, USA, July 2011.

M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, February 2002.

T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, September 2006.

Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Mathematics in science and engineering. Academic Press, Orlando, FL, USA, 1988.

Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11(5):451–460, September 1975.

Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, New York, NY, USA, 2001.

M. Baum and U. D. Hanebeck. Random hypersurface models for extended object tracking. In *IEEE International Symposium on Signal Processing and Information Technology*, pages 178–183, Ajman, United Arab Emirates, December 2009.

M. Baum and U. D. Hanebeck. Shape tracking of extended objects and group targets with star-convex RHMs. In *Proceedings of the International Conference on Information Fusion*, pages 338–345, Chicago, IL, USA, July 2011.

M. Baum, B. Noack, and U. D. Hanebeck. Extended object and group tracking with elliptic random hypersurface models. In *Proceedings of the International Conference on Information Fusion*, pages 1–8, Edinburgh, UK, July 2010.

T. Bayes. An essay towards solving a problem in the doctrine of chances. *The Philosophical Transactions*, 53:370–418, 1763.

R. Behringer. *Visuelle Erkennung und Interpretation des Fahrspurverlaufes durch Rechnersehen für ein autonomes Straßenfahrzeug*, volume 310 of *Fortschrittsberichte VDI, Reihe 12*. VDI Verlag, Düsseldorf, Germany, 1997. Also as: PhD Thesis, Universität der Bundeswehr, 1996.

F. Bengtsson. *Models for tracking in automotive safety systems*. Licentiate Thesis No R012/2008, Department of Signals and Systems, Chalmers University of Technology, 2008.

F. Bengtsson and L. Danielsson. Designing a real time sensor data fusion system with application to automotive safety. In *Proceedings of the World Congress on Intelligent Transportation Systems and Services*, New York, USA, November 2008.

D. P. Bertsekas. The auction algorithm for assignment and other network flow problem: a tutorial. *Interfaces*, 20(4):133–149, July 1990.

S. S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, Norwood, MA, USA, 1999.

Y. Boers, H. Driessen, J. Torstensson, M. Trieb, R. Karlsson, and F. Gustafsson. Track-before-detect algorithm for tracking extended targets. *IEE Proceedings of Radar, Sonar and Navigation*, 153(4):345–351, August 2006.

M. Bühren and B. Yang. Simulation of automotive radar target lists using a novel approach of object representation. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 314–319, Tokyo, Japan, 2006.

O. Cappe, S. J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5): 899–924, May 2007.

S. Challa, M. R. Morelande, D. Musicki, and R. J. Evans. *Fundamentals of Object Tracking*. Cambridge University Press, Cambridge, UK, 2011.

J. Civera, A. J. Davison, and J. Montiel. Inverse depth parametrization for monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932–945, October 2008.

D. E. Clark. First-moment multi-object forward-backward smoothing. In *Proceedings of the International Conference on Information Fusion*, pages 1–6, Edinburgh, UK, July 2010.

D. J. Daley and D. Vere-Jones. *An introduction to the theory of point processes. Vol. 1 , Elementary theory and method*. Springer, New York, NY, USA, 2 edition, 2003.

L. Danielsson. *Tracking Theory for Preventive Safety Systems*. Licentiate Thesis No R004/2008, Department of Signals and Systems, Chalmers University of Technology, 2008.

L. Danielsson. *Tracking and radar sensor modelling for automotive safety systems*. PhD thesis No 3064, Department of Signals and Systems, Chalmers University of Technology, 2010.

J. Degerman, J. Wintenby, and D. Svensson. Extended target tracking using principal components. In *Proceedings of the International Conference on Information Fusion*, pages 1–8, Chicago, IL, USA, July 2011.

J. C. Dezert. Tracking maneuvering and bending extended target in cluttered environment. In *Proceedings of Signal and Data Processing of Small Targets*, volume 3373, pages 283–294, Orlando, FL, USA, April 1998. SPIE.

E. D. Dickmanns. *Dynamic Vision for Perception and Control of Motion*. Springer, London, UK, 2007.

P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez. Particle filtering. *Signal Processing Magazine, IEEE*, 20(5):19–38, September 2003.

A. Doucet, S. J. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, New York, USA, 2001.

O. E. Drummond, S. S. Blackman, and G. C. Pretrisor. Tracking clusters and extended objects with multiple sensors. In *Proceedings of Signal and Data Processing of Small Targets*, volume 1305, pages 362–375, Orlando, FL, USA, January 1990. SPIE.

H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping (SLAM): Part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, June 2006.

A. Eidehall. *Tracking and threat assessment for automotive collision avoidance*.

PhD thesis No 1066, Linköping Studies in Science and Technology, Linköping, Sweden, January 2007.

A. Eidehall and F. Gustafsson. Obtaining reference road geometry parameters from recorded sensor data. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 256–260, Tokyo, Japan, June 2006.

A. Eidehall, J. Pohl, and F. Gustafsson. Joint road geometry estimation and vehicle tracking. *Control Engineering Practice*, 15(12):1484–1494, December 2007.

O. Erdinc, P. Willett, and Y. Bar-Shalom. The bin-occupancy filter and its connection to the PHD filters. *IEEE Transactions on Signal Processing*, 57(11): 4232–4246, November 2009.

M. Feldmann, D. Fränken, and W. Koch. Tracking of extended objects and group targets using random matrices. *IEEE Transactions on Signal Processing*, 59(4): 1409–1420, April 2011.

R. A. Fisher. On an absolute criterion for fitting frequency curves. *Messenger of Mathematics*, 41:155–160, 1912.

R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society Series A*, 222:309–368, 1922.

T. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Ocean Engineering*, 8(3):173–184, July 1983.

A. Gern, U. Franke, and P. Levi. Advanced lane recognition - fusing vision and radar. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 45–51, Dearborn, MI, USA, October 2000.

A. Gern, U. Franke, and P. Levi. Robust vehicle tracking fusing radar and vision. In *Proceedings of the international conference of multisensor fusion and integration for intelligent systems*, pages 323–328, Baden-Baden, Germany, August 2001.

K. Gilholm and D. Salmond. Spatial distribution model for tracking extended objects. *IEE Proceedings of Radar, Sonar and Navigation*, 152(5):364–371, October 2005.

K. Gilholm, S. Godsill, S. Maskell, and D. Salmond. Poisson models for extended target and group tracking. In *Proceedings of Signal and Data Processing of Small Targets*, volume 5913, pages 230–241, San Diego, CA, USA, August 2005. SPIE.

T. D. Gillespie. *Fundamentals of Vehicle Dynamics*. SAE Society of Automotive Engineers, Warrendale, PA, USA, 1992.

I. R. Goodman, R. P. S. Mahler, and H. T. Nguyen. *Mathematics of data fusion*. Kluwer Academic, Dordrecht, 1997.

N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings on Radar and Signal Processing*, 140(5):107–113, April 1993.

K. Granström, C. Lundquist, and U Orguner. A Gaussian mixture PHD filter for extended target tracking. In *Proceedings of the International Conference on Information Fusion*, Edinburgh, UK, July 2010.

K. Granström, C. Lundquist, and U. Orguner. Extended target tracking using a Gaussian-mixture PHD filter. *IEEE Transactions on Aerospace and Electronic Systems*, 2011a. Under review.

K. Granström, C. Lundquist, and U Orguner. Tracking rectangular and elliptical extended targets using laser measurements. In *Proceedings of the International Conference on Information Fusion*, Chicago, IL, USA, July 2011b.

J. Gunnarsson. *Models and Algorithms - with applications to vehicle tracking and frequency estimation*. PhD thesis No 2628, Department of Signals and Systems, Chalmers University of Technology, June 2007.

J. Gunnarsson, L. Svensson, E Bengtsson, and L. Danielsson. Joint driver intention classification and tracking of vehicles. In *IEEE Nonlinear Statistical Signal Processing Workshop*, pages 95–98, September 2006.

J. Gunnarsson, L. Svensson, L. Danielsson, and F. Bengtsson. Tracking vehicles using radar detections. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 296–302, Istanbul, Turkey, June 2007.

F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons, New York, USA, 2000.

F. Gustafsson. Automotive safety systems. *IEEE Signal Processing Magazine*, 26 (4):32–47, July 2009.

F. Gustafsson. Particle filter theory and practice with positioning applications. *IEEE Transactions on Aerospace and Electronic Systems*, 25(7):53–82, July 2010.

G. Hendeby. *Performance and Implementation Aspects of Nonlinear Filtering*. PhD thesis No 1161, Linköping Studies in Science and Technology, Linköping, Sweden, February 2008.

J. Jansson. *Collision Avoidance Theory with Applications to Automotive Collision Mitigation*. PhD thesis No 950, Linköping Studies in Science and Technology, Linköping, Sweden, June 2005.

A. H. Jazwinski. *Stochastic processes and filtering theory*. Mathematics in science and engineering. Academic Press, New York, USA, 1970.

K. H. Johansson, M. Törngren, and L. Nielsen. Vehicle applications of controller area network. In D. Hristu-Varsakelis and W. S. Levine, editors, *Handbook of Networked and Embedded Control Systems*, pages 741–765. Birkhäuser, 2005.

S. J. Julier. The scaled unscented transformation. In *Proceedings of the American Control Conference*, volume 6, pages 4555–4559, 2002.

S. J. Julier and J. K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *Signal Processing, Sensor Fusion, and Target Recognition VI*, volume 3068, pages 182–193, Orlando, FL, USA, 1997. SPIE.

S. J. Julier and J. K. Uhlmann. Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations. In *Proceedings of the American Control Conference*, volume 2, pages 887–892, 2002.

S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, March 2004.

S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *American Control Conference, 1995. Proceedings of the*, volume 3, pages 1628–1632, Jun 1995.

T. Kailath. *Linear systems*. Prentice Hall, Englewood Cliffs, NJ, USA, 1980.

T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Information and System Sciences Series. Prentice Hall, Upper Saddle River, NJ, USA, 2000.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82:35–45, 1960.

R. Karlsson. *Particle Filtering for Positioning and Tracking Applications*. PhD thesis No 924, Linköping Studies in Science and Technology, Linköping, Sweden, March 2005.

S. M. Kay. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Prentice Hall Signal Processing. Prentice Hall, Upper Saddle River, NJ, USA, 1993.

U. Kiencke, S. Dais, and M. Litschel. Automotive serial controller area network. Technical Report 860391, SAE International Congress, 1986.

J. W. Koch. Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44 (3):1042–1059, July 2008.

M. E. Liggins, D. L. Hall, and J. Llinas. *Handbook of multisensor data fusion : theory and practice*. CRC Press, Boca Raton, FL, USA, 2 edition, 2009.

L. Lin, Y. Bar-Shalom, and T. Kirubarajan. Data association combined with the probability hypothesis density filter for multitarget tracking. In *Proceedings of Signal and Data Processing of Small Targets*, volume 5428, pages 464–475, Orlando, FL, USA, 2004. SPIE.

L. Ljung. *System identification, Theory for the user*. System sciences series. Prentice Hall, Upper Saddle River, NJ, USA, 2 edition, 1999.

C. Lundquist. *Automotive Sensor Fusion for Situation Awareness*. Licentiate Thesis No 1422, Department of Electrical Engineering, Linköping University, Sweden, 2009.

C. Lundquist. Method for stabilizing a vehicle combination. U.S. Patent US 8010253 August 30, 2011 and German Patent Application DE 102007008342 August 21, 2008, Priority date February 20, 2007.

C. Lundquist and R. Großheim. Method and device for determining steering angle information. German Patent Application DE 10 2007 000 958 Mai 14, 2009, International Patent Application WO 2009 047 020 April 16, 2009 and European Patent Application EP 2205478 April 16, 2009, Priority date October 2, 2007.

C. Lundquist and W. Reinelt. Back driving assistant for passenger cars with trailer. In *Proceedings of the SAE World Congress*, SAE paper 2006-01-0940, Detroit, MI, USA, April 2006a.

C. Lundquist and W. Reinelt. Rückwärtsfahrassistent für PKW mit Aktive Front Steering. In *Proceedings of the AUTOREG (Steuerung und Regelung von Fahrzeugen und Motoren*, VDI Bericht 1931, pages 45–54, Wiesloch, Germany, March 2006b.

C. Lundquist and W. Reinelt. Electric motor rotor position monitoring method for electrically aided steering system e.g. steer by wire, for motor vehicle, involves outputting alarm when difference between measurement value and estimated value of motor exceeds threshold. German Patent Application DE 102005016514 October 12, 2006, Priority date April 8, 2006.

C. Lundquist and T. B. Schön. Road geometry estimation and vehicle tracking using a single track model. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 144–149, Eindhoven, The Netherlands, June 2008.

C. Lundquist and T. B. Schön. Estimation of the free space in front of a moving vehicle. In *Proceedings of the SAE World Congress*, SAE paper 2009-01-1288, Detroit, MI, USA, April 2009a.

C. Lundquist and T. B. Schön. Recursive identification of cornering stiffness parameters for an enhanced single track model. In *Proceedings of the IFAC Symposium on System Identification*, pages 1726–1731, Saint-Malo, France, July 2009b.

C. Lundquist and T. B. Schön. Joint ego-motion and road geometry estimation. *Information Fusion*, 12:253–263, October 2011.

C. Lundquist, U. Orguner, and T. B. Schön. Tracking stationary extended objects for road mapping using radar measurements. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 405–410, Xi'an, China, June 2009.

C. Lundquist, L. Danielsson, and F. Gustafsson. Random set based road mapping using radar measurements. In *Proceedings of the European Signal Processing Conference*, pages 219–223, Aalborg, Denmark, August 2010a.

C. Lundquist, U. Orguner, and F Gustafsson. Estimating polynomial structures from radar data. In *Proceedings of the International Conference on Information Fusion*, Edinburgh, UK, July 2010b.

C. Lundquist, K. Granström, and U. Orguner. Estimating the shape of targets with a PHD filter. In *Proceedings of the International Conference on Information Fusion*, Chicago, IL, USA, July 2011a.

C. Lundquist, L. Hammarstrand, and F. Gustafsson. Road intensity based mapping using radar measurements with a probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 59(4):1397–1408, April 2011b.

C. Lundquist, U. Orguner, and F. Gustafsson. Extended target tracking using polynomials with applications to road-map estimation. *IEEE Transactions on Signal Processing*, 59(1):15–26, January 2011c.

C. Lundquist, E. Özkan, and F. Gustafsson. Tire radii estimation using a marginalized particle filter. *IEEE Transactions on Intelligent Transportation Systems*, 2011d. Submitted.

C. Lundquist, T. B. Schön, and F. Gustafsson. Situational awareness and road prediction for trajectory control applications. In A. Eskandarian, editor, *Handbook of Intelligent Vehicles*, chapter 24. Springer, November 2011e.

C. Lundquist, M. Skoglund, K. Granström, and T. Glad. How peer-review affects student learning. In *Utvecklingskonferens för Sveriges ingenjörsutbildingar*, Norrköping, Sweden, November 2011f.

R. P. S. Mahler. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, October 2003.

R. P. S. Mahler. Statistics 101 for multisensor, multitarget data fusion. *IEEE Transactions on Aerospace and Electronic Systems*, 19(1):53–64, January 2004.

R. P. S. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, Boston, MA, USA, 2007a.

R. P. S. Mahler. PHD filters of higher order in target number. *IEEE Transactions on Aerospace and Electronic Systems*, 43(4):1523–1543, October 2007b.

R. P. S. Mahler. PHD filters for nonstandard targets, I: Extended targets. In *Proceedings of the International Conference on Information Fusion*, pages 915–921, Seattle, WA, USA, July 2009a.

R. P. S. Mahler. Random set theory for multisource-multitarget information fusion. In M. E. Liggins, D. L. Hall, and J. Llinas, editors, *Handbook of multisensor data fusion: theory and practice*, chapter 16. CRC Press, Boca Raton, FL, 2 edition, November 2009b.

R. P. S. Mahler, B.-N. Vo, and B.-T. Vo. The forward-backward probability hypothesis density smoother. In *Proceedings of the International Conference on Information Fusion*, pages 1–8, Edinburgh, UK, July 2010.

S. Malinen, C. Lundquist, and W. Reinelt. Fault detection of a steering wheel sensor signal in an active front steering system. In *Preprints of the IFAC Symposium on SAFEPROCESS*, pages 547–552, Beijing, China, August 2006.

M. Mitschke and H. Wallentowitz. *Dynamik der Kraftfahrzeuge*. Springer, Berlin, Heidelberg, 4 edition, 2004.

E. Nilsson, C. Lundquist, T. B. Schön, D. Forslund, and J. Roll. Vehicle motion estimation using an infrared camera. In *Proceedings of the World Congress of the International Federation of Automatic Control*, Milan, Italy, August 2011.

U. Orguner, C. Lundquist, and K. Granström. Extended target tracking with a cardinalized probability hypothesis density filter. In *Proceedings of the International Conference on Information Fusion*, pages 65–72, Chicago, IL, USA, July 2011.

E. Özkan, C. Lundquist, and F. Gustafsson. A Bayesian approach to jointly estimate tire radii and vehicle trajectory. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, Washington DC, USA, October 2011.

K. Panta, B.-N. Vo, S. Singh, and A. Doucet. Probability hypothesis density filter versus multiple hypothesis tracking. In *Signal Processing, Sensor Fusion, and Target Recognition XIII*, volume 5429, pages 284–295, Orlando, FL, USA, 2004. SPIE.

R. Rajamani. *Vehicle Dynamics and Control*. Springer, Boston, MA, USA, 2006. ISBN 978-0-387-28823-9.

G. Reimann and C. Lundquist. Method for operating electronically controlled servo steering system of motor vehicle, involves determining steering wheel angle as measure for desired steering handle angle by steering handle for steering wheels of motor vehicle. German Patent Application DE 102006053029 May 15, 2008, Priority date November 10, 2006.

W. Reinelt and C. Lundquist. Observer based sensor monitoring in an active front steering system using explicit sensor failure modeling. In *Proceedings of the IFAC World Congress*, Prague, Czech Republic, July 2005.

W. Reinelt and C. Lundquist. Mechatronische Lenksysteme: Modellbildung und Funktionalität des Active Front Steering. In R. Isermann, editor, *Fahrdynamik Regelung - Modellbildung, Fahrassistenzsysteme, Mechatronik*, pages 213–236. Vieweg Verlag, September 2006a.

W. Reinelt and C. Lundquist. Controllability of active steering system hazards: From standards to driving tests. In Juan R. Pimintel, editor, *Safety Critical Automotive Systems*, pages 173–178. SAE International, 400 Commonwealth Drive, Warrendale, PA, USA, August 2006b.

W. Reinelt and C. Lundquist. Method for assisting the driver of a motor vehicle with a trailer when reversing. German Patent DE 10 2006 002 294 February 24, 2011, European Patent Application EP 1810913 July 25, 2007 and Japanese Patent Application JP 2007191143 August 2, 2007, Priority date January 18, 2006.

W. Reinelt, W. Klier, G. Reimann, C. Lundquist, W. Schuster, and R. Großheim. Active front steering for passenger cars: System modelling and functions. In *Proceedings of the IFAC Symposium on Advances in Automotive Control*, Salerno, Italy, April 2004.

W. Reinelt, C. Lundquist, and H. Johansson. On-line sensor monitoring in an active front steering system using extended Kalman filtering. In *Proceedings of the SAE World Congress*, SAE paper 2005-01-1271, Detroit, MI, USA, April 2005.

W. Reinelt, W. Schuster, R. Großheim, and C. Lundquist. Motor vehicle's electronically regulated servo steering system operating method, involves comparing actual value of measured value with stored practical value of corresponding measured value. German Patent DE 10 2006 040 443 January 27, 2011, Priority date August 29, 2006.

W. Reinelt, W. Schuster, R. Großheim, and C. Lundquist. Operating method for electronic servo steering system of vehicle, involves presetting steering wheel angle by steering mechanism as measure for desired wheel turning angle for steering wheel of vehicle. German Patent Application DE 102006052092 May 8, 2008, Priority date November 4, 2006.

W. Reinelt, C. Lundquist, and S. Malinen. Automatic generation of a computer program for monitoring a main program to provide operational safety. German Patent Application DE 102005049657 April 19, 2007, Priority date October 18, 2005.

W. Reinelt, W. Schuster, R. Großheim, and C. Lundquist. Electronic servo steering system operating method for motor vehicle, involves recognizing track misalignment of vehicle when forces differentiate around preset value from each other at preset period of time in magnitude and/or direction. German Patent DE 102006043069 December 3, 2009, Priority date September 14, 2006.

W. Reinelt, W. Schuster, R. Großheim, and C. Lundquist. Operating method for electronic power-assisted steering system of vehicle, involves overlapping additional angle, which is disabled after re-start of utility function. German Patent Application DE 102006041236 Mars 6, 2008, Priority date September 2, 2006a.

W. Reinelt, W. Schuster, R. Großheim, and C. Lundquist. Operating method for electronic power-assisted steering system of vehicle, involves re-starting utility function, and after re-start of utility function superimposition of additional angle is unlatched. German Patent DE 102006041237 December 3, 2009, Priority date September 2, 2006b.

B. Ristic and D. J. Salmond. A study of a nonlinear filtering problem for tracking an extended target. In *Proceedings of the International Conference on Information Fusion*, pages 503–509, Stockholm, Sweden, June 2004.

B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle filters for tracking applications*. Artech House, London, UK, 2004.

X. Rong Li and V. P. Jilkov. Survey of maneuvering target tracking: Part III. Measurement models. In *Proceedings of Signal and Data Processing of Small Targets*, volume 4473, pages 423–446, San Diego, CA, USA, 2001. SPIE.

X. Rong Li and V. P. Jilkov. Survey of maneuvering target tracking: Part I. Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4): 1333–1364, October 2003.

M. Roth and F. Gustafsson. An efficient implementation of the second order extended Kalman filter. In *Proceedings of the International Conference on Information Fusion*, pages 1–6, July 2011.

W. J. Rugh. *Linear System Theory*. Information and system sciences series. Prentice Hall, Upper Saddle River, NJ, USA, 2 edition, 1996.

D. J. Salmond and M. C. Parr. Track maintenance using measurements of target extent. *IEE Proceedings of Radar, Sonar and Navigation*, 150(6):389–395, December 2003.

S. F. Schmidt. Application of state-space methods to navigation problems. *Advances in Control Systems*, 3:293–340, 1966.

B. Schofield. *Model-Based Vehicle Dynamics Control for Active Safety*. PhD thesis, Department of Automatic Control, Lund University, Sweden, September 2008.

T. B. Schön. *Estimation of Nonlinear Dynamic Systems – Theory and Applications*. PhD thesis No 998, Linköping Studies in Science and Technology, Department of Electrical Engineering, Linköping University, Sweden, February 2006.

T. B. Schön and J. Roll. Ego-motion and indirect road geometry estimation using night vision. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 30–35, Xi'an, China, June 2009.

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

T. B. Schön, A. Eidehall, and F. Gustafsson. Lane departure detection for improved road geometry estimation. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 546–551, Tokyo, Japan, June 2006.

T. B. Schön, D. Törnqvist, and F. Gustafsson. Fast particle filters for multi-rate sensors. In *Proceedings of the 15th European Signal Processing Conference*, Poznań, Poland, September 2007.

H. Sidenbladh. Multi-target particle filtering for the probability hypothesis density. In *Proceedings of the International Conference on Information Fusion*, volume 2, pages 800–806, Cairns, Australia, March 2003.

G. L. Smith, S. F. Schmidt, and L. A. McGee. Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle. Technical Report TR R-135, NASA, 1962.

J. Sörstedt, L. Svensson, F. Sandblom, and L. Hammarstrand. A new vehicle motion model for improved predictions and situation assessment. *IEEE Transactions on Intelligent Transportation Systems*, 2011. ISSN 1524-9050. doi: 10.1109/TITS.2011.2160342.

R. L. Streit. Multisensor multitarget intensity filter. In *Proceedings of the International Conference on Information Fusion*, pages 1–8, Cologne, Germany, July 2008.

R. L. Streit. *Poisson Point Processes*. Springer, 1 edition, 2010.

R. L. Streit and L. D. Stone. Bayes derivation of multitarget intensity filters. In *Proceedings of the International Conference on Information Fusion*, pages 1–8, Cologne, Germany, July 2008.

L. Svensson and J. Gunnarsson. A new motion model for tracking of vehicles. In *Proceedings of the 14th IFAC Symposium on System Identification*, Newcastle, Australia, 2006.

A. Swain and D. Clark. Extended object filtering using spatial independent cluster processes. In *Proceedings of the International Conference on Information Fusion*, pages 1–8, Edinburgh, UK, July 2010.

S. Thrun. Robotic mapping: A survey. In *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.

H. L. van Trees. *Detection, Estimation, and Modulation Theory*. John Wiley & Sons, New York, NY, USA, 1968.

J. Vermaak, N. Ikoma, and S. J. Godsill. Sequential Monte Carlo framework for extended object tracking. *IEE Proceedings of Radar, Sonar and Navigation*, 152 (5):353–363, October 2005.

B.-N. Vo and W.-K. Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, November 2006.

B.-N. Vo, S. Singh, and A. Doucet. Random finite sets and sequential Monte Carlo methods in multi-target tracking. In *Proceedings of the International Radar Conference*, pages 486–491, Adelaide, Australia, September 2003.

B.-N. Vo, B.-T. Vo, and R. P. S. Mahler. A closed form solution to the probability hypothesis density smoother. In *Proceedings of the International Conference on Information Fusion*, pages 1–8, Edinburgh, UK, July 2010.

B.-N. Vo, B.-T. Vo, and R. P. S. Mahler. Closed form solutions to forward-backward smoothing. *IEEE Transactions on Signal Processing*, 2011. doi: 10.1109/TSP.2011.2168519.

B.-T. Vo, B.-N. Vo, and A. Cantoni. Analytic implementations of the cardinalized probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 55(7):3553–3567, July 2007.

N. Wahlström, J. Callmer, and F. Gustafsson. Single target tracking using vector magnetometers. In *IEEE Conference on Acoustics, Speech and Signal Processing*, pages 4332–4335, Prague, Czech Republic, May 2011.

A. Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, June 1945.

M. J. Waxman and O. E. Drummond. A bibliography of cluster (group) tracking. In *Proceedings of Signal and Data Processing of Small Targets*, volume 5428, pages 551–560, Orlando, FL, USA, April 2004. SPIE.

M. Wieneke and S. J. Davey. Histogram PMHT with target extent estimates based on random matrices. In *Proceedings of the International Conference on Information Fusion*, pages 1–8, Chicago, IL, USA, July 2011.

J. Y. Wong. *Theory Of Ground Vehicles*. John Wiley & Sons, New York, USA, 3 edition, 2001.

T. Zajic and R. P. S. Mahler. Particle-systems implementation of the PHD multitarget-tracking filter. In *Signal Processing, Sensor Fusion, and Target Recognition XII*, volume 5096, pages 291–299, Orlando, FL, USA, April 2003. SPIE.

Z. Zomotor and U. Franke. Sensor fusion for improved vision based lane recognition and object tracking with range-finders. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 595–600, Boston, MA, USA, November 1997.

Part II

Publications

Paper A

Situational Awareness and Road Prediction for Trajectory Control Applications

Authors: Christian Lundquist, Thomas B. Schön and Fredrik Gustafsson

Edited version of the paper:

C. Lundquist, T. B. Schön, and F. Gustafsson. Situational awareness and road prediction for trajectory control applications. In A. Eskandarian, editor, *Handbook of Intelligent Vehicles*, chapter 24. Springer, November 2011e.

Situational Awareness and Road Prediction for Trajectory Control Applications

Christian Lundquist, Thomas B. Schön and Fredrik Gustafsson

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

Abstract

This chapter is concerned with the problem of estimating a map of the immediate surroundings of a vehicle as it moves. In order to compute these map estimates sensor measurements from radars, lasers and/or cameras are used together with the standard proprioceptive sensors present in a car. Four different types of maps are discussed. Feature based maps are represented by a set of salient features. Road maps make use of the fact that roads are highly structured, since they are built according to clearly specified road construction standards, which allows relatively simple and powerful models of the road to be employed. Location based maps, where occupancy grid maps belong and finally intensity based maps, which can be viewed as a continuous version of the location based maps. The aim is to provide a self-contained presentation of how these maps can be built from measurements. Real data from Swedish roads are used throughout the chapter to illustrate the methods introduced.

1 Introduction

Most automotive original equipment manufacturers today offer longitudinal control systems, such as adaptive cruise control (ACC) or collision mitigation systems. Lateral control systems, such as lane keeping assistance (LKA), emergency lane assist (ELA) (Eidehall et al., 2007) and curve speed warning, are currently developed and released. These systems can roughly be split into safety applications, which aim to mitigate vehicular collisions such as rear end or blind spot detection; and comfort applications such as ACC and LKA, which aim at reducing the driver's work load. The overview article by Caveney (2010) describes the current development of trajectory control systems. The requirements on the position accuracy of the ego vehicle in relation to other vehicles, the road and the surrounding environment increases with those control applications that are currently under development and expected to be introduced to the market.

The systems available or in development today are based on two basic tracking and decision principles: longitudinal systems use a radar, possibly supported by a camera, to track leading vehicles and they decide on braking warnings or interventions. On the other hand, lateral systems use a camera to track the lane markers and they decide on steering warnings or interventions. Future safety and comfort functions need more sophisticated situational awareness and decision functionality:

- A combination of lateral and longitudinal awareness will be needed, where all lanes are monitored, all of their vehicles are tracked, and the road-side conditions are modeled to allow for emergency maneuvers. The result is a situational awareness map, which is the topic for this chapter.
- This will allow for more sophisticated decision functionality. First, the possible evasive driver maneuvers are computed, and only if the driver has no or very little time for evasive actions, the system will intervene. Second, more complex automatic evasive maneuvers can be planned using the situational awareness map, including consecutive lateral and braking actions.

It should be remarked that the accuracy of the navigation systems today and in the near future, see Chapters “In-car Navigation Basics" and “The evolution of in-car navigation systems", are not of much assistance for situational awareness. The reason is that satellite based navigation gives an accuracy of 10-20 meters, which is not sufficient for lateral awareness. Even in future systems, including reference base stations, enabling meter accuracy, the standard road maps will limit the performance since they are not of sufficient accuracy. Thus, two leaps in development are needed before positioning information and standard maps can be used to improve situational awareness maps. Another technical enabler is car to car communication (C2C), which may improve tracking of other vehicles and in the end change the transportation system as has already been done with the transponder systems for aircraft and commercial surface ships. Still, there will always be vehicles and obstacles without functioning communication systems. The need for accurate situation awareness and road prediction to be able to automatically position the car in a lane and derive drivable trajectories will evolve and remain important.

The different types of situation awareness maps used to represent the environment are introduced in Section 2. Details of these maps are presented in Sections 3 to 6. The chapter is concluded in Section 7.

2 Modeling the Environment with a Map

The transportation system may be described and represented by a number of variables. These variables include state variables describing the position, orientation, velocity and size of the vehicles. Here one can distinguish between the own vehicle, the so called ego vehicle, and the other vehicles, referred to as the targets.

The state variable of the ego vehicle at time k is denoted $\mathbf{x}_{\mathrm{E},k}$. The trajectory of the ego vehicle is recorded in $\mathbf{x}_{\mathrm{E},1:k} = \{\mathbf{x}_{\mathrm{E},1}, \ldots, \mathbf{x}_{\mathrm{E},k}\}$, and it is assumed to be *a priori* known in this work. This is a feasible assumption, since the absolute trajectory in world coordinates and the relative position in the road network are separable problems.

The state variable of the targets at time k is denoted $\mathbf{x}_{\mathrm{T},k}$. The road and the environment may be modeled by a map, which is represented by a set of variables describing $N_{\mathbf{m}}$ landmarks in the environment according to

$$\mathbf{M}_k = \left\{\mathbf{m}_k^{(1)}, \mathbf{m}_k^{(2)}, \ldots, \mathbf{m}_k^{(N_{\mathbf{m}})}\right\}. \tag{1}$$

According to Thrun et al. (2005), there exists primarily two types of indexing for probabilistic maps. In a *feature based map* each $\mathbf{m}^{(n)}$ specifies the properties and location of one object, whereas in a *location based map* the index n corresponds to a location and $\mathbf{m}^{(n)}$ is the property of that specific coordinate. The *occupancy grid map* is a classical location based representation of a map, where each cell of the grid is assigned a binary occupancy value that specifies if the location n is occupied ($\mathbf{m}^{(n)} = 1$) or not ($\mathbf{m}^{(n)} = 0$), see e.g., Elfes (1987); Moravec (1988).

The ego vehicle perceives information about the other vehicles and the environment trough its sensors. The sensors provide a set of noisy measurements

$$\mathbf{Z}_k = \left\{\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(2)}, \ldots, \mathbf{z}_k^{(N_{\mathbf{z},k})}\right\} \tag{2}$$

at each discrete time instant $k = 1, \ldots, K$. Common sensors used for automotive navigation and mapping measure either range and bearing angle, as for example radar and laser, or bearing and elevation angles, as for the case of a camera. A signal preprocessing is always included in automotive radar sensors and the sensor provides a list of detected features, defined by the range r, range rate $\dot{r}$ and bearing ψ. The preprocessing, the waveform design and the detection algorithms of the radar is well described by e.g., Rohling and Möller (2008); Rohling and Meinecke (2001). Laser sensors typically obtain one range measurement per beam, and there exists both sensors which emit several beams at different angles and those which have a rotating beam deflection system. They all have in common that the angles at which they measure range are quantized, thus providing a list of range and bearings of which only the ones which are less than the maximum range shall be considered. Another commonly used automotive sensor is the camera. The camera measurements are quantized and the data is represented in a pixel matrix as an image.

Note that the indexing of sensor data is analogous to the representation of maps. Each range and bearing measurement $\mathbf{z}^{(n)}$ from a radar or laser specifies the properties and location of one observation, i.e., it is a feature based measurement. However, the indexing of camera measurement is location based, since the index n corresponds to a pixel in the image and $\mathbf{z}^{(n)}$ is the property of that specific coordinate.

The aim of all stochastic mapping algorithms, independent of indexing, is to estimate the posterior density of the map

$$p(\mathbf{M}_k|\mathbf{Z}_{1:k}, \mathbf{x}_{\mathrm{E},1:k}), \tag{3}$$

given all the measurements $\mathbf{Z}_{1:k}$ from time 1 to k and the trajectory of the ego vehicle $\mathbf{x}_{\mathrm{E},1:k}$. The conditioning on $\mathbf{x}_{\mathrm{E},1:k}$ is implicit in this chapter, since it is assumed to be *a priori* known. To be able to estimate the map, a relation between the map and the measurements must first be found and modeled. This model h is referred to as the measurement equation and for one combination of a measurement $\mathbf{z}_k^{(i)}$ and a map variable $\mathbf{m}_k^{(j)}$ it may be written according to

$$\mathbf{z}_k^{(i)} = h(\mathbf{m}_k^{(j)}) + \mathbf{e}_k, \tag{4}$$

where $\mathbf{e}_k$ is the measurement noise. The primary aim in this chapter is to create a momentary map of the environment currently surrounding the ego vehicle. Hence, it is just the present map data that is recorded in the vehicle. As soon as a part of the environment if sufficiently far from the ego vehicle the corresponding map entries are deleted. Environmental models must be compact, so that they can be transmitted to and used efficiently by other automotive systems, such as path planners. The maps must be adapted to the type of environment they aim to model. For this reason four different map representations, which are relevant for modeling the environment surrounding a vehicle, are described in the following sections.

Feature based map The map is represented by a number of salient features in the scene. Feature representation and tracking as part of a map is described in Section 3.

Road map This is a special case of the feature based map, where the map variables model the geometry of the road. Roads are highly structured; they are built according to road construction standards and contain primarily, straight lines, curves and clothoids. Maps of road lanes and edges are described in Section 4.

Location based map One of the most well established location based map is the occupancy grid map, which is described in Section 5. The map is defined over a continuous space, but discretized with a grid approximation.

Intensity based map The intensity density may be interpreted as the probability that one object is located in an infinitesimal region of the state space. The intensity based map is a continuous approximation of a location based map and it is described in Section 6.

The estimated maps can be used to increase the localization accuracy of the ego vehicle with respect to its local environment. Furthermore, the maps may be used to derive a trajectory, which enables a collision free path of the vehicle.

3 Feature Based Map

Features corresponding to distinct objects in the physical world, such as tree trunks, corners of buildings, lampposts and traffic signs are commonly denoted *landmarks*. The procedure of extracting features reduces the computational complexity of the system, as the features are on a more compact format than the original measurement. The form of the measurement equations (4) depends on the type of sensor used, and the signals measured by the sensor. In this section we will briefly describe the use of features and the corresponding measurement equations in both radar and laser sensors (Section 3.1) as well as cameras (Section 3.2).

The feature based approach may together with existing road maps be used to supplement the GPS-based position of the vehicle. This approach is also commonly referred to as visual odometry, see e.g., Nistér et al. (2006).

3.1 Radar and Laser

As mentioned in the introduction, radar and laser sensors measure at least range and bearing of the landmark relative to the vehicles local coordinate, i.e., the measurement vector is composed of

$$\mathbf{z}^{(i)} = \begin{bmatrix} r^{(i)} & \psi^{(i)} \end{bmatrix}^{\mathrm{T}}. \tag{5}$$

Notice that, for the sake of simplicity, the subscripts k specifying the time stamps of the quantities is dropped throughout this chapter. The assumption will be made that the measurements of the features are independent, i.e., the noise in each individual measurement $\mathbf{z}^{(i)}$ is independent of the noise in the other measurements $\mathbf{z}^{(j)}$, for $i \neq j$. This assumption makes it possible to process one feature at a time in the algorithms. Assume that the ego vehicle pose is defined by

$$\mathbf{x}_{\mathrm{E}} = \begin{bmatrix} \mathrm{x}_E & \mathrm{y}_E & \psi_E \end{bmatrix}^{\mathrm{T}}, \tag{6}$$

where x_E, y_E denote the horizontal position of the vehicle and ψ_E denotes the heading angle of the vehicle. Furthermore, let us assume that one feature j in the map is defined by its Cartesian coordinate,

$$\mathbf{m}^{(j)} = \begin{bmatrix} \mathrm{x}_m^{(j)} & \mathrm{y}_m^{(j)} \end{bmatrix}^{\mathrm{T}}. \tag{7}$$

The measurement model (4) is then written as

$$\begin{bmatrix} r^{(i)} \\ \psi^{(i)} \end{bmatrix} = \begin{bmatrix} \sqrt{(\mathrm{x}_m^{(j)} + \mathrm{x}_E)^2 + (\mathrm{y}_m^{(j)} + \mathrm{y}_E)^2} \\ \arctan \frac{\mathrm{y}_m^{(j)} - \mathrm{y}_E}{\mathrm{x}_m^{(j)} + \mathrm{x}_E} - \psi_E \end{bmatrix} + \begin{bmatrix} e_r \\ e_\psi \end{bmatrix}, \tag{8}$$

where e_r and e_ψ are noise terms. The i^{th} measurement feature corresponds to the j^{th} map feature. The data association problem arises when the correspondence between the measurement feature and the map feature cannot be uniquely identified. A correspondence variable, which describes the relation between measurements and map features is introduced. This variable is also estimated at each time step k and there exists a number of different algorithms to do this, see e.g., Blackman and Popoli (1999). There are quite a few different possibilities on how to define features when radars and lasers are used.

3.2 Cameras and Computer Vision

Features form the bases in many computer vision algorithms, especially when it comes to building maps. There exists a myriad of feature detectors, which extract edges, corners or other distinct patterns. Some of the most well know are the Harris corner detector (Harris and Stephens, 1988), SIFT (Lowe, 2004) and MSER (Matas et al., 2004), see Figure 1 for an example, where the Harris corner detector is used. For a more complete account of various features used, see e.g., Szeliski (2010). Using features to build maps from camera images has been studied for a long time and a good account of this is provided by Davison et al. (2007).

A key component in building maps using features is a good mathematical description of how the features detected in the image plane are related to the corresponding positions in world coordinates. The distance (commonly referred to as the depth) to a landmark cannot be determined from a single image and this fact should be encoded by the mathematical parameterization of the landmark. The so called *inverse depth parameterization* by Civera et al. (2008) provides an elegant uncertainty description of the fact that the depth (i.e., distance) to the landmark is unknown. Here, the landmark (lm) state vector is given by

$$\mathbf{m}^{(j)} = \begin{bmatrix} \mathbf{c}^{(j)} \\ \psi^{(j)} \\ \phi^{(j)} \\ \rho^{(j)} \end{bmatrix} = \begin{bmatrix} \text{camera position first time lm was seen} \\ \text{azimuth angle of lm seen from } \mathbf{c}^{(j)} \\ \text{elevation angle of lm seen from } \mathbf{c}^{(j)} \\ \text{inverse distance (depth) from } \mathbf{c}^{(j)} \text{ to lm} \end{bmatrix} \tag{9a}$$

$$\mathbf{c}^{(j)} = \begin{bmatrix} \mathsf{x}^{(j)} & \mathsf{y}^{(j)} & \mathsf{z}^{(j)} \end{bmatrix}^{\mathrm{T}}, \tag{9b}$$

where $\mathbf{c}^{(j)}$ is the position of the camera expressed in world coordinates at the time when landmark j was first seen, $\psi^{(j)}$ is the azimuth angle of the landmark as seen from $\mathbf{c}^{(j)}$, relative to the world coordinate frame. The elevation angle of the landmark as seen from $\mathbf{c}^{(j)}$, relative to world coordinate frame directions is denoted $\phi^{(j)}$, and the inverse depth, which is the inverse of the distance, from $\mathbf{c}^{(j)}$ to the landmark is denoted $\rho^{(j)}$.

The landmark state $\mathbf{m}^{(j)}$ is a parametrization of the Cartesian position $\mathbf{p}^{(j)}$ of landmark j, see Figure 2. The relationship between the position of the landmark and the inverse depth state representation is given by

$$\mathbf{p}^{(j)} = \mathbf{c}_k^{(j)} + \frac{1}{\rho^{(j)}} \underbrace{\begin{bmatrix} \cos\phi^{(j)} \cos\psi^{(j)} \\ \cos\phi^{(j)} \sin\psi^{(j)} \\ \sin\phi^{(j)} \end{bmatrix}}_{\mathbf{q}^{(j)}}. \tag{10}$$

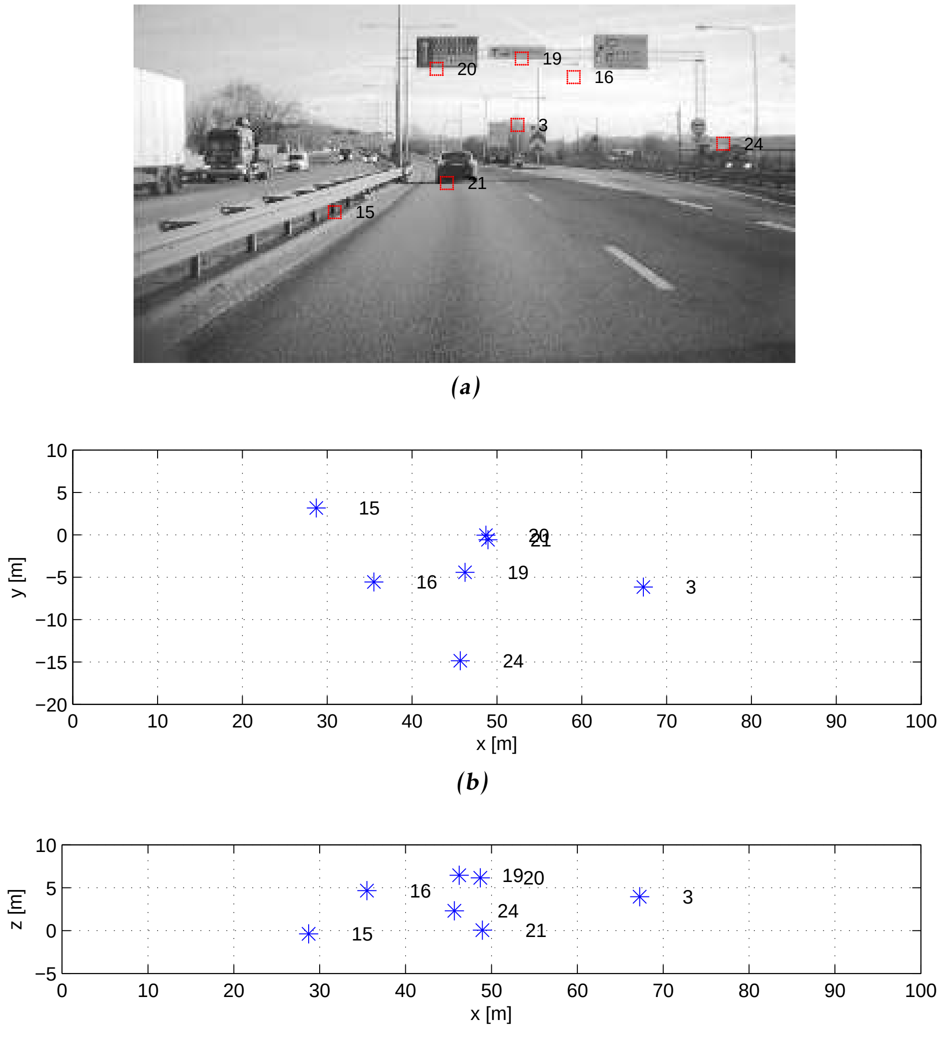


Figure 1: *Features detected using the Harris corner detector are shown in Figure (a). Figure (b) and (c) shows the estimated position of the landmarks in the* $\mathsf{x}-\mathsf{y}$ *and* $\mathsf{x}-\mathsf{z}$ *plane, respectively.*

The measurement model (4) for the landmarks is given by

$$h(\mathbf{m}^{(j)}) = \mathcal{P}_n(\mathbf{p}^{C,(j)}) = \frac{1}{\mathsf{x}_p^{C,(j)}} \begin{bmatrix} \mathsf{y}_p^{C,(j)} \\ \mathsf{z}_p^{C,(j)} \end{bmatrix}, \tag{11}$$

where $\mathcal{P}_n(\mathbf{p}^{C,(j)})$ is used to denote the normalized pinhole projection and $\mathbf{p}^{C,(j)} =$

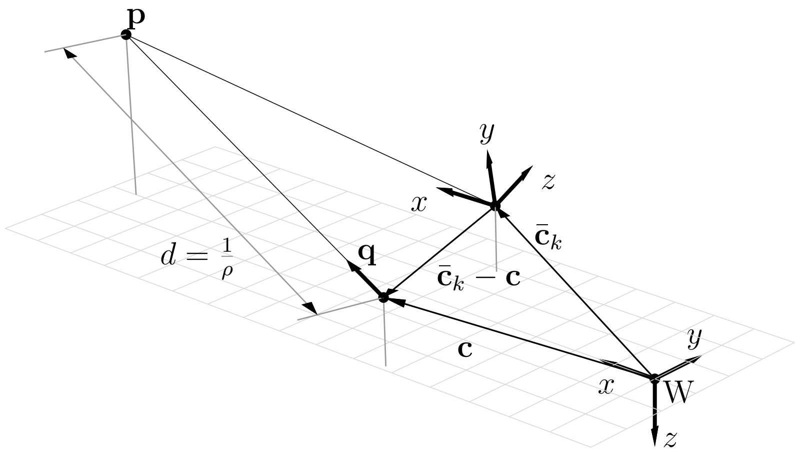


Figure 2: The inverse depth parameterization used for the landmarks. The position of the landmark **p** *is parameterized using the position* **c** *of the camera the first time the feature was seen, the direction* $\mathbf{q}(\phi, \psi)$ *and the inverse depth* ρ*. The position of the camera at time step* k *is denoted* $\bar{\mathbf{c}}_k$*.*

$\begin{bmatrix} \mathsf{x}_p^{C,(j)} & \mathsf{y}_p^{C,(j)} & \mathsf{z}_p^{C,(j)} \end{bmatrix}^{\mathrm{T}}$ denotes the position of feature j at time k in the camera coordinate frame C. Note that before an image position can be used as a measurement together with the measurement equation (11), the image position is adjusted according to the camera specific parameters, such as focal length, pixel sizes, etc. The transformation between pixel coordinates $\begin{bmatrix} \mathsf{u} & \mathsf{v} \end{bmatrix}^{\mathrm{T}}$ and normalized camera coordinates $\begin{bmatrix} \mathsf{y} & \mathsf{z} \end{bmatrix}^{\mathrm{T}}$, which is the kind of coordinates landmark measurements $\mathbf{z}^{(i)}$ (see (11)) are given in, is

$$\begin{bmatrix} \mathsf{y} \\ \mathsf{z} \end{bmatrix} = \begin{bmatrix} \frac{\mathsf{u}-\mathsf{u}^{\mathrm{ic}}}{f_{\mathsf{u}}} \\ \frac{\mathsf{v}-\mathsf{v}^{\mathrm{ic}}}{f_{\mathsf{v}}} \end{bmatrix}, \tag{12}$$

where $\begin{bmatrix} \mathsf{u}^{\mathrm{ic}} & \mathsf{v}^{\mathrm{ic}} \end{bmatrix}^{\mathrm{T}}$ denotes the image center, and f_{u} and f_{v} are the focal lengths (given in pixels) in the lateral u-direction and the vertical v-direction, respectively. The transformation between world and camera coordinates is given by

$$\mathbf{p}^{C,(j)} = R^{CE}\left(R^{EW}\left(\left(\mathbf{c}^{(j)} - \begin{bmatrix} \mathsf{x}_E \\ \mathsf{y}_E \end{bmatrix}\right) + \frac{1}{\rho^{(j)}}\mathbf{q}^{(j)}\right) - \mathbf{c}^E\right), \tag{13a}$$

$$R^{CE} = R(\psi^C, \phi^C, \gamma^C), \tag{13b}$$

$$R^{EW} = R(\psi_E, \phi_E, \gamma_E), \tag{13c}$$

where $\mathbf{c}^E$ denotes the position of the camera expressed in the ego vehicle body coordinate frame. The rotation matrix $R(\alpha, \beta, \gamma)$ transforms coordinates from coordinate frame B to coordinate frame A, where the orientation of B relative to A is ψ (yaw), ϕ (pitch) and γ (roll). Furthermore, ψ^C, ϕ^C and γ^C are the constant yaw, pitch and roll angles of the camera, relative to the vehicle body coordinate frame.

The landmarks are estimated recursively using e.g., a Kalman filter. An example of estimated landmarks is shown in Figure 1. The estimated position $\mathbf{p}^{(j)}$ for seven landmarks is shown in the image plane, as well as in the world $\mathrm{x}-\mathrm{y}$ and $\mathrm{x}-\mathrm{z}$ plane, where the ego vehicle is in origin.

4 Road Map

A road map describe the shape of the road. The roads are mainly modeled using polynomial functions which describe the lane and the road edges. The advantages of road models are that they require sparse memory and are still very accurate, since they do not suffer from discretization problems. General road models are presented in Section 4.1. Lane estimation using camera measurements is described in Section 4.2 and finally road edge estimation based on feature measurements is described in Section 4.3.

4.1 Road Model

The road, as a construction created by humans, possesses no dynamics; it is a static time invariant object in the world coordinate frame. The building of roads is subject to road construction standards, hence, the modeling of roads is geared to these specifications. However, if the road is described in the ego vehicle's coordinate frame and the vehicle is moving along the road it is possible and indeed useful to describe the characteristics of the road using time varying state variables.

A road consists of straight and curved segments with constant radius and of varying length. The sections are connected through transition curves, so that the driver can use smooth and constant steering wheel movements instead of stepwise changes when passing through road segments. More specifically, this means that a transition curve is formed as a clothoid, whose curvature c changes linearly with its curve length x_c according to

$$c(\mathrm{x}_c) = c_0 + c_1 \cdot \mathrm{x}_c. \tag{14}$$

Note that the curvature c is the inverse of the radius. Now, suppose x_c is fixed to the ego vehicle, i.e., $\mathrm{x}_c = 0$ at the position of the ego vehicle. When driving along the road and passing through different road segments c_0 and c_1 will not be constant, but rather time varying state variables

$$\mathbf{m} = \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} \text{curvature at the ego vehicle} \\ \text{curvature derivative} \end{bmatrix}. \tag{15}$$

In section 3 the map features were expressed in a fixed world coordinate frame. However, note that in this section the road map is expressed as seen from the moving ego vehicle. Using (14), a change in curvature at the position of the vehicle is given by

$$\left.\frac{dc(\mathrm{x}_c)}{dt}\right|_{\mathrm{x}_c=0} = \dot{c}_0 = \frac{dc_0}{d\mathrm{x}_c} \cdot \frac{d\mathrm{x}_c}{dt} = c_1 \cdot \mathrm{v}, \tag{16}$$

where v is the ego vehicle's velocity. Furthermore, the process model is given by

$$\begin{bmatrix} \dot{c}_0 \\ \dot{c}_1 \end{bmatrix} = \begin{bmatrix} 0 & \mathrm{v}_\mathrm{x} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} + \begin{bmatrix} 0 \\ w_{c_1} \end{bmatrix}. \tag{17}$$

This model is referred to as the *simple clothoid model* and it is driven by the process noise w_{c_1}. Note that the road is modeled in a road aligned coordinate frame, with the components $(\mathrm{x}_c, \mathrm{y}_c)$, and the origin at the position of the ego vehicle. There are several advantages of using road aligned coordinate frames, especially when it comes to the process models of the other vehicles on the same road, these models are greatly simplified in road aligned coordinates. However, the flexibility of the process model is reduced and basic dynamic relations such as Newton's and Euler's laws cannot be directly applied. The road model (14) is transformed into Cartesian coordinates (x, y) using

$$\mathrm{x}(\mathrm{x}_c) = \int_0^{\mathrm{x}_c} \cos\left(\chi(x)\right) dx \approx \mathrm{x}_c, \tag{18a}$$

$$\mathrm{y}(\mathrm{x}_c) = \int_0^{\mathrm{x}_c} \sin\left(\chi(x)\right) dx \approx \frac{c_0}{2}\mathrm{x}_c^2 + \frac{c_1}{6}\mathrm{x}_c^3, \tag{18b}$$

where the heading angle χ is defined as

$$\chi(x) = \int_0^{x} c(\lambda) d\lambda = c_0 x + \frac{c_1}{2} x^2. \tag{18c}$$

The origin of the two frames is fixed to the ego vehicle, hence, integration constants $(\mathrm{x}_0, \mathrm{y}_0)$ are omitted.

A problem appears when two or more clothoid segments, with different parameters c_0 and c_1, are observed in the same camera view. The parameter c_0 will change continuously during driving, whereas c_1 will be constant in each segment and change stepwise at the segment transition. This leads to a dirac impulse in $\dot{c}_1$ at the transition. The problem can be solved by assuming a high process noise w_{c_1} in (17), but this leads to less precise estimates of the state variables when no segment transitions occur in the camera view. To solve this problem an averaging curvature model was proposed by Dickmanns (1988), which is perhaps best described with an example. Assume that the ego vehicle is driving on a straight road (i.e., $c_0 = c_1 = 0$) and that the look ahead distance of the camera is $\bar{\mathrm{x}}_c$. A new segment begins at the position $\mathrm{x}'_c < \bar{\mathrm{x}}_c$, which means that there is a step in c_1, and c_0 is ramped up, see Figure 3. The penetration into the next segment is $l_c = \bar{\mathrm{x}}_c - \mathrm{x}'_c$. The idea of this model, referred to as *averaging* or *spread-out dynamic curvature model*, with the new state variables c_{0m} and c_{1m}, is that it generates the true

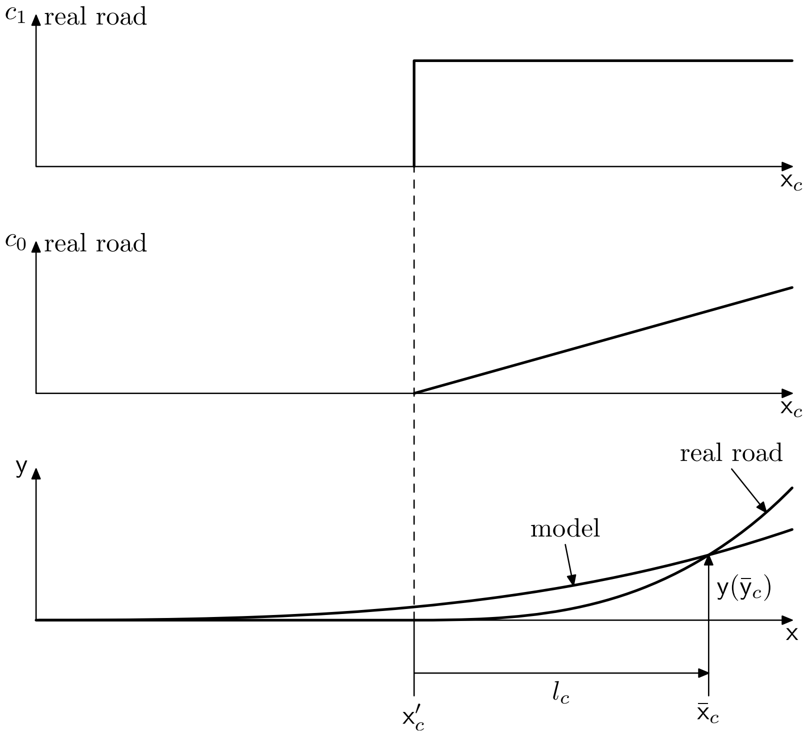


Figure 3: *A straight and a curved road segment are modeled with the averaging road model. The two upper plots show the parameters c_1 and c_0 of the real road, the bottom plot shows the real and the modeled roads in a Cartesian coordinate frame.*

lateral offset $y(\bar{x}_c)$ at the look ahead distance $\bar{x}_c$, i.e.,

$$y_{\text{real}}(\bar{x}_c) = y_{\text{model}}(\bar{x}_c), \tag{19}$$

but it is continuously spread out in the range $(0, \bar{x}_c)$. The lateral offset of the real road as a function of the penetration l_c, for $0 \leq l_c \leq \bar{x}_c$, is

$$y_{\text{real}}(l_c) = \frac{c_1}{6} l_c^3, \tag{20}$$

since the first segment is straight. The lateral offset of the averaging model as a function of the penetration l_c is

$$y_{\text{model}}(l_c) = \frac{c_{0m}(l_c)}{2}\bar{x}_c^2 + \frac{c_{1m}(l_c)}{6}\bar{x}_c^3, \tag{21}$$

at the look ahead distance $\bar{x}_c$. The equation

$$c_1 \frac{l_c^3}{\bar{x}_c^2} = 3c_{0m}(l_c) + c_{1m}(l_c)\bar{x}_c, \tag{22}$$

is obtained by inserting (20) and (21) into (19). By differentiating (22) with respect to l_c and using the relations $\frac{dc_1}{dl_c} = 0$, $\frac{dc_{0m}(l_c)}{dl_c} = c_{1m}(l_c)$ and $\frac{d(\cdot)}{dl_c} = \frac{d(\cdot)}{dt} \cdot \frac{dt}{dl_c}$ the following equation is obtained

$$\dot{c}_{1m} = 3\frac{\mathrm{v}}{\bar{\mathrm{x}}_c}(c_1(l_c/\bar{\mathrm{x}}_c)^2 - c_{1m}), \tag{23}$$

for $l_c < \bar{\mathrm{x}}_c$. Since $(l_c/\bar{\mathrm{x}}_c)^2$ is unknown it is usually set to 1 (Dickmanns, 2007), which finally yields

$$\dot{c}_{1m} = 3\frac{\mathrm{v}}{\bar{\mathrm{x}}_c}(c_1 - c_{1m}). \tag{24}$$

The state variable vector of the averaging model is defined as

$$\mathbf{m} = \begin{bmatrix} c_{0m} \\ c_{1m} \\ c_1 \end{bmatrix} = \begin{bmatrix} \text{curvature at the ego vehicle} \\ \text{averaged curvature derivative} \\ c \text{ derivative of the foremost segment} \end{bmatrix}, \tag{25}$$

and the process model is given by augmenting the simple clothoid model (17) with (24) according to

$$\begin{bmatrix} \dot{c}_{0m} \\ \dot{c}_{1m} \\ \dot{c}_1 \end{bmatrix} = \begin{bmatrix} 0 & \mathrm{v} & 0 \\ 0 & -3\frac{\mathrm{v}}{\bar{\mathrm{x}}_c} & 3\frac{\mathrm{v}}{\bar{\mathrm{x}}_c} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_{0m} \\ c_{1m} \\ c_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ w_{c_1} \end{bmatrix}. \tag{26}$$

The model is driven by the process noise w_{c_1}, which also influences the other states. The averaging model is well described in the recent textbook Dickmanns (2007) and some early results using the model are presented by e.g., Dickmanns and Mysliwetz (1992).

4.2 Mapping of the Road Lanes

The problem of mapping road lanes or lane estimation as it is often called, is a *curve estimation* problem. The task is to obtain the best possible estimate of the curve describing the lane by exploiting the measurements provided by the onboard sensors. The most important sensor type here is exteroceptive sensors, such as for example cameras and lasers. Currently the camera is the most commonly used sensor for the lane estimation problem and in this section we will show how camera images can be used to obtain lane estimates.

The lane estimation problem is by no means a new problem, it has been studied for more than 25 years, see e.g., Waxman et al. (1987); Dickmanns and Mysliwetz (1992) for some early work and Wang et al. (2008) for more recent contributions. A complete overview of what has been done on this problem is available in the survey paper McCall and Trivedi (2006). In this section the problem is broken down into its constituents and one way of solving the lane estimation problem when a camera is used as a sensor is shown.

The lane estimation problem can be separated into two subproblems, commonly referred to as the *lane detection* problem and the *lane tracking* problem. As the name reveals, the lane detection problem deals with detecting lanes in an image.

The lane tracking problem then makes use of the detected lanes together with information about the temporal and the spatial dependencies over time in order to compute lane estimates. These dependencies are mathematically described using a road model. Traditionally, lane tracking is done using an extended Kalman filter, see e.g., Dickmanns and Mysliwetz (1992); Guiducci (1999). There are also interesting approaches based on the particle filter (PF) by Gordon et al. (1993) available, see e.g., Zhou et al. (2006); Wang et al. (2008); Kim (2008).

The lane is here modeled in the image plane (cf. Section 3.2) as a linear function close to the ego vehicle and as a quadratic function far away, i.e.,

$$\mathbf{l}_\theta(\mathsf{v}) = \begin{cases} a + b(\mathsf{v} - \mathsf{v}_s) & \mathsf{v} > \mathsf{v}_s \\ a + b(\mathsf{v} - \mathsf{v}_s) + c(\mathsf{v} - \mathsf{v}_s)^2 & \mathsf{v} \leq \mathsf{v}_s, \end{cases} \tag{27a}$$

where v_s denotes the (known) vertical separation in pixels between the linear and the quadratic model (illustrated by the horizontal line in Figure 4) and subindex θ is used to denote the dependence on the parameters

$$\theta = \begin{bmatrix} a & b & c \end{bmatrix}^{\mathrm{T}}, \tag{27b}$$

which are to be estimates. These parameters all have geometrical interpretations in terms of offset (a), local orientation (b) and curvature (c) of the lane in the image plane. The lane estimates (here, the estimates of the parameters θ in (27)) carry important information about the states in the road model introduced in Section 4.1, which are expressed in world coordinates (in contrast to pixel coordinates u, v). These road model states are typically what we are interested in and we will return to this important connection at the end of this section.

Given the fact that the problem of lane detection has been studied for more than 25 year there are many ideas on how to solve this problem available. Rather than trying to give a complete account of all the different methods available, we will here be very specific and explain one way in which lanes can be detected and show some results on real sequences. The solution presented here is very much along the lines of Jung and Kelber (2005); Lee (2002).

The initial lane detection is performed using a linear lane model, which is found from the image using a combination of the edge distribution function (EDF) (Lee, 2002) and the Hough transform (Hough, 1962; Duda and Hart, 1972). The EDF is defined as the gradient orientation

$$\varphi(\mathsf{u}, \mathsf{v}) = \arctan\left(\frac{D_\mathsf{u}}{D_\mathsf{v}}\right), \tag{28}$$

where D_u and D_v are approximations of the gradient function

$$\nabla I(\mathsf{u}, \mathsf{v}) = \begin{bmatrix} \frac{\partial I}{\partial \mathsf{u}} & \frac{\partial I}{\partial \mathsf{v}} \end{bmatrix}^{\mathrm{T}} \approx \begin{bmatrix} D_u & D_v \end{bmatrix}^{\mathrm{T}} \tag{29}$$

for the gray scale image $I(\mathsf{u}, \mathsf{v})$. The two largest peaks (α^{l}, α^{r},) of $\varphi(\mathsf{u}, \mathsf{v})$ provide the most probable orientations of the lanes in the image. This is used to form an edge image $g(\mathsf{u},\mathsf{v})$ as

$$g(\mathsf{u},\mathsf{v}) = \begin{cases} |\nabla I(\mathsf{u},\mathsf{v})| \approx |D_\mathsf{u}| + |D_\mathsf{v}|, & |\varphi(\mathsf{u},\mathsf{v}) - \alpha^\mathrm{l}| < T_\alpha \text{ or } |\varphi(\mathsf{u},\mathsf{v}) - \alpha^\mathrm{r}| < T_\alpha \\ 0, & \text{otherwise} \end{cases} \tag{30}$$

where T_α is a threshold, here $T_\alpha = 2°$. Applying the Hough transform to the edge image $g(\mathsf{u},\mathsf{v})$ provides two initial linear models $\mathbf{l}(\mathsf{v}) = a + b\mathsf{v}$, one for the left lane markings and one for the right lane markings. These models are used to form a region which will serve as the search region in the subsequent image. This region, which we refer to as the lane boundary region of interest (LBROI) is simply defined by extending the linear model w pixels to the right and w pixels to the left (here $w = 10$).

Given that an initial LBROI is found, the task is now to make use of this information in order to compute estimates of the parameters in the lane model (27),

$$\theta = \begin{bmatrix}(\theta^\mathrm{l})^\mathrm{T} & (\theta^\mathrm{r})^\mathrm{T}\end{bmatrix}^\mathrm{T}, \quad \text{where} \quad \theta^\mathrm{l} = \begin{bmatrix}a^\mathrm{l} & b^\mathrm{l} & c^\mathrm{l}\end{bmatrix}^\mathrm{T}, \quad \theta^\mathrm{r} = \begin{bmatrix}a^\mathrm{r} & b^\mathrm{r} & c^\mathrm{r}\end{bmatrix}^\mathrm{T}, \tag{31}$$

where superscript l and r have been used to indicate the left lane marking and the right lane marking, respectively. Estimates of these parameters θ are obtained by solving a constrained weighted least squares problem for each image. The cost function is given by

$$V(\theta) = \sum_{i=1}^{N} \left(M_i^\mathrm{l}(\mathsf{u}_i^\mathrm{l} - \mathbf{l}_{\theta^\mathrm{l}}(\mathsf{v}_i^\mathrm{l}))^2 + (M_i^\mathrm{r}(\mathsf{u}_i^\mathrm{r} - \mathbf{l}_{\theta^\mathrm{r}}(\mathsf{v}_i^\mathrm{r}))^2 \right), \tag{32}$$

where N denotes the number of relevant pixels in the lateral u direction, $\mathbf{l}$ denotes the lane model given in (27) and M_i denotes the magnitude in the thresholded edge-image $g_2(\mathsf{u},\mathsf{v})$, $M_i = g_2(\mathsf{u}_i,\mathsf{v}_i)$, defined as

$$g_2(\mathsf{u},\mathsf{v}) = \begin{cases} g(\mathsf{u},\mathsf{v}), & g(\mathsf{u},\mathsf{v}) \geq \frac{1}{2} M_\mathrm{mean} \\ 0, & \text{otherwise} \end{cases} \tag{33}$$

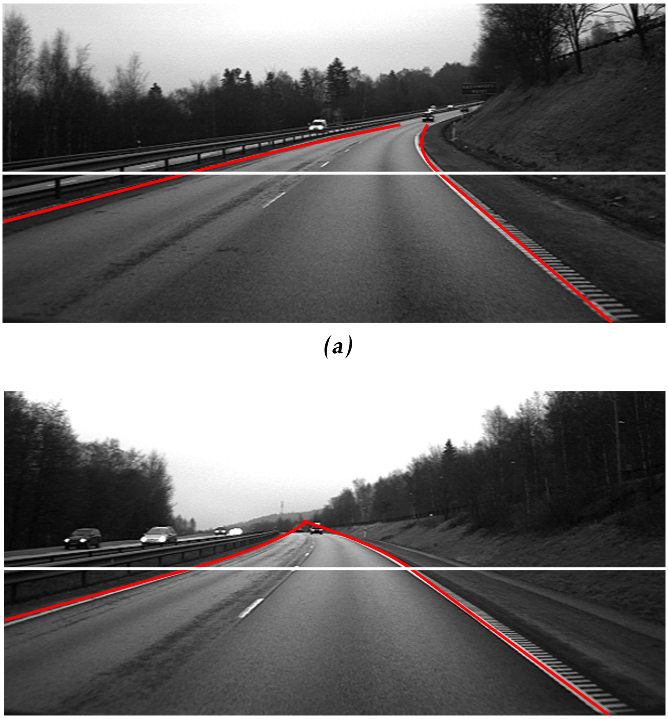
where M_mean denotes the mean magnitude of the $g(\mathsf{u},\mathsf{v})$.

Constraints are introduced in order to account for the fact that the right lane and the left lane are related to each other. This is modelled according to the following linear (in θ) inequality constraint

$$a^\mathrm{r} - a^\mathrm{l} + (b^\mathrm{r} - b^\mathrm{l})(\mathsf{v}_1 - \mathsf{v}_\mathrm{m}) + (c^\mathrm{r} - c^\mathrm{l})(\mathsf{v}_1 - \mathsf{v}_\mathrm{m})^2 \leq \delta, \tag{34}$$

which states that the left and the right lanes cannot be more than δ pixels apart furthest away (at v_1) from the host vehicle. In other words, (34) encodes the fact that the left and the right lanes must have similar geometry in the sense that the quadratic parts in (27) are strongly related.

From (32)–(34) it is now clear that lane estimation boils down to a curve estimation problem, which here is quadratic in the unknown parameters θ. More specifically, inserting the lane model (27) into the cost function (32) and writing the problem on matrix form results in a constrained weighted least squares



(a)

(b)

Figure 4: *Lane estimation results (in red) overlayed onto the camera image. From this figure the lane model* (27) *is clearly illustrated, the model is linear for* $\mathsf{v} > \mathsf{v}_s$ *and quadratic for* $\mathsf{v} \leq \mathsf{v}_s$*. The method assumes that the road surface is flat and when this assumption is true the results are good, see Figure (a). However, when this assumption does not hold the estimates are not that good on a longer horizon, see Figure (b).*

problem on the form

$$\begin{array}{rl} \min\limits_{\theta} & \frac{1}{2}\theta^{\mathrm{T}} H\theta + f^{\mathrm{T}}\theta \\ \text{s.t.} & L\theta \quad \leq \quad \delta. \end{array} \tag{35}$$

This is a quadratic program (QP) implying that a global minimum will be found and there are very efficient solvers available for this type of problems. Here, we have made use of a dual active set method[1] according to Gill et al. (1991). An illustration of the lane estimation results is provided in Figure 4. The estimate of θ is then used to form the LBROI for the new image, simply as region defined by $\mathbf{l}_{\theta}(\mathsf{v}) \pm w$ for each lane.

[1] The QP code was provided by Dr. Adrian Wills at the University of Newcastle, Australia, see http://sigpromu.org/quadprog. This code implements the method described by Goldfarb and Idnani (1983); Powell (1985).

The lane estimates that are now obtained as the solution to (35) can be expressed in world coordinates, seen from the ego vehicle, using geometrical transformation along the lines of what has already been described in Section 3. These transformation are discussed in detail in Guiducci (2000). Once the lane estimates are available in the world coordinates they can be used as camera measurements in a *sensor fusion* framework to make a very important contribution to the estimate of map variables $\mathbf{m}$ (i.e., (15) or (24)) in the road model (perhaps most importantly the curvature c_0 and the curvature derivative c_1) as it is derived in Section 4.1.

4.3 Mapping of the Road Edges

Feature based measurements of landmarks along the road may be used to map the road edges. This section describes a method to track line shaped objects, such as guardrails using point measurements from radar, laser or extracted camera features. Tracking point objects, was covered in Section 3 and is not repeated here. The line shaped and curved guardrails are described using the polynomial road model (18) and tracked as extended targets in a Cartesian frame. However, to allow a more general treatment of the problem in this section the extended targets are modeled using n^{th} order polynomials given as

$$\mathsf{y} = a_0 + a_1\mathsf{x} + a_2\mathsf{x}^2 + \ldots + a_n\mathsf{x}^n, \tag{36}$$

in the range $[\mathsf{x}_{\text{start}}, \mathsf{x}_{\text{end}}]$ where $\mathbf{m}_a \triangleq \begin{bmatrix} a_0 & a_1 & \cdots a_n \end{bmatrix}^{\mathrm{T}}$ are the polynomial coefficients and $\begin{bmatrix} \mathsf{x} & \mathsf{y} \end{bmatrix}^{\mathrm{T}}$ are planar Cartesian coordinates. Note that the coordinate y is a function of x and that the direction of the coordinate frame is chosen dependent on the application in mind. The state vector of a map object j is defined as

$$\mathbf{m}^{(j)} \triangleq \begin{bmatrix} (\mathbf{m}_a^j)^{\mathrm{T}} & \mathsf{x}_{\text{start}}^j & \mathsf{x}_{\text{end}}^j \end{bmatrix}^{\mathrm{T}}. \tag{37}$$

The map $\mathbf{M}$ is modeled by a set of such polynomial shapes according to (1).

Suppose the 2-dimensional noisy feature based sensor measurements are given in batches of Cartesian x and y coordinates as follows

$$\left\{ \mathbf{z}_k^{(i)} \triangleq \begin{bmatrix} \mathsf{x}^{(i)} & \mathsf{y}^{(i)} \end{bmatrix}_k^{\mathrm{T}} \right\}_{i=1}^{N_{\mathbf{z},k}}, \tag{38}$$

for discrete time instants $k = 1, \ldots, K$. In many cases in reality (e.g., radar, laser and stereo vision cf. (5)) and in the practical application considered in this section, the sensor provides range r and azimuth angle ψ given as,

$$\left\{ \bar{\mathbf{z}}_k^{(i)} \triangleq \begin{bmatrix} r^{(i)} & \psi^{(i)} \end{bmatrix}_k^{\mathrm{T}} \right\}_{i=1}^{N_{\mathbf{z},k}}. \tag{39}$$

In such a case some suitable standard polar to Cartesian conversion algorithm is used to convert these measurements into the form (38).

The state model considered in this section is described, in general, by the state space equations

$$\mathbf{m}_{k+1} = f(\mathbf{m}_k, \mathbf{u}_k) + \mathbf{w}_k, \tag{40a}$$

$$\mathbf{y}_k = h(\mathbf{m}_k, \mathbf{u}_k) + \mathbf{e}_k, \tag{40b}$$

where $\mathbf{m}$, $\mathbf{u}$ and $\mathbf{y}$ denote the state, the input signal, and the output signal, while $\mathbf{w} \sim \mathcal{N}(0, Q)$ and $\mathbf{e} \sim \mathcal{N}(0, R)$ are the process and measurement noise, respectively. The use of an input signal $\mathbf{u}$ is important in this framework. For the sake of simplicity, the tracked objects are assumed stationary, resulting in very simple motion models (40a).

A polynomial is generally difficult to handle in a filter, since the noisy measurements are distributed arbitrarily along the polynomial. In this respect, the measurement models considered contain parts of the actual measurement vector as parameters. The methodology takes into account the errors caused by using the actual noisy measurements as model parameters. This scheme is an example of the so called "errors in variables" (EIV) framework, see e.g., Söderström (2007); Diversi et al. (2005); Björck (1996).

The general convention in modeling is to make the definitions

$$\mathbf{y} \triangleq \mathbf{z}, \qquad \mathbf{u} \triangleq \emptyset, \tag{41}$$

where $\emptyset$ denotes the empty set meaning that there is no input. Notice that the subscripts k, specifying the time stamps of the quantities, is omitted for the sake of simplicity, In this setting, it is extremely difficult, if not impossible, to find a measurement model connecting the outputs $\mathbf{y}$ to the states $\mathbf{m}_a$ in the form of (40b). Therefore, other selections for $\mathbf{y}$ and $\mathbf{u}$, need to be used. Here, the selection

$$\mathbf{y} \triangleq \mathsf{y}, \qquad \mathbf{u} \triangleq \mathsf{x}. \tag{42}$$

is made. Although being quite a simple selection, this choice results in a rather convenient linear measurement model in the state partition $\mathbf{m}_a$,

$$\mathbf{y} = H_a(\mathbf{u})\mathbf{m}_a + \mathbf{e}, \tag{43}$$

where $H_a(\mathbf{u}) = \begin{bmatrix}1 & \mathsf{x} & \mathsf{x}^2 & \cdots & \mathsf{x}^n\end{bmatrix}^{\mathrm{T}}$. It is the selection in (42) rather than (41) that allows to use the standard methods in target tracking with clever modifications. Such a selection as (42) is also in accordance with the EIV representations where measurement noise are present in both the outputs and inputs, i.e.,, the observation $\mathbf{z}$ can be partitioned according to

$$\mathbf{z} = \begin{bmatrix}\mathbf{u} \\ \mathbf{y}\end{bmatrix}. \tag{44}$$

The measurement vector given in (38) is expressed in terms of a noise free variable $\mathbf{z}_0$ which is corrupted by additive measurement noise $\tilde{\mathbf{z}}$ according to

$$\mathbf{z} = \mathbf{z}_0 + \tilde{\mathbf{z}}, \quad \tilde{\mathbf{z}} \sim \mathcal{N}(0, \Sigma_{\mathsf{c}}), \tag{45}$$

where the covariance Σ_{c} can be decomposed as

$$\Sigma_{\mathsf{c}} = \begin{bmatrix}\Sigma_{\mathsf{x}} & \Sigma_{\mathsf{xy}} \\ \Sigma_{\mathsf{xy}} & \Sigma_{\mathsf{y}}\end{bmatrix}. \tag{46}$$

Note that, in the case the sensor provides measurements only in polar coordinates

(39), one has to convert both the measurement $\bar{\mathbf{z}}$ and the measurement noise covariance

$$\Sigma_{\mathsf{p}} = \operatorname{diag}(\sigma_d^2, \sigma_\delta^2) \tag{47}$$

into Cartesian coordinates. This is a rather standard procedure. Note that, in such a case, the resulting Cartesian measurement covariance Σ_{c} is, in general, not necessarily diagonal and hence Σ_{xy} of (46) might be non-zero.

Since the model (43) is linear, the Kalman filter measurement update formulas can be used to incorporate the information in $\mathbf{z}$ into the extended source state $\mathbf{m}_a$. An important question in this regard is what would be the measurement covariance of the measurement noise term $\mathbf{e}$ in (43).

Neglecting the errors in the model parameters $H_a(\mathbf{u})$ can cause overconfidence in the estimates of recursive filters and can actually make data association difficult in tracking applications (by causing too small gates). A simple methodology is used to take the uncertainties in $H_a(\mathbf{u})$ into account in line with the EIV framework. Assuming that the elements of the noise free quantity $\mathbf{z}_0$ satisfy the polynomial equation exactly according to

$$\mathbf{y} - \tilde{\mathbf{y}} = H_a(\mathbf{u} - \tilde{\mathbf{u}})\mathbf{m}_a, \tag{48a}$$

$$\mathsf{y} - \tilde{\mathsf{y}} = \begin{bmatrix} 1 & \mathsf{x} - \tilde{\mathsf{x}} & (\mathsf{x} - \tilde{\mathsf{x}})^2 & \cdots & (\mathsf{x} - \tilde{\mathsf{x}})^n \end{bmatrix} \mathbf{m}_a, \tag{48b}$$

which can be approximated using a first order Taylor expansion resulting in

$$\mathbf{y} \approx H_a(\mathbf{u})\mathbf{m}_a - \widetilde{H}_a(\mathbf{u})\tilde{\mathsf{x}}\mathbf{m}_a + \tilde{\mathsf{y}} \tag{49a}$$

$$= H_a(\mathbf{u})\mathbf{m}_a + \tilde{h}_a(\mathbf{m}_a, \mathbf{u}) \begin{bmatrix} \tilde{\mathsf{x}} \\ \tilde{\mathsf{y}} \end{bmatrix}, \tag{49b}$$

with

$$H_a(\mathbf{u}) = \begin{bmatrix} 1 & \mathsf{x} & \mathsf{x}^2 & \cdots & \mathsf{x}^n \end{bmatrix}, \tag{49c}$$

$$\widetilde{H}_a(\mathbf{u}) = \begin{bmatrix} 0 & 1 & 2\mathsf{x} & \cdots & n\mathsf{x}^{n-1} \end{bmatrix}, \tag{49d}$$

$$\tilde{h}_a(\mathbf{m}_a, \mathbf{u}) = \begin{bmatrix} -a_1 - 2a_2\mathsf{x} - \cdots - na_n\mathsf{x}^{n-1} & 1 \end{bmatrix}. \tag{49e}$$

Hence, the noise term $\mathbf{e}$ of (43) is given by

$$\mathbf{e} = \tilde{\mathsf{y}} - \widetilde{H}_a\tilde{\mathsf{x}}\mathbf{m}_a = \tilde{h}(\mathbf{m}_a, \mathbf{u}) \begin{bmatrix} \tilde{\mathsf{x}} \\ \tilde{\mathsf{y}} \end{bmatrix} \tag{50}$$

and its covariance is given by

$$\begin{aligned} \Sigma_a &= \mathrm{E}(\mathbf{e}\mathbf{e}^{\mathrm{T}}) = \Sigma_{\mathsf{y}} + \mathbf{m}_a\widetilde{H}_a\Sigma_{\mathsf{x}}\widetilde{H}_a^{\mathrm{T}}\mathbf{m}_a^{\mathrm{T}} - 2\widetilde{H}_a\Sigma_{\mathsf{xy}} \\ &= \tilde{h}(\mathbf{m}_a, \mathbf{u})\Sigma_{\mathsf{c}}\tilde{h}^{\mathrm{T}}(\mathbf{m}_a, \mathbf{u}). \end{aligned} \tag{51}$$

Note that the EIV covariance Σ_a depends on the state variable $\mathbf{m}_a$, which is substituted by its last estimate in recursive estimation.

Up to this point, only the relation of the observation $\mathbf{z}$ to the state component $\mathbf{m}_a$ has been considered. It remains to discuss the relation between the observation and the start $\mathrm{x}_{\text{start}}$ and the end points x_{end} of the polynomial. The measurement information must only be used to update these components of the state if the new observations of the extended source lie outside the range of the polynomial. The following (measurement dependent) measurement matrix can be defined for this purpose:

$$H_{se} = \begin{cases} \begin{bmatrix} 1 & 0 \end{bmatrix} & \text{if } \mathrm{x} \leq \mathrm{x}_{\text{start},k|k-1} \\ \begin{bmatrix} 0 & 1 \end{bmatrix} & \text{if } \mathrm{x} \geq \mathrm{x}_{\text{end},k|k-1} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & \text{otherwise.} \end{cases} \tag{52}$$

The complete measurement model of an extended object can now be summarized by

$$\mathbf{z} = H\mathbf{m} + \mathbf{e}, \quad \mathbf{e} \sim \mathcal{N}(0, R(\mathbf{m})), \tag{53a}$$

with

$$H = \begin{bmatrix} \mathbf{0}^{1\times n} & H_{se} \\ H_a & \mathbf{0}^{1\times 2} \end{bmatrix}, \tag{53b}$$

$$R(\mathbf{m}) = \operatorname{blkdiag}(\Sigma_{\mathrm{x}}, \Sigma_a(\mathbf{m})). \tag{53c}$$

Put in words, if the x-component of a new measurement is closer to the sensor than the start point of the line $\mathrm{x}_{\text{start}}$ it is considered in the measurement equation (52) and can used to update this state variable. Analogously, if a new measurement is more distant than the end point of the line x_{end} it is considered in (52). Further, if a measurements is in between the start and end point of the line, the measurement model is zero in (52) and there is no relation between this measurement and the state variables $\mathrm{x}_{\text{start}}$ or x_{end}.

Any model as e.g., the standard constant velocity or the coordinated turn model may be used for the targets. For simplicity it is assumed that the targets are stationary in this contribution, thus the process model on the form (40a) is linear and may be written

$$\mathbf{m}_{k+1} = F\mathbf{m}_k + \mathbf{w}_k. \tag{54}$$

To increase the flexibility of the extended object an assumption about the dynamic behavior of its size is made. The size of the extended object is modeled to shrink with a factor $0.9 < \lambda < 1$ according to

$$\mathrm{x}_{\text{start},k+1} = \mathrm{x}_{\text{start},k} + \lambda(\mathrm{x}_{\text{end},k} - \mathrm{x}_{\text{start},k}), \tag{55a}$$

$$\mathrm{x}_{\text{end},k+1} = \mathrm{x}_{\text{end},k} - \lambda(\mathrm{x}_{\text{end},k} - \mathrm{x}_{\text{start},k}), \tag{55b}$$

leading to the following process model for the polynomial

$$F = \begin{bmatrix} \boldsymbol{I}^{n\times n} & \mathbf{0}^{n\times 2} & \\ \mathbf{0}^{2\times n} & 1-\lambda & \lambda \\ & \lambda & 1-\lambda \end{bmatrix}. \tag{56}$$

This shrinking behavior for the polynomials allows for automatic adjustment of the start and end points of the polynomials according to the incoming measurements.

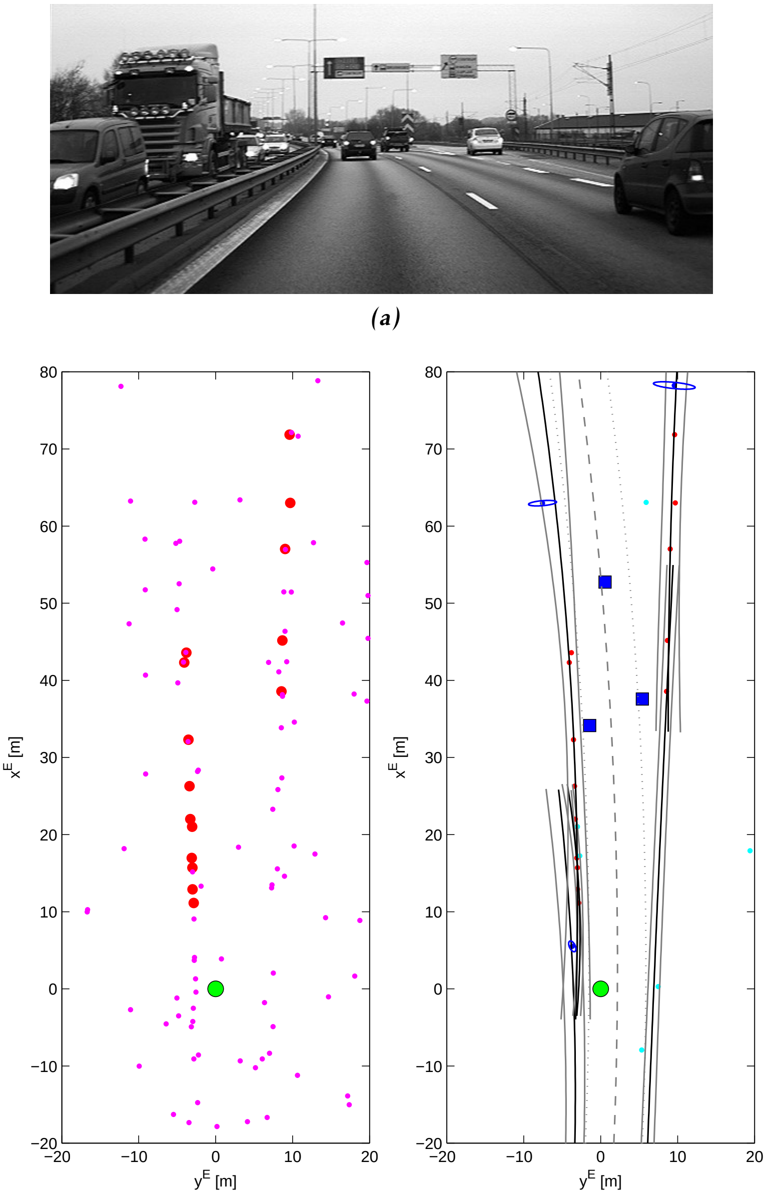
The association of measurements to state estimates is treated in Lundquist et al. (2011b), where a generalized nearest neighbor method is applied.

The section is concluded with some results based on the information given by an ordinary automotive ACC radar, for the traffic situation shown in Figure 5a. The ego vehicle, indicated by a green circle, is situated at the $(0, 0)$-position in Figure 5b, and the red dots are the radar reflections, or stationary observations, at one time sample. The smaller magenta colored dots are former radar reflections, obtained at earlier time samples. Figure 5c shows the estimated points and lines for the same scenario using the KF EIV method presented in this contribution. The mean values of the states are indicated by solid black lines or blue points. Furthermore, the state variance, by means of the 90% confidence interval, is illustrated by gray lines or cyan colored ellipses, respectively. The estimate of the lane markings (18), illustrated by the gray dashed lines and derived according to the method presented in Lundquist and Schön (2010), is shown here as a comparison. Tracked vehicle in front of the ego vehicle are illustrated by blue squares.

5 Occupancy Grid Map

An occupancy grid map is defined over a continuous space and it can be discretized with, e.g., a grid approximation. The size of the map can be reduced to a certain area surrounding the ego vehicle. In order to keep a constant map size while the vehicle is moving, some parts of the map are thrown away and new parts are initiated. Occupancy grid mapping (OGM) is one method for tackling the problem of generating consistent maps from noisy and uncertain data under the assumption that the ego vehicle pose, i.e., position and heading, is known. These maps are very popular in the robotics community, especially for all sorts of autonomous vehicles equipped with laser scanners. Indeed several of the DARPA urban challenge vehicles used OGM's, see Buehler et al. (2008). This is because they are easy to acquire, and they capture important information for navigation. The OGM was introduced by Elfes (1987) and an early introduction is given by Moravec (1988). To the best of the author's knowledge Borenstein and Koren (1991) were the first to utilize OGM for collision avoidance. Examples of OGM in automotive applications are given in Vu et al. (2007). A solid treatment can be found in the recent textbook by Thrun et al. (2005).

This section begins with a brief introduction to occupancy grid maps, according to Thrun et al. (2005). Using this theory and a sensor with high resolution usually gives a nice looking bird eye's view map. However, since a standard automotive radar is used, producing only a few range and bearing measurements at every time sample, some modifications are introduced as described in the following sections.



(a)

(b)

(c)

Figure 5: *A traffic situation is shown in Figure (a). Figure (b) shows the radar measurements, and Figure (c) the resulting tracked points and lines. The circle in the origin is the ego vehicle, the square is the tracked vehicle in front and the dashed gray lines illustrate the tracked road curvature.*

5.1 Background

The planar map $\mathbf{M}$ is defined in the world coordinate frame W and is represented by a matrix. An occupancy grid map is partitioned into finitely many grid cells

$$\mathbf{M} = \{\mathbf{m}^{(j)}\}_{j=1}^{N_\mathbf{m}}. \tag{57}$$

The probability of a cell being occupied $p(\mathbf{m}^{(j)})$ is specified by a number ranging from 1 for occupied to 0 for free. The notation $p(\mathbf{m}^{(j)})$ will be used to refer to the probability that a grid cell is occupied. A disadvantage with this design is that it does not allow for dependencies between neighboring cells.

The occupancy grid map was originally developed to primarily be used with measurements from a laser scanner. A laser is often mounted on a rotating shaft and it generates a range measurement for every angular step of the mechanical shaft, i.e. a bearing angle. This means that the continuously rotating shaft produces many range and bearing measurements during every cycle. The OGM algorithms transform the polar coordinates of the measurements into Cartesian coordinates in a fixed world or map frame. After completing one mechanical measurement cycle the sensor provides the measurements for use.

The algorithm loops through all cells and increases the occupancy probability $p(\mathbf{m}^{(j)})$ if the cell was occupied according to the measurement $\mathbf{z}_k^{(i)}$. Otherwise the occupancy value either remains unchanged or is decreased, depending on if the range to the cell is greater or less than the measured range. The latter implies that the laser beam did pass this cell without observing any obstacles. If the measured range is too large or the cell size is too small, it might be necessary to consider the angular spread of the laser beam and increase or decrease the occupancy probability of several cells with respect to the beam width.

The map is assumed to be static, i.e., it does not change during sensing. In this section the map estimation problems is solved with a binary Bayes filter, of which OGM is one example. In this case the estimation problem is solved with the binary Bayes filter, of which OGM is one example. the state can either be free $\mathbf{m}^{(j)} = 0$ or occupied $\mathbf{m}^{(j)} = 1$. A standard technique to avoid numerical instabilities for probabilities close to 0 and to avoid truncation problems close to 0 and 1 is to use the log odds representation of occupancy

$$\ell_{j,k} = \log \frac{p(\mathbf{m}^{(j)}|\mathbf{Z}_{1:k}, \mathbf{x}_{\mathrm{E},1:k})}{1 - p(\mathbf{m}^{(j)}|\mathbf{Z}_{1:k}, \mathbf{x}_{\mathrm{E},1:k})}, \tag{58}$$

or put in words, the odds of a state is defined as the ratio of the probability of this event $p(\mathbf{m}^{(j)}|\mathbf{Z}_{1:k}, \mathbf{x}_{\mathrm{E},1:k})$ divided by the probability of its complement $1 - p(\mathbf{m}^{(j)}|\mathbf{Z}_{1:k}, \mathbf{x}_{\mathrm{E},1:k})$. The probabilities are easily recovered using

$$p(\mathbf{m}^{(j)}|\mathbf{Z}_{1:k}, \mathbf{x}_{\mathrm{E},1:k}) = 1 - \frac{1}{1 + \exp \ell_{j,k}}. \tag{59}$$

Note that the filter uses the inverse measurement model $p(\mathbf{m}|\mathbf{z}, \mathbf{x})$. Using Bayes' rule it can be shown that the binary Bayes filter in log odds form is

$$\ell_{j,k} = \ell_{j,k-1} + \log \frac{p(\mathbf{m}^{(j)}|\mathbf{Z}_k, \mathbf{x}_{\mathrm{E},k})}{1 - p(\mathbf{m}^{(j)}|\mathbf{Z}_k, \mathbf{x}_{\mathrm{E},k})} - \log \frac{p(\mathbf{m}^{(j)})}{1 - p(\mathbf{m}^{(j)})}, \tag{60}$$

where $p(\mathbf{m}^{(j)})$ represents the prior probability. The log odds ratio of the prior before processing any measurements is defined as

$$\ell_{j,0} = \log \frac{p(\mathbf{m}^{(j)})}{1 - p(\mathbf{m}^{(j)})}. \tag{61}$$

Typically $p(\mathbf{m}^{(j)}) = 0.5$ is assumed, since before having measurements nothing is known about the surrounding environment. This value yields $\ell_0 = 0$.

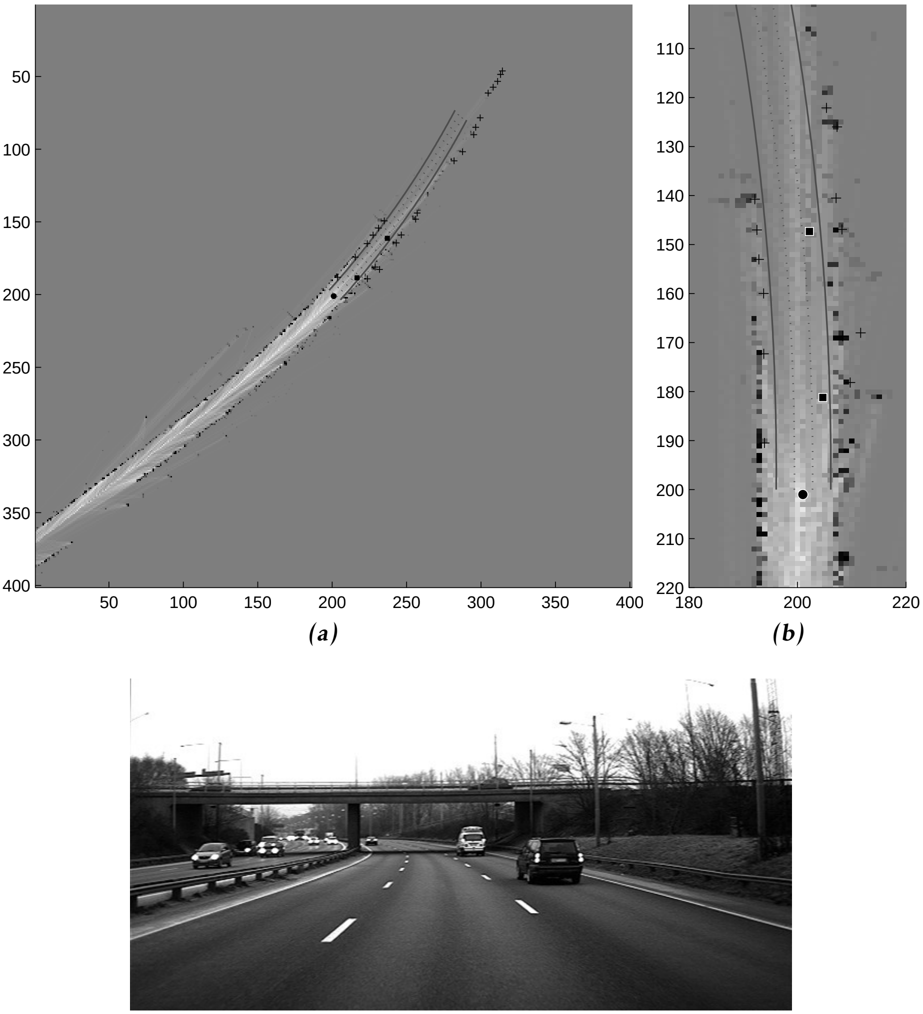
5.2 OGM with Radar Measurements

The radar system provides range and bearing measurements for observed targets at every measurement cycle. The main difference to a laser is that there is not one range measurement for every angular position of the moving sensor. The number of observations depends on the environment. In general there are much fever observations compared to a laser sensor. There is also a limit (usually around $32-64$) on the number of objects transmitted by the radar equipment on the CAN-bus, and a proprietary selection is perform in the radar. Moving objects, which are distinguished by measurements of the Doppler shift, are prioritized and more likely to be transmitted than stationary objects. Furthermore, it is assumed that the opening angle of the radar beam is small compared to the grid cell size. With these the OGM algorithm is changed to loop through the measurements instead of the cells, in order to decrease the computational load. A radar's angular uncertainty is usually larger than its range uncertainty. When transforming the polar coordinates of the radar measurements into the Cartesian coordinates of the map, the uncertainties can either be transformed in the same manner or it can simply be assumed that the uncertainty increases with the range.

5.3 Experiments and Results

Figure 6a shows an OGM example of a highway situation. The ego vehicle's camera view is shown in Figure 6c. The size of the OGM is 401 × 401 m, with the ego vehicle in the middle cell. Each cell represents a 1×1 m square. The gray-level in the occupancy map indicates the probability of occupancy $p(\mathbf{M}|\mathbf{Z}_{1:k}, \mathbf{x}_{\mathrm{E},1:k})$, the darker the grid cell, the more likely it is to be occupied. The map shows all major structural elements as they are visible at the height of the radar. This is a problem if the road is undulated and especially if the radar observes obstacles over and behind the guardrail. In this case the occupancy probability of a cell might be decreased even though it was previously believed to be occupied, since the cell is between the ego vehicle and the new observation. The impact of this problem can be reduced by tuning the filter well.

It is clearly visible in Figure 6a that the left border is sharper than the right. The only obstacle on the left side is the guardrail, which gives rise to the sharp edge,



(a)

(b)

(c)

Figure 6: *The filled circle at position* (201, 201) *in the occupancy grid map in Figure (a) is the ego vehicle, the* + *are the radar observations obtained at this time sample, the black squares are the two leading vehicles that are currently tracked. Figure (b) shows a zoom of the* OGM *in front of the ego vehicle. The gray-level in the figure indicates the probability of occupancy, the darker the grid cell, the more likely it is to be occupied. The shape of the road is given as solid and dashed lines, calculated as described in Section 4. The camera view from the ego vehicle is shown in Figure (c), the concrete walls, the guardrail and the pillar of the bridge are interesting landmarks. Furthermore, the two tracked leading vehicles are clearly visible in the right lane.*

whereas on the right side there are several obstacles behind the guardrail, which also cause reflections, e.g., noise barrier and vegetation. A closer look in Figure 6b reveals that there is no black line of occupied cells representing the guardrail as expected. Instead there is a region with mixed probability of occupancy and after about 5 m the gray region with initial valued cells tell us that nothing is known about these cells.

6 Intensity Based Map

The bin-occupancy filter, which is described in Erdinc et al. (2009), aims at estimating the probability of a target being in a given point. The approach is derived via a discretized state-space model of the surveillance region, where each grid cell (denoted bin in this approach) can or may not contain a target. One of the important assumptions is that the bins are sufficiently small so that each bin is occupied by maximum one target. In the limiting case, when the volume of the bins $|\nu|$ tends to zero, it is possible to define the bin-occupancy density

$$D_{k|k} \triangleq \lim_{|\nu|\to 0} \frac{\Pr(\mathbf{m}_k^{(j)} = 1|\mathbf{Z}_{1:k})}{|\nu|}, \tag{62}$$

where $\Pr(\mathbf{m}_k^{(j)} = 1|\mathbf{Z}_{1:k})$ is the probability that bin j is occupied by one target. The continuous form of the bin-occupancy filter prediction and update equations are the same as the probability hypothesis density (PHD) filter equations (Erdinc et al., 2009). Furthermore, the PHD is the first moment density or *intensity density* in point process theory, see e.g., Mahler (2007), and a physical interpretation is given in Daley and Vere-Jones (2003) as the probability that one target is located in the infinitesimal region $(\mathbf{x}, \mathbf{x} + d\mathbf{x})$ of the state space, divided by $d\mathbf{x}$. The continuous form of the physical bin model leads us to a continuous location based map which we denote *intensity-based map*, and intend to estimate with the PHD filter.

The bin occupancy filter or the PHD filter was developed for target tracking of point sources, however the aim in this section is to create a probabilistic location based map of the surroundings of a moving vehicle. One of the main differences between standard target tracking problems and the building of a location based map, is that many objects such as, guardrails or walls, are typically not point targets, but extended targets (Mahler, 2007; Gilholm and Salmond, 2005). Furthermore, there is no interest in estimating the number of objects in the map, and there is also no interest in keeping track of specific objects. Nevertheless, the bin-occupancy filter attempts to answer the important question: "Is there an object (target) at a given point?". Erdinc et al. (2009) poses the following assumptions for the bin occupancy filter:

1. The bins are sufficiently small so that each bin is occupied by at most one target.
2. One target gives rise to only one measurement.

3. Each target generates measurements independently.
4. False alarms are independent of target originated measurements.
5. False alarms are Poisson distributed.

Here, only point 2 needs some extra treatment if the aim of the algorithm is mapping and not target tracking. It can be argued that the measurements of point sources belongs to extended objects and that the aim is to create a map of those point sources. Also for mapping purposes, the assumption that there will not be two measurements from the same point at the same time is justified. The relation described is modeled by a likelihood function $p(\mathbf{Z}_k|\mathbf{M}_{k|k})$, which maps the Cartesian map to polar point measurements.

So far in this section the discussion has been quite general and the PHD or the intensity has only been considered as a surface over the surveillance region. The first practical algorithms to realize the PHD filter prediction and measurement update equations were based on the particle filter (PF), see e.g., Vo et al. (2003); Sidenbladh (2003), where the PHD is approximated by a large set of random samples (particles). A Gaussian mixture approximation of the PHD (GM-PHD) was proposed by Vo and Ma (2006). The mixture is represented by a sum of weighted Gaussian components and in particular the mean and covariance of those components are propagated by the Kalman filter. In this work we represent the intensity by a Gaussian mixture, since the parametrization and derivation is simpler than for a particle filter based solution. The modeling of the intensity through a number of Gaussian components also makes it simpler to account for structures in the map. We will return to these structures in the next two sections.

The GM-PHD filter estimates the posterior intensity, denoted $D_{k|k}$, as a mixture of Gaussian densities as,

$$D_{k|k} = \sum_{i=1}^{J_{k|k}} w_{k|k}^{(i)} \mathcal{N}\left(m_{k|k}^{(i)}, P_{k|k}^{(i)}\right), \tag{63}$$

where $J_{k|k}$ is the number of Gaussian components and $w_{k|k}^{(i)}$ is the expected number of point sources covered by the density $\mathcal{N}\left(m_{k|k}^{(i)}, P_{k|k}^{(i)}\right)$. In Lundquist et al. (2011a) it is shown how the intensity is estimated with the GM-PHD filter. The Gaussian components are parametrized by a mean $m_{k|k}^{(i)}$ and a covariance $P_{k|k}^{(i)}$, which are expressed in a planar Cartesian coordinate frame, according to

$$m_k^{(i)} = \begin{bmatrix} \mathsf{x}_k^{(i)} & \mathsf{y}_k^{(i)} \end{bmatrix}^{\mathrm{T}}. \tag{64}$$

The aim of the mapping algorithm is to estimate the posterior density (3). The considered intensity based map is continuous over the surveillance region, thus, for the number of elements in (1) it holds that $N_{\mathbf{m}} \to \infty$. Furthermore, the intensity is a summary statistic of the map according to

$$p(\mathbf{M}_k|\mathbf{Z}_{1:k}) \sim p(\mathbf{M}_k; D_{k|k}), \tag{65}$$

see e.g., Mahler (2003), and the estimated intensity $D_{k|k}$ is parametrized by

$$\mu_k^{(i)} \triangleq \left\{ w_k^{(i)}, m_k^{(i)}, P_k^{(i)} \right\} \tag{66}$$

of the Gaussian sum (63). The intensity based map is a multimodal surface with peaks around areas with many sensor reflections or point sources. It is worth observing that the map $\mathbf{M}$ is described by a location based function (63), with feature based parametrization (66).

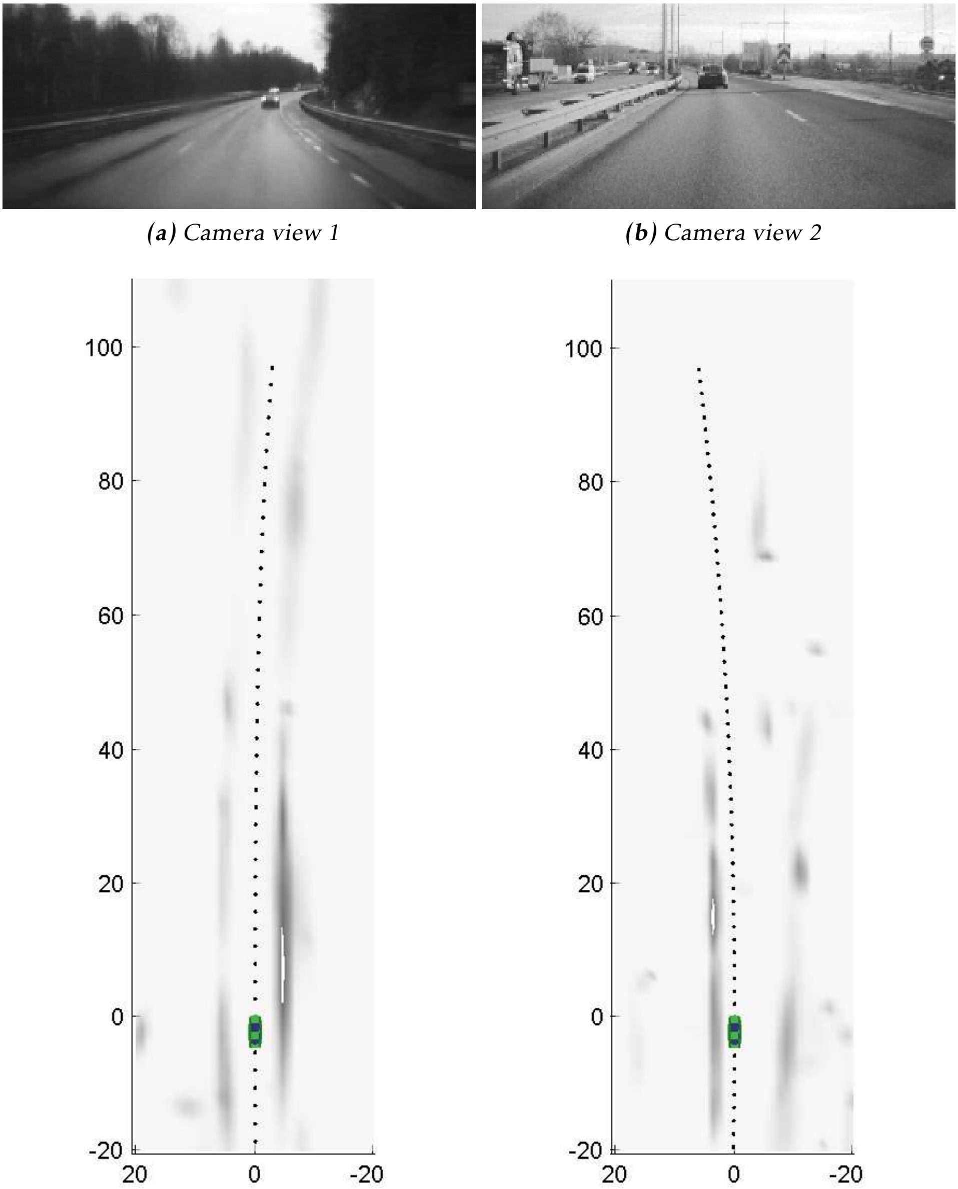
Experiments were conducted with a prototype passenger car. One example of the estimated intensity at a freeway traffic scenario is shown as a bird's eye view in Figure 7c. Darker regions illustrate higher concentrations of point sources, which in this figure stem from the guardrails to the left and the right of the road. As expected, the path of the ego vehicle, indicated by the black dots, is in between the two regions of higher object concentration. The driver's view is shown in Figure 7a.

A second example is shown in Figure 7d and 7b. Here, the freeway exit is clearly visible in the intensity map, which shows that the proposed method to create maps is very conformable.

The Gaussian components are generally removed from the filter when the vehicle passed those parts of the map. However, to give a more comprehensive overview, these components are stored and the resulting intensity based map is shown together with an occupancy grid map (OGM) and a flight photo in Figure 8. The top figure is the map produced as described in this section. The OGM, described in the precious Section 5, is based on the same data set and used as a comparison. The gray-level of the OGM indicates the probability of occupancy, the darker the grid cell the more likely it is to be occupied. As seen in the figure the road edges are not modeled as distinct with the OGM. The OGM representation of the map is not very efficient, since huge parts of the map are gray indicating that nothing is known about these areas. An OGM matrix with often more than 10000 elements must be updated and communicated to other safety functions of a car at each time step. The compact representation is an advantage of the intensity based map. Each Gaussian components is parametrized with 7 scalar values according to (66). Since most maps are modeled with $10 - 30$ components it summarizes to around $70 - 210$ scalar values, which easily can be sent on the vehicles CAN bus to other safety functions. Finally, the bottom photo is a very accurate flight photo (obtained from the Swedish mapping, cadastral and land registration authority), which can be used as ground truth to visualize the quality of the intensity based map.

7 Conclusion

The use of radar, laser and camera for situation awareness is gaining popularity in automotive safety applications. In this chapter it has been shown how sensor data perceived from the ego vehicle is used to estimate a map describing the local



***(a)** Camera view 1*

***(b)** Camera view 2*

***(c)** Intensity map 1*

***(d)** Intensity map 2*

***Figure 7:** The image in (a) shows the driver's view of the intensity map in (c), and the image in (b) is the driver's view of the intensity map in (d). The darker the areas in the intensity map, the higher the concentration of objects. The drivers path is illustrated with black dots and may be used as a reference. Note that snap shot in (d) and (b) is obtained only some meters after the situation shown in Figure 1.*

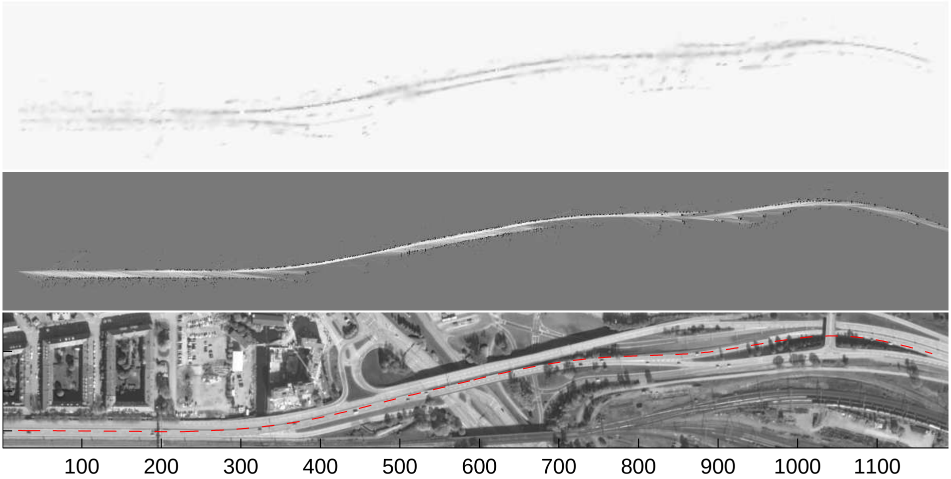


Figure 8: *The top figure shows the intensity based map obtained from radar measurements collected on a freeway. The* OGM *in the middle figure serves as a comparison of an existing algorithm. The bottom figure is a flight photo used as ground truth, where the driven trajectory is illustrated with a dashed line (©Lantmäteriet Gävle 2010. Medgivande I 2011/0100. Reprinted with permission). Note that the drivers view at* 295 meter *is shown in Figure 7b, and about the same position is also shown in Figure 1 and Figure 5.*

surroundings of a vehicle. The map may be modeled in various different ways, of which four major approaches have been described. In a feature based map each element of the map specifies the properties and location of one object. This can either be a point source in the space; or it can be an extended object such as the position and shape of the lane or the road edges. Furthermore, in a location based map the index of each element corresponds to a location and the value of the map element describes the property in that position. One example is the occupancy grid map, which is defined over a continuous space but discretized with a grid approximation. Another example is the intensity based map, which is a continuous approximation, describing the density of objects in the map. The four approaches presented in this chapter have all been evaluated on real data from both freeways and rural roads in Sweden.

The current accuracy of GPS receivers is acceptable only for route guidance, where the provided global position is sufficient. For automotive active safety systems, the local position of the ego vehicle with respect to its surroundings is more important. The estimated maps, described in this chapter, can be used to increase the localization accuracy of the ego vehicle. Furthermore, the maps may be used to derive a collision free trajectory for the vehicle.

Bibliography

Å Björck. *Numerical methods for least squares problems.* SIAM, Philadelphia, PA, USA, 1996.

S. S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems.* Artech House, Norwood, MA, USA, 1999.

J. Borenstein and Y. Koren. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288, June 1991.

M. Buehler, K. Iagnemma, and S. Singh, editors. *Special Issue on the 2007 DARPA Urban Challenge, Part I-III*, volume 25 (8–10). Journal of Field Robotics, 2008.

D. Caveney. Cooperative vehicular safety applications. *IEEE Control Systems Magazine*, 30(4):38–53, August 2010.

J. Civera, A. J. Davison, and J. Montiel. Inverse depth parametrization for monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932–945, October 2008.

D. J. Daley and D. Vere-Jones. *An introduction to the theory of point processes. Vol. 1 , Elementary theory and method.* Springer, New York, NY, USA, 2 edition, 2003.

A. J. Davison, I. Reid, N. Molton, and O. Strasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Patterns Analysis and Machine Intelligence*, 29(6):1052–1067, June 2007.

E. D. Dickmanns. Dynamic computer vision for mobile robot control. In *Proceedings of the International Symposium on Industrial Robots*, Sydney, Australia, November 1988.

E. D. Dickmanns. *Dynamic Vision for Perception and Control of Motion.* Springer, London, UK, 2007.

E. D. Dickmanns and B. D. Mysliwetz. Recursive 3-D road and relative ego-state recognition. *IEEE Transactions on pattern analysis and machine intelligence*, 14(2):199–213, February 1992.

R. Diversi, R. Guidorzi, and U. Soverini. Kalman filtering in extended noise environments. *IEEE Transactions on Automatic Control*, 50(9):1396–1402, September 2005.

R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, January 1972.

A. Eidehall, J. Pohl, F. Gustafsson, and J. Ekmark. Toward autonomous collision avoidance by steering. *IEEE Transactions on Intelligent Transportation Systems*, 8(1):84–94, March 2007.

A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 3(3):249–265, June 1987.

O. Erdinc, P. Willett, and Y. Bar-Shalom. The bin-occupancy filter and its connection to the PHD filters. *IEEE Transactions on Signal Processing*, 57(11): 4232–4246, November 2009.

K. Gilholm and D. Salmond. Spatial distribution model for tracking extended objects. *IEE Proceedings of Radar, Sonar and Navigation*, 152(5):364–371, October 2005.

P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. Inertia-controlling methods for general quadratic programming. *SIAM Review*, 33(1):1–36, March 1991.

D. Goldfarb and A. Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, 27(1):1–33, 1983.

N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings on Radar and Signal Processing*, 140(5):107–113, April 1993.

A. Guiducci. Parametric model of the perspective projection of a road with applications to lane keeping and 3D road reconstruction. *Computer Vision and Image Understanding*, 73(3):414–427, March 1999.

A. Guiducci. Camera calibration for road applications. *Computer Vision and Image Understanding*, 79(2):250–266, August 2000.

C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference*, pages 147–151, Manchester, UK, September 1988.

P. V. C. Hough. A method and means for recognizing complex patterns. U.S. Patent No. 3069654, December 1962.

C. R. Jung and C. R. Kelber. Lane following and lane departure using a linear-parabolic model. *Image and Vision Computing*, 23(13):1192–1202, November 2005.

Z. W. Kim. Robust lane detection and tracking in challenging scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):16–26, March 2008.

J. W. Lee. A machine vision system for lane-departure detection. *Computer vision and image understanding*, 86(1):52–78, April 2002.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

C. Lundquist and T. B. Schön. Joint ego-motion and road geometry estimation. *Information Fusion*, 2010. DOI: 10.1016/j.inffus.2010.06.007.

C. Lundquist, L. Hammarstrand, and F. Gustafsson. Road intensity based mapping using radar measurements with a probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 59(4):1397–1408, April 2011a.

C. Lundquist, U. Orguner, and F. Gustafsson. Extended target tracking using polynomials with applications to road-map estimation. *IEEE Transactions on Signal Processing*, 59(1):15–26, January 2011b.

C. Lundquist, T. B. Schön, and F. Gustafsson. Situational awareness and road prediction for trajectory control applications. In A. Eskandarian, editor, *Handbook of Intelligent Vehicles*, chapter 24. Springer, November 2011c.

R. P. S. Mahler. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, October 2003.

R. P. S. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, Boston, MA, USA, 2007.

J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):731–767, September 2004.

J. C. McCall and M. M. Trivedi. Video-based lane estimation and tracking for driver assistance: Servey, system, and evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):20–37, March 2006.

H. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9 (2):61–74, 1988. ISSN 0738-4602.

D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20, January 2006.

M. J. D. Powell. On the quadratic programming algorithm of Goldfarb and idnani. *Mathematical Programming Study*, 25(1):46–61, October 1985.

H. Rohling and M.-M. Meinecke. Waveform design principles for automotive radar systems. In *Proceedings on CIE International Conference on Radar*, pages 1–4, Beijing, China, October 2001.

H. Rohling and C. Möller. Radar waveform for automotive radar systems and applications. In *IEEE Radar Conference*, pages 1–4, Rome, Italy, May 2008.

H. Sidenbladh. Multi-target particle filtering for the probability hypothesis density. In *Proceedings of the International Conference on Information Fusion*, volume 2, pages 800–806, Cairns, Australia, March 2003.

T. Söderström. Survey paper: Errors-in-variables methods in system identification. *Automatica*, 43(6):939–958, June 2007.

R. Szeliski. *Computer vision : algorithms and applications*. Springer, New York, NY, USA, 2010.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, Cambridge, MA, USA, 2005.

B.-N. Vo and W.-K. Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, November 2006.

B.-N. Vo, S. Singh, and A. Doucet. Random finite sets and sequential Monte Carlo methods in multi-target tracking. In *Proceedings of the International Radar Conference*, pages 486–491, Adelaide, Australia, September 2003.

T. D. Vu, O. Aycard, and N. Appenrodt. Online localization and mapping with moving object tracking in dynamic outdoor environments. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 190–195, Istanbul, Turkey, June 2007.

Y. Wang, L. Bai, and M. Fairhurst. Robust road modeling and tracking using condensation. *IEEE Transactions on Intelligent Transportation Systems*, 9(4): 570–579, December 2008.

A. Waxman, J. LeMoigne, L. Davis, B. Srinivasan, T. Kushner, Eli Liang, and T. Siddalingaiah. A visual navigation system for autonomous land vehicles. *IEEE Journal of Robotics and Automation*, 3(2):124–141, April 1987.

Y. Zhou, R. Xu, X. Hu, and Q. Ye. A robust lane detection and tracking method based on computer vision. *Measurement science and technology*, 17(4):736–745, April 2006.

Paper B

Joint Ego-Motion and Road Geometry Estimation

Authors: Christian Lundquist and Thomas B. Schön

Edited version of the paper:

C. Lundquist and T. B. Schön. Joint ego-motion and road geometry estimation. *Information Fusion*, 12:253–263, October 2011.

The paper presents data that was previously published in:

C. Lundquist and T. B. Schön. Road geometry estimation and vehicle tracking using a single track model. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 144–149, Eindhoven, The Netherlands, June 2008.

Preliminary version:

Technical Report LiTH-ISY-R-2844, Dept. of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden.

Joint Ego-Motion and Road Geometry Estimation

Christian Lundquist and Thomas B. Schön

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

Abstract

We provide a sensor fusion framework for solving the problem of joint ego-motion and road geometry estimation. More specifically we employ a sensor fusion framework to make systematic use of the measurements from a forward looking radar and camera, steering wheel angle sensor, wheel speed sensors and inertial sensors to compute good estimates of the road geometry and the motion of the ego vehicle on this road. In order to solve this problem we derive dynamical models for the ego vehicle, the road and the leading vehicles. The main difference to existing approaches is that we make use of a new dynamic model for the road. An extended Kalman filter is used to fuse data and to filter measurements from the camera in order to improve the road geometry estimate. The proposed solution has been tested and compared to existing algorithms for this problem, using measurements from authentic traffic environments on public roads in Sweden. The results clearly indicate that the proposed method provides better estimates.

1 Introduction

We are in this paper concerned with the problem of integrated ego-motion and road geometry estimation using information from several sensors. The sensors used to this end are a forward looking camera and radar, together with inertial sensors, a steering wheel sensor and wheel speed sensors. The solution is obtained by casting the problem within an existing sensor fusion framework. An important part of this solution is the nonlinear state-space model. The state-space model contains the dynamics of the ego vehicle, the road geometry, the leading vehicles and the measurement relations. It can then be written in the form

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k, \tag{1a}$$

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{e}_k, \tag{1b}$$

where $\mathbf{x}_k \in \mathbb{R}^{n_x}$ denotes the state vector, $\mathbf{u}_k \in \mathbb{R}^{n_u}$ denotes the input signals, $\mathbf{y}_k \in \mathbb{R}^{n_y}$ denotes the measurements, $\mathbf{w}_k \in \mathbb{R}^{n_w}$ and $\mathbf{e}_k \in \mathbb{R}^{n_e}$ denote the process and measurement noise, respectively. The process model equations, describing the evolution of the state over time are denoted by $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$. Furthermore, the measurement model describing how the measurements from the vision system, the radar and the inertial sensors relate to the state is given by $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_y}$. When we have a model in the form (1) we have transformed the problem into a standard nonlinear state estimation problem, where the task is to compute estimates of the state based on the information in the measurements. There are many different ways of solving this problem and we will in this work make use of the popular Extended Kalman Filter (EKF), described in e.g., Smith et al. (1962); Schmidt (1966); Anderson and Moore (1979).

The problem studied in this paper is by no means new, it is the proposed solution that is new. For some early, still very interesting and relevant work on this problem we refer to Dickmanns and Zapp (1986); Dickmanns and Mysliwetz (1992). From the camera we can produce estimates of the road geometry based on measurements of the lane markings. This problem is by now rather mature, see e.g., the survey McCall and Trivedi (2006) and the recent book Dickmanns (2007) for solid accounts. The next step in the development was to make use of the radar information as well. Using radar measurements we can track the leading vehicles, that is, we can estimate the position and velocity of the leading vehicles. Under the assumption that the leading vehicles drive on the same road as the ego vehicle, their positions contain valuable information about the road geometry. This idea was introduced by Zomotor and Franke (1997); Gern et al. (2000, 2001) and has been further refined in Eidehall et al. (2007); Eidehall (2007). The combination of radar and vision as well as the advantages and disadvantages of these sensors are discussed in Hofmann et al. (2000, 2003). Furthermore, the ego vehicle model in Hofmann et al. (2000, 2003) is comparable with the one used in the present work. The four wheel speeds are used to estimate the path of the ego vehicle, which unlike the present work is separated from the leading vehicles dynamics and the lane estimate.

The leading vehicles are used to improve the road geometry in the present work; however the opposite is also possible as the recent work Schubert et al. (2009); Weigel et al. (2009) shows, where the vehicle detection algorithm benefits from the lane information. Vision and radar are used in Schubert et al. (2009), whereas vision and lidar are used in Weigel et al. (2009). In Muller et al. (2009) lidar is used to detect the leading vehicle, and the movement of the leading vehicle is then used to estimate the lane and the driven path, which in turn is used to autonomously follow this vehicle. This works well even for curved and narrow roads. Unmarked and winding rural roads may be hard to detect, recent research in this area is presented in Loose et al. (2009), where stereo vision and image radar are used within a marginalized particle filter to obtain 3D information and improve the task of lane recognition. Information obtained from road-side structures may be used to improve the estimate of the lane shape and the position of the vehicle within the lane, as showed in Watanabe et al. (2009), where only a monocular camera is used. Furthermore, at construction sites it is hard to identify the temporary lanes, a method for this using color images and beacon extraction is presented in Gumpp et al. (2009). In Wedel et al. (2008) the authors present an algorithm for free space estimation, capable of handling non-planar roads, using a stereo camera system.

Lane tracking has also been tackled using radar sensors, see e.g., Kaliyaperumal et al. (2001); Lakshmanan et al. (1997); Nikolova and Hero (2000); Ma et al. (2000) and laser sensors, see e.g. Wijesoma et al. (2004). There have been several approaches making use of reflections from the road boundary, such as crash barriers and reflection posts, to compute information about the free space, see e.g. Kirchner and Heinrich (1998); Kirchner and Ameling (2000); Sparbert et al. (2001) for some examples using laser scanners and Lundquist et al. (2009), where radar is used.

To summarize, our approach is able to improve the performance by making use of a dynamic model of the ego vehicle and a new dynamic model of the road at the same time as we make use of the motion of the leading vehicles. The new road process model describes the curvature of the ego vehicle's currently driven path. This should be compared with existing road models, used in most of the publications mentioned above, where the road's curvature is modeled according to road construction standards. The advantage of our new road model is that we are able to directly include information of the ego vehicles motion into the estimate of the road geometry.

In the subsequent section we provide a brief introduction to the sensor fusion framework we work with and explain how the present problem fits into this framework. An essential part of this framework is the dynamical model (1a), which is derived in Section 3. Furthermore, the corresponding measurement model (1b) is introduced in Section 4. In Section 5 the proposed solution is evaluated using measurements from real and relevant traffic environments from public roads in Sweden. Finally, the conclusions are given in Section 6. For convenience we provide a list of the relevant notation in the appendix.

2 Sensor Fusion

In order to successfully solve the problem under study in this work it is imperative to have a good understanding of sensor fusion. Sensor fusion is defined as the process of using information from *several different sensors* to compute an *estimate* of the state of a *dynamical system*.

We need a dynamic model and a measurement model in the form (1) in order to be able to produce an estimate of the state. These models are derived in detail in Section 3 and Section 4. However, for the sake of the present discussion we will briefly discuss the model here. The state vector $\mathbf{x}_k$ consists of three parts according to

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{\mathrm{E},k} \\ \mathbf{x}_{\mathrm{R},k} \\ \mathbf{x}_{\mathrm{T},k} \end{bmatrix}, \tag{2}$$

where $\mathbf{x}_{\mathrm{E},k}$ denotes the state of the ego vehicle, $\mathbf{x}_{\mathrm{R},k}$ denotes the state of the road and $\mathbf{x}_{\mathrm{T},k}$ denotes the state of one leading vehicle (also referred to as a target). In deriving the evolution of these states over time we will end up with continuous-time differential equations in the form

$$\dot{\mathbf{x}}(t) = g(\mathbf{x}(t), \mathbf{u}(t)). \tag{3}$$

However, according to (1) we required the model to be in discrete time. The simplest way of obtaining a difference equation from (3) is to make use of the standard forward Euler method, which approximates (3) at time t according to

$$\mathbf{x}(t+T) = \mathbf{x}(t) + Tg(\mathbf{x}(t), \mathbf{u}(t)) \triangleq f(\mathbf{x}_t, \mathbf{u}_t), \tag{4}$$

where T denotes the sample time. The measurement model is of course already in discrete time.

The *estimate* of the state is computed by a state estimator of some kind. This state estimator makes use of the measurements from the different sensors to produce an estimate of the so called filtering probability density function (PDF) $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, where $\mathbf{y}_{1:k} \triangleq \{\mathbf{y}_i\}_{i=1}^k$ denotes all the measurements from time 1 to time k. This density function contains all there is to know about the state $\mathbf{x}_k$, given the information in the measurements $\mathbf{y}_{1:k}$. Once an approximation of $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ is available it can be used to form many different estimates and the most commonly used estimate is the conditional mean estimate

$$\hat{\mathbf{x}}_{k|k} = \mathrm{E}(\mathbf{x}_k|\mathbf{y}_{1:k}). \tag{5}$$

This estimate will be used in the present work as well.

Since we are looking for an algorithm capable of working in real-time it is important to understand how the filtering PDF evolves over time. Now, it is well-known (see e.g., Jazwinski (1970)) that a sequential solution can be obtained according to

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{\int p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})d\mathbf{x}_k}, \tag{6a}$$

$$p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k}) = \int p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k})d\mathbf{x}_k. \tag{6b}$$

Here, it is also worth mentioning that since we have assumed additive noise in the model (1), we have explicit expressions for $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and $p(\mathbf{y}_k|\mathbf{x}_k)$ according to

$$p(\mathbf{x}_{k+1}|\mathbf{x}_k) = p_{\mathbf{w}_k}(\mathbf{x}_{k+1} - f(\mathbf{x}_k, \mathbf{u}_k)), \tag{7a}$$

$$p(\mathbf{y}_k|\mathbf{x}_k) = p_{\mathbf{e}_k}(\mathbf{y}_k - h(\mathbf{x}_k, \mathbf{u}_k)), \tag{7b}$$

where $p_{\mathbf{w}_k}(\cdot)$ and $p_{\mathbf{e}_k}(\cdot)$ denote the PDF's for the process and the measurement noise, respectively.

In the special case, where the equations in the model (1) are linear and the noise is Gaussian, the multidimensional integrals in (6) allows for an analytical solution, the Kalman filter Kalman (1960). For a derivation of this kind, see e.g., Schön (2006). However, the problem is that for the general nonlinear, non-Gaussian case that we are facing, there does not exist any closed form solution to (6). Hence, we are forced to make approximations of some kind. The most commonly used approximation is provided by the extended Kalman filter (EKF). The idea underlying the EKF is very simple, approximate the nonlinear model with a linear model subject to Gaussian noise and apply the Kalman filter to this approximation. For a solid account of the EKF we refer to Anderson and Moore (1979); Kailath et al. (2000). Lately the so called particle filter, introduced in Gordon et al. (1993), has become increasingly popular. This filter often provides a better solution, but it typically requires much more computational effort. For the present application the EKF provides an approximation that is good enough. For a more thorough account of the framework for nonlinear estimation briefly introduced above we refer to Schön (2006).

Before we end our brief overview on the sensor fusion problem it is important to stress that a successful sensor fusion framework will, besides the modeling and filtering parts mentioned above, rely on a certain surrounding infrastructure. This surrounding infrastructure deals with issues such as time synchronization between the various sensors, calibration, sensor-near signal processing, track handling, etc. This part of the framework should not be overlooked and a solid treatment of the provided infrastructure is accounted for in Bengtsson and Danielsson (2008) for the problem at hand. Despite this it is worth mentioning that the leading vehicles are incorporated into the estimation problem using rather standard techniques from target tracking, such as nearest neighbor data association and track counters in order to decide when to stop tracking a certain vehicle, etc. These are all important parts of the system we have implemented, but it falls outside the scope of this paper and since the techniques are rather standard we simply refer to the general treatments given in e.g., Blackman and Popoli (1999); Bar-Shalom et al. (2001).

3 Dynamic Models

As mentioned in the introduction our sensor fusion framework needs a state-space model describing the dynamics of the ego vehicle, the road and the leading vehicles. In this section we will derive the differential equations describing the motion of the ego vehicle (Section 3.2), the road (Section 3.3) and the leading vehicles (Section 3.4), also referred to as targets. Finally, in Section 3.5 we summarize these equations and form the process model of the state-space model. However, before we embark on deriving these equations we introduce the overall geometry and some necessary notation in Section 3.1.

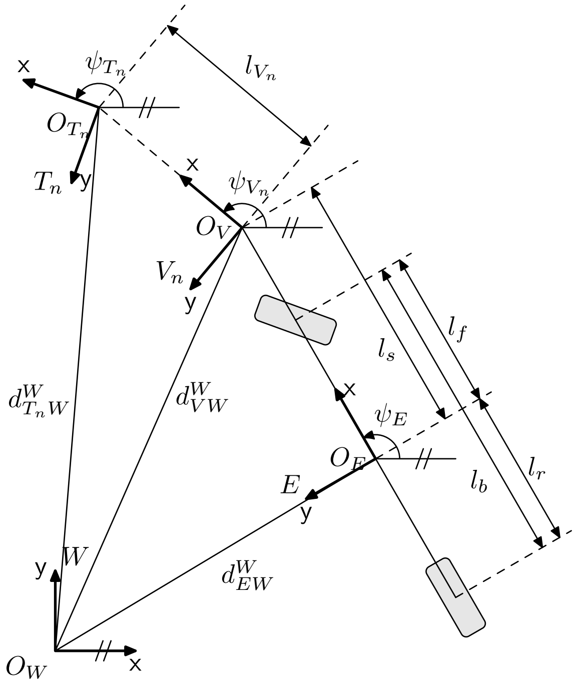


Figure 1: *Coordinate frames describing the ego vehicle, with center of gravity in O_E and the radar and camera sensors mounted in O_V. One leading vehicle is positioned in O_{T_n}.*

3.1 Geometry and Notation

The coordinate frames describing the ego vehicle and one leading vehicle are defined in Figure 1. The inertial world reference frame is denoted by W and its origin is O_W. The ego vehicle's coordinate frame E is located in the center of gravity (CoG). Furthermore, V_n is associated to the observed leading vehicle n, with O_V at the vision and radar sensor of the ego vehicle. Finally, T_n is also associated with the observed and tracked leading vehicle n, but its origin O_{T_n} is located at the leading vehicle. In this work we will use the planar coordinate transformation matrix

$$R^{WE} = \begin{bmatrix} \cos\psi_E & -\sin\psi_E \\ \sin\psi_E & \cos\psi_E \end{bmatrix} \tag{8}$$

to transform a vector, represented in E, into a vector, represented in W, where the yaw angle of the ego vehicle ψ_E is the angle of rotation from W to E. The geometric displacement vector d^W_{EW} is the direct straight line from O_W to O_E represented with respect to the frame W. Velocities are defined as the movement of a frame E relative to the inertial reference frame W, but typically resolved in the frame E, for example v^E_x is the velocity of the E frame in its x-direction. The same convention holds for the acceleration a^E_x. In order to simplify the notation we leave out E when referring to the ego vehicle's velocity and acceleration. This notation will be used when referring to the various coordinate frames. However, certain frequently used quantities will be renamed, in the interest of readability. The measurements are denoted using superscript m. Furthermore, the notation used for the rigid body dynamics is in accordance with Hahn (2002).

3.2 Ego Vehicle

We will only be concerned with the ego vehicle motion during normal driving situations and not at the adhesion limit. This implies that the single track model is sufficient for the present purposes. This model is also referred to as the bicycle model, see e.g., Mitschke and Wallentowitz (2004); Wong (2001) for a solid treatment. The geometry of the single track model with slip angles is shown in Figure 2. It is here worth to point out that the velocity vector of the ego vehicle is typically not in the same direction as the longitudinal axis of the ego vehicle. Instead the vehicle will move along a path at an angle β with the longitudinal direction of the vehicle. Hence, the angle β is defined as,

$$\tan\beta = \frac{\mathrm{v}_\mathrm{y}}{\mathrm{v}_\mathrm{x}}, \tag{9}$$

where v_x and v_y are the ego vehicle's longitudinal and lateral velocity components, respectively. This angle β is referred to as the float angle Robert Bosch GmbH (2004) or the vehicle body side slip angle Kiencke and Nielsen (2005).

The slip angle α_i is defined as the angle between the central axis of the wheel and the path along which the wheel moves. The phenomenon of side slip is mainly due to the lateral elasticity of the tire. For reasonably small slip angles, at maximum 3 deg, it is a good approximation to assume that the lateral friction force of the tire F_i is proportional to the slip angle,

$$F_i = C_{\alpha i}\alpha_i. \tag{10}$$

The parameter $C_{\alpha i}$ is called cornering stiffness and describes the cornering behavior of the tire. The load transfer to the front axle when braking or to the outer wheels when driving trough a curve influences the parameter value. A model considering these influences is given in Lundquist and Schön (2009).

Following this brief introduction to the ego vehicle geometry, we are now ready to give an expression describing the evolution of yaw angle ψ_E and the float angle β over time

$$\ddot{\psi}_E = \beta\frac{-C_{\alpha f}l_f\cos\delta_f + C_{\alpha r}l_r}{I_{zz}} - \dot{\psi}_E\frac{C_{\alpha f}l_f^2\cos\delta_f + C_{\alpha r}l_r^2}{I_{zz}\mathrm{v}_\mathrm{x}} + \frac{C_{\alpha f}l_f\tan\delta_f}{I_{zz}}, \tag{11a}$$

$$\dot{\beta} = -\beta\frac{C_{\alpha f}\cos\delta_f + C_{\alpha r} + \dot{\mathrm{v}}_\mathrm{x}m}{m\mathrm{v}_\mathrm{x}} - \dot{\psi}_E\left(1 + \frac{C_{\alpha f}l_f\cos\delta_f - C_{\alpha r}l_r}{\mathrm{v}_\mathrm{x}^2 m}\right) + \frac{C_{\alpha f}\sin\delta_f}{m\mathrm{v}_\mathrm{x}}, \tag{11b}$$

where m denotes the mass of the vehicle and I_{zz} denotes the moment of inertia of the vehicle about its vertical axis in the center of gravity. These single track

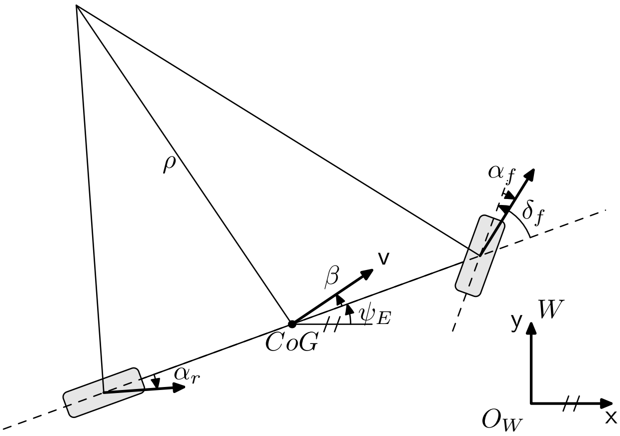


Figure 2: *In the single track model the wheels on each axle are modeled as single units. The velocity vector* v*, with the float angle β to the longitudinal axis of the vehicle, is attached at the center of gravity. Furthermore, the wheel slip angles are referred to as α_f and α_r. The front wheel angle is denoted by δ_f and the current radius is denoted by ρ.*

model equations are well-known in the literature, see e.g., Kiencke and Nielsen (2005).

3.3 Road Geometry

We start this section by defining the road variables and expressing a typical way to parameterize a road. The section is continued with a derivation of a new model for the road that makes use of the dynamic motion of the ego vehicle.

Background

The most essential component in describing the road geometry is the curvature c, which we will define as the curvature of the white lane marking to the left of the ego vehicle. An overall description of the road geometry is given in Figure 3. The heading angle ψ_R is defined as the tangent of the road at the level of the ego vehicle in the world reference frame W, see Figure 4. The angle δ_r is the angle between the tangent of the road curvature and the longitudinal axis of the ego vehicle. Note that this angle can be measured by sensors mounted on the ego vehicle. Furthermore, we define δ_R as

$$\delta_R \triangleq \delta_r - \beta, \tag{12}$$

i.e., the angle between the ego vehicles direction of motion (velocity vector) and the road curvature tangent.

The road curvature c is typically parameterized according to

$$c(\mathrm{x}_c) = c_0 + c_1 \mathrm{x}_c, \tag{13}$$

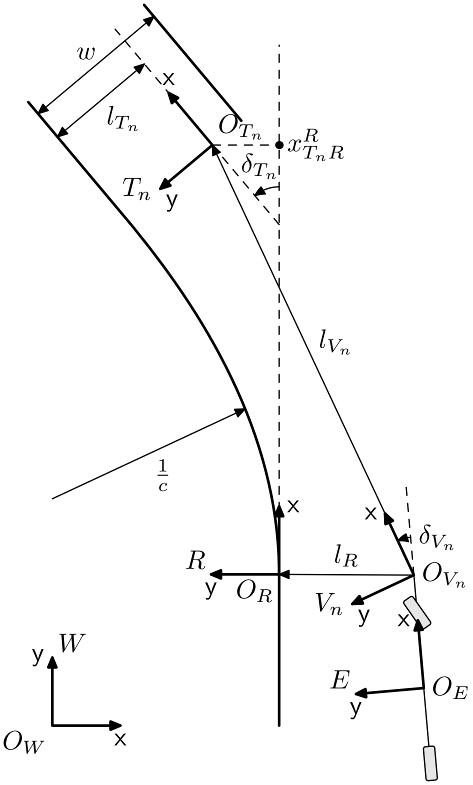


Figure 3: *Relations between one leading vehicle in O_{T_n}, the ego vehicle and the road. The distance between the ego vehicle's longitudinal* x*-axis and the white lane to its left is $l_R(t)$. The leading vehicle's distance to the lane marking is l_{T_n} and its heading angle in the road frame R is δ_{T_n}. The lane width is w.*

where x_c is the position along the road in a road aligned coordinate frame and $\mathrm{x}_c = 0$ at the vehicles center of gravity. Furthermore, c_0 describes the local curvature at the ego vehicle position and c_1 is the distance derivative (hence, the rate of change) of c_0. It is common to make use of a road aligned coordinate frame when deriving an estimator for the road geometry, a good overview of this approach is given in Eidehall (2007). There are several advantages using road aligned coordinate frames, particularly the motion models of the other vehicles on the same road can be greatly simplified. However, the flexibility of the motion models is reduced and basic dynamic relations such as Newton's and Euler's laws cannot be directly applied. Since we are using a single track model of the ego vehicle, we will make use of a Cartesian coordinate frame. A good polynomial approximation of the shape of the road curvature is given by

$$\mathrm{y}^E = l_R + \mathrm{x}^E \tan \delta_r + \frac{c_0}{2}(\mathrm{x}^E)^2 + \frac{c_1}{6}(\mathrm{x}^E)^3, \tag{14}$$

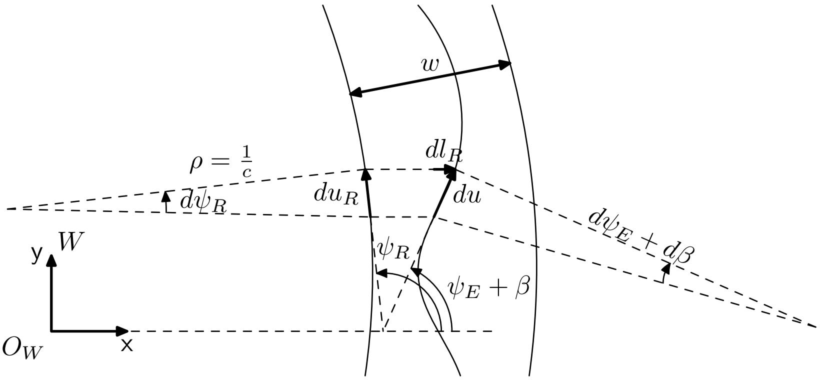


Figure 4: *Infinitesimal segments of the road curvature du_R and the driven path du are shown together with the angles $\delta_R = \psi_R - (\psi_E + \beta)$.*

where $l_R(t)$ is defined as the time dependent distance between the ego vehicle and the lane marking to the left, see e.g., Dickmanns and Mysliwetz (1992); Eidehall (2007).

The following dynamic model is often used for the road

$$\dot{c}_0 = \mathrm{v}c_1, \tag{15a}$$

$$\dot{c}_1 = 0, \tag{15b}$$

which can be interpreted as a velocity dependent integration. It is interesting to note that (15) reflects the way in which roads are commonly built Dickmanns and Mysliwetz (1992). However, we will now derive a new dynamic model for the road, that makes use of the road geometry introduced above.

A New Dynamic Road Model

Assume that du_R is an infinitesimal part of the road curvature or an arc of the road circle with the angle $d\psi_R$, see Figure 4. A segment of the road circle can be described as

$$du_R = \frac{1}{c_0} d\psi_R, \tag{16}$$

which after division with the infinitesimal change in time dt is given by

$$\frac{du_R}{dt} = \frac{1}{c_0}\frac{d\psi_R}{dt}. \tag{17}$$

Assuming that the left hand side can be reformulated according to

$$\frac{du_R}{dt} = \mathrm{v_x} \cos\left(\psi_R - \psi_E\right) \approx \mathrm{v_x}, \tag{18}$$

this yields

$$\mathrm{v_x} = \frac{1}{c_0}\dot{\psi}_R. \tag{19}$$

The angle ψ_R can be expressed as

$$\psi_R = \psi_E + \beta + \delta_R, \tag{20}$$

by rewriting (12). Re-ordering equation (19) and using the derivative of (20) to substitute $\dot{\psi}_R$ yields

$$\dot{\delta}_R = c_0 \mathsf{v}_\mathsf{x} - (\dot{\psi}_E + \dot{\beta}), \tag{21}$$

which by substituting $\dot{\beta}$ with (11b) according to

$$\begin{aligned}\dot{\delta}_R = c_0 \mathsf{v}_\mathsf{x} - \beta \frac{-C_{\alpha f} \cos \delta_f - C_{\alpha r} - \dot{\mathsf{v}}_\mathsf{x} m}{m \mathsf{v}_\mathsf{x}} \\ + \dot{\psi}_E \frac{C_{\alpha f} l_f \cos \delta_f - C_{\alpha r} l_r}{\mathsf{v}_\mathsf{x}^2 m} - \frac{C_{\alpha f} \sin \delta_f}{m \mathsf{v}_\mathsf{x}}\end{aligned} \tag{22}$$

results in a differential equation of the road angle δ_R. A similar relation has been used in Dickmanns and Mysliwetz (1992); Litkouhi et al. (1993).

We also need a differential equation for the road curvature, which can be found by differentiating (21) w.r.t. time,

$$\ddot{\delta}_R = \dot{c}_0 \mathsf{v}_\mathsf{x} + c_0 \dot{\mathsf{v}}_\mathsf{x} - \ddot{\psi}_E - \ddot{\beta}. \tag{23}$$

From the above equation we have

$$\dot{c}_0 = \frac{\ddot{\delta}_R + \ddot{\psi}_E + \ddot{\beta} - c_0 \dot{\mathsf{v}}_\mathsf{x}}{\mathsf{v}_\mathsf{x}}. \tag{24}$$

Let us assume that $\ddot{\delta}_R = 0$. Furthermore, differentiating $\dot{\beta}$, from (11b), w.r.t. time and inserting this together with $\ddot{\psi}_E$, given in (11a), into the above expression yields the differential equation

$$\begin{aligned}\dot{c}_0 = \frac{1}{(I_{zz} m^2 \mathsf{v}_\mathsf{x})^4} \Big(& C_{\alpha r}^2 (I_{zz} + l_r^2 m)(-\dot{\psi}_E l_r + \beta \mathsf{v}_\mathsf{x}) + C_{\alpha f}^2 (I_{zz} + l_f^2 m)(\dot{\psi}_E l_f + (\beta - \delta_f) \mathsf{v}_\mathsf{x}) \\ & + C_{\alpha r} I_{zz} m (-3 \dot{\psi}_E \dot{\mathsf{v}}_\mathsf{x} l_r + 3 \beta \dot{\mathsf{v}}_\mathsf{x} \mathsf{v}_\mathsf{x} + \dot{\psi}_E \mathsf{v}_\mathsf{x}^2) + \dot{\mathsf{v}}_\mathsf{x} I_{zz} m^2 \mathsf{v}_\mathsf{x} (2 \beta \dot{\mathsf{v}}_\mathsf{x} + \mathsf{v}_\mathsf{x} (\dot{\psi}_E - c_0 \mathsf{v}_\mathsf{x})) \\ & + C_{\alpha f} (C_{\alpha r} (I_{zz} + l_r(-l_f) m)(\dot{\psi}_E l_b - 2 \dot{\psi}_E l_r + 2 \beta \mathsf{v}_\mathsf{x} - \delta_f \mathsf{v}_\mathsf{x}) \\ & + I_{zz} m (3 \dot{\psi}_E \dot{\mathsf{v}}_\mathsf{x} l_f + (3\beta - 2\delta_f) \dot{\mathsf{v}}_\mathsf{x} \mathsf{v}_\mathsf{x} + (\dot{\delta}_f + \dot{\psi}_E) \mathsf{v}_\mathsf{x}^2)) \Big)\end{aligned} \tag{25}$$

for the road curvature.

In this model c_0 is defined at the ego vehicle and thus describes the currently driven curvature, whereas for the curvature described by the state-space model (15) and by the polynomial (13) it is not entirely obvious where c_0 is defined.

Finally, we need a differential equation describing how the distance $l_R(t)$ between the ego vehicle and the lane markings changes over time. Assume again an infinitesimal arc du of the circumference describing the ego vehicle's curvature. By contemplating Figure 4 we have

$$dl_R = du \sin \delta_R, \tag{26}$$

where δ_R is the angle between the ego vehicle's velocity vector and the road. Dividing this equation with an infinitesimal change in time dt and using $du/dt = v$ yield the differential equation

$$\dot{l}_R = \mathrm{v_x} \sin(\delta_R + \beta), \tag{27}$$

which concludes the derivation of the road geometry model.

3.4 Leading Vehicles

The leading vehicles are also referred to as targets T_n. The coordinate frame T_n moving with target n has its origin located in O_{T_n}, as we previously saw in Figure 3. It is assumed that the leading vehicles are driving on the road, quite possibly in a different lane. More specifically, it is assumed that they are following the road curvature and thus that their heading is in the same direction as the tangent of the road.

For each target T_n, there exists a coordinate frame V_n, with its origin O_V at the position of the sensor. Hence, the origin is the same for all targets, but the coordinate frames have different heading angles ψ_{V_n}. This angle, as well as the distance l_{V_n}, depend on the targets position in space. From Figure 3 it is obvious that,

$$d^W_{EW} + d^W_{V_nE} + d^W_{T_nV} - d^W_{T_nW} = 0, \tag{28}$$

or more explicitly,

$$\mathrm{x}^W_{EW} + l_s \cos\psi_E + l_{V_n} \cos\psi_{V_n} - \mathrm{x}^W_{T_nW} = 0, \tag{29a}$$

$$\mathrm{y}^W_{EW} + l_s \sin\psi_E + l_{V_n} \sin\psi_{V_n} - \mathrm{y}^W_{T_nW} = 0. \tag{29b}$$

Let us now define the relative angle to the leading vehicle as

$$\delta_{V_n} \triangleq \psi_{V_n} - \psi_E. \tag{30}$$

It is worth noticing that this angle can be measured by a sensor mounted on the vehicle.

The target T_n is assumed to have zero lateral velocity in the V_n frame, i.e., $\dot{y}^{V_n} = 0$, since it is always fixed to the x^{V_n}-axis. If we transform this relation to the world frame W, using the geometry of Figure 1 we have

$$R^{VW} \cdot \dot{d}^W_{T_nW} = \begin{bmatrix} \cdot \\ 0 \end{bmatrix}, \tag{31}$$

where the top equation of the vector equality is non-descriptive and the bottom equation can be rewritten as

$$-\dot{\mathrm{x}}^W_{T_nW} \sin\psi_{V_n} + \dot{\mathrm{y}}^W_{T_nW} \cos\psi_{V_n} = 0. \tag{32}$$

The velocity vector of the ego vehicles is applied in the center of gravity O_E. The derivative of (29) is used together with the velocity components of the ego vehicle and (32) to get an expression for the derivative of the relative angle to the leading vehicle w.r.t. time according to

$$(\dot{\delta}_{V_n} + \dot{\psi}_E)l_{V_n} + \dot{\psi}_E l_s \cos\delta_{V_n} + \mathrm{v}_\mathrm{x}\sin(\beta - \delta_{V_n}) = 0. \tag{33}$$

This equation is rewritten, forming the differential equation

$$\dot{\delta}_{V_n} = -\frac{\dot{\psi}_E l_s \cos\delta_{V_n} + \mathrm{v}_\mathrm{x}\sin(\beta - \delta_{V_n})}{l_{V_n}} - \dot{\psi}_E \tag{34}$$

of the relative angle $\dot{\delta}_{V_n}$ to the leading vehicles.

3.5 Summarizing the Dynamic Model

The state-space models derived in the previous sections are nonlinear and they are given in continuous time. Hence, in order to make use of these equations in the EKF we will first linearize them and then make use of (4) in order to obtain a state-space model in discrete time according to (1). This is a rather standard procedure, see e.g., Gustafsson (2000); Rugh (1996). At each time step k, the nonlinear state-space model is linearized by evaluating the Jacobian (i.e., the partial derivatives) of the $f(\mathbf{x}_k, \mathbf{u}_k)$-matrix introduced in (4) at the current estimate $\hat{\mathbf{x}}_{k|k}$. It is worth noting that this Jacobian is straightforwardly computed off-line using symbolic or numerical software, such as MATHEMATICA. Hence, we will not go through the details here. However, for future reference we will briefly summarize the continuous-time dynamic model here.

In the final state-space model the three parts (ego vehicle, road and leading vehicles) of the dynamic model are augmented, resulting in a state vector of dimension $6 + 4 \cdot$ (Number of leading vehicles). Hence, the size of the state vector varies with time, depending on the number of leading vehicles that are tracked at a specific instance of time.

The ego vehicle model is described by the following states,

$$\mathbf{x}_\mathrm{E} = \begin{bmatrix} \dot{\psi}_E & \beta & l_R \end{bmatrix}^T, \tag{35}$$

i.e., the yaw rate, the float angle and the distance to the left lane marking. The front wheel angle δ_f, which is calculated from the measured steering wheel angle, and the ego vehicle longitudinal velocity v_x and acceleration $\dot{\mathrm{v}}_x$ are modeled as input signals,

$$\mathbf{u}_k = \begin{bmatrix} \delta_f & \mathrm{v}_\mathrm{x} & \dot{\mathrm{v}}_x \end{bmatrix}^T. \tag{36}$$

The nonlinear state-space model $\dot{\mathbf{x}}_\mathrm{E} = g_\mathrm{E}(\mathbf{x}, \mathbf{u})$ is given by

$$g_\mathrm{E}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \beta\frac{-C_{\alpha f} l_f \cos\delta_f + C_{\alpha r} l_r}{I_{zz}} - \dot{\psi}_E \frac{C_{\alpha f} l_f^2 \cos\delta_f + C_{\alpha r} l_r^2}{I_{zz}\mathrm{v}_\mathrm{x}} + \frac{C_{\alpha f} l_f \tan\delta_f}{I_{zz}} \\ -\beta\frac{C_{\alpha f}\cos\delta_f + C_{\alpha r} + \dot{\mathrm{v}}_\mathrm{x} m}{m\mathrm{v}_\mathrm{x}} - \dot{\psi}_E\left(1 + \frac{C_{\alpha f} l_f \cos\delta_f - C_{\alpha r} l_r}{\mathrm{v}_\mathrm{x}^2 m}\right) + \frac{C_{\alpha f}\sin\delta_f}{m\mathrm{v}_\mathrm{x}} \\ \mathrm{v}_\mathrm{x}\sin(\delta_R + \beta) \end{bmatrix}. \tag{37}$$

The corresponding differential equations were previously given in (11a), (11b) and (27), respectively.

The states describing the road $\mathbf{x}_\mathrm{R}$ are the road curvature c_0 at the ego vehicle position, the angle δ_R between the ego vehicles direction of motion and the road curvature tangent and the width of the lane w, i.e.,

$$\mathbf{x}_\mathrm{R} = \begin{bmatrix} c_0 & \delta_R & w \end{bmatrix}^T. \tag{38}$$

The differential equations for c_0 and δ_R were given in (25) and (22), respectively. When it comes to the width of the current lane w, we have

$$\dot{w} = 0, \tag{39}$$

motivated by the fact that w does not change as fast as the other variables, i.e., the nonlinear state-space model $\dot{\mathbf{x}}_\mathrm{R} = g_\mathrm{R}(\mathbf{x}, \mathbf{u})$ is given by

$$g_\mathrm{R}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{c}_0 \\ c_0 \mathrm{v}_\mathrm{x} + \beta \frac{C_{\alpha f} \cos \delta_f + C_{\alpha r} + \dot{\mathrm{v}}_\mathrm{x} m}{m \mathrm{v}_\mathrm{x}} + \dot{\psi} \frac{C_{\alpha f} l_f \cos \delta_f - C_{\alpha r} l_r}{\mathrm{v}_\mathrm{x}^2 m} - \frac{C_{\alpha f} \sin \delta_f}{m \mathrm{v}_\mathrm{x}} \\ 0 \end{bmatrix}. \tag{40}$$

A target is described by the following states, azimuth angle δ_{V_n}, lateral position l_{T_n} of the target, distance between the target and the ego vehicle l_{V_n} and relative velocity between the target and the ego vehicle $\dot{l}_{V_n}$. Hence, the state vector is given by

$$\mathbf{x}_\mathrm{T} = \begin{bmatrix} \delta_{V_n} & l_{T_n} & \dot{l}_{V_n} & l_{V_n} \end{bmatrix}^T. \tag{41}$$

The derivative of the azimuth angle was given in (34). It is assumed that the leading vehicle's lateral velocity is small, implying that $\dot{l}_{T_n} = 0$ is a good assumption (compare with Figure 3). Furthermore, it can be assumed that the leading vehicle accelerates similar to the ego vehicle, thus $\ddot{l}_{V_n} = 0$ (compare with e.g., Eidehall (2007)). The state-space model $\dot{\mathbf{x}}_\mathrm{T} = g_\mathrm{T}(\mathbf{x}, \mathbf{u})$ of a leading vehicle (target) is

$$g_\mathrm{T}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} -\frac{\dot{\psi}_E l_s \cos \delta_{V_n} + \mathrm{v}_\mathrm{x} \sin(\beta - \delta_{V_n})}{l_{V_n}} - \dot{\psi}_E \\ 0 \\ 0 \\ \dot{l}_{V_n} \end{bmatrix}. \tag{42}$$

Note that the dynamic models given in this section are nonlinear in $\mathbf{u}$.

4 Measurement Model

The measurement model (1b) describes how the measurements $\mathbf{y}_k$ relates to the state variables $\mathbf{x}_k$. In other words, it describes how the measurements enter the estimator. We will make use of superscript m to denote measurements. Let us start by introducing the measurements relating directly to the ego vehicle motion, by defining

$$\mathbf{y}^1 = \begin{bmatrix} \dot{\psi}_E^m & \mathrm{a}_\mathrm{y}^m \end{bmatrix}^T, \tag{43}$$

where $\dot{\psi}_E^m$ and a_y^m are the measured yaw rate and the measured lateral acceleration, respectively. They are both measured with the ego vehicle's inertial sensor in the center of gravity (CoG). The ego vehicle lateral acceleration in the CoG is

$$\mathsf{a}_\mathsf{y} = \mathsf{v}_\mathsf{x}(\dot{\psi}_E + \dot{\beta}) + \dot{\mathsf{v}}_\mathsf{x}\beta. \tag{44}$$

By replacing $\dot{\beta}$ with the expression given in (11b) and at the same time assuming that $\dot{\mathsf{v}}_\mathsf{x}\beta \approx 0$ we obtain

$$\begin{aligned}\mathsf{a}_\mathsf{y} &= \mathsf{v}_\mathsf{x}(\dot{\psi}_E + \dot{\beta}) \\ &= -\beta\frac{C_{\alpha f}\cos\delta_f + C_{\alpha r} + m\dot{\mathsf{v}}_\mathsf{x}}{m} + \dot{\psi}_E\frac{-C_{\alpha f}l_f\cos\delta_f + C_{\alpha r}l_r}{m\mathsf{v}_\mathsf{x}} + \frac{C_{\alpha f}}{m}\sin\delta_f. \end{aligned}\tag{45}$$

From this it is clear that the measurement of the lateral acceleration contains information about the ego vehicle states. Hence, the measurement equation corresponding to (43) is given by

$$h^1 = \begin{bmatrix} \dot{\psi}_E \\ -\beta\frac{C_{\alpha f}\cos\delta_f + C_{\alpha r} + m\dot{\mathsf{v}}_\mathsf{x}}{m} + \dot{\psi}_E\frac{-C_{\alpha f}l_f\cos\delta_f + C_{\alpha r}l_r}{m\mathsf{v}_\mathsf{x}} + \frac{C_{\alpha f}}{m}\sin\delta_f \end{bmatrix}. \tag{46}$$

The vision system provides measurements of the road geometry and the ego vehicle position on the road according to

$$\mathbf{y}^2 = \begin{bmatrix} c_0^m & \delta_r^m & w^m & l_R^m \end{bmatrix}^T \tag{47}$$

and the corresponding measurement equations are given by

$$h^2 = \begin{bmatrix} c_0 & (\delta_R + \beta) & w & l_R \end{bmatrix}^T. \tag{48}$$

An obvious choice would have been to use the state δ_r, instead of the sum $\delta_R + \beta$, however, we have chosen to split these since we are interested in estimating both of these quantities.

In order to include measurements of a leading vehicle we require that it is detected both by the radar and the vision system. The range l_{V_n} and the range rate $\dot{l}_{V_n}$ are measured by the radar. The azimuth angle is also measured by the radar, but not used directly in this framework. Instead, the accuracy of the angle estimate is improved by using the camera information. We will not describe these details here, since it falls outside the scope of this work. The corresponding measurement vector is

$$\mathbf{y}^3 = \begin{bmatrix} \delta_{V_n}^m & \dot{l}_{V_n}^m & l_{V_n}^m \end{bmatrix}^T. \tag{49}$$

Since these are state variables, the measurement equation is obviously

$$h^3 = \begin{bmatrix} \delta_{V_n} & \dot{l}_{V_n} & l_{V_n} \end{bmatrix}^T. \tag{50}$$

The fact that the motion of the leading vehicles reveals information about the road geometry allows us to make use of their motion in order to improve the road geometry estimate. This will be accomplished by introducing a nontrivial artificial measurement equation according to

$$h^4 = l_R + (\delta_R + \beta) l_{V_n} \cos\delta_{V_n} + \frac{c_0}{2}(l_{V_n}\cos\delta_{V_n})^2 + \frac{l_{T_n}}{\cos\delta_{T_n}}, \tag{51}$$

which is derived from Figure 3 and describes the predicted lateral distance of a leading vehicle in the ego vehicles coordinate frame E. In order to model the road curvature we introduce the road coordinate frame R, with its origin O_R on the white lane marking to the left of the ego vehicle. This implies that the frame R is moving with the frame E of the ego vehicle. The angle $\delta_{T_n} \triangleq \psi_{T_n} - \psi_R$ is derived by considering the road's slope at the position of the leading vehicle, i.e.,

$$\delta_{T_n} = \arctan\frac{d\mathrm{y}^R}{d\mathrm{x}^R} = \arctan c_0 \mathrm{x}^R, \tag{52}$$

where $\mathrm{x}^R = \mathrm{x}^R_{T_nR}$, see Figure 3. The Cartesian x-coordinate of the leading vehicle T_n in the R-frame is

$$\mathrm{x}^R_{T_nR} = \mathrm{x}^E_{T_nE} - l_s \approx l_{V_n}\frac{\cos\delta_{V_n}}{\cos\delta_r}. \tag{53}$$

The sensors only provide range $l^m_{V_n}$ and azimuth angle $\delta^m_{V_n}$. Hence, the corresponding quasi-measurement is

$$\mathbf{y}^4 = l^m_{V_n}\sin(\delta^m_{V_n}), \tag{54}$$

describing the measured lateral distance to a leading vehicle in the ego vehicle's coordinate frame. This might seem a bit ad hoc at first. However, the validity of the approach has recently been justified in the literature, see e.g., Teixeira et al. (2007).

5 Experiments and Results

The experiments presented in this section are based on measurements acquired on public roads in Sweden during normal traffic conditions. The test vehicle is a Volvo S80 equipped with a forward looking 77 GHz mechanically scanning FMCW radar and a forward looking vision sensor (camera), measuring the distances and angles to the targets. The image sensor includes object and lane detection and provides for example the lane curvature. Information about the ego vehicle motion, such as the steering wheel angle, yaw rate, etc. were acquired directly from the CAN bus.

Before stating the main results in this section we outline how to estimate the parameters of the ego vehicle and how the filter is tuned. Subsequently we state the results of the ego vehicle validation. We compare our road curvature estimates with two other sensor fusion approaches as well as one road model.

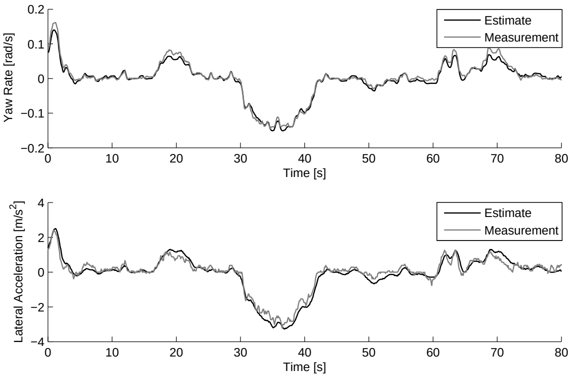


Figure 5: *Comparing the simulated result of the nonlinear state-space model (black) with measured data (gray) of a validation data set. The upper plot shows the yaw rate and the lower shows the lateral acceleration.*

5.1 Parameter Estimation and Filter Tuning

Most of the ego vehicle's parameters, such as the dimensions, the mass and the moment of inertia were provided by the vehicle manufacturer. Since the cornering stiffness is a parameter which describes the properties between road and tire it has to be estimated for the given set of measurements. An on-line method to estimate the cornering stiffness parameter using recursive least square is presented in Lundquist and Schön (2009). However, in the present work an exhaustive search was accomplished off-line using a batch of measurements to estimate $C_{\alpha f}$ and $C_{\alpha r}$. A state-space model with the differential equations given in (11a) and (11b) and with the yaw rate $\dot{\psi}_E$ and the float angle β in the state vector was used for this purpose. Furthermore, the front wheel angle δ_f and the ego vehicle longitudinal velocity v_x were modeled as input signals. The measurements were provided by the yaw rate $\dot{\psi}_E^m$ and the lateral acceleration a_y^m. The corresponding measurement equation was given in (46). The data used to identifying the cornering stiffness parameters was split into two parts, one estimation part and one validation part. This facilitates cross-validation, where the parameters are estimated using the estimation data and the quality of the estimates can then be assessed using the validation data Ljung (1999).

The approach is further described in Lundquist and Schön (2008b). The resulting state-space model with the estimated parameters was validated using the validation data and the result is given in Figure 5.

The process and measurement noise covariances are the design parameters in the extended Kalman filter (EKF). It is assumed that the covariances are diagonal and that there are no cross correlations between the measurement noise and the process noise. The present filter has ten states and ten measurement signals, which implies that 20 parameters have to be tuned. The tuning was started using physical intuition of the error in the process equations and the measurement signals. In a second step, the covariance parameters were tuned simply by trying to minimize the root mean square error (RMSE) of the estimated $\hat{c}_0$ and the reference curvature c_0. The estimated curvature was obtained by running the filter using the estimation data set. The calculation of the reference value is described in Eidehall and Gustafsson (2006). The chosen design parameters were validated on a different data set and the results are discussed in the subsequent sections.

5.2 Validation Using Ego Vehicle Signals

The state variables of the ego vehicle are according to (35), the yaw rate, the float angle and the distance to the left lane marking. The estimated and the measured yaw rate signals are, as expected, very similar. As described in Section 5.1, the parameters of the vehicle model were optimized with respect to the yaw rate, hence it is no surprise that the fusion method decreases the residual further. A measurement sequence acquired on a rural road is shown in Figure 6a. Note that the same measurement sequence is used in Figures 5 to 7, which will make it easier to compare the estimated states.

The float angle β is estimated, but there is no reference or measurement signal to compare it to. An example is shown in Figure 6b. For velocities above 30 – 40 km/h, the float angle appears more or less like the mirror image of the yaw rate, and by comparing with Figure 6a, we can conclude that the sequence is consistent.

The measurement signal of the distance to the left white lane marking l_R^m is produced by the vision system OLR (optical lane recognition). Bad lane markings or certain weather conditions can cause errors in the measurement signal. The estimated state l_R of the fusion approach is very similar to the pure OLR signal.

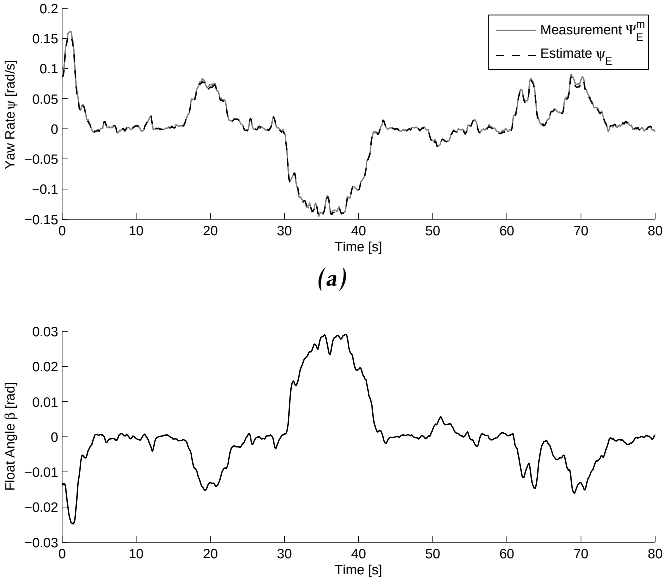
5.3 Road Curvature Estimation

An essential idea with the sensor fusion approach introduced in this paper is to make use of the single track ego vehicle model in order to produce better estimates of the road curvature. In this section we will compare this approach to approaches based on other models of the ego vehicle and the road geometry.

Fusion 1 is the sensor fusion approach shown in this paper.

Fusion 2 is a similar approach, thoroughly described in Eidehall (2007). An important difference to fusion 1 is that the ego vehicle is modeled with a constant velocity model, which is less complex. The float angle β is not estimated. Furthermore, the road is modeled according to (15) and a road aligned coordinate frame is used. This method is similar to the approaches used in e.g., Zomotor and Franke (1997); Gern et al. (2000, 2001).

Fusion 3 comprehends the ego vehicle model of fusion 1 and the road model of fusion 2, i.e., substituting (25) by (15) and introducing the seventh state c_1. Furthermore, a Cartesian coordinate frame is used. This method, but



(a)

(b)

Figure 6: *A comparison between the ego vehicle's measured (gray) and estimated yaw rate (black dashed) using the sensor fusion approach in this paper is shown in (a). The estimated float angle β for the same data sequence is shown in (b).*

without considering the leading vehicles is similar to the ones described in e.g., Dickmanns and Mysliwetz (1992) and Behringer (1997).

Model is the ego vehicle and road state-space model given in this paper, described by the motion models (37) and (40) and the measurement models (46) and (48), without the extended Kalman filter.

The curvature estimate $\hat{c}_0$ from the sensor fusion approaches, the model and the raw measurement from the optical lane recognition are compared to a reference value. The reference value is computed off-line using a geometric method described in Eidehall and Gustafsson (2006), which applies a least square curve fitting to a sliding window. The entire data set i.e., also future values of the ego vehicle movement, is used to derive the reference value. The accuracy of the method was validated on a test track, where the ground truth is well defined, and the results are good as reported in Eidehall and Gustafsson (2006).

A typical result of a comparison is shown in Figure 7. The data stems from a rural road, which explains the curvature values. It can be seen that the estimates from the sensor fusion approaches give better results than using the OLR alone, as was expected. The OLR estimate is rather noisy compared to the fused estimates. This is not surprising, since the raw OLR has less information. A camera view from the curve at time 32 s is shown in Figure 8a.

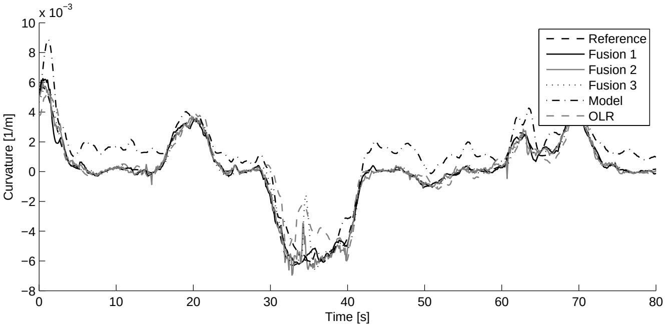
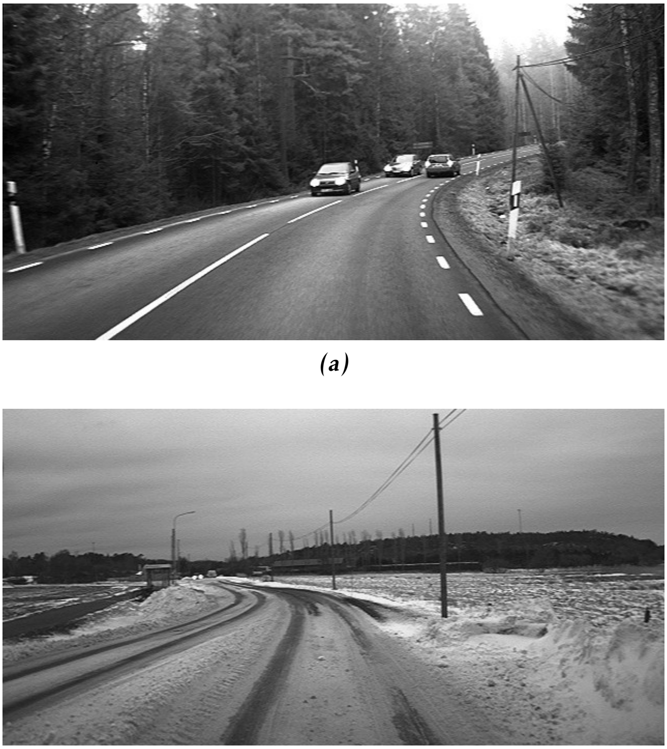


Figure 7: *Results from the three fusion approaches (fusion 1 solid black line, fusion 2 gray line and fusion 3 dotted line) and the* OLR *(dashed gray line), showing the curvature estimate $\hat{c}_0$. As can be seen, the curvature estimate can be improved by taking the other vehicles (gray line) and the ego vehicle's driven curvature in to account (solid black line). The model (dash-dotted) is estimating the derivative of the curvature and the absolute position is not measured, which leads to the illustrated bias. The dashed line is the reference curvature.*

The curvature estimate from the state-space model described in this paper is denoted by model and is shown as a dash-dotted black line. The absolute position is not measured, which leads to a clearly visible bias in the estimate of c_0. The bias is transparent in Figure 7, but it also leads to a large RMSE value in Table 1. Fusion 3 also delivers a decent result, but it is interesting to notice that the estimate seems to follow the incorrect OLR at time 35 s. The same behavior holds for fusion 2 in Figure 7.

To get a more aggregate view of the performance, we provide the root mean square error (RMSE) for longer measurement sequences in Table 1. The fusion approaches improve the road curvature estimate by making use of the information about the leading vehicles, that is available from the radar and the vision systems. However, since we are interested in the curvature estimate also when there are no leading vehicles in front of the ego vehicle, this case will be studied as well. It is straightforward to study this case, it is just a matter of not providing the measurements of the leading vehicles to the algorithms. The RMSE values found without information about the leading vehicles are given in the columns marked *no* in Table 1.

These results should ideally be compared to data where information about the leading vehicles is considered, but during the 78 min drive there were not always another car in front of us. Only for about 50 % of the time there existed other vehicles, which we could track. Hence, for the sake of comparability we give the RMSE values for those sequences where at least one leading vehicle was tracked, bearing in mind that these are based on only about 50 % of the data. The corresponding columns in Table 1 are marked *only*. Finally, we also give the RMSE



(a)

(b)

Figure 8: *Two different camera views are shown. In (a) the lane markings are excellent and the leading vehicles are close and clearly visible. This is the traffic situation at* 32 s *in the Figures 5 to 7. Although the circumstances seem perfect, the* OLR*, Fusion 2 and 3 have problems estimating the curvature, as seen in Figure 7. The traffic situation shown in (b) is more demanding, mainly due to the weather conditions and large distance to the leading vehicle.*

values for the complete data, where other vehicles were considered whenever *possible*.

It is interesting to see that the advantage of fusion 1, which uses a more accurate ego vehicle and road model, in comparison to fusion 2 is particularly noticeable when driving alone on a rural road, the RMSE for fusion 1 is then 1.18, whereas the RMSE for fusion 2 is 2.91. The reason for this is first of all that we are driving on a rather curvy road which implies that any additional information will help improving the curvature estimate. Here, the additional information is the improved ego vehicle and road models used in fusion 1. Furthermore, the fact that there are no leading vehicles that could aid the fusion algorithm when driving alone creates a greater disadvantage for fusion 2, since it is its main additional

Table 1: *Comparison of the root mean square error (RMSE) of the road curvature c_0 in $[1/m]$ for the three fusion approaches and the pure measurement signal OLR for two longer measurement sequences acquired on public roads in Sweden. Three cases were considered, using* only *those measurements where a leading vehicle could be tracked, using the knowledge of the leading vehicles position whenever* possible *or* not *at all and thereby simulating the lonely driver. Note that all RMSE values should be multiplied by 10^{-3}.*

	Highway			Rural road		
Time	44 min			34 min		
OLR $[10^{-3}/m]$	0.385			3.60		
Model $[10^{-3}/m]$	0.356			2.10		
Leading vehicles used?	only	possible	no	only	possible	no
Fusion 1 $[10^{-3}/m]$	0.176	0.184	0.189	1.48	1.13	1.18
Fusion 2 $[10^{-3}/m]$	0.231	0.228	0.230	1.53	2.84	2.91
Fusion 3 $[10^{-3}/m]$	0.203	0.210	0.205	1.32	2.01	1.94

information. Fusion 3, which uses the single track vehicle model of fusion 1, but the road model of fusion 2, seems to position itself between those two.

Comparing the rural road results based only on those measurements where other vehicles were tracked, we see an interesting pattern. The curvature estimate of fusion 2 and fusion 3 is improved by the additional information, but the estimate of fusion 1 is declined. The error values of the three fusion approaches are also in the same range. The explanation of this behavior can be found by analyzing the measurement sequences. If the leading vehicle is close-by, as for example in Figure 8a, it helps improving the results. However, if the leading vehicle is more distant, the curvature at this position might not be the same as it is at the ego vehicle's position, which leads to a degraded result. In Lundquist and Schön (2008a) the authors presented preliminary results based on much shorter measurement sequences, where the leading vehicles were more close-by and the estimate of fusion 1 was improved by the existence of leading vehicles. The problem could be solved by letting the measurement noise $\mathbf{e}$ of the measurement equation (51) depend on the distance to the leading vehicle.

The highway is rather straight and as expected not much accuracy could be gained in using an improved dynamic vehicle model. It is worth noticing that the OLR's rural road RMSE value is about 10 times higher than the highway value, but the model's RMSE increases only about six times when comparing the rural road values with the highway. Comparing the RMSE values in the columns marked *possible*; the RMSE for fusion 1 also increases about six times, but that of fusion 2 increases as much as twelve times when comparing the highway measurements with the rural road.

A common problem with these road estimation methods is that it is hard to distinguish between the case when the leading vehicle is entering a curve and the case when the leading vehicle is performing a lane change. With the approach in this paper the information about the ego vehicle motion, the OLR and the leading vehicles is weighted together in order to form an estimate of the road curvature. The fusion approach in this paper produces an estimate of the lateral position l_{T_n} of the leading vehicle which seems reasonable. The results are thoroughly described in Lundquist and Schön (2008a).

6 Conclusions

In this contribution we have derived a method for joint ego-motion and road geometry estimation. The presented sensor fusion approach combines the information from sensors present in modern premium cars, such as radar, camera and IMU, with a dynamic model. This model, which consists of a new dynamic motion model of the road, is the core of this contribution. The road geometry is estimated by considering the information from the optical lane recognition of the camera, the position of the leading vehicles, obtained by the radar and the camera, and by making use of a dynamic ego vehicle motion model, which takes IMU-data and the steering wheel angle as input. If one of these three parts fails, for example there might not be any leading vehicles or the lane markings are bad, as in Figure 8b, then the sensor fusion framework will still deliver an estimate.

The presented sensor fusion framework has been evaluated together with two other fusion approaches on real and relevant data from both highway and rural roads in Sweden. The data consists of 78 min driving on various road conditions, also including snow-covered pavement. The approach presented in this paper obtained the best results in all situations, when compared to the other approaches, but it is most prominent when driving alone on a rural road. If there are no leading vehicles that can be used, the improved road and ego vehicle models still supports the road geometry estimation and delivers a more accurate result.

Bibliography

B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Information and system science series. Prentice Hall, Englewood Cliffs, NJ, USA, 1979.

Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, New York, NY, USA, 2001.

R. Behringer. *Visuelle Erkennung und Interpretation des Fahrspurverlaufes durch Rechnersehen für ein autonomes Straßenfahrzeug*, volume 310 of *Fortschrittsberichte VDI, Reihe 12*. VDI Verlag, Düsseldorf, Germany, 1997. Also as: PhD Thesis, Universität der Bundeswehr, 1996.

F. Bengtsson and L. Danielsson. Designing a real time sensor data fusion system with application to automotive safety. In *Proceedings of the World Congress on Intelligent Transportation Systems and Services*, New York, USA, November 2008.

S. S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, Norwood, MA, USA, 1999.

E. D. Dickmanns. *Dynamic Vision for Perception and Control of Motion*. Springer, London, UK, 2007.

E. D. Dickmanns and B. D. Mysliwetz. Recursive 3-D road and relative ego-state recognition. *IEEE Transactions on pattern analysis and machine intelligence*, 14(2):199–213, February 1992.

E. D. Dickmanns and A. Zapp. A curvature-based scheme for improving road vehicle guidance by computer vision. In *Proceedings of the SPIE Conference on Mobile Robots*, volume 727, pages 161–198, Cambridge, MA, USA, 1986.

A. Eidehall. *Tracking and threat assessment for automotive collision avoidance*. PhD thesis No 1066, Linköping Studies in Science and Technology, Linköping, Sweden, January 2007.

A. Eidehall and F. Gustafsson. Obtaining reference road geometry parameters from recorded sensor data. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 256–260, Tokyo, Japan, June 2006.

A. Eidehall, J. Pohl, and F. Gustafsson. Joint road geometry estimation and vehicle tracking. *Control Engineering Practice*, 15(12):1484–1494, December 2007.

A. Gern, U. Franke, and P. Levi. Advanced lane recognition - fusing vision and radar. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 45–51, Dearborn, MI, USA, October 2000.

A. Gern, U. Franke, and P. Levi. Robust vehicle tracking fusing radar and vision. In *Proceedings of the international conference of multisensor fusion and integration for intelligent systems*, pages 323–328, Baden-Baden, Germany, August 2001.

N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings on Radar and Signal Processing*, 140(5):107–113, April 1993.

T. Gumpp, D. Nienhuser, R. Liebig, and J.M. Zollner. Recognition and tracking of temporary lanes in motorway construction sites. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 305–310, Xi'an, China, June 2009.

F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons, New York, USA, 2000.

H. Hahn. *Rigid body dynamics of mechanisms. 1, Theoretical basis*, volume 1. Springer, Berlin, Germany, 2002.

U. Hofmann, A. Rieder, and E.D. Dickmanns. Ems-vision: application to hybrid adaptive cruise control. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 468–473, Dearborn, MI, USA, 2000.

U. Hofmann, A. Rieder, and E.D. Dickmanns. Radar and vision data fusion for hybrid adaptive cruise control on highways. *Machine Vision and Applications*, 14(1):42–49, 2003.

A. H. Jazwinski. *Stochastic processes and filtering theory*. Mathematics in science and engineering. Academic Press, New York, USA, 1970.

T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Information and System Sciences Series. Prentice Hall, Upper Saddle River, NJ, USA, 2000.

K. Kaliyaperumal, S. Lakshmanan, and K. Kluge. An algorithm for detecting roads and obstacles in radar images. *IEEE Transactions on Vehicular Technology*, 50(1):170–182, January 2001.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82:35–45, 1960.

U. Kiencke and L. Nielsen. *Automotive Control Systems*. Springer, Berlin, Heidelberg, Germany, 2 edition, 2005.

A. Kirchner and C. Ameling. Integrated obstacle and road tracking using a laser scanner. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 675–681, Dearborn, MI, USA, October 2000.

A. Kirchner and T. Heinrich. Model based detection of road boundaries with a laser scanner. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 93–98, Stuttgart, Germany, October 1998.

S. Lakshmanan, K. Kaliyaperumal, and K. Kluge. Lexluther: an algorithm for detecting roads and obstacles in radar images. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 415–420, Boston, MA, USA, November 1997.

B.B. Litkouhi, A.Y. Lee, and D.B. Craig. Estimator and controller design for lanetrak, a vision-based automatic vehicle steering system. In *Proceedings of the*

IEEE Conference on Decision and Control, volume 2, pages 1868–1873, San Antonio, Texas, December 1993.

L. Ljung. *System identification, Theory for the user*. System sciences series. Prentice Hall, Upper Saddle River, NJ, USA, 2 edition, 1999.

H. Loose, U. Franke, and C. Stiller. Kalman particle filter for lane recognition on rural roads. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 60–65, Xi'an, China, June 2009.

C. Lundquist and T. B. Schön. Road geometry estimation and vehicle tracking using a single track model. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 144–149, Eindhoven, The Netherlands, June 2008a.

C. Lundquist and T. B. Schön. Road geometry estimation and vehicle tracking using a single track model. Technical Report LiTH-ISY-R-2844, Department of Electrical Engineering, Linköping University, Sweden, Linköping, Sweden, March 2008b.

C. Lundquist and T. B. Schön. Recursive identification of cornering stiffness parameters for an enhanced single track model. In *Proceedings of the IFAC Symposium on System Identification*, pages 1726–1731, Saint-Malo, France, July 2009.

C. Lundquist and T. B. Schön. Joint ego-motion and road geometry estimation. *Information Fusion*, 12:253–263, October 2011.

C. Lundquist, U. Orguner, and T. B. Schön. Tracking stationary extended objects for road mapping using radar measurements. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 405–410, Xi'an, China, June 2009.

B. Ma, S. Lakshmanan, and A. O. Hero. Simultaneous detection of lane and pavement boundaries using model-based multisensor fusion. *IEEE Transactions on Intelligent Transportation Systems*, 1(3):135–147, September 2000.

J. C. McCall and M. M. Trivedi. Video-based lane estimation and tracking for driver assistance: Servey, system, and evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):20–37, March 2006.

M. Mitschke and H. Wallentowitz. *Dynamik der Kraftfahrzeuge*. Springer, Berlin, Heidelberg, 4 edition, 2004.

A. Muller, M. Manz, M. Himmelsbach, and H.J. Wunsche. A model-based object following system. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 242–249, Xi'an, China, June 2009.

M. Nikolova and A Hero. Segmentation of a road from a vehicle-mounted radar and accuracy of the estimation. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 284–289, Dearborn, MI, USA, October 2000.

Robert Bosch GmbH, editor. *Automotive Handbook*. SAE Society of Automotive Engineers, 6 edition, 2004.

W. J. Rugh. *Linear System Theory*. Information and system sciences series. Prentice Hall, Upper Saddle River, NJ, USA, 2 edition, 1996.

S. F. Schmidt. Application of state-space methods to navigation problems. *Advances in Control Systems*, 3:293–340, 1966.

T. B. Schön. *Estimation of Nonlinear Dynamic Systems – Theory and Applications*. PhD thesis No 998, Linköping Studies in Science and Technology, Department of Electrical Engineering, Linköping University, Sweden, February 2006.

R. Schubert, G. Wanielik, and K. Schulze. An analysis of synergy effects in an omnidirectional modular perception system. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 54–59, Xi'an, China, June 2009.

G. L. Smith, S. F. Schmidt, and L. A. McGee. Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle. Technical Report TR R-135, NASA, 1962.

J. Sparbert, K. Dietmayer, and D. Streller. Lane detection and street type classification using laser range images. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 454–459, Oakland, CA, USA, August 2001.

B. O. S. Teixeira, J. Chandrasekar, L. A. B. Torres, L. A. Aguirre, and D. S. Bernstein. State estimation for equality-constrained linear systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 6220–6225, New Orleans, LA, USA, December 2007.

A. Watanabe, T. Naito, and Y. Ninomiya. Lane detection with roadside structure using on-board monocular camera. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 191–196, Xi'an, China, June 2009.

A. Wedel, U. Franke, H. Badino, and D. Cremers. B-spline modeling of road surfaces for freespace estimation. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 828–833, Eindhoven, The Netherlands, June 2008.

H. Weigel, P. Lindner, and G. Wanielik. Vehicle tracking with lane assignment by camera and Lidar sensor fusion. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 513–520, Xi'an, China, June 2009.

W. S. Wijesoma, K. R. S. Kodagoda, and A. P. Balasuriya. Road-boundary detection and tracking using ladar sensing. *IEEE Transactions on Robotics and Automation*, 20(3):456–464, June 2004.

J. Y. Wong. *Theory Of Ground Vehicles*. John Wiley & Sons, New York, USA, 3 edition, 2001.

Z. Zomotor and U. Franke. Sensor fusion for improved vision based lane recognition and object tracking with range-finders. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 595–600, Boston, MA, USA, November 1997.

Paper C

Extended Target Tracking Using Polynomials With Applications to Road-Map Estimation

Authors: Christian Lundquist, Umut Orguner and Fredrik Gustafsson

Edited version of the paper:

C. Lundquist, U. Orguner, and F. Gustafsson. Extended target tracking using polynomials with applications to road-map estimation. *IEEE Transactions on Signal Processing*, 59(1):15–26, January 2011c.

The paper presents data that was previously published in:

C. Lundquist, U. Orguner, and F Gustafsson. Estimating polynomial structures from radar data. In *Proceedings of the International Conference on Information Fusion*, Edinburgh, UK, July 2010b.
C. Lundquist, U. Orguner, and T. B. Schön. Tracking stationary extended objects for road mapping using radar measurements. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 405–410, Xi'an, China, June 2009.

Extended Target Tracking Using Polynomials With Applications to Road-Map Estimation

Christian Lundquist, Umut Orguner and Fredrik Gustafsson

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

Abstract

This paper presents an extended target tracking framework which uses polynomials in order to model extended objects in the scene of interest from imagery sensor data. State space models are proposed for the extended objects which enables the use of Kalman filters in tracking. Different methodologies of designing measurement equations are investigated. A general target tracking algorithm that utilizes a specific data association method for the extended targets is presented. The overall algorithm must always use some form of prior information in order to detect and initialize extended tracks from the point tracks in the scene. This aspect of the problem is illustrated on a real life example of road-map estimation from automotive radar reports along with the results of the study.

1 Introduction

This contribution relates to tracking applications where a sensor platform observes objects in local coordinates (x,y,z). We focus on the case of a moving sensor platform and stationary objects, but the framework can be generalized to any target tracking scenario. Some of these detected objects are points, while others are extended objects. Tracking point shaped objects have been the main focus in the tracking literature for a long time Blackman and Popoli (1999); Bar-Shalom and Li (1995). In particular in the computer vision literature, tracking extended objects has become popular during the last decades. The literature models extended objects as closed contours, like simple geometrical shapes as rectangles and ellipses, or arbitrary shapes as splines Bartels et al. (1987) and active contours Blake and Isard (1998). We propose to complement these models of extended objects with detection and data association methods for sensors providing point measurements. Letting x be the longitudinal, y the lateral and z the vertical direction, possible applications include the following three automotive and two naval examples:

- Road curvature models $y = f(x; \theta)$ based on lane markers detected by a vision sensor. Such models are needed in all future collision avoidance systems, in particular lane assist systems Gustafsson (2009).
- Similar extended objects along the road side $y = f(x; \theta)$ based on radar mainly. Such a model is a useful complement to the vision based road model above, or for autonomous vehicles driving in unstructured environments. This is our application example in Section 6.
- Road inclination models $z = f(x; \theta)$ based on a combination of vision and radar with good azimuth resolution. This can be used in look-ahead control Hellström (2010) for fuel efficient cruise controllers.
- Shore models $x = f(y; \theta)$ based on a radar to describe the shape of islands and parts of the coastline. This can be used in robust navigation systems based on map matching, see for instance the extended approach in Karlsson and Gustafsson (2006); Svensson (2009).
- Vertical shape models $z = f(y; \theta)$ based on a camera describing the altitude variations, for instance, of an island as seen from the sensor platform. Applications are similar to the previous point.

The common theme in these examples is that the model is parameterized in a given model class (we will study polynomials), and that both the input and output are observed variables, leading to an errors in variables (EIV) Söderström (2007) problem. This is, to the best of the authors knowledge, an original approach to describe extended objects. We derive a modified measurement update for the EIV model, and demonstrate its increased accuracy compared to the straightforward errors in output (EIO) approximation for filtering applications. The detection procedure requires some form of prior information on the shape of the extended object because many different models can suit the same set of measurements.

This paper is outlined as follows. We begin with an overview of the related literature in Section 2. Connections to several different fields of research are pointed out and this makes this section fairly long. However, the subsequent sections are organized such that the uninterested reader can skip this section without causing a loss of understanding for the the rest of the paper. A basic problem formulation of point and extended target tracking using polynomials as extended objects is given in Section 3. The state space representations that we are going to use in extended target tracking are introduced in Section 4. Our general algorithm for tracking point and extended targets is summarized in Section 5. Section 6 introduces the practical application. We apply the ideas presented in earlier sections to the road map estimation problem utilizing the automotive radar reports. The experiments done and the results obtained are presented in Section 6.3. The paper is concluded in Section 7.

2 Related Literature

Extended target tracking has been studied in the literature for some years and we give a brief overview in Section 2.1. Section 2.2 summarizes a couple of estimation methods proposed earlier for curves and contours. One of our proposed solutions is stated to be related to an estimation framework called as errors in variables problem and some references for this are given in Section 2.3. Finally, different solutions related to our motivating application example of road edge estimation are discussed in Section 2.4.

2.1 Extended Target Tracking

A target is denoted extended whenever the target extent is larger than the sensor resolution, i.e., it is large enough to give rise to more than one measurement per time step. Common methods used to track extended objects are very similar to the ones used for tracking a group of targets moving in formation, see e.g., Ristic et al. (2004). The bibliography Waxman and Drummond (2004) provides a comprehensive overview of existing literature in the area of group and cluster tracking. One conventional method is to model the target as a set of point features in a target reference frame, each of which may contribute at most one sensor measurement Mahler (2007). The exact location of a feature in the target reference frame is often assumed uncertain. The motion of an extended target is modeled through the process model in terms of the translation and rotation of the target reference frame relative to a world coordinate frame, see e.g., Dezert (1998). As is the case most of the time, not all measurements arise from features belonging to targets and some are due to false detections (clutter). The association hypotheses are derived through some data association algorithm. In Vermaak et al. (2005) a method is proposed where the association hypotheses are included in the state vector and the output of the tracking filter is a joint posterior density function of the state vector and the association hypotheses.

Instead of modeling the target as a number of point features, the target may be represented by a spatial probability distribution. It is more likely that a measurement comes from a region of high spatial density than from a sparse region. In Gilholm and Salmond (2005); Gilholm et al. (2005) it is assumed that the number of received target and clutter measurements are Poisson distributed, hence several measurements may originate from the same target. Each target related measurement is an independent sample from the spatial distribution. The spatial distribution is preferable where the point source models are poor representations of reality, that is, in the cases where the measurement generation is diffuse. In Boers et al. (2006) a similar approach is presented, but since raw data is considered, no data association hypotheses are needed.

In many papers dealing with the shape of a target it is assumed that the sensor is also able to measure one or more dimensions of the target's extent. A high-resolution radar sensor may provide measurements of a targets down-range extent, i.e., the extension of the objects along the line-of-sight. The information of the target's extent is incorporated in the tracking filter aiding the tracking process to maintain track on the target when it is close to other objects. An elliptical target model, to represent an extended target or a group of targets, is proposed in Drummond et al. (1990). The idea is improved in Salmond and Parr (2003) with a solution based on EKF, in Ristic and Salmond (2004) based on UKF and in Angelova and Mihaylova (2008) with a solution based on a Monte Carlo (MC) algorithm.

2.2 Contours and Curves

Modeling of extended targets in this work is very similar to active contours Blake and Isard (1998) and snakes Kass et al. (1988), which model the outline of an object based on 2 dimensional image data, studied extensively in computer vision. The active contour models are also polynomials represented by B-splines. Moreover, the probabilistic processing of *active contours* from image data pioneered mainly by Blake and Isard Blake and Isard (1998) uses similar estimation framework like Kalman or particle filters. The underlying low-level processing involved, on the other hand, assumes reasonably the existence of image data from which features (like Harris corners) or feature maps (like Harris measures etc.) can be extracted. The so-called Condensation algorithm Isard and Blake (1998), for example, searches for suitable sets of features in the image data (or in the feature map) iteratively for each of its different hypotheses (particles) in the calculation of the likelihoods. In this respect, the active contour framework would be an important candidate for doing tracking with the raw sensor data without any thresholding which is named as “track-before-detect” in the target tracking literature. The approach presented in this work carries the distinction of working only with a single set of features provided most of the time by thresholding of the raw sensor data (like conventional target tracking) and hence is mainly aimed at applications where the users (of the sensors) either are unable to reach or do not have the computation power to work with the raw sensor data.

The mapping of the boundaries of complex objects is also achieved in Lazarus et al. (2010) using splinegon techniques and a range sensor such as e.g., laser mounted on moving vehicles.

2.3 Errors In Variables

The use of “noisy” measurements as model parameters in this work makes this paper directly related to the errors in variables framework, where some of the independent variables are contaminated by noise. Such a case is common in the field of system identification Ljung (1999) when not only the system outputs, but also the inputs are measured imperfectly. Examples of such EIV representations can be found in Söderström (2007) and a representative solution is proposed in Guidorzi et al. (2003); Diversi et al. (2003). The Kalman filter cannot be directly applied to such EIV processes as discussed in Guidorzi et al. (2003),Roorda and Heij (1995). Nevertheless, an extension of the Kalman filter, where the state and the output are optimally estimated in the presence of state and output noise is proposed in Diversi et al. (2005). The EIV problem is also closely related to the total least squares methodology, which is well described in the papers van Huffel and Zha (1993); Boley and Sutherland (1993); De Moor (1993), in the survey paper Markovsky and van Huffel (2007) and in the textbook Björck (1996).

2.4 Application

An important part of this contribution is the application of the ideas to the real world example of road map estimation using automotive radar sensor reports. In this context, the point and extended targets existing in the scene would represent e.g., lamp posts and guardrails along the road, respectively. Our earlier work Lundquist et al. (2009) contains the findings from similar tracking scenarios. This also connects our work to the approaches presented in the literature for the problem of road edge estimation. The third order approximation of the two sided (left and right) "clothoid model" has been used in connection with Kalman filters in Kirchner and Heinrich (1998) and Polychronopoulos et al. (2004) for laser scanner measurements and radar measurements, respectively. In Lundquist and Schön (2009) two road edge models are proposed, one of which is very similar to the model proposed in Polychronopoulos et al. (2004), and used a constrained quadratic program to solve for the parameters. A linear model represented by its midpoint and orientation (one for each side of the road) is utilized by Wijesoma et al. (2004) with lidar sensing for tracking road-curbs. Later, the results of Wijesoma et al. (2004) were enhanced in Kodagoda et al. (2006) with the addition of image sensors. A similar extended Kalman filtering based solution is given in Fardi et al. (2003), where a circular road edge modeling framework is used. Recently, the particle filters (also referred to as condensation in image and video processing) have been applied to the road edge estimation problem in Wang et al. (2008) with an hyperbolic road model.

3 Problem Formulation

The examples mentioned in Section 1 all involve 2-dimensional extended objects, modeled by any configuration of two local coordinates (x, y or z). In the following parts of the paper, we describe all models in the x-y-plane without any loss of generality, i.e., the coordinates are easily exchanged to describe objects in any other space.

Suppose we are given the sensor measurements in batches of Cartesian x and y coordinates as follows:

$$\left\{\mathbf{z}_k^{(i)} \triangleq \begin{bmatrix} \mathrm{x}^{(i)} & \mathrm{y}^{(i)} \end{bmatrix}_k^{\mathrm{T}}\right\}_{i=1}^{N_{\mathbf{z}_k}} \tag{1}$$

for discrete time instants $k = 1, \ldots, K$. In many cases in reality (e.g., radar, laser and stereo vision) and in the practical application considered in this work, the sensor provides range r and azimuth angle ψ given as

$$\left\{\bar{\mathbf{z}}_k^{(i)} \triangleq \begin{bmatrix} r^{(i)} & \psi^{(i)} \end{bmatrix}_k^{\mathrm{T}}\right\}_{i=1}^{N_{\mathbf{z}_k}}. \tag{2}$$

In such a case we assume that some suitable standard polar to Cartesian conversion algorithm is used to convert these measurements into the form (1).

The measurements $\mathbf{z}$ are noisy point measurements originating from one of the following sources

- Point sources with state vector

$$\mathbf{x}_\mathrm{P} \triangleq \begin{bmatrix} \mathsf{x} & \mathsf{y} \end{bmatrix}^\mathrm{T}, \tag{3}$$

  which represents the true Cartesian position of the source.

- Extended sources which are represented by an n^th order polynomial given as

$$\mathsf{y} = a_0 + a_1\mathsf{x} + a_2\mathsf{x}^2 + \ldots + a_n\mathsf{x}^n \tag{4}$$

  in the range $[\mathsf{x}_\mathrm{start}, \mathsf{x}_\mathrm{end}]$ where $\mathbf{x}_a \triangleq \begin{bmatrix} a_0 & a_1 & \cdots a_n \end{bmatrix}^\mathrm{T}$ are the polynomial coefficients and $\begin{bmatrix} \mathsf{x} & \mathsf{y} \end{bmatrix}^\mathrm{T}$ are planar Cartesian coordinates. Note that the coordinate y is a function of x and that the direction of the coordinate frame is chosen dependent on the application in mind. The state vector is defined as

$$\mathbf{x}_\mathrm{L} \triangleq \begin{bmatrix} \mathbf{x}_a & \mathsf{x}_\mathrm{start} & \mathsf{x}_\mathrm{end} \end{bmatrix}^\mathrm{T}. \tag{5}$$

- False detections

The state models considered in this contribution are described, in general, by the state space equations

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k, \tag{6a}$$

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{e}_k, \tag{6b}$$

where $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{y}$ denotes the state, the input signal, and the output signal, while $\mathbf{w} \sim \mathcal{N}(0, Q)$ and $\mathbf{e} \sim \mathcal{N}(0, R)$ are the process and measurement noise, respectively. The use of an input signal is explained in Section 4.1. The time index is denoted with k.

One of the main contributions of our work is the specific state space representation we propose for extended targets which is presented in Section 4. A polynomial is generally difficult to handle in a filter, since the noisy measurements are distributed arbitrarily along the polynomial. In this respect, the measurement model (6b) contains parts of the actual measurement vector as parameters. For the sake of simplicity, the tracked objects are assumed stationary, resulting in very simple motion models (6a). However, the motion or process model may easily be substituted and chosen arbitrarily to best fit its purpose.

The aim is to obtain posterior estimates of the point sources $\{\mathbf{x}_{\mathrm{P},k|k}^{(i)}\}_{i=1}^{N_\mathrm{P}}$ and the extended sources $\{\mathbf{x}_{\mathrm{L},k|k}^{(i)}\}_{i=1}^{N_\mathrm{L}}$ given all the measurements $\left\{\{\mathbf{z}_\ell^{(i)}\}_{i=1}^{N_{\mathbf{z}_\ell}}\right\}_{\ell=1}^{k}$ recursively in time.

4 State Space Representation for an Extended Object

This section first investigates the measurement model to be used with extended sources in Section 4.1 and then makes short remarks about the state dynamics equation for polynomials in Section 4.2.

4.1 Measurement Model

This section describes how a point measurement $\mathbf{z}$ relates to the state $\mathbf{x}_\mathrm{L}$ of an extended object. For this purpose, we derive a measurement model in the form (6b), which describes the relation between the state variables $\mathbf{x}_\mathrm{L}$, defined in (5), and output signals $\mathbf{y}$ and input signals $\mathbf{u}$. Notice that, for the sake of simplicity, we also drop the subscripts k specifying the time stamps of the quantities.

The general convention in modeling is to make the definitions

$$\mathbf{y} \triangleq \mathbf{z}, \qquad \mathbf{u} \triangleq \emptyset, \tag{7}$$

where $\emptyset$ denotes the empty set meaning that there is no input. In this setting, it is extremely difficult, if not impossible, to find a measurement model connecting the outputs $\mathbf{y}$ to the states $\mathbf{x}_\mathrm{L}$ in the form of (6b). Therefore, we are forced to use other selections for $\mathbf{y}$ and $\mathbf{u}$. Here, we make the selection

$$\mathbf{y} \triangleq \mathsf{y}, \qquad \mathbf{u} \triangleq \mathsf{x}. \tag{8}$$

Although being quite a simple selection, this choice results in a rather convenient linear measurement model in the state partition $\mathbf{x}_a$,

$$\mathbf{y} = H_a(\mathbf{u})\mathbf{x}_a + \mathbf{e}_a, \tag{9}$$

where $H_a(\mathbf{u}) = \begin{bmatrix} 1 & \mathsf{x} & \mathsf{x}^2 & \cdots & \mathsf{x}^n \end{bmatrix}^\mathrm{T}$. It is the selection in (8) rather than (7) that allows us to use the standard methods in target tracking with clever modifications. Such a selection as (8) is also in accordance with the errors in variables representations where measurement noise are present in both the outputs and inputs, i.e., the observation $\mathbf{z}$ can be partitioned according to

$$\mathbf{z} = \begin{bmatrix} \mathbf{u} \\ \mathbf{y} \end{bmatrix}. \tag{10}$$

We express the measurement vector given in (1) in terms of a noise free variable $\mathbf{z}_0$ which is corrupted by additive measurement noise $\tilde{\mathbf{z}}$ according to

$$\mathbf{z} = \mathbf{z}_0 + \tilde{\mathbf{z}}, \quad \tilde{\mathbf{z}} \sim \mathcal{N}(0, \Sigma_\mathsf{c}), \tag{11}$$

where the covariance Σ_c can be decomposed as

$$\Sigma_\mathsf{c} = \begin{bmatrix} \Sigma_\mathsf{x} & \Sigma_\mathsf{xy} \\ \Sigma_\mathsf{xy} & \Sigma_\mathsf{y} \end{bmatrix}. \tag{12}$$

Note that, in the case the sensor provides measurements only in polar coordinates (2), one has to convert both the measurement $\bar{\mathbf{z}}$ and the measurement noise covariance

$$\Sigma_{\mathsf{p}} = \operatorname{diag}(\sigma_r^2, \sigma_\psi^2) \tag{13}$$

into Cartesian coordinates, which is a rather standard procedure. Note that, in such a case, the resulting Cartesian measurement covariance Σ_{c} is, in general, not necessarily diagonal and hence Σ_{xy} of (12) might be non-zero.

Since the model (9) is conditionally linear given the measurements, the Kalman filter measurement update formulas can be used to incorporate the information in $\mathbf{z}$ into the extended source state $\mathbf{x}_{\mathrm{L}}$. An important question in this regard is what would be the measurement covariance of the measurement noise term $\mathbf{e}_a$ in (9). This problem can be tackled in two ways.

Errors in Output (EIO) scheme

Although the input terms defined in (8) are measured quantities (and hence affected by the measurement noise), and therefore the model parameters $H_a(\mathbf{u})$ are uncertain, in a range of practical applications where parameters are obtained from measurements, such errors are neglected. Thus, the first scheme we present here neglects all the errors in $H_a(\mathbf{u})$. In this case, it can easily be seen that

$$\mathbf{e}_a = \tilde{\mathbf{y}} \tag{14}$$

and therefore the covariance Σ_a of $\mathbf{e}_a$ is

$$\Sigma_a = \Sigma_{\mathsf{y}}. \tag{15}$$

This type of approach was also used in our previous work Lundquist et al. (2009) which presented earlier versions of the findings in this paper.

Errors in Variables (EIV) scheme

Neglecting the errors in the model parameters $H_a(\mathbf{u})$ can cause overconfidence in the estimates of recursive filters and can actually make data association difficult in tracking applications (by causing too small gates). We, in this second scheme, use a simple methodology to take the uncertainties in $H_a(\mathbf{u})$ into account in line with EIV framework. Assuming that the elements of the noise free quantity $\mathbf{z}_0$ satisfy the polynomial equation exactly, we get

$$\mathbf{y} - \tilde{\mathbf{y}} = H_a(\mathbf{u} - \tilde{\mathbf{u}})\mathbf{x}_a, \tag{16a}$$

$$\mathsf{y} - \tilde{\mathsf{y}} = \begin{bmatrix} 1 & \mathsf{x} - \tilde{\mathsf{x}} & (\mathsf{x} - \tilde{\mathsf{x}})^2 & \cdots & (\mathsf{x} - \tilde{\mathsf{x}})^n \end{bmatrix} \mathbf{x}_a, \tag{16b}$$

which can be approximated using a first order Taylor expansion resulting in

$$\mathbf{y} \approx H_a(\mathbf{u})\mathbf{x}_a - \widetilde{H}_a(\mathbf{u})\tilde{\mathsf{x}}\mathbf{x}_a + \tilde{\mathsf{y}} \tag{17a}$$

$$= H_a(\mathbf{u})\mathbf{x}_a + \tilde{h}_a(\mathbf{x}_a, \mathbf{u}) \begin{bmatrix} \tilde{\mathsf{x}} \\ \tilde{\mathsf{y}} \end{bmatrix}, \tag{17b}$$

with

$$H_a(\mathbf{u}) = \begin{bmatrix} 1 & \mathsf{x} & \mathsf{x}^2 & \cdots & \mathsf{x}^n \end{bmatrix}, \tag{17c}$$

$$\widetilde{H}_a(\mathbf{u}) = \begin{bmatrix} 0 & 1 & 2\mathrm{x} & \cdots & n\mathrm{x}^{n-1} \end{bmatrix}, \tag{17d}$$

$$\tilde{h}_a(\mathbf{x}_a, \mathbf{u}) = \begin{bmatrix} -a_1 - 2a_2\mathrm{x} - \cdots - na_n\mathrm{x}^{n-1} & 1 \end{bmatrix}. \tag{17e}$$

Hence, the noise term $\mathbf{e}_a$ of (9) is given by

$$\mathbf{e}_a = \tilde{\mathrm{y}} - \widetilde{H}_a \tilde{\mathrm{x}} \mathbf{x}_a = \tilde{h}_a(\mathbf{x}_a, \mathbf{u}) \begin{bmatrix} \tilde{\mathrm{x}} \\ \tilde{\mathrm{y}} \end{bmatrix} \tag{18}$$

and its covariance is given by

$$\begin{aligned} \Sigma_a = \mathrm{E}(\mathbf{e}_a \mathbf{e}_a^{\mathrm{T}}) &= \Sigma_{\mathrm{y}} + \widetilde{H}_a \mathbf{x}_a \Sigma_{\mathrm{x}} \mathbf{x}_a^{\mathrm{T}} \widetilde{H}_a^{\mathrm{T}} - 2\widetilde{H}_a \mathbf{x}_a \Sigma_{\mathrm{xy}} \\ &= \tilde{h}_a(\mathbf{x}_a, \mathbf{u}) \Sigma_{\mathrm{c}} \tilde{h}_a^{\mathrm{T}}(\mathbf{x}_a, \mathbf{u}). \end{aligned} \tag{19}$$

Note that the EIV covariance Σ_a depends on the state variable $\mathbf{x}_a$.

Example

An example is used to compare the performances of the EIO and the EIV schemes. A second order polynomial with the true states

$$\mathbf{x}_a = \begin{bmatrix} a_0 & a_1 & a_2 \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} -20 & -0.5 & 0.008 \end{bmatrix}^{\mathrm{T}} \tag{20}$$

is used and 100 uniformly distributed measurements are extracted in the range $\mathrm{x} = [0, 200]$. The measurements are given on polar form as in (2) and zero mean Gaussian measurement noise with covariance as in (13) is added using the parameters

$$\text{Sensor 1:} \quad \sigma_{r1} = 0.5, \quad \sigma_{\psi 1} = 0.05, \tag{21a}$$

$$\text{Sensor 2:} \quad \sigma_{r2} = 10, \quad \sigma_{\psi 2} = 0.05, \tag{21b}$$

$$\text{Sensor 3:} \quad \sigma_{r3} = 10, \quad \sigma_{\psi 3} = 0.005, \tag{21c}$$

to simulate different type of sensors. Sensor 1 represents a sensor with better range than bearing accuracy, whereas the opposite holds for sensor 3. Sensor 2 has about the same range and bearing accuracies. The true polynomial and the measurements are shown in Figure 1. The measurements are transformed into Cartesian coordinates. The following batch methods ($N_{\mathbf{z}_k} = 100$, $K = 1$) are applied to estimate the states

- Least squares (LS EIO) estimator,
- Weighted least squares (WLS EIO) with EIO covariance,
- Weighted least squares (WLS EIV) with EIV covariance. The state $\mathbf{x}_a$ used in (19) is estimated through a least squares solution in advance.

Furthermore, the states are estimated recursively ($N_{\mathbf{z}_k} = 1$, $K = 100$) using

- Kalman filter (KF EIO) with EIO covariance,
- Kalman filter (KF EIV) with EIV covariance. The predicted state estimate $\hat{\mathbf{x}}_{a,k|k-1}$ is used in (19) to derive $R_{a,k}$.

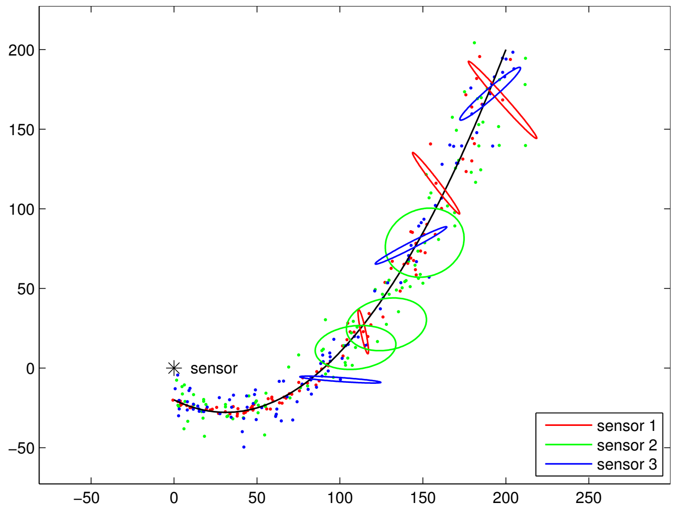


***Figure 1:** The true polynomial and the extracted measurements are shown. Covariance ellipses are drawn for a few measurements to show the direction of the uncertainties.*

- Unscented Kalman filter (UKF EIV) with sigma points derived by augmenting the state vector with the noise terms

$$\tilde{\mathbf{z}} = \begin{bmatrix} \tilde{\mathbf{u}} & \tilde{\mathbf{y}} \end{bmatrix}^{\mathrm{T}}, \quad \tilde{\mathbf{z}} \sim \mathcal{N}(0, \Sigma_{\mathrm{c}}), \tag{22}$$

  to consider the error in all directions (as in the EIV case), and to deal with the nonlinear transformations of the noise terms.

The covariance of the process noise is set to zero, i.e., $Q = 0$, since the target is not moving. The initial conditions of the state estimate are selected as

$$\hat{\mathbf{x}}_{a,0} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}}, \tag{23a}$$

$$P_{a,0} = \left(\frac{\sigma_a^2}{\kappa} \operatorname{diag}(\mathbf{x}_a) \right)^2, \tag{23b}$$

where $\kappa = 3$ and $\sigma_a^2 = 8$. Note that every estimate's initial uncertainty is set to be a scaled version of its true value in (23b).

The RMSE values for 1000 Monte Carlo (MC) simulations are given in Table 1. The RMSE of the EIV schemes is clearly lower than the other methods, especially for sensor 1 and 3 with non-symmetric measurement noise. This result justifies the extra computations required for calculating the EIV covariance. There is only a small difference in the performance between the KF EIV and the UKF EIV for the second order polynomial. This is a clear indication of that the simple Taylor series approximation used in deriving the EIV covariance is accurate enough.

Table 1: *RMSE values for the extended object in Figure 1. The sensor configurations are defined in* (21)

Sensor		LS	WLS	WLS	KF	KF	UKF
		EIO	EIO	EIV	EIO	EIV	EIV
	a_0	5.10	0.55	0.45	0.55	0.48	0.49
1	a_1	0.18	0.034	0.024	0.034	0.029	0.029
	$a_2 \cdot 10^{-3}$	1.06	0.29	0.24	0.29	0.31	0.32
	a_0	4.90	3.54	3.35	2.90	2.44	2.36
2	a_1	0.20	0.11	0.099	0.10	0.11	0.10
	$a_2 \cdot 10^{-3}$	1.21	0.77	0.66	0.78	0.83	0.80
	a_0	2.53	30.51	3.51	30.47	4.81	4.35
3	a_1	0.068	0.45	0.072	0.45	0.12	0.12
	$a_2 \cdot 10^{-3}$	0.39	1.27	0.40	1.26	0.62	0.60

1 Remark (Robustness). We use only Gaussian noise representations in this paper which would directly connect our work to the (implicit) minimization of LS-type cost functions which are known to be non-robust against outliers. The algorithms we are going to propose for estimation are going to be protected, to some extent, against this by classical techniques like gating Blackman and Popoli (1999); Bar-Shalom and Li (1995). Extra robustness features can be accommodated by the use of Huber Huber and Ronchetti (2009) and ℓ_1 norms etc. in the estimation which were considered to be outside the scope of this paper.

Complete Measurement Model for an Extended Object

Up to this point, we have only considered how the observation $\mathbf{z}$ relates to the state component $\mathbf{x}_a$. It remains to discuss the relation between the observation and the start $\mathrm{x}_{\text{start}}$ and the end points x_{end} of the polynomial. The measurement information must only be used to update these components of the state if the new observations of the extended source lie outside the range of the polynomial. We can define the following (measurement dependent) measurement matrix for this purpose:

$$H_{se} = \begin{cases} \begin{bmatrix} 1 & 0 \end{bmatrix} & \text{if } \mathrm{x} \leq \mathrm{x}_{\text{start},k|k-1} \\ \begin{bmatrix} 0 & 1 \end{bmatrix} & \text{if } \mathrm{x} \geq \mathrm{x}_{\text{end},k|k-1} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & \text{otherwise.} \end{cases} \tag{24}$$

The complete measurement model of an extended object can now be summarized by

$$\mathbf{z} = H_{\mathrm{L}}\mathbf{x}_{\mathrm{L}} + \mathbf{e}, \quad \mathbf{e} \sim \mathcal{N}(0, R_{\mathrm{L}}(\mathbf{x}_{\mathrm{L}})), \tag{25a}$$

with

$$H_{\mathrm{L}} = \begin{bmatrix} \mathbf{0}^{1\times n} & H_{se} \\ H_a & \mathbf{0}^{1\times 2} \end{bmatrix}, \tag{25b}$$

$$R_{\mathrm{L}}(\mathbf{x}_{\mathrm{L}}) = \operatorname{blkdiag}(\Sigma_{\mathrm{x}}, \Sigma_a(\mathbf{x}_{\mathrm{L}})). \tag{25c}$$

Put in words, if the x-component of a new measurement is closer to the sensor than the start point of the line $\mathsf{x}_{\text{start}}$ it is considered in the measurement equation (24) and can used to update this state variable. Analogously, if a new measurement is more distant than the end point of the line x_{end} it is considered in (24). Further, if a measurements is in between the start and end point of the line, the measurement model is zero in (24) and there is no relation between this measurement and the state variables $\mathsf{x}_{\text{start}}$ or x_{end}.

4.2 Process Model

Any model as e.g., the standard constant velocity or coordinated turn model may be used for the targets. For simplicity it is assumed that the targets are stationary in this contribution, thus the process model on the form (6a) is linear and may be written

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{w}_k. \tag{26}$$

To increase the flexibility of the extended object an assumption about the dynamic behavior of its size is made. The size of the extended object is modeled to shrink with a factor $0.9 < \lambda < 1$ according to

$$\mathsf{x}_{\text{start},k+1} = \mathsf{x}_{\text{start},k} + \lambda(\mathsf{x}_{\text{end},k} - \mathsf{x}_{\text{start},k}), \tag{27a}$$

$$\mathsf{x}_{\text{end},k+1} = \mathsf{x}_{\text{end},k} - \lambda(\mathsf{x}_{\text{end},k} - \mathsf{x}_{\text{start},k}), \tag{27b}$$

leading to the following process model for the polynomial

$$F_{\mathrm{L}} = \begin{bmatrix} \boldsymbol{I}^{n\times n} & \mathbf{0}^{n\times 2} & \\ \mathbf{0}^{2\times n} & 1-\lambda & \lambda \\ & \lambda & 1-\lambda \end{bmatrix}. \tag{28}$$

This shrinking behavior for the polynomials allows for automatic adjustment of the start and end points of the polynomials according to the incoming measurements.

Some dynamics is included into the process model of the polynomials for the start and the end point components of the state, $\mathsf{x}_{\text{start}}$ and x_{end}.

5 Multi-Target Tracking Algorithm

In this section, we describe the target tracking algorithm we use to track polynomial shaped extended objects and point objects at the same time. Suppose at time $k-1$, we have $N_{\mathrm{P},k-1}$ estimated point objects whose states and covariances are given by $\{\hat{\mathbf{x}}^{(i)}_{\mathrm{P},k-1|k-1}, P^{(i)}_{\mathrm{P},k-1|k-1}\}_{i=1}^{N_{\mathrm{P},k-1}}$ and $N_{\mathrm{L},k-1}$ estimated extended objects whose states and covariances are given by $\{\hat{\mathbf{x}}^{(j)}_{\mathrm{L},k-1|k-1}, P^{(j)}_{\mathrm{L},k-1|k-1}\}_{j=1}^{N_{\mathrm{L},k-1}}$. Basically, the task of the tracking algorithm is to obtain the updated states and the covariances $\{\hat{\mathbf{x}}^{(i)}_{\mathrm{P},k|k}, P^{(i)}_{\mathrm{P},k|k}\}_{i=1}^{N_{\mathrm{P},k}}$, $\{\hat{\mathbf{x}}^{(j)}_{\mathrm{L},k|k}, P^{(j)}_{\mathrm{L},k|k}\}_{j=1}^{N_{\mathrm{L},k}}$ when the set of $N_{\mathbf{z}_k}$ measurements given by $\{\mathbf{z}_k^{(m)}\}_{m=1}^{N_{\mathbf{z}_k}}$ is obtained. Notice that the number of these updated summary statistics kept in the algorithm can change with appearance or disappearance of the objects in the environment. Hence, the number of each type of objects must also be estimated by the tracking algorithm.

We, in this work, consider a conventional multi-target tracking algorithm that uses a specific track initiation and deletion logic for handling the variable number of objects to track. Each cycle of estimation starts with the classical (association) hypotheses reduction technique "gating" which is followed by a data association process which determines the measurement to (point or extended) track associations. When the measurement to track associations are known, the current tracks are updated with their associated measurement. In this measurement update step, the results of Section 4.1 are used for updating the extended object states and covariances. After the updates are completed one has to do track handling. This operation deals with

- Extended object generations from established point targets. This part of the tracking algorithm must depend heavily upon the problem under investigation and hence the prior information that one has about the possible extended objects because the mapping from a number of points to a number of polynomials is not unique.
- Track deletion. This operation deletes the old tracks that have not been associated to any measurements for a significant amount of time.

Below, in separate subsections, we investigate the steps of the tracking algorithm we propose in more detail.

5.1 Gating and Data Association

Each of the $N_{\mathbf{z}_k}$ observations $\mathbf{z}_k^{(m)}$, $m = 1, \ldots, N_{\mathbf{z}_k}$ from the sensor measurements can be associated to one of the existing point tracks $\mathbf{x}_{\mathrm{P}}^{(i)}$, the extended tracks $\mathbf{x}_{\mathrm{L}}^{(j)}$ or a new point track is initiated. The number of association events (hypotheses) is extremely large. The classical technique to reduce the number of these hypotheses is called gating, see e.g., Bar-Shalom and Fortmann (1988). In this section we show how to apply gating and to make a nearest-neighbor type data association based on likelihood ratio tests. Other more complicated data association methods like multiple hypothesis tracking, according to e.g., Reid (1979), or joint probabilistic data association, as described by e.g., Bar-Shalom and Fortmann (1988), can also be used in this framework with appropriate modifications for the extended sources. However, these are quite complicated and computationally costly approaches and the nearest neighbor type algorithm has been found to give sufficiently good performance for our application. The gating and the data association are performed according to the following calculations. The likelihood $\ell_{m,i}$ that the observation $\mathbf{z}^{(m)}$ corresponds to the i^{th} point track $\mathrm{P}^{(i)}$ is given by

$$\ell_{m,i} = \begin{cases} \mathcal{N}(\mathbf{z}_k^{(m)}; \hat{\mathbf{z}}_{\mathrm{P},k|k-1}^{(i)}, S_{\mathrm{P},k|k-1}^{(i)}), & \text{if } \mathbf{z}_k^{(m)} \in \mathcal{G}_{\mathrm{P}}^{(i)} \\ 0, & \text{otherwise} \end{cases} \tag{29}$$

where $\hat{\mathbf{z}}^{(i)}_{\mathrm{P},k|k-1}$ is the predicted measurement of the point P_i according to the model (39) and $S^{(i)}_{\mathrm{P},k|k-1}$ is its covariance (innovation covariance) in the Kalman filter. The gate $\mathcal{G}^{(i)}_{\mathrm{P}}$ is defined as the region

$$\mathcal{G}^{(i)}_{\mathrm{P}} \triangleq \left\{ \mathbf{z} \middle| \left(\mathbf{z} - \hat{\mathbf{z}}^{(i)}_{\mathrm{P},k|k-1}\right)^{\mathrm{T}} \left(S^{(i)}_{\mathrm{P},k|k-1}\right)^{-1} \left(\mathbf{z} - \hat{\mathbf{z}}^{(i)}_{\mathrm{P},k|k-1}\right) \leq \delta_{\mathrm{P}} \right\} \tag{30}$$

where δ_{P} is the gating threshold.

The likelihood $\ell_{m,j}$ that the observation m belongs to the j^{th} line is given by

$$\ell_{m,j} = \begin{cases} \mathcal{N}\left(\mathbf{y}^{L,(m)}_k ; \hat{\mathbf{y}}^{(j)}_{\mathrm{L},k|k-1}, S^{(j)}_{\mathrm{L},k|k-1}\right), & \text{if } \mathbf{z}^{L,(m)}_k \in \mathcal{G}_{\mathrm{L}_j} \\ 0, & \text{otherwise} \end{cases} \tag{31}$$

where $\hat{\mathbf{y}}^{(j)}_{\mathrm{L},k|k-1}$ is the predicted output of the measurement model (9) and $S^{(j)}_{\mathrm{L},k|k-1}$ is its covariance, both for the state estimates $\hat{\mathbf{x}}^{(j)}_{\mathrm{L}}$. The quantity $\mathbf{z}^{L,(m)}$ is representing the original measurement transformed into the line's corresponding coordinate frame and $\mathbf{y}^{L,(m)}_k$ is the y component of it. The gate $\mathcal{G}_{\mathrm{L}_j}$ is defined as

$$\mathcal{G}_{\mathrm{L}_j} \triangleq \left\{ \mathbf{z} = \begin{bmatrix} \mathbf{u} \\ \mathbf{y} \end{bmatrix} \middle| \frac{\left(\mathbf{y} - \hat{\mathbf{y}}^{(j)}_{L,k|k-1}\right)^2}{S^{(j)}_{\mathrm{L},k|k-1}} \leq \delta_{\mathrm{L}},\ \mathbf{x}^{(j)}_{\text{start},k|k-1} - \delta_s < \mathbf{u} < \mathbf{x}^{(j)}_{\text{end},k|k-1} + \delta_e \right\}. \tag{32}$$

and the innovation covariance is given by

$$S^{(j)}_{\mathrm{L},k|k-1} = H_a(\mathbf{u}) P^{(j)}_{a,k|k-1} H_a(\mathbf{u})^{\mathrm{T}} + \Sigma^{(j)}_{a,k|k-1}, \tag{33}$$

where $P^{(j)}_{a,k|k-1}$ is the state covariance of $\mathbf{x}^{(j)}_{a,k|k-1}$.

Having calculated the likelihood values, two matrices of likelihood values are formed, one matrix $\Gamma_{\mathrm{P}} \in \mathbb{R}^{N_{\mathbf{z}} \times N_{\mathrm{P}}}$ with the combinations of observations and points, according to (29), and one matrix $\Gamma_{\mathrm{L}} \in \mathbb{R}^{N_{\mathbf{z}} \times N_{\mathrm{L}}}$ with the combinations of observations and lines, according to (31).

The association procedure using Γ_{P} and Γ_{L} must be quite unconventional because in conventional multi-target tracking, one of the well-known assumptions is that at most a single measurement can originate from a target. However, obviously, in our problem polynomials can result in multiple measurements. On the other hand, we still have the assumptions that no two measurements may originate from the same point source and no two sources may give rise to the same measurements.

Our association mechanism which uses nearest neighbor type ideas with likelihood ratio tests is summarized as follows: First find the the maximum value of Γ_{P} and call the corresponding point state $i_{\max}$ and measurement $m_{\max}$. Then, find the maximum value of the m^{th} row, corresponding to measurement $m_{\max}$ of matrix Γ_{L} and call the corresponding polynomial state $j_{\max}$. The likelihood ratio denoted by $\Lambda(\mathbf{z}^{(m)})$ is now given by

$$\Lambda(\mathbf{z}^{(m)}) \triangleq \frac{\sqrt{\ell_{m,i_{\max}}}}{\ell_{m,j_{\max}}}, \tag{34}$$

where we take the square root of the point likelihood for unit matching, since the unit of (29) is $(\text{distance})^{-2}$ and the unit of (31) is $(\text{distance})^{-1}$. The corresponding *likelihood ratio test* is

$$\Lambda(\mathbf{z}^{(m)}) \underset{H_1}{\overset{H_0}{\gtrless}} \eta \tag{35}$$

where H_0 and H_1 correspond to hypotheses that the measurement $\mathbf{z}^{(m)}$ is associated to the point $\mathbf{x}_{\mathrm{P}}^{(i_{\max})}$ and to the line $\mathbf{x}_{\mathrm{L}}^{(j_{\max})}$, respectively. The threshold η is to be selected experimentally. More theory about likelihood ratio tests is given by e.g., van Trees (1968). When this test is performed, if the measurement $\mathbf{z}^{(m)}$ is associated to a point source P_i, then the values in the m^{th} row of the two matrices as well as the i^{th} column of the point likelihood matrix must be set to zero to exclude the measurement and the point source from further association. However, if $\mathbf{z}^{(m)}$ is associated to line L_j, then only the values in the m^{th} rows of the two matrices are set to zero because the line L_j can still be associated to other measurements. The procedure is repeated until all measurements with non-zero likelihood have been associated to either a point or a line. A new point track is to be initiated if the observations could not be associated to an existing state. This is true when a measurement is not in the gate of a non-associated point or an extended source.

5.2 Track Handling

Initially, our algorithm tries to generate point tracks from all incoming measurements, using any standard initialization logic. When the point tracks are established, at the end of each estimation cycle, the point track positions are examined and decisions are made on the existence of any extended sources. This process must be quite application dependent and it requires prior information on the possible forms of the polynomials to be extracted. For the purpose of giving some examples of such prior information, we go back to the five motivating application examples in Section 1 in the same order and list possible available prior information in each case.

- The model class of road curvatures is determined in road construction standards, and prior values may be based on the ego vehicle's motion.
- Extended objects along the road side may be based on prior knowledge of the road curvature. An example application with such details is presented in Section 6.
- The class of road inclination models are based on road construction standards and initial values may be based on maps or GPS data, to be improved in the tracking filter.
- A prior for the shore model is given by digital sea charts.

- The altitude of an island is initiated roughly based on the topographic information of a sea chart or as a standard model for islands in a given archipelago.

If some extended sources are detected in the detection process, states and covariances for this extended sources are calculated from the corresponding point tracks which are removed from the point track list after this procedure. The created extended source states and covariances, from then on, are treated as extended tracks.

When the environment changes some tracks might expire and get no measurements. When this happens, such tracks (point or extended) must be removed from the track list. For this purpose, for each track we keep a counter that holds the number of time instants that the corresponding track has not been associated to any measurement. When this counter exceeds a threshold, the corresponding track is removed from the track list.

6 Application Example and Use of Prior Information

As an application of the ideas presented, we consider the road map estimation problem. A good road map is important for collision avoidance systems, path planing and trajectory control. Measurements from an automotive radar mounted on a moving vehicle which we refer to as the ego vehicle are used in this example which makes it possible to map the road and its surrounding infrastructure even if the line markings are bad or missing. We consider that we have the 2D world coordinate frame shown as W whose origin is denoted by O_W. The state vector of the vehicle at least consists in the position and heading angle, i.e.,

$$\mathbf{x}_\mathrm{E} \triangleq \begin{bmatrix} \mathrm{x}_{EW}^{W} & \mathrm{y}_{EW}^{W} & \psi_E \end{bmatrix}^\mathrm{T} \tag{36}$$

with respect to the world coordinate frame W.

In our scenario, radar echoes typically stem from delineators or guardrails, which we would like to track as points or polynomials, respectively, with the framework presented in Section 5. We use the polynomial model (4) with $n = 2$ for our extended sources. This model has also been used for sequentially estimating similar parameters for the road's white lane markings in Lundquist and Schön (2010).

In the formulations of the previous sections, the extended sources are modeled as polynomials in the common x-y (world) coordinates of the problem. This can be considered as a significant limitation for the applications because in real scenes the curves to be tracked might not satisfy such polynomial equations in the common x-y coordinates of the problem. Here, this restriction is overcome by equipping each detected extended target with its own Cartesian x-y coordinates. Each extended object with state $\mathbf{x}_\mathrm{L}^{(j)}$ has an associated coordinate frame denoted by $L^{(j)}$. The position of the origin and the orientation of this coordinate frame is given as

$$\xi_\mathrm{L}^{(j)} \triangleq \begin{bmatrix} \mathrm{x}_{L^{(j)}W}^{W} & \mathrm{y}_{L^{(j)}W}^{W} & \psi_{L^{(j)}} \end{bmatrix}^\mathrm{T} \tag{37}$$

with respect to the world reference frame W where the superscripts are suppressed in the vector $\xi_{\mathrm{L}}^{(j)}$ for simplicity.

The measurements from the sensor mounted on the vehicle are obtained in polar coordinates (as in (2)) and need to be transformed first into the Cartesian coordinates of the sensor's frame E_s and then into the extended source's coordinate frame in order to be used. The measurement, expressed in the extended sources coordinate frame L in Cartesian coordinates is given by

$$\mathbf{z}^L = \mathcal{T}^{LE_s}(\mathbf{z}^{E_s}, \mathbf{x}_{\mathrm{E}}, \xi_{\mathrm{L}}), \tag{38}$$

where the transformation $\mathcal{T}^{LE_s}$ is described in the Appendix. Note that the measurement $\mathbf{z}^L$, which fits into the framework discussed in Section 4.1, is now not only affected by the original measurement $\mathbf{z}^{E_s}$, but also by the position of the ego vehicle, given by the state variable $\mathbf{x}_{\mathrm{E}}$, and the origin and orientation of the lanes coordinate frame ξ_{L}. We refer to the Appendix for details about the covariance calculation for $\mathbf{z}^L$ involving the transformation (38).

6.1 State Space Model

The state space model of the extended objects was thoroughly described in Section 4 and is not repeated here. The process model for the points on the form (6a) is linear and since the points are stationary, the process matrix F_{P} is the identity matrix. The polar measurements (2) are related to the Cartesian point states (3), i.e., $\mathbf{x}_{\mathrm{P}} = \begin{bmatrix} \mathsf{x}_{PW}^W & \mathsf{y}_{PW}^W \end{bmatrix}^{\mathrm{T}}$, through the measurement model according to

$$\bar{\mathbf{z}}_k = \begin{bmatrix} \sqrt{\left(\mathsf{x}_{PW}^W - \mathsf{x}_{E_sW}^W\right)^2 + \left(\mathsf{y}_{PW}^W - \mathsf{y}_{E_sW}^W\right)^2} \\ \arctan \frac{\mathsf{y}_{PW}^W}{\mathsf{x}_{PW}^W} - \psi_{E_s} \end{bmatrix} + \mathbf{e}_k, \tag{39}$$

where $\begin{bmatrix} \mathsf{x}_{E_sW}^W & \mathsf{y}_{E_sW}^W \end{bmatrix}^{\mathrm{T}}$ and ψ_{E_s} are the mounting position and orientation of the sensor. The measurement noise is assumed Gaussian $\mathbf{e} \sim \mathcal{N}(0, \Sigma_{\mathrm{p}})$.

6.2 Using Prior Information in Extended Track Generation

An extended track is initiated from tracked points under the assumption that a number of points form a line parallel to the road. In this section we are returning to the track handling discussion started in Section 5.2, in order to make the initialization procedure for extended sources more concrete. A prior information of the road's shape, or more specifically the lane's shape, in the vehicle coordinate frame is used for generating extended tracks from point tracks. The prior information is obtained by a lane tracking filter that estimates the parameters of the following lane polynomial

$$\mathsf{y}^E = l_E + \psi_{RE}\mathsf{x}^E + \frac{c_0}{2}(\mathsf{x}^E)^2, \tag{40}$$

where x^E and y^E are expressed in the ego vehicle's coordinate frame E. The lane tracker we use is described in Lundquist and Schön (2010) and it is based on a ego vehicle motion model which estimates the lane given measurements from a camera, an inertia sensor and the motion of other vehicles tracked by the radar. This implies that the camera easily can be removed and that the prior solely is based on the motion model of the ego vehicle and other moving or stationary objects tracked by the radar. The parameters in (40) can be interpreted as physical sizes. The angle between the longitudinal axis of the vehicle and the road lane is ψ_{RE}. It is assumed that this angle is small and hence the approximation $\sin \psi_{RE} \approx \psi_{RE}$ is used. The curvature parameter is denoted by c_0 and the offset between the ego vehicle and the white lane is denoted by l_E.

All point state estimates $\hat{\mathbf{x}}_{\mathrm{P}_i}$ are transformed into the ego vehicles coordinate frame since the priors's geometry (40) is given in this frame. We consider hypothetical parabolas passing through each point $\hat{\mathbf{x}}_{\mathrm{P}_k}$ parallel to the prior (40), i.e., the parameters ψ_{RE} and c_0 are just inherited from the lane tracker and the lateral distance l_{P_k} is given by

$$l_{\mathrm{P}_k} = \hat{\mathrm{y}}^E_{\mathrm{P}_k} - \psi_{RE}\hat{\mathrm{x}}^E_{\mathrm{P}_k} - \frac{c_0}{2}\left(\hat{\mathrm{x}}^E_{\mathrm{P}_k}\right)^2. \tag{41}$$

The likelihood $\ell_{\mathrm{P}_i\mathrm{P}_k}$ that a point $\mathbf{x}_{\mathrm{P}_i}$ is on the hypothetical line of point P_k is then given by

$$\ell_{\mathrm{P}_i\mathrm{P}_k} = \begin{cases} \mathcal{N}\left(\epsilon_{\mathrm{P}_i\mathrm{P}_k}; 0, P^E_{\mathrm{P}_k,(2,2)}\right), & \text{if } \hat{\mathbf{x}}^E_{\mathrm{P}_i} \in \mathcal{G}_{\mathrm{P}_k} \\ 0, & \text{otherwise,} \end{cases} \tag{42}$$

where the lateral distance between the point P_i and the proposed new line of point P_k is given by

$$\epsilon_{ik} = \hat{\mathrm{y}}^E_{\mathrm{P}_i} - \hat{\mathrm{y}}^E_{\mathrm{P}_k}, \tag{43}$$

with

$$\hat{\mathrm{y}}^E_{\mathrm{P}_k} = l_{\mathrm{P}_k} + \psi_{RE}\hat{\mathrm{x}}^E_{\mathrm{P}_i} + \frac{c_0}{2}\left(\hat{\mathrm{x}}^E_{\mathrm{P}_i}\right)^2. \tag{44}$$

The notation $P^E_{\mathrm{P}_k,(2,2)}$ refers to the lower-right element, i.e., the variance in the diagonal corresponding to y^E. The gate $\mathcal{G}_{\mathrm{P}_k}$ is defined as

$$\mathcal{G}_{\mathrm{P}_k} \triangleq \left\{ \begin{bmatrix} x \\ y \end{bmatrix} \middle| \frac{\left(y - \hat{y}^E_{\mathrm{P}_k}\right)^2}{P^E_{\mathrm{P}_k,(2,2)}} \leq \delta_{\mathrm{L}}, -\delta_s < x - \hat{x}^E_{\mathrm{P}_k} < \delta_e \right\}. \tag{45}$$

From all combinations of likelihoods we form a symmetric matrix $\Gamma_{\mathcal{I}}$. The columns of $\Gamma_{\mathcal{I}}$ are summed and the maximum value corresponding to column k_m is chosen. If this column contains more than a certain number κ of non-zero rows a parabola is formed from these points. The new line's states $\mathbf{x}_a$ are estimated by solving a least square problem using the corresponding points. The states $\mathbf{x}_{\mathrm{start}}$ and $\mathbf{x}_{\mathrm{end}}$ are the minimum and maximum x-coordinate value of the points, respectively. All elements in column k_m and rows i_m are set to zero and the procedure is repeated until no column contains more than κ non-zero elements.

6.3 Experiments and Results

Let us start by showing the information given by an ordinary automotive ACC radar, for the traffic situation shown in Figure 2a. The ego vehicle, indicated by a green circle, is situated at the (0, 0)-position in Figure 2b, and the red dots are the radar reflections, or stationary observations, at one time sample. The smaller magenta colored dots are former radar reflections, obtained at earlier time samples. Figure 2c shows the estimated points and lines for the same scenario using the KF EIV method presented in this contribution. The mean values of the states are indicated by solid black lines or blue points. Furthermore, the state variance, by means of the 90% confidence interval, is illustrated by gray lines or cyan colored ellipses, respectively. The estimate of the lane markings (40), illustrated by the gray dashed lines and derived according to the method presented in Lundquist and Schön (2010), is shown here as a comparison. We also show the tracked vehicle in front of the ego vehicle illustrated by a blue square.

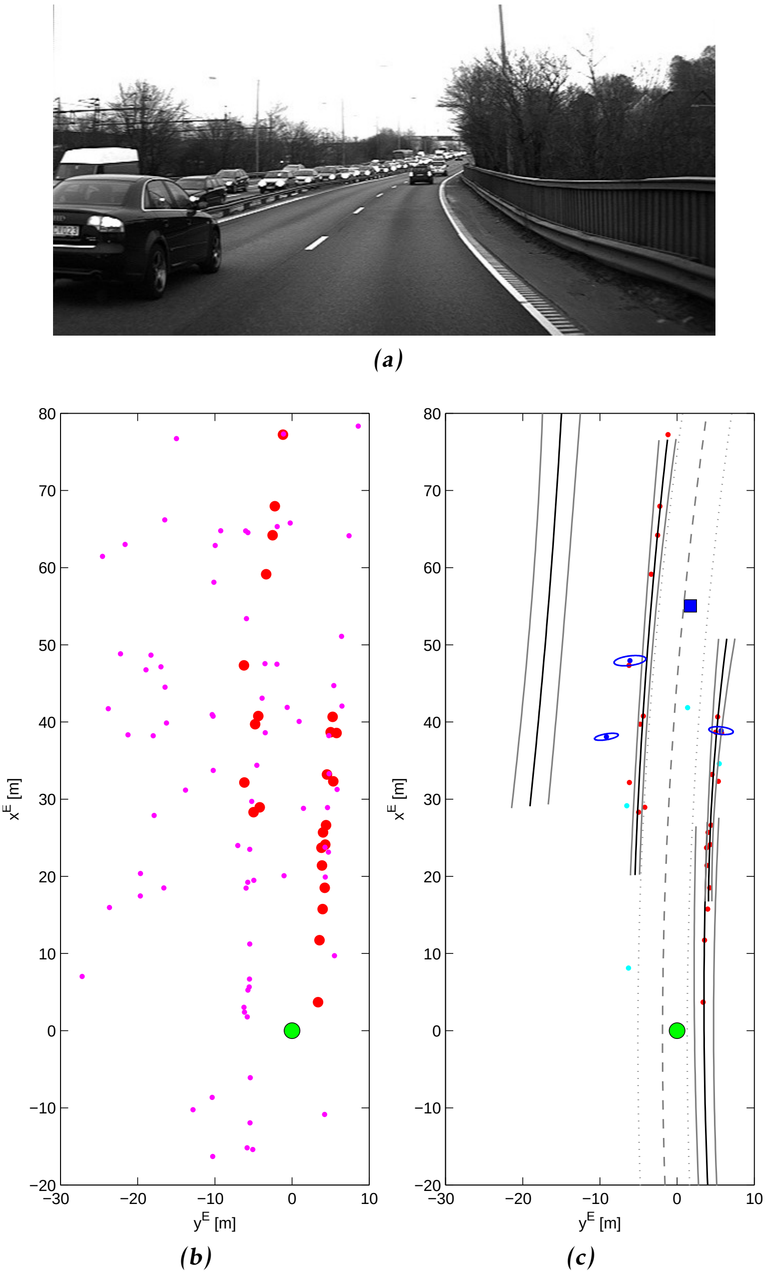
In Figure 3a we see a traffic scenario with a freeway exit. The corresponding bird's eye view is shown in Figure 3c. The rightmost line indicates the guardrail to the right of the exit, the middle line is the guardrail starting at the exit sign. The radar measurements are indicated with red dots, the confirmed point targets with blue dots and the unconfirmed points with cyan colored dots.

Our last example shows a situation from a rural road, see Figure 3d and 3b. Here the road edges are not as distinct as on the freeway and the amount of clutter is higher. This leads to parallel lines with larger covariance representing the guardrail, trees etc. Appearance of many lines may already be confusing and only the mean values of the lines are plotted for sake of clarity.

These results are only snapshots from different traffic situations. To show a more rigorous result of our approach the shape of the estimated lines is compared with the recorded driven trajectory from a 3 min drive on a freeway. The recorded positions 100 m in front of the car are used to estimate a polynomial with parameters a_1 and a_2 at each time step k. It is then compared with the estimated parameters $\hat{a}_1$ and $\hat{a}_2$ of the extended objects (lines) according to the mean absolute error

$$\text{MAE} = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{N_{\text{L},k}}\sum_{j=1}^{N_{\text{L},k}}\|\hat{a}_1^{(j)} - a_1\|\right) \tag{46}$$

for a_1 and similarly for a_2. The polynomials are transformed to be represented in the same coordinate frame. The ego vehicle is equipped with a vision system which measures the white lanes using computer vision and represents them according to (40). The vision based parameters ψ_{RE} and $c_0/2$ are also compared with the trajectory as in (46) and all results are shown in Table 2. Here, of course, the vision system produces better results since the driven trajectory should be parallel with the lane markings and the objects along the road, e.g., the guardrails, might not always be parallel. However, the table shows that our radar based approach is realistic and its performance is comparable to that of the vision based algorithm enabling the possibility of fusion with complementary information or

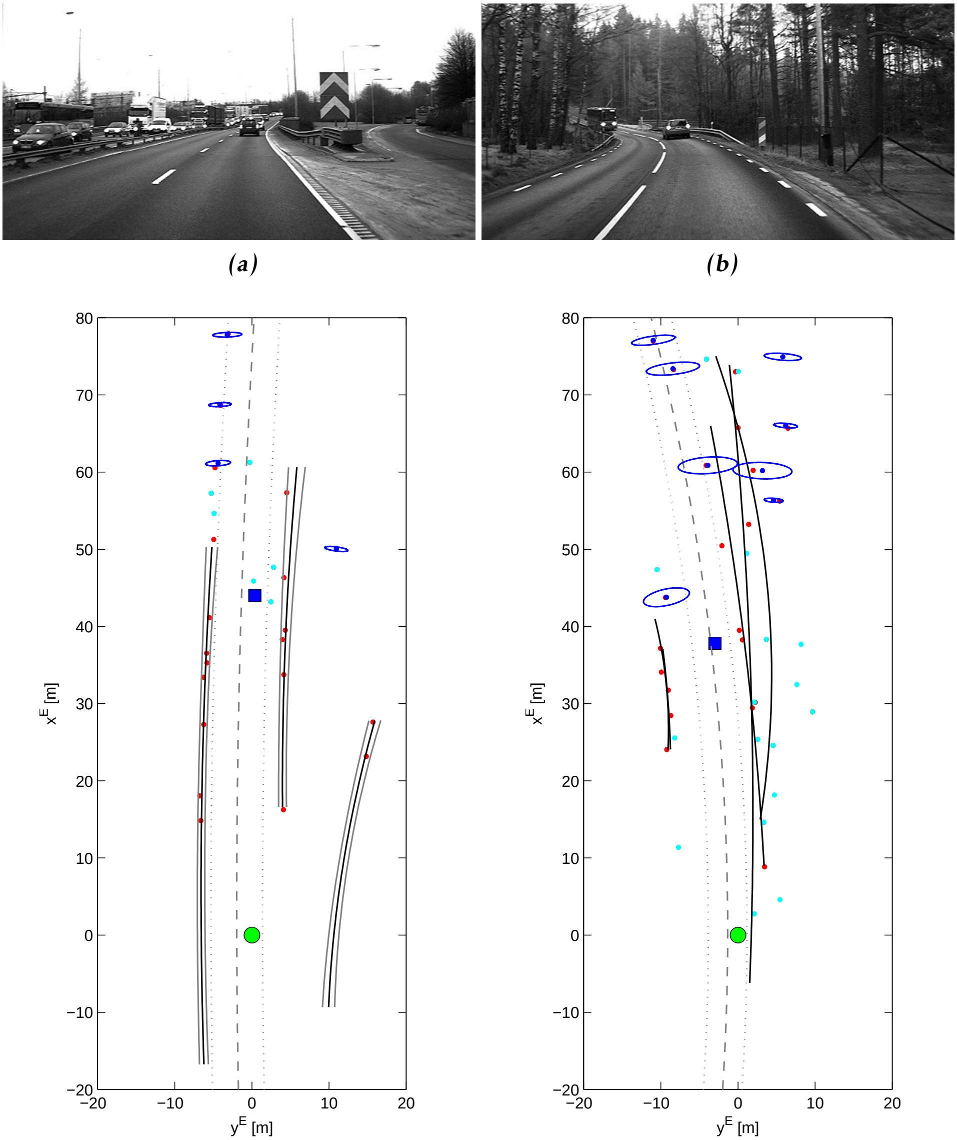


(a)

(b)

(c)

Figure 2: *A traffic situation is shown in Figure (a). Figure (b) shows the radar measurements, and Figure (c) the resulting tracked points and lines. The circle in the origin is the ego vehicle, the square is the tracked vehicle in front and the dashed gray lines illustrate the tracked road curvature.*



(a) *(b)*

(c) *(d)*

Figure 3: *Freeway exit with guardrails, the camera view is shown in Figure (a) and the bird's eye view with the estimated states in Figure (c). A traffic scenario from a rural road, with guardrails on both sides of a bridge is shown in Figure (d) and (b).*

Table 2: *MAE values for the estimated lines based on radar measurements and the lane estimate given by the computer vision, both with respect to the driven trajectory.*

Parameter	Radar	Vision
a_1 $(\cdot 10^{-3})$	8.22	6.93
a_2 $(\cdot 10^{-3})$	0.16	0.10

providing estimates when other approaches fail.

7 Conclusion

In this contribution we have considered the use of polynomials in extended target tracking. State space representations that enable the use of Kalman filters for the polynomial shaped extended objects are introduced. We have also described a multi-target tracking algorithm which is as general as possible. The loss of generality of the algorithm we presented lies in that the detection and initialization of the extended tracks from point tracks would always require some application dependent prior information. In the real world application example we have presented, we tried to be as illustrative as possible about the use of this prior information.

The approach has been evaluated on real and relevant data from both freeways and rural roads in Sweden. The results are not perfect, but surprisingly good at times, and of course contain much more information than just the raw measurements. Furthermore, the standard state representation of the objects should not be underestimated since it is compact and easy to send on a vehicle CAN-bus.

Some promising future investigations include the use of non-thresholded radar data in a track-before-detect type of curve estimation and better representation of curves in terms of intrinsic coordinates.

A Appendix

The need for a transformation from the sensor's frame E_s into the extended source's coordinate frame L was accentuated in (38) and described here in detail. The transformation $\mathcal{T}^{LE_s}(\mathbf{z}^{E_s}, \mathbf{x}_\mathrm{E}, \xi_\mathrm{L})$ is given by

$$\mathbf{z}^L = R^{LE}\left(R^{EE_s}\mathbf{z}^{E_s} + \boldsymbol{d}^E_{E_sE}\right) + R^{LW}(\boldsymbol{d}^W_{EW} - \boldsymbol{d}^W_{LW}), \tag{47}$$

where the displacement vector $\boldsymbol{d}^E_{E_sE}$ is the mounting position of the sensor in the vehicles coordinate frame, the rotation matrix R^{EE_s} describes the mounting orientation of the sensor in the vehicle and $\mathbf{z}^{E_s}$ is the Cartesian measurement given by the sensor.

The covariance is derived by first extracting the noise term from the stochastic variables according to

$$\hat{\mathbf{x}}_{\mathrm{E}} = \mathbf{x}_{\mathrm{E}} + \tilde{\mathbf{x}}_{\mathrm{E}}, \qquad \tilde{\mathbf{x}}_{\mathrm{E}} \sim \mathcal{N}(0, P_{\mathrm{E}}) \tag{48a}$$

$$\hat{\xi}_{\mathrm{L}} = \xi_{\mathrm{L}} + \tilde{\xi}_{\mathrm{L}}, \qquad \tilde{\xi}_{\mathrm{L}} \sim \mathcal{N}(0, P_{\xi_{\mathrm{L}}}) \tag{48b}$$

for the ego vehicle's state variable $\mathbf{x}_{\mathrm{E}}$ and the origin of the line's coordinate frame ξ_{L}. The noise term of the sensor measurement $\tilde{\mathbf{z}}^{E_s}$ was defined in (11). Equation (47) can now be rewritten according to

$$\begin{aligned}\mathbf{z}^{L} = {} & \widehat{R}^{LE}\left(R^{EE_s}\hat{\mathbf{z}}^{E_s} + \boldsymbol{d}^{E}_{E_sE}\right) + \widehat{R}^{LW}(\widehat{\boldsymbol{d}}^{W}_{EW} - \widehat{\boldsymbol{d}}^{W}_{LW}) \\ & + \widehat{R}^{LE}R^{EE_s}\tilde{\mathbf{z}}^{E_s} + A_2\tilde{\mathbf{x}}_{\mathrm{E}} + A_3\tilde{\xi}_{\mathrm{L}},\end{aligned} \tag{49a}$$

where the rotation matrices are given by

$$A_2 \triangleq \begin{bmatrix} \cos\hat{\psi}_L & \sin\hat{\psi}_L & -\mathrm{y}^{E_s}\cos\hat{\psi}_{EL} - \mathrm{x}^{E_s}\sin\hat{\psi}_{EL} \\ -\sin\hat{\psi}_L & \cos\hat{\psi}_L & \mathrm{x}^{E_s}\cos\hat{\psi}_{EL} - \mathrm{y}^{E_s}\sin\hat{\psi}_{EL} \end{bmatrix}, \tag{49b}$$

$$A_3 \triangleq \begin{bmatrix} -\cos\hat{\psi}_L & -\sin\hat{\psi}_L & a_3^{(1,3)} \\ \sin\hat{\psi}_L & -\cos\hat{\psi}_L & a_3^{(2,3)} \end{bmatrix}, \tag{49c}$$

and where

$$\begin{aligned}a_3^{(1,3)} &= \mathrm{y}^{E_s}\cos\hat{\psi}_{EL} + (\mathrm{y}^{W}_{EW} - \mathrm{y}^{W}_{LW})\cos\hat{\psi}_L \\ &\quad + \mathrm{x}^{E_s}\sin\hat{\psi}_{EL} - (\mathrm{x}^{W}_{EW} - \mathrm{x}^{W}_{LW})\sin\hat{\psi}_L,\end{aligned} \tag{49d}$$

$$\begin{aligned}a_3^{(2,3)} &= -\mathrm{x}^{E_s}\cos\hat{\psi}_{EL} - (\mathrm{x}^{W}_{EW} - \mathrm{x}^{W}_{LW})\cos\hat{\psi}_L \\ &\quad + \mathrm{y}^{E_s}\sin\hat{\psi}_{EL} - (\mathrm{y}^{W}_{EW} - \mathrm{y}^{W}_{LW})\sin\hat{\psi}_L.\end{aligned} \tag{49e}$$

To summarize these calculations the covariance of $\mathbf{z}^{L}$ is given by

$$\Sigma_{\mathrm{c}}^{L} = \widehat{R}^{LE_s}\Sigma_{\mathrm{c}}^{E_s}(\widehat{R}^{LE_s})^{\mathrm{T}} + A_2 P_{\mathrm{E}} A_2^{\mathrm{T}} + A_3 P_{\xi_{\mathrm{L}}} A_3^{\mathrm{T}} \tag{50}$$

where $\Sigma_{\mathrm{c}}^{E_s}$ is the Cartesian measurement covariance, P_{E} is the state covariance of the ego vehicles pose and $P_{\xi_{\mathrm{L}}}$ is the line state covariance corresponding to the position and orientation of the line's coordinate frame.

Bibliography

D. Angelova and L. Mihaylova. Extended object tracking using Monte Carlo methods. *IEEE Transactions on Signal Processing*, 56(2):825–832, February 2008.

Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Mathematics in science and engineering. Academic Press, Orlando, FL, USA, 1988.

Y. Bar-Shalom and X. R. Li. *Multitarget-Multisensor Tracking: Principles, Techniques*. YBS Publishing, Storrs, CT, USA, 1995.

R. H. Bartels, J. C. Beatty, and B. A. Barsky. *An introduction to splines for use in computer graphics and geometric modeling*. M. Kaufmann Publishers, Los Altos, CA, USA, 1987. ISBN 0-934613-27-3 ;.

Å Björck. *Numerical methods for least squares problems*. SIAM, Philadelphia, PA, USA, 1996.

S. S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, Norwood, MA, USA, 1999.

A. Blake and M. Isard. *Active contours*. Springer, London, UK, 1998.

Y. Boers, H. Driessen, J. Torstensson, M. Trieb, R. Karlsson, and F. Gustafsson. Track-before-detect algorithm for tracking extended targets. *IEE Proceedings of Radar, Sonar and Navigation*, 153(4):345–351, August 2006.

D. L. Boley and K. Sutherland. Recursive total least squares: An alternative to the discrete Kalman filter. In *AMS Meeting on Applied Linear Algebra*, DeKalb, IL, USA, May 1993.

B. De Moor. Structured total least-squares and L_2 approximation-problems. *Linear algebra and its applications*, 188–189:163–207, 1993.

J. C. Dezert. Tracking maneuvering and bending extended target in cluttered environment. In *Proceedings of Signal and Data Processing of Small Targets*, volume 3373, pages 283–294, Orlando, FL, USA, April 1998. SPIE.

R. Diversi, R. Guidorzi, and U. Soverini. Algorithms for optimal errors-in-variables filtering. *Systems & Control Letters*, 48(1):1–13, 2003.

R. Diversi, R. Guidorzi, and U. Soverini. Kalman filtering in extended noise environments. *IEEE Transactions on Automatic Control*, 50(9):1396–1402, September 2005.

O. E. Drummond, S. S. Blackman, and G. C. Pretrisor. Tracking clusters and extended objects with multiple sensors. In *Proceedings of Signal and Data Processing of Small Targets*, volume 1305, pages 362–375, Orlando, FL, USA, January 1990. SPIE.

B. Fardi, U. Scheunert, H. Cramer, and G. Wanielik. Multi-modal detection and parameter-based tracking of road borders with a laser scanner. In *Proceedings* *of the IEEE Intelligent Vehicles Symposium*, pages 95–99, Columbus, OH, USA, June 2003.

K. Gilholm and D. Salmond. Spatial distribution model for tracking extended objects. *IEE Proceedings of Radar, Sonar and Navigation*, 152(5):364–371, October 2005.

K. Gilholm, S. Godsill, S. Maskell, and D. Salmond. Poisson models for extended target and group tracking. In *Proceedings of Signal and Data Processing of Small Targets*, volume 5913, pages 230–241, San Diego, CA, USA, August 2005. SPIE.

R. Guidorzi, R. Diversi, and U. Soverini. Optimal errors-in-variables filtering. *Automatica*, 39(2):281–289, 2003. ISSN 0005-1098.

F. Gustafsson. Automotive safety systems. *IEEE Signal Processing Magazine*, 26 (4):32–47, July 2009.

E. Hellström. *Look-ahead Control of Heavy Vehicles*. PhD thesis No 1315, Linköping Studies in Science and Technology, Linköping, Sweden, May 2010.

P. J. Huber and E. Ronchetti. *Robust Statistics*. John Wiley & Sons, New York, NY, USA, 2009.

M. Isard and A. Blake. CONDENSATION-conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, August 1998.

R. Karlsson and F. Gustafsson. Bayesian surface and underwater navigation. *IEEE Transactions on Signal Processing*, 54(11):4204–4213, November 2006.

M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, January 1988.

A. Kirchner and T. Heinrich. Model based detection of road boundaries with a laser scanner. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 93–98, Stuttgart, Germany, October 1998.

K. R. S. Kodagoda, W. S. Wijesoma, and A. P. Balasuriya. CuTE: curb tracking and estimation. *IEEE Transactions on Control Systems Technology*, 14(5):951–957, September 2006.

S. Lazarus, A. Tsourdos, B.A. White, P.M.G. Silson, and R. Zandbikowski. Airborne vehicle mapping of curvilinear objects using 2-D splinegon. *IEEE Transactions on Instrumentation and Measurement*, 59(7):1941–1954, July 2010.

L. Ljung. *System identification, Theory for the user*. System sciences series. Prentice Hall, Upper Saddle River, NJ, USA, 2 edition, 1999.

C. Lundquist and T. B. Schön. Estimation of the free space in front of a moving vehicle. In *Proceedings of the SAE World Congress*, SAE paper 2009-01-1288, Detroit, MI, USA, April 2009.

C. Lundquist and T. B. Schön. Joint ego-motion and road geometry estimation. *Information Fusion*, 2010. DOI: 10.1016/j.inffus.2010.06.007.

C. Lundquist, U. Orguner, and T. B. Schön. Tracking stationary extended objects for road mapping using radar measurements. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 405–410, Xi'an, China, June 2009.

C. Lundquist, U. Orguner, and F Gustafsson. Estimating polynomial structures from radar data. In *Proceedings of the International Conference on Information Fusion*, Edinburgh, UK, July 2010.

C. Lundquist, U. Orguner, and F. Gustafsson. Extended target tracking using polynomials with applications to road-map estimation. *IEEE Transactions on Signal Processing*, 59(1):15–26, January 2011.

R. P. S. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, Boston, MA, USA, 2007.

I. Markovsky and S. van Huffel. Overview of total least-squares methods. *Signal Processing*, 87(10):2283–2302, 2007. ISSN 0165-1684.

A. Polychronopoulos, A. Amditis, N. Floudas, and H. Lind. Integrated object and road border tracking using 77 GHz automotive radars. *IEE Proceedings of Radar, Sonar and Navigation*, 151:375–381, December 2004.

D. B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.

B. Ristic and D. J. Salmond. A study of a nonlinear filtering problem for tracking an extended target. In *Proceedings of the International Conference on Information Fusion*, pages 503–509, Stockholm, Sweden, June 2004.

B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle filters for tracking applications*. Artech House, London, UK, 2004.

B. Roorda and C. Heij. Global total least squares modeling of multivariable time series. *IEEE Transactions on Automatic Control*, 40(1):50–63, January 1995.

D. J. Salmond and M. C. Parr. Track maintenance using measurements of target extent. *IEE Proceedings of Radar, Sonar and Navigation*, 150(6):389–395, December 2003.

T. Söderström. Survey paper: Errors-in-variables methods in system identification. *Automatica*, 43(6):939–958, June 2007.

H. Svensson. Simultaneous localization and mapping in marine environment using radar images. Master's Thesis No LiTH-ISY-EX-4285, Department of Electrical Engineering, Linköping University, Sweden, 2009.

S. van Huffel and H. Zha. An efficient Total Least Squares algorithm based on a rank-revealing two-sided orthogonal decomposition. *Numerical Algorithms*, 4: 101–133, February 1993.

H. L. van Trees. *Detection, Estimation, and Modulation Theory.* John Wiley & Sons, New York, NY, USA, 1968.

J. Vermaak, N. Ikoma, and S. J. Godsill. Sequential Monte Carlo framework for extended object tracking. *IEE Proceedings of Radar, Sonar and Navigation*, 152 (5):353–363, October 2005.

Y. Wang, L. Bai, and M. Fairhurst. Robust road modeling and tracking using condensation. *IEEE Transactions on Intelligent Transportation Systems*, 9(4): 570–579, December 2008.

M. J. Waxman and O. E. Drummond. A bibliography of cluster (group) tracking. In *Proceedings of Signal and Data Processing of Small Targets*, volume 5428, pages 551–560, Orlando, FL, USA, April 2004. SPIE.

W. S. Wijesoma, K. R. S. Kodagoda, and A. P. Balasuriya. Road-boundary detection and tracking using ladar sensing. *IEEE Transactions on Robotics and Automation*, 20(3):456–464, June 2004.

Paper D

Road Intensity Based Mapping using Radar Measurements with a Probability Hypothesis Density Filter

Authors: Christian Lundquist, Lars Hammarstrand and Fredrik Gustafsson

Edited version of the paper:

C. Lundquist, L. Hammarstrand, and F. Gustafsson. Road intensity based mapping using radar measurements with a probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 59(4):1397–1408, April 2011b.

The paper presents data that was previously published in:

C. Lundquist, L. Danielsson, and F. Gustafsson. Random set based road mapping using radar measurements. In *Proceedings of the European Signal Processing Conference*, pages 219–223, Aalborg, Denmark, August 2010a.

Preliminary version:

Technical Report LiTH-ISY-R-2994, Dept. of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden.

Road Intensity Based Mapping using Radar Measurements with a Probability Hypothesis Density Filter

Christian Lundquist[*], Lars Hammarstrand[**] and Fredrik Gustafsson[*]

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

[**]Volvo Car Corporation
Active Safety Electronics
SE-405 31 Göteborg, Sweden
lhammar5@volvocars.com

Abstract

Mapping stationary objects is essential for autonomous vehicles and many autonomous functions in vehicles. In this contribution the probability hypothesis density (PHD) filter framework is applied to automotive imagery sensor data for constructing such a map, where the main advantages are that it avoids the detection, the data association and the track handling problems in conventional multiple-target tracking, and that it gives a parsimonious representation of the map in contrast to grid based methods. Two original contributions address the inherent complexity issues of the algorithm: First, a data clustering algorithm is suggested to group the components of the PHD into different clusters, which structures the description of the prior and considerably improves the measurement update in the PHD filter. Second, a merging step is proposed to simplify the map representation in the PHD filter. The algorithm is applied to multi-sensor radar data collected on public roads, and the resulting map is shown to well describe the environment as a human perceives it.

1 Introduction

Autonomous vehicles and autonomous functionality in vehicles require situation awareness. Situation awareness is traditionally split into two main tasks: tracking of moving objects and navigation relative to stationary objects represented by a map. The precision in commercial maps and position services today is not sufficient for autonomous functions. For that reason, imagery sensors as radar, vision and laser scanners have been used to build local maps of stationary objects on the fly, which requires sophisticated estimation algorithms. Similarly to the classic target tracking application, going from single-sensor single-object estimation to multi-sensor multi-object estimation increases the complexity of the algorithms substantially. For that reason, there is a strong need for structured maps rather than point maps, where single objects that together form structures as lines or curves are described as one extended object. This contribution describes an efficient algorithm to create maps that are particularly useful to describe structures along road sides. Such a map is promising for a range of applications, from collision avoidance maneuvers to true autonomy on public roads and off-road.

There exist many algorithms and representations of maps using imagery sensor data. However, so far methods based on probabilistic theory are most successful, see e.g., Thrun et al. (2005) or Adams et al. (2007) for a recent overview. Feature based maps have become very popular in the related field simultaneous localization and mapping (SLAM) Durrant-Whyte and Bailey (2006); Bailey and Durrant-Whyte (2006), where a vehicle builds a map of environmental features, while simultaneously using that map to localize itself. Location based maps, such as the occupancy grid map (OGM) Elfes (1987), were primarily developed to be used with laser scanners to generate consistent maps. They have also been used with radar sensors, where the data comprises a complete signal power profile Mullane et al. (2009); Foessel-Bunting et al. (2001), or a thresholded radar report Lundquist et al. (2009). The OGM was very popular at the DARPA urban challenge, see the special issues Buehler et al. (2008). Generally, line shaped and curved objects, such as roads and coast lines can be identified in topographic maps. Road edge and obstacle detection have been tackled using raw radar images, see e.g., Kaliyaperumal et al. (2001); Lakshmanan et al. (1997); Nikolova and Hero (2000); Ma et al. (2000). There have been several approaches making use of reflections from the road edges, such as guard rails and reflection posts, to compute information about the free space, see e.g. Kirchner and Heinrich (1998); Kirchner and Ameling (2000); Sparbert et al. (2001) for some examples using laser scanners. Radar reflections were used to track the road edges as extended targets in Lundquist et al. (2011b). This method is promising but a drawback is the large data association problem, which arises since it is hard to create a general model for the various types of objects.

The bin occupancy filter is devised via a quantized state space model, where each cell is denoted bin Erdinc et al. (2009). In the limit, when the volume of the bins become infinitesimal, the filter equations are identical to the probability hypothesis density (PHD) filter, proposed by Mahler Mahler (2000, 2003, 2007). The PHD is the first moment density or intensity density of a random set. In Section 2 we propose to represent the map by the surface described by the intensity density of point sources of stationary objects. Individual sources are not tracked, but the PHD filter framework makes it possible to estimate the intensity of sources in a given point. To describe the map by a continuous surface distinguishes our method from the other mapping approaches mentioned above. An approach to make use of the PHD filter to solve the related SLAM problem is presented in Mullane et al. (2008). In contrast to the intensity based map, the map in Mullane et al. (2008) is represented by a finite set of map features, which are extracted from the intensity density.

Section 3 summarizes the Gaussian mixture PHD (GM-PHD) filter recursion. We investigate the possibility to improve the efficiency of the GM-PHD filter by utilizing structure in the map. These structures are first identified and modeled in Section 4, to then be used to improve the prior in the GM-PHD recursion and to create an efficient representation of the intensity in Section 5. Road edges and guardrails are typical examples of such map structures. All parts of the road mapping example are put together in Section 6 to present the complete map, based on measurements from automotive radar sensors. The map may be used by automotive safety functions, for instance trajectory control, which aims at minimizing the probability to hit objects. Other examples are functions which can perform automatic evasive maneuvers, and need map information for threat assessment Eidehall et al. (2007b). The conclusions are in Section 7.

2 Mapping

Sensors which measure range r and bearing ψ, e.g., radar and laser, are commonly used for automotive and robotics navigation and mapping. These type of sensors typically do not give the user access to their detection and decision parameters. Automotive radar sensors threshold the amplitude and deliver only those detections with the highest amplitude, whereas a laser sensor delivers the first range detection along a bearing angle. Furthermore, cameras measure two angles and represent them in a pixel matrix. The considered sensors provide a set of noisy measurements (thresholded detections)

$$\mathbf{Z}_k = \left\{\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(2)}, \ldots, \mathbf{z}_k^{(N_{\mathbf{z},k})}\right\} \tag{1}$$

at each discrete time instant $k = 1, \ldots, K$.

Landmark detections from the noisy sensor data are used to build a probabilistic map, represented by a set of $N_{\mathbf{m}}$ objects in the environment

$$\mathbf{M}_k = \left\{\mathbf{m}_k^{(1)}, \mathbf{m}_k^{(2)}, \ldots, \mathbf{m}_k^{(N_{\mathbf{m}})}\right\}. \tag{2}$$

There exists primarily two types of indexing for probabilistic maps Thrun et al. (2005). In a *feature based map* each $\mathbf{m}^{(n)}$ specifies the properties and location of one object Smith et al. (1988), whereas in a *location based map* the index n corresponds to a location and $\mathbf{m}^{(n)}$ is the property of that specific coordinate. Note that the indexing is analogous to the representation of sensor data, the range and bearing measurements from a radar or laser are feature based, whereas the pixels of a camera are location based measurements. The *Occupancy grid map* is a classical location based representation of a map, where each cell of the grid is assigned a binary occupancy value that specifies if the location n is occupied ($\mathbf{m}^{(n)} = 1$) or not ($\mathbf{m}^{(n)} = 0$) Elfes (1987); Moravec (1988). The aim of all stochastic mapping algorithms, independent of indexing, is to estimate the posterior density of the map

$$p(\mathbf{M}_k|\mathbf{Z}_{1:k}), \tag{3}$$

given all the measurements from time 1 to k.

The bin-occupancy filter, which is described in Erdinc et al. (2009), aims to estimate the probability of a target being in a given point. The approach is derived via a discretized state-space model of the surveillance region, where each grid cell (denoted bin in this approach) can or may not contain a target. One of the important assumptions in Erdinc et al. (2009) is that the bins are sufficiently small so that each bin is occupied by maximum one target. In the limiting case, when the volume of the bins $|\nu|$ goes to zero, it is possible to define the *bin-occupancy density*

$$D_{k|k} \triangleq \lim_{|\nu| \to 0} \frac{\Pr(\mathbf{m}_k^{(n)} = 1 | \mathbf{Z}_{1:k})}{|\nu|}, \tag{4}$$

where $\Pr(\mathbf{m}_k^{(n)} = 1|\mathbf{Z}_{1:k})$ is the probability that bin n is occupied by one target. The continuous form of the bin-occupancy filter prediction and update equations are the same as the PHD filter equations Erdinc et al. (2009). Furthermore, the PHD is the first moment density or *intensity density* in point process theory, see e.g., Mahler (2000, 2007), and a physical interpretation is given in Daley and Vere-Jones (2003) as the probability that one target is located in the infinitesimal region $(\mathbf{x},\mathbf{x} + d\mathbf{x})$ of the state space, divided by $d\mathbf{x}$. The continuous form of the physical bin model leads us to a continuous location based map which we denote *intensity based map*, and we intend to estimate it with the PHD filter.

The bin occupancy filter and the PHD filter were developed for target tracking of point sources; however, the aim of the present contribution is to create a probabilistic location based map of the surroundings of a moving vehicle. One of the main differences between standard target tracking problems and the building of a location based map, is that many objects such as, guardrails or walls, are typically not point targets, but extended targets Mahler (2007); Gilholm et al. (2005); Gilholm and Salmond (2005). Furthermore, there is no interest in keeping track of the identity of specific objects. Nevertheless, the bin-occupancy filter attempts to answer the important question: “Is there an object (target) at a given point?". Erdinc et al. Erdinc et al. (2009) pose the following assumptions for the bin occupancy filter:

1. The bins are sufficiently small so that each bin is occupied by at most one target.
2. One target gives rise to only one measurement.
3. Each target generates measurements independently.
4. False alarms are independent of target originated measurements.
5. False alarms are Poisson distributed.

Here, only point 2 needs some extra treatment if the aim of the algorithm is mapping and not target tracking. It can be argued that the measurements of the point sources belong to extended objects and that the aim is to create a map of those point sources. Also for mapping cases the assumption that there will not be two measurements from the same point at the same time is justified. The described relation is modeled by a likelihood function $p(\mathbf{Z}_k|\mathbf{M}_{k|k})$, which maps the Cartesian map to polar point measurements.

So far in this section the discussion has been quite general and the PHD or the intensity has only been considered as a surface over the surveillance region. The first practical algorithms to realize the PHD filter prediction and measurement update equations were based on the particle filter, see e.g., Vo et al. (2003); Zajic and Mahler (2003); Sidenbladh (2003), where the PHD is approximated by a large set of random samples (particles). A Gaussian mixture approximation of the PHD (GM-PHD) was proposed by Vo and Ma Vo and Ma (2006). The mixture is represented by a sum of weighted Gaussian components and in particular the mean and covariance of the components are propagated by the Kalman filter. In this work the intensity is represented by a Gaussian mixture, since the parametrization and derivation is simpler than for a particle filter based solution. The modeling of the intensity through a number of Gaussian components makes it also simpler to account for structures in the map. We will return to these structures in the next two sections.

The GM-PHD filter estimates the posterior intensity, denoted $D_{k|k}$, as a mixture of Gaussian densities as,

$$D_{k|k} = \sum_{i=1}^{J_{k|k}} w_{k|k}^{(i)} \mathcal{N}\left(m_{k|k}^{(i)}, P_{k|k}^{(i)}\right), \tag{5}$$

where $J_{k|k}$ is the number of Gaussian components and $w_{k|k}^{(i)}$ is the expected number of point sources covered by the density $\mathcal{N}(m_{k|k}^{(i)}, P_{k|k}^{(i)})$. In Section 3 it is shown how the intensity is estimated with the GM-PHD filter. The Gaussian components are parametrized by a mean $m_{k|k}^{(i)}$ and a covariance value $P_{k|k}^{(i)}$, which are expressed in a planar Cartesian coordinate frame, according to

$$m_k^{(i)} = \begin{bmatrix} \mathsf{x}_k^{(i)} & \mathsf{y}_k^{(i)} \end{bmatrix}^{\mathrm{T}}. \tag{6}$$

The aim of the mapping algorithm is to estimate the posterior density (3). The considered intensity based map is continuous over the surveillance region, thus, the volume of the bins becomes infinitesimal, and the number of elements in (2) tends to infinity ($N_{\mathbf{m}} \rightarrow \infty$). Furthermore, the intensity is a summary statistic of the map according to

$$p(\mathbf{M}_k|\mathbf{Z}_{1:k}) \sim p(\mathbf{M}_k; D_{k|k}), \tag{7}$$

see e.g., Mahler (2003), and the estimated intensity $D_{k|k}$ is parametrized by

$$\mu_k^{(i)} \triangleq \left\{ w_k^{(i)}, m_k^{(i)}, P_k^{(i)} \right\} \tag{8}$$

of the Gaussian sum (5). The intensity based map is a multimodal surface with

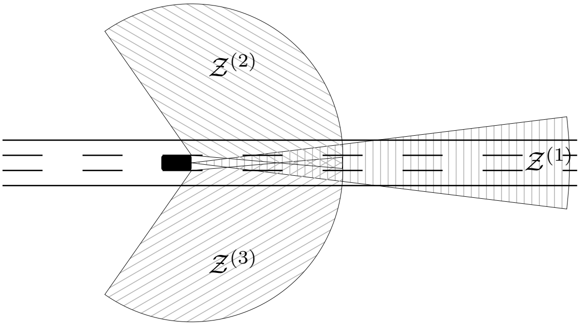


Figure 1: *The observation space of the forward looking radar is denoted $\mathcal{Z}^{(1)}$ and the observation spaces of the side-looking radars are $\mathcal{Z}^{(2)}$ and $\mathcal{Z}^{(3)}$.*

peaks around areas with many sensor reflections or point sources. It is worth observing that the map $\mathbf{M}$ is described by a location based function (5), with feature based parametrization (8). The intensity is a distribution and the graphical representation provides an intuitive description of the map.

An automotive mapping example will be studied to exemplify the methods described in each section. In this example, an intensity based map is constructed of the infrastructure surrounding an ego vehicle equipped with three radar sensors. The aim is to represent stationary radar reflectors, i.e. reflection points on road side objects, such as, guard rails or lampposts, using these radar measurements. Each sensor ($s = 1, 2, 3$) collects a set of individual observations $m = 1, \ldots, N_{\mathbf{z},k}^{(s)}$ of range $r^{(m)}$, range rate $\dot{r}^{(m)}$ and bearing angle $\psi^{(m)}$ to strong radar reflectors according to

$$\left\{\mathbf{z}_k^{(m)} = \begin{bmatrix} r^{(m)} & \dot{r}^{(m)} & \psi^{(m)} \end{bmatrix}_k^{\mathrm{T}}\right\}_{m=1}^{N_{\mathbf{z},k}^{(s)}}. \tag{9}$$

The three sensors are one forward looking 77 GHz mechanically scanning FMCW radar, with measurement space $\mathcal{Z}^{(1)}$, and two medium distance 24 GHz radars, with measurement spaces $\mathcal{Z}^{(2)}$ and $\mathcal{Z}^{(3)}$, which are mounted on the vehicle as illustrated in Figure 1. The radars constitute a multi-sensor system, measuring at different ranges and in different directions, but with an overlap of observation spaces. One of the most significant advantages with the PHD filter is that this framework scales very well for multi-sensor multi-target problem, and the classically hard sensor fusion task is solved more or less automatically. A coordinate frame E is attached to the moving ego vehicle, see Figure 3. The trajectory $\mathbf{x}_{1:k} = \mathbf{x}_1, \ldots, \mathbf{x}_k$ of the ego vehicle is assumed to be *a priori* known in this work, and it is defined through the sequence of all its positions, orientations and velocities relative the world coordinate frame W up to time k.

Note, that the map is parametrized in a very compact format (8), and that the parameters easily can be sent over the CAN-bus to decision systems in a car.

3 GM-PHD Filter

The PHD filter recursion is described in e.g., Mahler (2003), the bin occupancy filter is described in Erdinc et al. (2009) and the GM-PHD filter recursion is described in Vo and Ma (2006). This section intents to summarize the main algorithmic concept of those contributions and apply the filter to the mapping application. The mentioned filters are primarily developed for multiple-target tracking, where, at least for the PHD filter, the targets are defined as a random finite set, and where the extraction of the most likely target states at the end of each recursion cycle is an important output. However, the output for the proposed mapping is the intensity $D_{k|k}$, which describes the density of sensor reflections or point sources. The filter recursion is given by the time update in Section 3.1 and the measurement update in Section 3.2 below.

3.1 Time Evolution

Consider the bin occupancy model, a bin can be occupied at time $k+1$, if it already was occupied at time k, or if a new point source appears in the map at time $k+1$. A new source can appear either by spontaneous birth, as a completely new object, or by spawning from an existing object.

In the mapping application the following definitions are made: If a bin turns occupied independent of other bins' occupancy it is denoted *birth*, whereas if a bin turns occupied dependent on the occupancy of other bins it is denoted *spawn*. The latter is for example the case if radar reflections stem from an extended object, such as a guardrail, and new point sources appear on other positions on the same object. The prediction of the intensities is given by the sum of the existing intensity $D_{s,k+1|k}$, the spawn intensity $D_{\beta,k+1|k}$ and the birth intensity $D_{b,k+1}$ according to

$$D_{k+1|k} = D_{s,k+1|k} + D_{\beta,k+1|k} + D_{b,k+1}. \tag{10}$$

The prediction for the existing intensity is given by

$$D_{s,k+1|k} = p_{S,k+1} \sum_{i=1}^{J_{k|k}} w_{k|k}^{(i)} \, \mathcal{N}\left(m_{k+1|k}^{(i)}, P_{k+1|k}^{(i)}\right), \tag{11}$$

where the Gaussian components $\mathcal{N}(m_{k+1|k}^{(i)}, P_{k+1|k}^{(i)})$ are derived using the time update step of, e.g., the Kalman filter. The probability of survival is denoted p_S. The spawn intensity is a Gaussian mixture of the form

$$D_{\beta,k+1|k} = \sum_{j=1}^{J_{\beta,k+1}} w_{\beta,k+1|k}^{(j)} \mathcal{N}\left(m_{\beta,k+1|k}^{(j)}, P_{\beta,k+1|k}^{(j)}\right), \tag{12}$$

where $w_{\beta,k+1|k}^{(j)}, m_{\beta,k+1|k}^{(j)}$ and $P_{\beta,k+1|k}^{(j)}$ for $j = 1, \dots, J_{\beta,k+1}$, are model parameters that determine the shape of the spawn intensity. In the original Gaussian mixture PHD filter Vo and Ma (2006) the components $\mathcal{N}(m_{\beta,k+1|k}^{(j)}, P_{\beta,k+1|k}^{(j)})$ are in the proximity of its parents. However, in the mapping application the existing stationary point sources are likely to belong to extended object, which describe some structure in the map, from which new point sources are likely to be spawned. Typical examples of such structures are guard rails in automotive examples or coast lines in terrain mapping. An approach is proposed where the components are derived using a model-based process, which takes the existing components $\mathcal{N}(m_{k|k}^{(i)}, P_{k|k}^{(i)})$ at time step k as input. A method, which clusters the components and estimates the model parameters simultaneously is described in Section 4. The Gaussian mixture components $\mathcal{N}(m_{\beta,k+1|k}^{(j)}, P_{\beta,k+1|k}^{(j)})$ are then sampled from the estimated models.

The birth intensity is assumed to be a Gaussian mixture of the form

$$D_{b,k+1} = \sum_{j=1}^{J_{b,k+1}} w_{b,k+1}^{(j)} \, \mathcal{N}\left(m_{b,k+1}^{(j)}, P_{b,k+1}^{(j)}\right), \tag{13}$$

where $w_{b,k+1}^{(j)}$, $m_{b,k+1}^{(j)}$ and $P_{b,k+1}^{(j)}$ for $j = 1, \dots, J_{b,k+1}$, are model parameters that determine the shape of the birth intensity. The components $\mathcal{N}(m_{b,k+1}^{(j)}, P_{b,k+1}^{(j)})$ are uniformly distributed over the state space.

3.2 Measurement Update

The measurement update is given by a sum of intensities according to

$$D_{k|k} = (1 - p_{D,k})D_{k|k-1} + \sum_{\mathbf{z}\in\mathbf{Z}_k} D_{d,k|k}, \tag{14}$$

where p_D is the probability of detection. The updated intensity $D_{d,k|k}$ is given by

$$D_{d,k|k} = \sum_{i=1}^{J_{k|k-1}} w_k^{(i)}(\mathbf{z})\mathcal{N}\left(m_{k|k}^{(i)}(\mathbf{z}), P_{k|k}^{(i)}(\mathbf{z})\right), \tag{15}$$

where the the number of Gaussian components is $J_{k|k-1} = J_{k-1|k-1} + J_{\beta,k|k-1} + J_{b,k}$. The Gaussian components $\mathcal{N}(m_{k|k}^{(i)}(\mathbf{z}), P_{k|k}^{(i)}(\mathbf{z}))$ are calculated by using the measurement update step of, e.g., the Kalman filter. Furthermore, the weights are updated according to

$$w_k^{(i)}(\mathbf{z}) = \frac{p_{D,k} w_{k|k-1}^{(i)} q_k^{(i)}(\mathbf{z})}{\kappa_k + p_{D,k} \sum_{\ell=1}^{J_{k|k-1}} w_{k|k-1}^{(\ell)} q_k^{(\ell)}(\mathbf{z})}, \tag{16a}$$

$$q^{(i)}(\mathbf{z}) = \mathcal{N}\left(\mathbf{z}; \eta_{k|k-1}^{(i)}, S_{k|k-1}^{(i)}\right), \tag{16b}$$

where $\eta_{k|k-1}^{(i)}$ is the predicted measurement and $S_{k|k-1}^{(i)}$ is the innovation covariance from the i^{th} component in the predicted intensity. The clutter intensity at time k is denoted κ_k, and the clutter measurements are assumed to be Poisson distributed.

4 Joint Clustering and Estimation

The spawning process was briefly introduced in Section 3.1. Provided that there exists a certain structure in between the single point objects in the map, then this structure can be used to improve the spawning process. This section describes how the map structure can be found, modeled and estimated. We propose a regression clustering algorithm, which alternates between clustering the Gaussian components and estimating the model parameters, using a simple weighted least squares method. The standard K-means cluster algorithm is first described in Section 4.1 followed by the joint clustering and estimation algorithm in Section 4.2. Road edges are typical structures in a map, and it is shown in Section 4.3 how to apply the joint clustering and estimation algorithm to estimate these.

4.1 K-Means Clustering

Given a set of $N_{\mathbf{y}}$ observations $\mathbf{y}$, then the clustering algorithm aims to find a partition $\mathbf{Y} = \{\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(K)}\}$ of the observations, where $K < N_{\mathbf{y}}$. Note, that in this section $\mathbf{y}$ represents any arbitrary observation, which should not be confused with the sensor measurements $\mathbf{z}$. The original K-means clustering algorithm, introduced by Lloyd Lloyd (1982), aims at minimizing the within-cluster sum of squares of the distances according to

$$\min_{\substack{\mathbf{Y}^{(1)},\ldots,\mathbf{Y}^{(K)} \\ \mathbf{Y}^{(1)} \uplus \ldots \uplus \mathbf{Y}^{(K)} = \mathbf{Y}}} \sum_{k=1}^{K} \sum_{\mathbf{y}^{(i)} \in \mathbf{Y}^{(k)}} \|\mathbf{y}^{(i)} - \mu^{(k)}\|^2, \tag{17}$$

where $\mu^{(k)}$ is the mean value of the set $\mathbf{Y}^{(k)}$. A solution is found by alternating between the assignment step, which assigns each observation to the cluster with the closest mean, and the update step, which calculate the new means, until convergence, see e.g., the textbooks Bishop (2006); Hastie et al. (2009); Duda et al. (2001). A study of the convergence properties of the K-means algorithm is given in MacQueen (1967).

4.2 Regression Clustering

Now, suppose that the observations $\mathbf{y}^{(i)}$ of the set $\mathbf{Y}^{(k)}$ are produced by a regression model, with the regressors $\varphi^{(i)}$, according to

$$\mathbf{y}^{(i)} = (\varphi^{(i)})^{\mathrm{T}} \boldsymbol{\theta}^{(k)} + \mathbf{e}^{(i)}, \quad \mathbf{y}^{(i)} \in \mathbf{Y}^{(k)}, \quad \mathbf{e}^{(i)} \sim \mathcal{N}(0, R^{(i)}), \tag{18}$$

and that the objective is to identify the parameters $\boldsymbol{\theta}^{(k)}$ for every expected cluster $k = 1, \ldots, K$. The observations $\mathbf{y}^{(i)}$ within one set $\mathbf{Y}^{(k)}$ are chosen to fit the common parameter $\boldsymbol{\theta}^{(k)}$, i.e., the aim is to minimize the sum of squared residuals within each cluster according to

$$\min_{\substack{\mathbf{Y}^{(1)},\dots,\mathbf{Y}^{(K)}\\ \mathbf{Y}^{(1)}\uplus\dots\uplus\mathbf{Y}^{(K)}=\mathbf{Y}}} \sum_{k=1}^{K} \min_{\boldsymbol{\theta}^{(k)}} \sum_{\mathbf{y}^{(i)}\in\mathbf{Y}^{(k)}} \|\mathbf{y}^{(i)} - (\varphi^{(i)})^{\mathrm{T}}\boldsymbol{\theta}^{(k)}\|^2_{R^{(i)}}, \tag{19}$$

using the norm $\|x\|_R^2 = x^{\mathrm{T}}R^{-1}x$. This can be done by first clustering all observations using K-means (17) and thereafter estimating the parameters $\boldsymbol{\theta}^{(k)}$ for each cluster. However, we propose a joint clustering and estimation algorithm, which instead of clustering the observations around mean values $\mu^{(k)}$, clusters around models $(\varphi^{(i)})^{\mathrm{T}}\boldsymbol{\theta}^{(k)}$. Similar methods are used to solve for piecewise affine systems, see e.g., Roll (2003); Ferrari-Trecate et al. (2003), and for switching dynamic systems, see e.g., Petridis and Kehagias (1998), where the observation regions and submodels are identified iteratively. A solution is found by alternating between an assignment step and an estimation step until convergence. These steps are described in detail below.

Assignment Step

Given an initial guess of K parameters $\hat{\boldsymbol{\theta}}_0^{(k)}$ and covariances $P_0^{(k)}$, for $k = 1\dots,K$; each observation $\mathbf{y}^{(i)}$ is first assigned to the cluster with the smallest estimation error according to

$$\mathbf{Y}_\tau^{(k)} = \Big\{\mathbf{y}^{(i)} : \Big\|\mathbf{y}^{(i)} - (\varphi^{(i)})^{\mathrm{T}}\hat{\boldsymbol{\theta}}_\tau^{(k)}\Big\|_{S_\tau^{(k,i)}} \leq \Big\|\mathbf{y}^{(i)} - (\varphi^{(i)})^{\mathrm{T}}\hat{\boldsymbol{\theta}}_\tau^{(j)}\Big\|_{S_\tau^{(k,i)}},\ \forall j\Big\}, \tag{20a}$$

using the norm $\|x\|_S^2 = x^{\mathrm{T}}S^{-1}x$, where

$$S_\tau^{(k,i)} = (\varphi^{(i)})^{\mathrm{T}}P_\tau^{(k)}\varphi^{(i)} + R^{(i)}. \tag{20b}$$

The covariance of the parameter estimate, denoted $P_\tau^{(k)} = \operatorname{Cov}\hat{\boldsymbol{\theta}}_\tau^{(k)}$, is calculated in the estimation step. Note, that the discrete iteration index is denoted τ here.

Estimation Step

The second step is the update step, where new parameters $\hat{\boldsymbol{\theta}}^{(k)}$ are estimated for each cluster. Any estimation algorithm may be used, e.g., the weighted least squares according to

$$\hat{\boldsymbol{\theta}}_{\tau+1}^{(k)} = \left(\sum_{\mathbf{y}^{(i)}\in\mathbf{Y}_\tau^{(k)}} \varphi^{(i)}(R^{(i)})^{-1}(\varphi^{(i)})^{\mathrm{T}}\right)^{-1} \sum_{\mathbf{y}^{(i)}\in\mathbf{Y}_\tau^{(k)}} \varphi^{(i)}(R^{(i)})^{-1}(\varphi^{(i)})^{\mathrm{T}}\mathbf{y}^{(i)}, \tag{21a}$$

$$P_{\tau+1}^{(k)} = \left(\sum_{\mathbf{y}^{(i)}\in\mathbf{Y}_\tau^{(k)}} \varphi^{(i)}(R^{(i)})^{-1}(\varphi^{(i)})^{\mathrm{T}}\right)^{-1}. \tag{21b}$$

The assignment step (20) and estimation step (21) are iterated until the assignments no longer change.

4.3 Road Edge Estimation

To illustrate the joint clustering and estimation algorithm proposed above, we return to the example discussed in Section 2. In this example, it is likely that new stationary objects are spawned from road-side objects, such as guard rails. Consequently, we propose to use a simplified version of the road edge model in Lundquist and Schön (2009) in the joint clustering and estimation algorithm to find the road edge in the current map. From this estimate of the road edge the spawning intensity can be described.

The road edge is described by a polynomial model, linear in its parameters

$$\mathrm{y} = a_0^{(k)} + a_1\mathrm{x} + a_2\mathrm{x}^2 + a_3\mathrm{x}^3, \tag{22}$$

given the x and y-coordinates of the Gaussian components, and expressed in the vehicle's coordinate frame E. Furthermore, assume that the left and right edge of the road are approximately parallel and that they can be modeled using at most K polynomials (22). The polynomials only differ by the lateral distances $a_0^{(k)}$ for the $k = 1, \dots, K$ clusters.

Given the $J_{k|k}$ Gaussian components, the parameters

$$\boldsymbol{\theta}^{(k)} = \begin{bmatrix} a_0^{(k)} & a_1 & a_2 & a_3 \end{bmatrix}^{\mathrm{T}} \tag{23}$$

are estimated by rewriting the linear predictor (22) according to

$$\mathrm{y}^{(i)} = \left(\varphi^{(i)}\right)^{\mathrm{T}} \boldsymbol{\theta}^{(k)}, \tag{24}$$

where the regressors are given by

$$\varphi^{(i)} = \begin{bmatrix} 1 & \mathrm{x}^{(i)} & (\mathrm{x}^{(i)})^2 & (\mathrm{x}^{(i)})^3 \end{bmatrix}^{\mathrm{T}} \tag{25}$$

for the mean values of the components (8) in cluster k. The values of the parameters $\boldsymbol{\theta}^{(k)}$ are found by minimizing the estimation error within each cluster according to (19), i.e.,

$$\min_{\substack{\mathbf{Y}^{(1)},\dots,\mathbf{Y}^{(K)} \\ \mathbf{Y}^{(1)} \uplus \dots \uplus \mathbf{Y}^{(K)} = \mathbf{Y}}} \sum_{k=1}^{K} \min_{\boldsymbol{\theta}^{(k)}} \sum_{\mathbf{y}^{(i)} \in \mathbf{Y}^{(k)}} \|\mathrm{y}^{(i)} - (\varphi^{(i)})^{\mathrm{T}} \hat{\boldsymbol{\theta}}^{(k)}\|^2_{R^{(i)}}. \tag{26}$$

In this case the observations $\mathbf{y}^{(i)}$ are the mean values $m^{(i)} = \begin{bmatrix} \mathrm{x}^{(i)} & \mathrm{y}^{(i)} \end{bmatrix}^{\mathrm{T}}$ of the Gaussian components and the noise covariances $R^{(i)}$ are the estimated weighted covariances $P^{(i)}/w^{(i)}$. The parameters $\boldsymbol{\theta}^{(k)}$ are estimated iteratively using the assignment step (20) and the estimation step (21). Since radar measurements also are obtained by objects on the opposite side of the freeway, the states are clustered into four sets. A fair initial guess for $K = 4$ clusters could be $a_0^{(1)} = 10$, $a_0^{(2)} = -10$, $a_0^{(3)} = 30$, $a_0^{(4)} = -30$ and $a_1 = a_2 = a_3 = 0$.

Figure 2 shows the four resulting clusters in two traffic situations. The stars are the mean values of the Gaussian components, which are used as input to the joint clustering and estimation algorithm, and the lines are the resulting road edge models $\varphi^{\mathrm{T}}\boldsymbol{\theta}^{(k)}$. The past and future path of the ego vehicle is shown by black dots in the figure and can be used as a good reference of the road edges being correctly estimated. The number of components in each cluster indicates how many objects that should be spawned from the cluster. The spawning along the road edges is described in Section 6.2.

5 Merging

The Gaussian mixture formulation of the PHD filter tends to suffer from a high computational complexity as the number of mixture components, $J_{k|k}$, in (5) grows rapidly. To keep the complexity under control, methods are typically used to prune mixture components with too low weight $w_{k|k}^{(i)}$, which makes small contribution to the posterior density, and to merge components that are similar. The general merging process is discussed in Section 5.1. However, while keeping the complexity in check, it is important not to lose too much information in the process. The errors introduced in pruning and merging are analyzed in Clark and Vo (2007). In Section 5.2, we propose a general algorithm, which can be used for merging Gaussian components if more information about the system is available. The state space is often chosen since it is a convenient space to perform tracking in, i.e., to predict and update tracks in. However, as described in Section 5.2, Gaussian components may be better merged in another space, in which the system requirements are easier to describe.

Typically there exists some structure in a map. By exploiting this structure in the merging of the components, it is possible to maintain the information in the map, while reducing the number of components needed to describe it. The algorithm proposed in Section 5.2 is applied on the road map example in Section 5.3. The model-based merging method exploits the maps structure to find a more efficient representation of the posterior intensity in (5).

5.1 Background

The Gaussian mixture posterior intensity is given in (14). The aim of the merging is to compute a reduced Gaussian mixture, with $\ell < J_{k|k}$ components, that is close to the original intensity according to some distance measure d.

Some different methods for Gaussian mixture reduction by means of merging components exists, and have mainly been proposed for Gaussian sum filters. Alspach proposed a method to merge two Gaussian components with the same covariance Sorenson and Alspach (1971). The joining algorithm, proposed by Salmond Salmond (1990), evaluates the distances between every pair of components and successively pairs those components with the smallest distance. To avoid scaling problems, which may occur depending on the size of the elements in the state vector, Mahalanobis distance measure is often used, see e.g., Bar-Shalom et al. (2001).

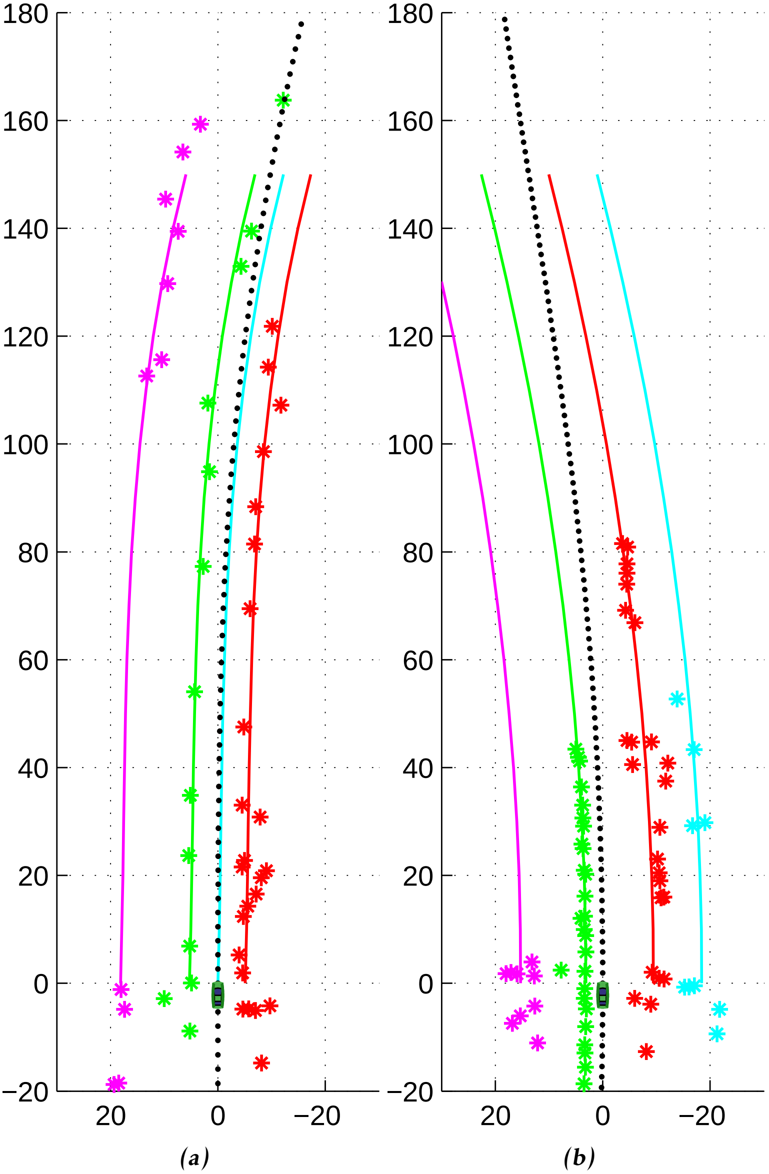


Figure 2: The four clusters, describing the road edges, are illustrated with different colors, and can be compared with the vehicles driven path indicated with black dots. The lines are a "snapshot" estimate, based on the Gaussian components at one time step k. The drivers view is shown in Figure 5a and 5b for the two traffic scenarios in (a) and (b), respectively.

The clustering algorithm, also presented in Salmond (1990), merges Gaussian components in groups rather than in pairs. A cluster center (centroid) is chosen as the component with the largest weight and the algorithm then merges all surrounding components to the centroid. This process is repeated with the remaining components until all have been clustered. The distance measure which is used to represent the closeness of a components i to the centroid j is given by

$$\left(d_k^{(i,j)}\right)^2 = \frac{w_k^{(j)} w_k^{(i)}}{w_k^{(j)} + w_k^{(i)}} \left(m_k^{(j)} - m_k^{(i)}\right)^{\mathrm{T}} \left(P_k^{(j)}\right)^{-1} \left(m_k^{(j)} - m_k^{(i)}\right), \tag{27}$$

where the distance is normalized to the covariance of the centroid. The factor $w_k^{(j)} w_k^{(i)}/(w_k^{(j)} + w_k^{(i)})$ makes it easier to cluster small components while large components retain as they are. The clustering algorithm is used in the original GM-PHD filter Vo and Ma (2006).

The joining and clustering algorithms are local methods in the sense that closely lying individual components are merged without considering the total density. During the last years some global methods, which take the whole Gaussian mixture into account, were published. A cost-function based approach was presented in Williams and Maybeck (2003), a global cluster approach in Schieferdecker and Huber (2009) and a bottom-up approach which starts with one component and successively builds up components to approximate the original mixture in Huber and Hanebeck (2008). The global methods are typically more accurate at the expense of higher computational effort, and are therefore not a good choice for the GM-PHD filter.

5.2 Algorithm

Let us introduce an invertible, possibly nonlinear, mapping function $\mathcal{T}(\,\cdot\,)$, which transforms the Gaussian components μ, defined in (8), from the state space to the merging space. The proposed algorithm consists of the following steps

1. Transform $\bar{\mu}_k^{(i)} := \mathcal{T}(\mu_k^{(i)})$.
2. Merge $\bar{\mu}_k^{(i)}$ using the cluster algorithm in Algorithm 10, resulting in $\tilde{\bar{\mu}}_k^{(i)}$.
3. Inverse transform $\tilde{\mu}_k^{(i)} := \mathcal{T}^{-1}(\tilde{\bar{\mu}}_k^{(i)})$.

The merging algorithm and the transformation is exemplified on the the road mapping example in the subsequent Section 5.3.

5.3 Road Mapping

The state space is spanned by the Cartesian world coordinate frame. However, for road mapping it is advantageous to merge Gaussian components lying along the road. Hence, the merging space is spanned by a road-aligned coordinate frame. The shape of the road can be described by a polynomial according to (cf. (22))

$$\mathrm{y}^E = \psi_{RE}\mathrm{x}^E + \frac{c_0}{2}(\mathrm{x}^E)^2 + \frac{c_1}{6}(\mathrm{x}^E)^3, \tag{28}$$

Algorithm 10 Merging Gaussian mixture by means of the cluster algorithm

```
Require: the Gaussian components $\{\bar{\mu}_k^{(i)}\}_{i=1}^{\bar{J}_{k|k}}$ and the threshold $\delta_\mathrm{m}$
initiate: by setting $\ell = 0$, $I = \{i = 1, \dots, J_{k|k}\}$ and then
repeat
    $\ell := \ell + 1$
    $j := \arg\max_{i \in I} w_k^{(i)}$
    $L := \left\{ i \in I \middle| d_k^{(i,j)} \leq \delta_\mathrm{m} \right\}$, where $d^{(i,j)}$ is given in (27)
    $\tilde{w}_k^{(\ell)} = \sum_{i\in L} w_k^{(i)}$
    $\tilde{m}_k^{(\ell)} = \frac{1}{\tilde{w}_k^{(\ell)}} \sum_{i\in L} w_k^{(i)} m_k^{(i)}$
    $\tilde{P}_k^{(\ell)} = \frac{1}{\tilde{w}_k^{(\ell)}} \sum_{i\in L} w_k^{(i)} (P_k^{(i)} + (\tilde{m}_k^{(\ell)} - m_k^{(i)})(\tilde{m}_k^{(\ell)} - m_k^{(i)})^\mathrm{T})$
    $I := I \backslash L$
until $I = \emptyset$
return the Gaussian components $\{\tilde{\bar{\mu}}_k^{(i)}\}_{i=1}^{\ell}$
```

where x^E and y^E are expressed in the ego vehicle's coordinate frame E. This model was introduced by Dickmanns Dickmanns and Zapp (1986) and a neat feature of the model is that the parameters can be interpreted as physical sizes. The angle between the longitudinal axis of the vehicle and the road lane is ψ_{RE}, see Figure 3. It is assumed that this angle is small and hence the approximation $\sin \psi_{RE} \approx \psi_{RE}$ is used. The curvature parameter is denoted c_0. The parameters can be estimated using computer vision and a lane tracker, or taken from the joint clustering and estimation method described in the previous section.

The non-linear transformation $\mathcal{T}$ from the world coordinate frame to the road fixed coordinate frame is calculated in two steps. First the components $\begin{bmatrix}\mathrm{x}^W & \mathrm{y}^W\end{bmatrix}^\mathrm{T}$, represented in the world coordinate frame W are transformed to $\begin{bmatrix}\mathrm{x}^E & \mathrm{y}^E\end{bmatrix}^\mathrm{T}$, represented in the ego vehicle coordinate frame E, using

$$\begin{bmatrix}\mathrm{x}^E \\ \mathrm{y}^E\end{bmatrix} = \left(R^{WE}\right)^\mathrm{T} \left(\begin{bmatrix}\mathrm{x}^W \\ \mathrm{y}^W\end{bmatrix} - \boldsymbol{d}_{EW}^W \right), \tag{29}$$

where the planar rotation matrix is given by

$$R^{WE} = \begin{bmatrix}\cos\psi_E & -\sin\psi_E \\ \sin\psi_E & \cos\psi_E\end{bmatrix}, \tag{30}$$

and where ψ_E is the angle of rotation from W to E. This angle is referred to as the yaw angle of the vehicle. In this work it is assumed that the position $\boldsymbol{d}_{EW}^W$ and orientation ψ_E of the ego vehicle is known. In the second step the components

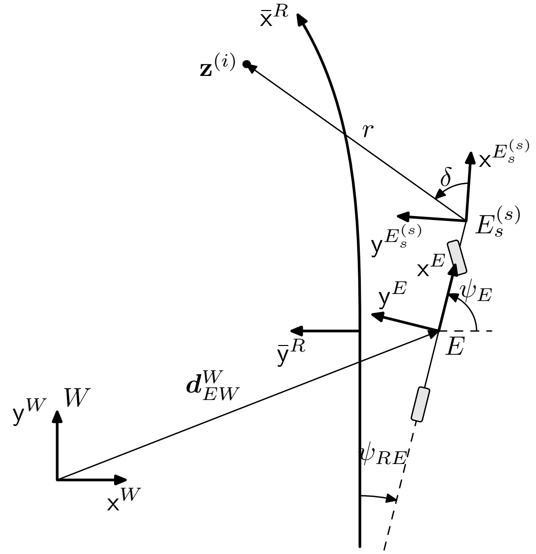


Figure 3: *The world coordinate frame is denoted W, the ego vehicle's coordinate frame is denoted E and the curved road coordinate frame is denoted R.*

are transformed into the road aligned coordinate frame according to

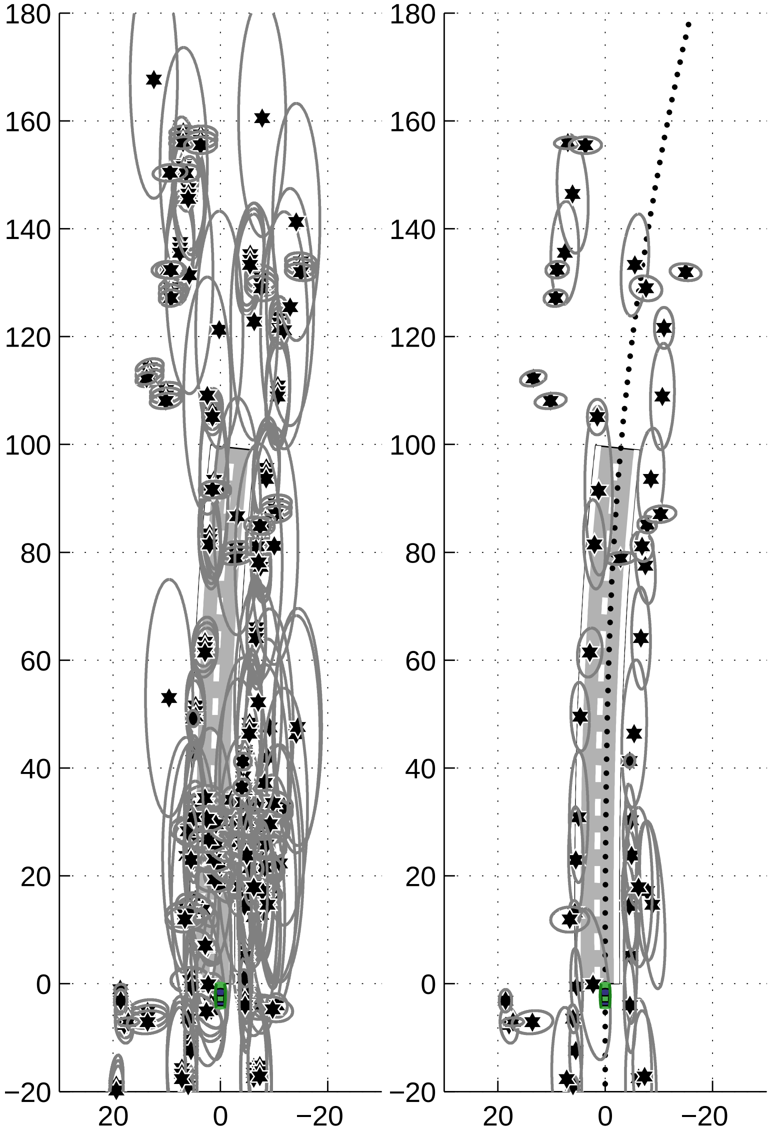
$$\begin{bmatrix} \bar{\mathrm{x}}^R \\ \bar{\mathrm{y}}^R \end{bmatrix} = \begin{bmatrix} \mathrm{x}^E \\ \mathrm{y}^E + \psi_{RE}\mathrm{x}^E + \frac{c_0}{2}(\mathrm{x}^E)^2 + \frac{c_1}{6}(\mathrm{x}^E)^3 \end{bmatrix}. \tag{31}$$

Some other approximations of the transformation $\mathcal{T}$ are described in Eidehall et al. (2007a). Since the transformation is nonlinear, the unscented transform, see e.g., Julier and Uhlmann (2004), is used to propagate the Gaussian components between the two spaces.

The covariance $P_k^{(j)}$ in the distance measure (27) is artificially formed to make it more likely for components lying along the road to be merged. A diagonal covariance matrix $\Sigma \in \mathbb{R}^{2\times 2}$ is created, with its two elements well separated, resulting in a pin-shaped covariance ellipse, and added to the state covariance according to

$$\bar{P}_k^{(j)} = P_k^{(j)} + \Sigma. \tag{32}$$

The merging is performed according to Algorithm 10, and the components are transformed back to the world coordinate frame using the inverse transform $\mathcal{T}^{-1}$, cf. Section 5.2. An example showing the components before and after the merging is illustrated in Figure 4. Compared with the camera view in Figure 5a, it is clearly shown that the resulting components are following the road edge.



***(a)** Before Pruning and Merging* ***(b)** After Pruning and Merging*

Figure 4: *Figure (a) shows the Gaussian mixture before pruning and merging and Figure (b) shows the reduced mixture. The illustrated road is estimated from the camera image, and can be used as a reference here. This is the same example as in Figure 5a-5c, where also a photo of the driver's view is given.*

6 Road Mapping Example

In this section all parts of the road mapping example, which was introduced in Section 2 and configured in the preceding sections, are put together. Especially the spawning of new target is exemplified in Section 6.2. The application dependent parts of GM-PHD filter recursion are described in Section 6.3. The capability of the posterior intensity to map the surroundings of the vehicle is shown in Section 6.4. However, we start with an overview and definition of the system in Section 6.1.

6.1 System Model

In this work, variables are defined from different perspectives. The measurements are obtained by the sensor on the ego vehicle and are therefore expressed in the ego vehicles coordinate frame, which is denoted E. The states, i.e., the position of the stationary objects, are expressed in a world fixed reference frame W. The radars are mounted at the front bumper at the positions $E_s^{(s)}$, for the sensors $s = 1, 2, 3$.

The linear process model

$$m_{k+1} = m_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(0, Q), \tag{33a}$$

of the Gaussian components allows for some motion, even though the objects are stationary. The reason is that one Gaussian component covers many stationary point sources and the mean value will change if sources are added or removed. The nonlinear measurement equation

$$\mathbf{z}_k = h(m_k) + \mathbf{e}_k, \quad \mathbf{e}_k \sim \mathcal{N}(0, R), \tag{33b}$$

describes how a measurement (9) i.e., the range r, the range rate $\dot{r}$ and the bearing angle ψ relates to the Gaussian components m_k. The nonlinear sensor model is given by

$$h(\mathbf{x}_k^{(i)}) = \begin{bmatrix} \sqrt{\left(\mathrm{x}^{(i)} - \mathrm{x}^W_{E_s^{(s)}W}\right)^2 + \left(\mathrm{y}^{(i)} - \mathrm{y}^W_{E_s^{(s)}W}\right)^2} \\ -\mathrm{v}\cos\left(\arctan \frac{\mathrm{y}^{(i)} - \mathrm{x}^W_{E_s^{(s)}W}}{\mathrm{x}^{(i)} - \mathrm{y}^W_{E_s^{(s)}W}} - \psi_{E_s^{(s)}}\right) \\ \arctan \frac{\mathrm{y}^{(i)} - \mathrm{x}^W_{E_s^{(s)}W}}{\mathrm{x}^{(i)} - \mathrm{y}^W_{E_s^{(s)}W}} - \psi_{E_s^{(s)}} \end{bmatrix}, \tag{34}$$

where $\begin{bmatrix} \mathrm{x}^W_{E_s^{(s)}W} & \mathrm{y}^W_{E_s^{(s)}W} \end{bmatrix}^\mathrm{T}$ is the position and $\psi_{E_s^{(s)}}$ the orientation of the radar sensor $s = 1, 2, 3$ in W, see Figure 3. The velocity of the ego vehicle is denoted v. The range rate is the relative velocity between the source and the moving vehicle. It is primarily included in the measurement vector to be used to sort stationary sources from moving target in the gate process, cf. (39).

6.2 Spawn Process

The left and right road edges are modeled with the polynomial (22) and with the parameter vectors $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$ which were estimated in Section 4.3. The parameters $\boldsymbol{\theta}^{(3)}$ and $\boldsymbol{\theta}^{(4)}$ are assumed to belong to lines further away, e.g., the other side of the road, and are therefore not considered here. The models are used to create a prior intensity, which is a Gaussian mixture of the form (12) along the road edges. The road model (22) is expressed in vehicle coordinates and after having derived the Gaussian mixture, the components are transformed into the world frame W.

A number $J_{\beta,k+1}/2$ of x^E-coordinates, denoted $\left\{\mathsf{x}^{E,(j)}\right\}_{j=1}^{J_{\beta,k+1}/2}$, are chosen in the range from 0 to the maximum range of the radar sensor. The corresponding y^E-coordinates $\left\{\mathsf{y}^{E,(j)}\right\}_{j=1}^{J_{\beta,k+1}/2}$ and $\left\{\mathsf{y}^{E,(j)}\right\}_{j=J_{\beta,k+1}/2+1}^{J_{\beta,k+1}}$ are derived by using the road border model (22) and the parameters $\hat{\boldsymbol{\theta}}^{(1)}$ and $\hat{\boldsymbol{\theta}}^{(2)}$, respectively. The coordinates form the mean values of the position of the Gaussian components on the road edges according to

$$m_{\beta,k+1}^{E,(j)} = \begin{bmatrix} \mathsf{x}^{E,(j)} & \mathsf{y}^{E,(j)} \end{bmatrix}^{\mathrm{T}}. \tag{35a}$$

The covariance of the Gaussian components is given by the diagonal matrix

$$P_{\beta,k+1}^{E,(j)} = \begin{bmatrix} \left(\sigma_{\mathsf{x}}^{E}\right)^2 & 0 \\ 0 & \left(\sigma_{\mathsf{y}}^{E}(\mathsf{x}^{E(j)})\right)^2 \end{bmatrix}, \tag{35b}$$

where it is assumed that the variance of the y-coordinate increases with the x-distance, i.e., $\sigma_y^E(\mathsf{x}^{E,(j)})$ depends on $\mathsf{x}^{E(j)}$, to model the increased uncertainty in the shape of the road at long distances.

So far the derivations are accomplished in the ego vehicles coordinate frame E, but since the map is expressed in the world frame W, the components of the Gaussian mixture are transformed into the world coordinate frame according to

$$m_{\beta,k+1}^{(j)} = R^{WE} m_{\beta,k+1}^{E,(j)} + \boldsymbol{d}_{EW}^{W}, \tag{36a}$$

$$P_{\beta,k+1}^{(j)} = R^{WE} P_{\beta,k+1}^{E,(j)} \left(R^{WE}\right)^{\mathrm{T}}, \tag{36b}$$

to be used in (12). The weight $w_{\beta,k+1}^{(j)}$ represents the expected number of new targets originating from $m_{\beta,k+1}^{(j)}$.

6.3 GM-PHD Filter Recursion

The general GM-PHD filter recursion was described in Section 3. Additional details for the road mapping example are given in this section. The merging of Gaussian components, described in Section 5.3, is performed in each iteration.

Prediction

The process model (33a) is linear and the Gaussian components in (11) are derived using the Kalman filter prediction step according to

$$m^{(i)}_{k+1|k} = m^{(i)}_{k|k}, \tag{37a}$$

$$P^{(i)}_{k+1|k} = P^{(i)}_{k|k} + Q_{k-1}. \tag{37b}$$

The probability of survival is $p_S = 0.99$.

Measurement Update

The measurement equation (34) is nonlinear and the unscented transform (UT) is used to propagate the state variables through the measurement equation, see e.g., Julier and Uhlmann (2004). A set of L sigma points and weights, denoted by $\left\{\chi^{(\ell)}_k, u^{(\ell)}\right\}^L_{\ell=0}$ are generated from each Gaussian component $\mathcal{N}(x^{(i)}_{k|k-1}, P^{(i)}_{k|k-1})$ using the method described in Julier and Uhlmann (2004). The sigma points are transformed to the measurement space using (34) to obtain the propagated sigma point $\zeta^{(\ell)}_{k|k-1}$ and the first and second order moments of the measurement density are approximated as

$$\eta^{(i)}_{k|k-1} = \sum_{\ell=0}^{L} u^{(\ell)} \zeta^{(\ell)}_{k|k-1}, \tag{38a}$$

$$S^{(i)}_{k|k-1} = \sum_{\ell=0}^{L} u^{(\ell)} \left(\zeta^{(\ell)}_{k|k-1} - \eta^{(i)}_{k|k-1}\right)\left(\zeta^{(\ell)}_{k|k-1} - \eta^{(i)}_{k|k-1}\right)^{\mathrm{T}} + R_k, \tag{38b}$$

$$G^{(i)}_{k|k-1} = \sum_{\ell=0}^{L} u^{(\ell)} \left(\chi^{(\ell)}_{k|k-1} - m^{(i)}_{k|k-1}\right)\left(\zeta^{(\ell)}_{k|k-1} - \eta^{(i)}_{k|k-1}\right)^{\mathrm{T}}. \tag{38c}$$

Note that these variables also are used to derive the updated weights in (16). The Gaussian components are updated given the information from each new measurement $\mathbf{z}_k$, which is in the gate

$$\mathcal{G}_i = \left\{\mathbf{z}_k \middle| \left(\mathbf{z}_k - \eta^{(i)}_{k|k-1}\right)^{\mathrm{T}} \left(S^{(i)}_{k|k-1}\right)^{-1} \left(\mathbf{z}_k - \eta^{(i)}_{k|k-1}\right) < \delta_{\mathcal{G}}\right\}, \tag{39}$$

for the gate threshold $\delta_{\mathcal{G}} = 11.3$, according to

$$m^{(i)}_{k|k}(\mathbf{z}) = m^{(i)}_{k|k-1} + K^{(i)}_k \left(\mathbf{z}_k - \eta^{(i)}_{k|k-1}\right), \tag{40a}$$

$$P^{(i)}_{k|k} = P^{(i)}_{k|k-1} - K^{(i)} S^{(i)}_{k|k-1} \left(K^{(i)}\right)^{\mathrm{T}}, \tag{40b}$$

$$K^{(i)} = G^{(i)}_{k|k-1} \left(S^{(i)}_{k|k-1}\right)^{-1}. \tag{40c}$$

The updated components are together forming the updated intensity in (15).

The posterior intensity at time step k was given in (14). The probability of detection is adjusted according to

$$p_{D,k}(x) = \begin{cases} p_{D,k} & \eta_{k|k-1}^{(i)} \in \mathcal{Z}, \\ 0 & \eta_{k|k-1}^{(i)} \notin \mathcal{Z}, \end{cases} \tag{41}$$

and the value is $p_D = 10^{-3}$. Gaussian components which are outside field of view of the sensor, i.e., the measurement space $\mathcal{Z}^{(s)}$, are hence not updated. The clutter intensity is $\kappa = 10^{-8}$.

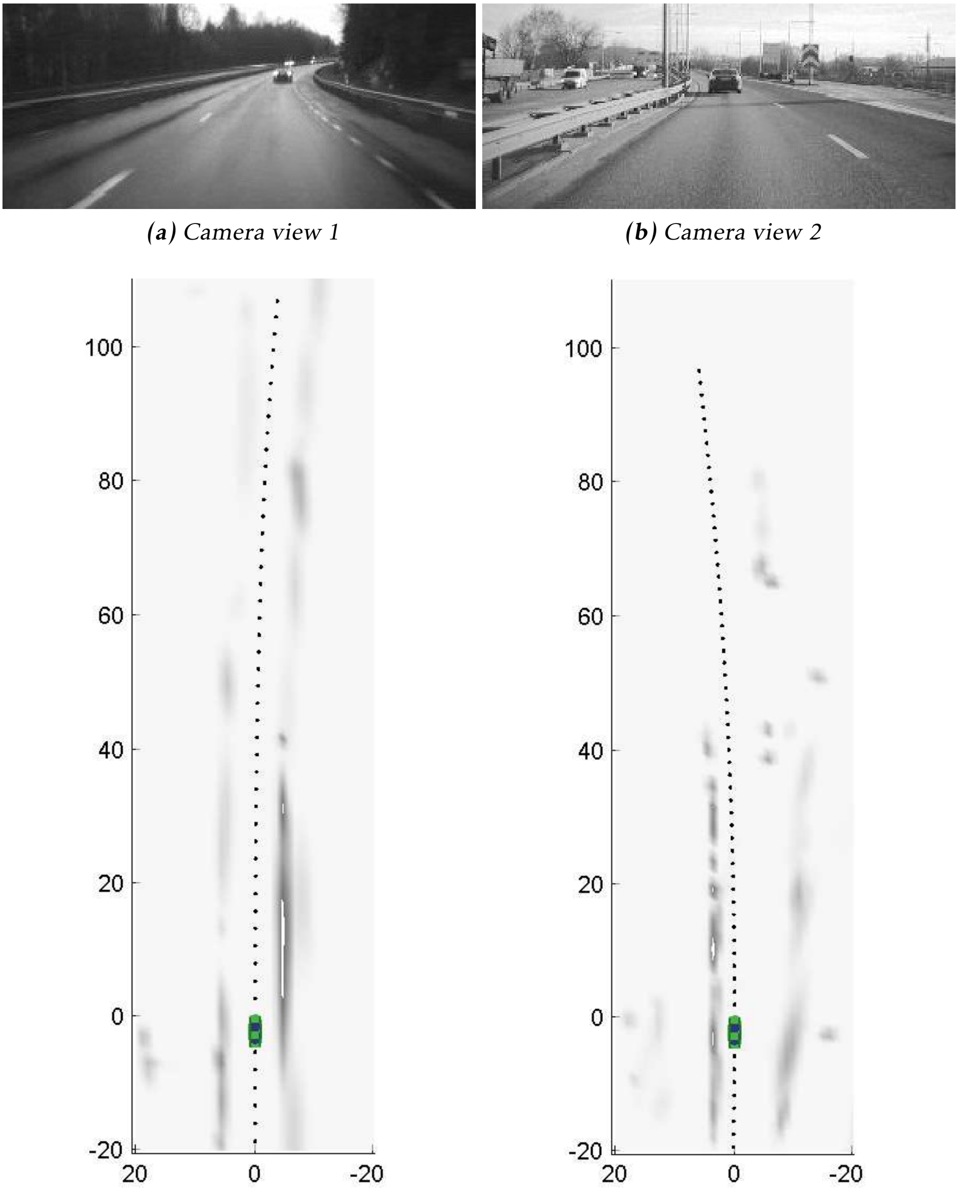
The measurements from the three radar sensors are fused by considering their measurement times, synchronizing the data, and updating the filter accordingly.

6.4 Experiments and Results

The experiments were conducted with a prototype passenger car, equipped with the radar sensors configured as shown in Figure 1. No reference data of the road borders exist, but the vehicle's position was recorded and may be used to illustrate that the resulting map is reasonable. One example of the estimated intensity at a freeway traffic scenario is shown as a bird's eye view in Figure 5c. Darker regions illustrate higher concentrations of point sources, which in this figure stem from the guardrails to the left and the right of the road. As expected, the path of the ego vehicle, indicated by the black dots, is in between the two regions of higher object concentration. The driver's view is shown in Figure 5a.

A second example is shown in Figure 5d and 5b. Here, the freeway exit is clearly visible in the intensity map, which shows that the proposed method to create maps is very conformable.

The Gaussian components are generally removed from the filter when the vehicle passed those parts of the map. However, to give a more comprehensive overview, these components are stored and the resulting intensity based map is shown together with an occupancy grid map (OGM) and a flight photo in Figure 6. The top figure is the map produced as described in this contribution. The OGM is based on the same data set and used as a comparison of an existing and well established algorithm, see e.g., Lundquist et al. (2009). The gray-level of the OGM indicates the probability of occupancy, the darker the grid cell the more likely it is to be occupied. As seen in the figure the road edges are not modeled as distinct with the OGM. The OGM representation of the map is not very efficient, since huge parts of the map are gray indicating that nothing is known about these areas. An OGM matrix with often more than 10000 elements must be updated and communicated to other safety functions of a car at each time step. The compact representation is an advantage of the intensity based map. Each Gaussian components is parametrized with 7 scalar values according to (8). Since most maps are modeled with $10-30$ components it summarizes to around $70-210$ scalar values, which easily can be send on the vehicles CAN-bus to other safety functions. Finally, the bottom photo is a very accurate flight photo (obtained from the Swedish mapping, cadastral and land registration authority), which can be used as ground truth to visualize the quality of the intensity based map. Note for



***(a)** Camera view 1*

***(b)** Camera view 2*

***(c)** Intensity map 1*

***(d)** Intensity map 2*

***Figure 5:** The image in (a) shows the driver's view of the intensity map in (c), and the image in (b) is the driver's view of the intensity map in (d). The darker the areas in the intensity map, the higher the concentration of objects. The drivers path is illustrated with black dots and may be used as a reference.*

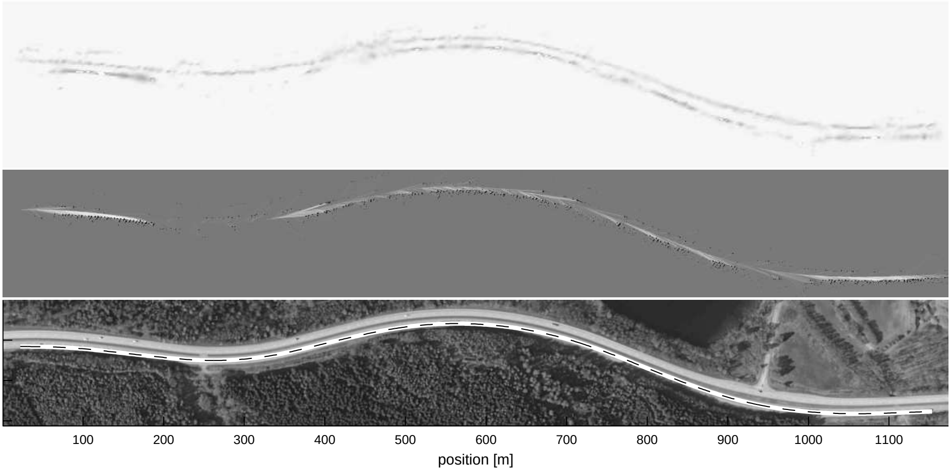


Figure 6: *The top figure shows the intensity based map obtained from radar measurements collected on a highway (drivers view in Figure 5a). The OGM in the middle figure serves as a comparison of an existing algorithm. The bottom figure is a flight photo used as ground truth, where the driven trajectory is illustrated with a dashed line (©Lantmäteriet Gävle 2010. Medgivande I 2010/1540, reprinted with permission).*

example the junction at 1000 m.

7 Conclusion

In this work it is shown how the GM-PHD filter can be used to create a map of the environment using noisy point measurements. The map is represented by the intensity, which describes the concentration of point sources. Methods to identify and utilize typical map structures and to create an efficient prior in the GM-PHD-filter have been presented. The structures are identified by a clustering algorithm which alternates between assigning existing map features to clusters, and estimating structures within each cluster, until convergence. Furthermore, these structures are also used to find an efficient representation of the resulting map.

The usefulness of the proposed mapping approach is demonstrated using automotive radar measurements collected on Swedish freeways. The resulting intensity map of stationary objects, such as guardrails and lampposts, can be viewed as a map over the concentration of radar reflectors. The parametrization is compact and the map is easily sent over the CAN-bus to active safety functions, which can use it to derive drivable trajectories with low probability of occupancy.

The intensity is a distribution and its graphical representation provides an intuitive description of the map, which is useful to decision making and path planning algorithms. However, the intensity is an approximation of a true map. It remains for future work to find an expression for the real map and also to develop a metric to quantify the accuracy of the intensity map.

Bibliography

M. Adams, W. S. Wijesoma, and A. Shacklock. Autonomous navigation: Achievements in complex environments. *IEEE Instrumentation & Measurement Magazine*, 10(3):15–21, June 2007.

T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, September 2006.

Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, New York, NY, USA, 2001.

C. M. Bishop. *Pattern recognition and machine learning*. Springer, New York, NY, USA, 2006. ISBN 0-387-31073-8.

M. Buehler, K. Iagnemma, and S. Singh, editors. *Special Issue on the 2007 DARPA Urban Challenge, Part I-III*, volume 25 (8–10). Journal of Field Robotics, 2008.

D. Clark and B.-N. Vo. Convergence analysis of the Gaussian mixture PHD filter. *IEEE Transactions on Signal Processing*, 55(4):1204–1212, April 2007.

D. J. Daley and D. Vere-Jones. *An introduction to the theory of point processes. Vol. 1 , Elementary theory and method*. Springer, New York, NY, USA, 2 edition, 2003.

E. D. Dickmanns and A. Zapp. A curvature-based scheme for improving road vehicle guidance by computer vision. In *Proceedings of the SPIE Conference on Mobile Robots*, volume 727, pages 161–198, Cambridge, MA, USA, 1986.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, New York, NY, USA, 2 edition, 2001.

H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping (SLAM): Part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, June 2006.

A. Eidehall, J. Pohl, and F. Gustafsson. Joint road geometry estimation and vehicle tracking. *Control Engineering Practice*, 15(12):1484–1494, December 2007a.

A. Eidehall, J. Pohl, F. Gustafsson, and J. Ekmark. Toward autonomous collision avoidance by steering. *IEEE Transactions on Intelligent Transportation Systems*, 8(1):84–94, March 2007b.

A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 3(3):249–265, June 1987.

O. Erdinc, P. Willett, and Y. Bar-Shalom. The bin-occupancy filter and its connection to the PHD filters. *IEEE Transactions on Signal Processing*, 57(11): 4232–4246, November 2009.

G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2):205–217, February 2003.

A. Foessel-Bunting, J. Bares, and W. L. Whittaker. Three-dimensional map building with mmw radar. In R. Chatila A. Halme and E. Prassler, editors, *Proceedings of the International Conference on Field and Service Robotics*, Helsinki, Finland, June 2001.

K. Gilholm and D. Salmond. Spatial distribution model for tracking extended objects. *IEE Proceedings of Radar, Sonar and Navigation*, 152(5):364–371, October 2005.

K. Gilholm, S. Godsill, S. Maskell, and D. Salmond. Poisson models for extended target and group tracking. In *Proceedings of Signal and Data Processing of Small Targets*, volume 5913, pages 230–241, San Diego, CA, USA, August 2005. SPIE.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning : data mining, inference, and prediction*. Springer, New York, NY, USA, 2 edition, 2009.

M. F. Huber and U. D. Hanebeck. Progressive Gaussian mixture reduction. In *Proceedings of the International Conference on Information Fusion*, pages 1–8, Cologne, Germany, July 2008.

S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, March 2004.

K. Kaliyaperumal, S. Lakshmanan, and K. Kluge. An algorithm for detecting roads and obstacles in radar images. *IEEE Transactions on Vehicular Technology*, 50(1):170–182, January 2001.

A. Kirchner and C. Ameling. Integrated obstacle and road tracking using a laser scanner. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 675–681, Dearborn, MI, USA, October 2000.

A. Kirchner and T. Heinrich. Model based detection of road boundaries with a laser scanner. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 93–98, Stuttgart, Germany, October 1998.

S. Lakshmanan, K. Kaliyaperumal, and K. Kluge. Lexluther: an algorithm for detecting roads and obstacles in radar images. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 415–420, Boston, MA, USA, November 1997.

S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982.

C. Lundquist and T. B. Schön. Estimation of the free space in front of a moving vehicle. In *Proceedings of the SAE World Congress*, SAE paper 2009-01-1288, Detroit, MI, USA, April 2009.

C. Lundquist, T. B. Schön, and U. Orguner. Estimation of the free space in front of a moving vehicle. Technical Report LiTH-ISY-R-2892, Department of Electrical Engineering, Linköping University, Sweden, Linköping, Sweden, April 2009.

C. Lundquist, L. Danielsson, and F. Gustafsson. Random set based road mapping using radar measurements. In *Proceedings of the European Signal Processing Conference*, pages 219–223, Aalborg, Denmark, August 2010.

C. Lundquist, L. Hammarstrand, and F. Gustafsson. Road intensity based mapping using radar measurements with a probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 59(4):1397–1408, April 2011a.

C. Lundquist, U. Orguner, and F. Gustafsson. Extended target tracking using polynomials with applications to road-map estimation. *IEEE Transactions on Signal Processing*, 59(1):15–26, January 2011b.

B. Ma, S. Lakshmanan, and A. O. Hero. Simultaneous detection of lane and pavement boundaries using model-based multisensor fusion. *IEEE Transactions on Intelligent Transportation Systems*, 1(3):135–147, September 2000.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley, CA, USA, July 1967.

R. P. S. Mahler. A theoretical foundation for the Stein-Winter probability hypothesis density (PHD) multitarget tracking approach. In *Proceedings of the national symposium on sensor data fusion*, San Antonio, TX, USA, June 2000.

R. P. S. Mahler. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, October 2003.

R. P. S. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, Boston, MA, USA, 2007.

H. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9 (2):61–74, 1988. ISSN 0738-4602.

J. Mullane, B.-N. Vo, M. D. Adams, and W. S. Wijesoma. A random set formulation for Bayesian SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1043–1049, Nice, France, September 2008.

J. Mullane, M. D. Adams, and W. S. Wijesoma. Robotic mapping using measurement likelihood filtering. *The International Journal of Robotics Research*, 28 (2):172–190, 2009. ISSN 0278-3649.

M. Nikolova and A Hero. Segmentation of a road from a vehicle-mounted radar and accuracy of the estimation. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 284–289, Dearborn, MI, USA, October 2000.

V. Petridis and A. Kehagias. Identification of switching dynamical systems using multiple models. In *Proceedings of the IEEE Conference on Decision and Control*, volume 1, pages 199–204, Tampa, Florida, USA, December 1998.

J. Roll. *Local and Piecewise Affine Approaches to System Identification*. PhD thesis No 802, Linköping Studies in Science and Technology, Linköping, Sweden, April 2003.

D. J. Salmond. Mixture reduction algorithms for target tracking in clutter. *Proceedings of Signal and Data Processing of Small Targets*, 1305(1):434–445, January 1990.

D. Schieferdecker and M.F. Huber. Gaussian mixture reduction via clustering. In *Proceedings of the International Conference on Information Fusion*, pages 1536–1543, Seattle, WA, USA, July 2009.

H. Sidenbladh. Multi-target particle filtering for the probability hypothesis density. In *Proceedings of the International Conference on Information Fusion*, volume 2, pages 800–806, Cairns, Australia, March 2003.

R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *Proceedings of the International Symposium on Robotics Research*, pages 467–474, Cambridge, MA, USA, 1988. MIT Press.

H. W. Sorenson and D. L. Alspach. Recursive Bayesian estimation using Gaussian sum. *Automatica*, 7:465–479, July 1971.

J. Sparbert, K. Dietmayer, and D. Streller. Lane detection and street type classification using laser range images. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 454–459, Oakland, CA, USA, August 2001.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, Cambridge, MA, USA, 2005.

B.-N. Vo and W.-K. Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, November 2006.

B.-N. Vo, S. Singh, and A. Doucet. Random finite sets and sequential Monte Carlo methods in multi-target tracking. In *Proceedings of the International Radar Conference*, pages 486–491, Adelaide, Australia, September 2003.

J. L. Williams and P. S. Maybeck. Cost-function-based Gaussian mixture reduction for target tracking. In *Proceedings of the International Conference on Information Fusion*, volume 2, pages 1047–1054, Cairns, Australia, July 2003.

T. Zajic and R. P. S. Mahler. Particle-systems implementation of the PHD multitarget-tracking filter. In *Signal Processing, Sensor Fusion, and Target Recognition XII*, volume 5096, pages 291–299, Orlando, FL, USA, April 2003. SPIE.

Paper E

Extended Target Tracking Using a Gaussian-Mixture PHD Filter

Authors: Karl Granström, Christian Lundquist and Umut Orguner

Edited version of the paper:

K. Granström, C. Lundquist, and U. Orguner. Extended target tracking using a Gaussian-mixture PHD filter. *IEEE Transactions on Aerospace and Electronic Systems*, 2011a. Under review.

The paper presents data that was previously published in:

K. Granström, C. Lundquist, and U Orguner. A Gaussian mixture PHD filter for extended target tracking. In *Proceedings of the International Conference on Information Fusion*, Edinburgh, UK, July 2010.

Preliminary version:

Technical Report LiTH-ISY-R-3028, Dept. of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden.

Extended Target Tracking Using a Gaussian-Mixture PHD Filter

Karl Granström, Christian Lundquist and Umut Orguner

Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
karl@isy.liu.se, lundquist@isy.liu.se, umut@isy.liu.se

Abstract

This paper presents a Gaussian-mixture implementation of the PHD filter for tracking extended targets. The exact filter requires processing of all possible measurement set partitions, which is generally infeasible to implement. A method is proposed for limiting the number of considered partitions and possible alternatives are discussed. The implementation is used on simulated data and in experiments with real laser data, and the advantage of the filter is illustrated. Suitable remedies are given to handle spatially close targets and target occlusion.

1 Introduction

In most multi-target tracking applications it is assumed that each target produces at most one measurement per time step. This is true for cases when the distance between the target and the sensor is large in comparison to the target's size. In other cases however, the target size may be such that multiple resolution cells of the sensor are occupied by the target. Targets that potentially give rise to more than one measurement per time step are categorized as extended. Examples include the cases when vehicles use radar sensors to track other road-users, when ground radar stations track airplanes which are sufficiently close to the sensor, or in mobile robotics when pedestrians are tracked using laser range sensors.

Gilholm and Salmond Gilholm and Salmond (2005) have presented an approach for tracking extended targets under the assumption that the number of received target measurements in each time step is Poisson distributed. Their algorithm was illustrated with two examples where point targets which may generate more than one measurement and objects that have a 1-D extension (infinitely thin stick of length l) are tracked. In Gilholm et al. (2005) a measurement model was suggested which is an inhomogeneous Poisson point process. At each time step, a Poisson distributed random number of measurements are generated, distributed around the target. This measurement model can be understood to imply that the extended target is sufficiently far away from the sensor for its measurements to resemble a cluster of points, rather than a geometrically structured ensemble. A similar approach is taken in Boers et al. (2006) where track-before-detect theory is used to track a point target with a 1-D extent.

Using the rigorous finite set statistics (FISST), Mahler has pioneered the recent advances in the field of multiple target tracking with a set theoretic approach where the targets and measurements are treated as random finite sets (RFS). This type of approach allows the problem of estimating multiple targets in clutter and uncertain associations to be cast in a Bayesian filtering framework Mahler (2007), which in turn results in an optimal multi-target Bayesian filter. As is the case in many nonlinear Bayesian estimation problems, the optimal multi-target Bayesian filter is infeasible to implement except for simple examples and an important contribution of FISST is to provide structured tools in the form of the statistical moments of a RFS. The first order moment of a RFS is called *probability hypothesis density* (PHD), and it is an intensity function defined over the state space of the targets. The so called PHD filter Mahler (2003, 2007) propagates in time PHDs corresponding to the set of target states as an approximation of the optimal multi-target Bayesian filter. A practical implementation of the PHD filter is provided by approximating the PHDs with Gaussian-mixtures (GM) Vo and Ma (2006) which results in the Gaussian-mixture PHD (GM-PHD) filter. In the recent work Mahler (2009), Mahler presented an extension of the PHD filter to also handle extended targets of the type presented in Gilholm et al. (2005).

In this paper, we present a Gaussian-mixture implementation of the PHD-filter for extended targets Mahler (2009), which we call the extended target GM-PHD-filter (ET-GM-PHD). In this way, we, to some extent, give a practical extension of the series of work in Gilholm et al. (2005); Vo and Ma (2006); Mahler (2009). An earlier version of this work was presented in Granström et al. (2010) and the current, significantly improved, version includes also practical examples with real data. For space considerations, we do not repeat the derivation of the PHD-filter equations for extended targets and instead refer the reader to Mahler (2009).

The document is outlined as follows. The multiple extended target tracking problem is defined in Section 2. The details of the Gaussian-mixture implementation are given in Section 3. For the measurement update step of the ET-GM-PHD-filter, different partitions of the set of measurements have to be considered. A measurement clustering algorithm used to reduce the combinatorially exploding number of possible measurement partitions is described in Section 4. The proposed approaches are evaluated using both simulations and experiments. The target tracking setups for these evaluations are described in Section 5, the simulation results are presented in Section 6 and results using data from a laser sensor are presented in Section 7. Finally, Section 8 contains conclusions and thoughts on future work.

2 Target Tracking Problem Formulation

In previous work, extended targets have often been modeled as targets having a spatial extension or shape that would lead to multiple measurements, as opposed to at most a single measurement. On the other hand, the extended target tracking problem can be simplified by the assumption that the measurements originating from a target are distributed approximately around a target reference point Gilholm and Salmond (2005) which can be e.g. the centroid or any other point depending on the extent (or the shape) of the target. Though all targets obviously have a spatial extension and shape, in the latter type of modeling, only the target reference point is important and the target extent does not need to be estimated.

The relevant target characteristics that are to be estimated form the target's state vector $\mathbf{x}$. Generally, beside the kinematic variables as position, velocity and orientation, the state vector may also contain information about the target's spatial extension. As mentioned above, when the target's state does not contain any variables related to the target extent, though the estimation is done as if the target was a point (i.e. the target reference point), the algorithms should still take care of the multiple measurements that originate from a target. Hence, in this study, we use a generalized definition of an extended target, given below, which does not depend on whether the target extent is estimated or not.

1 Definition (Extended Target). A target which potentially gives rise to more than one measurement per time step irrespective of whether the target's extent is explicitly modeled (and/or estimated) or not.

In this work, to simplify the presentation, no information about the size and shape of the target is kept in the state vector $\mathbf{x}$, i.e. the target extent is not explicitly estimated. Nevertheless, it must be emphasized that this causes no loss of generality as shown by the recent work Granström et al. (2011b) where the resulting ET-GM-PHD filter is used to handle the joint estimation of size, shape and kinematic variables for rectangular and elliptical extended targets. We model both the target states to be estimated, and the measurements collected, as RFSs . The motivation behind this selection is two-fold. First, in many practical systems, although the sensor reports come with a specific measurement order, the results of the target tracking algorithms are invariant under the permutations of this order. Hence, modeling the measurements as the elements of a set in which the order of the elements is irrelevant makes sense. Second, this work unavoidably depends on the previous line of work Mahler (2009), which is based on such a selection.

The initial GM-PHD work Vo and Ma (2006) does not provide tools for ensuring track continuity, for which some remedies are described in the literature, see e.g. Panta et al. (2009). It has however been shown that labels for the Gaussian components can be included into the filter in order to obtain individual target tracks, see e.g. Clark et al. (2006). In this work, for the sake of simplicity, labels are not used, however they can be incorporated as in Clark et al. (2006) to provide track continuity.

We denote the unknown number of targets as $N_{\mathbf{x},k}$, and the set of target states to be estimated at time k is $\mathbf{X}_k = \{\mathbf{x}_k^{(i)}\}_{i=1}^{N_{\mathbf{x},k}}$. The measurement set obtained at time k is $\mathbf{Z}_k = \{\mathbf{z}_k^{(i)}\}_{i=1}^{N_{\mathbf{z},k}}$ where $N_{\mathbf{z},k}$ is the number of the measurements.

The dynamic evolution of each target state $\mathbf{x}_k^{(i)}$ in the RFS $\mathbf{X}_k$ is modeled using a linear Gaussian dynamical model,

$$\mathbf{x}_{k+1}^{(i)} = F_k \mathbf{x}_k^{(i)} + G_k \mathbf{w}_k^{(i)}, \tag{1}$$

for $i = 1, \ldots, N_{\mathbf{x},k}$, where $\mathbf{w}_k^{(i)}$ is Gaussian white noise with covariance $Q_k^{(i)}$. Note that each target state evolves according to the same dynamic model independent of the other targets.

The number of measurements generated by the ith target at each time step is a Poisson distributed random variable with rate $\gamma\left(\mathbf{x}_k^{(i)}\right)$ measurements per scan, where $\gamma(\,\cdot\,)$ is a known non-negative function defined over the target state space. The probability of the ith target generating at least one measurement is then given as

$$1 - e^{-\gamma\left(\mathbf{x}_k^{(i)}\right)}. \tag{2}$$

The ith target is detected with probability $p_{\mathrm{D}}\left(\mathbf{x}_k^{(i)}\right)$ where $p_{\mathrm{D}}(\,\cdot\,)$ is a known non-negative function defined over the target state space. This gives the effective probability of detection

$$\left(1 - e^{-\gamma\left(\mathbf{x}_k^{(i)}\right)}\right) p_{\mathrm{D}}\left(\mathbf{x}_k^{(i)}\right). \tag{3}$$

The measurements originating from the ith target are assumed to be related to the target state according to a linear Gaussian model given as

$$\mathbf{z}_k^{(j)} = H_k \mathbf{x}_k^{(i)} + \mathbf{e}_k^{(j)}, \tag{4}$$

where $\mathbf{e}_k^{(j)}$ is white Gaussian noise with covariance R_k. Each target is assumed to give rise to measurements independently of the other targets. We here emphasize, that in an RFS framework both the set of measurements $\mathbf{Z}_k$ and the set of target states $\mathbf{X}_k$ are unlabeled, and hence no assumptions are made regarding which target gives rise to which measurement.

The number of clutter measurements generated at each time step is a Poisson distributed random variable with rate $\beta_{FA,k}$ clutter measurements per surveillance volume per scan. Thus, if the surveillance volume is V_s, the mean number of clutter measurements is $\beta_{FA,k} V_s$ clutter measurements per scan. The spatial distribution of the clutter measurements is assumed uniform over the surveillance volume.

The aim is now to obtain an estimate of the sets of the target states $\mathbf{X}^K = \{\mathbf{X}_k\}_{k=1}^K$ given the sets of measurements $\mathbf{Z}^K = \{\mathbf{Z}_k\}_{k=1}^K$. We achieve this by propagating the predicted and updated PHDs, denoted $D_{k|k-1}(\cdot)$ and $D_{k|k}(\cdot)$, respectively, of the set of target states $\mathbf{X}_k$, using the PHD filter presented in Mahler (2009).

3 Gaussian-Mixture Implementation

In this section, following the derivation of the GM-PHD-filter for standard single measurement targets in Vo and Ma (2006), a PHD recursion for the extended target case is described. Since the prediction update equations of the extended target PHD filter are the same as those of the standard PHD filter Mahler (2009), the Gaussian mixture prediction update equations of the ET-GM-PHD filter are the same as those of the standard GM-PHD filter in Vo and Ma (2006). For this reason, here we only consider the measurement update formulas for the ET-GM-PHD filter.

The predicted PHD has the following Gaussian-mixture representation

$$D_{k|k-1}(\mathbf{x}) = \sum_{j=1}^{J_{k|k-1}} w_{k|k-1}^{(j)} \mathcal{N}\left(\mathbf{x} \,|\, m_{k|k-1}^{(j)}, P_{k|k-1}^{(j)}\right) \tag{5}$$

where

- $J_{k|k-1}$ is the predicted number of components;
- $w_{k|k-1}^{(j)}$ is the weight of the jth component;
- $m_{k|k-1}^{(j)}$ and $P_{k|k-1}^{(j)}$ are the predicted mean and covariance of the jth component;
- the notation $\mathcal{N}(x \,|\, m, P)$ denotes a Gaussian distribution defined over the variable x with mean m and covariance P.

The PHD measurement update equation for the extended target Poisson model of Gilholm et al. (2005) was derived in Mahler (2009). The corrected PHD-intensity is given by the multiplication of the predicted PHD and a measurement pseudo-likelihood function Mahler (2009) $L_{\mathbf{Z}_k}$ as

$$D_{k|k}(\mathbf{x}|\mathbf{Z}) = L_{\mathbf{Z}_k}(\mathbf{x})\, D_{k|k-1}(\mathbf{x}|\mathbf{Z}). \tag{6}$$

The measurement pseudo-likelihood function $L_{\mathbf{Z}_k}$ in (6) is defined as

$$\begin{aligned} L_{\mathbf{Z}_k}(\mathbf{x}) \triangleq & \left(1 - e^{-\gamma(\mathbf{x})}\right) p_{\mathrm{D}}(\mathbf{x}) + e^{-\gamma(\mathbf{x})} p_{\mathrm{D}}(\mathbf{x}) \\ & \times \sum_{\mathrm{p} \angle \mathbf{Z}_k} \omega_{\mathrm{p}} \sum_{W \in \mathrm{p}} \frac{\gamma(\mathbf{x})^{|W|}}{d_W} \cdot \prod_{\mathbf{z}_k \in W} \frac{\phi_{\mathbf{z}_k}(\mathbf{x})}{\lambda_k c_k(\mathbf{z}_k)}. \end{aligned} \tag{7}$$

where

- $\lambda_k \triangleq \beta_{FA,k} V_s$ is the mean number of clutter measurements;
- $c_k(\mathbf{z}_k) = 1/V_s$ is the spatial distribution of the clutter over the surveillance volume;
- the notation $\mathrm{p} \angle \mathbf{Z}_k$ means that p partitions the measurement set $\mathbf{Z}_k$ into non-empty cells W;
- the quantities ω_p and d_W are non-negative coefficients defined for each partition p and cell W respectively.
- $\phi_{\mathbf{z}_k}(\mathbf{x}) = p(\mathbf{z}_k|\mathbf{x})$ is the likelihood function for a single target generated measurement, which would be a Gaussian density in this work.

The first summation on the right hand side of (7) is taken over all partitions p of the measurement set $\mathbf{Z}_k$. The second summation is taken over all cells W in the current partition p.

In order to derive the measurement update of the GM-PHD-filter, six assumptions were made in Vo and Ma (2006), which are repeated here for the sake of completeness.

2 Assumption. Each target evolves and generates observations independently of one another.

3 Assumption. Clutter is Poisson and independent of target-originated measurements.

4 Assumption. The predicted multi-target RFS is Poisson.

5 Assumption. Each target follows a linear Gaussian dynamical model, cf. (1), and the sensor has a linear Gaussian measurement model, cf. (4).

6 Assumption. The survival and detection probabilities are state independent, i.e. $p_\mathrm{S}(\mathbf{x}) = p_\mathrm{S}$ and $p_\mathrm{D}(\mathbf{x}) = p_\mathrm{D}$.

7 Assumption. The intensities of the birth and spawn RFS are Gaussian-mixtures.

In this paper we adopt all of the above assumptions except that we relax the assumption on detection probability as follows.

8 Assumption. The following approximation about probability of detection function $p_\mathrm{D}(\cdot)$ holds.

$$p_\mathrm{D}(\mathbf{x})\,\mathcal{N}\left(\mathbf{x}\,|\,m_{k|k-1}^{(j)}, P_{k|k-1}^{(j)}\right) \approx p_\mathrm{D}\left(m_{k|k-1}^{(j)}\right)\mathcal{N}\left(\mathbf{x}\,|\,m_{k|k-1}^{(j)}, P_{k|k-1}^{(j)}\right) \tag{8}$$

for all $\mathbf{x}$ and for $j = 1, \ldots, J_{k|k-1}$.

Assumption 8 is weaker than Assumption 6 in that (8) is trivially satisfied when $p_{\mathrm{D}}(\cdot) = p_{\mathrm{D}}$, i.e. when $p_{\mathrm{D}}(\cdot)$ is constant. In general, Assumption 8 holds approximately when the function $p_{\mathrm{D}}(\cdot)$ does not vary much in the uncertainty zone of a target determined by the covariance $P^{(j)}_{k|k-1}$. This is true either when $p_{\mathrm{D}}(\cdot)$ is a sufficiently smooth function or when the signal to noise ratio (SNR) is high enough such that $P^{(j)}_{k|k-1}$ is sufficiently small. We still note here that Assumption 8 is only for the sake of simplification rather than approximation, since $p_{\mathrm{D}}(\mathbf{x})$ can always be approximated as a mixture of exponentials of quadratic functions (or equivalently as Gaussians) without losing the Gaussian-mixture structure of the corrected PHD, see Vo and Ma (2006). This, however, would cause a multiplicative increase in the number of components in the corrected PHD, which would in turn make the algorithm need more aggressive pruning and merging operations. A similar approach to variable probability of detection has been taken in order to model the *clutter notch* in ground moving target indicator target tracking Ulmke et al. (2007).

For the expected number of measurements from the targets, represented by $\gamma(\cdot)$, similar remarks apply and we use the following assumption.

9 Assumption. The following approximation about $\gamma(\cdot)$ holds

$$e^{-\gamma(\mathbf{x})}\gamma^n(\mathbf{x})\mathcal{N}\left(\mathbf{x} \,|\, m^{(j)}_{k|k-1}, P^{(j)}_{k|k-1}\right) \approx e^{-\gamma\left(m^{(j)}_{k|k-1}\right)}\gamma^n\left(m^{(j)}_{k|k-1}\right)\mathcal{N}\left(\mathbf{x} \,|\, m^{(j)}_{k|k-1}, P^{(j)}_{k|k-1}\right) \quad (9)$$

for all $\mathbf{x}$, $j = 1, \ldots, J_{k|k-1}$ and $n = 1, 2, \ldots$.

The trivial situation $\gamma(\cdot) = \gamma$, i.e. when $\gamma(\cdot)$ is constant, is again a special case where Assumption 9 is satisfied. In general, satisfying Assumption 9 is more difficult than Assumption 8 and a Gaussian mixture assumption for $\gamma(\cdot)$ would not work due to the exponential function. Nevertheless Assumption 9 is expected to hold approximately either when $\gamma(\cdot)$ is a sufficiently smooth function or when the signal to noise ratio (SNR) is high enough such that $P^{(j)}_{k|k-1}$ is sufficiently small.

With the assumptions presented above, the posterior intensity at time k is a Gaussian-mixture given by

$$D_{k|k}(\mathbf{x}) = D^{\mathrm{ND}}_{k|k}(\mathbf{x}) + \sum_{\mathrm{p}\angle\mathbf{Z}_k}\sum_{W\in\mathrm{p}} D^{\mathrm{D}}_{k|k}(\mathbf{x}, W). \quad (10)$$

The Gaussian-mixture $D^{\mathrm{ND}}_{k|k}(\cdot)$, handling the no detection cases, is given by

$$D^{\mathrm{ND}}_{k|k}(\mathbf{x}) = \sum_{j=1}^{J_{k|k-1}} w^{(j)}_{k|k}\mathcal{N}\left(\mathbf{x} \,|\, m^{(j)}_{k|k}, P^{(j)}_{k|k}\right), \quad (11a)$$

$$w^{(j)}_{k|k} = \left(1 - \left(1 - e^{-\gamma^{(j)}}\right)p^{(j)}_{\mathrm{D}}\right)w^{(j)}_{k|k-1}, \quad (11b)$$

$$m^{(j)}_{k|k} = m^{(j)}_{k|k-1}, \qquad P^{(j)}_{k|k} = P^{(j)}_{k|k-1}. \quad (11c)$$

where we used the short hand notations $\gamma^{(j)}$ and $p_{\mathrm{D}}^{(j)}$ for $\gamma\left(m_{k|k-1}^{(j)}\right)$ and $p_{\mathrm{D}}\left(m_{k|k-1}^{(j)}\right)$ respectively.

The Gaussian-mixture $D_{k|k}^{\mathrm{D}}(\mathbf{x}, W)$, handling the detected target cases, is given by

$$D_{k|k}^{\mathrm{D}}(\mathbf{x}, W) = \sum_{j=1}^{J_{k|k-1}} w_{k|k}^{(j)} \mathcal{N}\left(\mathbf{x} \,|\, m_{k|k}^{(j)}, P_{k|k}^{(j)}\right), \tag{12a}$$

$$w_{k|k}^{(j)} = \omega_{\mathrm{p}} \frac{\Gamma^{(j)} p_{\mathrm{D}}^{(j)}}{d_W} \Phi_W^{(j)} w_{k|k-1}^{(j)}, \tag{12b}$$

$$\Gamma^{(j)} = e^{-\gamma^{(j)}} \left(\gamma^{(j)}\right)^{|W|}, \tag{12c}$$

$$\Phi_W^{(j)} = \phi_W^{(j)} \prod_{\mathbf{z}_k \in W} \frac{1}{\lambda_k c_k(\mathbf{z}_k)}, \tag{12d}$$

where the product is over all measurements $\mathbf{z}_k$ in the cell W and $|W|$ is the number of elements in W. The coefficient $\phi_W^{(j)}$ is given by

$$\phi_W^{(j)} = \mathcal{N}\left(\mathbf{z}_W \,|\, \mathbf{H}_W m_{k|k-1}^{(j)}, \mathbf{H}_W P_{k|k-1}^{(j)} \mathbf{H}_W^{\mathrm{T}} + \mathbf{R}_W\right) \tag{12e}$$

and is calculated using

$$\mathbf{z}_W \triangleq \bigoplus_{\mathbf{z}_k \in W} \mathbf{z}_k, \qquad \mathbf{H}_W = [\underbrace{H_k^{\mathrm{T}}, H_k^{\mathrm{T}}, \cdots, H_k^{\mathrm{T}}}_{|W| \text{ times}}]^{\mathrm{T}},$$

$$\mathbf{R}_W = \mathrm{blkdiag}(\underbrace{R_k, R_k, \cdots, R_k}_{|W| \text{ times}}).$$

The operation $\bigoplus$ denotes vertical vectorial concatenation. The partition weights ω_{p} can be interpreted as the probability of the partition p being true and are calculated as

$$\omega_{\mathrm{p}} = \frac{\prod_{W \in \mathrm{p}} d_W}{\sum_{\mathrm{p}' \angle \mathbf{Z}_k} \prod_{W' \in \mathrm{p}'} d_{W'}}, \tag{12f}$$

$$d_W = \delta_{|W|,1} + \sum_{\ell=1}^{J_{k|k-1}} \Gamma^{(\ell)} p_{\mathrm{D}}^{(\ell)} \Phi_W^{(\ell)} w_{k|k-1}^{(\ell)}, \tag{12g}$$

where $\delta_{i,j}$ is the Kronecker delta. The mean and covariance of the Gaussian components are updated using the standard Kalman measurement update,

$$m_{k|k}^{(j)} = m_{k|k-1}^{(j)} + \mathbf{K}_k^{(j)} \left(\mathbf{z}_W - \mathbf{H}_W m_{k|k-1}^{(j)}\right), \tag{13a}$$

$$P_{k|k}^{(j)} = \left(I - \mathbf{K}_k^{(j)} \mathbf{H}_W\right) P_{k|k-1}^{(j)}, \tag{13b}$$

$$\mathbf{K}_k^{(j)} = P_{k|k-1}^{(j)} \mathbf{H}_W^{\mathrm{T}} \left(\mathbf{H}_W P_{k|k-1}^{(j)} \mathbf{H}_W^{\mathrm{T}} + \mathbf{R}_W \right)^{-1}. \tag{13c}$$

In order to keep the number of Gaussian components at a computationally tractable level, pruning and merging is performed as in Vo and Ma (2006).

4 Partitioning the Measurement Set

As observed in the previous section, an integral part of extended target tracking with the PHD filter is the partitioning of the set of measurements Mahler (2009). The partitioning is important, since more than one measurement can stem from the same target. Let us exemplify[1] the process of partitioning with a measurement set containing three individual measurements, $\mathbf{Z}_k = \left\{\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(2)}, \mathbf{z}_k^{(3)}\right\}$. This set can be partitioned in the following different ways;

$$\begin{aligned}
\mathrm{p}_1 &: W_1^1 = \left\{\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(2)}, \mathbf{z}_k^{(3)}\right\},\\
\mathrm{p}_2 &: W_1^2 = \left\{\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(2)}\right\},\ W_2^2 = \left\{\mathbf{z}_k^{(3)}\right\},\\
\mathrm{p}_3 &: W_1^3 = \left\{\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(3)}\right\},\ W_2^3 = \left\{\mathbf{z}_k^{(2)}\right\},\\
\mathrm{p}_4 &: W_1^4 = \left\{\mathbf{z}_k^{(2)}, \mathbf{z}_k^{(3)}\right\},\ W_2^4 = \left\{\mathbf{z}_k^{(1)}\right\},\\
\mathrm{p}_5 &: W_1^5 = \left\{\mathbf{z}_k^{(1)}\right\},\ W_2^5 = \left\{\mathbf{z}_k^{(2)}\right\},\ W_3^5 = \left\{\mathbf{z}_k^{(3)}\right\}.
\end{aligned}$$

Here, p_i is the ith partition, and W_j^i is the jth cell of partition i. Let $|\mathrm{p}_i|$ denote the number of cells in the partition, and let $|W_j^i|$ denote the number of measurements in the cell. It is quickly realized that as the size of the measurement set increases, the number of possible partitions grows very large. In order to have a computationally tractable target tracking method, only a subset of all possible partitions can be considered. In order to achieve good extended target tracking results, this subset of partitions must represent the most likely ones of all possible partitions.

In Section 4.1, we propose a simple heuristic for finding this subset of partitions, which is based on the distances between the measurements. We here note that our proposed method is only one instance of a vast number of other clustering algorithms found in the literature, and that other methods could have been used. Some well-known alternatives are pointed out, and compared to the proposed partitioning method, in Section 4.2. A modification of the partitioning approach to better handle targets which are spatially close is described in Section 4.3.

4.1 Distance Partitioning

Consider a set of measurements $\mathbf{Z} = \{\mathbf{z}^{(i)}\}_{i=1}^{N_\mathbf{z}}$. Our partitioning algorithm relies on the following theorem.

[1] This example was also utilized in Mahler (2009).

10 Theorem. *Let $\mathbf{d}(\cdot,\cdot)$ be a distance measure and $d_\ell \geq 0$ be an arbitrary distance threshold. Then there is one and only one partition in which any pair of measurements $\mathbf{z}^{(i)}, \mathbf{z}^{(j)} \in \mathbf{Z}$ that satisfy*

$$\mathbf{d}\left(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}\right) \leq d_\ell \tag{15}$$

are in the same cell.

Proof: The proof is given in Appendix A.1 for the sake of clarity. □

Given a distance measure $\mathbf{d}(\cdot,\cdot)$, the distances between each pair of measurements can be calculated as

$$\Delta_{ij} \triangleq \mathbf{d}(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}), \text{ for } 1 \leq i \neq j \leq N_{\mathbf{z}}. \tag{16}$$

Theorem 10 says that there is a unique partition that leaves all pairs (i,j) of measurements satisfying $\Delta_{ij} \leq d_\ell$ in the same cell. We give an example algorithm that can be used to obtain this unique partition in Algorithm 11. We use this algorithm to generate N_d alternative partitions of the measurement set $\mathbf{Z}$, by selecting N_d different thresholds

$$\{d_\ell\}_{\ell=1}^{N_d}, \; d_\ell < d_{\ell+1}, \text{ for } \ell = 1, \ldots, N_d - 1. \tag{17}$$

The alternative partitions contain fewer cells as the d_ℓ's are increasing, and the cells typically contain more measurements.

The thresholds $\{d_\ell\}_{\ell=1}^{N_d}$ are selected from the set

$$\mathcal{D} \triangleq \{0\} \cup \{\Delta_{ij} | 1 \leq i < j \leq N_{\mathbf{z}}\} \tag{18}$$

where the elements of $\mathcal{D}$ are sorted in ascending order. If one uses all of the elements in $\mathcal{D}$ to form alternative partitions, $|\mathcal{D}| = N_{\mathbf{z}}(N_{\mathbf{z}} - 1)/2 + 1$ partitions are obtained. Some partitions resulting from this selection might still turn out to be identical, and must hence be discarded so that each partition at the end is unique. Among these unique partitions, the first (corresponding to the threshold $d_1 = 0$) would contain $N_{\mathbf{z}}$ cells with one measurement each. The last partition would have just one cell containing all $N_{\mathbf{z}}$ measurements. Notice that this partitioning methodology already reduces the number of partitions tremendously.

The smallest and the largest thresholds in the set $\mathcal{D}$ on the other hand can still contain very similar values due to the fact that the measurements are generally clustered around the targets. In order to further reduce the computational load, partitions in this work are computed only for a subset of thresholds in the set $\mathcal{D}$. This subset is determined based on the statistical properties of the distances between the measurements belonging to the same target.

Suppose we select the distance measure $\mathbf{d}(\cdot,\cdot)$ as the Mahalanobis distance, given by

$$\mathbf{d}_M\left(\mathbf{z}_k^{(i)}, \mathbf{z}_k^{(j)}\right) = \sqrt{\left(\mathbf{z}_k^{(i)} - \mathbf{z}_k^{(j)}\right)^{\mathrm{T}} R_k^{-1} \left(\mathbf{z}_k^{(i)} - \mathbf{z}_k^{(j)}\right)}. \tag{19}$$

Algorithm 11 Distance Partitioning

```
Require: $d_\ell$, $\Delta_{i,j}$, $1 \leq i \neq j \leq N_{\mathbf{z}}$.
CellNumber($i$) = 0, $1 \leq i \leq N_{\mathbf{z}}$ {Set cells of all measurements to null}
CellId = 1 {Set the current cell id to 1}
%Find all cell numbers
for $i = 1 : N_{\mathbf{z}}$ do
    if CellNumbers($i$) = 0 then
        CellNumbers($i$) = CellId
        CellNumbers = FindNeighbors($i$,CellNumbers,CellId)
        CellId = CellId+1
    end if
end for
```

The recursive function FindNeighbors($\cdot$, $\cdot$, $\cdot$) is given as

```
function CellNumbers = FindNeighbors($i$,CellNumbers,CellId)
for $j = 1 : N_{\mathbf{z}}$ do
    if $j \neq i$ & $\Delta_{ij} \leq d_\ell$ & CellNumbers($j$) = 0 then
        CellNumbers($j$) = CellId
        CellNumbers = FindNeigbors($j$,CellNumbers,CellId)
    end if
end for
```

Then, for two target-originated measurements $\mathbf{z}_k^{(i)}$ and $\mathbf{z}_k^{(j)}$ belonging to the same target, $\mathbf{d}_M\left(\mathbf{z}_k^{(i)}, \mathbf{z}_k^{(j)}\right)$ is χ^2 distributed with degrees of freedom equal to the measurement vector dimension. Using the inverse cumulative χ^2 distribution function, which we denote here as invchi2($\cdot$), a unitless distance threshold δ_{P_G} can be computed as

$$\delta_{P_\mathrm{G}} = \mathrm{invchi2}(P_\mathrm{G}), \tag{20}$$

for a given probability P_G. We have seen in early empirical simulations that good target tracking results are achieved with partitions computed using the subset of distance thresholds in $\mathcal{D}$ satisfying the condition $\delta_{P_\mathrm{L}} < d_\ell < \delta_{P_\mathrm{U}}$ for lower probabilities $P_\mathrm{L} \leq 0.3$ and upper probabilities $P_\mathrm{U} \geq 0.8$.

As a simple example, if there are four targets present, each with expected number of measurements 20, and clutter measurements are generated with $\beta_{FA} V_s = 50$, then the mean number of measurements collected each time step would be 130. For 130 measurements, the number of all possible partitions is given by the Bell number $B_{130} \propto 10^{161}$ Rota (1964). Using all of the thresholds in the set $\mathcal{D}$, 130 different partitions would be computed on average. Using the upper and lower probabilities $P_\mathrm{L} = 0.3$ and $P_\mathrm{U} = 0.8$, Monte Carlo simulations show that on average only 27 partitions are computed, representing a reduction of computational complexity several orders of magnitude.

4.2 Alternative Partitioning Methods

An alternative to using the proposed algorithm is to use a method which takes as input the final desired number of cells, denoted K, and then divides the set of measurements into K cells. The perhaps most well known example of such a method is K-means clustering, see e.g. the the textbooks Bishop (2006); Hastie et al. (2009). In the ET-GM-PHD-filter, one needs to generate alternative partitions, corresponding to values of K between a lower and an upper threshold, denoted K_{L} and K_{U}. While the values for the partitioning parameters $\delta_{P_{\text{L}}}$ and $\delta_{P_{\text{U}}}$ in distance partitioning can be chosen using some intuitive arguments as above, it is less clear how K_{L} and K_{U} should be selected. One idea is to set $K_{\text{L}} = 1$ and $K_{\text{U}} = |\mathbf{Z}_k|$, which corresponds to $\delta_{P_{\text{U}}} = \infty$ and $\delta_{P_{\text{L}}} = 0$ in Distance Partitioning. Doing so would significantly increase the computational complexity compared to Distance Partitioning, since a considerably higher number of partitions must be considered.

Another major difference between the suggested distance partitioning and K-means clustering is highlighted in Figure 1, which shows a measurement set that consists of $N_{\mathbf{z},k} = 13$ measurements, 10 of which are clustered in the northeast of the surveillance region and the other three are scattered individually. The intuitive way to cluster this set of measurements is into 4 clusters, which is achieved by Distance Partitioning using a distance threshold of about 25 m, as shown in the left plot of Figure 1. When there is a large number of measurements concentrated in one part of the surveillance area, as is the case in this example, K-means clustering tends to split those measurements into smaller cells, and merge remaining but far away measurements into large cells which is illustrated in the right plot of Figure 1.

The main reason behind this shortcoming of K-means is the initialization of the algorithm, where the initial cluster centers are chosen by uniform sampling. In order to overcome this problem, modifications to the standard K-means algorithm have been suggested, where initial clusters are chosen as separated as possible, see Arthur and Vassilvitskii (2007); Ostrovsky et al. (2006). This improved version of K-means is called K-means++.

In simulations, Distance Partitioning was compared to K-means (using MATLAB's `kmeans`) and K-means++ (using an implementation available online Sorber (2011)). The results show that K-means++ in fact outperforms K-means, however it still fails to compute the correct partitions much too often, except in scenarios with very low $\beta_{FA,k}$. This can be attributed to the existence of counter-intuitive local optima for the implicit cost function involved with K-means (or K-means++). Distance Partitioning on the other hand can handle both high and low $\beta_{FA,k}$ and always gives an intuitive and unique partitioning for a given d_ℓ.

Therefore, we argue that a hierarchical method, such as the suggested distance partitioning, should be preferred over methods such as K-means. However, it is important to note here again, that regarding partitioning of the measurement set, the contribution of the current work lies mainly not in the specific method that

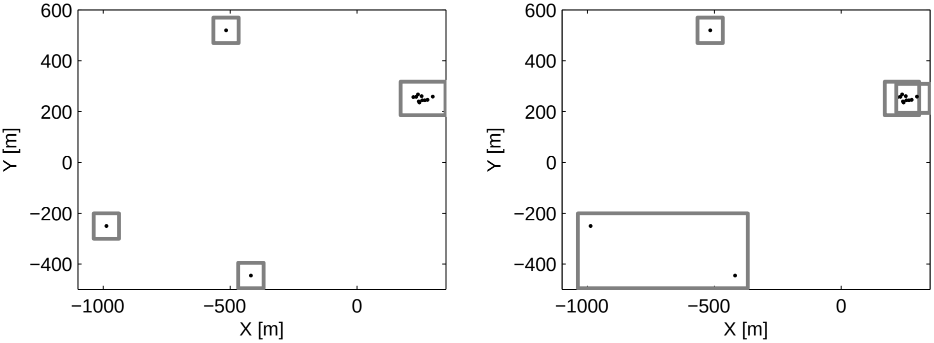


Figure 1: *Set of $N_{\mathbf{z},k} = 13$ measurements.* ***Left:*** *The measurements partitioned using the suggested distance partitioning method with a distance threshold of* $25\,\mathrm{m}$. ***Right:*** *The measurements partitioned using K-means clustering with $K = 4$.*

is suggested, but more in showing that all possible partitions can efficiently be approximated using a subset of partitions.

4.3 Sub-Partitioning

Initial results using ET-GM-PHD showed problems with underestimation of target set cardinality in situations where two or more extended targets are spatially close Granström et al. (2010). The reason for this is that when targets are spatially close, so are their resulting measurements. Thus, using Distance Partitioning, measurements from more than one measurement source will be included in the same cell W in all partitions p, and subsequently the ET-GM-PHD filter will interpret measurements from multiple targets as having originated from just one target. In an ideal situation, where one could consider all possible partitions of the measurement set, there would be alternative partitions which would contain the subsets of a wrongly merged cell. Such alternative partitions would dominate the output of the ET-GM-PHD filter towards the correct estimated number of targets. Since we eliminate such partitions completely using Distance Partitioning, the ET-GM-PHD filter lacks the means to correct its estimated number of targets.

One remedy for this problem is to form additional sub-partitions after performing Distance Partitioning, and to add them to the list of partitions that ET-GM-PHD filter considers at the current time step. Obviously, this should be done only when there is a risk of having merged the measurements belonging to more than one target, which can be decided based on the expected number of measurements originating from a target. We propose the following procedure for the addition of the sub-partitions.

Suppose that we have computed a set of partitions using Distance Partitioning, cf. the algorithm in Algorithm 11. Then, for each partition generated by the Distance Partitioning, say for p_i, we calculate the maximum likelihood (ML) estimates $\hat{N}_x^j$

Algorithm 12 Sub-Partition

```
Require: Partitioned set of measurements $\mathbf{Z}^p = \left\{\mathrm{p}_1, \ldots, \mathrm{p}_{N_\mathrm{p}}\right\}$, where $N_\mathrm{p}$ is the
    number of partitions.
Initialise: Counter for new partitions $\ell = N_\mathrm{p}$.
for $i = 1, \ldots, N_\mathrm{p}$ do
    for $j = 1, \ldots, |\mathrm{p}_i|$ do
        $\hat{N}_x^j = \arg\max_n \, p\left(\left|W_j^i\right| \middle| N_\mathbf{x}^j = n\right)$
        if $\hat{N}_x^j > 1$ then
            $\ell = \ell + 1$ {Increase the partition counter}
            $\mathrm{p}_\ell = \mathrm{p}_i \backslash W_j^i$ {Current partition except the current cell}
            $\left\{W_k^+\right\}_{k=1}^{\hat{N}_x^j} = \mathrm{split}\left(\hat{N}_x^j, W_j^i\right)$ {Split the current cell}
            $\mathrm{p}_\ell = \mathrm{p}_\ell \cup \left\{W_k^+\right\}_{k=1}^{\hat{N}_x^j}$ {Augment the current partition}
        end if
    end for
end for
```

of the number of targets for each cell W_j^i. If this estimate is larger than one, we split the cell W_j^i into $\hat{N}_x^j$ sub-cells, denoted as

$$\left\{W_s^+\right\}_{s=1}^{\hat{N}_x^j}. \tag{21}$$

We then add a new partition, consisting of the new sub-cells along with the other cells in p_i, to the list of partitions obtained by the Distance Partitioning.

We illustrate the sub-partitioning algorithm in Algorithm 12, where the splitting operation on a cell is shown by a function

$$\mathrm{split}\left(\hat{N}_x^j, W_j^i\right). \tag{22}$$

We give the details for obtaining the ML estimate $\hat{N}_x^j$ and choosing the function split$(\,\cdot\,,\,\cdot\,)$ in the subsections below.

Computing $\hat{N}_x^j$

For this operation, we assume that the function $\gamma(\,\cdot\,)$ determining the expected number measurements generated by a target is constant i.e., $\gamma(\,\cdot\,) = \gamma$. Each target generates measurements independently of the other targets, and the number of generated measurements by each target is distributed with the Poisson distribution, Pois$(\,\cdot\,, \gamma)$. The likelihood function for the number of targets corresponding to a cell W_j^i is given as

$$p\left(\left|W_j^i\right| \middle| N_\mathbf{x}^j = n\right) = \mathrm{Pois}\left(\left|W_j^i\right|, \gamma n\right). \tag{23}$$

Here, we assumed that the volume covered by a cell is sufficiently small such that the number of false alarms in the cell is negligible, i.e. there are no false alarms in W_j^i. The ML estimate $\hat{N}_x^j$ can now be calculated as

$$\hat{N}_x^j = \arg\max_n \, p\left(\left|W_j^i\right| \middle| N_{\mathbf{x}}^j = n\right). \tag{24}$$

Note that other alternatives can be found for calculating the estimates of $N_{\mathbf{x}}^j$, e.g. utilizing specific knowledge about the target tracking setup, however, both simulations and experiments have shown that the above suggested method works well.

The split (· , ·) Function

An important part of the Sub-Partition function in Algorithm 12 is the function split (· , ·), which is used to divide the measurements in a cell into smaller cells. In both simulations and experiments, we have used K-means clustering to split the measurements in the cell, results shows that this works well. Note however that other methods to split the measurements are possible.

11 Remark (Limitations of Sub-Partitioning). Notice that the Sub-Partition algorithm given in this section can be interpreted to be only a first-order remedy to the problem, and hence have limited correction capabilities. This is because we do not consider the combinations of the cells when we are adding sub-partitions. In the case, for example, where there are two pairs of close targets whose cells are merged wrongly by Distance Partitioning, the sub-partitioning algorithm presented above would add an additional partition for each of the target pairs (i.e. for each of the wrongly merged cells), but not an additional partition that contains the split versions of both cells. Consideration of all combinations of (the wrongly merged) cells seems again to be prohibitive, due to the combinatorial growth in the number of additional partitions. An idea for the cases where there can be more than one wrongly merged cells is to add a single additional partition, which contains split versions of all such cells.

5 Target Tracking Setup

The presented tracking approach is exemplified with a laser sensor tracking humans at short distance. In this section the tracking setup is defined for both a pure simulation environment and an experimental realisation with laser data. The targets are modeled as points with state variables

$$\mathbf{x}_k = \begin{bmatrix} \mathsf{x}_k & \mathsf{y}_k & \mathsf{v}_k^{\mathsf{x}} & \mathsf{v}_k^{\mathsf{y}} \end{bmatrix}^{\mathrm{T}}, \tag{25}$$

where x_k and y_k are the planar position coordinates of the target, and $\mathsf{v}_k^{\mathsf{x}}$ and $\mathsf{v}_k^{\mathsf{y}}$ are the corresponding velocities. The sensor measurements are given in batches of Cartesian x and y coordinates as follows;

$$\mathbf{z}_k^{(j)} \triangleq \begin{bmatrix} \mathsf{x}_k^{(j)} & \mathsf{y}_k^{(j)} \end{bmatrix}^{\mathrm{T}}. \tag{26}$$

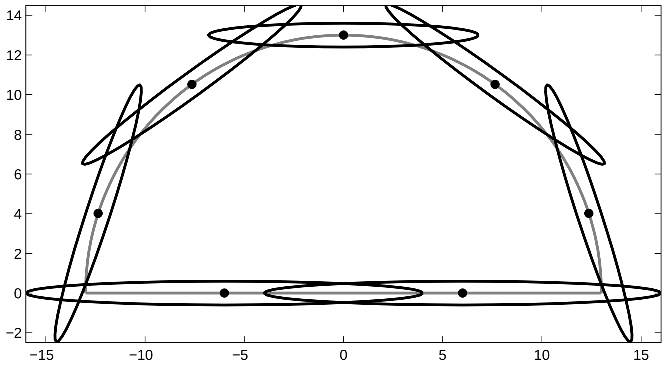


Figure 2: *Birth intensity used in experiments.*

Table 1: *Parameter values used for simulations (s) and experiments (e).*

	T	Q_k	R_k	$\gamma^{(i)}$	w_β	Q_β
S	1	$2^2\mathbf{I}_2$	$20^2\mathbf{I}_2$	10	0.05	$\text{blkdiag}((100\mathbf{I}_2, 400\mathbf{I}_2))$
E	0.2	$2^2\mathbf{I}_2$	$0.1^2\mathbf{I}_2$	12	0.01	$0.01\mathbf{I}_4$

A constant velocity model Rong Li and Jilkov (2003), with sampling time T is used. In all simulations the probability of detection and probability of survival are set as $p_\text{D} = 0.99$ and $p_\text{S} = 0.99$, respectively. The algorithm parameters for the simulation and experiment are given in Table 1. The surveillance area is $[-1000, 1000](\text{m}) \times [-1000, 1000](\text{m})$ for the simulations, and for the real data experiments the surveillance area is a semi circle located at the origin with range 13 m. In the simulations, clutter was generated with a Poisson rate of 10 clutter measurements per scan, and (unless otherwise stated) each target generated measurements with a Poisson rate of 10 measurements per scan. The birth intensity in the simulations is

$$D_b(\mathbf{x}) = 0.1\mathcal{N}(\mathbf{x}\,;\, m_b, P_b) + 0.1\mathcal{N}(\mathbf{x}\,;\, -m_b, P_b), \tag{27a}$$

$$m_b = [250, 250, 0, 0]^\text{T},\ P_b = \text{diag}([100, 100, 25, 25]). \tag{27b}$$

For the experiments, the birth intensity Gaussian components are illustrated with their corresponding one standard deviation ellipsoids in Fig. 2. Each birth intensity component has a weight $w_b^{(j)} = \frac{0.1}{J_b}$, where the number of components is $J_b = 7$. The spawn intensity is

$$D_\beta(\mathbf{x}|\mathbf{y}) = w_\beta\mathcal{N}(\mathbf{x}\,;\, \xi, Q_\beta), \tag{28}$$

where ξ is the target from which the new target is spawned and the values for w_β and Q_β are given in Table 1.

6 Simulation Results

This section presents the results from the simulations using the presented extended target tracking method. In Section 6.1 a comparison of the partitioning methods is presented, the results show the increased performance when using Sub-Partition. A comparison between ET-GM-PHD and GM-PHD is presented in Section 6.2, where it is shown that ET-GM-PHD as expected outperforms GM-PHD for extended targets. Section 6.3 presents a comparison of ET-GM-PHD and GM-PHD for targets that give rise to at most one measurement per time step. Finally, detailed investigations are carried out about the effects of the possibly unknown parameter γ in Section 6.4.

6.1 Partitioning Methods

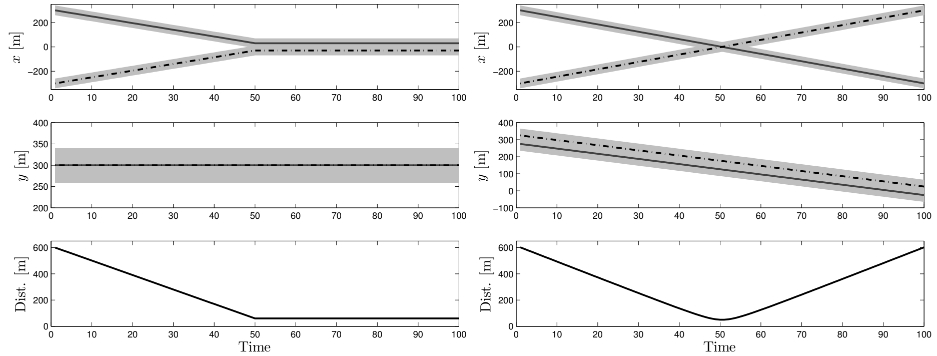
As was noted in Section 4.3, as well as in previous work Granström et al. (2010), using only Distance Partitioning to obtain a subset of all possible partitions is insufficient when the extended targets are spatially close. For this reason, Sub-Partition was introduced to obtain more partitions. In this section, we present the results from simulations that compare the performance of ET-GM-PHD tracking with partitions computed using only Distance Partitioning and with partitions computed using Distance Partitioning and Sub-Partition. Two scenarios are considered, both with two targets. The true x, y positions and the distance between the targets are shown in Fig. 3a and Fig. 3b. At the closest points the targets were 60m and 50m apart, respectively. In the simulations, the targets were modeled as points that generated measurements with standard deviation of 20m in both x and y direction. Thus, a measure of target extent can be taken as the two standard deviation measurement covariance ellipses, which here are circles of radius 40m. In both scenarios these circles partly overlap when the targets are closest to each other.

Each scenario was simulated 100 times with a constant expected number of measurements per target ($\gamma(\,\cdot\,) = \gamma$) of 5, 10 and 20, respectively. Fig. 4 shows the resulting sum of weights of the ET-GM-PHD algorithm. As can be seen, using Sub-Partition the average sum of weights is closer to the true number of targets. This is especially clear for targets that generate more measurements per time step, i.e. when γ is higher.

6.2 Comparison with GM-PHD

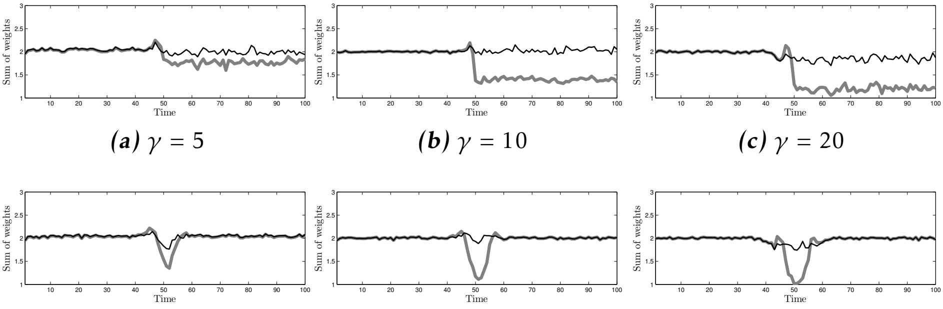
This section presents results from a simulation comparison of ET-GM-PHD and GM-PHD. Note here that the GM-PHD filter is applied naively to the simulated measurements, i.e. it is applied under the (false) assumption that each target produces at most one measurement per time step. The true targets are shown in black in Figure 5a, around time 50–52 two target tracks cross at a distance of just over 50m, at time 67 a new target is spawned 20m from a previous one. Thus the scenario presents challenges that are typical to multiple target applications.

To compare the ET-GM-PHD filter with the GM-PHD filter, 100 Monte Carlo simu-



(a) *(b)*

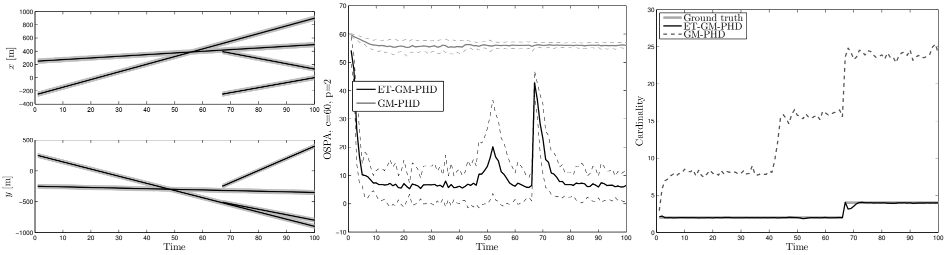
Figure 3: *Two simulation scenarios with spatially close targets. To the left, in (a), is a scenario where the two targets move closer to each other and then stand still at a distance of* 60m *apart. Note that the true* y *position was* 300m *for both targets for the entire simulation. To the right, in (b), is a scenario where the two targets cross paths, at the closest point they are* 50m *apart. The top and middle rows show the true* x *and* y *positions over time as a gray solid line and a black dash-dotted line. The light gray shaded areas show the target extent, taken as two measurement noise standard deviations (*40m*). The bottom row shows the distance between the two targets over time.*



(a) $\gamma = 5$ ***(b)*** $\gamma = 10$ ***(c)*** $\gamma = 20$

(d) $\gamma = 5$ ***(e)*** $\gamma = 10$ ***(f)*** $\gamma = 20$

Figure 4: *Simulation results for the two scenarios in Fig. 3, comparing different partitioning methods for different values of the expected number of measurements* γ*. The top row, (a), (b) and (c), is for the true tracks in Fig. 3a. The bottom row, (d), (e) and (f), is for the true tracks in Fig. 3b. Black shows Distance Partitioning with Sub-Partition, gray is only Distance Partitioning. It is clear from the plots that using Sub-Partition gives significantly better results, especially when* γ *is higher.*



(a) *(b)* *(c)*

Figure 5: *Results from multiple target tracking using synthetic data. (a) The true* x *and* y *positions are shown in black, the light gray shaded areas show the target extent, taken as two measurement noise standard deviations (*40m*). (b) Mean* OSPA *(solid lines)* ±1 *standard deviation (dashed lines). (c) Mean cardinality compared to the true cardinality.*

lations were performed, each with new measurement noise and clutter measurements. The results are shown in Figure 5b and Figure 5c, which show the corresponding multi-target measure optimal subpattern assignment metric (OSPA) Schuhmacher et al. (2008), and the cardinality, respectively. In the OSPA metric the parameters are set to $p = 2$, corresponding to using the 2-norm which is a standard choice, and $c = 60$, corresponding to a maximum error equal to three measurement noise standard deviations. Here, the cardinality is computed as $\sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)}$. This sum can be rounded to obtain an integer estimate of target number Vo and Ma (2006).

It is evident from the two figures that the presented ET-GM-PHD significantly outperforms the standard GM-PHD, which does not take into account the possibility of the multiple measurements from single targets. The main difference between the two filters is the estimation of cardinality, i.e. the number of targets. The ET-GM-PHD-filter correctly estimates the cardinality with the exception of when the new target is spawned – after time 67 there is a small dip in the mean estimated cardinality, even though Sub-Partition is used. The reason for this is that the targets are 20m apart which is smaller than the effective target extent determined by the measurement standard deviation (20m). As in the previous section, assuming that the effective target extent is two standard deviations, which correspond to circular targets of 40m radius, at 20m distance the measurements overlap significantly and the probability that the new target's measurements were also generated by the old target, as computed in (12e), is large. As the targets separate, this probability decreases and the ET-GM-PHD filter recovers the correct cardinality. It should still be noted that, in reality, where the targets would probably be rigid bodies, this type of very close situation is highly unlikely and the results of the ET-GM-PHD filter with Sub-Partition would be close to those presented in Section 6.1.

Similar simulations have been performed which compare Distance Partitioning with K-means clustering. Over 1000 Monte Carlo simulations, the tracking results in terms of mean OSPA and mean cardinality are significantly better for distance partitioning, mainly due to the problem with K-means clustering shown in Fig. 1.

6.3 Standard Single Measurement Targets

This section investigates how ET-GM-PHD handles standard targets that produce at most one measurement per time step, in comparison to standard GM-PHD which is designed under the standard target measurement generation assumption. Note that the measurement set cardinality distribution (i.e. the probability mass function for the number of measurements generated by a target) for a standard target contains only a single nonzero element[2] at cardinality 1, which is impossible to model with a Poisson distribution underlying the ET-GM-PHD filter. Hence, the case where each target generates measurements whose number is Poisson distributed with rate $\gamma = 1$ is very different from the standard target measurement generation.

Four targets were simulated in 100 Monte Carlo simulations, and all the targets were separated, i.e. there were no track crossings or new target spawn. Initially, in the ET-GM-PHD filter, $\gamma^{(j)}$ are all set as $\gamma^{(j)} = 1$ in the comparison. The average sum of weights and the average number of extracted targets (obtained by rounding the weights to the nearest integer) for the algorithms are shown in Fig. 6a and Fig. 6b respectively. As is shown, the sum of weights and number of extracted targets are too large for the ET-GM-PHD filter. The reason for this is that when the expected number of measurements per target (i.e. $\gamma^{(j)}$) is small, the effective probability of detection[3]

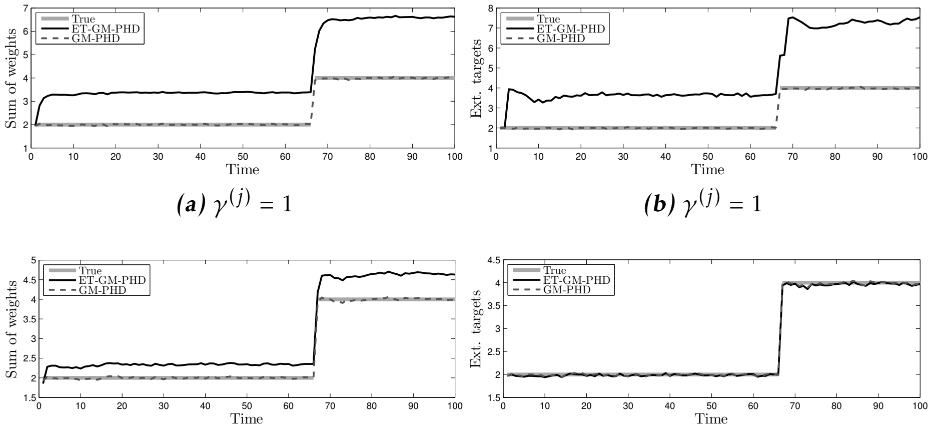
$$p_{\mathrm{D,eff}}^{(j)} = \left(1 - e^{-\gamma^{(j)}}\right) p_{\mathrm{D}}^{(j)} \tag{29}$$

becomes significantly smaller than one. For example, the case $\gamma^{(j)} = 1$ and $p_{\mathrm{D}}^{(j)} = 0.99$ gives $p_{\mathrm{D,eff}}^{(j)} = 0.6258$. This low effective probability of detection is what causes the weights in the ET-GM-PHD filter to become too large.

Actually, this problem has been seen to be inherited by the ET-GM-PHD filter from the standard PHD filter. We here give a simple explanation to the problem with low (effective) probability of detection in the PHD filter. Assuming no false alarms, and a single target with existence probability p_{E}, ideally a single detection should cause the expected number of targets to be unity. However, applying the standard PHD formulae to this simple example, one can calculate the updated expected number of targets to be $1 + p_{\mathrm{E}}(1 - p_{\mathrm{D}})$ whose positive bias increases as p_{D} decreases. We have seen that when the (effective) probability of detection is

[2] Note that a standard target always generates a single measurement. Whether no measurements or a single measurement is obtained from the standard target is determined by the detection process.

[3] More correctly, $p_{\mathrm{D,eff}}^{(j)}$ in (29) is the probability of the event that at least one measurement from the (jth) target is obtained by the sensor.



(a) $\gamma^{(j)} = 1$

(b) $\gamma^{(j)} = 1$

(c) $\gamma^{(j)} = 2$

(d) $\gamma^{(j)} = 2$

Figure 6: *Simulation results, comparison of ET-GM-PHD and GM-PHD for standard targets that produce at most one measurement per time step. The top row shows results when the parameter $\gamma^{(j)}$ is set to one, the bottom row shows results when it is set to two. Due to the low effective probability of detection, the ET-GM-PHD weights become too large, resulting in sum of weights larger than the true number of targets. When each weight is rounded to the nearest integer to extract targets, results for $\gamma^{(j)} = 2$ gives the correct number of extracted targets.*

low, the increase in $\sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)}$ is a manifestation of this type of sensitivity of the PHD type filters.[4] A similar sensitivity issue is mentioned in Erdinc et al. (2009) for the case of no detection.

This problem can be overcome by increasing $\gamma^{(j)}$ slightly in the ET-GM-PHD filter, e.g. $\gamma^{(j)} = 2$ gives $p_{\mathrm{D,eff}}^{j} = 0.8560$ which gives sum of weights and number of extracted targets that better match the results from GM-PHD, see Fig. 6c and Fig. 6d. Using $\gamma^{(j)} = 3$ gives results that are more or less identical to GM-PHD, thus a conclusion that can be drawn is that when tracking standard targets with an ET-GM-PHD filter, the parameter $\gamma^{(j)}$ should not be set too small. The following subsection investigates the issue of selecting the parameter γ in more detail.

6.4 Unknown Expected Number of Measurements γ

In the simulations above, the parameters $\gamma = \gamma^{(j)}$ was assumed to be known *a priori*. Further, in Section 4.3 where Sub-Partitioning is presented, the knowledge of the Poisson rate γ was used to determine whether a cell should be split or not

[4] Some extreme versions of this phenomenon for lower P_{D} values are illustrated and investigated in detail in the recent work Orguner et al. (2011).

to create an additional partition. In this section, some scenarios where γ is not known *a priori* are investigated. For the sake of clarity, γ is used to denote the true Poisson rate with which measurements were generated, and $\hat{\gamma}$ is used to denote the corresponding parameter in the ET-GM-PHD-filter.

In many real world scenarios, the number of measurements generated by a target is dependent on the distance between the target and the sensor. Typically, the longer the distance, the lower the number of measurements generated by the targets. This is true for sensors such as the laser range sensor, the radar sensor and even cameras. Thus, it is of interest to evaluate the ET-GM-PHD-filter in a scenario where the number of generated measurements varies with the target to sensor distance. This is simulated in Section 6.4, where the ET-GM-PHD filter is compared for the cases when the parameter $\hat{\gamma}$ is constant, and when the parameter is modeled as distance varying. Section 6.4 presents results from simulations where the parameter $\hat{\gamma}$ is set incorrectly, and Section 6.4 presents results with targets of different sizes. Finally, Section 6.4 presents a discussion about the results from the simulations, and supplies some guidelines into the selection of $\hat{\gamma}$.

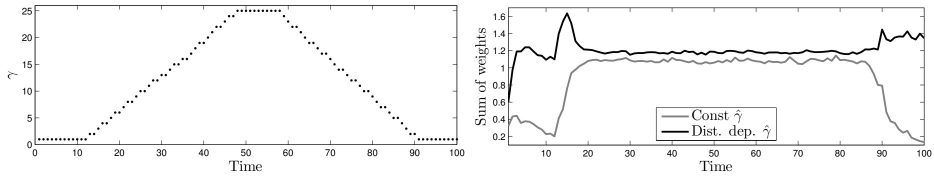
Distance Varying γ

A scenario was constructed where a target moved such that the target to sensor distance first decreased, and then increased. The sensor was simulated such that the true parameter γ depended on the target to sensor distance ρ as follows.

$$\gamma(\rho) = \begin{cases} 1, & \text{if } \rho > 400\text{m} \\ \lfloor -0.08\rho + 33.5 \rfloor, & \text{if } 100\text{m} \le \rho \le 400\text{m} \\ 25, & \rho < 100\text{m} \end{cases} \tag{30}$$

where $\lfloor \cdot \rfloor$ is the floor function. Thus, at distances larger than 400m, with $p_{\text{D}}^{(j)} = 0.99$, the effective probability of detection is only 0.6258 (as in the previous subsection). Note that the scenario is different from a target that always generates one measurement, which is detected with probability $p_{\text{D}}^{(j)} = 0.99$.

Monte Carlo simulations were made with two ET-GM-PHD-filters: one with constant value $\hat{\gamma} = 10$ and another where $\hat{\gamma}$ was selected to be dependent on the state of the targets via the function (30). The results are shown in Figure 7. For constant $\hat{\gamma}$, the number of targets is underestimated when the true γ is low. This is due to the fact that the filter expects a target to generate more measurements, and thus the likelihood that some small number of measurements are all clutter is higher. However, at distances ρ such that $\gamma(\rho) > 5$, $\hat{\gamma} = 10$ works quite well. When the model (30) for distance dependent γ is assumed known, the results are much more reasonable and acceptable. The only, and possibly negligible, drawback seems to be the number of targets being slightly overestimated. There are two main reasons for this. The first reason is the low effective probability of detection when $\hat{\gamma}$ is low. When $\hat{\gamma}$ becomes smaller than 5, this behavior is more evident. The second reason is that the clutter falling into the region $\Delta > 400$m (i.e., when the true parameter is $\gamma = 1$) is interpreted as targets to some extent, which causes a positive, though small, bias in the estimation of number of tar-



(a) *(b)*

Figure 7: *Results from the simulation scenario where γ is dependent on the target to sensor distance. In (a), the true γ is plotted against time, and in (b) the mean sum of weights is plotted against time. The ET-GM-PHD-filter is compared for the cases when the parameter $\hat{\gamma}$ is held constant (gray) or is set to the true distance dependent value (black). The correct target number is one, thus the sum of weights should be around one. In both cases, at the beginning and the end of the scenario when the distance is largest and $\gamma = 1$, tracking performance gets worse.*

gets. In that region, the target behavior is fairly similar to the clutter behavior which results in some Gaussian components with small weights surviving until the situation is resolved.

Incorrect γ Parameter

In this simulation study, the target tracks in Figure 3b were used. Each target generated measurements with a Poisson rate of $\gamma = 20$ and eleven different ET-GM-PHD-filters, each using a different $\hat{\gamma}$ value, were run. The set of $\hat{\gamma}$ values used is given as

$$\hat{\gamma} = 10, 12, \ldots, 28, 30. \tag{31}$$

The results, in terms of the sum of weights averaged over the Monte Carlo runs, are shown in Figure 8. The figure shows that for sufficiently separated targets, the ET-GM-PHD-filter correctly estimates the number of targets for all values of $\hat{\gamma}$. However, for spatially close targets, the ET-GM-PHD-filter overestimates the number of targets when $\hat{\gamma}$ is set too low, and underestimates the number of targets when $\hat{\gamma}$ is set too high. This result is actually expected, and is a direct consequence of the simple sub-partitioning algorithm we use. When $\hat{\gamma}$ is too low, Sub-Partitioning creates an additional partition with too many cells, causing the overestimation of number of targets. Conversely, when $\hat{\gamma}$ is too high, Sub-Partitioning does not create partitions with sufficient number of cells, causing the underestimation of number of targets. It is very important to note here that Sub-Partitioning operation actually runs even when the targets are well separated and does not cause any overestimation. Our observations show that this is a result of the fact that additional partitions created (when $\hat{\gamma}$ is selected too low) cannot win over single target partitions when the targets measurements are distributed in a blob shape. It is only when the two targets approach each other resulting in an eight-shaped cluster of measurements that the additional partition can gain

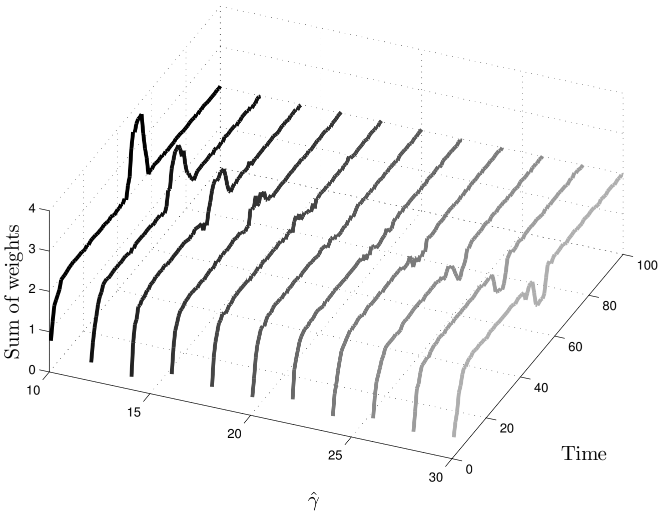


Figure 8: Simulation results for various values of the ET-GM-PHD-filter parameter $\hat{\gamma}$. There are two targets, the true Poisson rate used to generate measurements for both targets was $\gamma = 20$.

dominance over the single target partition. This property, though not proved mathematically, is considered to be a manifestation of the Poisson property and the Gaussianity assumptions underlying the measurements.

If the cardinality estimates of the algorithms are rounded to the nearest integer, an interesting property observed with Figure 8 is that no cardinality error appears for the cases that satisfy

$$\hat{\gamma} - \sqrt{\hat{\gamma}} \leq \gamma \leq \hat{\gamma} + \sqrt{\hat{\gamma}}. \tag{32}$$

Thus, when the true parameter γ lies in the interval determined by the mean ($\hat{\gamma}$) ± one standard deviation ($\sqrt{\hat{\gamma}}$), cardinality is expected to be estimated correctly even for close targets.

Targets of Different Size

In many scenarios, it is possible to encounter multiple targets of different sizes, thus producing a different number of measurements. This means that two targets would not have the same Poisson rate γ. In this section, the results are presented for a scenario with two targets with measurement generating Poisson rates of 10 and 20, respectively. In Monte Carlo simulations, three ET-GM-PHD-filter were run with the parameter $\hat{\gamma}$ set to 10, 15 and 20, respectively. This corresponds to using either the true value of the smaller target, the mean of both, or the true value of the larger target. The results, in terms of average sum of weights over time are shown in Figure 9. When the targets are spatially separated, all three filters perform equally well. However, when the targets are spatially close, the target with $\hat{\gamma}$ set to the mean of the true γs performs better than the others.

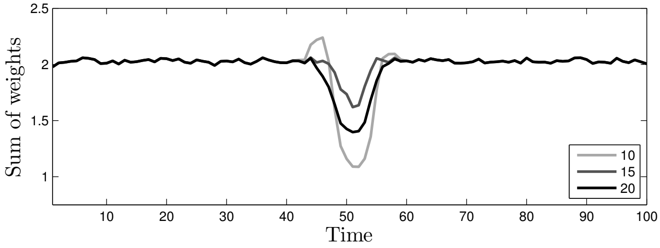


Figure 9: Simulation results from a scenario with two targets of different sizes. The two targets have the true Poisson rates $\gamma = 10$ and $\gamma = 20$, respectively. The legend refers to the filter parameter $\hat{\gamma}$.

Discussion

The simulation results above show that the ET-GM-PHD-filter works well even when $\hat{\gamma} \neq \gamma$, except when $\gamma < 5$ or targets are spatially close. For $\gamma < 5$, the filter is more sensitive, and a correct value for $\hat{\gamma}$ is increasingly important. For targets that are spatially close, it is important for $\hat{\gamma}$ to be a good estimate of γ, since the Sub-Partition algorithm relies on $\hat{\gamma}$. When such a good estimate is unavailable, a more advanced sub-partitioning algorithm seems to be necessary for robustness. With the proposed sub-partitioning procedure in this work, our findings support the intuitive conclusion that the true parameter γ should be in one standard deviation uncertainty region around the mean $\hat{\gamma}$ of the Poisson distribution for a reasonable performance for close targets.

The simulation with different target sizes shows that the close target case in this example is harder to tackle than the others. A possible solution is to adaptively estimate the parameters $\hat{\gamma}$ for each Gaussian mixture component based on the previous measurements. Another solution, which is possibly more straightforward, is to use a state dependent $\hat{\gamma}$ parameter, where the state contains information about the target extent, which can readily be estimated, see e.g. Granström et al. (2011b); Lundquist et al. (2011); Baum et al. (2010); Baum and Hanebeck (2011); Zhu et al. (2011). Using the estimated shape and size, and a model of the sensor that is used, $\hat{\gamma}$ can then be estimated with reasonable accuracy. This has indeed recently been performed using an ET-GM-PHD-filter Granström et al. (2011b).

7 Experiment Results

This section presents results from experiments with data from two data sets acquired with a laser range sensor. The experiments are included more as a proof of concept and as a potential application, rather than as an exhaustive evaluation of the presented target tracking filter. The measurements were collected using a SICK LMS laser range sensor. The sensor measures range every 0.5° over a 180° surveillance area. Ranges shorter than 13 m were converted to (x, y) measurements using a polar to Cartesian transformation.

The two data sets contain 411 and 400 laser range sweeps in total, respectively. During the data collection humans moved through the surveillance area, entering the surveillance area at different times. The laser sensor was at the waist level of the humans.

Since there is no ground truth available it is difficult to obtain a definite measure of target tracking quality, however by examining the raw data we were able to observe the true cardinality, which can thus be compared to the estimated cardinality.

Section 7.1 presents results from an experiment with spatially close targets, and Section 7.2 presents results from an experiment with both spatially close targets and occlusion.

7.1 Experiment with Close Targets

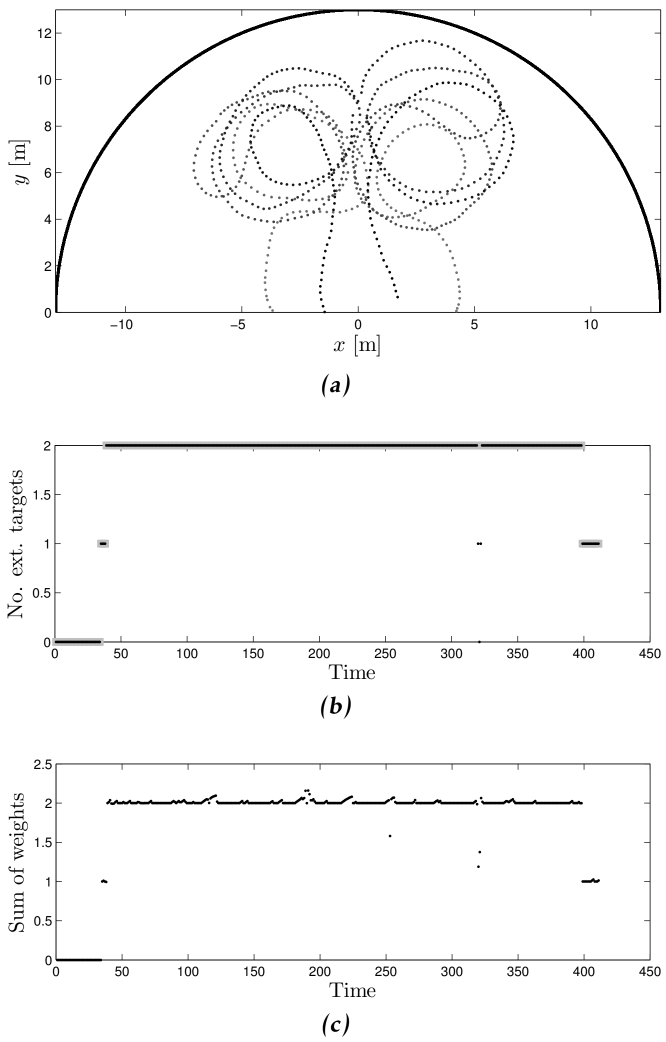
In this experiment, a data set containing 411 laser range scans was used. The data set contains two human targets that repeatedly move towards and away from each other, moving right next to each other at several times. The two targets passed each other at close distance moving in the opposite direction, representing instances in time when the targets were close for short periods of time. The targets also moved close to each other moving in the same direction, representing instances in time when the targets were close for longer periods of time.

The locations of the extracted Gaussian components are shown in Fig. 10a, the number of extracted targets is shown in Fig. 10b and the sum of weights are shown in Fig. 10c. Around time 320 there is a decrease in the number of extracted targets for three time steps, in all other situations the filter handles the two targets without problem. Thus, using Sub-Partition with K-means as split$(\cdot, \cdot)$ function, the ET-GM-PHD filter can be said to handle most of the spatially close target cases.

7.2 Experiment with Occlusion

In this experiment, a dataset containing 400 laser range scans was used. The data set contains four human targets that move through the surveillance area, however there are at most three targets present at any one time. The first target enters the surveillance area at time $k = 22$ and proceeds to the center of the surveillance area where he remains still for the remainder of the experiment. The second target enters the surveillance area at time $k = 38$ and repeatedly moves in front of and behind the first target. The third target enters and exits at time $k = 283$ and $k = 310$, respectively. The last target enters and exits at time $k = 345$ and $k = 362$, respectively.

This case requires a state dependent (i.e. variable) probability of detection $p_{\mathrm{D}}(\cdot)$ selection for the targets. Otherwise, i.e. with a constant probability of detection assumption, when a target is occluded, this would be interpreted as the exit of the target from the area of surveillance while it is only the disappearance of the target behind another. The variable p_{D} is modeled as a function of the mean,



(a)

(b)

(c)

Figure 10: *Experiment results, two human targets moving close to each other. Note that in (a) the gray scale indicates the time line, lighter gray is earlier time steps, darker is later time steps. In (b), the number of extracted targets (black) is compared to the ground truth (gray). In (c) the sum of weights is shown. Around time* 320 *the cardinality is underestimated for three time steps.*

covariance and the weights of the Gaussian components. The intuition behind this idea is that the knowledge of the targets that are present, i.e. the knowledge of the estimated Gaussian components of the PHD, can be used to determine what parts of the surveillance area are likely to be in view of the sensor, and which parts are not. Leaving the details of the variable p_{D} calculation to Appendix A.2, we present the results below.

The locations of the extracted Gaussian components are shown in Fig. 11a, the number of the extracted targets is shown in Fig. 11b and the sum of weights are shown in Fig. 11c. In total, there are six situations where one target is occluded by another. The extracted number of targets is wrong in two of these situations, where the occluded target is spatially very close to (right next to) the target which is causing the occlusion. The ET-GM-PHD filter correctly estimates the cardinality in four out of six occlusions.

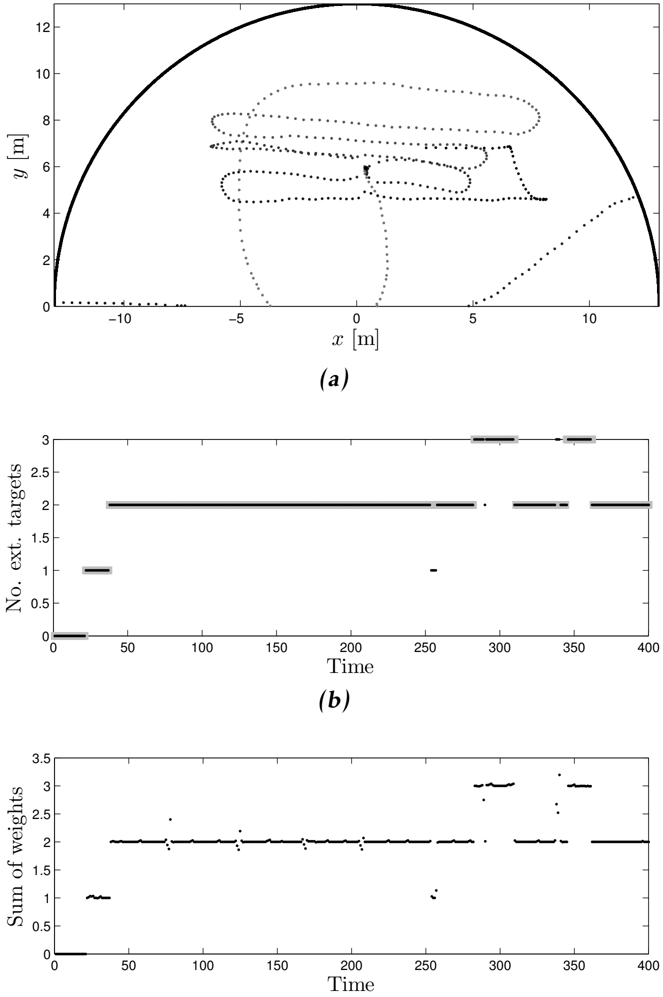
Thus, using the suggested variable p_{D}, the filter can correctly predict the target while it is occluded, provided that it is not very close to another target while the occlusion is happening. If $\sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)}$ is rounded to the nearest integer there is no cardinality error for the first four occlusions. However, as the target exits the occluded area there is a "jump" in $\sum_{j=1}^{J_{k|k}} w_{k|k}^{(j)}$ around times $k = 75$, $k = 125$, $k = 175$ and $k = 210$, see Fig. 11c. We have seen that this "jumping" behavior is caused by the sensitivity of the cardinality estimates of the PHD filter to detections when $p_{\mathrm{D}}^{(j)}$ is set to a low value, which is the case when the target is half occluded while it gets out of the occluded region. This is the same phenomenon observed with low effective probability of detection in Section 6.3.

8 Conclusions and Future Work

In this paper a Gaussian mixture implementation of the probability hypothesis density filter for tracking extended targets was presented. It was shown that all possible partitions of the measurement set does not need to be considered, instead it is sufficient to consider a subset of partitions, as long as this subset contain the most probable partitions. A simple method for finding this subset of all measurement set partitions was described. This partitioning method is complemented with a sub-partitioning strategy to handle the cases that involve close targets better. Simulations and experiments have shown that the proposed filter is capable of tracking extended targets in cluttered measurements. The number of targets is estimated correctly even for most of the cases when tracks are close. The detailed investigations carried out gave some guidelines about the selection of the Poisson rate for the cases when it is unknown. Using inhomogeneous detection probabilities in the surveillance region, it was shown that targets can be tracked as they move through occluded parts of the surveillance area.

In recent work, a cardinalized PHD filter Mahler (2007) for extended targets has been presented Orguner et al. (2011). This filter has less sensitive estimates of the number of targets. Initial step have also been taken towards including estimation of target extent in the ET-GM-PHD-filter Granström et al. (2011b). More work is needed in both of these research directions.

A further interesting research can be to see the potential use of the partitioning algorithms presented in this work with more conventional multiple target track-



(a)

(b)

(c)

Figure 11: *Experiment results, targets moving in and out of occluded regions of the surveillance area. Note that in (a) the gray scale indicates the time line, lighter gray is earlier time steps, darker is later time steps. In (b), the number of extracted targets (black) is compared to the ground truth (gray). In (c) the sum of weights is shown.*

ing algorithms. A comparison of such algorithms with the ET-GM-PHD filter can illustrate the advantage coming from the use of the random set framework.

Algorithm 13 Find partition p that satisfies the conditions of theorem 10

```
Require: Set of measurements $\mathbf{Z} = \left\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(N_\mathbf{z})}\right\}$, where $N_\mathbf{z}$ is the number of measurements.
1: Set $\mathrm{p}^0 = \left\{\{\mathbf{z}^{(1)}\}, \{\mathbf{z}^{(2)}\}, \ldots, \{\mathbf{z}^{(N_\mathbf{z})}\}\right\}$ i.e., set $W_j^0 = \{\mathbf{z}^{(j)}\}$ for $j = 1, \ldots, N_\mathbf{z}$.
2: Set $i = 1$.
3: Calculate all the pairwise distances between the cells of $\mathrm{p}_{i-1}$ as
       $$\eta_{st}^{i-1} = \min_{\substack{z_m \in W_s^{i-1} \\ z_n \in W_t^{i-1}}} \mathbf{d}\left(\mathbf{z}^{(m)}, \mathbf{z}^{(n)}\right) \qquad (33)$$
4: If $\min_{1 \leq s \neq t \leq |\mathrm{p}^{i-1}|} \eta_{st}^{i-1} > d_\ell$, then stop the algorithm, since $\mathrm{p}^{i-1}$ is a partition satisfying the conditions of the theorem.
5: Otherwise, combine all cells that satisfy $\eta_{st}^{i-1} \leq d_\ell$ to form a single cell.
6: Set $\mathrm{p}^i$ to be the set of cells obtained in Step 5.
7: Set $i = i + 1$ and go to Step 3.
```

A Appendix

A.1 Proof of Theorem 10

The proof is composed of two parts.

- We first prove that there is a partition satisfying the conditions of the theorem. The proof is constructive. Consider the algorithm in Algorithm 13.

  In the algorithm, one first forms a partition formed of singleton sets of the individual measurements and then combine the cells of this cluster until conditions of the theorem are satisfied. □

- We need to prove that the partition satisfying the conditions of the theorem is unique. The proof is by contradiction. Suppose that there are two partitions p_i and p_j satisfying the conditions of the theorem. Then, there must be a measurement $\mathbf{z}_m \in \mathbf{Z}$ such that the cells $W_{m_i}^i \ni \mathbf{z}_m$ and $W_{m_j}^j \ni \mathbf{z}_m$ are different, i.e., $W_{m_i}^i \neq W_{m_j}^j$. This requires an additional measurement $\mathbf{z}_n \in \mathbf{Z}$ that is in one of $W_{m_i}^i, W_{m_j}^j$ but not in the other. Without loss of generality, suppose $\mathbf{z}_n \in W_{m_i}^i$ and $\mathbf{z}_n \notin W_{m_j}^j$. Then since $\mathbf{z}_n \in W_{m_i}^i$, we know that $\mathbf{d}\left(\mathbf{z}^{(m)}, \mathbf{z}^{(n)}\right) \leq d_\ell$. However, this contradicts the fact that $\mathbf{z}_n \notin W_{m_j}^j$ which means that p_j does not satisfy the conditions of the theorem. Then our initial assumption that there are two partitions satisfying the conditions of the theorem must be wrong. The proof is complete. □

A.2 Variable Probability of Detection for the Laser Sensor

With the variable probability of detection function we reduce p_D behind (i.e. at larger range from the sensor) each component of the PHD according to the weight and bearing standard deviation of the Gaussian components.

For a given point $\mathbf{x}$ in the surveillance area, the probability of detection $p_{\mathrm{D}}(\mathbf{x})$ is computed as

$$\begin{aligned} p_{\mathrm{D}}(\mathbf{x}) &= \max\left(p_{\mathrm{D,min}}\,,\ p_{\mathrm{D}}^{\mathrm{v}}\right) \\ p_{\mathrm{D}}^{\mathrm{v}} &= p_{\mathrm{D},0} - \sum_{i:r>r^{(i)}} w^{(i)} \sqrt{\frac{\sigma_s}{\bar{\sigma}_{\varphi^{(i)}}}} \exp\left(\frac{-(\varphi-\varphi^{(i)})^2}{2\bar{\sigma}_{\varphi^{(i)}}}\right) \end{aligned} \tag{34}$$

where

- $p_{\mathrm{D,min}}$ is the minimum probability of detection value a target can have;
- $p_{\mathrm{D},0}$ is the nominal probability of detection of the targets when they are not occluded;
- r and φ are the range and bearing of point $\mathbf{x}$ with respect to the sensor respectively;
- $r^{(i)}$ and $\varphi^{(i)}$ are the range and bearing of the ith Gaussian component with respect to the sensor respectively;
- $w^{(i)}$ is the weight of the ith component;
- $\bar{\sigma}_{\varphi^{(i)}}$ is defined as

  $$\bar{\sigma}_{\varphi^{(i)}} \triangleq \begin{cases} \sigma_{\max}, & \text{if } \sigma_{\varphi^{(i)}} > \sigma_{\max} \\ \sigma_{\min}, & \text{if } \sigma_{\varphi^{(i)}} < \sigma_{\min} \\ \sigma_{\varphi^{(i)}}, & \text{otherwise} \end{cases} \tag{35}$$

  where $\sigma_{\varphi^{(i)}}$ is the bearing standard deviation of the ith component given as

  $$\sigma_{\varphi^{(i)}} \triangleq \sqrt{u_{\varphi^{(i)}}^{\mathrm{T}} P_p^{(i)} u_{\varphi^{(i)}}} \tag{36}$$

  Here, $P_p^{(i)}$ is the position covariance of the ith component and $u_{\varphi^{(i)}}$ is the unit vector orthogonal to the range direction from the ith component to the sensor.
- The constant term σ_s is used to scale the bearing standard deviation.

Intuitively, the operation of (34) is to reduce the nominal probability of detection at a point depending on the weights, means and standard deviations of the components of the last estimated PHD which have smaller range values than the range of the point also taking into account the angular proximity of such components to the bearing of the point in the exponential function in (34).

Bibliography

D. Arthur and S. Vassilvitskii. k-means++: The Advantages of Careful Seeding. In *Proceedings of the ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, Philadelphi, PA, USA, January 2007.

M. Baum and U. D. Hanebeck. Shape tracking of extended objects and group targets with star-convex RHMs. In *Proceedings of the International Conference on Information Fusion*, pages 338–345, Chicago, IL, USA, July 2011.

M. Baum, B. Noack, and U. D. Hanebeck. Extended object and group tracking with elliptic random hypersurface models. In *Proceedings of the International Conference on Information Fusion*, pages 1–8, Edinburgh, UK, July 2010.

C. M. Bishop. *Pattern recognition and machine learning*. Springer, New York, NY, USA, 2006. ISBN 0-387-31073-8.

Y. Boers, H. Driessen, J. Torstensson, M. Trieb, R. Karlsson, and F. Gustafsson. Track-before-detect algorithm for tracking extended targets. *IEE Proceedings of Radar, Sonar and Navigation*, 153(4):345–351, August 2006.

D. E. Clark, K. Panta, and B.-N. Vo. The GM-PHD Filter Multiple Target Tracker. In *Proceedings of the International Conference on Information Fusion*, Florence, Italy, July 2006.

O. Erdinc, P. Willett, and Y. Bar-Shalom. The bin-occupancy filter and its connection to the PHD filters. *IEEE Transactions on Signal Processing*, 57(11): 4232–4246, November 2009.

K. Gilholm and D. Salmond. Spatial distribution model for tracking extended objects. *IEE Proceedings of Radar, Sonar and Navigation*, 152(5):364–371, October 2005.

K. Gilholm, S. Godsill, S. Maskell, and D. Salmond. Poisson models for extended target and group tracking. In *Proceedings of Signal and Data Processing of Small Targets*, volume 5913, pages 230–241, San Diego, CA, USA, August 2005. SPIE.

K. Granström, C. Lundquist, and U Orguner. A Gaussian mixture PHD filter for extended target tracking. In *Proceedings of the International Conference on Information Fusion*, Edinburgh, UK, July 2010.

K. Granström, C. Lundquist, and U. Orguner. Extended target tracking using a Gaussian-mixture PHD filter. *IEEE Transactions on Aerospace and Electronic Systems*, 2011a. Under review.

K. Granström, C. Lundquist, and U Orguner. Tracking rectangular and elliptical extended targets using laser measurements. In *Proceedings of the International Conference on Information Fusion*, Chicago, IL, USA, July 2011b.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning :*

data mining, inference, and prediction. Springer, New York, NY, USA, 2 edition, 2009.

C. Lundquist, K. Granström, and U. Orguner. Estimating the shape of targets with a PHD filter. In *Proceedings of the International Conference on Information Fusion*, Chicago, IL, USA, July 2011.

R. P. S. Mahler. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, October 2003.

R. P. S. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, Boston, MA, USA, 2007.

R. P. S. Mahler. PHD filters for nonstandard targets, I: Extended targets. In *Proceedings of the International Conference on Information Fusion*, pages 915–921, Seattle, WA, USA, July 2009.

U. Orguner, C. Lundquist, and K. Granström. Extended target tracking with a cardinalized probability hypothesis density filter. In *Proceedings of the International Conference on Information Fusion*, pages 65–72, Chicago, IL, USA, July 2011.

R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy. The Effectiveness of Lloyd-Type Methods for the k-Means Problem. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 165–174, Berkeley, CA, USA, October 2006.

K. Panta, D. E. Clark, and Ba-Ngu Vo. Data association and track management for the Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Aerospace and Electronic Systems*, 45(3):1003–1016, July 2009.

X. Rong Li and V. P. Jilkov. Survey of maneuvering target tracking: Part I. Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4): 1333–1364, October 2003.

G.-C. Rota. The number of partitions of a set. *The American Mathematical Monthly*, 71(5):498–504, May 1964. ISSN 00029890.

D. Schuhmacher, B.-T. Vo, and B.-N. Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing*, 56 (8):3447–3457, August 2008.

L. Sorber. k-means++, 2011. URL `[http://www.mathworks.com/matlabcentral/fileexchange/28804-k-means++]`. Accessed 30-August-2011.

M. Ulmke, O. Erdinc, and P. Willett. Gaussian Mixture Cardinalized PHD Filter for Ground Moving Target Tracking. In *Proceedings of the International Conference on Information Fusion*, pages 1–8, Quebec City, Canada, July 2007.

B.-N. Vo and W.-K. Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, November 2006.

H. Zhu, C. Han, and C. Li. An extended target tracking method with random finite set observations. In *Proceedings of the International Conference on Information Fusion*, pages 73–78, Chicago, IL, USA, July 2011.

Paper F

Estimating the Shape of Targets with a PHD Filter

Authors: Christian Lundquist, Karl Granström and Umut Orguner

Edited version of the paper:

C. Lundquist, K. Granström, and U. Orguner. Estimating the shape of targets with a PHD filter. In *Proceedings of the International Conference on Information Fusion*, Chicago, IL, USA, July 2011a.

Estimating the Shape of Targets with a PHD Filter

Christian Lundquist, Karl Granström and Umut Orguner

Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
lundquist@isy.liu.se, karl@isy.liu.se, umut@isy.liu.se

Abstract

This paper presents a framework for tracking extended targets which give rise to a structured set of measurements per each scan. The concept of a measurement generating point (MGP) which is defined on the boundary of each target is introduced. The tracking framework contains an hybrid state space where MGPs and the measurements are modeled by random finite sets and target states by random vectors. The target states are assumed to be partitioned into linear and nonlinear components and a Rao-Blackwellized particle filter is used for their estimation. For each state particle, a probability hypothesis density (PHD) filter is utilized for estimating the conditional set of MGPs given the target states. The PHD kept for each particle serves as a useful means to represent information in the set of measurements about the target states. The early results obtained show promising performance with stable target following capability and reasonable shape estimates.

1 Introduction

In target tracking, the task is to detect, track and identify an unknown number of targets using measurements that are affected by noise and clutter. In recent years, so called extended target tracking has received increasing research attention. In extended target tracking, the classic point target assumption is relaxed and the target tracking framework is modified to handle multiple measurements per target and time step. The multiple measurements per target raise interesting possibilities to estimate the target's size and shape in addition to the target's position, velocity and heading. Typical sensors where targets cause multiple measurements are cameras, automotive radars and laser range sensors – especially cameras and laser range sensors give measurements with a high degree of structure.

Using finite set statistics (FISST) Mahler has derived rigorous tools for multiple target tracking, see Mahler (2003) and Mahler (2007). In the Probability Hypothesis Density filter (PHD-filter) the targets and measurements are treated as random finite sets (RFS); an implementation where the PHD-intensity is approximated using a mixture of Gaussians has been presented in Vo and Ma (2006). An extension of the PHD-filter to handle extended targets that give Poisson distributed measurements, as in Gilholm et al. (2005), was given in Mahler (2009). A Gaussian mixture implementation of this extended target PHD-filter was recently presented in Granström et al. (2010).

In the recent work by Mullane et al. Mullane et al. (2011), the Simultaneous Localization and Mapping (SLAM) problem was addressed using a Bayesian approach. The measurements and feature map is modeled using RFS, giving a framework in which the number and location of map features is jointly estimated with the vehicle trajectory. A Rao-Blackwellized implementation is suggested where the vehicle trajectory is estimated with a particle filter and the map is handled using a Gaussian-mixture (GM-PHD) filter.

In this paper, the ideas presented in Mullane et al. (2011) are utilized to solve the problem of estimating the shape of an extended target. The sensors mentioned earlier typically obtain point measurements from reflection points on the surface of the target. The realizations of the reflection points in this framework are called *measurement generating points* (MGPs) and their positions on the target are estimated as a means to estimate the shape, size and position of the target. By considering the MGP as an RFS it is possible to create a measurement model which better adapts to the actual and visible reflection points of the target. The target state vector is estimated using a particle filter, and the RFS of MGPs are estimated with a GM-PHD-filter. The target's state vector is too large to be efficiently estimated with a particle filter, and the linear and nonlinear parts are therefore partitioned and estimated in a marginalized or Rao-Blackwellized particle filter, see e.g., Schön et al. (2005). The joint estimation of the target density and the density of the MGPs on the boundary leads to a Rao-Blackwellized implementation of the joint particle and PHD filter.

Modeling of extended targets in this work is very similar to active contours Blake and Isard (1998) and snakes Kass et al. (1988), which model the outline of an object based on 2 dimensional image data, studied extensively in computer vision. Detecting and identifying shapes in cluttered point clouds has been studied in Srivastava and Jermyn (2009), where the MGPs on the surface of the target are denoted samples. The underlying low-level processing involved assumes reasonably the existence of image data from which features (like Harris corners) or feature maps (like Harris measures etc.) can be extracted. The so-called Condensation algorithm Isard and Blake (1998), for example, searches for suitable sets of features in the image data (or in the feature map) iteratively for each of its different hypotheses (particles) in the calculation of the likelihoods. The approach presented in this work carries the distinction of working only with a single set of measurement points provided most of the time by thresholding of the raw sensor data (like conventional target tracking) and hence is mainly aimed at applications where the users (of the sensors) either are unable to reach or do not have the computation power to work with the raw sensor data. The mapping of the boundaries of complex objects is also achieved in Lazarus et al. (2010) using splinegon techniques and a range sensor such as e.g., laser mounted on moving vehicles.

There are also several other approaches to extended target tracking in the literature. Gilholm presents in Gilholm and Salmond (2005); Gilholm et al. (2005) an approach where it is assumed that the number of received target measurements is Poisson distributed, hence several measurements may originate from the same target. In Boers et al. (2006) a similar approach is presented where raw data is considered in a track-before-detect framework and no data association is needed. Monte Carlo methods are applied to solve the extended target tracking problem in Vermaak et al. (2005); Angelova and Mihaylova (2008) and random matrices are used by Koch in Koch (2008).

The paper is organized as follows. Section 2 introduces the RFS of measurements and MGPs as well as the model of the target. The filter framework is described in Section 3 and an implementation is given in Section 4. A simple example with simulation results is shown in Section 5. Section 6 contains conclusions and some thoughts on future work.

2 The RFS Extended Target Model

Consider an extended target whose characteristics, e.g., position, heading, velocity and shape are described by a state vector $\mathbf{x}_k$. The target is observed by a sensor, which provides point observations $\mathbf{z}_k^{(m)}$ of the target. The number of measurements z_k at any given time is not fixed due to detection uncertainty, spurious measurements and unknown number of reflection points on the surface of the target. Since the target is extended, it potentially gives rise to more than one measurement per time step. Hence, the measurements obtained from one target may be represented by a random finite set (RFS) of observations

$$\mathbf{Z}_k = \left\{\mathbf{z}_k^{(1)}, \dots, \mathbf{z}_k^{(z_k)}\right\} \tag{1}$$

where k refers to discrete time. One way to associate the point measurements to the target is to consider a number of reflection points, called measurement generating points (MGP) in this work, on the surface of the target. A MGP is defined on a one dimensional coordinate axis denoted as $\mathbf{s} \in \mathbb{R}$ on the boundary of the target, which can generally be restricted to an interval $[\mathbf{s}_{\min}, \mathbf{s}_{\max}]$ called the MGP-space. All MGPs belonging to a single target may be modeled as a finite set shown as

$$\mathbf{S}_k = \left\{\mathbf{s}_k^{(1)}, \dots, \mathbf{s}_k^{(s_k)}\right\}. \tag{2}$$

A simple example is shown in Figure 1, where visible MGPs and point measurements are illustrated.

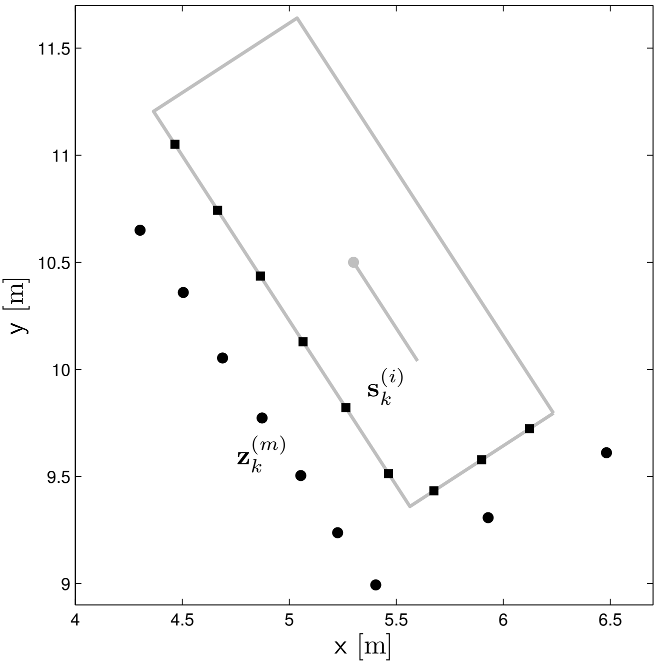


Figure 1: *A shape with visible MGPs $\mathbf{S}_k$ (■) and measurements $\mathbf{Z}_k$ (●).*

In a Bayesian framework, the states and the measurements are treated as realizations of random variables. Since in this case both the measurements and the MGPs are represented by finite sets, the concept of RFS is required for the Bayesian estimation of the target state. An RFS is a finite set valued random variable, whose elements are random variables and the number of elements are treated as a random non-negative integer variable. In a filtering and estimation framework the probability density is a very useful descriptor of an RFS. Standard tools and notation to compute densities and integrations of random variables are not appropriate for RFS and therefore finite set statistics FISST, developed by Mahler Mahler (2003) as a practical mathematical tool to deal with RFS, has to be applied.

Consider again the target in Figure 1, and let $\mathbf{S}$ be the RFS of all MGPs on the surface of the target. Furthermore let $\mathbf{S}_k \subset \mathbf{S}$ be the detectable MGPs. As the target moves new MGPs might become visible to the sensor, either by observing a new side of the object or by observing new reflection points, e.g., a wheel arch of a car, becoming visible when the angle between the target and the sensor changes. The new MGPs are modeled by an independent RFS $\mathbf{B}_k(\mathbf{x}_k) \subset \mathbf{S}$ depending on the target state $\mathbf{x}_k$. The set of visible MGPs evolves in time as

$$\mathbf{S}_k = \mathbf{F}_k(\mathbf{S}_{k-1}) \cup \mathbf{B}_k(\mathbf{x}_k). \tag{3}$$

where $\mathbf{F}_k(\cdot)$ is a set function modeling the survival (or death) of the previous visible MGPs. The RFS transition density is then given by

$$p(\mathbf{S}_k|\mathbf{S}_{k-1}, \mathbf{x}_k) = \sum_{\bar{\mathbf{S}}\subseteq\mathbf{S}_k} p_{\mathbf{S}}(\bar{\mathbf{S}}|\mathbf{S}_{k-1}) p_{\mathbf{B}}(\mathbf{S}_k - \bar{\mathbf{S}}|\mathbf{x}_k) \tag{4}$$

where $p_{\mathbf{S}}(\,\cdot\,|\mathbf{S}_{k-1})$ is the transition density of the observable set of MGPs, i.e., those who are in the field of view of the sensor, and $p_{\mathbf{B}}(\,\cdot\,|\mathbf{x}_k)$ is the density of the RFS $\mathbf{B}_k(\mathbf{x}_k)$ of the new visible MGPs. Furthermore, the target dynamics is modeled by a standard Markov process with transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and the joint transition density of the target and its MGPs is

$$p(\mathbf{S}_k, \mathbf{x}_k|\mathbf{S}_{k-1}, \mathbf{x}_{k-1}) = p(\mathbf{S}_k|\mathbf{S}_{k-1}, \mathbf{x}_k) p(\mathbf{x}_k|\mathbf{x}_{k-1}). \tag{5}$$

The RFS of measurements $\mathbf{Z}_k$ received at time k by the sensor from a target with state vector $\mathbf{x}_k$ and RFS $\mathbf{S}_k$ of MGP is modeled as

$$\mathbf{Z}_k = \bigcup_{\mathbf{s}\in\mathbf{S}_k} \mathbf{H}_k(\mathbf{s}, \mathbf{x}_k) \cup \mathbf{C}_k \tag{6}$$

where $\mathbf{H}_k(\mathbf{s}, \mathbf{x}_k)$ is the RFS of measurement produced by the MGPs modeling also the non-perfect detection process and $\mathbf{C}$ is the clutter RFS. The RFS of measurements produced by a MGP is modeled by a Bernoulli RFS, i.e., it is $\mathbf{H}_k(\mathbf{s}, \mathbf{x}_k) = \emptyset$ with probability $1 - P_D(\mathbf{s}|\mathbf{x}_k)$ and $\mathbf{H}_k(\mathbf{s}, \mathbf{x}_k) = \{\mathbf{z}\}$ with the probability density $P_D(\mathbf{s}|\mathbf{x}_k) p(\mathbf{z}|\mathbf{s}, \mathbf{x}_k)$, cf. Mahler (2007). The probability of detection is denoted by $P_D(\mathbf{s}|\mathbf{x}_k)$ defined over the MGP-space, and $p(\mathbf{z}|\mathbf{s}, \mathbf{x}_k)$ is the likelihood that the MGP $\mathbf{s}$ produces the measurement $\mathbf{z}$. The likelihood function of the measurement RFS $\mathbf{Z}_k$ is

$$p(\mathbf{Z}_k|\mathbf{x}_k, \mathbf{S}_k) = \sum_{\bar{\mathbf{Z}}\subseteq\mathbf{Z}_k} p_{\mathbf{Z}}(\bar{\mathbf{Z}}|\mathbf{S}_k, \mathbf{x}_k) p_{\mathbf{C}}(\mathbf{Z}_k - \bar{\mathbf{Z}}|\mathbf{x}_k) \tag{7}$$

where $p_{\mathbf{Z}}(\bar{\mathbf{Z}}|\mathbf{S}_k, \mathbf{x}_k)$ is the density representing the RFS of detected observations. The density of the clutter RFS $\mathbf{C}$ is denoted $p_{\mathbf{C}}(\,\cdot\,|\mathbf{x}_k)$ which is Poisson in cardinality and uniform in space.

The relation between the shape of the target and some of the state components is highly nonlinear and this makes an application of the particle filter suitable. Suppose that the target state $\mathbf{x}_k$ can be partitioned into a nonlinear and a linear part according to

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_k^n & \mathbf{x}_k^l \end{bmatrix}^{\mathrm{T}}, \tag{8}$$

the state space model for $\mathbf{x}_k$ is given by

$$\mathbf{x}_{k+1}^n = f_k^n(\mathbf{x}_k^n) + F_k^n(\mathbf{x}_k^n)\mathbf{x}_k^l + \mathbf{w}_k^n \tag{9a}$$

$$\mathbf{x}_{k+1}^l = f_k^l(\mathbf{x}_k^n) + F_k^l(\mathbf{x}_k^n)\mathbf{x}_k^l + \mathbf{w}_k^l. \tag{9b}$$

For a single measurement, the measurement model is given by

$$\mathbf{z}_k \sim \begin{cases} \mathcal{N}(\,\cdot\,; \hat{\mathbf{z}}_k(\mathbf{x}_k^n, \mathbf{s}_k), R_k) & \text{if } \mathbf{z} \text{ target generated} \\ c_k(\,\cdot\,) & \text{if } \mathbf{z} \text{ clutter generated} \end{cases} \tag{9c}$$

where

$$\hat{\mathbf{z}}_k(\mathbf{x}_k^n, \mathbf{s}_k) = h_k^n(\mathbf{x}_k^n, \mathbf{s}_k) + H_k(\mathbf{x}_k^n)\mathbf{x}_k^l \tag{9d}$$

The process noise $\mathbf{w}_k$ is assumed to be

$$\mathbf{w}_k = \begin{bmatrix} \mathbf{w}_k^n \\ \mathbf{w}_k^l \end{bmatrix} \sim \mathcal{N}(0, Q_k), \quad Q_k = \begin{bmatrix} Q_k^n & 0 \\ 0 & Q_k^l \end{bmatrix} \tag{10}$$

and the process covariance is here assumed to be diagonal. The measurement noise is modeled as $\mathbf{e}_k \sim \mathcal{N}(0; R_k)$. The MGPs $\mathbf{s}_k^{(i)}$ are assumed to be stationary on the boundary of the target and the motion model for them is selected to be a random walk model.

3 Filtering Framework

The aim of the filtering algorithm is to estimate $p(\mathbf{x}_{1:k}^n, \mathbf{x}_{1:k}^l, \mathbf{S}_k | \mathbf{Z}_{0:k})$, the joint posterior given the set of measurements $\mathbf{Z}_{0:k}$. The posterior may be factorized according to

$$p(\mathbf{x}_{1:k}^n, \mathbf{x}_k^l, \mathbf{S}_k | \mathbf{Z}_{0:k}) = p(\mathbf{x}_k^l | \mathbf{x}_{0:k}^n, \mathbf{S}_k, \mathbf{Z}_{0:k}) p(\mathbf{S}_k | \mathbf{x}_{0:k}^n, \mathbf{Z}_{0:k}) p(\mathbf{x}_{1:k}^n | \mathbf{Z}_{0:k}) \tag{11}$$

and the three posteriors for the states of a target $\mathbf{x}_k^l$, $\mathbf{x}_k^n$ and the MGPs $\mathbf{S}_k = \{\mathbf{s}_k^{(1)}, \dots, \mathbf{s}_k^{(\mathfrak{s}_k)}\}$ can be computed separately. However, since the filter recursion needed to estimate the probability density of the MGP RFS $p(\mathbf{S}_k | \mathbf{x}_{0:k}^n, \mathbf{Z}_{0:k})$ includes set integrals it is numerically intractable, and it is necessary to find a tractable approximation. The first order moment of an RFS is represented by a PHD, denoted D. For the RFS $\mathbf{S}_k$, the PHD is a nonnegative function, such that for each region S in the space of MGPs

$$\int_S D_k(\mathbf{x}) d\mathbf{x} = \mathrm{E}(|\mathbf{S}_k \cap S|) = \hat{\mathfrak{s}}_k \tag{12}$$

where $\hat{\mathfrak{s}}_k$ is the expected number of elements in the region S, and the peaks of the PHD indicates locations of MGPs with high probability. If the cardinality of elements in the RFS $\mathbf{S}_k$ is Poisson distributed and the elements are i.i.d., then the probability density of $\mathbf{S}_k$ can be computed from the PHD according to

$$p(\mathbf{S}_k) = \frac{\prod_{\mathbf{s} \in \mathbf{S}_k} D_k(\mathbf{s})}{\exp(\int D_k(\mathbf{s}) d\mathbf{s})}. \tag{13}$$

The PHD-filter propagates the PHD of an RFS in time and provides a reasonable approximation to the multitarget Bayesian filter. The filter recursion for the joint RFS and target state posterior density is described in the following. First the target state and the MGPs are predicted as described in section 3.1 and thereafter a measurement update of the MGPs and the target states is performed as described in Section 3.2.

3.1 Prediction

Given the prior

$$\begin{aligned} &p(\mathbf{x}^n_{1:k-1}, \mathbf{x}^l_{1:k-1}, \mathbf{S}_{k-1}|\mathbf{Z}_{0:k-1}) \\ &\quad = p(\mathbf{x}^l_{1:k-1}|\mathbf{x}^n_{0:k-1}, \mathbf{S}_{k-1}, \mathbf{Z}_{0:k-1})p(\mathbf{S}_{k-1}|\mathbf{x}^n_{0:k-1}, \mathbf{Z}_{0:k-1})p(\mathbf{x}^n_{1:k-1}|\mathbf{Z}_{0:k-1}) \end{aligned} \tag{14}$$

1. Predict first the nonlinear state components $\mathbf{x}^n$, i.e.,

$$p(\mathbf{x}^n_{1:k-1}|\mathbf{Z}_{0:k-1}) \to p(\mathbf{x}^n_{1:k}|\mathbf{Z}_{0:k-1}) \tag{15}$$

using (9a).

2. Predict the linear state components $\mathbf{x}^l$, i.e.,

$$p(\mathbf{x}^l_{1:k-1}|\mathbf{x}^n_{0:k-1}, \mathbf{Z}_{0:k-1}) \to p(\mathbf{x}^l_{1:k}|\mathbf{x}^n_{0:k}, \mathbf{Z}_{0:k-1}) \tag{16}$$

using (9a) and (9b). Compare with the marginalized particle filter Schön et al. (2005), where the linear components are predicted and updated with a Kalman filter, see Line 3-6 in Algorithm 14.

3. Finally predict the MGPs according to

$$p(\mathbf{S}_{k-1}|\mathbf{x}^n_{0:k}, \mathbf{Z}_{0:k-1}) \to p(\mathbf{S}_k|\mathbf{x}^n_{0:k}, \mathbf{Z}_{0:k-1}). \tag{17}$$

The predicted RFS is approximated by a Poisson RFS with PHD $D(\mathbf{s}_k|\mathbf{x}_{0:k}, \mathbf{Z}_{0:k-1})$ as

$$p(\mathbf{S}_k|\mathbf{x}_{0:k}, \mathbf{Z}_{0:k-1}) \approx \frac{\prod_{\mathbf{s}\in\mathbf{S}_k} D(\mathbf{s}|\mathbf{x}_{0:k}, \mathbf{Z}_{0:k-1})}{\exp(\int D(\mathbf{s}|\mathbf{x}_{0:k}, \mathbf{Z}_{0:k-1})d\mathbf{s})} \tag{18}$$

cf. (13). Using this approximation and the motion model (3), the PHD prediction equation is given by

$$D_{k|k-1}(\mathbf{s}|\mathbf{x}_{0:k}) = P_S(\mathbf{s})D_{k-1}(\mathbf{s}|\mathbf{x}_{0:k}) + D_{b,k}(\mathbf{s}) \tag{19}$$

where $D_{b,k}(\mathbf{s})$ is the PHD of the birth RFS $\mathbf{B}_k$ and $P_S(\cdot)$ models the probability of survival of a MGP. Note that the abbreviation

$$D_{k|k-1}(\mathbf{s}|\mathbf{x}_{0:k}) \triangleq D(\mathbf{s}_k|\mathbf{x}_{0:k}, \mathbf{Z}_{0:k-1}) \tag{20}$$

is adopted for clarity.

3.2 Measurement Update

4. Update the MGPs according to

$$p(\mathbf{S}_k|\mathbf{x}^n_{0:k}, \mathbf{Z}_{0:k-1}) \to p(\mathbf{S}_k|\mathbf{x}^n_{0:k}, \mathbf{Z}_{0:k}) \tag{21}$$

where the update is described by

$$p(\mathbf{S}_k|\mathbf{x}^n_{0:k},\mathbf{Z}_{0:k})=\frac{p(\mathbf{Z}_k|\mathbf{S}_k,\mathbf{x}^n_k)p(\mathbf{S}_k|\mathbf{x}^n_{0:k},\mathbf{Z}_{0:k-1})}{p(\mathbf{Z}_k|\mathbf{Z}_{0:k-1},\mathbf{x}^n_{0:k})}. \tag{22}$$

Like the case in the prediction, the posterior is approximated by a Poisson RFS with PHD $D(\mathbf{s}_k|\mathbf{x}_{0:k},\mathbf{Z}_{0:k})$ as

$$p(\mathbf{S}_k|\mathbf{x}_{0:k},\mathbf{Z}_{0:k})\approx\frac{\prod_{\mathbf{s}\in\mathbf{S}_k}D(\mathbf{s}|\mathbf{x}_{0:k},\mathbf{Z}_{0:k})}{\exp(\int D(\mathbf{s}|\mathbf{x}_{0:k},\mathbf{Z}_{0:k})d\mathbf{s})} \tag{23}$$

cf. (13). Using this approximation and the measurement model (6), the PHD corrector equation is given by

$$\begin{aligned}D_k(\mathbf{s}|\mathbf{x}_{0:k})=D_{k|k-1}(\mathbf{s}|\mathbf{x}_{0:k})\Bigg[&1-P_D(\mathbf{s}|\mathbf{x}_k)\\&+\sum_{\mathbf{z}\in\mathbf{Z}_k}\frac{\Lambda(\mathbf{z}_k|\mathbf{s}_k,\mathbf{x})}{\lambda c_k(\mathbf{z})+\int\Lambda(\mathbf{z}_k|\zeta,\mathbf{x}_k)D_{k|k-1}(\zeta|\mathbf{x}_{0:k})d\zeta}\Bigg]\end{aligned} \tag{24}$$

where

$$\Lambda(\mathbf{z}_k|\mathbf{s}_k,\mathbf{x}_k)=P_D(\mathbf{s}_k|\mathbf{x}_k)p(\mathbf{z}_k|\mathbf{s}_k,\mathbf{x}_k). \tag{25}$$

For clarity the abbreviation

$$D_k(\mathbf{s}|\mathbf{x}_{0:k})\triangleq D(\mathbf{s}_k|\mathbf{x}_{0:k},\mathbf{Z}_{0:k}) \tag{26}$$

is used.

5. Update the nonlinear state components $\mathbf{x}^n$

$$p(\mathbf{x}^n_{1:k}|\mathbf{Z}_{0:k-1})\rightarrow p(\mathbf{x}^n_{1:k}|\mathbf{Z}_{0:k}) \tag{27}$$

where the update is modeled as

$$p(\mathbf{x}^n_{1:k}|\mathbf{Z}_{0:k})=p(\mathbf{Z}_k|\mathbf{Z}_{0:k-1},\mathbf{x}^n_{0:k})\frac{p(\mathbf{x}^n_k|\mathbf{x}^n_{k-1})p(\mathbf{x}^n_{1:k-1}|\mathbf{Z}_{0:k-1})}{p(\mathbf{Z}_k|\mathbf{Z}_{0:k-1})}. \tag{28}$$

The term $p(\mathbf{Z}_k|\mathbf{Z}_{0:k-1},\mathbf{x}^n_{0:k})$ is given by a set integration over $\mathbf{S}_k$, according to

$$p(\mathbf{Z}_k|\mathbf{Z}_{0:k-1},\mathbf{x}^n_{0:k})=\int p(\mathbf{Z}_k,\mathbf{S}_k|\mathbf{Z}_{0:k-1},\mathbf{x}^n_{0:k})\delta\mathbf{S}_k. \tag{29}$$

To avoid the set integral, note that $p(\mathbf{Z}_k|\mathbf{Z}_{0:k-1},\mathbf{x}^n_{0:k})$ is the normalization constant in (22), hence it may be rewritten according to

$$p(\mathbf{Z}_k|\mathbf{Z}_{0:k-1},\mathbf{x}^n_{0:k})=\frac{p(\mathbf{Z}_k|\mathbf{S}_k,\mathbf{x}^n_k)p(\mathbf{S}_k|\mathbf{x}^n_{0:k},\mathbf{Z}_{0:k-1})}{p(\mathbf{S}_k|\mathbf{x}^n_{0:k},\mathbf{Z}_{0:k})}. \tag{30}$$

Note that $\mathbf{S}_k$ is only contained in the RHS, thus the relation holds for arbitrary choices of $\mathbf{S}_k$. We substitute here the last estimated $\mathbf{S}_k$ by making further approximations as follows. Using the Poisson RFS approximations (18) and (23) and the

RFS measurement likelihood (7) it holds that

$$
\begin{aligned}
p(\mathbf{Z}_k|\mathbf{Z}_{0:k-1}, \mathbf{x}^n_{0:k}) &= \frac{p(\mathbf{Z}_k|\mathbf{S}_k, \mathbf{x}^n_k)p(\mathbf{S}_k|\mathbf{x}^n_{0:k}, \mathbf{Z}_{0:k-1})}{p(\mathbf{S}_k|\mathbf{x}^n_{0:k}, \mathbf{Z}_{0:k})} \\
&\approx \underbrace{p(\mathbf{Z}_k|\mathbf{S}_k, \mathbf{x}^n_k)}_{\triangleq A} \frac{\frac{\prod_{\mathbf{s}\in\mathbf{S}_k} D_{k|k-1}(\mathbf{s}|\mathbf{x}_{0:k})}{\exp(\int D_{k|k-1}(\mathbf{s}|\mathbf{x}_{0:k})d\mathbf{s})}}{\frac{\prod_{\mathbf{s}\in\mathbf{S}_k} D_k(\mathbf{s}|\mathbf{x}_{0:k})}{\exp(\int D_k(\mathbf{s}|\mathbf{x}_{0:k})d\mathbf{s})}} \\
&= A \underbrace{\frac{\prod_{\mathbf{s}\in\mathbf{S}_k} D_{k|k-1}(\mathbf{s}|\mathbf{x}_{0:k})}{\prod_{\mathbf{s}\in\mathbf{S}_k} D_k(\mathbf{s}|\mathbf{x}_{0:k})}}_{\triangleq B} \underbrace{\frac{\exp(\int D_k(\mathbf{s}|\mathbf{x}_{0:k})d\mathbf{s})}{\exp(\int D_{k|-1}(\mathbf{s}|\mathbf{x}_{0:k})d\mathbf{s})}}_{\triangleq C} \qquad (31a)
\end{aligned}
$$

where

$$
A \approx \prod_{\mathbf{z}\in\mathbf{Z}} (\lambda c(\mathbf{z}) + P_D(\mathbf{s}^{(i)}_k)p(\mathbf{z}^{(i)}|\mathbf{s}^{(i)}_k, \mathbf{x}^n_k)) \prod_{\mathbf{s}\in\bar{\mathbf{S}}_k} (1 - P_D(\mathbf{s}^{(i)}_k)) \qquad (31b)
$$

with the ith MGP being the one closest to measurement i according to

$$
\begin{aligned}
\mathbf{s}^{(i)} &= \arg\min_{\mathbf{s}\in\mathbf{S}_k} \|\mathbf{z}^{(i)}_k - \mathbf{s}\| \qquad (31c) \\
\bar{\mathbf{S}}_k &= \mathbf{S}_k \backslash \{\mathbf{s}^{(i)}\}_{i=1}^{\mathfrak{z}_k} \qquad (31d)
\end{aligned}
$$

Furthermore,

$$
\begin{aligned}
B &= \prod_{i=1}^{\hat{\mathfrak{s}}_k} \frac{1}{(1 - P_D(\mathbf{s}^{(i)})) + \sum_{\mathbf{z}\in\mathbf{Z}} \frac{\Lambda(\mathbf{z}|\mathbf{s}^{(i)}_k, \mathbf{x}_k)}{\lambda c(\mathbf{z}) + \int \Lambda(\mathbf{z}|\zeta, \mathbf{x}_k) D_k(\zeta|\mathbf{x}_{0:k}) d\zeta}} \qquad (31e) \\
C &= \exp(\hat{\mathfrak{s}}_k - \hat{\mathfrak{s}}_{k|k-1}) \qquad (31f)
\end{aligned}
$$

and $\Lambda(\,\cdot\,)$ is given in (25). The number of predicted and updated MGPs are

$$
\begin{aligned}
\hat{\mathfrak{s}}_k &= \int D_k(\mathbf{s}_k|\mathbf{x}_{0:k}, \mathbf{Z}_{0:k}) d\mathbf{s}_k \qquad (32a) \\
\hat{\mathfrak{s}}_{k|k-1} &= \int D_k(\mathbf{s}_k|\mathbf{x}_{0:k}, \mathbf{Z}_{0:k-1}) d\mathbf{s}_k. \qquad (32b)
\end{aligned}
$$

6. Update the linear state components $\mathbf{x}^l$

$$
p(\mathbf{x}^l_{1:k}|\mathbf{x}^n_{0:k}, \mathbf{S}_k, \mathbf{Z}_{0:k-1}) \rightarrow p(\mathbf{x}^l_{1:k}|\mathbf{x}^n_{0:k}, \mathbf{S}_k, \mathbf{Z}_{0:k}) \qquad (33)
$$

The update of $\mathbf{x}^l_{1:k}$ is conditioned on the RFS of the MGPs $\mathbf{S}_k$. As an approximation, the posterior of $\mathbf{x}^l$ is written as

$$
p(\mathbf{x}^l_{1:k}|\mathbf{x}^n_{0:k}, \mathbf{S}_k \mathbf{Z}_{0:k}) \approx p(\mathbf{x}^l_{1:k}|\mathbf{x}^n_{0:k}, \widehat{\mathbf{S}}_k, \mathbf{Z}_{0:k}) \qquad (34)
$$

where the estimate $\widehat{\mathbf{S}}_k$ is calculated from the posterior PHD $D_k(\mathbf{s}|\mathbf{x}_{0:k})$. The linear components are updated as described in Line 46 to 51 in Algorithm 14.

7. The joint posterior is now given by multiplying the separate posteriors from (21), (27) and (33) as

$$p(\mathbf{x}^n_{1:k}, \mathbf{x}^l_{1:k}, \mathbf{S}_k | \mathbf{Z}_{0:k}) \approx p(\mathbf{x}^l_{1:k} | \mathbf{x}^n_{0:k}, \widehat{\mathbf{S}}_k, \mathbf{Z}_{0:k}) p(\mathbf{S}_k | \mathbf{x}^n_{0:k}, \mathbf{Z}_{0:k}) p(\mathbf{x}^n_{1:k} | \mathbf{Z}_{0:k}) \tag{35}$$

An implementation of the proposed filter recursion is presented in the next section.

4 RBPF-PHD Implementation

A Rao-Blackwellized implementation is utilized to estimate the target state. The nonlinear target state $\mathbf{x}^n$ is propagated with a particle filter and the linear state vector is propagated with a Kalman filter. The MGPs are described by a set $\mathbf{S}_k$, which are estimated with a PHD-filter. There exists one PHD $D^{(i)}_k(\cdot)$ for each particle i of the target state. The overall summary statistics is represented by

$$\left\{ w^{(i)}_k, \mathbf{x}^{n,(i)}_{0:k}, \hat{\mathbf{x}}^{l,(i)}_k, P^{l,(i)}_k, D^{(i)}_k(\mathbf{s}|\mathbf{x}^{(i)}_{0:k}) \right\}_{i=1}^N \tag{36}$$

where

- $\mathbf{x}^{n,(i)}_{0:k}$ is the ith particle for the nonlinear part of the target state.
- $\hat{\mathbf{x}}^{l,(i)}_k$, $P^{l,(i)}_k$ are the ith mean and covariance for the linear part of the target state.
- $w^{(i)}_k$ is the weights for the ith particle of the target state.
- $D^{(i)}_k(\mathbf{s}|\mathbf{x}^{(i)}_{0:k})$ is the PHD representing the MGPs for the ith particle of the target state.

In this work the PHD $D^{(i)}_k(\mathbf{s}|\mathbf{x}^{(i)}_k)$ is represented with by a Gaussian mixture and a realization of the GM-PHD filter recursion is described in Section 4.1, see also Vo and Ma (2006). The update of the nonlinear state components based on the PHD is described in Section 4.2, and a pseudo code for the proposed filter is given in Section 4.3.

4.1 PHD Prediction and Update

The MGPs on the surface of the ith particle are approximated with a PHD $D^{(i)}_k(\mathbf{s}|\mathbf{x}_k)$ represented by a Gaussian mixture. The prior PHD is given by

$$D^{(i)}_{k-1}(\mathbf{s}|\mathbf{x}^{(i)}_{k-1}) = \sum_{j=1}^{J^{(i)}_{k-1}} \eta^{(i,j)}_{k-1} \mathcal{N}\left(\mathbf{s}, \mu^{(i,j)}_{k-1}, P^{(i,j)}_{k-1}\right). \tag{37}$$

This is a mixture of $J_{k-1}^{(i)}$ Gaussian components, with $\eta_k^{(i,j)}$, $\mu_k^{(i,j)}$ and $P_k^{(i,j)}$ being the weights, means and covariances, respectively, for the jth Gaussian component of the ith particle.

The prediction of MGPs in the motion model (3) is given by the union of prior MGPs and new MGP, which is approximated by the PHD prediction equation (19). The resulting predicted PHD represented by a Gaussian mixture is then given as

$$D_{k|k-1}^{(i)}(\mathbf{s}|\mathbf{x}_k^{(i)}) = \sum_{j=1}^{J_{k|k-1}^{(i)}} \eta_{k|k-1}^{(i,j)} \mathcal{N}\left(\mathbf{s}, \mu_{k|k-1}^{(i,j)}, P_{k|k-1}^{(i,j)}\right). \tag{38}$$

where $J_{k|k-1}^{(i)} = J_{k-1}^{(i)} + J_{b,k}^{(i)}$ and $J_{b,k}^{(i)}$ is the number of new Gaussian components. The posterior PHD is computed in the GM form as

$$D_k^{(i)}(\mathbf{s}|\mathbf{x}_k^{n,(i)}) = D_{k|k-1}^{n,(i)}(\mathbf{s}|\mathbf{x}_k^{n,(i)})\left(1 - p_D(\mathbf{s}|\mathbf{x}_k^{n,(i)})\right) + \sum_{\mathbf{z}\in\mathbf{Z}_k} \sum_{j=1}^{J_{k|k-1}^{(i)}} D_{G,k}^{(i,j)}(\mathbf{z},\mathbf{s}|\mathbf{x}_k^{n,(i)}) \tag{39}$$

where the components are given by

$$D_{G,k}^{(i,j)}(z,\mathbf{s}|\mathbf{x}_k^{n,(i)}) = \eta_{k|k}^{(i,j)} \mathcal{N}\left(\mathbf{s}; \mu_{k|k}^{(i,j)}, P_{k|k}^{(i,j)}\right) \tag{40a}$$

$$\eta_{k|k}^{(i,j)} = \frac{P_D(\mathbf{s})\eta_{k|k-1}^{(i,j)} q^{(i,j)}(\mathbf{z},\mathbf{x}_k^{n,(i)})}{\lambda c(\mathbf{z}) + \sum_{\ell=1}^{J_{k|k-1}^{(i)}} P_D(\mathbf{s})\eta_{k|k-1}^{(i,\ell)} q^{(i,\ell)}(\mathbf{z},\mathbf{x}_k^{n,(i)})} \tag{40b}$$

where $q^{(i,j)}(\mathbf{z},\mathbf{x}_k^{n,(i)}) = \mathcal{N}\left(\mathbf{z}; \hat{\mathbf{z}}(\mathbf{x}_k^{n(i)}, \eta_{k|k-1}^{(i,j)}), S_k^{(i,j)}\right)$ and where $\hat{\mathbf{z}}(\,\cdot\,)$ is given in (9d). The terms $\eta_{k|k-1}^{(i,j)}$, $P_{k|k}^{(i,j)}$ and $S_k^{(i,j)}$ can be obtained using standard filtering techniques, such as EKF or UKF. The clutter density is $c(\mathbf{z}) = \mathcal{U}(\mathbf{z})$, where λ_c is the average number of clutter measurements and $\mathcal{U}(\mathbf{z})$ is a uniform distribution on the measurement space.

4.2 Particle Update

The transition density is chosen as the proposal distribution and hence the particles are sampled as

$$\mathbf{x}_k^{n,(i)} \sim p(\mathbf{x}_k^{n,(i)}|\mathbf{x}_{0:k-1}^{n,(i)}) \tag{41a}$$

$$w_k^{(i)} = p(\mathbf{Z}_k|\mathbf{Z}_{0:k-1}, \mathbf{x}_{0:k}^{(i)})w_{k-1}^{(i)} \tag{41b}$$

where the prediction can be done using (9a) with the substitution of the last estimated values of the linear components. Note that the linear components are treated as noise here and that therefore the process noise is

$$Q_k^n + F_k^n(\mathbf{x}_{k-1}^{n,(i)})P_{k-1}^{l,(i)}(F_k^n(\mathbf{x}_{k-1}^{n,(i)}))^{\mathrm{T}}$$

when sampling the particles. The update of the weights can be computed according to (31), where

$$\hat{\mathfrak{s}}_{k|k-1}^{(i)} = \sum_{j=1}^{J_{k|k-1}^{(i)}} \eta_{k|k-1}^{(i,j)} \quad \text{and} \quad \hat{\mathfrak{s}}_{k}^{(i)} = \sum_{j=1}^{J_{k|k}^{(i)}} \eta_{k}^{(i,j)} \tag{42}$$

are the sums of the predicted and updated PHD weights for the ith particle.

4.3 Algorithm

The algorithm is given in Algorithm 14. The state estimates are extracted by taking the weighted mean

$$\hat{\mathbf{x}}_k = \frac{1}{\sum w_k^{(i)}} \sum_{i=1}^{N} w_k^{(i)} \mathbf{x}_k^{(i)}. \tag{43}$$

The MGPs must not be extracted since they are only considered as a mean to estimate the target state.

Algorithm 14 Prediction and Update

```
Require: $\{w_{k-1}^{(i)}, \mathbf{x}_{k-1}^{n,(i)}, \hat{\mathbf{x}}_{k-1}^{l,(i)}, P_{k-1}^{(i)}, D_{k-1}^{(i)}\}_{i=1}^{N}$, where the PHD is composed of $D_{k-1}^{(i)} =$
    $\{\eta_{k-1}^{i,j}, \mu_{k-1}^{(i,j)}, \Sigma_{k-1}^{(i,j)}\}_{j=1}^{J_{k-1}^{(i)}}$
for $i = 1$ to $N$ do
    Predict $\mathbf{x}^n$
    $\mathbf{x}_{k|k-1}^{n,(i)} \sim p(\mathbf{x}_{1:k}^{n,(i)} | \mathbf{x}_{0:k-1}^{n,(i)}, \hat{\mathbf{x}}_{k-1}^{l,(i)})$ {cf. (9a)}
    Predict $\mathbf{x}^l$
    $S_{k-1} = F_{k-1}^n P_{k-1}^{(i)} (F_{k-1}^n)^{\mathrm{T}} + Q_{k-1}^n$
    $K_{k-1} = F_k^l P_{k-1}^{(i)} (F_{k-1}^n)^{\mathrm{T}} S_{k-1}^{-1}$
    $\hat{\mathbf{x}}_{k|k-1}^{l,(i)} = F_k^l \hat{\mathbf{x}}_{k-1}^{l,(i)} + f_{k-1}^l + K_{k-1}(\mathbf{x}_{k|k-1}^{n,(i)} - f_{k-1}^n - F_{k-1}^n \hat{\mathbf{x}}_{k-1}^{l,(i)})$
    $P_{k|k-1}^{(i)} = F_{k-1}^l P_{k-1}^{(i)} (F_{k-1}^l)^{\mathrm{T}} + Q_{k-1}^l - K_{k-1} S_{k-1} K_{k-1}^{-1}$
    Predict PHD $D$
    $\ell = 0$
    for $j = 1$ to $J_{b,k}^{(i)}$ do
        $\ell = \ell + 1$
        $\eta_{k|k-1}^{(i,\ell)} = \eta_{b,k}^{(i,j)}$, $\mu_{k|k-1}^{(i,\ell)} = \mu_{b,k}^{(i,j)}$, $\Sigma_{k|k-1}^{(i,\ell)} = \Sigma_{b,k}^{(i,j)}$
    end for
    for $j = 1$ to $J_k^{(i)}$ do
        $\ell = \ell + 1$
        $\eta_{k|k-1}^{(i,\ell)} = P_S \eta_{k-1}^{(i,j)}$, $\mu_{k|k-1}^{(i,\ell)} = \mu_{k-1}^{(i,j)}$, $\Sigma_{k|k-1}^{(i,\ell)} = \Sigma_{k-1}^{(i,j)} + Q_{k-1}^s$
    end for
    $J_{k|k-1}^{(i)} = l$, $\hat{\mathfrak{s}}_{k|k-1}^{(i)} = \sum_{j=1}^{J_{k|k-1}^{(i)}} \eta_{k|k-1}^{(i,j)}$
    Update PHD $D$
    for $j = 1$ to $J_{k|k-1}^{(i)}$ do
```

```
        $\hat{\mathbf{z}}_{k|k-1}^{(i,j)} = h_k(\mathbf{x}_{k|k-1}^{n,(i)}, \mu_{k|k-1}^{(i,j)}) + H_k(\mathbf{x}_{k|k-1}^{n,(i)})\hat{\mathbf{x}}_{k|k-1}^{l,(i)}$ {cf. (9d)}
        $\nabla H_k = \frac{\partial}{\partial \mathbf{s}} h_k(\mathbf{x}, \mathbf{s})\Big|_{\mathbf{x}=\mathbf{x}_{k|k-1}^{n,(i)}, \mathbf{s}=\mu_{k|k-1}^{(i,j)}}$ {cf. (48)}
        $S_k^{n,(i)} = H_k(\mathbf{x}_{k|k-1}^{n,(i)}) P_{k|k-1}^{(i)} H_k^{\mathrm{T}}(\mathbf{x}_{k|k-1}^{n,(i)})$
        $S_k^{(i,j)} = \nabla H_k \Sigma_{k|k-1}^{(i,j)} (\nabla H_k)^{\mathrm{T}} + S_k^{n,(i)} + R$
        $K_k^{(i,j)} = \Sigma_{k|k-1}^{(i,j)} (\nabla H_k)^{\mathrm{T}} (S_k^{(i,j)})^{-1}$
    end for
    for missed detections $j = 1$ to $J_{k|k-1}^{(i)}$ do
        $\eta_k^{(i,j)} = (1 - P_D)\eta_{k|k-1}^{(i,j)}, \mu_k^{(i,j)} = \mu_{k|k-1}^{(i,j)}, \Sigma_k^{(i,j)} = \Sigma_{k|k-1}^{(i,j)}$
    end for
    $\ell = 0$
    for each detection $\mathbf{z}_k \in \mathbf{Z}_k$ do
        $\ell = \ell + 1$
        for $j = 1$ to $J_{k|k-1}^{(i)}$ do
            $\tau^{(j)} = P_D \eta_{k|k-1}^{(i,j)} \mathcal{N}(\mathbf{z}_k; \hat{\mathbf{z}}_{k|k-1}^{(i,j)}, S_k^{(i,j)})$
            $\mu_k^{(i,\ell J_{k|k-1}^{(i)}+j)} = \mu_{k|k-1}^{(i,j)} + K_k^{(i,j)}(\mathbf{z}_k - \mathbf{z}_{k|k-1}^{(i,j)})$
            $\Sigma_k^{(i,\ell J_{k|k-1}^{(i)}+j)} = \Sigma_{k|k-1}^{(i,j)} + K_k^{(i,j)} S_k^{(i,j)} (K_k^{(i,j)})^{\mathrm{T}}$
        end for
        for $j = 1$ to $J_{k|k-1}^{(i)}$ do
            $\eta_k^{(i,\ell J_{k|k-1}^{(i)}+j)} = \tau^{(i)} / (\lambda c(\mathbf{z}_k) + \sum_{m=1}^{J_{k|k-1}^{(i)}} \tau^{(m)})$
        end for
    end for
    $J_k^{(i)} = (\ell + 1) J_{k|k-1}^{(i)}, \hat{\mathfrak{s}}_k^{(i)} = \sum_{j=1}^{J_k^{(i)}} \eta_k^{(i,j)}$
    merge and prune Gaussians
    Update $\mathbf{x}^n$
    $\tilde{w}_k^{(i)} = p(\mathbf{Z}_k | \mathbf{Z}_{0:k-1}, \mathbf{x}_{0:k}^n) w_{k-1}^{(i)}$ {cf. (31)}
end for
$w_k^{(i)} = \tilde{w}_k^{(i)} / \sum_{j=1}^{N} \tilde{w}_k^{(j)}$ {Normalize}
resample if necessary
Update $\mathbf{x}^l$
for $i = 1$ to $N$ do
    extract MGPs $\widehat{\mathbf{S}}_k^{(i)} = \{\hat{\mathbf{s}}_k^{(i,j)}\}_{j=1}^{\hat{\mathfrak{s}}_k^{(i)}}$ from $D_k^{(i)}$
    compute association matrix $A$ using e.g., NN
    $S_k = H_k^l P_{k|k-1}^{(i)} (H_k^l)^{\mathrm{T}} + R_k$
    $K_k = P_{k|k-1}^{(i)} (H_k^l)^{\mathrm{T}} (S_k)^{-1}$
    $\hat{\mathbf{x}}_k^{l,(i)} = \hat{\mathbf{x}}_{k|k-1}^{l,(i)} + K_k(A\mathbf{Z} - h_k(\mathbf{x}_{k|k}^{n,(i)}, \widehat{\mathbf{S}}_k^{(i)}) - H_k(\mathbf{x}_{k|k}^{n,(i)})\hat{\mathbf{x}}_{k|k-1}^{l,(i)})$
    $P_{k|k}^{(i)} = P_{k|k-1}^{(i)} - K_k S_k (K_k)^{\mathrm{T}}$
end for
```

```
return $\{w_k^{(i)}, \mathbf{x}_k^{n,(i)}, \hat{\mathbf{x}}_k^{l,(i)}, P_k^{(i)}, D_k^{(i)}\}_{i=1}^{N}$ and $\hat{\mathfrak{s}}_k^{(i)}$, with $D_k^{(i)} = \{\eta_k^{i,j}, \mu_k^{(i,j)}, \Sigma_k^{(i,j)}\}_{j=1}^{J_k^{(i)}}$
```

5 Simulation Example

In this section, we are going to use a simple example to illustrate the filtering solution we propose. In this example, we use the following specific target and measurement models.

- **Target State Model:** The state vector of the target is given by

$$\mathbf{x} = \begin{bmatrix} \mathsf{x} & \mathsf{y} & \mathsf{v} & \psi & \mathbf{t} \end{bmatrix}^{\mathrm{T}}, \tag{44}$$

  where (x, y) is the planar Cartesian position of the target. It may be any point related to the target, however in this example it is assumed to be the center position. Furthermore, v is the velocity and ψ is the heading angle. The shape and size of the target is described by the shape component $\mathbf{t}$. Considering a simple but common shape e.g., a rectangle, its size may be represented by a length l and a width w. In this example a simple coordinated turn motion model is used

$$\begin{aligned}
\mathsf{x}_{k+1} &= \mathsf{x}_k + T\mathsf{v}_k \cos(\psi) && \text{(45a)}\\
\mathsf{y}_{k+1} &= \mathsf{y}_k + T\mathsf{v}_k \sin(\psi) && \text{(45b)}\\
\mathsf{v}_{k+1} &= \mathsf{v}_k + \mathbf{w}_{\mathsf{v}} && \text{(45c)}\\
\psi_{k+1} &= \psi_k + \mathbf{w}_{\psi} && \text{(45d)}\\
\mathbf{t}_{k+1} &= \mathbf{t}_k + \mathbf{w}_{\mathbf{t}} && \text{(45e)}
\end{aligned}$$

  where T is the sample time. The heading angle ψ is clearly modeled as a nonlinear component.

- **Measurements and Shape Model:** Let the point measurement be a Cartesian position, i.e.,

$$\mathbf{z} = \begin{bmatrix} \bar{\mathsf{x}} & \bar{\mathsf{y}} \end{bmatrix}^{\mathrm{T}}. \tag{46}$$

  on the border of the target. Common point measurement sensors, such as radar and laser typically measure range and bearing. The polar representation of the sensor data has here been converted to a Cartesian representation. This also means that the measurement noise covariance must be converted and that it not necessarily is diagonal.

  A MGP is defined on a coordinate $\mathbf{s}$ along the border of the target which has a spline representation of order d. In this case, the measurement model (9d) may be written according to

$$\hat{\mathbf{z}} = h(\mathbf{x}^n, \mathbf{S}) + \begin{bmatrix} \mathsf{x} & \mathsf{y} \end{bmatrix}^{\mathrm{T}} \tag{47a}$$

with

$$h(\mathbf{x}^n, \mathbf{S}) = \mathrm{Rot}(\psi) \begin{bmatrix} \Pi B^\sigma G^\sigma & 0 \\ 0 & \Pi B^\sigma G^\sigma \end{bmatrix} \Gamma \tag{47b}$$

$$\Pi = \begin{bmatrix} 1 & \mathbf{s} & \mathbf{s}^2 & \cdots & \mathbf{s}^{d-1} \end{bmatrix} \tag{47c}$$

where Rot is a rotation matrix, B^σ and G^σ is a basis matrix and a placement matrix, respectively. The vector Γ includes the shape parameters $\mathbf{t}$. Spline curves are well described in e.g., Blake and Isard (1998).

A nice property using spline curves is that they are continuously differentiable, which is important if the EKF is utilized to update the Gaussian components of the PHD, compare with Line 19 in Algorithm 14. The derivative of the measurement model (47a) with respect to MGPs is

$$\nabla H_k = \frac{\partial}{\partial \mathbf{s}} h_k(\mathbf{x}^n, \mathbf{s}) \tag{48a}$$

$$= \mathrm{Rot}(\psi) \begin{bmatrix} \nabla \Pi B^\sigma G^\sigma & 0 \\ 0 & \nabla \Pi B^\sigma G^\sigma \end{bmatrix} \Gamma \tag{48b}$$

$$\nabla \Pi = \begin{bmatrix} 0 & 1 & \mathbf{s} & \cdots & (d-1)\mathbf{s}^{d-2} \end{bmatrix} \tag{48c}$$

In the models above, it is obvious that the center position x, y and velocity v of the target are linear in the motion and in the measurement model. The heading ψ and target shape state $\mathbf{t}$ are nonlinear, hence the state vector may be partitioned as

$$\mathbf{x}^l = \begin{bmatrix} \mathsf{x} & \mathsf{y} & \mathsf{v} \end{bmatrix}^{\mathrm{T}} \tag{49a}$$

$$\mathbf{x}^n = \begin{bmatrix} \psi & \mathbf{t} \end{bmatrix}^{\mathrm{T}}. \tag{49b}$$

Comparing (9d) and (47a) the linear measurement matrix is

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{50}$$

Note that the MGPs $\mathbf{s}$ are given by a one dimensional position along the border, and that they are highly nonlinear in the measurement model (47a) which justifies the use of particle filter.

A rectangle target is considered in this simulation. The shape state is the length l and width w. The vector Γ is

$$\Gamma = \begin{matrix} [\mathsf{l} & 0 & -\mathsf{l} & -\mathsf{l} & -\mathsf{l} & 0 & \mathsf{l} & \mathsf{l} & \mathsf{l} & \ldots \\ \ldots & -\mathsf{w} & -\mathsf{w} & -\mathsf{w} & 0 & \mathsf{w} & \mathsf{w} & \mathsf{w} & 0 & -\mathsf{w}]^{\mathrm{T}} \end{matrix} \tag{51}$$

A $d = 3$ order spline is considered. In the simulation the extended target moves from right to left throughout the surveillance area. The orientation of the target is 0rad and the length and width are $\mathsf{l} = 5$m and $\mathsf{w} = 2.5$m respectively. The target trajectory and the surveillance region of the sensor located at the origin are illustrated in Figure 2. As seen from the figure, the target starts far from the

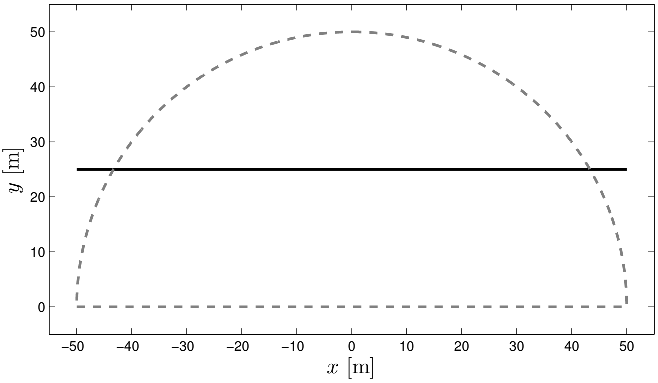


Figure 2: *The trajectory used in the simulation. The target trajectory is showed in black. The sensor is located in the origin, the surveillance area boundary is showed with a dashed gray line.*

sensor and hence has few measurements initially. Neither width nor the length of the target is observable. As the target travels towards the sensor, the length and the width observability increases first but when the target starts to get close to the sensor, the width observability is lost again in which case only the length of the target is visible. When the target passes by the sensor the width becomes once more observable. With the distance between the target and the sensor increasing towards the end of the scenario, observability of both quantities decrease.

The position estimation results are shown in Figure 3. Clearly the algorithm can follow the target along the x-axis. The shape estimates are illustrated in Figure 4. As predicted the shape estimates degrade as the target gets further from the sensor. Though the variance of the estimates are large at times, the algorithm seems to be capable of keeping them at a smaller level than the initial values. As the target approaches the sensor, the general trend in the variances is to decrease though there exist also occasional increases. Further investigations must be done with various initializations and stability and the robustness must be evaluated more thoroughly in the future work.

6 Conclusions

In this work a new approach to track extended targets has been presented. Point measurements are produced by an unknown number of measurement generating points (MGPs) on the surface of the target. These MGPs are modeled as a random finite set RFS and used as a means to estimate the shape, the size and the position of a target. The state of the target is propagated with Rao-Blackwellized particle filter (nonlinear part) and Kalman filters (linear part); the measurements are processed with a GM-PHD-filter.

First simulation results show that this approach is promising. A simple rectangu-

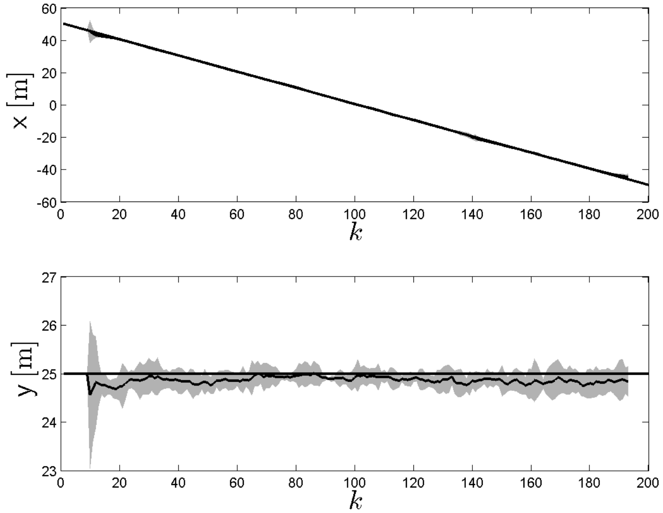


Figure 3: *The position estimates (*x *and* y*) of the algorithm. The true quantities are shown with grey lines. The mean* x *and* y *estimates are in black lines and their 4 standard deviation uncertainty calculated from the particles are illustrated with grey clouds around the estimates.*

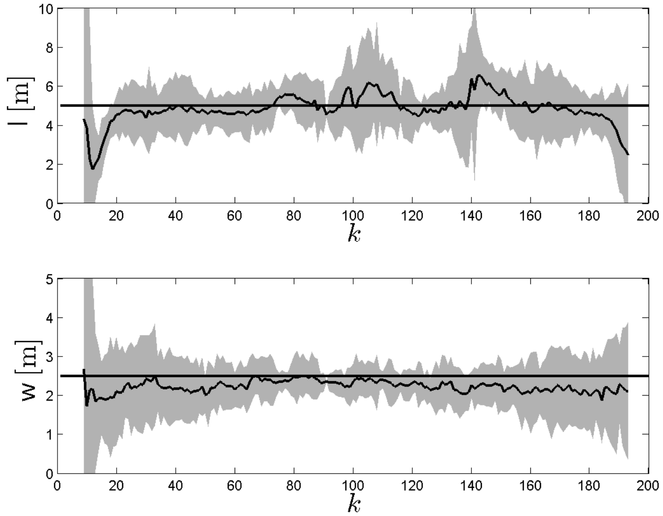


Figure 4: *The length and the width estimates of the algorithm. The true quantities are shown with grey lines. The mean length and width estimates are in black lines and their 4 standard deviation uncertainty calculated from the particles are illustrated with grey clouds around the estimates.*

lar target is followed with a rather stable performance and it remains for future fork to estimate the shape of more complex targets. The next step will be to validate the proposed method on more challenging real data collected on e.g., a road, where different type of shapes, such as pedestrians and vehicles are visible. The stability and robustness must also be evaluated thoroughly with different initializations and settings.

Bibliography

D. Angelova and L. Mihaylova. Extended object tracking using Monte Carlo methods. *IEEE Transactions on Signal Processing*, 56(2):825–832, February 2008.

A. Blake and M. Isard. *Active contours*. Springer, London, UK, 1998.

Y. Boers, H. Driessen, J. Torstensson, M. Trieb, R. Karlsson, and F. Gustafsson. Track-before-detect algorithm for tracking extended targets. *IEE Proceedings of Radar, Sonar and Navigation*, 153(4):345–351, August 2006.

K. Gilholm and D. Salmond. Spatial distribution model for tracking extended objects. *IEE Proceedings of Radar, Sonar and Navigation*, 152(5):364–371, October 2005.

K. Gilholm, S. Godsill, S. Maskell, and D. Salmond. Poisson models for extended target and group tracking. In *Proceedings of Signal and Data Processing of Small Targets*, volume 5913, pages 230–241, San Diego, CA, USA, August 2005. SPIE.

K. Granström, C. Lundquist, and U Orguner. A Gaussian mixture PHD filter for extended target tracking. In *Proceedings of the International Conference on Information Fusion*, Edinburgh, UK, July 2010.

M. Isard and A. Blake. CONDENSATION-conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, August 1998.

M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, January 1988.

J. W. Koch. Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44 (3):1042–1059, July 2008.

S. Lazarus, A. Tsourdos, B.A. White, P.M.G. Silson, and R. Zandbikowski. Airborne vehicle mapping of curvilinear objects using 2-D splinegon. *IEEE Transactions on Instrumentation and Measurement*, 59(7):1941–1954, July 2010.

C. Lundquist, K. Granström, and U. Orguner. Estimating the shape of targets with a PHD filter. In *Proceedings of the International Conference on Information Fusion*, Chicago, IL, USA, July 2011.

R. P. S. Mahler. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, October 2003.

R. P. S. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, Boston, MA, USA, 2007.

R. P. S. Mahler. PHD filters for nonstandard targets, I: Extended targets. In

Proceedings of the International Conference on Information Fusion, pages 915–921, Seattle, WA, USA, July 2009.

J. Mullane, B-.N. Vo, M. D. Adams, and B-.T. Vo. A random-finite-set approach to bayesian slam. *IEEE Transactions on Robotics*, PP(99):1–15, February 2011.

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

A. Srivastava and I. H. Jermyn. Looking for shapes in two-dimensional cluttered point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:1616–1629, September 2009. ISSN 0162-8828.

J. Vermaak, N. Ikoma, and S. J. Godsill. Sequential Monte Carlo framework for extended object tracking. *IEE Proceedings of Radar, Sonar and Navigation*, 152 (5):353–363, October 2005.

B.-N. Vo and W.-K. Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, November 2006.

Paper G

Tire Radii and Vehicle Trajectory Estimation Using a Marginalized Particle Filter

Authors: Christian Lundquist, Emre Özkan and Fredrik Gustafsson

Edited version of the paper:

C. Lundquist, E. Özkan, and F. Gustafsson. Tire radii estimation using a marginalized particle filter. *IEEE Transactions on Intelligent Transportation Systems*, 2011d. Submitted.

The paper presents data that was previously published in:

E. Özkan, C. Lundquist, and F. Gustafsson. A Bayesian approach to jointly estimate tire radii and vehicle trajectory. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, Washington DC, USA, October 2011.

Preliminary version:

Technical Report LiTH-ISY-R-3029, Dept. of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden.

Tire Radii and Vehicle Trajectory Estimation Using a Marginalized Particle Filter

Christian Lundquist, Emre Özkan and Fredrik Gustafsson

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

Abstract

Measurements of individual wheel speeds and absolute position from a global navigation satellite system (GNSS) are used for high-precision estimation of vehicle tire radii in this work. The radii deviation from its nominal value is modeled as a Gaussian process and included as noise components in a vehicle model. The novelty lies in a Bayesian approach to estimate online both the state vector of the vehicle model and noise parameters using a marginalized particle filter. No model approximations are needed such as in previously proposed algorithms based on the extended Kalman filter. The proposed approach outperforms common methods used for joint state and parameter estimation when compared with respect to accuracy and computational time. Field tests show that the absolute radius can be estimated with millimeter accuracy, while the relative wheel radius on one axle is estimated with submillimeter accuracy.

1 Introduction

Tire pressure monitoring has become an integral part of todays' automotive active safety concept Velupillai and Guvenc (2007). With the announcement of the US standard (FMVSS 138) and the European standard (ECE R-64) vehicle manufacturer must provide a robust solution to early detect tire pressure loss. A direct way to measure the tire pressure is to equip the wheel with a pressure sensor and transmit the information wireless to the vehicle body. In the harsh environment that the tires are exposed to, e.g., water, salt, ice and heat, the sensors are error-prone. Furthermore, the pressure sensors in each wheel is expensive and their existence complicates the wheel changes. Therefore, indirect solutions have been introduced on the market lately, see e.g., Persson et al. (2001). In this paper an indirect approach is presented where the tire radius is estimated simultaneously with the vehicle trajectory. This is done under the assumption that there is a relation between a reduction in tire radius and tire pressure.

The indirect approach presented in Persson et al. (2001) is only based on the wheel speed sensors and it is shown how a tire pressure loss in one wheel leads to a relative radii error between the wheels. In later publications GPS measurements have also been included to improve the radius estimation and even make it possible to estimate the absolute radius of one tire. The effective tire radius is estimated using a simple least-squares regression technique in Miller et al. (2001). A non-linear observer approach to estimate the tire radius is presented in Carlson and Gerdes (2005), and a second order sliding mode observer is used to estimate the tire radius in M'sirdi et al. (2006); Shraim et al. (2006). A simultaneous maximum likelihood calibration and sensor position and orientation estimation approach for mobile robots is presented in Censi et al. (2008), where among other parameters the wheel radii are estimated. An observer based fault detection algorithm, which detects tire radius errors using yaw rate measurement and a bicycle model of the vehicle, is described in Patwardhan et al. (1997). An extended Kalman filter based approach is presented in Ersanilli et al. (2009), where the tire radius is estimated via vertical chassis accelerations.

In the present contribution the difference between the nominal and the actual value of the tire radius is modeled as a Gaussian stochastic process, where both the mean and the covariance are unknown and time varying. The vehicle dynamics and the measurements are described by a general state space model and the noise statistics are treated as the unknown parameters of the model. Hence the joint estimation of the state vector and the unknown model parameters based on available measurements is required. Such a problem is hard to treat as both the state estimation and the parameter estimation stages affects the performance of the other. The structure of this nonlinear problem with biased and unknown noise requires utilizing approximative algorithms. The particle filter (PF) provides one generic approach to non-linear non-Gaussian filtering. Probably the most common way to handle a joint parameter and state estimation problem is by augmenting the state vector with the unknown parameters and redefine the problem as a filtering problem, see e.g., Liu and West (2001); Julier and Durrant-Whyte (1995); Ersanilli et al. (2009). The approach has some major disadvantages as it requires artificial dynamics for the static parameters and it leads also to an increase in the state dimension which is not preferable for particle filters. This is particularly important to stress in automotive applications, where the computational cost must be kept low. In this work, an efficient Bayesian method is proposed for approximating the joint density of the unknown parameters and the state based on the particle filters and the marginalization concept, introduced in Saha et al. (2010). The statistics of the posterior distribution of the unknown noise parameters is propagated recursively conditional on the nonlinear states of the particle filter. An earlier version of this work was presented in Özkan et al. (2011), and the current, significantly improved version also includes comparisons with other methods.

The proposed method is implemented and tested on real vehicle data. The state augmentation technique is also implemented and tested, both using the PF and the extended Kalman Filter (EKF) variants. Furthermore, a method where the unknown noise enters in the measurement signal instead of, as proposed, in the input signal is presented for comparison. The proposed method, the two augmented state methods and the measurement noise method are compared with respect to estimation accuracy and robustness. The proposed method outperforms the competitors both in the simulations and the real data.

The paper is outlined as follows. The problem is formulated together with the vehicle model and with the description of the sensors in Section 2. The estimation procedure and the filter approach is described in Section 3. The proposed method is compared to common and similar methods described in the literature, and these are summarized and put into the context in Section 4. Results based on real data collected with a production type passenger car are presented in Section 5. The work is concluded in Section 6.

2 Model

The aim of this work is to jointly estimate the vehicle trajectory and the unknown tire radius errors based on the available measurements in a Bayesian framework. Furthermore, the aim is to find appropriate models that can be recast into the general state space model

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + g(\mathbf{x}_k, \mathbf{u}_k)\mathbf{w}_k, \tag{1a}$$

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{u}_k) + d(\mathbf{x}_k, \mathbf{u}_k)\mathbf{e}_k. \tag{1b}$$

This model suits our noise parameter marginalized particle filter framework.

A four wheeled vehicle model, which is used to estimate the unknown variables given the velocity and the position measurements, is introduced in this section. More specifically, the angular velocities of the wheels and GPS positions are considered as the inputs and the measurements, respectively. It is assumed that the unknown wheel radii affect the vehicle state trajectory through the wheel speed sensors. The state vector is defined as the planar position and the heading angle of the vehicle,

$$\mathbf{x} = \begin{bmatrix} \mathsf{x} & \mathsf{y} & \psi \end{bmatrix}^{\mathrm{T}}. \tag{2}$$

The discrete time motion model for the evolution of the state is given as,

$$\mathsf{x}_{k+1} = \mathsf{x}_k + T\mathsf{v}_k \cos\psi_k \tag{3a}$$

$$\mathsf{y}_{k+1} = \mathsf{y}_k + T\mathsf{v}_k \sin\psi_k \tag{3b}$$

$$\psi_{k+1} = \psi_k + T\dot{\psi}_k, \tag{3c}$$

where v_k is the vehicle longitudinal velocity, $\dot{\psi}_k$ is the yaw rate of the vehicle and T is the sampling time.

Furthermore, the angular velocities of the wheels ω_i, $i = 1 \ldots 4$, which are measured by the ABS sensors at each wheel, can be converted to virtual measurements of the absolute longitudinal velocity and yaw rate as described in Gustafsson et al. (2001); Gustafsson (2010a). Assuming a front wheel driven vehicle with slip-free

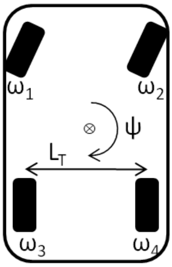


Figure 1: *The notation of the vehicle variables.*

motion of the rear wheels the virtual measurements become

$$\mathsf{v}^{\text{virt}} = \frac{\omega_3 r + \omega_4 r}{2} \tag{4a}$$

$$\dot{\psi}^{\text{virt}} = \frac{\omega_4 r - \omega_3 r}{l_t}, \tag{4b}$$

where r is the nominal value of the wheel radius and l_t is the wheel track; see Figure 1 for the notation. In practice, the actual tire radii differ from their nominal value. They are unknown and needed to be estimated on the run. The wheel radius errors are defined as the difference between the actual and the nominal values of the rear left and right wheel radii $\delta_3 \triangleq r_3 - r$ and $\delta_4 \triangleq r_4 - r$, respectively. The actual tire radii are

$$r_3 = r + \delta_3 \tag{5a}$$

$$r_4 = r + \delta_4. \tag{5b}$$

Multiplying r_3 with ω_3 and r_4 with ω_4 in equations (4) to derive the actual longitudinal velocity v and yaw raw rate $\dot{\psi}$ results in

$$\mathsf{v} = \frac{\omega_3 r_3 + \omega_4 r_4}{2} = \mathsf{v}^{\text{virt}} + \frac{\omega_3 \delta_3}{2} + \frac{\omega_4 \delta_4}{2} \tag{6a}$$

$$\dot{\psi} = \frac{\omega_4 r_4 - \omega_3 r_3}{l_t} = \dot{\psi}^{\text{virt}} + \frac{\omega_4 \delta_4}{l_t} - \frac{\omega_3 \delta_3}{l_t}. \tag{6b}$$

Substituting these into the motion model (3) gives

$$\mathsf{x}_{k+1} = \mathsf{x}_k + T(\mathsf{v}_k^{\text{virt}} + \frac{\omega_3 \delta_{3,k}}{2} + \frac{\omega_4 \delta_{4,k}}{2}) \cos \psi_k \tag{7a}$$

$$\mathsf{y}_{k+1} = \mathsf{y}_k + T(\mathsf{v}_k^{\text{virt}} + \frac{\omega_3 \delta_{3,k}}{2} + \frac{\omega_4 \delta_{4,k}}{2}) \sin \psi_k \tag{7b}$$

$$\psi_{k+1} = \psi_k + T(\dot{\psi}_k^{\text{virt}} + \frac{\omega_4 \delta_{4,k}}{l_t} - \frac{\omega_3 \delta_{3,k}}{l_t}), \tag{7c}$$

where the input vector $\mathbf{u}$ only consists of the wheel speeds, i.e.,

$$\mathbf{u} = \begin{bmatrix} \omega_3 & \omega_4 \end{bmatrix}^{\mathrm{T}} \tag{8}$$

since $\mathsf{v}_k^{\text{virt}}$ and $\dot{\psi}_k^{\text{virt}}$ are functions of the wheel speeds.

The motion model (7) can now be written in the form (1a), with

$$f(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} \mathrm{x}_k + T\mathrm{v}_k^{\mathrm{virt}} \cos \psi_k \\ \mathrm{y}_k + T\mathrm{v}_k^{\mathrm{virt}} \sin \psi_k \\ \psi_k + T\dot{\psi}_k^{\mathrm{virt}} \end{bmatrix} \tag{9a}$$

$$g(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} \frac{T\omega_3 \cos \psi_k}{2} & \frac{T\omega_4 \cos \psi_k}{2} \\ \frac{T\omega_3 \sin \psi_k}{2} & \frac{T\omega_4 \sin \psi_k}{2} \\ -\frac{T\omega_3}{l_t} & \frac{T\omega_4}{l_t} \end{bmatrix} \tag{9b}$$

where the process noise term $\mathbf{w}_k$ is defined as,

$$\mathbf{w}_k = \begin{bmatrix} \delta_{3,k} \\ \delta_{4,k} \end{bmatrix} \sim \mathcal{N}(\mu_k, \Sigma_k) = \mathcal{N}\left(\begin{bmatrix} \mu_{3,k} \\ \mu_{4,k} \end{bmatrix}, \begin{bmatrix} \Sigma_{3,k} & 0 \\ 0 & \Sigma_{4,k} \end{bmatrix} \right). \tag{9c}$$

The process noise, or the tire radius error, is described by two parameters, the mean value μ and the covariance Σ for the left and the right wheel. Intuitively the mean value corresponds to the slow time variations from the nominal tire radius, while the variance corresponds to the fast variations due to tire vibrations. One interpretation is that the mean value μ models the change in the wheel radii due to abrasion, tire pressure changes and effects of cornering, and the covariance Σ can account for the vibrations arising from an uneven road surface. An interesting special case is when $\Sigma_3 = \Sigma_4$, which represents homogeneous road conditions, in comparison with split road surface when $\Sigma_3 \neq \Sigma_4$.

The measurement model (1b) defines the relation between the GPS position and the state variables as follows.

$$\mathbf{y}_k = \begin{bmatrix} \mathrm{x}_k^{\mathrm{GPS}} & \mathrm{y}_k^{\mathrm{GPS}} \end{bmatrix}^{\mathrm{T}} \tag{10a}$$

$$h(\mathbf{x}_k, \mathbf{u}_k) = h(\mathbf{x}_k) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{x}_k \tag{10b}$$

$$d(\mathbf{x}_k, \mathbf{u}_k) = I \tag{10c}$$

where I is the identity matrix and the measurement noise is assumed to be Gaussian with zero mean and known constant covariance R, i.e.,

$$\mathbf{e}_k = \begin{bmatrix} \mathbf{e}_{\mathrm{x}} & \mathbf{e}_{\mathrm{y}} \end{bmatrix}^{\mathrm{T}} = \mathcal{N}(0, R), \quad R = \Sigma^{\mathrm{GPS}} I_2. \tag{10d}$$

Other sensor measurements are also plausible to include in the measurement vector, at the cost of a more complex model. For instance, using a yaw rate gyro to measure the third state, also requires to model the drifting offset in the sensor. The steering angle can also be converted to yaw rate, but it suffers from dynamic lag and other dynamic states of the vehicle.

The unknown parameters $\boldsymbol{\theta}$ are the radii error biases and the covariances,

$$\boldsymbol{\theta}_k \triangleq \{\mu_{3,k}, \mu_{4,k}, \Sigma_{3,k}, \Sigma_{4,k}\}. \tag{11}$$

The unknown parameters are all subject to change in time. The underlying evolution model for the mean and covariance, i.e. $p(\mu_k|\mu_{k-1})$ and $p(\Sigma_k|\Sigma_{k-1})$ are also unknown. We later use forgetting factor principle on the statistics to account for this factor.

In the following section the joint estimation of the unknown parameters and the state is described in a Bayesian framework.

3 Parameter and State Estimation

This section focuses on the evaluation of the joint density $p(\mathbf{x}_k, \boldsymbol{\theta}_k|\mathbf{y}_{1:k})$ of the state variable $\mathbf{x}_k$ defined in (2) and the parameters $\boldsymbol{\theta}_k$ defined in (11), conditioned on all measurements $\mathbf{y}_{1:k}$ from time 1 to k. In order to simplify the calculations, the target density is decomposed into conditional densities as follows

$$p(\mathbf{x}_k, \boldsymbol{\theta}_k|\mathbf{y}_{1:k}) = p(\boldsymbol{\theta}_k|\mathbf{x}_{0:k}, \mathbf{y}_{1:k})p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}). \tag{12}$$

The resulting two densities are estimated recursively. The implementation of the estimation algorithm is described in three steps. In Section 3.1 we define the approximations made to derive the state trajectory $p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$ and in Section 3.2 we describe the estimation of the sufficient statistics of the parameter distribution $p(\boldsymbol{\theta}_k|\mathbf{x}_{0:k}, \mathbf{y}_{1:k})$. The predicted trajectory is derived conditional on the parameter estimates and the estimation of the joint density and the marginal density of the states $p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$ is finalized in Section 3.3.

3.1 State Trajectory

The state trajectory $p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$ is approximated by an empirical density of N particles as follows:

$$p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) \approx \sum_{i=1}^{N} w_k^{(i)} \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^{(i)}) \tag{13}$$

where $\mathbf{x}_{0:k}^{(i)}$ is a trajectory sample and $w_k^{(i)}$ is its related weight. The approximated state trajectory (13) is propagated with a particle filter, using the sequential importance sampling scheme. In this scheme, at any time k, first the samples, which are also denoted by particles, are generated from a proposal distribution $q(\mathbf{x}_k|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k})$ by using the particles from time $k-1$. Then the weight update step follows the sampling step. The weights are updated according to

$$w_k^{(i)} \propto \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k-1})}{q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k})} w_{k-1}^{(i)}. \tag{14a}$$

In the sampling step one can use the state prediction density as the proposal density while generating the samples. The posterior distribution of the unknown parameters can be used in computing the state prediction density $p(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k-1})$ according to

$$p(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k-1})d\boldsymbol{\theta}. \tag{15}$$

The estimation of the parameters is described in the next section, and we will therefore return to this equation in Section 3.3 and give the final expression.

3.2 Parameter Estimation

In the factorization of the joint density given in equation (12), the distribution of the unknown parameters (which corresponds to the first term) is computed conditional on the realization of the state trajectory and the measurements. For a specific realization of the trajectory, the posterior density can also be written conditional on the realization of the noise terms

$$p(\boldsymbol{\theta}_k|\mathbf{x}_{0:k}^{(i)}, \mathbf{y}_{1:k}) = p(\boldsymbol{\theta}_k|\mathbf{w}_{0:k}^{(i)}). \tag{16}$$

It can be decomposed into a likelihood function and a prior according to Bayes rule

$$p(\boldsymbol{\theta}_k|\mathbf{w}_{0:k}) \propto p(\mathbf{w}_k|\boldsymbol{\theta}_k)p(\boldsymbol{\theta}_k|\mathbf{w}_{0:k-1}). \tag{17}$$

The likelihood function $p(\mathbf{w}_k|\boldsymbol{\theta}_k)$ is assumed to be multivariate Gaussian in this work, as previously mentioned in (9c), and the mean μ_k and the covariance Σ_k are considered unknown parameters $\boldsymbol{\theta}$. In this case a Normal-inverse-Wishart distribution defines a conjugate prior[1] $p(\boldsymbol{\theta}_k|\mathbf{w}_{0:k-1})$. Normal-inverse-Wishart distribution defines a hierarchical Bayesian model given as

$$\mathbf{w}_k|\mu_k, \Sigma_k \sim \mathcal{N}(\mu_k, \Sigma_k) \tag{18a}$$

$$\mu_k|\Sigma_k \sim \mathcal{N}(\hat{\mu}_{k|k}, \gamma_{k|k}\Sigma_{k|k}) \tag{18b}$$

$$\Sigma_k \sim \mathrm{iW}(\nu_{k|k}, \Lambda_{k|k}) \tag{18c}$$

$$\propto |\Sigma_k|^{-\frac{1}{2}(\nu+d+1)} \exp\left(-\frac{1}{2}\,\mathrm{Tr}(\Lambda_{k|k}\Sigma_k^{-1})\right) \tag{18d}$$

where iW(.) denotes the inverse Wishart distribution and d denotes the dimension of the noise vector $\mathbf{w}_k$. The statistics $S_{\mathbf{w},k} \triangleq \{\gamma_k, \hat{\mu}_k, \nu_k, \Lambda_k\}$ can according to Peterka (1981); Kárný (2006) be recursively updated as follows. The measurement update is

$$\gamma_{k|k} = \frac{\gamma_{k|k-1}}{1+\gamma_{k|k-1}} \tag{19a}$$

$$\hat{\mu}_{k|k} = \hat{\mu}_{k|k-1} + \gamma_{k|k}(\mathbf{w}_k - \hat{\mu}_{k|k-1}) \tag{19b}$$

$$\nu_{k|k} = \nu_{k|k-1} + 1 \tag{19c}$$

$$\Lambda_{k|k} = \Lambda_{k|k-1} + \frac{1}{1+\gamma_{k|k-1}}(\hat{\mu}_{k|k-1} - \mathbf{w}_k)(\hat{\mu}_{k|k-1} - \mathbf{w}_k)^{\mathrm{T}} \tag{19d}$$

where the statistics of the predictive distributions are given by the time update step according to

$$\gamma_{k|k-1} = \frac{1}{\lambda}\gamma_{k-1|k-1} \tag{20a}$$

$$\hat{\mu}_{k|k-1} = \hat{\mu}_{k-1|k-1} \tag{20b}$$

$$\nu_{k|k-1} = \lambda\nu_{k-1|k-1} \tag{20c}$$

$$\Lambda_{k|k-1} = \lambda\Lambda_{k-1|k-1}. \tag{20d}$$

[1] A family of prior distributions is conjugate to a particular likelihood function if the posterior distribution belongs to the same family as the prior.

The scalar real number $0 \leq \lambda \leq 1$ is the forgetting factor. The forgetting factor here helps in the estimation of the dynamic variables. The statistics relies on roughly the measurements within the last $h = \frac{1}{1-\lambda}$ frames/time instances. That allows the algorithm to adapt the changes in the noise statistics in time. Such an approach is appropriate when the unknown parameters are slowly varying, and the underlying parameter evolution is unknown.

3.3 Noise Marginalization

Let us now return to the state prediction in (15). One important advantage of using the conjugate priors reveals itself here as it is possible to integrate out the unknown noise parameters as they follow Normal-inverse-Wishart distribution. The integrand in (15) is the product of a Gaussian distribution and a NiW distribution and the result of the integral is a Student-t distribution which can be evaluated analytically. The predictive distribution of $\mathbf{w}_k$ becomes a multivariate Student-t density, according to

$$p(\mathbf{w}_k|V_{k-1}, \nu_{k-1}) = \mathrm{St}(\hat{\mu}_{k|k-1}, \Lambda_{k|k-1}, \nu_{k|k-1} - d + 1) \tag{21a}$$

$$\propto \left|1 + \frac{1}{\nu_{k|k-1} - d + 1} \tilde{\mathbf{w}}_k^{\mathrm{T}} \Lambda_{k|k-1}^{-1} \tilde{\mathbf{w}}_k \right|^{-\frac{1}{2}(\nu_{k|k-1}+1)} \tag{21b}$$

with $\tilde{\mathbf{w}}_k \triangleq \mathbf{w}_k - \hat{\mu}_{k|k-1}$ and with $\nu_{k|k-1} - d + 1$ degrees of freedom. The Student-t distribution is located at $\hat{\mu}_{k|k-1}$ with the scale parameter $\Lambda_{k|k-1}$, and these statistics are given in (20). Furthermore, if the state transition density $p(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k-1})$ is used as the proposal distribution $q(\mathbf{x}_k|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k})$, then the the weight update equation (14) reduces to,

$$w_k^{(i)} = w_{k-1}^{(i)} p(\mathbf{y}_k|\mathbf{x}_k^{(i)}). \tag{22}$$

In the implementation, the noise is first sampled from (21) and used in (1a) in order to create samples $\mathbf{x}_k^{(i)}$. The samples from (21) can be used directly in the statistics update (19). The pseudo code of the algorithm used in the simulations is given in Table 15.

In the proposed method, each particle i keeps its own estimate for the parameters $\theta^{(i)}$ of the unknown process noise. In the importance sampling step, the particles use their own posterior distribution of the unknown parameters. The weight update of the particles is made according to the measurement likelihood. The particles are keeping the unknown parameters, and those which best explain the observed measurement sequence will survive in time.

The marginal posterior density of the unknown parameters can be computed by integrating out the states in the joint density

$$p(\theta|\mathbf{y}_{1:k}) = \int p(\theta|\mathbf{x}_{0:k}, \mathbf{y}_{1:k}) p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) d\mathbf{x}_{0:k} \approx \sum_{i=1}^{N} \omega_k^{(i)} p(\theta|\mathbf{x}_{0:k}^{(i)}, \mathbf{y}_{1:k}). \tag{23}$$

Then the estimate of the unknown parameters could be computed according to

Algorithm 15 Pseudo Code of the Algorithm

```
Initialization:
for each particle $i = 1, .., N$ do
    Sample $\mathbf{w}_0^{(i)}$ from (21)
    Compute $\mathbf{x}_0^{(i)}$ from (1a)
    Set initial weights $\omega_0^{(i)} = \frac{1}{N}$
    Set initial noise statistics $S_{\mathbf{w},0}^{(i)}$ corresponding to each particle
end for
Iterations:
for $k = 1, 2, \ldots$ do
    for For each particle $i = 1, .., N$ do
        Predict noise statistics $S_{\mathbf{w},k|k-1}^{(i)}$ using (20)
        Sample $\mathbf{w}_{k|k-1}^{(i)}$ from (21)
        Compute $\mathbf{x}_{k|k-1}^{(i)}$ from (1a)
        update the weights, $\tilde{w}_k^{(i)} = w_{k-1}^{(i)} p(\mathbf{y}_k | \mathbf{x}_k^{(i)})$
        Update noise statistics $S_{\mathbf{w},k}^{(i)}$ using (19)
    end for
    Normalize weights, $w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{i=1}^N \tilde{w}_k^{(i)}}$.
    Compute $N_{\text{eff}} = \frac{1}{\sum_{i=1}^N (w_k^{(i)})^2}$.
    If $N_{\text{eff}} \leq \eta$, Resample the particles.
    Compute state estimate $\hat{\mathbf{x}} = \sum_{i=1}^N w_k^{(i)} \mathbf{x}_k^{(i)}$
    Compute the parameter estimates using (24)
end for
```

a chosen criterion. As an example, according to the minimum mean square error (MMSE) criterion, the noise mean and variance estimates at time k could be computed as

$$\hat{\mu}_k = \sum_{i=1}^{N} \omega_k^{(i)} \hat{\mu}_k^{(i)} \tag{24a}$$

$$\widehat{\Sigma}_k = \sum_{i=1}^{N} \omega_k^{(i)} \frac{\nu_k^{(i)} - d + 1}{\nu_k^{(i)} - d - 1} \Lambda_k^{(i)}, \tag{24b}$$

where the weights are inherited from the particles.

4 Models for Comparison

In order to evaluate the performance of the proposed method it will be compared with two other methods. These two methods are described in this section followed by the results and discussions in Section 5. The three methods are based on the same type of information and model. All of them can be described by the general state space model (1), however, the order and the definition of the model components and variables vary between them. The approach presented in the preceding sections will be denoted as *marginalized particle filter process noise* (MPF-PN) estimation.

A model where the state vector is augmented with the parameters is described in Section 4.1, followed by a model where the parameters appear in the measurement noise, instead of in the process noise, in Section 4.2. The augmented state space model in Section 4.1 can be implemented using both a PF and a Kalman filter, i.e., summing up to in total four different filters implementations to compare. The characteristics of the four different methods are summarized in Section 4.3. Finally, Section 4.4 discusses the velocity measurements, which are used as inputs in the system.

4.1 Augmented State Vector

One common approach in joint state and parameter estimation is to augment the state vector with the unknown parameters Liu and West (2001). In this case the state vector is defined as

$$\mathbf{x} = \begin{bmatrix} \mathsf{x} & \mathsf{y} & \psi & \mu_3 & \mu_4 & \Sigma_3 & \Sigma_4 \end{bmatrix}^{\mathrm{T}}. \tag{25}$$

The motion model (7) contains the variables δ_3 and δ_4 and in order to express the motion model using the state variables the relation in (9c) is considered, i.e., $\delta_3 = \mu_3 + \sqrt{\Sigma_3}\mathcal{N}(0,1)$ and vice versa for δ_4. The state space model can again be written in the general form (1) with the components according to

$$f(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} \mathsf{x}_k + T\left(\mathsf{v}_k^{\text{virt}} + \frac{\omega_3\mu_{3,k}+\omega_4\mu_{4,k}}{2}\right)\cos\psi_k \\ \mathsf{y}_k + T\left(\mathsf{v}_k^{\text{virt}} + \frac{\omega_3\mu_{3,k}+\omega_4\mu_{4,k}}{2}\right)\sin\psi_k \\ \psi_k + T\left(\dot{\psi}_k^{\text{virt}} - \frac{\omega_3\mu_{3,k}-\omega_4\mu_{4,k}}{l_t}\right) \\ \mu_{3,k} \\ \mu_{4,k} \\ 0 \\ 0 \end{bmatrix} \tag{26a}$$

$$g(\mathbf{x}_k, \mathbf{u}_k) = \begin{bmatrix} T\frac{\omega_3\sqrt{\Sigma_{3,k}}}{2}\cos\psi_k & T\frac{\omega_4\sqrt{\Sigma_{4,k}}}{2}\cos\psi_k & 0 & 0 \\ T\frac{\omega_3\sqrt{\Sigma_{3,k}}}{2}\sin\psi_k & T\frac{\omega_4\sqrt{\Sigma_{4,k}}}{2}\sin\psi_k & 0 & 0 \\ T\frac{-\omega_3\sqrt{\Sigma_{3,k}}}{l_t} & T\frac{\omega_4\sqrt{\Sigma_{4,k}}}{l_t} & 0 & 0 \\ 0 & 0 & I_2 & 0 \\ 0 & 0 & 0 & I_2 \end{bmatrix} \tag{26b}$$

$$\mathbf{w}_k = \begin{bmatrix}\mathbf{w}_n^{\mathrm{T}} & \mathbf{w}_{\mu,k}^{\mathrm{T}} & \mathbf{w}_{\Sigma,k}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \tag{26c}$$

where $\mathbf{w}_n = \mathcal{N}(0, I_2)$. The process noises $\mathbf{w}_{\mu,k}$ is zero mean Gaussian noise. i.e.,

$$\mathbf{w}_\mu = \mathcal{N}(0, Q). \tag{26d}$$

In order to preserve the positive definite property, the following Markovian model with Inverse-Gamma distribution is used to propagate the unknown variances

$$p(\Sigma_{3,k}|\Sigma_{3,k-1}) = i\Gamma(\alpha_{3,k}, \beta_{3,k}) \tag{27a}$$

$$p(\Sigma_{4,k}|\Sigma_{4,k-1}) = i\Gamma(\alpha_{4,k}, \beta_{4,k}). \tag{27b}$$

The parameters α and β are chosen such that the mean value is preserved and the standard deviation is equal to 5 percent of the previous value of the parameter.

$$\mathbb{E}\{\Sigma_{3,k}|\Sigma_{3,k-1}\} = \Sigma_{3,k-1} \tag{28a}$$

$$\mathbb{E}\{\Sigma_{4,k}|\Sigma_{4,k-1}\} = \Sigma_{4,k-1} \tag{28b}$$

$$\mathrm{Std}\{\Sigma_{3,k}|\Sigma_{3,k-1}\} = 0.05\Sigma_{3,k-1} \tag{28c}$$

$$\mathrm{Std}\{\Sigma_{4,k}|\Sigma_{4,k-1}\} = 0.05\Sigma_{4,k-1}. \tag{28d}$$

Finally, the components of the measurement model (1b) are

$$\mathbf{y}_k = \begin{bmatrix}\mathrm{x}_k^{\mathrm{GPS}} & \mathrm{y}_k^{\mathrm{GPS}}\end{bmatrix}^{\mathrm{T}} \tag{29a}$$

$$h(\mathbf{x}_k, \mathbf{u}_k) = h(\mathbf{x}_k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}\mathbf{x}_k \tag{29b}$$

$$d(\mathbf{x}_k, \mathbf{u}_k) = d = I_2 \tag{29c}$$

and the measurement noise is zero mean and the covariance is assumed to be constant, i.e.,

$$\mathbf{e}_k = \begin{bmatrix}\mathbf{e}_{\mathrm{x}} & \mathbf{e}_{\mathrm{y}}\end{bmatrix}^{\mathrm{T}} = \mathcal{N}(0, R), \quad R = \Sigma^{\mathrm{GPS}} I_2. \tag{29d}$$

The input signals are the same as defined in (8). Since the unknown parameters are included in the state vector, there is no need to marginalize the noise parameters. Standard particle filter is used to estimate the state. This method will be referred to as *augmented particle filter* (AUG-PF). The filtering problem can also be solved with nonlinear Kalman filter (KF) based algorithms such as extended KF (EKF) or unscented KF (UKF), and this method will be referred to as *augmented Kalman filter* (AUG-KF).

4.2 Measurement Noise Estimation

In the fourth method the bias terms are assumed to appear in the noise again, but in this case they will appear in the measurement noise instead of the process noise, as in Section 2. Hence, this method is referred to as *marginalized particle filter measurement noise* (MPF-MN). The motion model is written as in (3), but since the radii error now appears in the measurement model, the state vector must be augmented with the yaw rate and velocity, i.e.,

$$\mathbf{x} = \begin{bmatrix} \mathrm{x} & \mathrm{y} & \psi & \dot{\psi} & \mathrm{v} \end{bmatrix}^{\mathrm{T}}. \tag{30}$$

The motion model, in (3), is augmented with the two additional states and the components of the motion model (1a) become

$$f(\mathbf{x}_k, \mathbf{u}_k) = f(\mathbf{x}_k) = \begin{bmatrix} \mathrm{x}_k + T\mathrm{v}_k \cos\psi_k \\ \mathrm{y}_k + T\mathrm{v}_k \sin\psi_k \\ \psi_k + T\dot{\psi}_k \\ \dot{\psi}_k \\ \mathrm{v}_k \end{bmatrix} \tag{31a}$$

$$g(\mathbf{x}_k, \mathbf{u}_k) = g(\mathbf{x}_k) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ T & 0 \\ 0 & T \end{bmatrix} \tag{31b}$$

with the zero mean measurement noise $\mathbf{w}$ with constant covariance

$$\mathbf{w}_k = \begin{bmatrix} \mathbf{w}_{\dot{\psi}} \\ \mathbf{w}_{\mathrm{v}} \end{bmatrix} = \mathcal{N}(0, Q), \quad Q = \mathrm{diag}(\sigma^2_{\delta_3}, \sigma^2_{\delta_4}). \tag{31c}$$

The measurement model is now nonlinear and the components in (1b) are

$$\mathbf{y}_k = \begin{bmatrix} \mathrm{x}^{\mathrm{GPS}} & \mathrm{y}^{\mathrm{GPS}} & \dot{\psi}_k^{\mathrm{virt}} & \mathrm{v}_k^{\mathrm{virt}} \end{bmatrix}^{\mathrm{T}} \tag{32a}$$

$$h(\mathbf{x}_k, \mathbf{u}_k) = h(\mathbf{x}_k) = \begin{bmatrix} \mathrm{x} & \mathrm{y} & \dot{\psi} & \mathrm{v} \end{bmatrix}^{\mathrm{T}} \tag{32b}$$

$$g(\mathbf{x}_k, \mathbf{u}_k) = g(\mathbf{u}_k) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & -\frac{\omega_3}{l_t} & \frac{\omega_4}{l_t} \\ 0 & \frac{\omega_3}{2} & \frac{\omega_4}{2} \end{bmatrix} \tag{32c}$$

where the noise is

$$\mathbf{e}_k = \begin{bmatrix} \mathbf{e}_{\mathrm{x}} & \mathbf{e}_{\mathrm{y}} & \delta_{3,k} & \delta_{4,k} \end{bmatrix}^{\mathrm{T}} = \mathcal{N}(\mu_k, \Sigma_k) \tag{32d}$$

$$\mu_k = \begin{bmatrix} 0 & 0 & \mu_{3,k} & \mu_{4,k} \end{bmatrix}^{\mathrm{T}} \tag{32e}$$

$$\Sigma_k = \mathrm{diag}([\Sigma_{\mathrm{x},k}, \Sigma_{\mathrm{y},k}, \Sigma_{3,k}, \Sigma_{4,k}]) \tag{32f}$$

The algorithm described in Section 3 applies to solve the estimation problem.

In this model the process noise parameters are known but the measurement noise parameters are to be estimated. The same approach described in Section 3 also

Table 1: *Summary of Estimation Methods*

Method	ω_i	$\dot{\psi}^{\text{virt}}$, v^{virt}	$\dot{\psi}$, v	$\boldsymbol{\theta}$	n_x	n_y	n_u
MPF-PN	**u**	**u**	-	**w**	3	2	2
AUG-PF/ AUG-KF	**u**	**u**	-	**x**	7	2	2
MPF-MN	**u**	**y**	**x**	**e**	5	4	0

applies to this model. At the sampling stage, the state prediction distribution can be used as the importance distribution. Then the weight update equation simplifies to,

$$w_k^{(i)} = w_{k-1}^{(i)} p(\mathbf{y}_k | \mathbf{x}_{0:k}, \mathbf{y}_{1:k-1}), \tag{33}$$

likewise the equation (22). The likelihood computation can be done by marginalizing the unknown noise parameters, as it is done for the process noise in equation (15)

$$p(\mathbf{y}_k | \mathbf{x}_{0:k}, \mathbf{y}_{1:k-1}) = \int p(\mathbf{y}_k | \mathbf{x}_k, \boldsymbol{\theta}) p(\boldsymbol{\theta} | \mathbf{x}_{0:k-1}, \mathbf{y}_{1:k-1}) d\boldsymbol{\theta}. \tag{34}$$

The resulting distribution is again a Student-t distribution, whose parameters can again be computed by conditioning on the realization of the measurement noise terms according to

$$\mathbf{e}_k^{(i)} = d^{\dagger}(\mathbf{x}_k^{(i)}, \mathbf{u}_k)(\mathbf{y}_k^{(i)} - h(\mathbf{x}_k^{(i)}, \mathbf{u}_k)), \tag{35}$$

The sufficient statistics update equations follow the same pattern. Here, † denotes the pseudo-inverse.

4.3 Summarizing the Four Methods

The four methods are summarized in Table 1. The way in which a variable is treated in the model is described in the table. A variable can be considered input **u**, output (measurement) **y**, state **x**, process noise **w** or measurement noise **e**. The dimension of the state vector, the input vector and the measurement vector are n_x, n_u and n_y, respectively. Note that the proposed method MPF-PN is the one with the smallest state dimension.

4.4 Wheel Velocity

The principle for rotational speed sensors is that a toothed wheel is attached to the rotating shaft. The teeth are also referred to as cogs, and the number of cogs are denoted N_{cog}. A magnet attached to one side causes a variation in the magnetic field that can be sensed by a Hall sensor. The variation is converted to a square wave signal with constant amplitude, where each edge corresponds to one edge of the toothed wheel.

The time between two or more edges is then registered and converted into angular speed as follows. The angle between each tooth is $\frac{2\pi}{N_{\text{cog}}}$. The time when tooth ℓ is passing is denoted τ_ℓ, and the corresponding angle is denoted $\varphi_\ell = \ell \frac{2\pi}{N_{\text{cog}}}$. The angular speed $\omega(kT)$ at time kT, where T is the sampling time and k is the time step number, can now be approximated as

$$\hat{\omega}(kT) = \frac{2\pi(\ell_k - \ell_{k-1})}{N_{\text{cog}}(\tau_{\ell_k} - \tau_{\ell_{k-1}})}. \tag{36}$$

Here, τ_{ℓ_k} denotes the time when the last cog ℓ_k passed before time $\tau = kT$. Note that the angle can be recovered by summing $\hat{\omega}(kT)$ over time.

In the motion model (7) the angular speed ω is multiplied with the sample time T, which gives an angle. However, since the sensor initially measures the number of cogs passed in a certain time according to equation (36) there is no need to transform it into an angular velocity. It is more efficiently to instead use

$$T\omega_k = \frac{2\pi(\ell_k - \ell_{k-1})}{N_{\text{cog}}T}, \tag{37}$$

where $\ell_k - \ell_{k-1}$ is the number of cogs passed in the sampling time T. More details about sampling and quantization effects in wheel speed sensors are described in Gustafsson (2010b).

5 Results

In the experiments, the measurements were collected with a passenger car equipped with standard vehicle sensors, such as wheel speed sensors, and a GPS receiver, see Figure 2. The vehicle is further equipped with an additional and more accurate IMU, besides the standard IMU already mounted in the car, and an optical velocity sensor. These two additional sensors were used to calibrate the setup, but were not further used to produce the results presented.

In regions where the car moves at low velocities, the steering wheel angle measurement was utilized as follows, in order to avoid quantization problems of the wheel cogs

$$\mathsf{x}_{k+1} = \mathsf{x}_k + T(\mathsf{v}_k^{\text{virt}} + \frac{\omega_3\delta_3}{2} + \frac{\omega_4\delta_4}{2})\cos\psi_k \tag{38a}$$

$$\mathsf{y}_{k+1} = \mathsf{y}_k + T(\mathsf{v}_k^{\text{virt}} + \frac{\omega_3\delta_3}{2} + \frac{\omega_4\delta_4}{2})\sin\psi_k \tag{38b}$$

$$\psi_{k+1} = \begin{cases} \psi_k + T(\dot{\psi}_k^{\text{virt}} + \frac{\omega_4\delta_4}{l_t} - \frac{\omega_3\delta_3}{l_t}) & \text{if } \mathsf{v} > \gamma \\ \psi_k + T\delta_F(\mathsf{v}_k^{\text{virt}} + \frac{\omega_3\delta_3}{2} + \frac{\omega_4\delta_4}{2})/l_b & \text{if } \mathsf{v} < \gamma \end{cases} \tag{38c}$$

The GPS measurements of the 12 km test round is shown as a red solid line in Figure 3. It is overlayed by the estimated trajectory, which is black-white dashed. The photo is a very accurate flight photo (obtained from the Swedish mapping, cadastral and land registration authority), which can be used as ground truth to visualize the quality of the trajectory estimate. The round took about 18 min to drive and it starts and ends in urban area of Linköping, in the upper right corner in Figure 3. The test vehicle is driving clockwise, first on a small rural

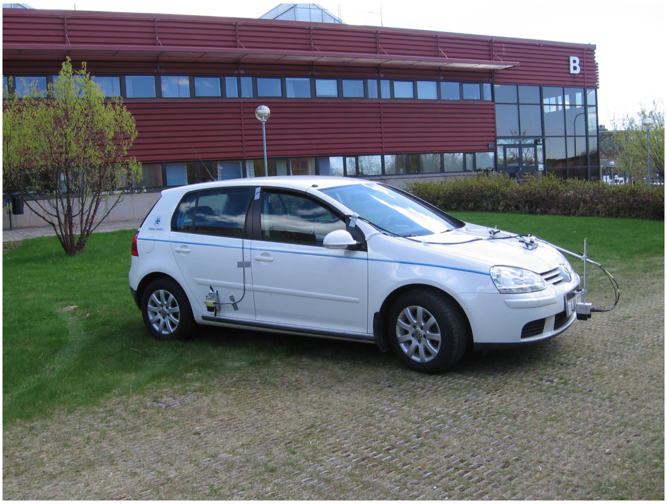


Figure 2: The test vehicle of Linköping University is logging standard CAN data. The vehicle is in addition equipped with a GPS receiver, an IMU and an optical velocity sensor.

road, and then on the left side of the figure entering a straight national highway, before driving back to urban area on the top of the figure. The test was performed two times, first with balanced tire pressure and thereafter with unbalanced tire pressure, where the pressure of the rear left tire was released to be approximately 50% of the right tire pressure.

For the first round the pressure of the rear wheel tires was adjusted to be equal 2.8 bar on both tires. The estimated parameters, i.e., the mean and the covariance for the left and the right wheel are shown in the Figure 4a and 4b, respectively. It is clearly visible that the radius error is similar for the left and the right wheel and and all the methods perform well in estimating the mean values μ_3 and μ_4 except the MPF-MN method, which makes a jump around time $t = 540$. The performances of the methods differ in estimating the radius error difference between the left and right wheels which is shown in Figure 5. The proposed method, MPF-PN, performs the best among the four estimators. The AUG-PF, and MPF-MN methods produce more erratic estimates than the MPF-PN and the AUG-KF. The peak value of the estimation error of MPF-PN is smaller than that of AUG-KF.

For the second round the pressure of the rear left tire was released to 1.5 bar. Comparing Figure 6a with Figure 6b it is visible that the pressure reduction leads to a smaller μ_3 than μ_4 value. The behavior of the algorithms are similar to the balanced case. The difference among the methods can be observed better in Figure 7, where the difference between the left and the right tire radii errors $\mu_3 - \mu_4$ is shown. Here again, The AUG-PF, and MPF-MN methods produce more erratic estimates. The MPF-PN and the AUG-PF produces smoother estimates and the MPF-PN results are better than AUG-KF. All four methods reach a value of a rela-

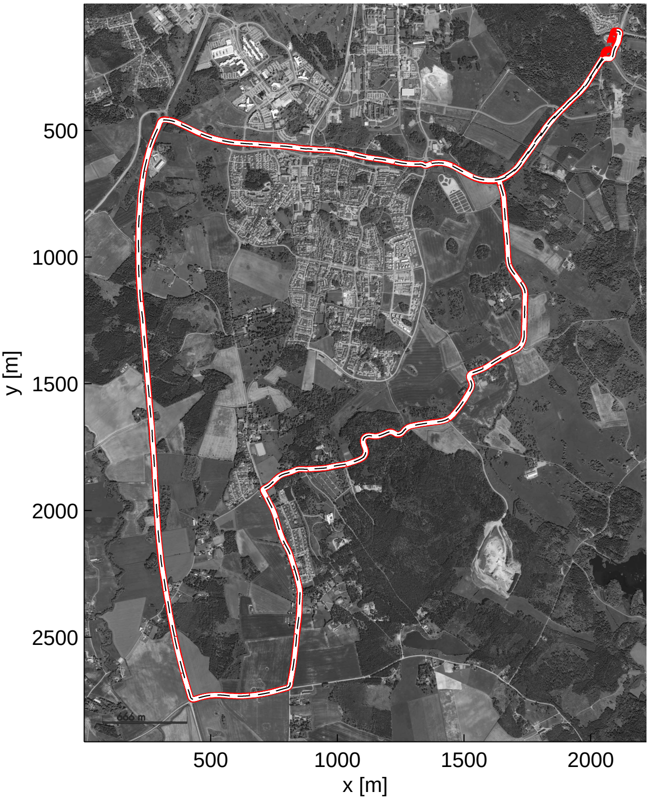
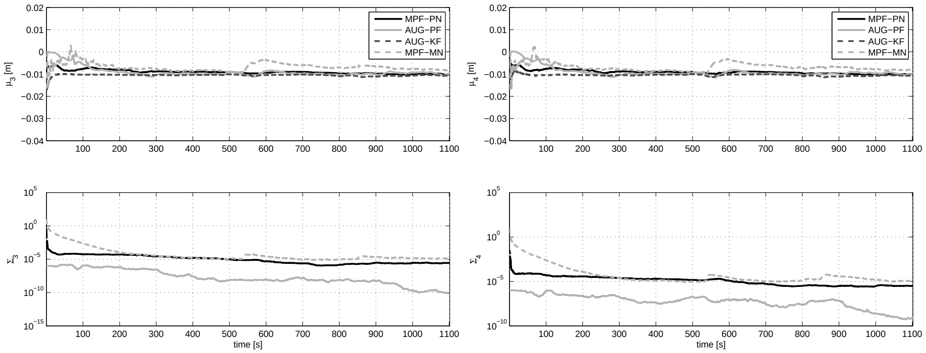


Figure 3: *The red line is GPS position measurements and the black-white dashed line is the estimated driven trajectory. The experiment starts and ends at a roundabout in the upper right corner. (©Lantmäteriet Medgivande I2011/1405, reprinted with permission)*



***(a)** Left Wheel* ***(b)** Right Wheel*

***Figure 4:** Tire radius error of the left and right rear wheels, in Figure (a) and (b), respectively. The upper plot in each sub figure shows the mean values μ and the lower plot the covariance estimates Σ. These plots show the situation with balanced wheel pressure, and the radius error in Figure (a) and (b) is very similar. The black solid line is the MPF-PN estimate, the gray solid line is the AUG-PF estimate, the black dashed line is the AUG-KF estimate and the gray dashed line is from the MPF-PN.*

tive difference of approximately 1.5 mm.

In order to analyze the methods numerically an artificial data set, which simulates a pressure drop, has been created. The data set with balanced wheels is used as a basis for that purpose, and an artificial wheel speed is computed according to

$$\bar{\omega}_4 = \omega_4 \frac{r}{r + \bar{\delta}_4} \tag{39}$$

where the artificial tire radius error is given by

$$\bar{\delta}_4 = \begin{cases} -\frac{k\delta_{\max}}{K_2} & k < K_2 \\ \delta_{\max} & k \geq K_2 \end{cases}. \tag{40}$$

The virtual measurements (4) are recalculated based on the artificial wheel speed as follows.

$$\bar{\mathrm{v}}^{\mathrm{virt}} = \frac{\omega_3 r + \bar{\omega}_4 r}{2} \tag{41a}$$

$$\bar{\dot{\psi}}^{\mathrm{virt}} = \frac{\bar{\omega}_4 r - \omega_3 r}{l_t}. \tag{41b}$$

For the example presented here, the values $\delta_{\max} = -2.5 \cdot 10^{-3}$m and $K_2 = 5808$ where chosen. The simulated tire radii difference versus time, and its estimates are plotted in Figure 8. Here again the MPF-PN produces the minimum minimum root mean square error (RMSE) among all methods. Further, we compare the

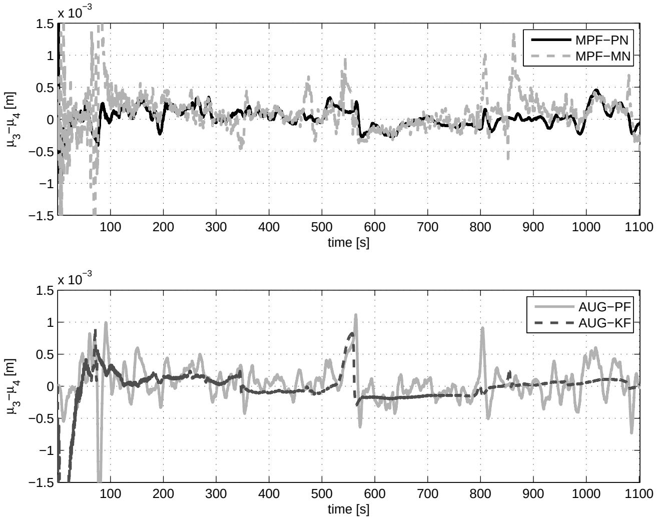


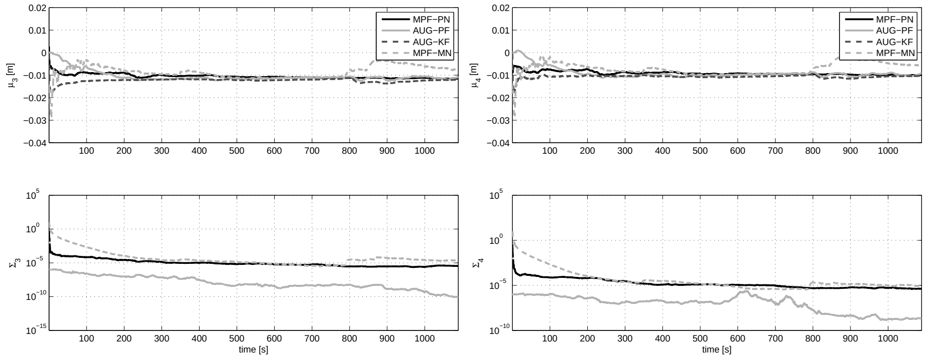
Figure 5: *The figure shows the tire radius error difference between the left and the right rear wheels, for the situation with balanced wheel pressures. Since the pressure is equal in the left and the right wheel the radius difference between them is zero as expected.*

average RMSE of the algorithms over 100 Monte Carlo (MC) runs and present the change in the RMSE errors while varying the number of particles.

Next results are based on 100 MC runs for all methods (except for AUG-KF, which is deterministic). The effects of changing the number of particles is examined in Figure 9a. The MC runs are repeated for 100, 500, 1000, and 5000 particles and the average RMSE of the difference between μ_3 and μ_4 is compared. This is performed under the assumption that $\mu_3 - \mu_4$ is equal to the artificially added error, according to (40), which serves as the true value. In Figure 9a the average RMSE are plotted with respect to the different number of particles. The proposed MPF-PN estimate has the smallest RMSE, and for instance the RMSE for 100 particles corresponds to the same RMSE of the AUG-PF using 500 particles. The AUG-KF approach produces here the worst RMSE.

Some of the methods are sensitive to divergence when the number of particles becomes to small. For this reason the divergence rate is analyzed for the three PF based methods and plotted with respect to the number of particles in Figure 9b. The algorithms are run on the complete 18 min data set and if the simulation diverges during one MC run it is counted and compared to the total number of 100 MC runs. In the figure it is shown that the proposed MPF-PN approach and the AUG-KF approach always converges, whereas the AUG-PF and the MPF-MN approaches converges for all 100 MC runs when using at least 1000 particles.

Finally the average runtime per sample of a single MC run are given in Table 2. Here it is obvious that the proposed process MPF-PN method and the AUG-PF



(a) Left Wheel *(b) Right Wheel*

Figure 6: *Tire radius error of the left and right rear wheels, in Figure (a) and (b), respectively. The upper plot in each sub figure shows the mean values μ and the lower plot the covariance estimates Σ. These plots show the situation with unbalanced wheel pressure, and as expected the radius error of the left wheel in Figure (a) is larger than the error of the right wheel in (b). The black solid line is the MPF-PN estimate, the gray solid line is the AUG-PF estimate, the black dashed line the AUG-KF estimate and the gray dashed line the MPF-PN estimate.*

Table 2: *Computation Time [ms]*

	Number of particles			
Method	100	500	1000	5000
MPF-PN	0.39	0.70	1.1	5.9
AUG-PF	0.41	0.77	1.2	7.4
AUG-KF	–	– 0.35 –		–
MPF-MN	6.2	29	56	290

estimate are in the same order of complexity, since the computation times are similar. The AUG-KF approach is in the same order of magnitude as the AUG-PF with 100 particles, but as seen in Figure 9a, with a much worse RMSE. The MPF-MN is by far the slowest method.

6 Conclusion

In this study, we address the problem of joint estimation of unknown tire radii and the trajectory of a four wheeled vehicle based on GPS and wheel angular velocity measurements. The problem is defined in a Bayesian framework and an efficient method that utilizes marginalized particle filters is proposed in order to accomplish the difficult task of joint parameter and state estimation. The algo-

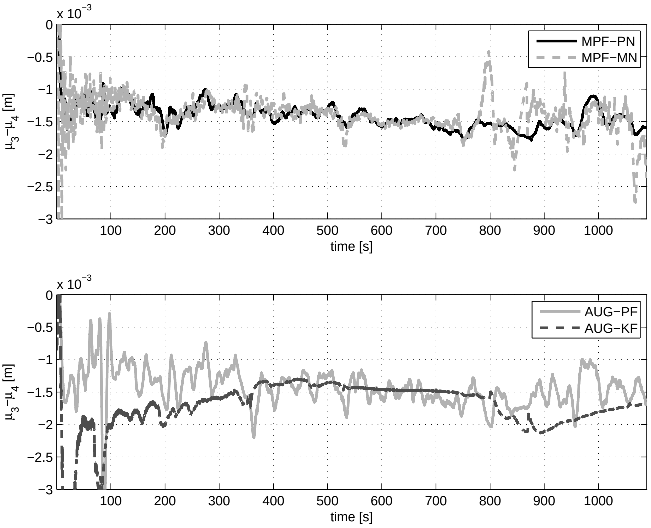


Figure 7: The figure shows the tire radius error difference between the left and the right rear wheels, for the situation with unbalanced wheel pressures. The pressure of the left wheel is reduced by 50%, which leads to a radius reduction of about 1.5 mm*.*

rithm is tested on real data experiments. The results show that it is possible to estimate relative tire radius difference within sub-millimeter accuracy.

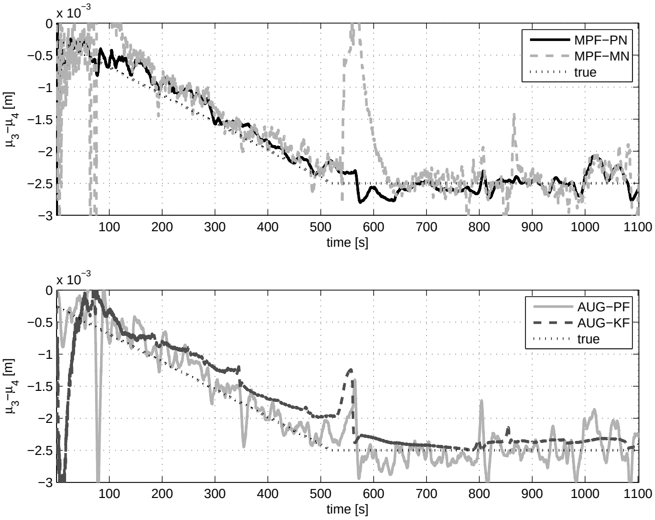
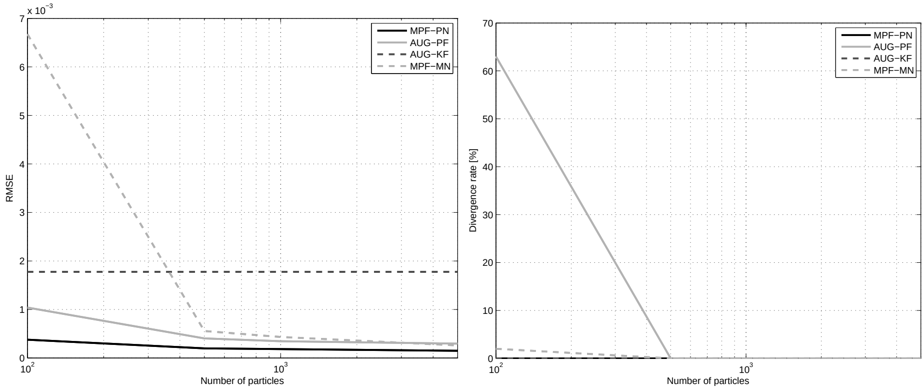


Figure 8: *The figure shows the tire radius error difference between the left and the right rear wheels, for the situation with artificial wheel radii error of* −2.5 mm. *The example is aimed at simulating a slow puncture. The resulting estimates are close to the expected values.*



*(**a**) RMSE* *(**b**) Divergence Rate*

Figure 9: *Figure (a) shows the RMSE over number of particles. Note that, the proposed MPF-PN approach has the same RMSE value for 100 particles as the AUG-PF has for 500 particles. Figure 9b shows the divergence rate over number of particles. The proposed MPF-PN approach did not diverge at any of the MC runs, however the AUG-PF converges only if more that 1000 particles are used.*

Bibliography

C. R. Carlson and J. C. Gerdes. Consistent nonlinear estimation of longitudinal tire stiffness and effective radius. *IEEE Transactions on Control Systems Technology*, 13(6):1010–1020, November 2005.

A. Censi, L. Marchionni, and G. Oriolo. Simultaneous maximum-likelihood calibration of odometry and sensor parameters. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2098–2103, Pasadena, Canada, May 2008.

V. E. Ersanilli, P. J. Reeve, K. J. Burnham, and P. J. King. A continuous-time model-based tyre fault detection algorithm utilising a Kalman state estimator approach. In *Proceedings of the 7th Workshop on Advanced Control and Diagnosis*, Zielona Góra, Poland, November 2009.

F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, Lund, Sweden, 2010a.

F. Gustafsson. Rotational speed sensors: Limitations, pre-processing and automotive applications. *IEEE Instrumentation & Measurement Magazine*, 13(2): 16–23, March 2010b.

F. Gustafsson, S. Ahlqvist, U. Forssell, and Ni. Persson. Sensor fusion for accurate computation of yaw rate and absolute velocity. In *Proceedings of the SAE World Congress*, SAE paper 2001-01-1064, Detroit, MI, USA, April 2001.

S. Julier and H. Durrant-Whyte. Process models for the high-speed navigation of road vehicles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 101–105, May 1995.

M. Kárný. *Optimized Bayesian Dynamic Advising: Theory and Algorithms*. Springer, London, 2006. ISBN 978-1-84628-254-6.

J. Liu and M. West. Combined parameter and state estimation in simulation-based filtering. In A. Doucet, N. De Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

C. Lundquist, E. Özkan, and F. Gustafsson. Tire radii estimation using a marginalized particle filter. *IEEE Transactions on Intelligent Transportation Systems*, 2011. Submitted.

S. L. Miller, B. Youngberg, A. Millie, P. Schweizer, and J. C. Gerdes. Calculating longitudinal wheel slip and tire parameters using gps velocity. In *IEEE American Control Conference*, volume 3, pages 1800–1805, June 2001.

N.K. M'sirdi, A. Rabhi, L. Fridman, J. Davila, and Y. Delanne. Second order sliding mode observer for estimation of velocities, wheel sleep, radius and stiffness. In *IEEE American Control Conference*, pages 3316–3321, June 2006.

E. Özkan, C. Lundquist, and F. Gustafsson. A Bayesian approach to jointly estimate tire radii and vehicle trajectory. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, Washington DC, USA, October 2011.

S. Patwardhan, H.-S. Tan, and M. Tomizuka. Experimental results of a tire-burst controller for ahs. *Control Engineering Practice*, 5(11):1615–1622, 1997.

N. Persson, S. Ahlqvist, U. Forssell, and F. Gustafsson. Low tire pressure warning system using sensor fusion. In *Proceedings of the Automotive and Transportation Technology Congress*, SAE paper 2001-01-3337, Barcelona, Spain, October 2001.

V. Peterka. Bayesian system identification. *Automatica*, 17(1):41–53, January 1981.

S. Saha, E. Özkan, F. Gustafsson, and V. Smidl. Marginalized particle filters for Bayesian estimation of Gaussian noise. In *Proceedings of the International Conference on Information Fusion*, Edinburgh, Scotland, July 2010.

H. Shraim, A. Rabhi, M. Ouladsine, N.K. M'Sirdi, and L. Fridman. Estimation and analysis of the tire pressure effects on the comportment of the vehicle center of gravity. In *International Workshop on Variable Structure Systems*, pages 268–273, June 2006.

S. Velupillai and L. Guvenc. Tire pressure monitoring. *IEEE Control Systems Magazine*, 27(6):22–25, December 2007.

Index

PhD Dissertations
Division of Automatic Control
Linköping University

M. Millnert: Identification and control of systems subject to abrupt changes. Thesis No. 82, 1982. ISBN 91-7372-542-0.

A. J. M. van Overbeek: On-line structure selection for the identification of multivariable systems. Thesis No. 86, 1982. ISBN 91-7372-586-2.

B. Bengtsson: On some control problems for queues. Thesis No. 87, 1982. ISBN 91-7372-593-5.

S. Ljung: Fast algorithms for integral equations and least squares identification problems. Thesis No. 93, 1983. ISBN 91-7372-641-9.

H. Jonson: A Newton method for solving non-linear optimal control problems with general constraints. Thesis No. 104, 1983. ISBN 91-7372-718-0.

E. Trulsson: Adaptive control based on explicit criterion minimization. Thesis No. 106, 1983. ISBN 91-7372-728-8.

K. Nordström: Uncertainty, robustness and sensitivity reduction in the design of single input control systems. Thesis No. 162, 1987. ISBN 91-7870-170-8.

B. Wahlberg: On the identification and approximation of linear systems. Thesis No. 163, 1987. ISBN 91-7870-175-9.

S. Gunnarsson: Frequency domain aspects of modeling and control in adaptive systems. Thesis No. 194, 1988. ISBN 91-7870-380-8.

A. Isaksson: On system identification in one and two dimensions with signal processing applications. Thesis No. 196, 1988. ISBN 91-7870-383-2.

M. Viberg: Subspace fitting concepts in sensor array processing. Thesis No. 217, 1989. ISBN 91-7870-529-0.

K. Forsman: Constructive commutative algebra in nonlinear control theory. Thesis No. 261, 1991. ISBN 91-7870-827-3.

F. Gustafsson: Estimation of discrete parameters in linear systems. Thesis No. 271, 1992. ISBN 91-7870-876-1.

P. Nagy: Tools for knowledge-based signal processing with applications to system identification. Thesis No. 280, 1992. ISBN 91-7870-962-8.

T. Svensson: Mathematical tools and software for analysis and design of nonlinear control systems. Thesis No. 285, 1992. ISBN 91-7870-989-X.

S. Andersson: On dimension reduction in sensor array signal processing. Thesis No. 290, 1992. ISBN 91-7871-015-4.

H. Hjalmarsson: Aspects on incomplete modeling in system identification. Thesis No. 298, 1993. ISBN 91-7871-070-7.

I. Klein: Automatic synthesis of sequential control schemes. Thesis No. 305, 1993. ISBN 91-7871-090-1.

J.-E. Strömberg: A mode switching modelling philosophy. Thesis No. 353, 1994. ISBN 91-7871-430-3.

K. Wang Chen: Transformation and symbolic calculations in filtering and control. Thesis No. 361, 1994. ISBN 91-7871-467-2.

T. McKelvey: Identification of state-space models from time and frequency data. Thesis No. 380, 1995. ISBN 91-7871-531-8.

J. Sjöberg: Non-linear system identification with neural networks. Thesis No. 381, 1995. ISBN 91-7871-534-2.

R. Germundsson: Symbolic systems – theory, computation and applications. Thesis No. 389, 1995. ISBN 91-7871-578-4.

P. Pucar: Modeling and segmentation using multiple models. Thesis No. 405, 1995. ISBN 91-7871-627-6.

H. Fortell: Algebraic approaches to normal forms and zero dynamics. Thesis No. 407, 1995. ISBN 91-7871-629-2.

A. Helmersson: Methods for robust gain scheduling. Thesis No. 406, 1995. ISBN 91-7871-628-4.

P. Lindskog: Methods, algorithms and tools for system identification based on prior knowledge. Thesis No. 436, 1996. ISBN 91-7871-424-8.

J. Gunnarsson: Symbolic methods and tools for discrete event dynamic systems. Thesis No. 477, 1997. ISBN 91-7871-917-8.

M. Jirstrand: Constructive methods for inequality constraints in control. Thesis No. 527, 1998. ISBN 91-7219-187-2.

U. Forssell: Closed-loop identification: Methods, theory, and applications. Thesis No. 566, 1999. ISBN 91-7219-432-4.

A. Stenman: Model on demand: Algorithms, analysis and applications. Thesis No. 571, 1999. ISBN 91-7219-450-2.

N. Bergman: Recursive Bayesian estimation: Navigation and tracking applications. Thesis No. 579, 1999. ISBN 91-7219-473-1.

K. Edström: Switched bond graphs: Simulation and analysis. Thesis No. 586, 1999. ISBN 91-7219-493-6.

M. Larsson: Behavioral and structural model based approaches to discrete diagnosis. Thesis No. 608, 1999. ISBN 91-7219-615-5.

F. Gunnarsson: Power control in cellular radio systems: Analysis, design and estimation. Thesis No. 623, 2000. ISBN 91-7219-689-0.

V. Einarsson: Model checking methods for mode switching systems. Thesis No. 652, 2000. ISBN 91-7219-836-2.

M. Norrlöf: Iterative learning control: Analysis, design, and experiments. Thesis No. 653, 2000. ISBN 91-7219-837-0.

F. Tjärnström: Variance expressions and model reduction in system identification. Thesis No. 730, 2002. ISBN 91-7373-253-2.

J. Löfberg: Minimax approaches to robust model predictive control. Thesis No. 812, 2003. ISBN 91-7373-622-8.

J. Roll: Local and piecewise affine approaches to system identification. Thesis No. 802, 2003. ISBN 91-7373-608-2.

J. Elbornsson: Analysis, estimation and compensation of mismatch effects in A/D converters. Thesis No. 811, 2003. ISBN 91-7373-621-X.

O. Härkegård: Backstepping and control allocation with applications to flight control. Thesis No. 820, 2003. ISBN 91-7373-647-3.

R. Wallin: Optimization algorithms for system analysis and identification. Thesis No. 919, 2004. ISBN 91-85297-19-4.

D. Lindgren: Projection methods for classification and identification. Thesis No. 915, 2005. ISBN 91-85297-06-2.

R. Karlsson: Particle Filtering for Positioning and Tracking Applications. Thesis No. 924, 2005. ISBN 91-85297-34-8.

J. Jansson: Collision Avoidance Theory with Applications to Automotive Collision Mitigation. Thesis No. 950, 2005. ISBN 91-85299-45-6.

E. Geijer Lundin: Uplink Load in CDMA Cellular Radio Systems. Thesis No. 977, 2005. ISBN 91-85457-49-3.

M. Enqvist: Linear Models of Nonlinear Systems. Thesis No. 985, 2005. ISBN 91-85457-64-7.

T. B. Schön: Estimation of Nonlinear Dynamic Systems — Theory and Applications. Thesis No. 998, 2006. ISBN 91-85497-03-7.

I. Lind: Regressor and Structure Selection — Uses of ANOVA in System Identification. Thesis No. 1012, 2006. ISBN 91-85523-98-4.

J. Gillberg: Frequency Domain Identification of Continuous-Time Systems Reconstruction and Robustness. Thesis No. 1031, 2006. ISBN 91-85523-34-8.

M. Gerdin: Identification and Estimation for Models Described by Differential-Algebraic Equations. Thesis No. 1046, 2006. ISBN 91-85643-87-4.

C. Grönwall: Ground Object Recognition using Laser Radar Data – Geometric Fitting, Performance Analysis, and Applications. Thesis No. 1055, 2006. ISBN 91-85643-53-X.

A. Eidehall: Tracking and threat assessment for automotive collision avoidance. Thesis No. 1066, 2007. ISBN 91-85643-10-6.

F. Eng: Non-Uniform Sampling in Statistical Signal Processing. Thesis No. 1082, 2007. ISBN 978-91-85715-49-7.

E. Wernholt: Multivariable Frequency-Domain Identification of Industrial Robots. Thesis No. 1138, 2007. ISBN 978-91-85895-72-4.

D. Axehill: Integer Quadratic Programming for Control and Communication. Thesis No. 1158, 2008. ISBN 978-91-85523-03-0.

G. Hendeby: Performance and Implementation Aspects of Nonlinear Filtering. Thesis No. 1161, 2008. ISBN 978-91-7393-979-9.

J. Sjöberg: Optimal Control and Model Reduction of Nonlinear DAE Models. Thesis No. 1166, 2008. ISBN 978-91-7393-964-5.

D. Törnqvist: Estimation and Detection with Applications to Navigation. Thesis No. 1216, 2008. ISBN 978-91-7393-785-6.

P-J. Nordlund: Efficient Estimation and Detection Methods for Airborne Applications. Thesis No. 1231, 2008. ISBN 978-91-7393-720-7.

H. Tidefelt: Differential-algebraic equations and matrix-valued singular perturbation. Thesis No. 1292, 2009. ISBN 978-91-7393-479-4.

H. Ohlsson: Regularization for Sparseness and Smoothness — Applications in System Identification and Signal Processing. Thesis No. 1351, 2010. ISBN 978-91-7393-287-5.

S. Moberg: Modeling and Control of Flexible Manipulators. Thesis No. 1349, 2010. ISBN 978-91-7393-289-9.

J. Wallén: Estimation-based iterative learning control. Thesis No. 1358, 2011. ISBN 978-91-7393-255-4.

J. Hol: Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS. Thesis No. 1368, 2011. ISBN 978-91-7393-197-7.

D. Ankelhed: On the Design of Low Order H-infinity Controllers. Thesis No. 1371, 2011. ISBN 978-91-7393-157-1.