

Mark Schiementz

---

# Postprocessing Architecture for an Automotive Radar Network

---



Cuvillier Verlag Göttingen

# Postprocessing Architecture for an Automotive Radar Network

Vom Promotionsausschuß der  
Technischen Universität Hamburg-Harburg  
zur Erlangung des akademischen Grades  
Doktor-Ingenieur (Dr.-Ing.)  
genehmigte Dissertation

von

Mark Schiementz

aus

Hamburg

2005

## **Bibliografische Information Der Deutschen Bibliothek**

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

1. Aufl. - Göttingen : Cuvillier, 2005  
Zugl.: Hamburg-Harburg, Univ., Diss., 2005  
ISBN 3-86537-517-0

1. Gutachter: Prof. Dr. Hermann Rohling
2. Gutachter: Prof. Dr. Herbert Werner

Tag der mündlichen Prüfung: 22. Juni 2005

© CUVILLIER VERLAG, Göttingen 2005  
Nonnenstieg 8, 37075 Göttingen  
Telefon: 0551-54724-0  
Telefax: 0551-54724-21  
[www.cuvillier.de](http://www.cuvillier.de)

Alle Rechte vorbehalten. Ohne ausdrückliche Genehmigung des Verlages ist es nicht gestattet, das Buch oder Teile daraus auf fotomechanischem Weg (Fotokopie, Mikrokopie) zu vervielfältigen.

1. Auflage, 2005  
Gedruckt auf säurefreiem Papier

ISBN 3-86537-517-0

# Danksagung

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter im Arbeitsbereich Nachrichtentechnik der Technischen Universität Hamburg-Harburg. Für die mir gegebene Unterstützung, aber auch die erheblichen Freiräume während dieser Zeit, möchte ich mich bei Herrn Prof. Dr. Hermann Rohling besonders herzlich bedanken. Ebenso gilt mein Dank den Herren Prof. Dr. Herbert Werner und Prof. Dr. Wolfgang Meyer für das Interesse an dieser Dissertation sowie für die Übernahme des Prüfungsvorsitzes.

Des weiteren bedanke ich mich bei allen Kolleginnen und Kollegen des Arbeitsbereiches für das sehr angenehme Arbeitsleben miteinander, speziell bei Herrn Dr.-Ing. Urs Lübbert für die interessanten Diskussionen sowie für eine außergewöhnlich gute Zusammenarbeit im Projekt RadarNet.

Mein besonderer Dank gilt meinen Eltern, die mich bei all meinen Entscheidungen während dieser Zeit unterstützt haben. Leider konnte mein Vater aufgrund schwerer Krankheit den Abschluß meiner Promotion nicht mehr miterleben. Ihm möchte ich diese Dissertation widmen.

Schließlich möchte ich mich ganz herzlich bei meiner Lebensgefährtin Katrin bedanken. Ohne ihre Unterstützung, Aufmunterung und Geduld wäre diese Arbeit unter den gegebenen Randbedingungen nicht möglich gewesen.

München, im Juni 2005

Mark Schiementz



Für meinen Vater



# Zusammenfassung

Vielfältige radarbasierte Applikationen für den Automobilbereich werden zur Zeit in der Forschung untersucht oder befinden sich bereits in der Vor- oder Serienentwicklung der Automobilindustrie. Seit ein paar Jahren sind radarbasierte Automatic Cruise Control (ACC) Systeme auf dem Markt, welche zunächst dem Luxussegment vorbehalten waren und sich dort etabliert haben. Nun werden neue radarbasierte Applikationen angestrebt, welche dem Fahrer weitere, unterstützende komfort- oder sicherheitsrelevante Funktionen zur Verfügung stellen sollen. So wird z.B. an Fahrerassistenz-Systemen wie dem Park Manöver Assistenten, dem Spurwechsellassistenten oder einem ACC/Stop & Go System gearbeitet. Die auf dem Markt befindlichen ACC-Systeme setzen vornehmlich Fernbereichssensorik im mm-Wellenbereich bei 77 GHz ein, welche in ihrer bestehenden Form den Anforderungen der neuen Applikationen nicht genügen. Deshalb wurden in den letzten Jahren auf dem Gebiet der radarbasierten Nahbereichssensorik verschiedene 24 GHz Sensoren entwickelt, die mit der vorhandenen Fernbereichssensorik zu einem Radarnetzwerk zusammengefaßt werden sollen, um die Anforderungen der angestrebten Applikationen zu erfüllen.

Im Straßenverkehr hat es die Radarsensorik mit ausgedehnten Zielen in Mehrzielsituationen mit einer großen Anzahl von bewegten und nicht bewegten Objekten zu tun. Diese werden von heutigen Sensoren immer besser detektiert. Um eine entsprechende Reaktion des Fahrers oder des Fahrerassistenzsystems auf die detektierten Objekte zu ermöglichen, müssen die Meßdaten der verschiedenen Sensoren mit Hilfe entsprechender Signalverarbeitung aufbereitet werden, so daß die übermittelten Daten einen hohen Anteil relevanter, fehlerfreier Information enthalten. Diese Signalverarbeitung findet im sogenannten Postprocessing statt.

Die vorgestellte Arbeit präsentiert eine Postprocessing Architektur für ein Multi-Sensor Radarnetzwerk. Diese extrahiert die von der Applikation benötigte Information aus den Radarmeßdaten, d.h. Fehldetektionen werden aussortiert, relevante Informationen stabilisiert und zusätzliche Informationen mit Hilfe der eigenen Fahrzeugdaten generiert. Einleitend werden die Meßrandbedingungen der Radarsensorik im Straßenverkehr erläutert und verschiedene 24 GHz Radarnetzwerkkonzepte untersucht sowie bewertet. Anschließend werden verschiedene Schätzalgorithmen zur Bestimmung der Position und Geschwindigkeit der detektierten Ziele sowie verschiedene Trackingfilter zu deren zeitlichen Verfolgung hergeleitet und evaluiert. Aus den gewonnenen Erkenntnissen der Untersuchung dieser Schätz- und Trackingverfahren wird eine mehrzielfähige Postprocessing Architektur für ein Radarnetzwerk bestehend aus zwei 24 GHz Monopulsradarsensoren hergeleitet. Die für ein mehrzielfähiges System benötigte Datenzuordnung wird anhand der Detektionseigenschaften der verwendeten Sensorik erläutert. Speziell die Trennung der großen Anzahl von detektierten Ziele in bewegte und unbewegte Ziele unter Einbeziehung der eigenen Fahrzeuginformation stellt für diese Datenzuordnung eine wichtige Grundlage dar. Außerdem wird der Tatsache Rechnung getragen, daß die beobachteten Objekte als ausgedehnt betrachtet werden müssen und somit mehrere Detektion in einem Meßzyklus erzeugen können. Neben den analytischer Untersuchungen der verschiedenen Postprocessing Algorithmen, stellt die empirische Evaluation der hergeleiteten Postpro-

cessing Architektur in einem Erprobungsfahrzeug einen Schwerpunkt dieser Arbeit dar.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Automotive Radar Applications . . . . .	2
1.2	Radar Measurement Principles . . . . .	9
1.2.1	Pulse Radar . . . . .	12
1.2.2	Linear Frequency Modulated Continuous Wave Radar (LFM-CW) . . . . .	14
1.2.3	Object Reflection Properties . . . . .	18
1.2.4	Summary . . . . .	21
1.3	Radar Sensors for Automotive Applications . . . . .	23
1.3.1	High Range Resolution Pulse Radar Sensor . . . . .	23
1.3.2	Monopulse Radar Sensor . . . . .	24
1.4	Multi-Sensor Radar Network . . . . .	26
1.4.1	Radar Network Concepts . . . . .	26
1.4.2	Radar Network Performance . . . . .	28
1.4.3	Radar Network Type 1 . . . . .	33
1.4.4	Radar Network Type 2 . . . . .	37
1.4.5	Single MR Sensor . . . . .	38
1.4.6	Summary . . . . .	40
1.5	A Postprocessing Structure for a Radar Network . . . . .	43
<b>2</b>	<b>Kinematic State Estimation</b>	<b>47</b>
2.1	Position and Velocity Estimation . . . . .	48
2.1.1	Geometric Preliminaries . . . . .	49
2.1.2	Kinematic State Determination from Single-Sensor Measurements	50

2.1.3	Linear Least Squares Estimation from Multiple Sensor Measurements	53
2.1.4	Nonlinear Least Squares Estimation from Multiple Sensor Measurements . . . . .	54
2.1.5	Incorporation of Host Vehicle Information . . . . .	59
2.1.6	Summary . . . . .	61
2.2	Target Tracking . . . . .	62
2.2.1	Kalman Filtering . . . . .	62
2.2.2	Linear Kalman Filter . . . . .	64
2.2.3	Target Dynamic Models . . . . .	67
2.2.4	Extended Kalman Filter . . . . .	71
2.2.5	Unscented Kalman Filter . . . . .	74
2.2.6	Comparison of Different Kinematic State Estimation Systems . .	78
2.2.7	Filter Decoupling . . . . .	86
2.2.8	Adaptive Filtering Based on Maneuver Detection . . . . .	89
2.3	Chapter Conclusions . . . . .	92
<b>3</b>	<b>A Multi-Target Postprocessing Architecture</b>	<b>93</b>
3.1	Postprocessing Architecture for a 2 UMRR Network . . . . .	94
3.2	Sensor-Level Postprocessing . . . . .	95
3.2.1	Simple Target Classification . . . . .	97
3.2.2	Sensor-Level Clustering . . . . .	101
3.2.3	Sensor-Level Data Association . . . . .	103
3.2.4	Sensor-Level Tracking . . . . .	112
3.3	Central-Level Postprocessing . . . . .	118
3.3.1	Central-Level Clustering . . . . .	119
3.3.2	Central-Level Data Association . . . . .	123
3.3.3	Central-Level Tracking . . . . .	124
3.4	Choice of Tracking Filter Parameters . . . . .	126
3.5	Chapter Conclusions . . . . .	128
<b>4</b>	<b>Experimental System</b>	<b>129</b>

---

4.1	Test Vehicle and Equipment . . . . .	129
4.2	Software Implementation Tool . . . . .	132
4.3	Real Measurement Situations . . . . .	134
4.3.1	Highway Scenario . . . . .	134
4.3.2	Urban Scenario . . . . .	138
4.3.3	Tunnel Scenario . . . . .	145
4.4	Chapter Conclusions . . . . .	145
<b>5</b>	<b>Conclusion and Future Outlook</b>	<b>147</b>
<b>A</b>	<b>Data Fusion</b>	<b>151</b>



# List of Figures

1.1	Automotive applications for a combined near- and far-distance sensor network . . . . .	5
1.2	Angle estimation techniques: (a) analysis of amplitude and phase information for overlapping antenna beams (Sequential Lobing, Monopulse) ; (b) multilateration of different range measurements . . . . .	11
1.3	Range measurements of a monostatic (a) and a bistatic radar network (b)	11
1.4	Measurement principle of a pulse radar . . . . .	13
1.5	Measurement principle of a simple LFM-CW radar . . . . .	15
1.6	LFM-CW transmit signal with two up- and downchirps . . . . .	16
1.7	Range-Doppler diagram for a two-target situation (as extracted from a LFM-CW receive signal with two successive up- and downchirps) . . . . .	16
1.8	Reflection amplitude of a car depending on the sensor's aspect angle (measured at an operating frequency of 77 GHz) . . . . .	19
1.9	Reflection amplitude of a human being depending on the sensor's aspect angle (measured at an operating frequency of 77 GHz) . . . . .	19
1.10	Point-target (a) and extended object (b) as detected by a two-sensor radar network . . . . .	21
1.11	Definition of a sensor's range and azimuth angle accuracy . . . . .	22
1.12	Picture and detailed view of the high range resolution (HRR) pulse radar sensor . . . . .	23
1.13	Universal Medium Range (UMRR) Radar . . . . .	24
1.14	Block diagram of the UMRR sensors . . . . .	25
1.15	4 HRR sensor (a) and 2 UMRR (b) radar network . . . . .	27
1.16	Single UMRR sensor . . . . .	27
1.17	Standard deviations describing the radar network's position estimation accuracy at the distance $r_1$ (a) and $r_2$ (b) relative to the origin of coordinates	31

1.18	Range resolution condition of one radar sensor (a) and of all four radar sensors (b) with two tables summarizing the possibility of resolving the considered two targets for each radar sensor . . . . .	32
1.19	The 4 HRR radar network's position estimation error due to the single-sensor measurement accuracies of $\sigma_r^{\text{HRR}} = 0.03\text{m}$ (a). Figures (b), (c), and (d) show the resulting radar network's position estimation accuracies $\sigma_r$ , $\sigma_\phi$ , and $\sigma_{tan}$ , respectively. . . . .	35
1.20	Complete single-sensor resolution conditions for the 4 HRR pulse sensor network (a) and the position dependent extension $D_{tan}$ of the rhomb-like area where the resolution of two targets is impossible (b) . . . . .	35
1.21	The 2 UMRR network's position estimation error due to the single-sensor measurement accuracies of $\sigma_r^{\text{UMRR}} = 0.12\text{m}$ and $\sigma_\phi^{\text{UMRR}} = 1^\circ$ (a). Figures (b), (c), and (d) show the resulting radar network's position estimation accuracies $\sigma_r$ , $\sigma_\phi$ , and $\sigma_{tan}$ , respectively. . . . .	36
1.22	Complete single-sensor resolution conditions for the 2 UMRR sensor network	36
1.23	1 UMRR sensor's position estimation error assuming measurement accuracies of $\sigma_r^{\text{UMRR}} = 0.12\text{m}$ and $\sigma_\phi^{\text{UMRR}} = 1^\circ$ (a). Figures (b), (c), and (d) show the position estimation accuracies $\sigma_r$ , $\sigma_\phi$ , and $\sigma_{tan}$ , respectively. . .	39
1.24	Single UMRR sensor resolution condition . . . . .	39
1.25	Example of a typical road traffic scenario . . . . .	42
1.26	General postprocessing structure for an automotive multi-sensors network	42
2.1	Geometrical calculation of an object's position by a trilateration (a) and a triangulation (b) . . . . .	48
2.2	Relation between the polar sensor and the global cartesian coordinate system	51
2.3	Example of the square error function $f_{\text{error}}(\mathbf{t}, \mathbf{m})$ for a target at the position $[t_x, t_y]^T = [12, -2]^T$ m of the 4 HRR pulse radar network (a) and the 2 UMRR network (b) . . . . .	55
2.4	Number of iterations until convergence occurs for different start values utilized in a 4 HRR pulse radar network (a), (c), (e) and a 2 UMRR sensor network (b), (d), (f). . . . .	57
2.5	Available information about the host vehicle's kinematic state ( yaw-rate $m_\omega^{\text{Host}}(k)$ and velocity $m_v^{\text{Host}}(k)$ ) . . . . .	60
2.6	Implementation of the Kalman filter algorithm . . . . .	65
2.7	Example of the principle Kalman filter estimation steps from the initialization $\hat{\mathbf{t}}(0 0)$ , $P(0 0)$ to the estimate $\hat{\mathbf{t}}(3 3)$ , $P(3 3)$ at cycle 3 . . . . .	68

2.8	The discrete sigma points $\mathcal{X}_i$ are capturing the mean and covariance of the random variable $x$ . After the nonlinear transformation $\mathcal{Y}_i = g(\mathcal{X}_i)$ , the mean and covariance of $y$ can be estimated from sigma points $\mathcal{Y}_i$ . . . . .	75
2.9	Tracking test scenario: The red car is turning into the lane of the host car and is accelerating (a). Then the host car is also accelerating (b) until it is getting close to the red car (c). After that, it is decelerating until it stops and the red car is turning to one's left (d). . . . .	79
2.10	Trajectory of the tracking test scenario with the raw measurement data of the 2 UMR sensors (a) and the processed data at the output of the tracking filter (b) . . . . .	81
2.11	Standard deviations of the raw measurement data (upper curve) and the LKF results (lower curve) achieved by directly feeding the measurements into the filter . . . . .	82
2.12	Standard deviations of the raw measurement data (upper curve) and the tracking results of the iterated least squares estimator combined with a LKF (lower curve) . . . . .	83
2.13	Standard deviations of the raw measurement data (upper curve) and the tracking results of the sequential EKF (lower curve) . . . . .	84
2.14	Standard deviations of the raw measurement data (upper curve) and the tracking results of the UKF (lower curve) . . . . .	85
2.15	Example of a decoupled Kalman filter by choosing the state vector in polar coordinates . . . . .	87
2.16	Standard deviations of the raw measurement data (upper curve) and the tracking results of postprocessing system no. 1 with a decoupled LKF (lower curve) . . . . .	89
2.17	Monitored statistical square distance in the example scenario of section 2.2.6	90
2.18	Standard deviations of the raw measurement data (upper curve) and the tracking results of the maneuver adaptive LKF (lower curve) achieved by directly feeding the measurements into the filter . . . . .	91
3.1	Postprocessing overview . . . . .	94
3.2	Sensor-level postprocessing structure . . . . .	95
3.3	Simple classification of moving and non-moving targets . . . . .	96
3.4	Road traffic scenario as seen by a video camera . . . . .	99
3.5	Road traffic scenario as seen by the left (a), the right (b), and both radar sensors (c) . . . . .	99
3.6	Road traffic scenario as interpreted after the simple target identification step by the left (a), the right (b), and both radar sensors (c) . . . . .	100

---

3.7	Clustering of multiple reflections and derivation of the object size . . . . .	100
3.8	Sensor-level data association overview . . . . .	103
3.9	Example of two objects driving parallel and turning to one's left . . . . .	108
3.10	Inverse of the extended normalized square distance for the different measurement-track combinations . . . . .	109
3.11	Approximated probability of the different measurement-track combinations	109
3.12	Data association traffic scenario example . . . . .	111
3.13	Traffic scenario example of figure 3.12 for the global nearest neighbor (b) and the all neighbors data association approach (c) . . . . .	111
3.14	Sensor-level tracking overview . . . . .	113
3.15	Track evaluation by number of detections in consecutive measurement cycles	117
3.16	Track evaluation with track quality monitoring . . . . .	117
3.17	Central-level postprocessing structure . . . . .	118
3.18	Video camera snapshot of the real road traffic example . . . . .	122
3.19	Moving target detections of both UMRR sensors (a) and the appropriate cluster at the end of the postprocessing chain (b) . . . . .	122
3.20	Central-level data association overview . . . . .	123
3.21	Central-level tracking overview . . . . .	124
4.1	Testcar equipped with 2 UMRR sensors in the front bumper . . . . .	130
4.2	Detailed view of one UMRR sensor in the front bumper . . . . .	130
4.3	Testcar radar network structure . . . . .	131
4.4	Turn rate sensor . . . . .	132
4.5	Signal processing modules in the graphical user interface of RASI . . . . .	133
4.6	Data viewer tools in RASI . . . . .	133
4.7	Typical ACC traffic situation . . . . .	135
4.8	Target traces of the typical ACC traffic situation in figure 4.7 . . . . .	135
4.9	Highway situation with medium dense traffic . . . . .	137
4.10	Target traces of the highway situation with medium dense traffic . . . . .	137
4.11	Urban traffic scenario with one passenger car overtaking another one . . . . .	138
4.12	Central-level target traces of the urban traffic scenario in figure 4.11 . . . . .	139
4.13	Sensor-level target traces of sensor no.1 and 2 of the detected moving objects	139

---

4.14	Cut-out situation in an urban traffic environment . . . . .	140
4.15	Target traces of the cut-out situation in an urban traffic environment . .	140
4.16	A second cut-out situation in an urban traffic environment . . . . .	141
4.17	Target traces of the second cut-out situation in an urban traffic environment	141
4.18	Cut-in situation in an urban traffic environment . . . . .	142
4.19	Target traces of the cut-in situation in an urban traffic environment . . .	142
4.20	Tunnel scenario with dense traffic moving at almost the same velocity . .	143
4.21	Target traces of the tunnel scenario with dense traffic . . . . .	143
4.22	Tunnel scenario with dense traffic moving at different velocities . . . . .	144
4.23	Target traces of the tunnel scenario with dense traffic moving at different velocities . . . . .	144



# List of Tables

1.1	Different sensor types and their suitability for automotive applications . . .	2
1.2	Suggested requirements for different automotive applications . . . . .	4
1.3	Typical values for the radar cross section [Roh01] . . . . .	18
1.4	Sensor Properties . . . . .	22
1.5	Comparison of two 24 GHz Radar Networks . . . . .	26
1.6	Radar network performances ( $p_{detect} = 1$ ) . . . . .	41
2.1	Number of iterations until convergence occurs and estimation percentage for the condition given in eq.(2.51) . . . . .	58
2.2	5 different postprocessing systems in a 2 UMR sensor network for single- target situations . . . . .	78
4.1	Information available on the CAN-bus . . . . .	132



# Chapter 1

## Introduction

The first invention of RADAR (radio detection and ranging) can be dated around the beginning of the twentieth century. In fact, several researchers in different countries announced the birth of radar at similar times. All these works were based on the equations of the british physicist James Clerk Maxwell describing the behavior of electromagnetic waves in 1864. Inherent in Maxwell's equations are the laws of radio-wave reflection, which were first demonstrated in 1886 in experiments by the German physicist Heinrich Hertz. Some years later, the German engineer Chistian Hülsmeier proposed the use of radio echoes in a detecting device designed to avoid collisions in marine navigation on the river Rhine. But until the 1930s no further interest in the radar technology was aroused. The threat of war along with growing technological advances all over the world prompted many countries to intensify their research in the field of radar.

Nowadays many peaceful radar applications can be found in our everyday life such as radar for weather analysis and prediction, for the discovery and rescue of avalanche victims, or just as the vandal-resistant automatic toilet flush. The first research activities in the field of automotive radar systems for road traffic situations were started in the 1970s with the aim of increasing the comfort and safety of road users in times of permanently growing traffic. But at that time, the geometrical size of a single radar sensor exceeded the size of a practical series product. The cost and the performance of such a system did also not meet the necessary requirements for comfort and safety applications in a passenger car. Since then, the fast development in microwave technology and microelectronic has led to an improvement concerning all these requirements. Today's radar front-ends are small enough to be integrated into a car, the necessary signal processing hardware is powerful enough to handle different algorithms and the reduced cost make it possible to develop new active comfort and safety applications.

This thesis will present a postprocessing architecture for an automotive radar network that has been implemented in a test vehicle and evaluated in real road traffic scenarios. Before this, different multi-sensor radar network concepts will be investigated and their characteristics in estimating the road users' kinematic states as well as interpreting typical road traffic scenarios will be discussed. It will be explained how different postprocessing algorithms can enhance the radar network's performance and what typical challenges

	Ultrasonic	Lidar	Video	Radar
Max. Range	--	+	<i>o</i>	++
Range Accuracy and Resolution	+	++	<i>o</i>	++
Angular Accuracy and Resolution	- multiple sensors needed	++ scanning principle	++	Accuracy:+ Resolution:- <sup>1</sup>
Range Rate/ Velocity	--	<i>o</i>	-	++
Extended Objects	--	++	++	+
Obscuration (Rain, Snow, Fog)	-	<i>o</i>	-	++
Size / Packaging	visible	visible	visible	invisible
Cost	++	+	+	<i>o</i>

Table 1.1: Different sensor types and their suitability for automotive applications

have to be faced in the field of automotive radar processing.

This chapter will first present an overview of the different automotive radar applications and will explain the appropriate requirements of these systems. To develop an understanding for the system properties and especially for the postprocessing task in an automotive radar network, the basic radar measurement principles will be outlined. After this, two high-performance radar sensors and their measurement properties will be presented. With these radar front-ends, three different multi-sensor radar network concepts will be established and compared in respect of the different estimation criteria for automotive applications. Finally, the basic steps of a postprocessing architecture will be outlined.

## 1.1 Automotive Radar Applications

The safety of road users in times of permanently growing traffic has become a major topic for automotive companies in recent years. Many systems such as airbags, ABS (Antilock Braking System) with EBD (Electronic Brake Distribution) and brake assistant, ESP (Electronic Stability Program), SRS (Supplemental Restraint System), or the preventive belt pretensioner have been invented and the properties of car frames and bodies concerning the energy absorption have been optimized to protect the passengers inside a car. Most of these systems reduce the risk of getting injured if an accident happens or even the probability of losing the control over the car, but the ultimate solution to achieve the reduction of accidents are active safety systems that prevent cars from smashing into

<sup>1</sup>Most of today's radar sensors are not able to resolve targets in the azimuth angular direction. To achieve such a resolution, a mechanically or electronically scanning antenna is required. Such radar sensors are under development, but are not available yet.

each other. The first simple collision avoidance features are already on the road as pricey adaptive cruise control options on a small group of luxury cars, but in the near future this technology will become widely available.

In the field of automotive comfort- and safety-applications, different sensors such as ultrasonic, infrared, lidar (light detection and ranging), or radar sensors as well as video cameras are currently under investigation. Table 1.1 shows the different criteria that are important to car manufacturers for their new applications and the appropriate suitability of the different sensor types. These requirements can be grouped into three main categories, namely sensor performance, sensor size/visibility, and cost. To be economically successful in all automotive segments, the cost of a single sensor should be around ten dollar in mass-production, and since design plays an important role in selling cars, the sensors must also be "invisible" in the car body. This means that the sensors should be small in size and that they can be hidden behind the front bumper, for example. No "holes" should be visible as it is the case for today's ultrasonic park distance control sensors. The requirements concerning the performance are depending strongly on the targeted application. Generally, it is advantageous to have a large set of measurement information available. For automotive applications, the measurements of today's sensors comprise

- range
- azimuth angle
- velocity
- object size

The evaluation of a sensor's measurement performance can be manifold. The characteristic values that are generally utilized to describe the sensor performance are

- the accuracy,
- the resolution,
- the maximum unambiguous range.

The measurement accuracy is defined as the standard deviation of the measurement relative to its expectation value. In the ideal measurement case, the expectation is the error-free measurement, i.e. the ideal measurement corresponding to the real target position. The resolution of a sensor is defined as the least necessary condition that leads to a separate detection of two neighboring objects of equal size with equal reflection properties, i.e. with the same radar cross section (see section 1.2.3). It depends strongly on the radar sensor's measurement principle. If no scanning receive antenna is utilized, then no resolution in the azimuth angular direction can be achieved. The range resolution of a sensor, i.e. the least necessary radial distance between two neighboring objects, depends directly on the bandwidth of the utilized transmit waveform (see section 1.2).

	ACC	Parking Aid	Blind Spot Surveillance	Lane Change Assistant
Update Rate: $T_{Cycle}$	$\geq 20$ Hz $\leq 50$ ms	$\geq 10$ Hz $\leq 100$ ms	$\geq 10$ Hz $\leq 100$ ms	$\geq 20$ Hz $\leq 50$ ms
Range:				
Min. Range	1.0 m	0.0 m	0.5 m	0.5 m
Max. Range	200.0 m	8.0 m	5.0 m	150.0 m
Range Accuracy	0.25 m	0.05 m	1.0 m	1.0 m
Range Resolution	1.0 m	0.1 m	0.5 m	0.5 m
Azimuth Angle				
Angular field of view of a single sensor	$\pm 15^\circ$	$\pm 60^\circ$	$\pm 60^\circ$	$\pm 60^\circ$
Angle Accuracy	$2^\circ$	$2^\circ$	–	$2^\circ$
Angular Resolution	$5^\circ$	$5^\circ$	–	$5^\circ$
Velocity				
Min. Velocity	–360 km/h	–50 km/h	–	–360 km/h
Max. Velocity	180 km/h	50 km/h	–	180 km/h
Velocity Accuracy	0.45 km/h	0.1 km/h	–	0.45 km/h
Velocity Resolution	1.8 km/h	1.0 km/h	–	1.8 km/h

	Pre-Crash Warning	Collision Avoidance	Stop & Go (incl. ACC)
Update Rate: $T_{Cycle}$	$\geq 100$ Hz $\leq 10$ ms	$\geq 100$ Hz $\leq 10$ ms	$\geq 50$ Hz $\leq 20$ ms
Range:			
Min. Range	0.3 m	1.0 m	0.3 m
Max. Range	30.0 m	200.0 m	150.0 m
Range Accuracy	0.05 m	0.25 m	0.25 m
Range Resolution	0.25 m	0.25 m	0.25 m
Azimuth Angle			
Angular field of view of a single sensor	$\pm 60^\circ$	$\pm 15^\circ$	$\pm 60^\circ$
Angle Accuracy	$2^\circ$	$2^\circ$	$2^\circ$
Angular Resolution	$5^\circ$	$5^\circ$	$5^\circ$
Velocity			
Min. Velocity	–250 km/h	–360 km/h	–360 km/h
Max. Velocity	100 km/h	180 km/h	180 km/h
Velocity Accuracy	1.0 km/h	0.45 km/h	0.45 km/h
Velocity Resolution	3.0 km/h	1.8 km/h	1.8 km/h

Table 1.2: Suggested requirements for different automotive applications

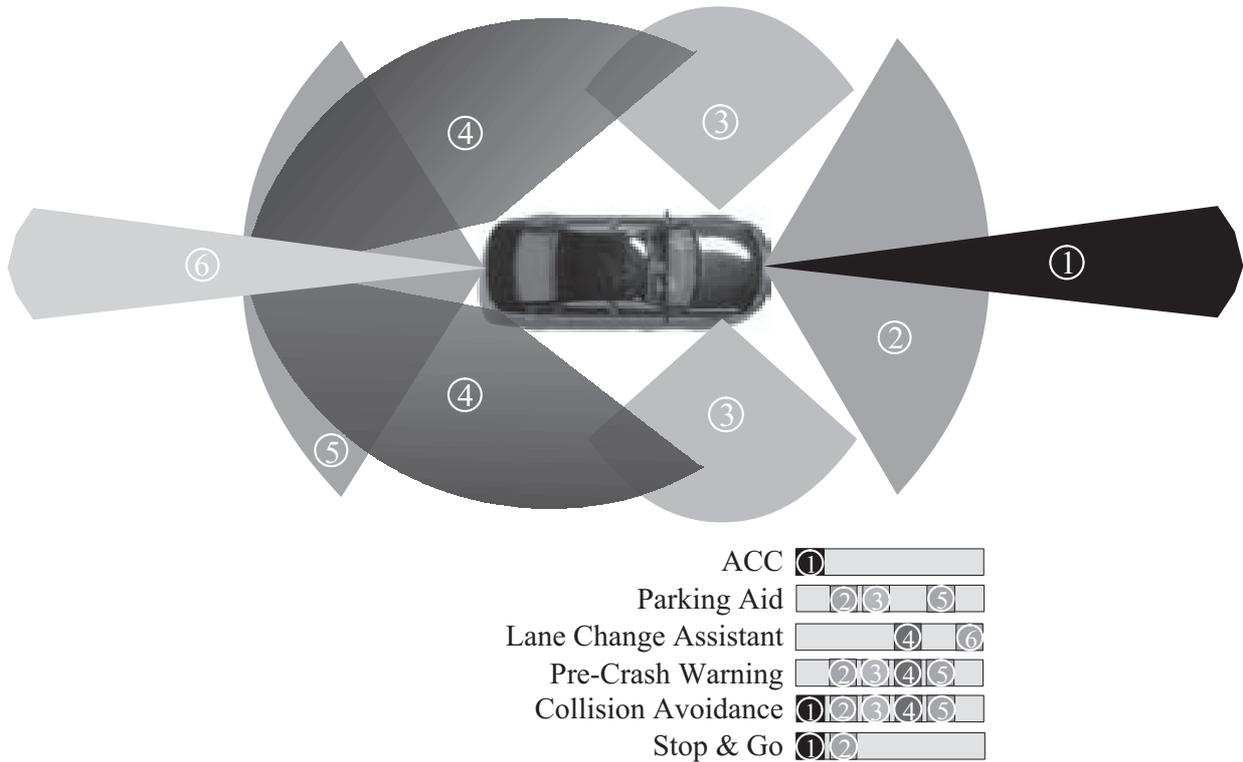


Figure 1.1: Automotive applications for a combined near- and far-distance sensor network

The maximum unambiguous range of the measurements is also an important criterion. Today’s radar sensors can be grouped into three categories referring to the maximum detectable range of an object. So-called far-distance sensor can achieve a maximum range that lies within 150 and 200 m with a small azimuth angular coverage of about 10°. Near- and medium distance sensors cover a range area up to 20-50 m and have a wider azimuth angular antenna coverage that of about 60°-120°. Finally, the sensors must be robust and work with the same high performance in all weather conditions.

As shown in table 1.1 the ultrasonic sensors are low in cost but their maximum range is too small for many wanted automotive applications. Another disadvantage of this sensor type is the visibility in the car body, e.g. the "holes in the front bumper" known from today’s park distance control systems. Nevertheless, these kind of sensors are favored for parking aid applications at the moment, because they are very low in cost and show an acceptable measurement performance for applications with very short distances. Lidar sensors and video cameras show a good range and angular measurement performance but this performance is not weather independent. This means that these sensors downgrade their measurement performance in rain or snow due to line-of-sight obstruction. In the case of lidar sensors many enhancements have been established by different sensor manufacturers, so that some of today’s lidar sensors show an acceptable bad weather performance. But the robustness and weather independence paired with the invisibility of the sensors in the car body, the compact sensor size, and a high measurement perfor-

mance are the big advantages of radar technology. Car manufacturers are investigating systems consisting of different sensor types in order to find the most versatile and low-cost combination. Figure 1.1 shows a versatile multi-sensor radar network covering the scan areas of different automotive applications. Such a complete radar network will probably not be implemented in a final product, nowadays. But since the number of driver assistance functions in a passenger car will increase in the future, this may be a possible sensor constellation in a few years. Figure 1.1 shows also a table summarizing the sensor combinations that are at least demanded by the different automotive application. Cost considerations will play a vital role in the implementation of each system. It will drive the system architectures to minimize the number of sensors on the vehicle in order to reduce component and installation cost. The performance requirements of a multi-sensor network also depend strongly on the targeted application. These application-dependent requirements are summarized in table 1.2. Concerning the functionality, the scan area, and the performance requirements, the different automotive applications can be characterized as follows:

- **ACC (Adaptive Cruise Control)**

An ACC system is a convenience application for the driver and can be considered as an enhancement of the standard cruise control. While the standard cruise control performs best in low traffic environments with a very low number of obstacles in the own driving path, the adaptive cruise control enhances its performance to more dense traffic situations. The ACC system is able to decelerate if the own vehicle is approaching another slower moving car situated in the same lane and to accelerate again if such a slower moving car is changing lanes in order to give way. The required detection area of an ACC system is mainly the own lane as well as the two neighboring lanes at a greater distance in front of the own car. The first ACC systems are on the market since 1998 and are now offered by several car manufacturers as an option for their luxury cars. Most of these systems operate with a 77 GHz far-distance radar sensor that can detect targets up to a maximum distance of at least 150 m ahead of the own car. The far-distance sensor is mounted in the front bumper, so that the scan area visualized in figure 1.1 is achieved. The update rate of an ACC system should be better than 20 Hz in order to decelerate in time if an obstacle is detected in the driving path of the own car. The first ACC systems were restricted to road traffic situations, where the own vehicle is moving faster than 80 km/h. But today's ACC systems are working properly at 30 km/h and will be enhanced in the future by additional near- or medium-distance sensors to achieve a stop & go functionality.

- **Parking Aid**

The purpose of a parking aid system is to support the driver in a parking maneuver. The general action is the movement of a vehicle from an area where the traffic flows to an area close to the traffic flow where the vehicle can be left stationary. This includes a planning of the parking maneuver by checking the size of the parking lot in comparison to the size of the own car and the own driving capability. Such a driver assistance functionality can be achieved at different levels. The basic level is to inform the driver by optical or acoustical means if an obstacle at the front or

rear gets close to the host vehicle. This can be combined with an alert signal at a certain minimum distance, for example. The driver has to check by his own if the desired parking lot is suitable for his car and the planned parking maneuver. This functionality is already achieved by today's park distance control. At the next level of a parking support system, the driver is given information about the desired parking lot and maybe a recommendation about the vehicle path (steering angle, car speed). At the final level, the complete parking maneuver is done automatically by the car. The required scan area for such a parking aid system is ideally the whole near-distance area around the car (see figure 1.1) or at least the near-distance area in front and behind the front and rear bumper, respectively. One car manufacturer presented a system that is relying basically on two side-looking sensors mounted in the vehicle's right mudguard and on sensors covering the near-distance areas in front and behind the car. The side-looking sensors are measuring the size of the parking lot while the car is driving past it. Then, the car is informing the driver if a parking maneuver is possible. If this is the case, the driver assistance system is automatically steering the car while the driver is only controlling the speed of the maneuver. Such a system requires only a low update rate because the car's movement is very slow compared to normal road traffic maneuvers. The necessary maximum range of the sensors is very small, e.g. about 15 m. But it is important that the range measurements are very accurate to enable a precise parking maneuver.

- **Blind Spot Surveillance/ Lane Change Assistant**

The purpose of these safety applications is to avoid a classical reason for accidents, a driver has overseen an obstacle being in the blind spot of his car or one that is approaching with high speed on a neighbored lane while the driver is maneuvering in the appropriate direction. Such an accident can simply be prevented if e.g. an acoustical or optical signal in the side rear mirrors informs the driver of the obstacle's presence in his car's blind spot area. The blind spot surveillance system requires a small detection area with a maximum range of 5 m at the location of the car's blind spot. Velocity measurements are not relevant for this application because it is a mere presence detection. The lane change assistant system can be considered as the logical extension of the blind spot surveillance system. At this, the driver is informed whether an obstacle will be situated on the right or left lane next to his car in the next moments. If he wants to change lanes in such a situation, a warning signal will appear. Since this application is preferred for highway-driving where high relative velocities may be measured, precise velocity measurements are necessary and the detection area of the side- and back-looking sensors must cover up to 150 m behind the own vehicle (see figure 1.1) for a lane change assistant with full functionality.

- **Pre-Crash Warning**

The general function of the safety application "pre-crash warning" is to detect an unavoidable crash with an obstacle a short time before the impact occurs. If there is a high probability of such an impact, the safety restraint systems will be provided with the appropriate information and thus can react a few milliseconds before the crash occurs. A possible reaction to the information of the radar network will be

a faster airbag deployment in a combination with some degree of car deceleration. Since a crash can occur from every direction, the pre-crash warning system requires the whole near-distance area around the car for detection (see figure 1.1). The required update rate of such a safety application is very high because collisions with other vehicles may occur at very high relative speeds. It is extremely important, that the pre-crash warning system detects an unavoidable crash as early as possible with an extreme low false alarm rate, so that the system can react properly.

- **Collision Avoidance/ Warning**

A collision warning system has to give the driver information that is indicating the need for urgent action to avoid a collision. The warning has to be provided in the advanced phases of a dangerous situation in order to draw the driver's attention to the need of performing an emergency braking, lane changing, or other collision avoidance maneuvers. If these necessary actions are performed automatically by the vehicle, the application is called collision avoidance system. Since collisions can occur from every direction, the scan area of the sensors must cover the whole near-distance area around the car. But in contradiction to the pre-crash warning system, the driver or system must be able to react quickly enough to avoid the collision. This gives a certain importance to the area where the obstacles with the highest relative velocity approach from, the area in front of the car. Thus, such a collision warning/avoidance system requires a far-distance sensor in the front bumper covering up to 200m in range. For the same reason, the update rate of such a safety system must be very high in order to detect possible collisions as early as possible.

- **Stop & Go**

The stop & go functionality is an enhancement of the adaptive cruise control enabling it to be utilized in more traffic scenarios than a simple ACC system. The main difference from today's ACC systems is that the stop & go system shall be able to bring the vehicle to a complete stop and also to accelerate it again. Because of this functionality enhancement, the stop & go system will further extend the usability of today's ACC systems to more dense traffic situations, such as city traffic and congested highway scenarios. Due to the demands of a close target detection and an early reaction in cut-in situations, the stop & go system requires additional near-distance sensors with a wide azimuthal scan area. These sensor are mounted in the front bumper as shown in figure 1.1. The update rate must be slightly higher than the one of an ACC system, because the stop & go system should work in dense traffic scenarios, where obstacles can suddenly appear in the driving path from different directions.

Because of the fact that all applications have a great impact either on the response of the driver to a warning signal or directly on the speed and maybe somewhere in the future on the steering angle of the car, they share a key requirement:

- high reliability

The newest field in today's radar signal processing research is the classification of road users by distinguishing between pedestrians, cyclists, passenger cars, and large trucks. The first results in this new automotive radar signal processing area have been promising and the above described applications will be enhanced by using more information about the detected objects in the future.

In the following, this thesis will concentrate on radar networks consisting of near- or medium-distance sensors that are needed for the parking aid, blind spot surveillance/lane change assistant, pre-crash warning, collision avoidance, and stop & go systems.

## 1.2 Radar Measurement Principles

Before introducing the considered radar sensors and the possible radar network concepts, this section will give a brief introduction to radar transmit signals and the underlying measurement principles. As already mentioned in the previous section, the most common measurements available from today's automotive radar sensors are

- range
- azimuth angle
- velocity
- object size

The principle of a radar bases on the reflection of a transmit signal at an object. A radar sensor is determining the range between itself and the detected object by measuring the time delay  $\tau$  between the transmit and the receive signal, i.e. the duration that it takes the electromagnetic wave to travel this two-way path with the speed of light  $c$  [Roh01]. The corresponding **range**  $r$  can be calculated from the time delay  $\tau$

$$r = \frac{c \cdot \tau}{2} \quad (1.1)$$

The **radial velocity** measurement of an object bases on the Doppler frequency shift  $f_{Doppler}$  between the transmit and the receive signal. This shift is given by

$$f_{Doppler} = \frac{-2 \cdot v_r}{\lambda} \quad (1.2)$$

where  $\lambda$  is the wavelength and  $v_r$  is the radial velocity of the object relative to the radar sensor.

To determine the **azimuth angle of the object's direction**, different techniques can be applied. The oldest one is the physical moving of an antenna beam through the desired azimuth angle range. A more modern approach is an electronic beam steering with array

antennas by differentially shifting the phases of the transmit signal emitted by the individual radiators. Both antenna steering techniques can be found in prototypes of today's automotive far-distance radar sensors and offer the big advantage of azimuth angular resolution of multiple targets. But since these methods are too complex or expensive for the implementation in near- or medium-distance radar sensors at the moment, simpler techniques are utilized to determine the azimuth angular information:

- **Sequential Lobing**

The sequential lobing angle estimation technique is often applied by far-distance radar sensors for ACC systems. At this, the radar switches sequentially between several overlapping receive antenna beams. Then, the difference between the measured receive signals is utilized to calculate an angular error signal and to extract the azimuth angular information from it.

- **Monopulse Technique**

Another angle estimation technique that is historically derived from sequential lobing is the monopulse technique [Mah98], [Sko80]. The difference to the sequential lobing is that all antenna beams are generated simultaneously (by the name-giving single pulse  $\Rightarrow$  monopulse). The idea is to use different receive paths at the same time and to extract the angular information from the comparison of the different receive signals [Klo02]. These receive signals can be compared in amplitude and phase, so that the angular information can be determined by the appropriately calculated error signals.

- **Trilateration / Multilateration**

The estimation of the azimuth angular direction of an object can also be achieved by using at least two high accuracy range-measuring sensors located at different reference points. For each measured range, the possible positions of the detected object can be found on a circular arc with the appropriate sensor location at the circle center. The position of the object is determined by intersecting all circular arcs corresponding to the range measurements of the different sensors. It is a simple geometric calculation in the case of two range measurements from two different sensors. This angle estimation technique is called trilateration and will be described in detail in the next chapter where different kinematic state estimation methods will be outlined.

The term trilateration is often mixed up with the term multilateration. Both are defined ambiguously in the literature, so that this thesis will follow the definition of [Föl01]. There, the trilateration is the determination of the object position by intersecting circular arcs as described before. The multilateration is an extension of this position estimation utilizing a radar network that consists of synchronized sensors. Every sensor knows the state of the other radar network sensors at every point in time, so that the transmit signal of one sensor can also be evaluated by the other ones. Thus, additional range measurements are available for the position estimation. Figure 1.3 shows that beside the monostatic range measurements  $r_1$  and  $r_2$ , two bistatic measurements of the summed up distance  $r_1 + r_2$  are available, because sensor 2 can evaluate the transmit signal of sensor 1 and vice versa.

The advantage of such a system is not only the greater number of distance measurements from different aspect angles, it is also the simultaneousness of the taken measurements [RHLS02].

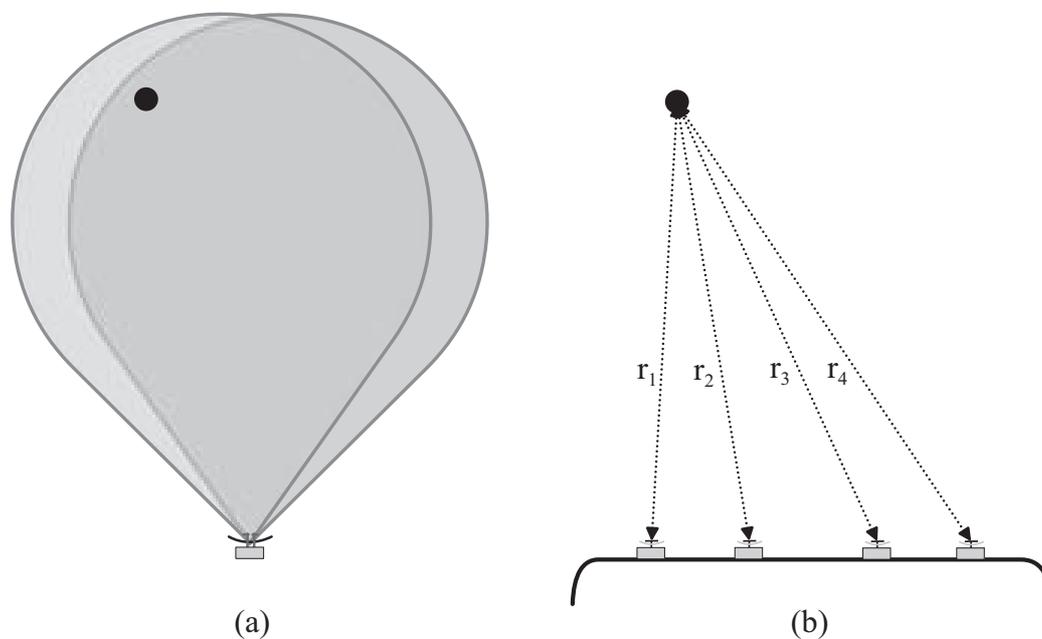


Figure 1.2: Angle estimation techniques: (a) analysis of amplitude and phase information for overlapping antenna beams (Sequential Lobing, Monopulse) ; (b) multilateration of different range measurements

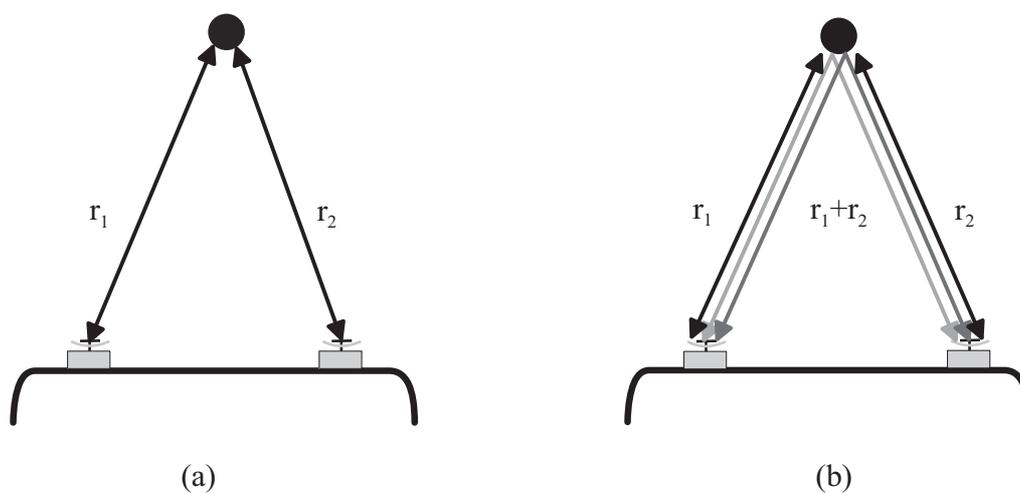


Figure 1.3: Range measurements of a monostatic (a) and a bistatic radar network (b)

So far, the basic radar measurement principles have been explained on the basis of a transmit signal that is an undefined sine-based electromagnetic waveform. But this transmit signal waveform has to be designed carefully in order to gain the wanted precise measurement information. Two basic, commonly implemented waveform principles will be explained in the following section as a representation for the variety of possible transmit signals. These two principles are implemented in the two automotive radar sensors that will be considered for a radar network later in this chapter.

### 1.2.1 Pulse Radar

The oldest transmit signal in radar history is the pulse radar waveform. Such a pulse is generated by amplitude modulating a sine-wave with an appropriate pulse shape. As shown in figure 1.4, the pulse is transmitted by the radar sensor antenna and the reflected signal is received after a time delay  $\tau$ . This time delay  $\tau$  is utilized to calculate the range  $r$  of an object as it is given in equation (1.1). This method is called pulse delay ranging. The measurement properties and thus the design parameters of such a pulse radar transmit signal are:

- **Pulse Width**

The pulse width has an impact on the range accuracy and resolution of a radar sensor. The range resolution  $\Delta R$  is defined as the minimum radial distance between two neighboring objects of equal reflection properties that leads to a separate detection of both. If the radial distance between the two targets is smaller than  $\Delta R$ , then the two reflected pulses superpose to a single pulse that cannot be resolved at the receiver. The range resolution  $\Delta R$  can be calculated as

$$\Delta R = \frac{c}{2} \cdot T_{pulse} \quad (1.3)$$

The accuracy  $\sigma_r$  of the range measurement  $r$  is influenced proportionally to the pulse width  $T_{pulse}$

$$\sigma_r \sim \frac{c}{2} \cdot T_{pulse} \quad (1.4)$$

For a high signal-to-noise ratio, the accuracy is about 20 – 30 % of the resolution [Roh01].

- **Pulse Repetition Frequency (PRF)**

The pulse repetition frequency  $PRF$  is given by

$$PRF = \frac{1}{T_{PulseRepetition}} \quad (1.5)$$

and defines the unambiguous range  $R_{max}$  of the radar sensor

$$R_{max} = \frac{c}{2} \cdot T_{PulseRepetition} \quad (1.6)$$

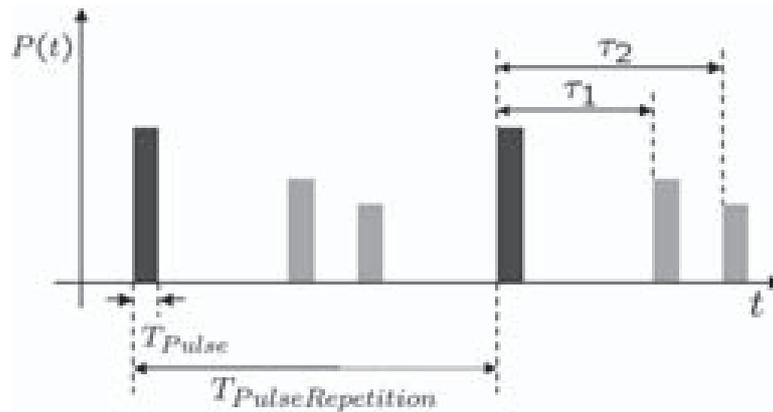


Figure 1.4: Measurement principle of a pulse radar

Objects outside this measurement range  $r > R_{max}$  can be detected, but the measurements are ambiguous. The measured range  $r_m > R_{max}$  will be projected onto the unambiguous measurement range  $r < R_{max}$  in such a case. This projection can be expressed by a modulo operation

$$r_m = r_{real} \bmod R_{max} \quad (1.7)$$

- **Pulse-Doppler Processing**

The above described basic pulse radar waveform is only measuring the delay of the received pulse echoes, which means that simultaneously received returns from different objects cannot be resolved. To achieve a separation of targets at almost the same range but with different radial velocities, a pulse-Doppler processing can be applied. At this, a train of pulses is transmitted and an analysis of the phase shifts in the reflected signal enables the determination of the different objects' velocities (pulse-doppler principle, see [Roh01], [Lud98]). Such a velocity determination in a single range gate containing multiple targets is usually achieved by a spectrum analysis with a fast Fourier transform (FFT). A high precision range-measuring pulse radar has a large number of range gates and thus a large number of FFT calculations are necessary.

Several further techniques, such as the use of staggered PRFs or the pulse compression technique, have been developed to enhance the basic pulse delay ranging principle. Please consider [Lud98], [Roh01], or [Wie03] for further details on this topic.

## 1.2.2 Linear Frequency Modulated Continuous Wave Radar (LFM-CW)

In contradiction to the pulse radar, the continuous wave (CW) radars have no switch between the transmit and receive mode, which means that they are transmitting and receiving simultaneously. Due to the non-existing switching time between these modes, the LFM-CW waveform principle has no physical restriction concerning the minimum detectable range. Many different CW transmit waveforms have been investigated in the last years (see [Mei01]) and are implemented in today's automotive radar sensors, e.g.

- the frequency shift keying (FSK)-CW transmit waveform,
- the linear frequency modulated (LFM)-CW transmit waveform,
- the intertwined LFM-CW waveform, and
- a waveform consisting of short steep chirps.

This thesis will briefly explain one often implemented class of CW-based measurement principles, the linear frequency modulated (LFM-CW) principle. This transmit waveform is implemented in one of the later presented automotive radar sensors that is utilized to establish a multi-sensor radar network. For further details on the other CW measurement principles, consider [Mei01].

The LFM-CW radar transmit signal is a linear sine sweep between the operating frequency  $f_{carrier}$  and  $f_{carrier} + f_{hub}$  as it is shown in figure 1.5 (a). The signal is reflected at the object and received with a time delay  $\tau$ . Since these radars are transmitting and receiving simultaneously, this causes a frequency difference  $df$  between the instantaneous frequencies of the transmit signal  $f_{trans}(t)$  and the receive signal  $f_{rec}(t)$  at time  $t$

$$df(t) = f_{trans}(t) - f_{rec}(t) \quad (1.8)$$

The received signal is mixed down to the baseband, so that the frequency difference  $df$  can be detected by performing a Fast Fourier Transform (FFT). If an object is not moving, then the range can be calculated directly from the detected difference frequency. But if this is not the case, the receive signal of figure 1.5 (a) is additionally frequency shifted by the Doppler-frequency  $f_{Doppler}$  (see eq.(1.2)) as it is shown in figure 1.5 (b). This superposition of range and velocity information in the measured frequency difference  $df$  is described by the following equation

$$df = f_{Doppler} - f_{\tau} \quad (1.9)$$

With  $f_{Doppler}$  as defined in eq.(1.2) and  $f_{\tau}$  given by

$$f_{\tau} = \frac{2}{c} \cdot \frac{f_{hub}}{T_{chirp}} \cdot r \quad (1.10)$$

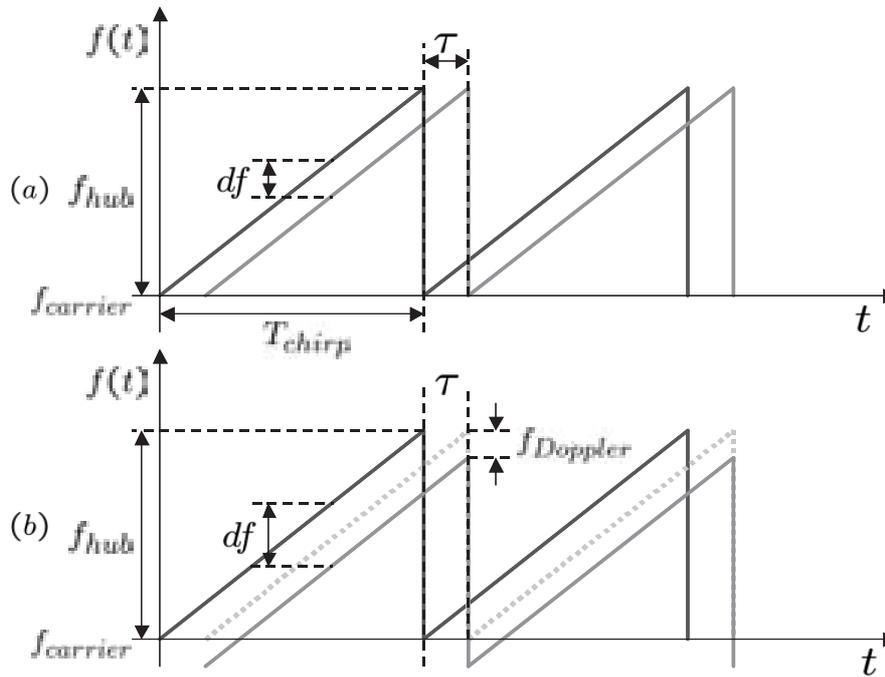


Figure 1.5: Measurement principle of a simple LFM-CW radar

the possible range and velocity values of the detected target are determined by the following linear equation

$$r = \frac{c \cdot T_{chirp}}{2 \cdot f_{hub}} \cdot \left( \frac{-2 \cdot v_r}{\lambda} - df \right) \quad (1.11)$$

and can be visualized as a straight line in a range-Doppler diagram. The ambiguity of the range and velocity information extracted from a single chirp can be solved by transmitting consecutive chirps with inverse gradients, so-called up- and down-chirps, as they are shown in figure 1.6. In a single-target situation, two chirps are sufficient to resolve the two linear equations and to determine the object's range and velocity. But in a multi-target situation, the resulting system of linear equations is under-determined. To improve the possibility of resolving these ambiguities, a four-chirp waveform is normally utilized (see figure 1.6). The third and the fourth chirp must have a different gradient as the first and second one in order to extend the system of linear equations. Each linear equation belongs to one chirp and can be visualized as a line in the range-Doppler diagram. Figure 1.7 shows these lines for a two-target situation measured with a transmit signal consisting of four chirps. The correct range and Doppler-frequency values of the two targets can be determined by finding the intersections of the four appropriate lines corresponding to the appropriate linear equations. In a multi-target situation with more than two targets, such a four-chirp transmit signal waveform will always lead to a great number of intersections, which have to be evaluated carefully. Although the number ghost-targets can be minimized by applying special intersection techniques (see [Ahr99]), they cannot be completely eliminated. This task has to be solved by different processing steps later in the signal processing chain.

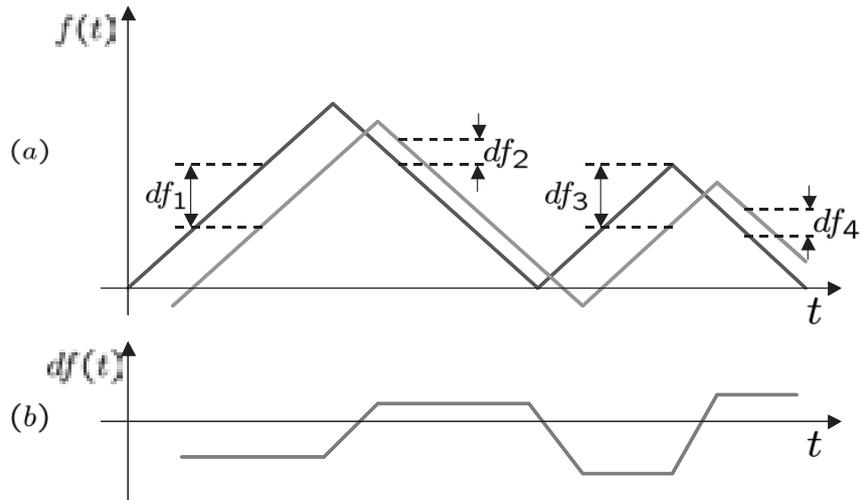


Figure 1.6: LFM-CW transmit signal with two up- and downchirps

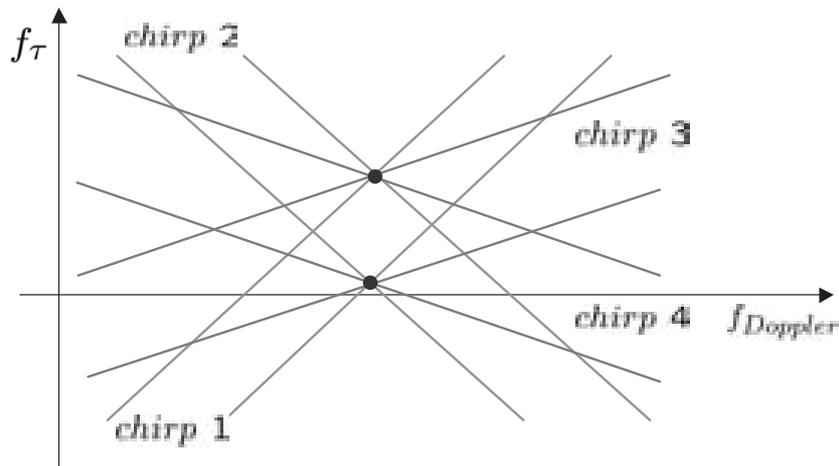


Figure 1.7: Range-Doppler diagram for a two-target situation (as extracted from a LFM-CW receive signal with two successive up- and downchirps)

The design parameters of an LFM-CW waveform are

- the chirp duration  $T_{chirp}$
- and the frequency bandwidth  $f_{hub}$

Both have a direct impact on the accuracy and the resolution achieved by the peak frequency estimation in the baseband receive signal spectrum. Thus, the question of range or velocity accuracy and resolution is answered by the consideration of the achievable frequency estimation accuracy and resolution. This frequency estimation accuracy and resolution depends on the applied frequency estimation algorithm. In the following, an

implementation of the fast fourier transform (FFT) is assumed for spectral estimation. Compared to parametric spectral estimation methods, the FFT processing is independent of the number of targets present in the spectrum and it has a determined low computation complexity.

The smallest resolvable difference between the frequencies  $df_1$  and  $df_2$  belonging to two peaks with equal magnitude in an FFT spectrum is at least 2 FFT-bins. If a non-rectangular window with a better side-lobe suppression is applied before the FFT calculation, then this separation by 2 FFT-bins is not sufficient to achieve a peak frequency resolution. Thus, this frequency separation of 2 FFT-bins can be considered as an upper bound for the frequency resolution of a system that is utilizing FFT-based spectrum estimation. With the FFT bin-width of an N-point FFT given by

$$\Delta f_{bin} = \frac{f_s}{N} \quad (1.12)$$

and the chirp duration  $T_{chirp}$  being the inverse of this bin-width, the upper bound for the frequency resolution is determined by

$$\Delta f = |df_1 - df_2| > \frac{1}{T_{chirp}} \quad (1.13)$$

According to eq.(1.9) the detected frequencies  $df_1$  and  $df_2$  are superpositions of the range and velocity information. Thus, it is difficult to generally speak of range and velocity resolution in the case of an LFM-CW radar, because both must be derived from the frequency resolution  $\Delta f$ . The assumption that the velocities of the two considered objects are similar ( $v_{r_1} = v_{r_2}$ ), leads to an often used value for the range resolution of an LFM-CW radar

$$\Delta R = \frac{c}{2} \cdot \frac{1}{f_{hub}} \quad (1.14)$$

This range resolution will only be determined by the bandwidth  $f_{hub}$  of the transmit signal. The same approach can also be applied to derive the velocity resolution. If two objects with the same reflection properties are positioned at the same range ( $r_1 = r_2$ ), then the resolution of the radial velocity is given by

$$\Delta V_r = \frac{\lambda}{2} \cdot \frac{1}{T_{chirp}} \quad (1.15)$$

The accuracy of a spectral peak frequency determination in an FFT spectrum depends also on the measurement time  $T_{chirp}$  of a single chirp. Thus, the achievable frequency accuracy is proportional to the achievable frequency resolution. At a high S/N value, it lies within of 10-20 % of the FFT-binwidth.

object	$RCS / \text{m}^2$
insect	$10^{-5}$
bird	$10^{-2}$
military aircraft	10
human	1
car	10

Table 1.3: Typical values for the radar cross section [Roh01]

### 1.2.3 Object Reflection Properties

The radar signal reflection behavior of different road users is very important and must be taken into account in the design of the radar network postprocessing architecture outlined in the next chapters. Thus, it is necessary to introduce some basic radar principles related to the reflection behavior. Assuming a lossless propagation medium, the transmitted power  $P_T$  of a radar sensor produces a power density  $S_T$  at the distance  $r$

$$S_T = \frac{P_T \cdot G_{ant}}{4\pi r^2} \quad (1.16)$$

where  $G_{ant}$  is the transmitting antenna gain relative to an omnidirectional antenna. If an object is located in the antenna beam, it is reflecting the radiated energy proportionally to its size. This reflection property is summarized in one object-specific, scalar parameter, the so-called radar cross section (RCS). This radar cross section is the equivalent value of an ideally, omnidirectional reflecting object that produces the same receive power at the radar sensor receiver as the real object. Thus, the reflected power density  $S_R$  at the receive antenna's position can be written as

$$S_R = \frac{P_T \cdot G_{ant} \cdot RCS}{(4\pi r^2)^2} \quad (1.17)$$

The receiving antenna has an effective aperture  $A_{ant}$  related to the antenna gain  $G_{ant}$  by

$$\frac{A_{ant}}{G_{ant}} = \frac{\lambda^2}{4\pi} \quad (1.18)$$

Inserting (1.18) in eq.(1.17) leads to the received power known as the radar equation

$$P_R = \frac{P_T \cdot G_{ant}^2 \cdot \lambda^2 \cdot RCS}{(4\pi)^3 r^4} \quad (1.19)$$

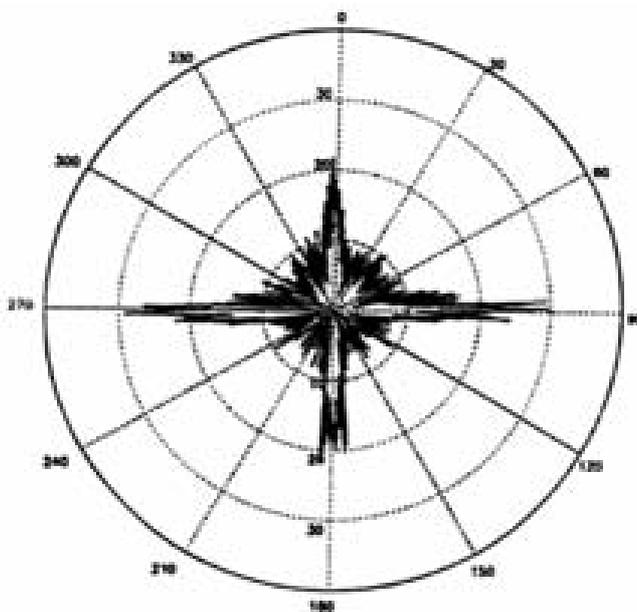


Figure 1.8: Reflection amplitude of a car depending on the sensor's aspect angle (measured at an operating frequency of 77 GHz)

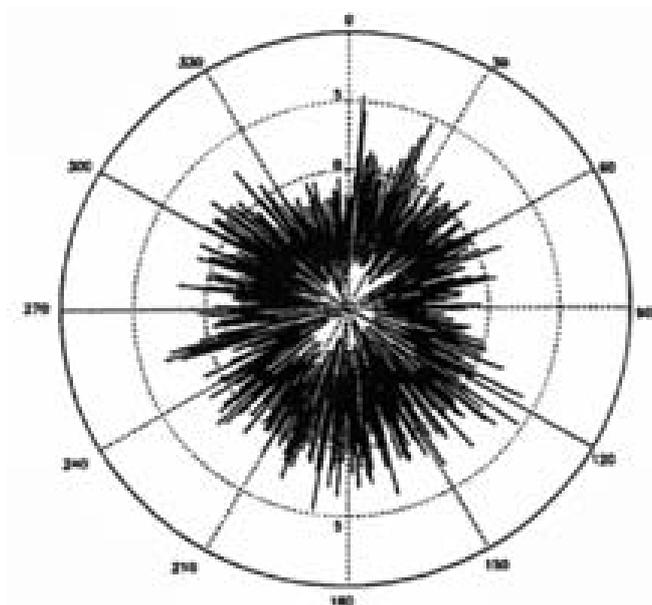


Figure 1.9: Reflection amplitude of a human being depending on the sensor's aspect angle (measured at an operating frequency of 77 GHz)

The *radar cross section* of a complex object is depending on several factors:

- object extension
- material
- shape
- aspect angle of the measuring radar sensor
- the radar's operating frequency

Typical RCS values of some objects can be found in table 1.3, [Roh01]. The figures 1.8 and 1.9 show the RCS of a car and a human being depending on the aspect angle of the measuring radar sensor. One important difference between both RCS characteristics is the aspect angle dependency. The human being can be considered as an approximately omnidirectional reflecting object while the passenger car shows a strong aspect angle dependency of its RCS. Thus, it can be expected that the receive signal amplitude will vary significantly if the passenger car will change its aspect angle relative to the measuring radar sensor.

If radar sensors are utilized in automotive applications, one important reflection property difference to the classical field of airborne radar must be taken into account: The distance between the radar sensors and the detected object is relative small (up to 200 m) compared to the object size. Thus, objects must be considered as extended targets, which especially becomes important in the case of distributed sensors in a near- or medium-range radar network. Because each radar sensor is observing the extended object from a different aspect angle, it is likely that each sensor will receive a reflection from a different reflection center on the car body and with a different amplitude. As an example, figure 1.10 (a) shows an object detection of a two-sensor radar network under the assumption that all reflections originate from the same reflection center and figure 1.10 (b) shows the more realistic detection scenario with an extended target, where the two radar sensors receive multiple reflections originating from different reflection centers on the car body. This important detection behavior of extended targets must be taken into account when it comes to the design of an automotive radar network with an appropriate postprocessing architecture. As a consequence, the experiences in developing postprocessing algorithms for airborne radar applications cannot be simply transferred to the field of automotive radar. They have to be enhanced with the focus on the fact that one object may appear as numerous detections that may not belong to the same reflection centers on the object. This important reflection behavior of extended targets will lead to a special clustering concept implemented in the later presented postprocessing architecture.

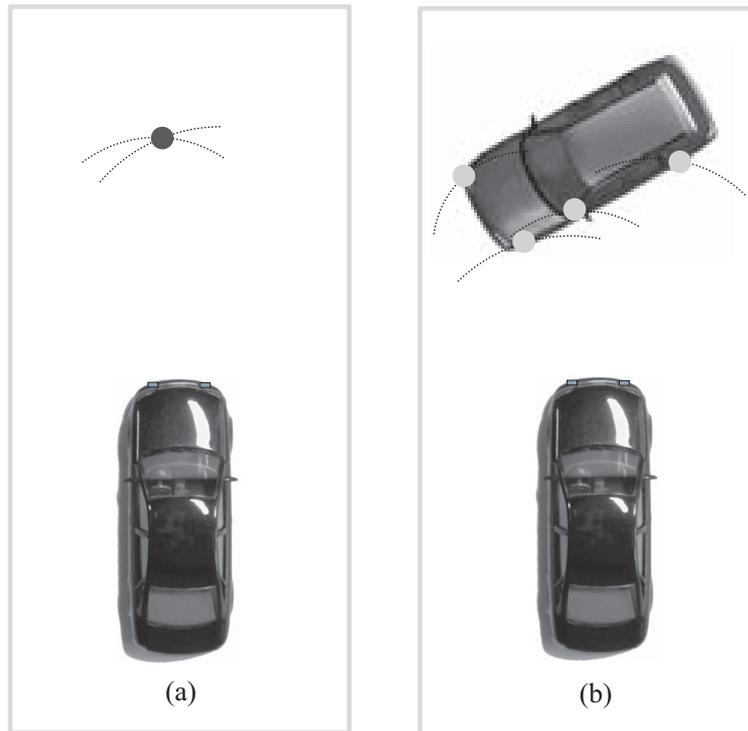


Figure 1.10: Point-target (a) and extended object (b) as detected by a two-sensor radar network

### 1.2.4 Summary

This section introduced some basic radar measurement principles determining range, velocity, and azimuth angle information from the received radar signal. It focused on two transmit signal waveforms being implemented in the two automotive radar sensors presented in the next section. Both transmit waveforms have advantages and disadvantages due to their measurement principle. The advantage of the LFM-CW radar is the achievable high resolution paired with a low implementation complexity while simultaneously measuring range and velocity. The pulse radar must have very short pulses to have the same high resolution performance. The generation of these ultra-short pulses combined with the necessary switching between the transmit and receive mode as well as the necessary signal processing bandwidth in the receiver are serious disadvantages of the pulse radar technique. The determination of the object's velocity requires also a complex pulse-Doppler signal processing. On the other hand, the CW-radar technique cannot offer the same high maximum measurement ranges of a pulse radar due to the concept of permanent transmission. Finally, the reflection behavior of different objects occurring in real road traffic scenarios have been outlined. At this, it has been stressed that vehicles must be considered as extended targets. This significant difference in automotive radar applications compared to the classical field of airborne radar will play an important role in the radar network design in the following sections.

Parameter	MR Sensor	HRR Pulse Sensor
Operating Frequency	24 GHz	24 GHz
Cycle Time	25 ms	20 ms
Range Coverage	0.75 ... 50 m	0.15 ... 30 m
Range Accuracy	0.12 m	0.03 m
Range Resolution (bandwidth)	0.6 m (FMCW: 250 MHz)	0.15 m (Pulse: 1 GHz)
Transmit Antenna	$6 \times 1$ patch array	$6 \times 1$ patch array
Receiving Antenna	$6 \times 2$ patch array	$6 \times 1$ patch array
Elevation Angle Coverage	$\pm 13^\circ$	$\pm 13^\circ$
Azimuth Angle Coverage	$\pm 40^\circ$	$\pm 40^\circ$
Azimuth Angular Accuracy	$1^\circ$	-
Azimuth Angular Resolution	<i>no</i>	-
Speed Coverage	$-250 \text{ km/h} \dots 91 \text{ km/h}$	-
Speed Accuracy	0.25 km/h	-
Speed Resolution	1.0 km/h	-

Table 1.4: Sensor Properties

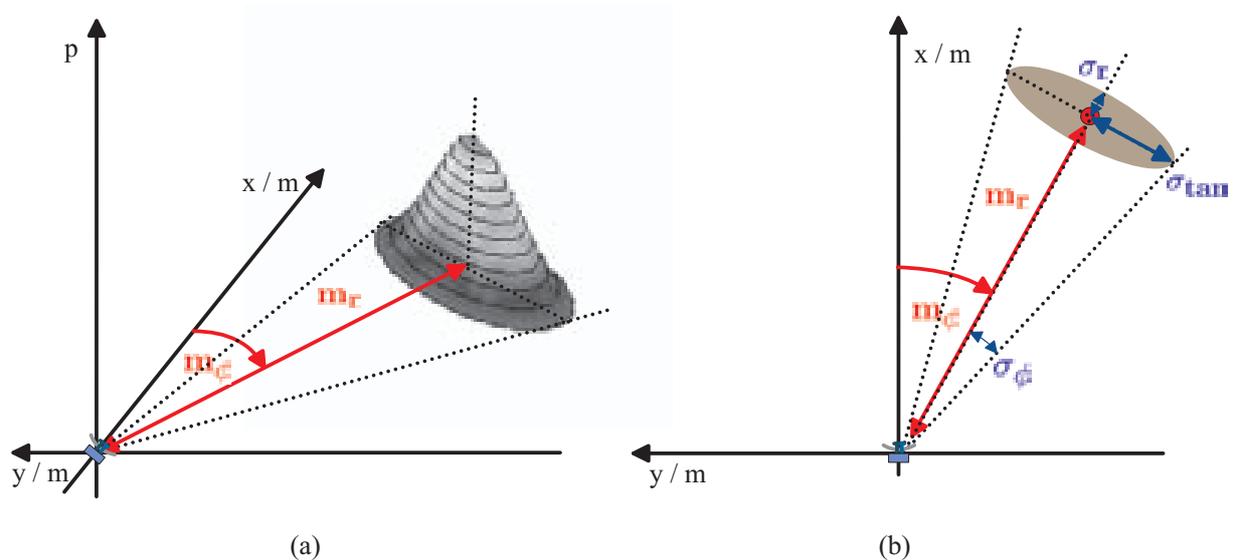


Figure 1.11: Definition of a sensor's range and azimuth angle accuracy

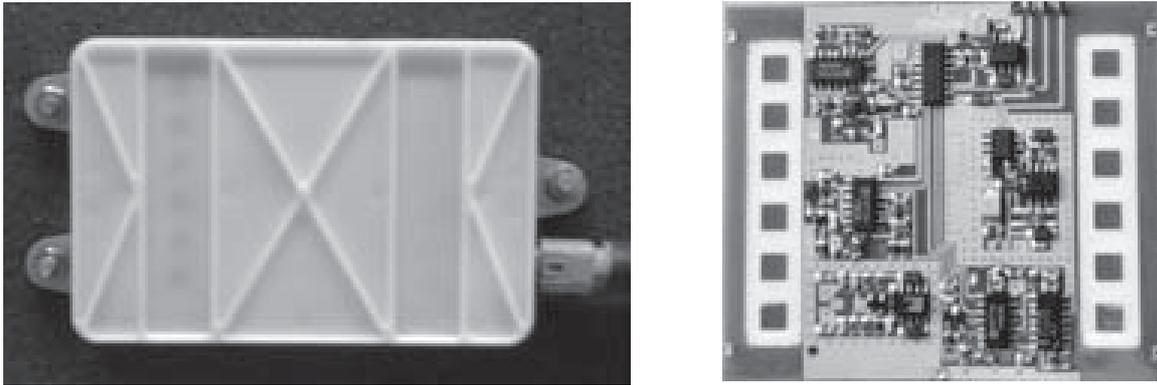


Figure 1.12: Picture and detailed view of the high range resolution (HRR) pulse radar sensor

## 1.3 Radar Sensors for Automotive Applications

The radar networks considered in this thesis are consisting of two different high-performance radar sensors applicable for near-distance automotive radar systems. Both sensors operate at a carrier frequency of 24 GHz. The first sensor is a high range resolution (HRR) pulse radar sensor, which is only measuring target ranges but with an extremely high accuracy. The second radar sensor is a so-called universal medium range radar (UMRR) sensor that measures target range and radial velocity as well as the azimuth angle of the target direction based on the monopulse radar (MR) technique. These two radar sensors will be utilized to establish three different types of radar networks that will be compared in the next section. Before that, the properties of the two sensors will be summarized briefly. At this, the accuracies of the single sensor measurements  $m_r^{\text{Sens}}$  and  $m_\phi^{\text{Sens}}$  are defined as their standard deviations  $\sigma_r^{\text{Sens}}$  and  $\sigma_\phi^{\text{Sens}}$  as visualized in figure 1.11.

### 1.3.1 High Range Resolution Pulse Radar Sensor

The HRR radar sensor as presented in [WS98] and [Klo02] is a pulse radar sensor measuring the range according to the pulse delay ranging principle described in section 1.2. Due to its ultra-short pulses (1 ns), it achieves very precise range measurements with an accuracy of a few centimeters. As described in [Klo02], velocity measurements are possible by applying pulse-Doppler frequency processing. But due to the great number of range gates, this would require a very high computation effort, so that the investigated HRR sensors are implemented as range only measuring sensor with no pulse-Doppler processing. As shown in table 1.4, the HRR pulse radar sensor requires a bandwidth of 1 GHz in order to achieve a range resolution of 0.15 m and a range accuracy of 0.03 m.



Figure 1.13: Universal Medium Range (UMRR) Radar

This bandwidth exceeds the available 250 MHz at 24 GHz in several European countries. But according to the latest political decisions, this ultra wide bandwidth will be available for a limited time period until the year 2013. As shown on the circuit board in figure 1.12, the HRR pulse radar sensor has two  $6 \times 1$  patch array antennas, i.e. one transmit and one receive antenna. The 3dB-beamwidth of such an antenna is  $\pm 40^\circ$  in the azimuthal and  $\pm 13^\circ$  in the elevation plane. A complete scan over the distance of 30 m, i.e. one measurement cycle, takes 20 ms. This corresponds to a target list update rate of 50 Hz. Due to its extremely high range accuracy and resolution the HRR pulse radar sensor is predestined to be implemented in a multi-sensor radar network that is determining the azimuth angle of an object by applying the trilateration technique described in section 1.2. The quality of the estimated azimuth angle is depending highly on the accurate measurement of the object's range. The position estimation properties of such a multi-sensor radar network will be discussed in detail in the next section.

### 1.3.2 Monopulse Radar Sensor

The universal medium range radar (UMRR) sensor as presented in [MBM03] and [MR02] can operate with different transmit signal waveforms. The UMRR sensors considered in this thesis are working with an LFM-CW narrowband transmit signal as it has been presented in section 1.2. It requires a bandwidth of 250 MHz and thus is compliant to the available bandwidth at 24 GHz throughout most countries of the European Union. It can be utilized for automotive applications without any limitation in the period of time as it is given for ultra wideband radar sensors at 24 GHz. A picture of the UMRR sensor is shown in figure 1.13. This sensor is capable of measuring a target at a radial distance up to 50 m with an accuracy of 0.12 m. It measures simultaneously the target's radial velocity with an accuracy of 0.25 km/h. Due to the transmit signal bandwidth of 250 MHz, it achieves a possible range resolution of 0.6 m between two targets with the same reflectivity and the same radial velocity. If two targets are located at the same range ( $r_1 = r_2$ ), both

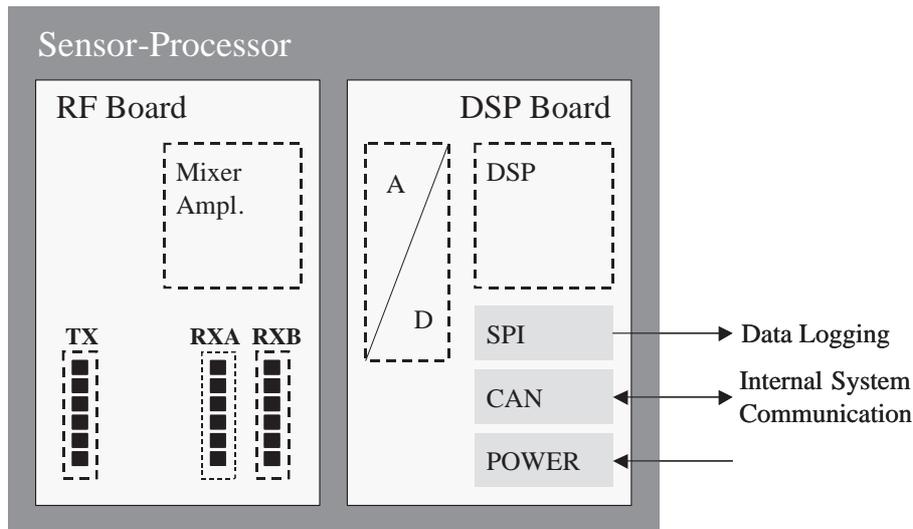


Figure 1.14: Block diagram of the UMRR sensors

can be resolved if their radial velocity difference is greater  $\Delta V_r \geq 1.0$  km/h. The block diagram in figure 1.14 shows schematically the RF frontend and the DSP module of the sensor. One transmit and two receive patch array antennas are integrated in the RF module with  $6 \times 1$  elements each. This dual receive antenna setup allows the direct azimuth angle measurement by applying the monopulse technique. The implementation of this angle estimation technique achieves an azimuth angular accuracy of  $1^\circ$ . As stated in [MBM03], a target separation in azimuth angular direction is not possible due to the actual monopulse concept. But in the future the UMRR antenna concept will be modified, so that a true angular resolution will be achieved. Thus, the only condition for resolving two targets with the same reflectivity is the radial distance or the radial velocity difference between them in case of the same radial velocity or the same distance, respectively. Although the UMRR sensor is not offering the precise range measurements of the HRR pulse radar sensor, it has the advantage of measuring almost the complete kinematic state of a detected target, namely the range, the azimuth angle, and the radial velocity. Due to this set of measurements, the UMRR sensor can either be implemented in a multi-sensor network or as a stand-alone system. These two possibilities will be investigated in the next section along with a multi-sensor HRR pulse radar network.

Radar Network Type 1	Radar Network Type 2	Single MR Sensor
4 HRR Puls radar Sensors	2 MR Sensors	1 MR Sensor
4 measurements: $m_r^{\text{Sens } n}$ ( $n = 1, \dots, 4$ )	4 measurements: $m_r^{\text{Sens } n}, m_\varphi^{\text{Sens } n}$ ( $n = 1, 2$ )	2 measurements: $m_r^{\text{Sens } n}, m_\varphi^{\text{Sens } n}$ ( $n = 1$ )

Table 1.5: Comparison of two 24 GHz Radar Networks

## 1.4 Multi-Sensor Radar Network

In this section, three different automotive radar network concepts will be compared. As already described in section 1.2, radar sensors can be grouped to a radar network in order to obtain a precise position estimation of a detected object. The azimuth angle of the object's direction can be estimated by applying a trilateration algorithm in a radar network consisting of only range measuring sensors or by using monopulse radar sensors, either in a multi-sensor network or as stand-alone sensors. In the following, two different types of radar networks as well as one stand-alone sensor will be compared in their position estimation performance under ideal measurement conditions. First, the concept of the radar networks will be outlined. Then, a brief derivation of the comparison criteria will follow, and finally, the performance as well as the advantages and disadvantages of the different radar network concepts will be presented.

### 1.4.1 Radar Network Concepts

The basis of this radar network comparison is the available number of measurements corresponding to one detected object. The first two different networks are designed with the precondition that four measurement values per target must be available if the detection probability of each sensor is ideal  $p_{\text{detect}} = 1$  (see table 1.5). The third system is a stand-alone MR sensor and delivers two measurements per target. This investigation will be restricted to an ideal detection probability as well as to the detection of point targets in order to investigate the basic position estimation behavior of the different radar systems.

The *first radar network* consists of four high performance HRR pulse radar sensors as described in table 1.4. These sensors have an ultra-short pulse length (1 ns) and can achieve very precise range measurements with an accuracy of a few centimeters. The available measurements determining the object position are given by

$$\mathbf{m}^{\text{HRR } n} = [m_r^{\text{HRR } n}] \quad n = 1, \dots, 4 \quad (1.20)$$

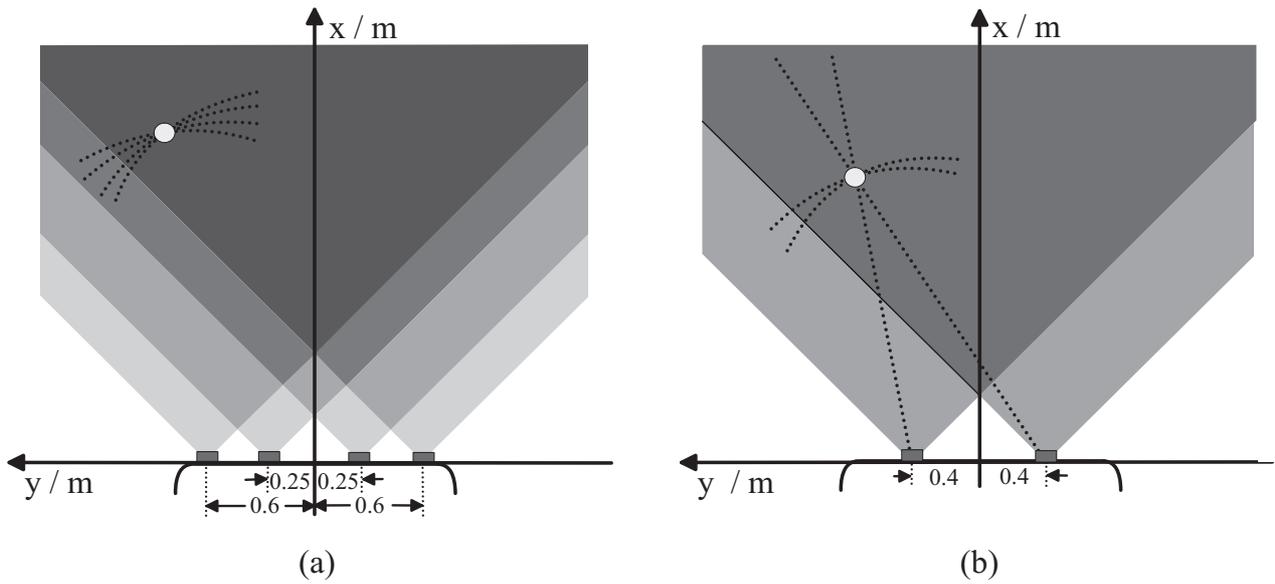


Figure 1.15: 4 HRR sensor (a) and 2 UMRR (b) radar network

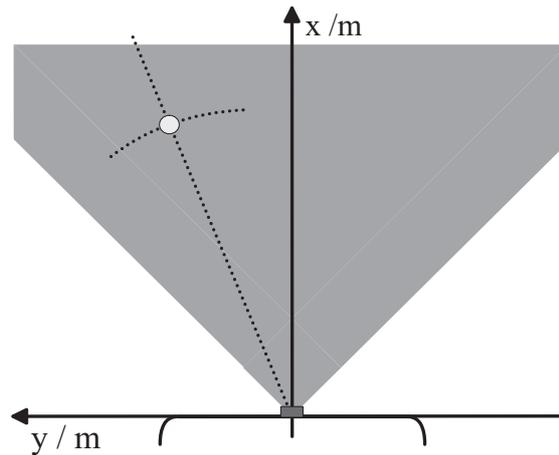


Figure 1.16: Single UMRR sensor

The sensors are mounted in the vehicle's front bumper as schematically shown in figure 1.15(a).

The *second radar network* consists of two monopulse radar sensors, which have the same position estimation properties as the described UMRR sensors in table 1.4. This MR sensor combination offers also four measurements values per detected object, namely two ranges and two estimated azimuth angles.

$$\mathbf{m}^{\text{MR } n} = \begin{bmatrix} m_{\varphi}^{\text{MR } n} \\ m_r^{\text{MR } n} \end{bmatrix} \quad n = 1, 2 \quad (1.21)$$

As shown in figure 1.15(b), the two monopulse radar sensors are mounted in a vehicle front bumper at 0.4 m on the left and right side of the origin of ordinates.

In order to get an impression of the estimation gain of these two multi-sensor radar networks, *a single MR sensor* will also be investigated. As already pointed out, this MR sensor is capable of simultaneously measuring the range and estimating the azimuth angle of a detected object

$$\mathbf{m}^{\text{MR}n} = \begin{bmatrix} m_{\phi}^{\text{MR}n} \\ m_r^{\text{MR}n} \end{bmatrix} \quad n = 1 \quad (1.22)$$

Thus, the object's position can be determined by a simple geometric calculation.

### 1.4.2 Radar Network Performance

The position estimation performances of the two radar networks as well as the one of the single MR sensor will be compared in accuracy and resolution. Since these two terms have only been defined for single radar sensors in section 1.2, the question must be answered what accuracy and resolution definitions are reasonable for a multi-sensor radar network.

A *radar network's position estimation accuracy* can be defined analog to the single sensor's measurement accuracy and will be derived in the following. Given the radar network's estimated position of a detected target by

$$\hat{\mathbf{t}} = \begin{bmatrix} \hat{t}_x \\ \hat{t}_y \end{bmatrix} \quad (1.23)$$

and its real position by

$$\mathbf{t} = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (1.24)$$

then the position estimation accuracy can be described by the expectation of the position estimate

$$E[\hat{\mathbf{t}}] \quad (1.25)$$

and the covariance matrix of the estimate  $\hat{\mathbf{t}}$

$$P = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{yx}^2 & \sigma_y^2 \end{bmatrix} \quad (1.26)$$

This covariance matrix of the radar network's position estimate can directly be obtained from the  $N$  measurement noise covariance matrices of the  $N$  radar sensors in the network. In the following these accuracy measures will be derived from the single sensor information:

Each radar network is described by the sensor placement in the front bumper. The position of the  $n$ -th sensor is defined by

$$\mathbf{s}^{\text{Sens}n} = \begin{bmatrix} s_x^{\text{Sens}n} \\ s_y^{\text{Sens}n} \end{bmatrix} \quad (1.27)$$

The measurements of each sensor are given relative to the appropriate sensor position. The geometric relation between the ideal measurements  $t_\varphi^{\text{Sens } n}, t_r^{\text{Sens } n}$  of sensor  $n$  and the real target position  $\mathbf{t}$  in a global cartesian coordinate system must be known. It is given by the following set of nonlinear equations

$$\begin{aligned} t_\varphi^{\text{Sens } n} &= f_\varphi(\mathbf{t}, \mathbf{s}^{\text{Sens } n}) = \arctan\left(\frac{s_y^{\text{Sens } n} - t_y}{s_x^{\text{Sens } n} - t_x}\right) \\ t_r^{\text{Sens } n} &= f_r(\mathbf{t}, \mathbf{s}^{\text{Sens } n}) = \sqrt{(t_x - s_x^{\text{Sens } n})^2 + (t_y - s_y^{\text{Sens } n})^2} \end{aligned} \quad (1.28)$$

By adding a sensor-dependent measurement error  $e^{\text{Sens } n}$ , we get realistic measurements and can rewrite the above equations for each of the two considered radar sensors.

The *measurements of the HRR radar sensor* are given by

$$\mathbf{m}^{\text{HRR } n} = [m_r^{\text{HRR } n}] = [f_r(\mathbf{t}, \mathbf{s}^{\text{Sens } n})] + [e_r^{\text{HRR } n}] \quad n = 1, \dots, 4 \quad (1.29)$$

It will be assumed that these measurements are unbiased

$$E[e_r^{\text{HRR } n}] = 0 \quad (1.30)$$

and that the measurement noise covariance matrix is given by

$$R^{\text{HRR } n} = [\sigma_r^{\text{HRR } 2}] \quad (1.31)$$

Analogously, the *measurements of the MR sensor* are given by

$$\mathbf{m}^{\text{MR } n} = \begin{bmatrix} m_r^{\text{MR } n} \\ m_\varphi^{\text{MR } n} \end{bmatrix} = \begin{bmatrix} f_r(\mathbf{t}, \mathbf{s}^{\text{Sens } n}) \\ f_\varphi(\mathbf{t}, \mathbf{s}^{\text{Sens } n}) \end{bmatrix} + \begin{bmatrix} e_r^{\text{MR } n} \\ e_\varphi^{\text{MR } n} \end{bmatrix} \quad i = 1, 2 \quad (1.32)$$

with

$$\begin{aligned} E[e_r^{\text{MR } n}] &= 0 \\ E[e_\varphi^{\text{MR } n}] &= 0 \end{aligned} \quad (1.33)$$

and the measurement noise covariance matrix

$$R^{\text{MR } n} = \begin{bmatrix} \sigma_r^{\text{MR } 2} & 0 \\ 0 & \sigma_\varphi^{\text{MR } 2} \end{bmatrix} \quad (1.34)$$

Considering an arbitrary  $N$ -sensor radar network with  $N$  different single sensor measurements per target, then a complete measurement set is given by the vector  $\mathbf{m}$

$$\mathbf{m} = \begin{bmatrix} \mathbf{m}^{\text{Sens } 1} \\ \mathbf{m}^{\text{Sens } 2} \\ \vdots \\ \mathbf{m}^{\text{Sens } N} \end{bmatrix} \quad (1.35)$$

with the covariance matrix containing the  $N$  single sensor measurement noise covariance matrices of the  $N$  radar network sensors

$$R = \begin{bmatrix} R^{\text{Sens } 1} & 0 & \dots & 0 \\ 0 & R^{\text{Sens } 2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R^{\text{Sens } N} \end{bmatrix} \quad (1.36)$$

Because the measurements of the HRR pulse radar sensor  $\mathbf{m}^{\text{HRR } n}$  as well as the ones of the MR sensor  $\mathbf{m}^{\text{MR } n}$  are unbiased, the radar network's estimated target position  $\hat{\mathbf{t}}$  calculated according to the set of nonlinear equations in (1.28) is also unbiased.

$$E[\hat{\mathbf{t}}] = \mathbf{t} \quad (1.37)$$

The radar network's position estimate covariance  $P$  of eq.(1.26) can be calculated from the sensors' measurement noise covariance matrices in the following way

$$P = (J^T R^{-1} J)^{-1}; \quad (1.38)$$

At this,  $J$  is the Jacobi matrix derived from the nonlinear measurement equations of the considered radar sensor (see (1.29), (1.32)) and evaluated at the estimated position  $\hat{\mathbf{t}}$

$$J = \left[ \frac{\partial \mathbf{m}}{\partial t_x} \quad \frac{\partial \mathbf{m}}{\partial t_y} \right]_{\mathbf{t}=\hat{\mathbf{t}}} \quad (1.39)$$

The position estimation covariance matrix in eq.(1.38) refers to cartesian coordinates. To obtain this accuracy measure in a more compact way, i.e. with negligible non-diagonal matrix elements, the covariance matrix  $P$  must be linearly transformed to polar coordinates by

$$\Sigma = U^T P U \quad (1.40)$$

with the transformation matrix

$$U = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix} \quad (1.41)$$

where

$$\varphi = \arctan\left(\frac{t_y}{t_x}\right) \quad (1.42)$$

The obtained polar coordinate covariance matrix  $\Sigma$  has diagonal elements  $\Sigma_{(i,i)}$  being much greater than non-diagonal elements  $\Sigma_{(i,j)}$

$$\Sigma_{(i,i)} \gg \Sigma_{(i,j)} \quad i \neq j \quad (1.43)$$

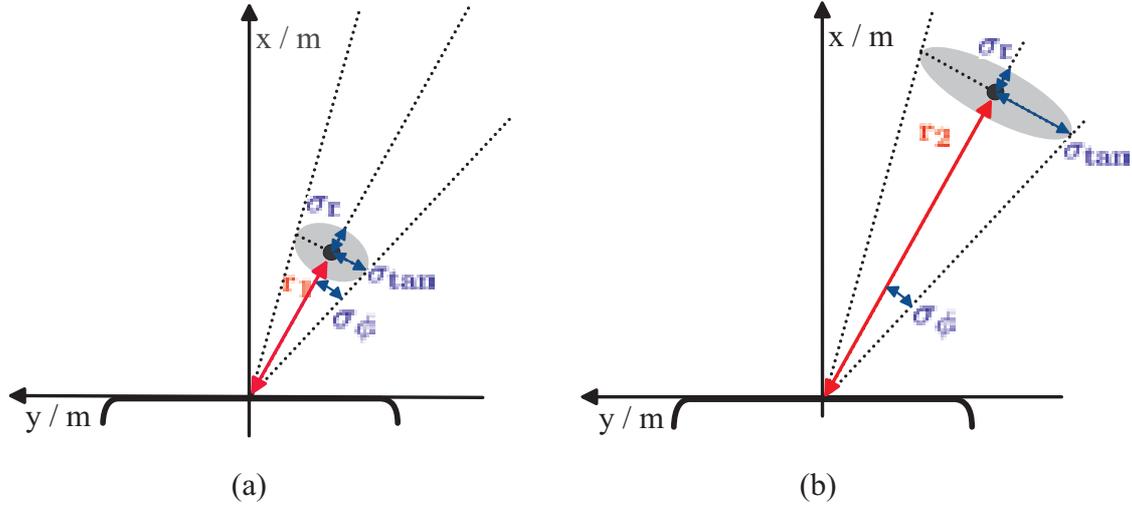


Figure 1.17: Standard deviations describing the radar network's position estimation accuracy at the distance  $r_1$  (a) and  $r_2$  (b) relative to the origin of coordinates

Due to this property, the diagonal elements of  $\Sigma$  can be interpreted as the nearly independent variances along the radial  $\vec{u}_r$  and the tangential direction  $\vec{u}_{tan}$  with respect to the true object position ( $t_x = E[\hat{t}_x], t_y = E[\hat{t}_y]$ ).

$$\Sigma = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_{tan}^2 \end{bmatrix} \quad (1.44)$$

Thus, the radial standard deviation is given by

$$\sigma_r = \sqrt{\Sigma_{(1,1)}} \quad (1.45)$$

and the standard deviation in the tangential direction is

$$\sigma_{tan} = \sqrt{\Sigma_{(2,2)}} \quad (1.46)$$

Both standard deviations of the radar network's position estimate are illustrated in figure 1.17. It can be seen that  $\sigma_{tan}$  is increasing, the higher the target range. The angular standard deviation  $\sigma_\varphi$  is position independent and can be calculated from  $\sigma_{tan}$  by

$$\sigma_\varphi = \arctan \left( \frac{\sigma_{tan}}{\sqrt{t_x^2 + t_y^2}} \right) \quad (1.47)$$

$\sigma_r$  and  $\sigma_\varphi$  will serve as position estimate accuracy measures in the radar network comparison in the next section:

*Radar Network Position Estimation Accuracy:*

$$\sigma_r = \sqrt{\Sigma_{(1,1)}} \quad (1.48)$$

$$\sigma_\varphi = \arctan \left( \frac{\sqrt{\Sigma_{(2,2)}}}{\sqrt{t_x^2 + t_y^2}} \right) \quad (1.49)$$

This analytical derived accuracy measures are equivalent to the accuracy measures presented in [SFR03], where a minimum mean square error (MMSE) position estimation was outlined and position estimation accuracy results were obtained in a Monte-Carlo simulation.

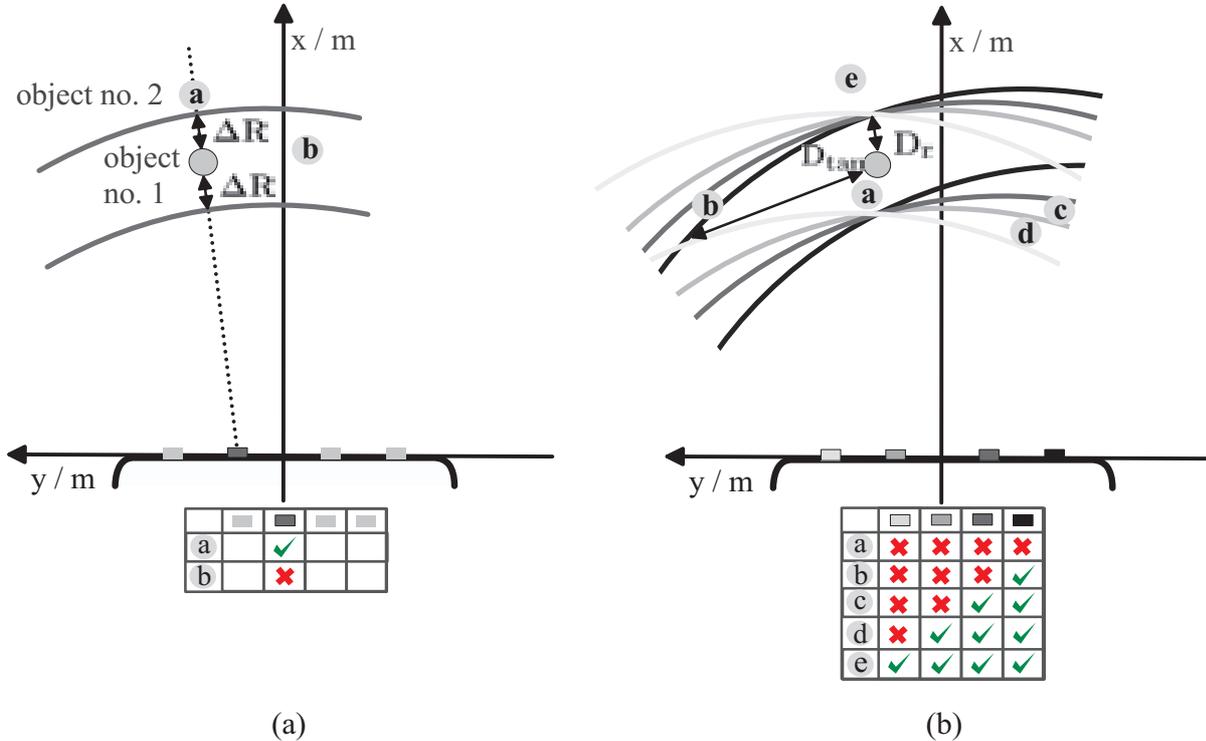


Figure 1.18: Range resolution condition of one radar sensor (a) and of all four radar sensors (b) with two tables summarizing the possibility of resolving the considered two targets for each radar sensor

The *resolution of a multi-sensor radar network* must be derived directly from the single sensor resolution conditions. As described in section 1.3, each of the considered radar sensors has its own resolution condition. In the case of the HRR pulse radar, the only resolution condition is the necessary radial distance difference  $\Delta R$  between two targets with the same reflection behavior. This resolution condition is visualized in figure 1.18 (a) as two circular arcs with the distance  $\Delta R$  to the position of target no. 1. This simple example demonstrates that target no. 1 and no. 2 can be resolved by the shown radar sensor if target no. 2 is located at a greater radial distance than  $\Delta R$  to target no. 1, e.g. at position  $a$ . The separation of the two targets is not possible if target no. 2 is placed at position  $b$ . The table below the coordinate system summarizes this example by denoting the possible resolution with a green hook and the impossibility of resolving the two targets with a red cross. Considering now the 4 HRR radar sensor network, the term resolution has to be defined two-dimensional, because the estimated position is also two-dimensional. Figure 1.18(b) shows the different single sensor resolution conditions

for a 4 HRR sensor radar network. The table below indicates if the resolution of the two targets is achieved by the different sensors depending on the position of target no. 2. Two targets with the same reflectivity are considered as resolved if one sensor in the radar network is detecting both separately. One can argue that at least two sensors must be able to separate the two targets in order to estimate two positions and one can also demand that the majority of the sensors, e.g. three of the four sensors, must resolve the two targets in order to achieve good quality position estimates. But the term resolution gives by definition only an indication about the possibility of resolving the two target situation, not about the quality of the estimation. Thus, this radar network resolution can be described by the only eye-catching characteristic in figure 1.18(b), the rhombic area around target no.1. This is the area where none of the sensors can resolve the two targets. If the second target is located outside, both can be resolved by at least one radar sensor. This rhombic area has a certain extension in the radial  $D_r$  and the lateral direction  $D_{tan}$ . Due to the sensor constellation in an automotive radar network,  $D_r$  will not change significantly at different positions in the radar network's scan area. But the extension of the rhombic area in the tangential direction  $D_{tan}$  will vary strongly, so that this quantity will draw the main attention in the radar network comparison in the next section.

The UMR sensor has a different resolution condition because it is a LFM-CW radar sensor. The target ranges and radial velocities are derived from the detected peak frequencies in the downconverted receive signal spectrum. As explained in section 1.2.2, the only resolution condition of an LFM-CW radar is the resolution of two peak frequencies corresponding to two targets in the frequency spectrum. Since these peak frequencies contain a superposition of the range and the velocity information, two resolution side conditions can be derived from the peak frequency resolution condition in eq.(1.13). If two targets have the same radial velocity ( $(v_{r_1} = v_{r_2})$ ), then the resolution of both depends only on the radial distance difference  $\Delta R$ . This leads to the same derivation of a multi-sensor network resolution condition as shown in figure 1.18(b) with a similar rhombic area where none of the sensors can resolve the two targets with the same velocity. Due to the UMR sensor's velocity measurement capability, it can additionally resolve two targets located at the same range ( $r_1 = r_2$ ) if the radial velocity difference between both is at least  $\Delta V_r \geq 1.0 \text{ km/h}$ . The possibility of separating two targets due to their different radial velocities is a clear advantage of the UMR sensor compared to the HRR pulse radar sensor, which also gives the appropriate radar network an edge over the HRR multi-sensor radar network. In the following radar network concept comparison, the extension of the rhombic area, where none of the sensors can resolve two targets with the same reflectivity will be discussed and the consequences for practical automotive applications will be outlined.

### 1.4.3 Radar Network Type 1

The HRR pulse radar network estimates the object position with distance measurements only. This is achieved by applying a trilateration algorithm in the MMSE sense as it is basically described in section 1.2. The principle distribution of the radar network's

position estimates is shown in figure 1.19(a). It can be seen that the range accuracy of the estimated position is excellent but the azimuth angle estimates are not as accurate. Figure 1.19(b) shows the range accuracy results of the radar network for the complete scan area. A position independent range accuracy of

$$\sigma_r = 0.015 \text{ m} \quad (1.50)$$

is accomplished. This result could be expected because 4 sensors each with a range accuracy of  $\sigma_r^{\text{HRR}} = 0.03 \text{ m}$  have been utilized to obtain the radar network's position estimate. The radar network's range accuracy can analytically be expressed as

$$\sigma_r = \frac{1}{\sqrt{4}} \cdot \sigma_r^{\text{HRR}} \quad (1.51)$$

The angular accuracy  $\sigma_\varphi$  achieved by the ideal trilateration in figure 1.19(c) is position dependent and varies between

$$1.9^\circ \leq \sigma_\varphi \leq 2.6^\circ \quad (1.52)$$

At first glance, these azimuth angular accuracy numbers seem to be very good. But having a closer look at the analogous standard deviation  $\sigma_{tan}$  (figure 1.19(d)), which is the deviation orthogonal to the radial direction in m, reveals that the tangential standard deviation of the position estimate is relative high. The position dependent standard deviation  $\sigma_{tan}$  has a bath-tub-like characteristic with maximum values at the detection area borders. These maxima reach values of 1.3 m.  $\sigma_{tan}$  is increasing linearly in the x-direction and has a parabolic behavior in the y-direction. This lack of azimuth angular accuracy has its cause in the geometric configuration. Considering the maximum distance between the single radar sensors in the front bumper, which is 1.2 m, compared to the covered distance of 30 m, it becomes clear that a position estimation retrieved by a trilateration of range measurements will suffer a strong limitation in angular accuracy. Small range measurement errors will result in significant azimuth angle errors of the position estimate. This phenomenon is known as a dilution of precision (DOP) from the research field of satellite navigation. Such a DOP cannot be avoided in the case of only range measuring automotive radar networks because the positioning of the sensors is limited by the extension of the vehicle's front bumper.

As already discussed in the previous section, the resolution of a multi-sensor radar network cannot be described in a simple way. Instead, an area has been defined where none of the network sensors can resolve two targets with the same reflectivity. This area can be derived from the single sensor resolution condition of the HRR pulse radar sensors as shown in figure 1.20(a). The only resolution condition of a single HRR pulse radar sensor is radial distance  $\Delta R$  between two targets, which has to be  $\Delta R \geq 0.15 \text{ m}$ . The resulting extension of the area of no resolution is given by the two values  $D_r$  and  $D_{tan}$ .  $D_r$  is position independent and has a value of

$$D_r = 0.15 \text{ m} \quad (1.53)$$

as could be expected from the single sensor resolution. The extension orthogonal to the radial direction  $D_{tan}$  is strongly position dependent, e.g. it varies between  $4 \text{ m} \leq D_{tan} \leq 9 \text{ m}$

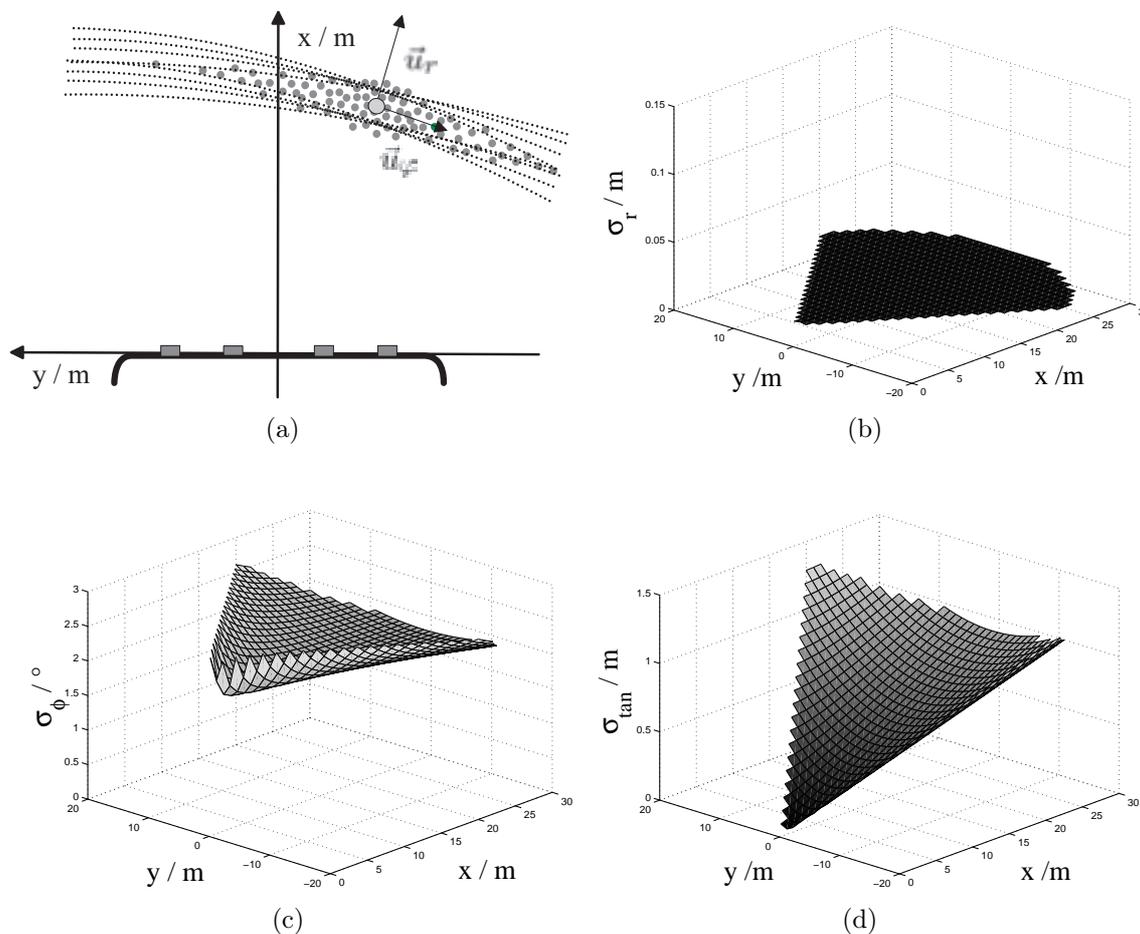


Figure 1.19: The 4 HRR radar network’s position estimation error due to the single-sensor measurement accuracies of  $\sigma_r^{\text{HRR}} = 0.03\text{m}$  (a). Figures (b), (c), and (d) show the resulting radar network’s position estimation accuracies  $\sigma_r$ ,  $\sigma_\phi$ , and  $\sigma_{tan}$ , respectively.

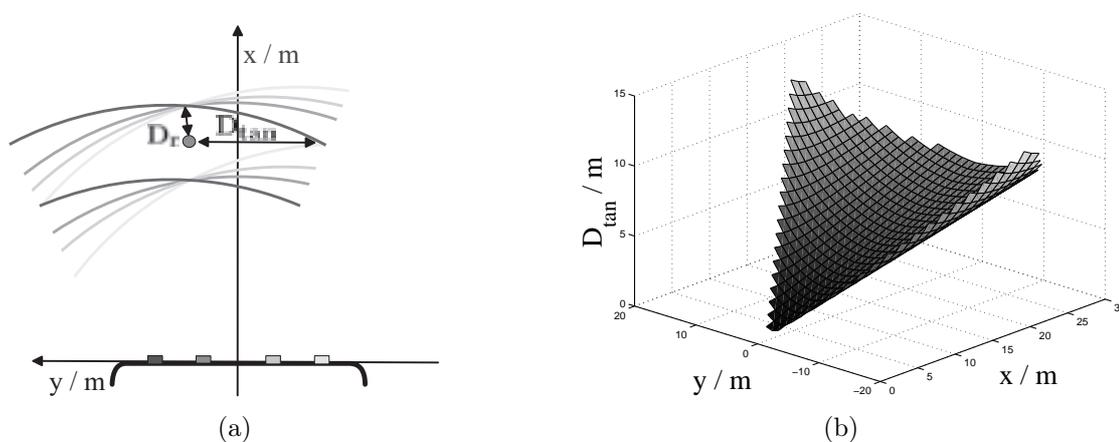


Figure 1.20: Complete single-sensor resolution conditions for the 4 HRR pulse sensor network (a) and the position dependent extension  $D_{tan}$  of the rhomb-like area where the resolution of two targets is impossible (b)

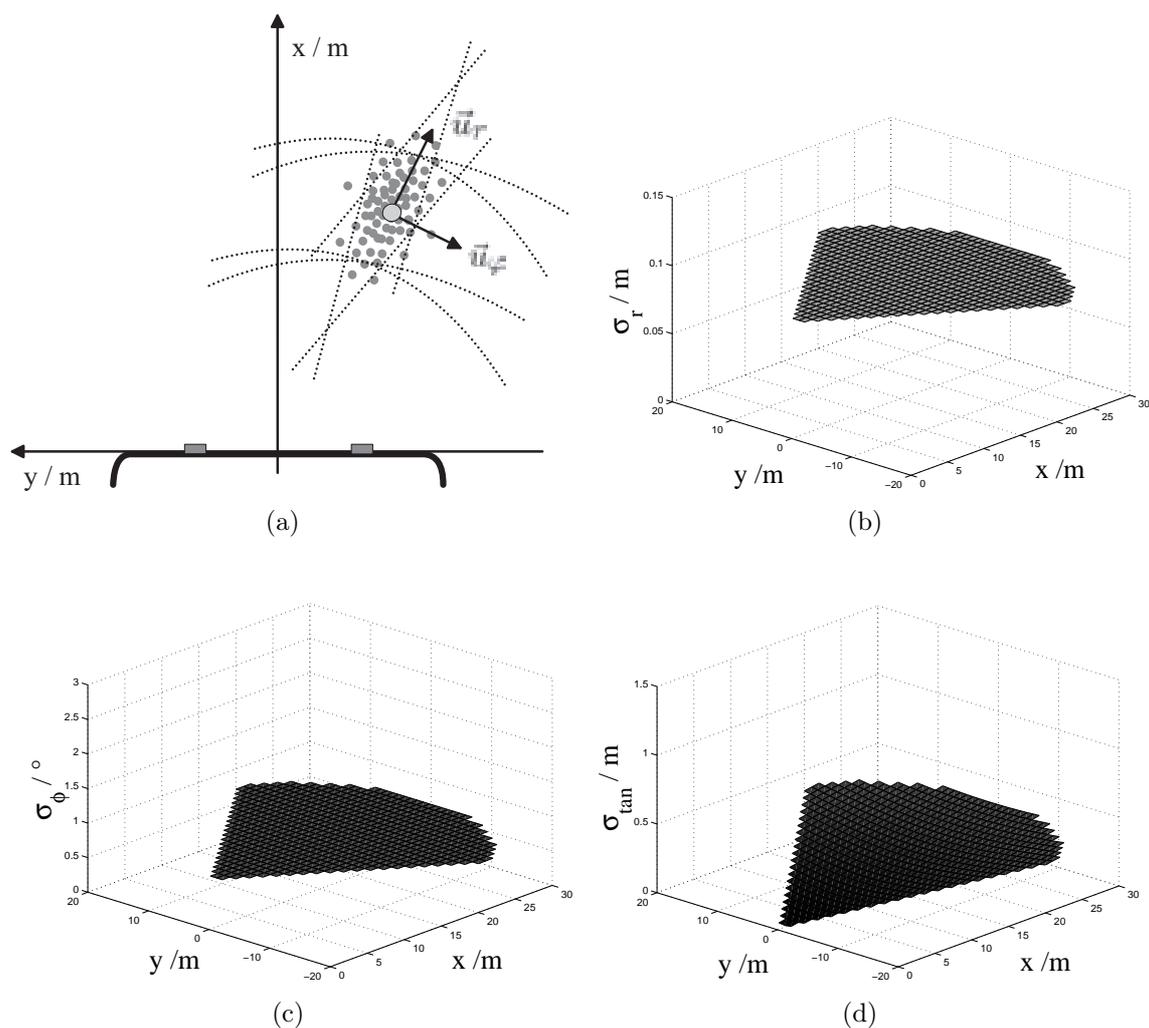


Figure 1.21: The 2 UMRR network's position estimation error due to the single-sensor measurement accuracies of  $\sigma_r^{\text{UMRR}} = 0.12\text{m}$  and  $\sigma_\phi^{\text{UMRR}} = 1^\circ$  (a). Figures (b), (c), and (d) show the resulting radar network's position estimation accuracies  $\sigma_r$ ,  $\sigma_\phi$ , and  $\sigma_{tan}$ , respectively.

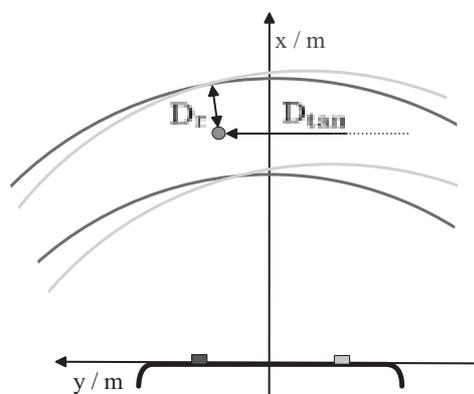


Figure 1.22: Complete single-sensor resolution conditions for the 2 UMRR sensor network

at a range  $r = 10$  m (figure 1.20(b)). The area of no resolution reaches its maximum lateral extension at the detection area boundaries with a maximum value of  $D_{tan} = 13$  m. Although this radar network comparison restricts itself to point targets, the following realistic example should illustrate the lateral resolution deficit of this network. If we consider a typical road traffic scenario, where the own car is following another car on the same lane and a third car from the oncoming traffic can be found on the left lane at about the same distance as the car ahead of us, then the HRR pulse radar network with a trilateration algorithm is not capable of resolving these two cars. Without any further signal processing of the estimated target positions, these dynamic situations cannot be interpreted correctly.

Overall, the HRR-pulse radar network delivers a very accurate range estimation but loses this accuracy in the azimuth angular direction due to the relative small distance between the sensor mounting positions compared to the covered distance. The resolution performance is limited due to its simple single sensor resolution condition, which allows only the separate detection of targets that have a different radial range of at least  $\Delta R \geq 0.15$  m.

#### 1.4.4 Radar Network Type 2

The advantage of the 2 UMRR sensor network is the possibility of measuring the azimuth angle of the target direction. Thus, the error in the angular direction  $\vec{u}_\varphi$  can be contained by the precise azimuth angle measurements of the UMRR sensors. The principle position estimate distribution of this radar network is shown in figure 1.21(a). Applying a MMSE position estimation to the available single sensor measurements leads to the network's position estimate accuracies shown in figure 1.21(b), (c), and (d). The position independent standard deviation  $\sigma_r$  of the UMRR network's estimate  $\hat{t}$  is not as accurate as the one of the HRR pulse radar network (see figure 1.21(b))

$$\sigma_r = 0.085 \text{ m} \quad (1.54)$$

but still can be considered as very precise. The azimuth angle estimation by the monopulse method helps to avoid angular errors. Hence, an extremely high, position independent azimuth angle accuracy  $\sigma_\varphi$  is achieved

$$\sigma_\varphi = 0.7^\circ \quad (1.55)$$

Considering the analogous standard deviation  $\sigma_{tan}$ , orthogonal to the radial direction  $\vec{u}_r$  in figure 1.21(d), it becomes clear that the 2 UMRR network shows a better performance than the 4 HRR pulse radar network. It has the same bath-tub-like characteristic as the one of the HRR pulse radar network and it is also increasing linearly in the x-direction, but with a significant lower gradient. Thus, the maximum value at the detection area boundaries is clearly smaller  $\sigma_{tan} = 0.4$  m.

Due to its LFM-CW measurement principle, the single UMRR sensor resolution condition for two targets with the same reflectivity is different to the one of a single HRR

sensor. Due to the superposition of range and radial velocity information in the detected peak frequency, two special resolution cases are considered. If the two targets have the same radial velocity ( $(v_{r_1} = v_{r_2})$ ), the resolution of both depends only on the radial distance difference  $\Delta R$ . This is the same resolution condition as the one of the HRR pulse radar network, except that it is only valid if the two targets have the same radial velocity. A similar area can be geometrically constructed for this special case, where none of the two UMRR sensors can resolve the targets (figure 1.22). The extension in the radial direction of this rhomb-like area is  $D_r = 0.6$  m. No value for the extension in the orthogonal direction can be given, because the circular arcs denoting the single sensor resolution conditions are not intersecting. This is caused by the mounting distance between the two sensors in the  $y$ -direction  $dy$ , which is smaller than twice the range resolution  $dy < 2 \cdot \Delta R$ . As a consequence, the UMRR radar network cannot resolve two targets with the same velocity if the radial distance between both is  $\Delta R \leq 0.6$  m. But due to the UMRR sensor's LFM-CW measurement principle, it also can resolve two targets located at the same range ( $r_1 = r_2$ ) if the radial velocity difference between both is  $\Delta V_r \geq 1.0$  km/h. This possibility of separating two targets due to their different radial velocities is a clear advantage of the UMRR sensor compared to the HRR pulse radar sensor. This also gives the UMRR network an edge over the HRR pulse radar network in the resolution performance in dynamic road traffic scenarios.

The UMRR network outperforms the HRR pulse radar network in position estimation accuracy because it takes advantage of the azimuth angle estimation of each sensor containing the loss of accuracy in the azimuth angular direction. Although the range accuracy of the HRR network is significantly higher than the one of the UMRR network, the azimuth angle accuracy gain is more important for the overall position estimation performance. The UMRR network also offers a better overall resolution. Both radar networks cannot achieve a lateral resolution of two targets situated at the same range in a realistic static measurement situation. But in dynamic multi-target scenarios where the detected targets have significantly different radial velocities, the 2 UMRR radar network achieves such a lateral resolution due to its LFM-CW measurement principle. Thus, the 2 UMRR radar network can be considered as the more versatile system.

### 1.4.5 Single MR Sensor

The purpose of a multi-sensor network is to determine the position of an object if only range measurements are available (4 HRR pulse radar network) or to achieve a more precise position estimation than in the single sensor case (2 UMRR network). The raising question is what position estimation performance is gained by a multi-sensor network compared to a single MR sensor. Therefore, the performance of such a sensor has also been investigated in accuracy and resolution. As already presented in table 1.4 the UMRR

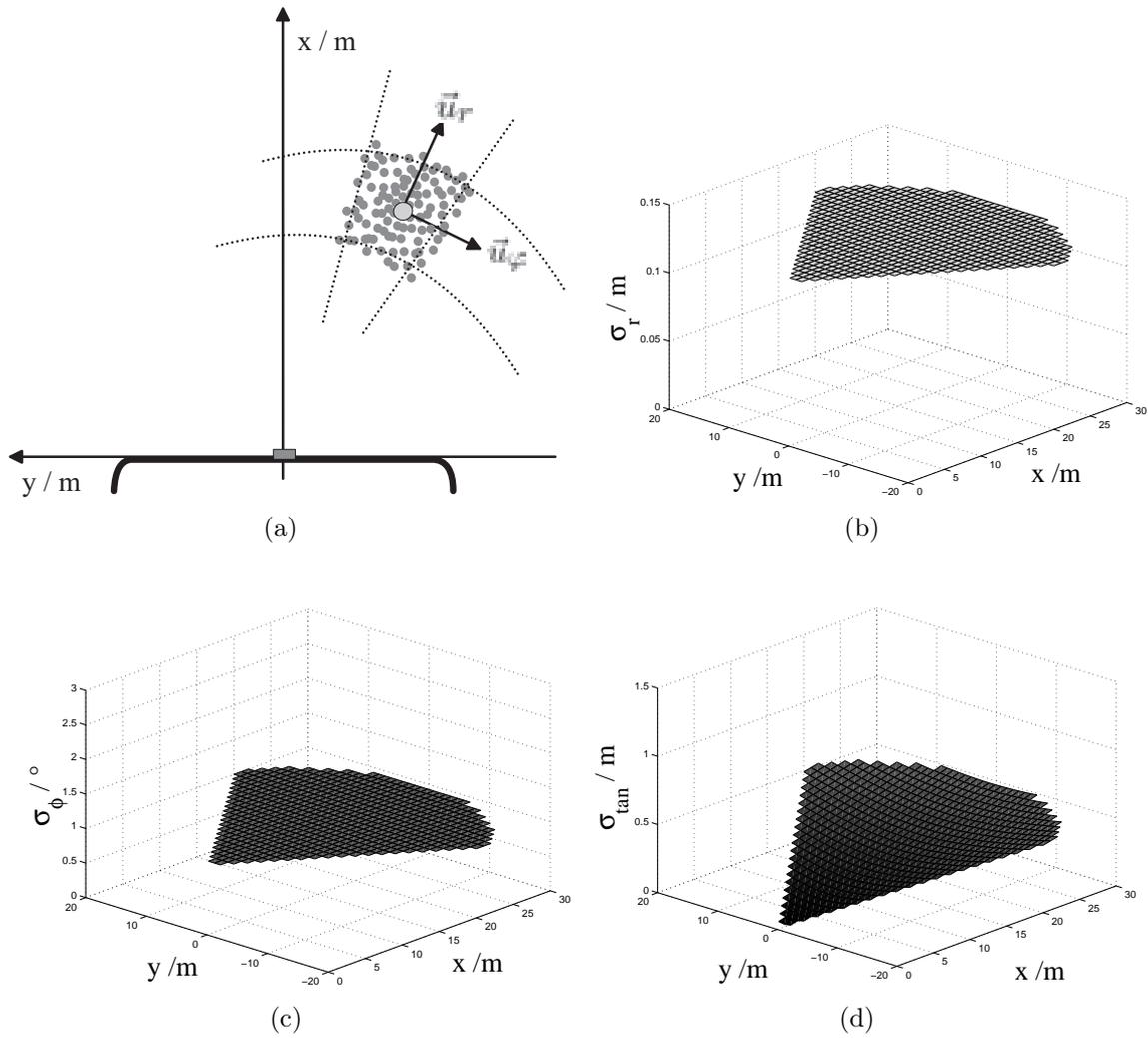


Figure 1.23: 1 UMRR sensor's position estimation error assuming measurement accuracies of  $\sigma_r^{\text{UMRR}} = 0.12\text{m}$  and  $\sigma_\phi^{\text{UMRR}} = 1^\circ$  (a). Figures (b), (c), and (d) show the position estimation accuracies  $\sigma_r$ ,  $\sigma_\phi$ , and  $\sigma_{tan}$ , respectively.

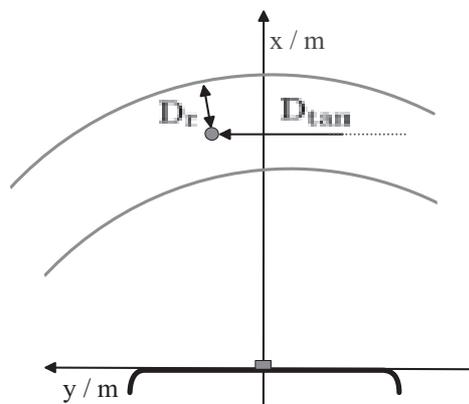


Figure 1.24: Single UMRR sensor resolution condition

sensor properties are

$$\sigma_r = 0.12 \text{ m} \quad (1.56)$$

$$\sigma_\varphi = 1^\circ \quad (1.57)$$

These standard deviations are position independent, so that the constant functions in the figures 1.23(b) and 1.23(c) could be expected. The analogous function  $\sigma_{tan}$  of the azimuth angular standard deviation  $\sigma_\varphi$  is shown in figure 1.23(d). It can be seen that the overall accuracy of the position estimation is clearly higher than the one of the HRR pulse radar network. This leads to the conclusion that one measured azimuth angle of a MR sensor enhances the azimuth angular estimation accuracy more than the high range accuracy of a few centimeters of each sensor in a 4 HRR multi-sensor network. Comparing the accuracy performance of the 2 UMRR radar network to the performance of the single UMRR sensor, the multi-sensor network accuracy is slightly higher by a factor of  $\sqrt{2}$ . This could be expected since number of available measurements is twice the one of the single sensor.

The resolution of one UMRR sensor has already been explained in the previous sections and was given in table 1.4. It is also visualized in figure 1.24. Since no significant resolution enhancement in the lateral direction is achieved by the 2 UMRR sensor network, the resolution performance of a single sensor and the one of the network can be considered to be the same.

Overall, the performance of a single MR sensor is higher than the one of the 4 HRR pulse radar network and slightly lower than the one of the 2 UMRR network. This gives the single UMRR sensor the advantage that it can be utilized as a stand-alone system either as a single sensor or in a distributed multi-sensor system, where all sensors deliver complete position estimates independent of the other sensor's target measurements. Overall, the single UMRR sensor offers flexible implementation possibilities, because it delivers the complete position information of a detected target.

### 1.4.6 Summary

This comparison of different radar network concepts has shown that a high position estimation performance could be achieved by all networks. At this, it has become clear that the availability of azimuth angle measurements enhances the position estimation accuracy significantly and gives the MR sensor network a clear advantage over the only range measuring 4 HRR pulse radar sensor network. Simultaneous velocity measurements give also the possibility to resolve different targets in dynamic road traffic scenarios. But some resolution limitations of the different radar networks have also become evident and will lead to merged observations for closely spaced targets. Such merged observations can be avoided if additional postprocessing steps are applied to achieve a high estimation performance over time. For example, it may occur that not every target is detected in each measurement cycle or that ghost-targets appear suddenly from cycle to cycle. In order to enhance this time-varying behavior of the different targets, different postprocessing

	Accuracy		Resolution		
	$\sigma_r/\text{m}$	$\sigma_\varphi/^\circ$	$D_r/\text{m}$	$D_{tan}/\text{m}$	$\Delta V_r/\text{km/h}$
Radar Network 1 (4 HRR)	0.015	2.1	0.15	$\leq 13$	-
Radar Network 2 (2 Mono)	0.085	0.7	0.6	-	1.0
Single MR sensor	0.12	1	0.6	-	1.0

Table 1.6: Radar network performances ( $p_{detect} = 1$ )

stages can be applied, so that ghost-targets, missed detections, measurement errors, or any additive noise process can be eliminated or smoothed. The presented comparison has been restricted to point targets, which will barely occur in real road traffic scenarios. As described in section 1.2, most of the road users must be considered as extended objects. As a consequence, each radar sensor will have a different aspect angle to the object and will receive one or more reflections of different reflection centers, .e.g. different measurements originating from different points on the car body. In order to achieve the same high position estimation performance as in the point target case, additional signal processing steps are necessary. Especially in multi-target situations, data association algorithms become essential to sort the single sensor's target lists according to their origin of reflection. All these different additional signal processing steps can be embraced in the term postprocessing algorithms. Such a principle postprocessing architecture for a multi-sensor radar network will be presented in the following section.

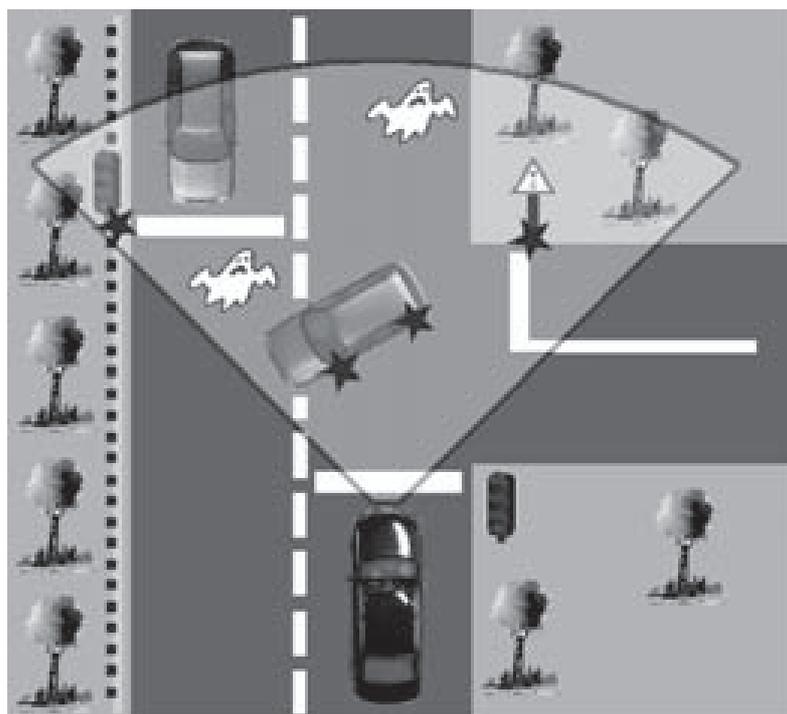


Figure 1.25: Example of a typical road traffic scenario

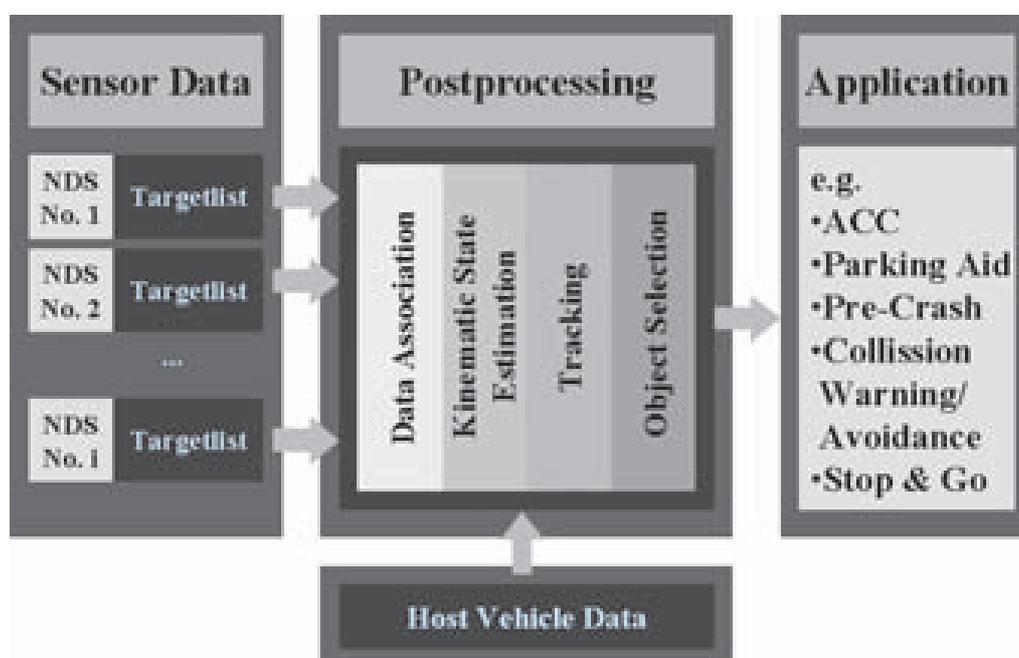


Figure 1.26: General postprocessing structure for an automotive multi-sensors network

## 1.5 A Postprocessing Structure for a Radar Network

As described in section 1.1, automotive radar applications demand high accuracy position and velocity estimates of road users with a high reliability. The task of postprocessing algorithms is to determine these stable and accurate position and velocity estimates as well as the extension of the detected objects from the error-prone or incomplete single-sensor measurements.

Figure 1.25 shows a typical snapshot of single-sensor target detections available in a single measurement cycle. The following cases may occur and have to be avoided, smoothed, or enhanced by applying different postprocessing steps:

- not all objects are detected in every measurement cycle (missed detections), e.g. the red car of the oncoming traffic in figure 1.25
- disturbance, clutter, or multipath propagation may cause echoes interpreted as targets that are not present in reality (ghost targets)
- due to the extension of road users one or more echoes may be received from different reflection centers on one object
- multi-target situations must be resolved and interpreted

Figure 1.26 shows a general postprocessing structure for an automotive multi-sensor network. The postprocessing algorithms have access to different information about the road traffic scenario and the host vehicle's kinematic state. The road traffic scenario information is given by the measurements of each single sensor in the radar network. For the radar sensors presented in section 1.3 this information may comprise

- target range  $r$
- radial velocity  $v_r$
- receive signal amplitude  $a$
- target azimuth angle  $\varphi$

The host vehicle data that is available for the postprocessing algorithms can be manifold. In this thesis, the utilized host vehicle information is

- the host vehicle velocity  $v_{host}$
- and the turning rate  $\omega_{host}$

The host vehicle velocity is measured by the ABS sensors of the four wheels and the turning rate is obtained from a yaw-rate sensor, which is typically implemented in today's ESP systems. This host vehicle information is available on the so-called powertrain CAN

of today's passenger cars.

In figure 1.26, the input data from the various radar sensor front-ends and from the host vehicle is first buffered. In the following postprocessing step, the detected targets are assigned to groups that originate from the same object. This is achieved by a data association algorithm that takes advantage of the knowledge about the actual target measurements of the radar sensors, the host vehicle information and the kinematic state prediction of previous, already evaluated objects formed to so-called object tracks. The actual target measurement data is either associated to already existing object tracks or is assigned to new target groups corresponding to a single object. At this, it is important to take the possible extension of the different road traffic users into account. This grouped data is then fed into the kinematic state estimation process calculating the position and velocity of the detected objects relative to the host vehicle from the single-cycle measurement data. The output of this kinematic state estimation step is fed into a tracking stage. The task of the tracking filter is to smooth the object's kinematic state estimate over time and to avoid detection discontinuities. For this purpose, it may assume a certain behavior of the object motion, a so-called target dynamic model, in order to achieve an optimal tracking filter estimate. If an object is detected for the first time, it forms a so-called unevaluated object track. If possible, this track will be updated by single cycle state estimates in the following measurement cycles, otherwise its position and velocity will be predicted according to the track's latest kinematic state. Such a linear prediction helps to avoid track discontinuities due to missed detections. The updating process of a tracking filter takes the accuracy of its input data as well as the uncertainty caused by the target dynamics into account. Thus, the tracking filter smoothes measurement errors without ignoring the object's dynamics, so that the kinematic state estimation performance of the overall system will be significantly enhanced. As the last postprocessing step, all detected objects must be sorted to an application dependent criterion. It is easy to understand that safety applications are more interested in objects that will have a short time-to-impact. Convenience applications like parking aid favor the object's with the shortest distance to the car. Hence, an application dependent object selection must be performed before the estimated object data is passed to an application processor that decides what kind of passive or active actions have to be taken.

The following chapters will outline and evaluate different postprocessing algorithms for an automotive radar network, which are finally implemented in a multi-target capable postprocessing architecture of an experimental car. The next chapter will give an overview of kinematic state estimation algorithms that can handle different sets of radar measurement data. It will also address the need of a tracking filter and discuss different tracking algorithms based on the general discrete Kalman filter principle. At this, it will present two non-linear Kalman filter approaches where the non-linear kinematic state estimation and the tracking filter are fused to one postprocessing step. Furthermore, different models of the object dynamics will be derived and their suitability for automotive applications will be discussed. Finally, different Kalman filter processing enhancements, either in computation complexity or in estimation performance, will be outlined. Chapter 3 will present a complete postprocessing architecture utilizing a special data association technique in order to enable the application of the presented kinematic state estimation

and tracking algorithms in multi-target road traffic scenarios. Chapter 4 will present the implementation of the presented postprocessing architecture in the experimental car and detailed experimental results in different real road traffic scenarios will be discussed. Finally, chapter 5 will conclude this thesis and will give an outlook on future research in the field of automotive radar applications.



## Chapter 2

# Kinematic State Estimation

This chapter will investigate different kinematic state estimation methods that can be applied in a multi-sensor radar network. These methods refer typically to the task of estimating the position, velocity, and acceleration of an object on the basis of either the actual radar sensor measurements or the actual and the past radar sensor measurements. All kinematic state estimation algorithms that will be presented are based on the geometric relation between the single-sensor measurements and the unknown kinematic states. Thus, this chapter begins with a brief description of the underlying measurement equations. This is followed by an introduction to the so-called batch processing approaches, where the kinematic states are estimated from the sensor measurements of a limited measurement time, e.g. a single measurement cycle. The presented type of this batch kinematic state estimation will be based on the principles of least squares estimation (LSE). Due to the possible detection of ghost targets, or missed detections of real targets in a single measurement cycle, the estimated kinematic states must be processed by a tracking filter that helps to avoid these missed detections, smoothes measurement errors, and enhances the time-depending performance of the system. This chapter will present tracking algorithms that are different derivations of the Kalman filter algorithm. All Kalman filter derivations offer the advantage of incorporating assumptions on the target dynamics in order to achieve an optimal estimation. Thus, different target dynamic models will also be introduced and their suitability for automotive radar applications will be discussed. An alternative to the kinematic state estimation plus linear tracking filter approach are the extended Kalman filter (EKF) and the unscented Kalman filter (UKF), where the estimation and the tracking step are fused to a single postprocessing stage. The performance of the two nonlinear tracking filters will be compared to the one of the combined least-squares kinematic state estimation plus linear Kalman filter approach. Finally, different Kalman filter signal processing enhancements either in computation complexity or in estimation performance will be outlined.

## 2.1 Position and Velocity Estimation

In the field of land surveying, two traditional techniques are utilized for the position determination of an object, the trilateration and the triangulation described in [Bau98], [Bau99]. As shown in figure 2.1(a) the position of a target can be determined by two range measurements, i.e. by intersecting the corresponding two circular arcs (trilateration). An analogous position determination, the so-called triangulation, can be achieved by two measured angles (figure 2.1(b)). At this, two straight lines defined by the measured angles are intersected and determine the wanted position. These simple methods are based on the solution of a set of nonlinear measurement equations. The two states, the object position  $[t_x, t_y]^T$  in the (x,y)-coordinate plane, are calculated from two measurements, either two ranges  $[r_1, r_2]^T$  or two angles  $[\varphi_1, \varphi_2]^T$ . The resulting set of measurement equations has a non-ambiguous solution. If the contributing measurements contain a certain error, than the non-ambiguous solution is error-prone as well.

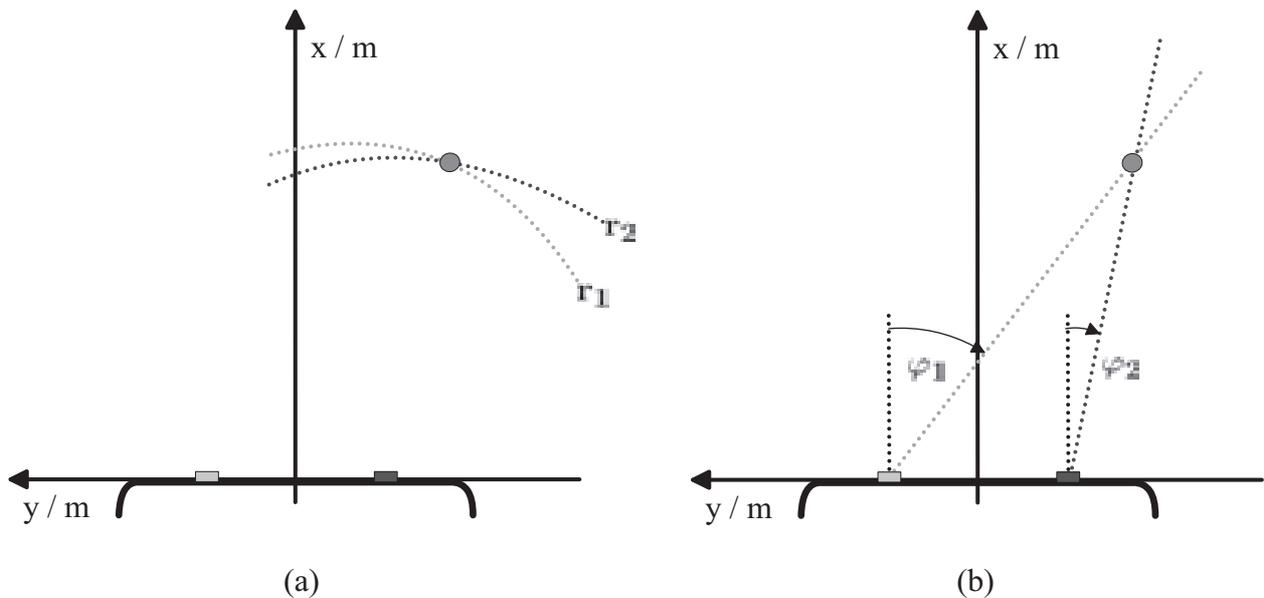


Figure 2.1: Geometrical calculation of an object's position by a trilateration (a) and a triangulation (b)

To achieve a high quality estimate of the unknown kinematic states of an object, it is advantageous that more measurements than unknown kinematic states are available. This extended number of measurements leads to an over-determined measurement equation system that can be solved by a least squares estimation approach. Such a kinematic state estimation approach requires a certain number of radar sensors that provide the estimation algorithm with an appropriate number of measurement values. In an ideal measurement case, it must be guaranteed that all measurements are taken at the same point in time and that they originate from the same reflection center on the extended

object's surface. These side conditions will only approximately be met in reality.

Before introducing different kinematic state estimation algorithms, the basic geometric relations between the available radar sensor measurements and the unknown kinematic states in the so-called measurement equations will be outlined in the following section.

### 2.1.1 Geometric Preliminaries

The real kinematic state  $\mathbf{t}$  of a target in an (x,y)-coordinate plane is defined by

$$\mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_{v_x} \\ t_{v_y} \\ t_{a_x} \\ t_{a_y} \end{bmatrix} \quad (2.1)$$

These target states are unknown and want to be estimated from the different single-sensor measurements contained in the vector  $\mathbf{m}$ . The geometric relation between the real target states and these measurements can be described by a function  $h(\mathbf{t})$  and an additive error vector  $\mathbf{e}$

$$\mathbf{m} = h(\mathbf{t}) + \mathbf{e} \quad (2.2)$$

The vector  $\mathbf{m}$  contains the measurements of the  $N$  different sensors

$$\mathbf{m} = \begin{bmatrix} \mathbf{m}^{\text{Sens } 1} \\ \mathbf{m}^{\text{Sens } 2} \\ \vdots \\ \mathbf{m}^{\text{Sens } N} \end{bmatrix} \quad (2.3)$$

At this, each single-sensor measurement vector  $\mathbf{m}^{\text{Sens } n}$  may contain only a single (e.g. HRR sensor) or multiple measurement values (e.g. UMR sensor). The additive error  $\mathbf{e}$  consists of sensor-dependent elements that represent the limited measurement accuracy of each radar sensor. The measurements in  $\mathbf{m}$  are assumed to be unbiased

$$E[\mathbf{e}] = 0 \quad (2.4)$$

and their uncertainty is described by the measurement noise covariance matrix  $R$  as given by

$$R = E[\mathbf{e}\mathbf{e}^T] \quad (2.5)$$

As an example, the geometric relation between the ideal, error-free measurements of the  $n$ -th radar sensor

$$\mathbf{t}^{\text{Sens } n} = [t_r^{\text{Sens } n}, t_\varphi^{\text{Sens } n}, t_{v_r}^{\text{Sens } n}]^T \quad (2.6)$$

and the unknown target states in the vector  $\mathbf{t}$  is given by the following set of nonlinear equations

$$t_\varphi^{\text{Sens } n} = h_\varphi(\mathbf{t}, \mathbf{s}^{\text{Sens } n}) = \arctan\left(\frac{s_y^{\text{Sens } n} - t_y}{s_x^{\text{Sens } n} - t_x}\right) \quad (2.7)$$

$$t_r^{\text{Sens } n} = h_r(\mathbf{t}, \mathbf{s}^{\text{Sens } n}) = \sqrt{(t_x - s_x^{\text{Sens } n})^2 + (t_y - s_y^{\text{Sens } n})^2} \quad (2.8)$$

$$t_{v_r}^{\text{Sens } n} = h_{v_r}(\mathbf{t}, \mathbf{s}^{\text{Sens } n}) = \frac{t_{v_x} \cdot (t_x - s_x^{\text{Sens } n}) + t_{v_y} \cdot (t_y - s_y^{\text{Sens } n})}{\sqrt{(t_x - s_x^{\text{Sens } n})^2 + (t_y - s_y^{\text{Sens } n})^2}} \quad (2.9)$$

The radar network itself is described by the sensor placement in the front bumper. The position of the  $n$ -th sensor is given by

$$\mathbf{s}^{\text{Sens } n} = \begin{bmatrix} s_x^{\text{Sens } n} \\ s_y^{\text{Sens } n} \end{bmatrix} \quad (2.10)$$

### 2.1.2 Kinematic State Determination from Single-Sensor Measurements

In the case of an UMRR sensor, the single-sensor measurements comprise the position and partly the velocity of the detected target. This kinematic state information is measured in polar coordinates

$$\mathbf{m}_{(r,\varphi)}^{\text{UMRR}} = [r, \varphi, v_r]^T \quad (2.11)$$

The velocity in the lateral direction as well as the object acceleration cannot directly be measured and thus are not available. Nevertheless, the available kinematic state information from one UMRR sensor offers the possibility of an implementation as a stand-alone system in a distributed radar sensor network. At this, each UMRR sensor is delivering its own target list that is later fused to a global object list by a central postprocessing unit. In order to simplify the geometric relation between the different single-sensor measurements contained in the measurement vector  $\mathbf{m}^{\text{UMRR}}$  and the unknown kinematic target state vector  $\mathbf{t}$ , it is reasonable to transform the measurements from each sensor to a global cartesian coordinate system. The relation between these two coordinate systems is visualized in figure 2.2.

The transformation basically consists of two steps:

- the transformation of the sensor coordinate system from polar to cartesian coordinates,
- and the linear transformation from the sensor to the global kinematic state coordinate system by taking the position and azimuthal orientation of the sensor relative to the global coordinate system into account.

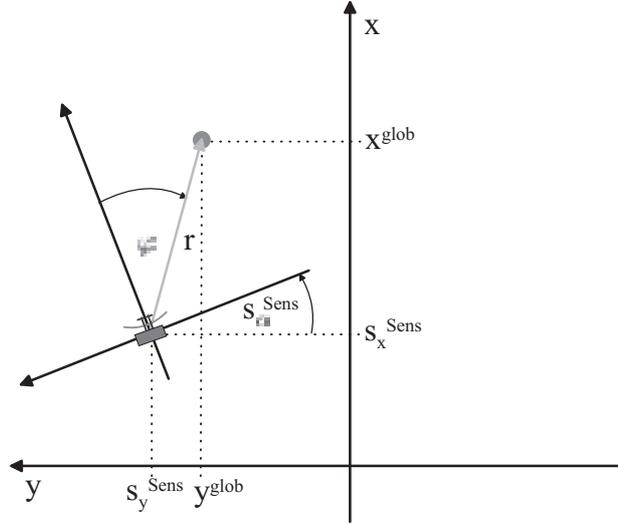


Figure 2.2: Relation between the polar sensor and the global cartesian coordinate system

The UMRR measurement values in cartesian coordinates can directly be determined by the following measurement equations

$$\begin{aligned}
 x &= r \cdot \cos(\varphi) \\
 y &= r \cdot \sin(\varphi) \\
 v_x &= v_r \cdot \cos(\varphi) + r \cdot \omega \cdot \sin(\varphi) \\
 v_y &= v_r \cdot \sin(\varphi) + r \cdot \omega \cdot \cos(\varphi)
 \end{aligned} \tag{2.12}$$

so that the transformed UMRR measurements are given by

$$\mathbf{m}_{(x,y)}^{\text{UMRR}} = [x, y, v_x, v_y]^T \tag{2.13}$$

As already mentioned, the complete target velocity information cannot be retrieved from the UMRR sensor measurements. Only the radial component of the target velocity is measured. Thus, the radial velocity information can only be split into its x- and y-components by incorporating the measured azimuth angular information. The orthogonal component of the object's velocity, i.e.  $r \cdot \omega$ , is still unknown and can only be obtained by combining measurements of different sensors, e.g. in the central postprocessing unit of the multi-sensor radar network. By arbitrarily setting the turning rate to zero ( $\omega = 0$ ), the velocity measurements in cartesian coordinates contain a certain error. This error must be taken into account in the covariance matrix of the transformed UMRR measurements.

The measurement covariance matrix  $R_{(x,y)}$  in cartesian coordinates can be derived from the original covariance matrix  $R_{(r,\varphi)}$  by a simple transformation

$$R_{(x,y)} = J R_{(r,\varphi)} J^T \tag{2.14}$$

with the transformation matrix

$$J = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \varphi} & \frac{\partial x}{\partial v_r} & \frac{\partial x}{\partial \omega} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \varphi} & \frac{\partial y}{\partial v_r} & \frac{\partial y}{\partial \omega} \\ \frac{\partial v_x}{\partial r} & \frac{\partial v_x}{\partial \varphi} & \frac{\partial v_x}{\partial v_r} & \frac{\partial v_x}{\partial \omega} \\ \frac{\partial v_y}{\partial r} & \frac{\partial v_y}{\partial \varphi} & \frac{\partial v_y}{\partial v_r} & \frac{\partial v_y}{\partial \omega} \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & -r \cdot \sin(\varphi) & 0 & 0 \\ \sin(\varphi) & r \cdot \cos(\varphi) & 0 & 0 \\ \omega \cdot \sin(\varphi) & -v_r \cdot \sin(\varphi) & \cos(\varphi) & r \cdot \sin(\varphi) \\ \omega \cdot \cos(\varphi) & v_r \cdot \cos(\varphi) & \sin(\varphi) & r \cdot \cos(\varphi) \end{bmatrix} \quad (2.15)$$

where  $r$ ,  $\varphi$ , and  $v_r$  are the actual UMRR sensor measurements. The measurement covariance matrix  $R_{(r,\varphi)}$  is given by

$$R_{(r,\varphi)} = \begin{bmatrix} \sigma_r^2 & 0 & 0 & 0 \\ 0 & \sigma_\varphi^2 & 0 & 0 \\ 0 & 0 & \sigma_{v_r}^2 & 0 \\ 0 & 0 & 0 & \sigma_\omega^2 \end{bmatrix} \quad (2.16)$$

where  $\sigma_r$ ,  $\sigma_\varphi$ , and  $\sigma_{v_r}$  are the measurement accuracies of the sensor. The variance of  $\omega$  is originally not part of the measurement noise covariance matrix  $R$ , but must be taken into account if the turning rate is assumed to be zero ( $\omega = 0$ ). It represents the possible error introduced by this assumption. If the maximum occurring turning rate is  $\omega_{max}$ , then the variance must be set to

$$\sigma_\omega^2 = \omega_{max}^2 \quad (2.17)$$

Since  $\omega_{max}$  is strongly depending on the targeted automotive application, it must be empirically adjusted in real road traffic scenarios.

The *second step of the transformation* from polar sensor to global cartesian coordinates is linear. The azimuthal orientation of the radar sensor  $s_\alpha^{\text{Sens}}$  and its relative position  $[s_x^{\text{Sens}}, s_y^{\text{Sens}}]^T$  to the origin of the global coordinate system are given by

$$\mathbf{s}^{\text{Sens}} = \begin{bmatrix} s_x^{\text{Sens}} \\ s_y^{\text{Sens}} \\ s_\alpha^{\text{Sens}} \end{bmatrix} \quad (2.18)$$

These parameters lead to the following linear transformation

$$\mathbf{m}_{(x,y)}^{\text{glob}} = T \cdot \mathbf{m}_{(x,y)}^{\text{UMRR}} + \begin{bmatrix} s_x^{\text{Sens}} \\ s_y^{\text{Sens}} \\ 0 \\ 0 \end{bmatrix} \quad (2.19)$$

where  $\mathbf{m}_{(x,y)}^{\text{glob}}$  is the measurement vector in global cartesian coordinates. The transformation is given by the  $4 \times 4$  turning matrix

$$T = \begin{bmatrix} \cos(s_\alpha^{\text{Sens}}) & -\sin(s_\alpha^{\text{Sens}}) & 0 & 0 \\ \sin(s_\alpha^{\text{Sens}}) & \cos(s_\alpha^{\text{Sens}}) & 0 & 0 \\ 0 & 0 & \cos(s_\alpha^{\text{Sens}}) & -\sin(s_\alpha^{\text{Sens}}) \\ 0 & 0 & \sin(s_\alpha^{\text{Sens}}) & \cos(s_\alpha^{\text{Sens}}) \end{bmatrix} \quad (2.20)$$

The appropriate covariance matrix  $P_{(x,y)}$  is obtained by

$$P_{(x,y)} = T \cdot R_{(x,y)} \cdot T^T \quad (2.21)$$

Having transformed the UMRR measurements to a global cartesian coordinate system, the measurement equation eq.(2.2) has become linear

$$\mathbf{m}_{(x,y)}^{\text{glob}} = \mathbf{t} + \mathbf{e} \quad (2.22)$$

so that the UMRR sensors can be utilized as stand-alone systems in a distributed radar network, for example. All measurements have been transformed to the same global coordinate system, so that the following postprocessing stages have a common data basis.

### 2.1.3 Linear Least Squares Estimation from Multiple Sensor Measurements

This section will present an approach for determining an object's kinematic state from different single-sensor measurements in global cartesian coordinates. It will be assumed that all measurement data serving as the input of the algorithm is originating from one reflection center of an object and that the kinematic state of this object is constant during the observation time. Furthermore, a linear relationship between the  $L$ -dimensional unknown kinematic state vector  $\mathbf{t}$  and the available  $M$ -dimensional measurement vector  $\mathbf{m}$  is assumed

$$\mathbf{m} = H\mathbf{t} + \mathbf{e} \quad (2.23)$$

where  $H$  is an  $M \times L$  measurement matrix. This system of linear equations can only be solved if the number of measurements  $M$  is greater or equal the number of unknown states  $L$  ( $M \geq L$ ). The criterion of such a *linear least squares estimation* can be written as

$$\begin{aligned} \mathbf{e} &= \mathbf{m} - H\mathbf{t} \\ \mathbf{e}^T \mathbf{e} &\stackrel{!}{=} \min \end{aligned} \quad (2.24)$$

and leads to the 'normal equations' of the least squares problem

$$(H^T H) \hat{\mathbf{t}}_{LS} = H^T \mathbf{m} \quad (2.25)$$

with the least squares estimate  $\hat{\mathbf{t}}_{LS}$  of the unknown vector  $\mathbf{t}$  given by

$$\hat{\mathbf{t}}_{LS} = (H^T H)^{-1} H^T \mathbf{m} \quad (2.26)$$

If the measurements  $\mathbf{m}$  are unbiased, then the least squares estimate  $\hat{\mathbf{t}}_{LS}$  is also unbiased

$$E[\hat{\mathbf{t}}_{LS}] = \mathbf{t} \quad (2.27)$$

Its covariance matrix  $P_{LS}$  can be calculated by

$$P_{LS} = E [(\hat{\mathbf{t}}_{LS} - E[\hat{\mathbf{t}}_{LS}])(\hat{\mathbf{t}}_{LS} - E[\hat{\mathbf{t}}_{LS}])^T] = (H^T H)^{-1} \quad (2.28)$$

These basic least squares calculations were first introduced by C.F.Gauß in order to determine the orbits of planets.

As an extension of this basic least squares estimation, the different measurements can be weighted by an  $m \times m$  weight matrix  $W$ . This so-called *weighted least squares approach* changes the least squares criterion in eq.(2.24) to

$$\mathbf{e}^T W \mathbf{e} \stackrel{!}{=} \min \quad (2.29)$$

The weighted least squares estimate  $\hat{\mathbf{t}}_{WLS}$  is then given by

$$\hat{\mathbf{t}}_{WLS} = (H^T W H)^{-1} H^T W \mathbf{m} \quad (2.30)$$

The estimation covariance matrix  $P_{WLS}$  is given by

$$P_{WLS} = (H^T W H)^{-1} \quad (2.31)$$

The estimate  $\hat{\mathbf{t}}_{WLS}$  is unbiased in the sense that

$$E [\hat{\mathbf{t}}_{WLS}] = E [\mathbf{t}] \quad (2.32)$$

The choice of the optimal weight matrix  $W$  is discussed in several publications, e.g. [CC98] or [BP99]. It is given by the inverse of the measurement covariance matrix  $R = E[ee^T]$

$$W = R^{-1} \quad (2.33)$$

In [CC98]  $\hat{\mathbf{t}}_{WLS}$  is called the least squares optimal estimate of  $\mathbf{t}$ , because it can be shown that  $R^{-1}$  is the optimal weight for all positive definite symmetric matrices  $W$  in the sense that

$$\text{var}(\mathbf{t} - \hat{\mathbf{t}}_{WLS}(R^{-1})) \leq \text{var}(\mathbf{t} - \hat{\mathbf{t}}_{WLS}(W)) \quad (2.34)$$

This means that  $\hat{\mathbf{t}}_{WLS}$  is the *minimum variance estimate* of  $\mathbf{t}$ . Generally, the direct computation of the kinematic state estimation  $\hat{\mathbf{t}}$  according to eq.(2.26) and (2.30) cannot be recommended, because this calculation is prone to rounding errors. Instead, the computation of the least squares estimate  $\hat{\mathbf{t}}$  can be achieved by a QR-decomposition as described in [MV93] or [Vol02].

### 2.1.4 Nonlinear Least Squares Estimation from Multiple Sensor Measurements

The linear least squares estimation approach described in the previous section can be extended to nonlinear measurement equations as given by

$$\mathbf{m} = h(\mathbf{t}) + \mathbf{e} \quad (2.35)$$

$h(\mathbf{t})$  is a vector of nonlinear functions, e.g. if the single-sensor measurements are given in the sensor's polar coordinate system and the unknown kinematic state vector is chosen

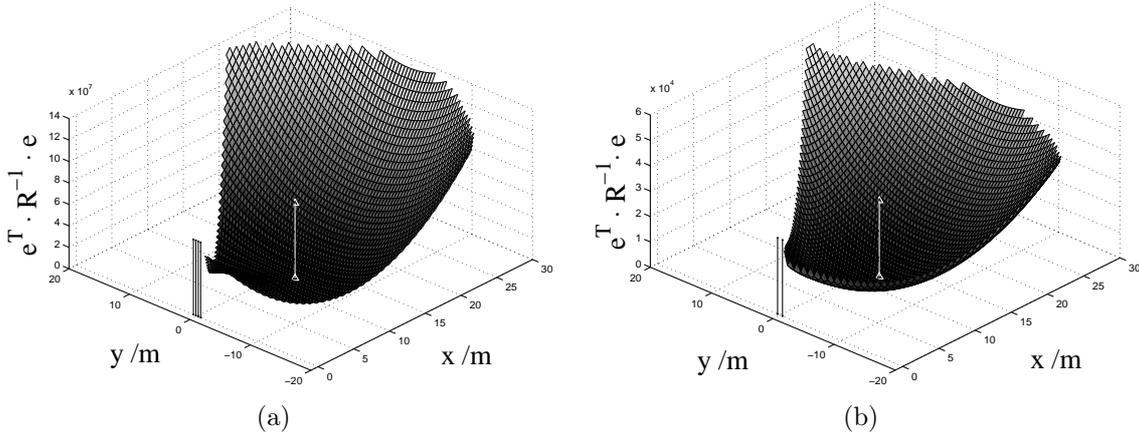


Figure 2.3: Example of the square error function  $f_{\text{error}}(\mathbf{t}, \mathbf{m})$  for a target at the position  $[t_x, t_y]^T = [12, -2]^T$  m of the 4 HRR pulse radar network (a) and the 2 UMRR network (b)

to be in global cartesian coordinates. The criterion for a least squares estimation is the same one as defined in eq.(2.29). It can be rewritten for the optimal weight  $W = R^{-1}$ , so that the resulting square error function that has to be minimized is given by

$$f_{\text{error}}(\mathbf{t}, \mathbf{m}) = [\mathbf{m} - h(\mathbf{t})]^T R^{-1} [\mathbf{m} - h(\mathbf{t})] \stackrel{!}{=} \min \quad (2.36)$$

In the figures 2.3(a) and (b) this square error function is exemplarily visualized for a target at the position  $[t_x, t_y]^T = [12, -2]^T$  m. Figure 2.3(a) shows  $f_{\text{error}}(\mathbf{t}, \mathbf{m})$  for a HRR pulse radar network, which is consisting of four only range measuring sensors. Figure 2.3(b) shows the appropriate square error function of a 2 UMRR sensor network. The sensor positions are denoted in both figures by the green lines and the real target position  $[t_x, t_y]^T = [12, -2]^T$  m by the red line. To retrieve a solution of the nonlinear weighted least squares criterion in eq.(2.36), the minimum  $\hat{\mathbf{t}}$  of such a square error function has to be found. As expected, only one minimum of the function  $f_{\text{error}}(\mathbf{t}, \mathbf{m})$  is existing in the scan area of the radar networks. This minimum is the solution of the over-determined set of nonlinear measurement equations and can be determined by applying an iterative algorithm called Gauß-Newton method [Voß02]. It is based on a first-order approximation of the nonlinear measurement function  $h(\mathbf{t})$  and will be outlined in the following:

### Iterative Algorithm:

In each iterative algorithm, a start value for the iteration process has to be chosen first. In this iterative algorithm, the choice of the start value has proven to be uncritical for the scan area of the considered multi-sensor networks. The target is expected to have a position with  $t_x > 0$  within the radar network's scan area, so that the initial kinematic state value of this iterative process can be set to

$$\hat{\mathbf{t}}_0 = [t_{x0}, 0, 0, 0]^T \text{ with } t_{x0} > 0 \quad (2.37)$$

One iteration step consists of the following calculations:

The approximate solution  $\hat{\mathbf{t}}_{n+1}$  of the over-determined system of nonlinear equations at step  $n + 1$  can be calculated by correcting a former solution  $\hat{\mathbf{t}}_n$  with a term  $\Delta\hat{\mathbf{t}}_{n+1}$

$$\hat{\mathbf{t}}_{n+1} = \hat{\mathbf{t}}_n + \Delta\hat{\mathbf{t}}_{n+1} \quad (2.38)$$

The correction term  $\Delta\hat{\mathbf{t}}_{n+1}$  is calculated in a weighted least squares sense

$$\Delta\hat{\mathbf{t}}_{n+1} = PH^T R^{-1} \Delta\mathbf{m} \quad (2.39)$$

The covariance matrix  $P$  is calculated in each step as follows

$$P = (H^T R^{-1} H)^{-1}; \quad (2.40)$$

$R$  is the measurement noise covariance matrix that is utilized for weighting each measurement with the inverse of the appropriate sensor accuracy.  $\Delta\mathbf{m}$  is a vector of innovations between the single-sensor measurements  $\mathbf{m}$  and the measurements  $\hat{\mathbf{m}}_n$  expected from the approximate solution  $\hat{\mathbf{t}}_n$  at iteration step  $n$ .

$$\Delta\mathbf{m} = \mathbf{m} - \hat{\mathbf{m}}_n \quad (2.41)$$

$H$  is the matrix of partial derivatives (the so-called Jacobi matrix) corresponding to the appropriate nonlinear measurement equations as given in eq.(2.7)-(2.9) evaluated at the approximate position  $\mathbf{t} = \hat{\mathbf{t}}_n$

$$H = \left[ \begin{array}{cccc} \frac{\partial \mathbf{m}}{\partial t_x} & \frac{\partial \mathbf{m}}{\partial t_y} & \frac{\partial \mathbf{m}}{\partial t_{vx}} & \frac{\partial \mathbf{m}}{\partial t_{vy}} \end{array} \right]_{t=\hat{\mathbf{t}}_n} \quad (2.42)$$

These calculation are repeated until a stop criterion is fulfilled. This stop criterion can be chosen to be either a maximum number of iteration steps or an arbitrarily chosen threshold  $\epsilon$ , so that the sum of square errors

$$SOS = \Delta\hat{\mathbf{t}}_{n+1}^T \cdot \Delta\hat{\mathbf{t}}_{n+1} \quad (2.43)$$

computed in each iteration step is fulfilling

$$SOS \leq \epsilon \quad (2.44)$$

The significance of the correction  $\Delta\hat{\mathbf{t}}_{n+1}$  in the last step  $n + 1$  can be tested by calculating the normalized square distance

$$d_{n+1}^2 = \Delta\mathbf{t}_{n+1}^T P^{-1} \Delta\mathbf{t}_{n+1} \quad (2.45)$$

A small value for  $d_{n+1}^2$  (less than about 0.01) is a clear indication that convergence has occurred.

The Gauss-Newton method can be applied for the kinematic state estimation in each of the radar network types presented in 1.4. Assuming the geometric preliminaries in 2.1.1, the estimated kinematic state  $\hat{\mathbf{t}}$  is unbiased. It also gives a minimum variance estimate of  $\mathbf{t}$  under the side conditions described for the linear least squares approach in the previous section.

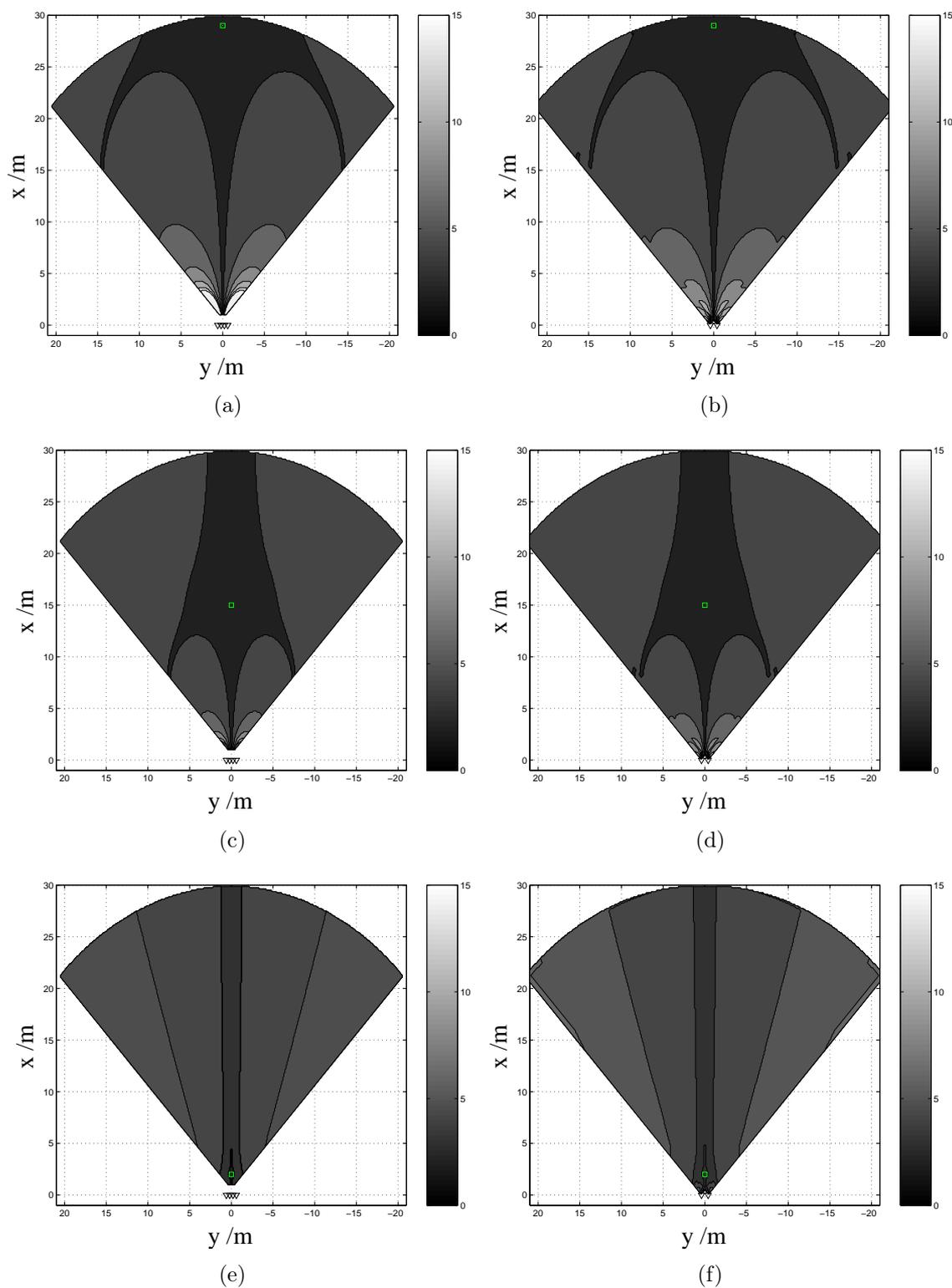


Figure 2.4: Number of iterations until convergence occurs for different start values utilized in a 4 HRR pulse radar network (a), (c), (e) and a 2 UMRR sensor network (b), (d), (f).

	$\hat{\mathbf{t}}_{01}$		$\hat{\mathbf{t}}_{02}$		$\hat{\mathbf{t}}_{03}$	
	$\bar{n}_{iterat}$		$\bar{n}_{iterat}$		$\bar{n}_{iterat}$	
Radar Network 1 (4 HRR)	4.25	78.9 %	3.92	79.7 %	4.35	79.5 %
Radar Network 2 (2 UMRR)	4.16	99.8 %	3.89	99.9 %	4.36	99.6 %

Table 2.1: Number of iterations until convergence occurs and estimation percentage for the condition given in eq.(2.51)

Different enhancements of the presented iterative nonlinear least squares approach have been investigated in [Voß02] and [Föl01]. In these publications, the introduction of a so-called damp factor has been suggested in order to achieve a faster convergence. This enhancement of the Gauss-Newton method is called damped Newton method. If the measurement matrix  $H$  is badly conditioned, then different regularization methods can be applied. But the author's experience with the presented Gauss-Newton method in automotive radar network applications leads to the conclusion that these enhancements are not necessary to achieve convergence or to avoid wrong estimates. The mentioned enhancements should be considered in more critical applications.

### Choice of Start Value

As already stated in the last section, the choice of the iterative algorithm's start value is uncritical for automotive radar network applications. Convergence is guaranteed if the initial value of  $\hat{\mathbf{t}}_0$  is chosen to be somewhere in the radar network's detection area. This statement should be exemplarily proven in the following. Therefore, the iteration depth necessary for convergence will be investigated for three different initial values of  $\hat{\mathbf{t}}_{0i}$  ( $i = 1, \dots, 3$ ):

$$\hat{\mathbf{t}}_{01} = [29, 0, 0, 0]^T \text{ m} \quad (2.46)$$

$$\hat{\mathbf{t}}_{02} = [15, 0, 0, 0]^T \text{ m} \quad (2.47)$$

$$\hat{\mathbf{t}}_{03} = [2, 0, 0, 0]^T \text{ m} \quad (2.48)$$

Again, the 4 HRR pulse radar network and the 2 UMRR network are compared in this example. The single sensor measurement values have been generated with the appropriate accuracies as described in section 1.4.2 in eq.(1.29) and (1.32). The figures 2.4(a)-(f) show the necessary number of iterations to achieve convergence in the iterative least squares estimation for each point of the radar networks' scanning areas. The start value of the iteration is visualized by a green square. The different colors in the scan area refer to the iteration depth necessary for convergence of a detected target at the appropriate position.

In the case of the 4 HRR pulse radar network, the number of iterations necessary for

convergence are shown in the figures 2.4(a), 2.4(c), and 2.4(e). It can be seen that the necessary iteration depth is not greater than 15, in fact, the average number of iterations  $\bar{n}_{iterat}$  considering one complete scanning area lies within

$$3.92 \leq \bar{n}_{iterat} \leq 4.25 \quad (2.49)$$

The appropriate iteration depth charts for the 2 UMRR sensor network are shown in the figures 2.4(b), 2.4(d), and 2.4(f). The achieved results are similar to the ones of the 4 HRR pulse radar network: The maximum number of necessary iterations is not exceeding 15. The average number of iterations  $\bar{n}_{iterat}$  for the three different initial values  $\hat{\mathbf{t}}_{0i}$  ( $i = 1, \dots, 3$ ) considering one complete scan area lies within the range

$$3.89 \leq \bar{n}_{iterat} \leq 4.36 \quad (2.50)$$

Until now, only the number of iteration until convergence occurs have been considered. But it may occur that convergence is achieved and the error of the estimated kinematic state  $\hat{\mathbf{t}} - \mathbf{t}$  is too large to consider this result as correct or precise enough for a target detection. Therefore, a maximum error is defined, so that all estimates fulfilling

$$\sqrt{(\hat{\mathbf{t}} - \mathbf{t})^T (\hat{\mathbf{t}} - \mathbf{t})} \leq 1 \text{ m} \quad (2.51)$$

are considered as correct target estimates. The percentage of correct estimations according to eq.(2.51) has been investigated by a Monte-Carlo simulation with 1000 simulations for each of the three start values in both radar networks. The results show a significant difference between the 4 HRR pulse radar network and the 2 UMRR network and are summarized in table 2.1. The ILSE algorithm applied in a 2 UMRR network is correctly converging with almost a perfect 100 %, whereas the correct estimate percentage in a 4 HRR radar network is reducing to 80 %. This result verifies the outcome of the radar network concept comparison in section 1.4 of the last chapter, where the 2 UMRR sensor network showed the better estimation performance due to the availability of complementary single-sensor measurements, such as range and azimuth angle.

Overall, these examples of different initial values of the iterative kinematic state estimation have shown that convergence is occurring for almost every initial value inside the radar network's scanning area. But it has also proven that convergence of the iterative kinematic state estimation does not mean that a reasonable kinematic state estimate is determined. At this, the 2 UMRR network has verified to have a better kinematic state estimation performance than the 4 HRR pulse radar network due to its complementary radar sensor measurements.

### 2.1.5 Incorporation of Host Vehicle Information

The previous sections have dealt with determining the relative kinematic state between the detected object and the host vehicle in global cartesian coordinates by either directly transforming the sensor measurements to this coordinate system or by combining different single-sensor measurements in a least squares sense. Since this relative kinematic

state information comprises the dynamics of two objects, it is reasonable to extract the absolute kinematic state of the detected object by incorporating the available host vehicle information. The information about the own vehicles motion is available on the powertrain CAN bus of today's vehicles and comprises the yaw-rate  $m_\omega^{\text{Host}}(k)$  and the vehicle speed  $m_v^{\text{Host}}(k)$ .

$$\mathbf{m}^{\text{Host}} = \begin{bmatrix} m_v^{\text{Host}} \\ m_\omega^{\text{Host}} \end{bmatrix} \quad (2.52)$$

Both are visualized in figure 2.5. The global cartesian coordinate system has been chosen in a way that the turning point of the host vehicle is the origin of coordinates. Such a choice simplifies the following calculations, so that the absolute kinematic state of the detected object is given by

$$\mathbf{t}^{\text{abs}}(k) = \mathbf{t}^{\text{rel}}(k) - D(k)\mathbf{m}^{\text{Host}} = \begin{bmatrix} t_x^{\text{rel}}(k) \\ t_y^{\text{rel}}(k) \\ t_{v_x}^{\text{rel}}(k) \\ t_{v_y}^{\text{rel}}(k) \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & r(k) \end{bmatrix} \cdot \begin{bmatrix} m_v^{\text{Host}} \\ m_\omega^{\text{Host}} \end{bmatrix} \quad (2.53)$$

with

$$r(k) = \sqrt{(t_x^{\text{rel}}(k))^2 + (t_y^{\text{rel}}(k))^2} \quad (2.54)$$

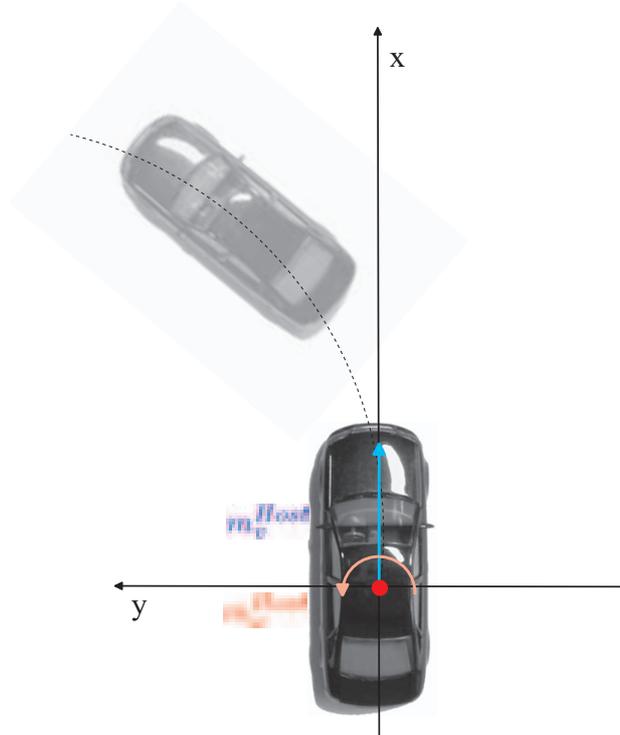


Figure 2.5: Available information about the host vehicle's kinematic state ( yaw-rate  $m_\omega^{\text{Host}}(k)$  and velocity  $m_v^{\text{Host}}(k)$ )

Since the yaw-rate  $m_{\omega}^{\text{Host}}(k)$  and the speed  $m_v^{\text{Host}}(k)$  of the host vehicle are also measured by sensors with a limited accuracy, the covariance matrix of the object's absolute kinematic state consists of two parts, the covariance matrix of the relative kinematic state estimate  $P^{\text{rel}}(k)$  and the one corresponding to the host vehicle measurements:

$$P^{\text{abs}}(k) = P^{\text{rel}}(k) + D(k)R^{\text{Host}}D(k)^T \quad (2.55)$$

The obtained absolute kinematic state of a detected object serves as the input of the tracking filter described in the next section. Because such a tracking filter makes certain assumptions on the absolute object dynamics, it is reasonable to feed it with the absolute kinematic states of an object.

### 2.1.6 Summary

The previous sections have dealt with the kinematic state estimation as a batch processing step. This means the estimates have been retrieved during a limited measurement time where the unknown kinematic state of a detected object remained unchanged. At this, two approaches have been described, the kinematic state determination from single-sensor measurements and a least squares estimation approach utilizing all available measurements corresponding to one reflection center of a detected object. Since the dynamic behavior of an object over time plays an important role in the kinematic state estimation, the following section will outline tracking algorithms, especially the Kalman filter approach. At this, it will be focussed on the 2 UMR sensor network. The reason for leaving out the 4 HRR pulse radar network in further investigations is the following:

As already stated in the radar network comparison in section 1.4, a limitation in the position estimation accuracy occurs if the aspect angles of the different sensors are almost similar and only one type of measurements is available. These side conditions meet exactly the measurement situation of a 4 HRR pulse radar network. Only range measurements are available for the position estimation and all sensors are mounted in the vehicle's front bumper, whose width is small compared to the covered range. In 1.4, it has been shown that the azimuth angle determination with four extremely precise range measurements still leads to a not negligible azimuth angle error in the estimated target position. Such an observation has also been made with an analogous measurement constellation in the research field of satellite navigation. The more the aspect angle of the satellites relative to the GPS receiver differ from each other, the more accurate is the resulting position estimate. Therefore, a quality measure for an arbitrary sensor-target constellation has been developed and has been named dilution of precision (DOP). It describes the quality of the estimated GPS receiver position determined by the pseudo measurements of the available satellites. Different DOP calculations have been defined in the literature, e.g. [PS96], and all have in common, that a large DOP value corresponds to a bad accuracy condition due to similar aspect angles of the different sensors. Exactly such a large DOP value is the outcome for almost the complete scan area of the 4 HRR pulse radar network. It can be explained by the limited extension of vehicle's front bumper, which is relative small compared to the covered scan distance of 30 or 50 m. The DOP of such a measurement constellation cannot be avoided in the case of

automotive radar networks with only one type of position measurements. This leads to the conclusion that complementary measurements, such as range and azimuth angle, are necessary to achieve reliable, precise position estimates. As a consequence, the following investigation of postprocessing algorithms will focus on the radar network concept with 2 UMRR sensors. This 2 UMRR sensor network and an adequate designed postprocessing architecture will be explored in detail.

## 2.2 Target Tracking

In the previous section, different methods of determining a detected object's kinematic state have been outlined. These methods have estimated the unknown kinematic state on the basis of the measurements of one measurement cycle. Measurements of multiple cycles could only be utilized under the assumption that the kinematic state of the object is constant during the observation time. The task of a tracking filter is to extend such an estimated "snapshot" of a dynamic scenario by taking the time-varying object behavior and measurement effects into account. The tracking filter smoothes kinematic state estimation errors and helps to avoid missed detections. It incorporates the kinematic state estimation task of the least squares estimator in its filter equations as well as a model of the observed object's dynamics. The most popular tracking filter approach is the Kalman filter named after its inventor R.E. Kalman [Kal60] and will be described in detail in the following. Different derivatives of this basic filter concept as well as different Kalman filter enhancements, either in tracking performance or computation complexity, will also be presented.

### 2.2.1 Kalman Filtering

The goal of the Kalman filter is to estimate its unknown filter states, i.e. the kinematic states in this thesis,  $\hat{\mathbf{t}}(k|k)$  at cycle  $k$  on the basis of the measurements taken in the previous  $k$  cycles

$$\mathbf{M}_0^k = \{\mathbf{m}(0), \dots, \mathbf{m}(k)\} \quad (2.56)$$

Without making any assumptions on the dynamic behavior of a target, the optimal estimate  $\hat{\mathbf{t}}(k|k)$  of the target's real kinematic state  $\mathbf{t}(k)$  at cycle  $k$  in the minimum mean square sense (MMSE) is given by the conditional mean [Kay93]

$$\hat{\mathbf{t}}(k|k) = E [\mathbf{t}(k)|\mathbf{M}_0^k] = \int \mathbf{t}(k) \cdot p(\mathbf{t}(k)|\mathbf{M}_0^k) dt(k) \quad (2.57)$$

where  $\mathbf{M}_0^k$  is the sequence of measurements up to cycle  $k$ . This determination of the posterior estimate  $\hat{\mathbf{t}}(k|k)$  is called Bayesian approach and requires the knowledge of the posterior probability density function (PDF)  $p(\mathbf{t}(k)|\mathbf{M}_0^k)$ . In most cases this density is unknown. But the Bayesian approach can be simplified under the assumption that the densities of all variables are Gaussian [WM01], so that the posteriori kinematic state estimate  $\hat{\mathbf{t}}(k|k)$  at cycle  $k$  is given by the following equations

*Optimal State Estimate Update:*

$$\hat{\mathbf{t}}(k|k) = \hat{\mathbf{t}}(k|k-1) + K(k) \cdot \tilde{\mathbf{m}}(k) \quad (2.58)$$

$$K(k) = C(k) \cdot S(k)^{-1} \quad (2.59)$$

$$P(k|k) = P(k|k-1) - K(k) \cdot R(k) \cdot K(k)^T \quad (2.60)$$

where  $R(k)$  is the measurement noise covariance matrix,  $\tilde{\mathbf{m}}(k)$  is the measurement innovation

$$\tilde{\mathbf{m}}(k) = \mathbf{m}(k) - \hat{\mathbf{m}}(k|k-1) \quad (2.61)$$

with its covariance matrix  $S(k)$ .  $C(k)$  is the cross-covariance matrix between the target's kinematic state and the measurements and is given by

$$C(k) = E [(\mathbf{t}(k) - E[\mathbf{t}(k)])(\mathbf{m}(k) - E[\mathbf{m}(k)])^T] \quad (2.62)$$

The prediction of the kinematic state  $\hat{\mathbf{t}}(k|k-1)$ , the appropriate covariance matrix  $P(k|k-1)$ , and the measurements  $\hat{\mathbf{m}}(k|k-1)$  are obtained from the following prediction equations:

*Optimal State Estimate Prediction:*

$$\hat{\mathbf{t}}(k|k-1) = E [\phi_{k-1}(\hat{\mathbf{t}}(k-1|k-1), \mathbf{f}(k-1), \mathbf{w}(k-1))] \quad (2.63)$$

$$P(k|k-1) = C(k) \cdot S(k)^{-1} \quad (2.64)$$

$$\hat{\mathbf{m}}(k|k-1) = E [h_{k-1}(\hat{\mathbf{t}}(k|k-1), \mathbf{f}(k), \mathbf{v}(k))] \quad (2.65)$$

In these prediction equations, the state-transition function  $\phi_{k-1}(\dots)$  describes the time-varying behavior of the kinematic filter states and the measurement function  $h_{k-1}(\dots)$  the relation between the measurements and the filter states. Both base on the following state-space model

$$\mathbf{t}(k|k-1) = \phi_{k-1}(\mathbf{t}_{k-1|k-1}, \mathbf{f}_{k-1}, \mathbf{w}_{k-1}) \quad (2.66)$$

$$\mathbf{m}(k) = h_k(\mathbf{t}(k|k), \mathbf{f}(k), \mathbf{v}(k)) \quad (2.67)$$

This model is a general system description assuming ...

- that the state prediction  $\mathbf{t}(k|k-1)$  is depending on the last posteriori state estimate  $\mathbf{t}(k-1|k-1)$  at time  $k-1$ , on a deterministic control input  $\mathbf{f}(k)$ , e.g. the known position change of the own vehicle, and on the Gaussian noise  $\mathbf{w}(k)$  representing the randomness in the target's motion.
- that the measurements  $\mathbf{m}(k)$  at time  $k$  are depending on the real kinematic state  $\mathbf{t}(k)$ , on the deterministic control input  $\mathbf{f}(k)$ , and on the Gaussian measurement noise  $\mathbf{v}(k)$ .

This general Kalman filter approach will be the starting point for the following derivations of three different types of Kalman filters, i.e. the linear Kalman filter (LKF), the extended Kalman filter (EKF), and the unscented Kalman filter (UKF). All these approaches will refer to the general recursive estimation equations (2.58)-(2.65) and simplify them according to each filter's assumptions.

### 2.2.2 Linear Kalman Filter

The linear Kalman filter is based on the general Kalman filter approach of the last section with two additional assumptions:

- First, it assumes the correlation between the target's kinematic state samples of consecutive scans  $k$  and  $k + 1$  to be given by a first-order Gauss-Markov process [Kay93].
- Second, the unknown states  $\mathbf{t}(k)$  and the measurements  $\mathbf{m}(k)$  are assumed to have a linear relationship.

Thus, the general state-space description of eq. (2.66) and (2.67) is simplified to the following equations

$$\mathbf{t}(k + 1) = \Phi(k) \cdot \mathbf{t}(k) + B(k) \cdot \mathbf{f}(k) + \mathbf{w}(k) \quad (2.68)$$

$$\mathbf{m}(k) = H(k) \cdot \mathbf{t}(k) + D(k) \cdot \mathbf{f}(k) + \mathbf{v}(k) \quad (2.69)$$

At this,  $\mathbf{t}$  is the state vector that wants to be estimated. The matrix  $\Phi(k)$  describes the state transition from scan  $k$  to  $k + 1$  and is assumed to be known.  $\mathbf{w}(k)$  is a zero-mean, white, Gaussian noise with assumed known covariance matrix  $Q$ .  $Q$  is also called process noise and represents the randomness in the target's motion, such as a random acceleration. The vector  $\mathbf{f}(k)$  is a deterministic input, such as a known kinematic state change of the own vehicle at scan  $k$ .  $B(k)$  is the appropriate state transition matrix for the deterministic control input  $\mathbf{f}(k)$ . In the measurement equation (2.69),  $H(k)$  is the measurement matrix and  $\mathbf{v}(k)$  is the zero-mean, white, Gaussian measurement noise with a covariance matrix  $R(k)$ . The deterministic control input  $\mathbf{f}(k)$  may also have an impact on the measurements  $\mathbf{m}(k)$ , so that this influence is described by the matrix  $D(k)$ .

Based on this state-space model, the LKF estimates the kinematic state vector  $\hat{\mathbf{t}}(k + 1)$  at scan  $k + 1$  on the basis of the previous  $k + 1$  measurements

$$\mathbf{M}_0^{k+1} = \{\mathbf{m}(0), \dots, \mathbf{m}(k + 1)\} \quad (2.70)$$

with their covariance matrices  $R(0), \dots, R(k + 1)$ . The following description of the LKF algorithm can directly be derived from the basic recursive estimation equations in (2.58)-(2.65) by inserting the underlying linear state-space description of eq.(2.68) and (2.69).

### Filter Algorithm

Given the target dynamics and the measurement model from the state-space description in eq.(2.68) and (2.69), the linear Kalman filter steps are as follows:

*State Estimate Initialization:*

$$P(0|0) = Cov(\mathbf{t}(0)) \quad (2.71)$$

$$\hat{\mathbf{t}}(0|0) = E(\mathbf{t}(0)) \quad (2.72)$$

*State Estimate Prediction:*

$$P(k|k-1) = \Phi(k)P(k-1|k-1)\Phi(k)^T + Q \quad (2.73)$$

$$\hat{\mathbf{t}}(k|k-1) = \Phi(k)\hat{\mathbf{t}}(k-1|k-1) + B(k-1)\mathbf{f}(k-1) \quad (2.74)$$

*State Estimate Update:*

$$K(k) = P(k|k-1)H(k)^T(H(k)P(k|k-1)H(k)^T + R(k))^{-1} \quad (2.75)$$

$$P(k|k) = (I - K(k)H(k))P(k|k-1) \quad (2.76)$$

$$\hat{\mathbf{t}}(k|k) = \hat{\mathbf{t}}(k|k-1) + K(k)(\mathbf{m}(k) - D(k)\mathbf{f}(k) - H(k)\hat{\mathbf{t}}(k|k-1)) \quad (2.77)$$

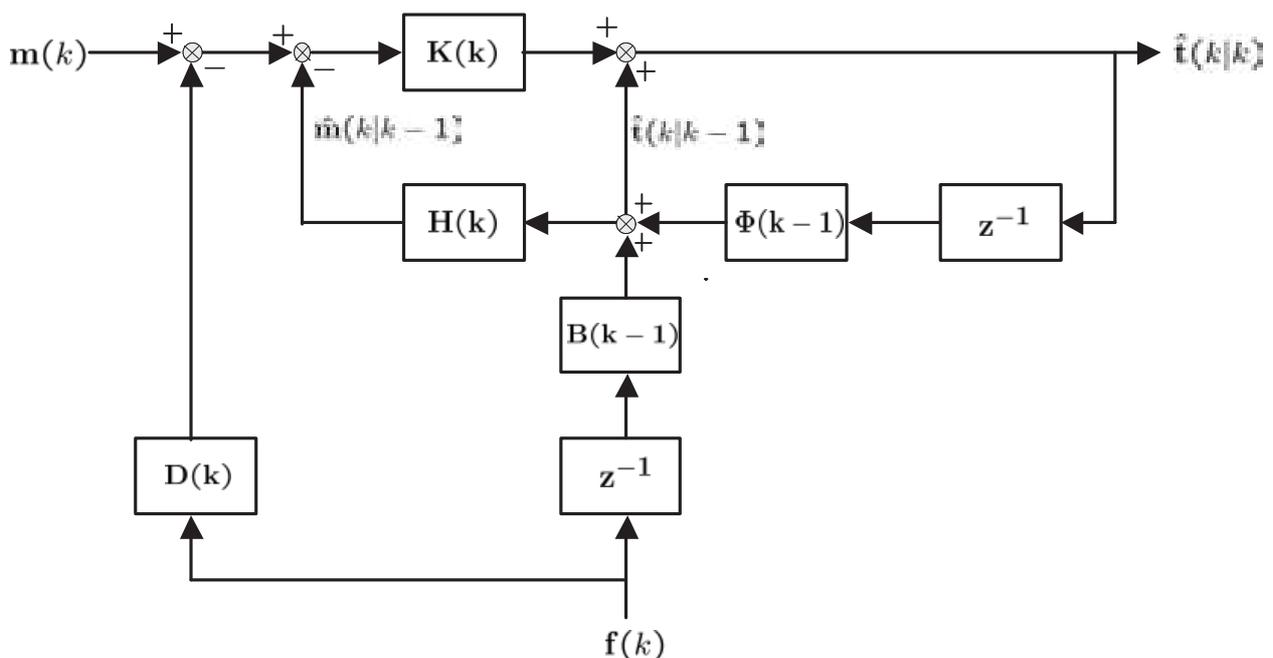


Figure 2.6: Implementation of the Kalman filter algorithm

The LKF is an optimal MMSE estimator under the assumption that the process and the measurement noise are Gaussian. Because both are assumed to be zero-mean, the resulting state estimate  $\hat{\mathbf{t}}(k|k)$  is also unbiased

$$E[\hat{\mathbf{t}}(k|k)] = E[\mathbf{t}(k|k)] \quad (2.78)$$

The LKF approach can be considered as an efficient method of analytically propagating Gaussian random variables through linear system dynamics.

A realization of the LKF is shown in figure 2.6. Beginning with the update equations (2.75), (2.76), and (2.77), the measurements  $\mathbf{m}(k)$  at time  $k$  are received from the different sensors and corrected by the deterministic control input  $D(k) \cdot \mathbf{f}(k)$ . Then, the innovation of the corrected measurements is calculated by subtracting the expected measurements  $\hat{\mathbf{m}}(k|k-1)$  derived from the information of the last cycle. After that, the Kalman gain  $K(k)$  is calculated according to eq.(2.75). Knowing the weight  $K(k)$  of the innovation term  $(\mathbf{m}(k) - D(k)\mathbf{f}(k) - H(k)\hat{\mathbf{t}}(k|k-1))$  in eq.(2.77), the new kinematic state estimate  $\hat{\mathbf{t}}(k|k)$  and the appropriate covariance matrix  $P(k|k)$  (see eq.(2.76)) are calculated. To obtain the predictions of the kinematic states and the covariance matrix for the next measurement cycle  $k+1$ , the prediction equations (2.73) and (2.74) are applied.

### Example of the Linear Kalman Filter Steps

To illustrate the cycle-to-cycle estimation process of a LKF and to give an impression of the impact of different Kalman parameters on its kinematic state estimation, a principle example will be explained in the following.

As shown in figure 2.7(a), a target with the kinematic state  $\mathbf{t}(0|0)$  is moving in the positive y-direction at scan  $k=0$ . After the first target detection, the Kalman filter is initialized and fed with the actual target measurement  $\mathbf{m}(0)$ , so that the first a posteriori kinematic state estimate  $\hat{\mathbf{t}}(0|0)$  with its appropriate covariance matrix  $P(0|0)$  is retrieved. In this example, the Kalman filter initialization has been a poor choice, so that the uncertainty of the first kinematic state estimate is very high. In the next cycle (figure 2.7(b)), the target has the real kinematic state  $\mathbf{t}(1|1)$  and the actual target measurement vector  $\mathbf{m}(1)$  has an uncertainty described by the covariance matrix  $R(1)$ . The predicted kinematic state  $\hat{\mathbf{t}}(1|0)$  is calculated from the information of the previous estimate  $\hat{\mathbf{t}}(0|0)$ . Due to the possible error of this prediction, its uncertainty is increasing, so that the predicted covariance matrix  $P(1|0)$  is much greater than the measurement noise covariance matrix  $R(1)$ . Due to the weighting according to the covariance matrices, the actual measurement will have a great impact on the new posteriori estimate  $\hat{\mathbf{t}}(1|1)$  in the Kalman filter update equations. The uncertainty of the new estimate given by the covariance matrix  $P(1|1)$  is significantly decreasing compared to  $P(0|0)$ . This prediction and updating procedure is repeated in every measurement cycle, so that the target's kinematic state estimate  $\hat{\mathbf{t}}(3|3)$  at cycle  $k=3$  has achieved a high accuracy denoted by small values in the covariance matrix  $P(3|3)$  (figure 2.7(d)).

Assuming precise single-sensor measurements, the quickness of convergence towards the

real target's kinematic state  $\mathbf{t}(k)$  depends only on the quality of filter state initialization. It is simple to understand that the closer the initialization of the filter's kinematic states  $\hat{\mathbf{t}}(0|0)$  is located to the real kinematic states  $\mathbf{t}(0)$ , the faster a precise kinematic state estimate can be retrieved. But in the case where the choice of the state initialization has been poor, the convergence time can also be kept short by increasing the predicted covariance matrix of the predicted kinematic state estimate in the first measurement cycles after initialization. Such an increase in the uncertainty causes the Kalman filter "to believe" more in the actual measurements than in its filter states and thus will quickly lead to an accurate kinematic state estimate. It should only be applied over a few cycles because otherwise the Kalman filter will not enhance the estimation accuracy compared to the pure measurement information.

### 2.2.3 Target Dynamic Models

The special advantage in Kalman filtering is the incorporation of a target dynamic model that describes the possible kinematic state changes between consecutive measurement cycles. The better the target dynamic model fits to the real target behavior, the more accurate are the filter state estimates. The target dynamic model is represented by the so-called process noise in the state-transition equation (2.68) of the underlying state-space model. The covariance matrix of this process noise expresses the uncertainty in the Kalman filter state estimation due to random target dynamics, or in other words, it determines the allowed target dynamic changes from cycle to cycle. It has to be derived from the target dynamics occurring in the targeted road traffic situations. The choice of the process noise covariance matrix  $Q$  has been discussed in many publications, e.g. in [BP99], [BSF88], and [BSL93]. These works concentrate on the field of airborne radar signal processing and not every target dynamic model in this field may be reasonable for the application in road traffic scenarios. This thesis will present two approaches that have proven to show a very good performance in real traffic scenarios, the piecewise constant white noise acceleration and the piecewise constant Wiener process acceleration model.

#### Piecewise Constant White Noise Acceleration Model

The state-transition equation of the LKF has already been given in (2.68) and can be rewritten for the stochastic part as

$$\mathbf{t}(k+1) = \Phi(k) \cdot \mathbf{t}(k) + \mathbf{w}(k) \quad (2.79)$$

Assuming  $\mathbf{t}(k)$  to be a 2-state vector

$$\mathbf{t}(k) = [t_x \quad t_y \quad t_{v_x} \quad t_{v_y}]^T \quad (2.80)$$

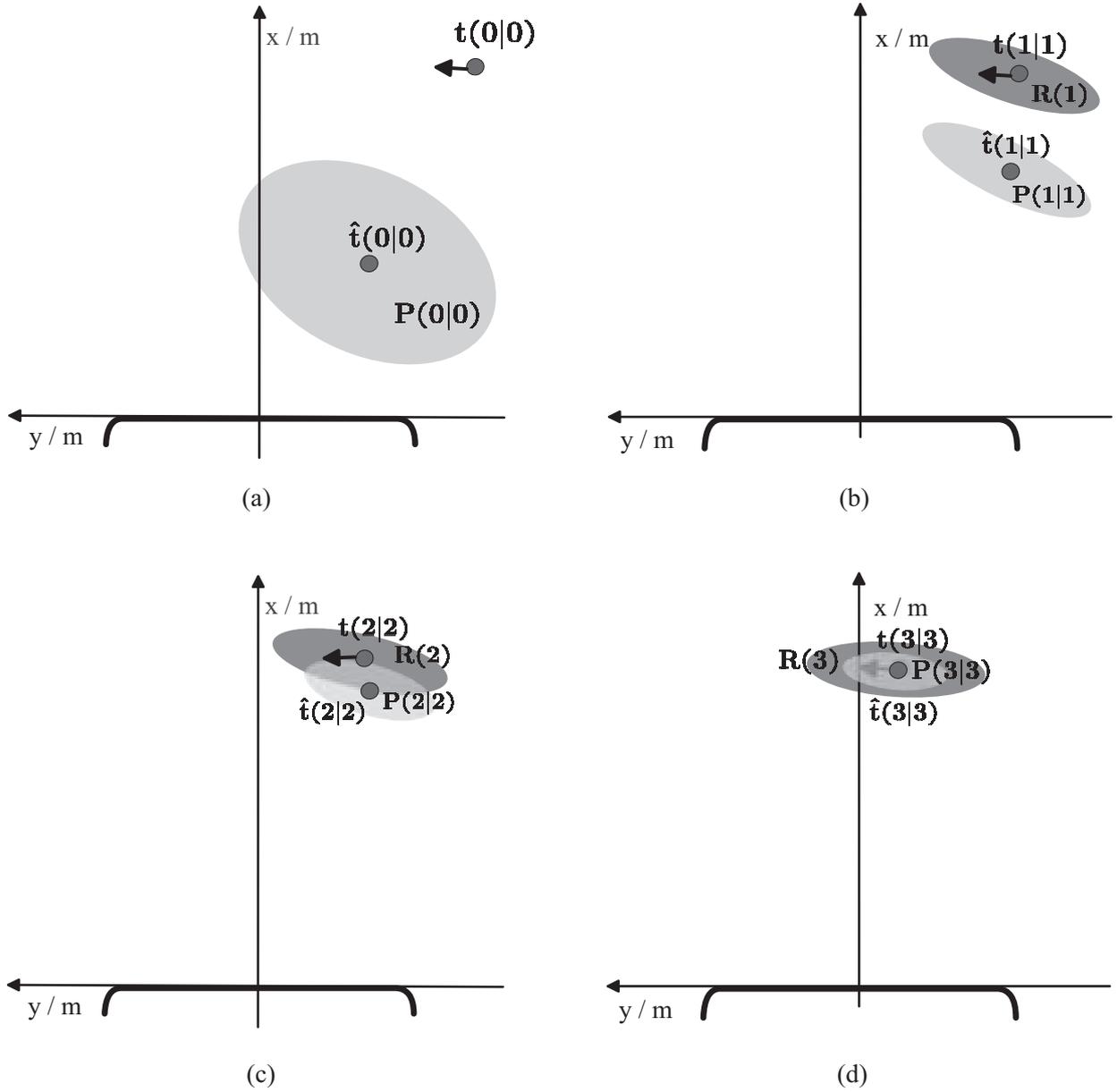


Figure 2.7: Example of the principle Kalman filter estimation steps from the initialization  $\hat{\mathbf{t}}(0|0)$ ,  $P(0|0)$  to the estimate  $\hat{\mathbf{t}}(3|3)$ ,  $P(3|3)$  at cycle 3

the transition matrix  $\Phi(k)$  is given by

$$\Phi(k) = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.81)$$

where  $T$  denotes the duration of the sampling period. In contradiction to many other target dynamic model derivations in the literature, the piecewise constant white noise acceleration model is directly defined in discrete time. It assumes that the target performs accelerations that are constant over the sampling interval  $T$ . This can be expressed by

a discrete time process noise  $w_a(k)$ , which is a scalar-valued, zero-mean white sequence with

$$E[w_a(k)w_a(l)] = \sigma_{w_a}^2 \delta(kl) \quad (2.82)$$

It is assumed that  $w_a(k)$  is the constant acceleration during the  $k$ -th sampling period of length  $T$ . Thus, the target's velocity increases during this period by  $w_a(k) \cdot T$ . The position is also affected by  $w_a(k) \cdot T^2/2$ . Therefore, the process noise  $\mathbf{w}(k)$  in eq.(2.79) is given by

$$\mathbf{w}(k) = \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \cdot w_a(k) \quad (2.83)$$

The appropriate process noise covariance matrix  $Q$  can be calculated as

$$\begin{aligned} Q &= E[\mathbf{w}(k)^T \mathbf{w}(k)] \\ &= \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \cdot \sigma_{w_a(k)}^2 \cdot \begin{bmatrix} T^2/2 & T \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{4}T^4 & \frac{1}{2}T^3 \\ \frac{1}{2}T^3 & T^2 \end{bmatrix} \cdot \sigma_{w_a(k)}^2 \end{aligned} \quad (2.84)$$

As a guideline for the process noise variance choice  $\sigma_{w_a(k)}^2$ , [BSL93] suggests to select the standard deviation  $\sigma_{w_a(k)}$  to be of the order of the maximum occurring acceleration  $a_{max}$ , so that a practical range of  $\sigma_{w_a(k)}$  is given by

$$0.4 \cdot a_{max} \leq \sigma_{w_a(k)} \leq a_{max} \quad (2.85)$$

In real road traffic situations, this guideline has proven to work properly.

### Piecewise Constant Wiener Process Acceleration Model

Some safety applications demand an excellent performance in traffic situation where great accelerations occur, e.g. in a situation where a road user performs an emergency braking. Utilizing a 2-state Kalman filter with a 2-state target dynamic model, i.e. without any estimated acceleration  $\hat{\mathbf{t}}_a$ , will lead to poor kinematic state predictions and thus to missed track updates. This tracking filter behavior is critical for the decisions/actions that have to be taken by the safety application and can be avoided by implementing a 3-state Kalman filter with an appropriate 3-state target dynamic model. Such a 3-state target dynamic model can analogously be developed to the 2-state piecewise constant white noise acceleration model derived in the previous section.

The so-called piecewise constant Wiener process noise acceleration model is a 3-state target dynamic model and assumes that the increments to acceleration  $\Delta a$  are constant over the sampling interval  $T$ . This leads to the process noise  $\mathbf{w}(k)$  in eq.(2.79) given by

$$\mathbf{w}(k) = \begin{bmatrix} T^2/2 \\ T \\ 1 \end{bmatrix} \cdot w_a(k) \quad (2.86)$$

with the appropriate process noise covariance matrix  $Q$  given by

$$\begin{aligned}
Q &= E[\mathbf{w}(k)^T \mathbf{w}(k)] \\
&= \begin{bmatrix} T^2/2 \\ T \\ 1 \end{bmatrix} \cdot \sigma_{w_a(k)}^2 \cdot \begin{bmatrix} T^2/2 & T & 1 \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{4}T^4 & \frac{1}{2}T^3 & \frac{1}{2}T^2 \\ \frac{1}{2}T^3 & T^2 & T \\ \frac{1}{2}T^2 & T & 1 \end{bmatrix} \cdot \sigma_{w_a(k)}^2
\end{aligned} \tag{2.87}$$

The guideline for choosing  $\sigma_{w_a(k)}$  can be given, so that

$$0.4 \cdot \Delta a_{max} \leq \sigma_{w_a(k)} \leq \Delta a_{max} \tag{2.88}$$

This guideline suggests that  $\sigma_{w_a(k)}$  should be smaller than the maximum acceleration increment expected over the sampling period  $T$ , but at least 40% of this maximum acceleration increment.

The main difference between the presented 2-state and 3-state model is the different assumption on the observed target's acceleration. In the piecewise constant white noise acceleration model, the input acceleration is uncorrelated from measurement cycle to measurement cycle while the piecewise constant Wiener process noise acceleration model assumes the acceleration to be a random walk process.

### Interactive Multiple Modeling (IMM)

Several publications (e.g. [BP99]) try to improve the target dynamic modeling of the standard Kalman filter implementation by applying multiple models, each representing a different target dynamic behavior. At this, these different models are processed in a parallel Kalman filter structure and the achieved tracking filter results are combined to a final filter state estimate. Such an approach is called interacting multiple model (IMM) Kalman filtering and has proven to achieve more accurate state estimates. But after investigating the side conditions of this superior performance, it became clear that the IMM approach will be no benefit in an automotive postprocessing structure. The IMM is advantageous if the measurement cycle time is relative long, so that the observed object could have performed a significant maneuver (which can be modeled appropriately by one of the different dynamic models). The considered radar sensors in this thesis have a target list update rate of 40 Hz, which refers to a measurement cycle time of 25 ms. During this time, none of the considered road users will perform such a significant kinematic state change, so that a well-chosen single-target dynamic model reveals to be not inferior to the IMM approach. Additionally, the IMM approach will linearly increase the computation complexity with the number of implemented models. As a conclusion, the IMM approach will not be considered for the postprocessing architecture in this thesis.

## Summary

The choice of the process noise matrix  $Q$  and thus the target dynamic model is a compromise between choosing a quick responding and a well smoothing, noise filtering Kalman filter. Starting from the present target dynamic model derivations is a good basis for adjusting each model according to the targeted real road traffic scenarios. But it is important that this theoretical derived process noise and its parameters are evaluated empirically and appropriately adjusted in real road traffic situations.

### 2.2.4 Extended Kalman Filter

The LKF requirement that all observation and process models are linear is rarely satisfied in practical applications. To apply the Kalman filter approach to a nonlinear measurement situation as given by the considered UMR sensors delivering measurements in polar sensor coordinates while estimating the kinematic state in global cartesian coordinates, one idea is to linearize the underlying nonlinear measurement equations. Such a tracking filter approach is the extended Kalman filter (EKF). It works with the same linear filter equations as the LKF while the measurement matrix  $H$  is obtained by a linearization of the nonlinear measurement equations  $h(\dots)$ .

Considering this case of nonlinear measurement equations, the LKF state-space description of eq.(2.68) and (2.69) changes to

$$\begin{aligned}\mathbf{t}(k+1) &= \Phi(k) \cdot \mathbf{t}(k) + B(k) \cdot \mathbf{f}(k) + \mathbf{q}(k) \\ \mathbf{m}(k) &= h_k(\mathbf{t}(k)) + D(k) \cdot \mathbf{f}(k) + \mathbf{v}(k)\end{aligned}\tag{2.89}$$

To achieve a linear state-space model, the EKF linearizes the nonlinear measurement function  $h_k(\mathbf{t}(k))$  in the moment where the measurements are taken under the assumption that the predicted kinematic state is not "too much" error-prone. This linearization of  $h_k(\mathbf{t}(k))$  is a first order approximation of the nonlinear measurement equations  $h_k(\mathbf{t}(k))$  at the target's predicted kinematic state  $\hat{\mathbf{t}}(k|k-1)$ . Thus, the LKF equations remain valid for the EKF with the measurement matrix  $H(k)$  being the linearization of  $h_k(\mathbf{t}(k))$ . The EKF equations are summarized on the next page.

*State Estimate Initialization:*

$$P(0|0) = Cov(\mathbf{t}(0)) \quad (2.90)$$

$$\hat{\mathbf{t}}(0|0) = E(\mathbf{t}(0)) \quad (2.91)$$

*State Estimate Prediction:*

$$P(k|k-1) = \Phi(k)P(k-1|k-1)\Phi(k)^T + Q \quad (2.92)$$

$$\hat{\mathbf{t}}(k|k-1) = \Phi(k)\hat{\mathbf{t}}(k-1|k-1) + B(k-1)\mathbf{f}(k-1) \quad (2.93)$$

*State Estimate Update:*

$$K(k) = P(k|k-1)H(k)^T(H(k)P(k|k-1)H(k)^T + R(k))^{-1} \quad (2.94)$$

$$P(k|k) = (I - K(k)H(k))P(k|k-1) \quad (2.95)$$

$$\hat{\mathbf{t}}(k|k) = \hat{\mathbf{t}}(k|k-1) + K(k)(\mathbf{m}(k) - D(k)\mathbf{f}(k) - H(k)\hat{\mathbf{t}}(k|k-1)) \quad (2.96)$$

*Linearization of  $h_k(\mathbf{t})$ :*

$$H(k) = \left. \frac{\partial h_k(\mathbf{t})}{\partial \mathbf{t}} \right|_{\mathbf{t}=\hat{\mathbf{t}}(k|k-1)} \quad (2.97)$$

The EKF has proven to be a powerful and popular approach, but must be considered as suboptimal compared to the LKF. The LKF can be viewed as an efficient method for analytically propagating Gaussian random variables through linear system dynamics. In the case of the nonlinear state-space model in eq.(2.89), the EKF approximates the optimal term of the measurement predictions in eq.(2.65) due to its linearization of the nonlinear measurement function  $h_k(\mathbf{t})$  by

$$\hat{\mathbf{m}}(k|k-1) \approx H(k)\hat{\mathbf{t}}(k|k-1) \quad (2.98)$$

with  $H(k)$  defined in eq.(2.97). Thus, it is introducing an error, which is the residual of the Taylor serie's first term given by

$$Res_k(\mathbf{t}(k)) = h_k(\mathbf{t}(k)) - h_k(\hat{\mathbf{t}}(k|k-1)) + \left. \frac{\partial h_k(\mathbf{t})}{\partial \mathbf{t}} \right|_{\mathbf{t}=\hat{\mathbf{t}}(k|k-1)} \cdot \frac{(\mathbf{t}(k) - \hat{\mathbf{t}}(k|k-1))}{1!} \quad (2.99)$$

Due to this error, the analytical propagation of Gaussian random variables through this linearized system results in a suboptimal state and covariance estimate,  $\hat{\mathbf{t}}(k|k)$  and  $P(k|k)$ . It may occur that large errors are introduced in the linearization process by error-prone state predictions, so that convergence is not guaranteed. To avoid such divergence and to obtain the best possible estimate with the EKF, different methods of feeding it with the actual measurement data have been developed:

**Parallel processing** of the actual measurement data is the direct approach as it is applied by the standard EKF equations (2.90)-(2.97). It means that the complete set of  $N$  measurements belonging to a single target

$$\mathbf{m} = \begin{bmatrix} \mathbf{m}^{\text{Sens } 1} \\ \mathbf{m}^{\text{Sens } 2} \\ \vdots \\ \mathbf{m}^{\text{Sens } N} \end{bmatrix} \quad (2.100)$$

is fed simultaneously in the EKF to correct the state prediction  $\hat{\mathbf{t}}(k|k-1)$  of the last cycle. The accuracy of the linearization (2.97) and thus the accuracy of the estimated filter state  $\hat{\mathbf{t}}(k|k)$  is depending on the quality of the state prediction  $\hat{\mathbf{t}}(k|k-1)$ . In case of a poor prediction, this approach introduces a large estimation error or may even lead to divergence in some situations.

**Sequential processing** tries to ensure the convergence and accuracy of the state estimation by taking advantage of the available  $N$  target measurements. These target measurements are sequentially fed into the EKF in the order of their measurement accuracy (most accurate measurement first). This means that the EKF is updated as many times as number of measurements are available.

$$\mathbf{m}_n = [\mathbf{m}^{\text{Sens } n}] \quad n = 1, \dots, N \quad (2.101)$$

In the first filter update step, the state prediction  $\hat{\mathbf{t}}(k|k-1)$  of the last measurement cycle is corrected by the first measurement  $m_1(k)$  to  $\hat{\mathbf{t}}_1(k|k)$ . If  $\hat{\mathbf{t}}(k|k-1)$  is of poor quality, it will be significantly corrected by  $m_1(k)$ . This corrected filter state prediction will again be corrected by the next available measurement  $m_2(k)$  in the following filter update step. But in contradiction to the parallel processing approach, the measurement matrix  $H_2(k)$  will not be determined at the first state prediction  $\hat{\mathbf{t}}(k|k-1)$ .  $H_2(k)$  will be calculated as a linearization of  $h(k)(\mathbf{t})$  at the corrected state prediction  $\hat{\mathbf{t}}_1(k|k)$  and thus will introduce a significantly smaller linearization error. This sequential update procedure is repeated until all  $N$  measurements are processed once. The  $n$ -th update step is given by

$$K_n(k) = P_n(k|k-1)H_n(k)^T(H_n(k)P_n(k|k)H_n(k)^T + R_n(k))^{-1} \quad (2.102)$$

$$P_n(k|k) = (I - K_n(k)H_n(k))P_n(k|k-1) \quad (2.103)$$

$$\hat{\mathbf{t}}_n(k|k) = \hat{\mathbf{t}}_n(k|k-1) + K_n(k)(\mathbf{m}_n(k) - D(k)\mathbf{f}(k) - H_n(k)\hat{\mathbf{t}}_n(k|k-1)) \quad (2.104)$$

where

$$H_n(k) = \left. \frac{\partial h_k(\mathbf{t})}{\partial \mathbf{t}} \right|_{\mathbf{t}=\hat{\mathbf{t}}_n(k|k)} \quad (2.105)$$

with

$$P_0(k|k) = P_n(k|k-1) \quad (2.106)$$

$$\hat{\mathbf{t}}_0(k|k) = \hat{\mathbf{t}}_n(k|k-1) \quad (2.107)$$

No Kalman filter state prediction must be calculated between the different sequential update steps within the same measurement cycle.

The sequential calculation of the EKF state update leads to an enhancement of the linearization accuracy in every sequential processing step, because the state prediction  $\hat{\mathbf{t}}(k|k-1)$  is corrected to a more accurate state estimate by each measurement value in each of the sequential processing steps. In addition to increasing the filter stability, this approach also leads to a significant computational simplification. Only one measurement value  $m^{\text{Sens}n}(k)$  and one measurement noise covariance matrix  $R(k)$  are simultaneously fed into the EKF update equation per sequential step. This causes a reduced size of all incorporated matrices and thus to a significantly reduced number of operations compared to the parallel approach.

### 2.2.5 Unscented Kalman Filter

The unscented Kalman filter (UKF) is another approach of applying the Kalman filter to nonlinear systems. The previously presented EKF is considered to be suboptimal because it simply replaces every nonlinear relationship with a linearized approximation, which leads to the described disadvantages. In [JU96] an alternative Kalman filter approach to nonlinear systems is derived. This approach is not approximating the nonlinear functions in eq.(2.63) and (2.65), instead, it chooses discrete representations of the mean and variance information of all Gaussian random variables contributing to the system description, so that these discrete representations can be propagated through the true nonlinear system. The advantage of this procedure is the avoidance of linearization errors that might lead to divergence or large estimation errors. According to [WM01], the achieved accuracy of the posteriori state estimate  $\hat{\mathbf{t}}((k|k))$  corresponds to the accuracy of a second order Taylor series approximating the functions in eq.(2.63) and (2.65) for any nonlinearity. The unscented transformation was first introduced by [JU96] and is discussed intensively in [WM01], [Jul00], [JU97a], [JU97b] and [Jul98] for different applications.

#### Unscented Transformation

The unscented transformation (UT) is a method for calculating the statistics of a random variable  $x$  that undergoes a nonlinear transformation  $y = g(x)$ . The idea of the UT is to approximate the statistical distribution of  $x$  by generating a discrete distribution having the same first and second statistical moment. Each point of this discrete distribution approximation can directly be propagated through the nonlinear function  $g(x)$ . The mean and covariance estimate of the nonlinear transformation result  $y$  can be extracted from the nonlinear transformation of the approximated discrete distribution of  $x$ . This basic principle of the UT approach is illustrated in figure 2.8.

Assuming that  $x$  has the mean value  $\bar{x}$  and covariance  $P_x$ , the statistics of  $x$  can be represented by a discrete approximation of its distribution. This discrete approximation

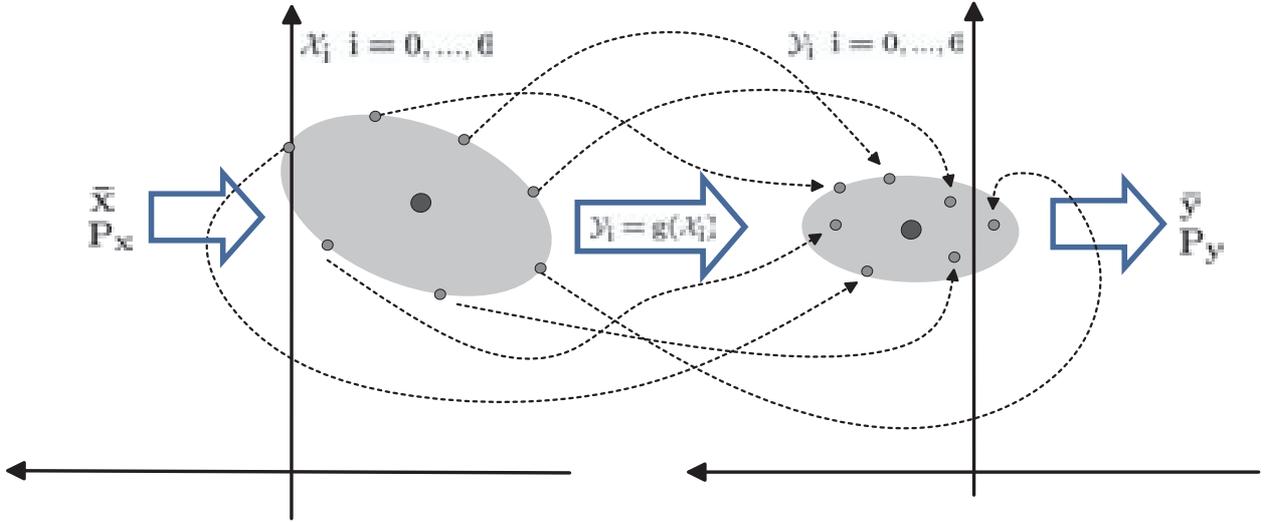


Figure 2.8: The discrete sigma points  $\mathcal{X}_i$  are capturing the mean and covariance of the random variable  $x$ . After the nonlinear transformation  $\mathcal{Y}_i = g(\mathcal{X}_i)$ , the mean and covariance of  $y$  can be estimated from sigma points  $\mathcal{Y}_i$

is obtained by forming a matrix  $\mathcal{X}$  of  $2L + 1$  so-called sigma vectors  $\mathcal{X}_i$

$$\mathcal{X}_0 = \bar{x} \quad (2.108)$$

$$\mathcal{X}_i = \bar{x} + \left( \sqrt{(L + \lambda)P_x} \right)_i, \quad i = 1, \dots, L \quad (2.109)$$

$$\mathcal{X}_i = \bar{x} - \left( \sqrt{(L + \lambda)P_x} \right)_{i-L}, \quad i = L + 1, \dots, 2L \quad (2.110)$$

$$(2.111)$$

with the scaling parameter  $\lambda$

$$\lambda = \alpha^2(L + \kappa) - L \quad (2.112)$$

The constant  $\alpha$  determines the spread of the discrete sigma points around the mean  $\bar{x}$  and is usually set to a small positive value in the range of  $10^{-4} \leq \alpha \leq 1$  (see [WM01]). The second scaling parameter  $\kappa$  must be set to either 0 or  $3 - L$ . The parameter  $\beta$  incorporates prior knowledge about the distribution of  $x$ , e.g. for Gaussian distributions  $\beta = 2$  has proven to be optimal. The term  $(\sqrt{(L + \lambda)P_x})_i$  is the  $i$ -th column of the matrix square root of  $(L + \lambda)P_x$ .

The discrete sigma points  $\mathcal{X}_i$  are now propagated through the true nonlinear function  $g(\dots)$ , so that the resulting discrete sigma points  $\mathcal{Y}_i$  approximating the distribution of  $y$  are

$$\mathcal{Y}_i = g(\mathcal{X}_i) \quad i = 0, \dots, 2L \quad (2.113)$$

The estimated mean and the covariance of the random variable  $y$  can be regained by calculating the weighted sample mean and covariance of the propagated sigma points  $\mathcal{Y}_i$

$$\bar{y} \approx \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_i \quad (2.114)$$

$$P_y \approx \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{Y}_i - \bar{y}][\mathcal{Y}_i - \bar{y}]^T \quad (2.115)$$

with weights  $W_i^{(m)}$  and  $W_i^{(c)}$  given by

$$\begin{aligned} W_0^{(m)} &= \lambda / (L + \lambda) \\ W_0^{(c)} &= \lambda / (L + \lambda) + (1 - \alpha^2 + \beta) \\ W_i^{(m)} = W_i^{(c)} &= 1 / [2(L + \lambda)] \quad i = 1, \dots, 2L \end{aligned} \quad (2.116)$$

### Filter Algorithm

The unscented Kalman filter is a straightforward application of the UT to the general Kalman filter equations (2.58)-(2.60) and (2.63)-(2.65). Assuming the same state-space model as in the case of the EKF

$$\begin{aligned} \mathbf{t}(k+1) &= \Phi(k) \cdot \mathbf{t}(k) + B(k) \cdot \mathbf{f}(k) + \mathbf{q}(k) \\ \mathbf{m}(k) &= h_k(\mathbf{t}(k)) + D(k) \cdot \mathbf{f}(k) + \mathbf{v}(k) \end{aligned} \quad (2.117)$$

with the process and the measurement noise being zero-mean, white, and Gaussian, the UKF filter equations are given by the following equations:

*State Estimate Initialization:*

$$P(0|0) = \text{Cov}(\mathbf{t}(0)) \quad (2.118)$$

$$\hat{\mathbf{t}}(0|0) = E(\mathbf{t}(0)) \quad (2.119)$$

*Sigma Points Calculation:*

$$\mathcal{X}(k-1|k-1) = \begin{bmatrix} \hat{\mathbf{t}}(k-1|k-1) \\ \hat{\mathbf{t}}(k-1|k-1) + \gamma \sqrt{P(k-1|k-1)} \\ \hat{\mathbf{t}}(k-1|k-1) - \gamma \sqrt{P(k-1|k-1)} \end{bmatrix}^T \quad (2.120)$$

*State Estimate Prediction:*

$$\mathcal{X}(k|k-1) = \Phi(k-1) \cdot \mathcal{X}(k-1|k-1) \quad (2.121)$$

$$\hat{\mathbf{t}}(k|k-1) = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{X}_{i,k|k-1} + B(k) \cdot \mathbf{f}(k) \quad (2.122)$$

$$P(k|k-1) = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{X}_{i,k|k-1} - \hat{\mathbf{t}}(k|k-1)] [\mathcal{X}_{i,k|k-1} - \hat{\mathbf{t}}(k|k-1)]^T + Q \quad (2.123)$$

$$\mathcal{Y}(k|k-1) = h_k(\mathcal{X}(k|k-1)) \quad (2.124)$$

$$\hat{\mathbf{m}}(k|k-1) = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_{i,k|k-1} \quad (2.125)$$

*State Estimate Update:*

$$P_{\tilde{y}(k)\tilde{y}(k)} = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{m}}(k|k-1)] [\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{m}}(k|k-1)]^T \quad (2.126)$$

$$P_{x(k)y(k)} = \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{X}_{i,k|k-1} - \hat{\mathbf{t}}(k|k-1)] [\mathcal{Y}_{i,k|k-1} - \hat{\mathbf{m}}(k|k-1)]^T \quad (2.127)$$

$$K(k) = P_{x(k)y(k)} P_{\tilde{y}(k)\tilde{y}(k)}^{-1} \quad (2.128)$$

$$\hat{\mathbf{t}}(k|k) = \hat{\mathbf{t}}(k|k-1) + K(k)(\mathbf{m}(k|k) + D(k) \cdot \mathbf{f}(k) - \hat{\mathbf{m}}(k|k-1)) \quad (2.129)$$

$$P(k|k) = P(k|k-1) - K(k)P_{\tilde{y}(k)\tilde{y}(k)}K(k)^T \quad (2.130)$$

where

$\gamma = \sqrt{(L + \lambda)}$ ,  $\lambda$  = composite scaling parameter,  $L$  = dimension of the state vector,  $Q(k)$  = process noise covariance matrix at scan  $k$ ,  $R(k)$  = measurement noise covariance matrix at scan  $k$ , and  $W_i$  = weights as calculated in eq.(2.116)

	Postprocessing System Type 1	Postprocessing System Type 2	Postprocessing System Type 3	Postprocessing System Type 4
Kinematic State Estimation Task	single sensor measurements	ILSE	seq. EKF	UKF
Tracking Task	LKF	LKF	seq. EKF	UKF

Table 2.2: 5 different postprocessing systems in a 2 UMR sensor network for single-target situations

### 2.2.6 Comparison of Different Kinematic State Estimation Systems

In the previous sections, two kinematic state estimation methods and three different types of Kalman filters have been presented and their properties have been outlined. Now, this section will exemplarily evaluate the kinematic state estimation plus tracking performance of four different single-target postprocessing systems consisting of the presented algorithms. This comparison will base on simulated measurements of the previously described 2 UMR sensor network. A brief overview of the four different postprocessing systems is given in table 2.2.

The *first postprocessing system* is a simple straightforward approach. Each UMR sensor is measuring one range, one azimuth angle, and one radial velocity per target. These measurements can directly be transformed to global cartesian coordinates (see section 2.1.2) and are sequentially fed in the LKF, which is fusing the measurements of the two sensors.

The *second postprocessing system* is estimating the kinematic state in global cartesian coordinates by applying the iterative least squares algorithm of section 2.1.4. The achieved, already fused estimate is then fed into the LKF where the final kinematic state estimate is retrieved.

The *postprocessing system no. 3* is fusing the two processing steps, i.e. the kinematic state estimation and the tracking filter, in a sequential EKF as it has been described in 2.2.4. All single-sensor measurements are fed sequentially into the tracking filter and the kinematic state estimate is updated in each processing step until all measurements are processed once.

The *4-th postprocessing system* consists also of a single postprocessing stage, namely the UKF presented in the previous section. It determines the kinematic state by approximating the stochastic behavior of the random variables in the system and propagating them through the true nonlinear system.

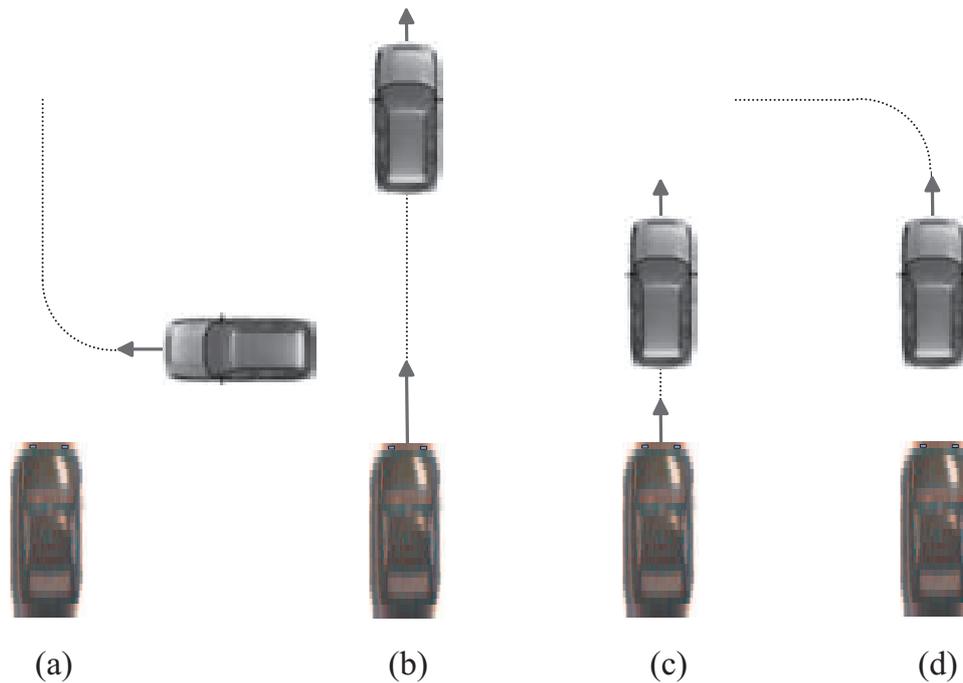


Figure 2.9: Tracking test scenario: The red car is turning into the lane of the host car and is accelerating (a). Then the host car is also accelerating (b) until it is getting close to the red car (c). After that, it is decelerating until it stops and the red car is turning to one's left (d).

### Test Scenario

The considered test scenario for comparing the different single-target postprocessing systems is the vehicle maneuver visualized in figure 2.9(a)-(d). The red car is turning into the lane of the host car with a velocity of 20 km/h and is accelerating with  $2 \text{ m/s}^2$  (a), then the host car is also accelerating with  $4 \text{ m/s}^2$  (b) until it is getting close to the red car (c). After that, both cars are decelerating with  $4 \text{ m/s}^2$ . Finally, the host vehicle is stopping and the red car is turning to the left side (d).

### Simulation Preliminaries

All postprocessing systems in this comparison will have the same simulated target measurements as inputs. Given the real target kinematic state

$$\mathbf{t} = [t_x, t_y, t_{v_x}, t_{v_y}]^T \quad (2.131)$$

the single-sensor measurements have been generated according to the following measurement equations

$$\begin{aligned}
 \mathbf{m}_r^{\text{Sens } n} &= \sqrt{(t_x - s_x^{\text{Sens } n})^2 + (t_y - s_y^{\text{Sens } n})^2} + n_r^{\text{Sens } n} \\
 \mathbf{m}_\varphi^{\text{Sens } n} &= \arctan\left(\frac{s_y^{\text{Sens } n} - t_y}{s_x^{\text{Sens } n} - t_x}\right) + n_\varphi^{\text{Sens } n} \\
 \mathbf{m}_{v_r}^{\text{Sens } n} &= \frac{t_{v_x} \cdot (t_x - s_x^{\text{Sens } n}) + t_{v_y} \cdot (t_y - s_y^{\text{Sens } n})}{\sqrt{(t_x - s_x^{\text{Sens } n})^2 + (t_y - s_y^{\text{Sens } n})^2}} + n_{v_r}^{\text{Sens } n}
 \end{aligned} \tag{2.132}$$

where  $s_x^{\text{Sens } n}$  and  $s_y^{\text{Sens } n}$  are the coordinates of the  $n$ -th sensor. The additive noise terms  $n_r^{\text{Sens } n}$ ,  $n_\varphi^{\text{Sens } n}$ , and  $n_{v_r}^{\text{Sens } n}$  are unbiased Gaussian random variables

$$E[n_r^{\text{Sens } n}] = E[n_\varphi^{\text{Sens } n}] = E[n_{v_r}^{\text{Sens } n}] = 0 \tag{2.133}$$

with standard deviations  $\sigma_r^{\text{Sens } n}$ ,  $\sigma_\varphi^{\text{Sens } n}$ , and  $\sigma_{v_r}^{\text{Sens } n}$  according to the appropriate measurement accuracies of an UMR sensor (see section 1.3.2). The number of UMR sensors in the radar network is  $N = 2$ . The measurements of these two UMR sensors are taken simultaneously. This synchronism is the ideal measurement case because the estimation accuracy is improving by a factor of  $1/\sqrt{2}$  due to the measurement redundancy.

All Kalman filters in the applied single-target postprocessing systems are based on the same underlying target dynamic model, namely the piecewise constant white noise acceleration model (see section 2.2.3) with the modeling parameter

$$\sigma_{w_a(k)} = 3 \text{ m/s} \tag{2.134}$$

The test scenario has been simulated 500 times per postprocessing system and the retrieved kinematic state estimation results have been utilized to calculate the different postprocessing systems' position estimation accuracies given by the standard deviations  $\sigma_r$  and  $\sigma_\phi$ .

## Expected Results

Before having a closer look at the estimation and tracking results of the single-target postprocessing systems, the expected performance of these systems should be discussed briefly. The relative trajectory of the considered target maneuver as it is observed by the host vehicle is shown in figures 2.10(a) and 2.10(b). In order to get an impression of the estimation accuracy gain of a single-target postprocessing system with a tracking filter, figure 2.10(a) visualizes also the positions directly retrieved from the single UMR sensor measurements (light grey dots) and 2.10(b) shows the appropriate enhanced tracking results (grey dots). But what accuracy gain can be expected from the different single-target postprocessing systems ?

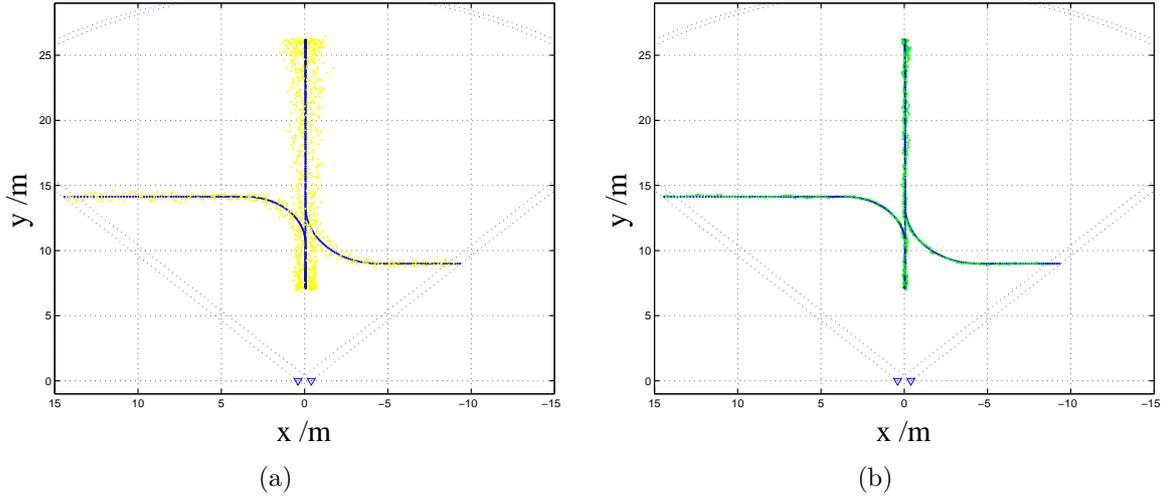


Figure 2.10: Trajectory of the tracking test scenario with the raw measurement data of the 2 UMRR sensors (a) and the processed data at the output of the tracking filter (b)

Given the single-sensor measurement accuracies

$$\begin{aligned}\sigma_r^{\text{Sens } n} &= 0.12 \text{ m} \\ \sigma_\varphi^{\text{Sens } n} &= 1^\circ\end{aligned}\quad (2.135)$$

with  $n = 1, 2$ , it can be expected that the estimation accuracies of the different postprocessing systems are significantly better. Thus, this single-sensor accuracies will serve as one reference for the systems' estimation performances.

In the ideal case of simultaneously taken measurements by the two UMRR sensors, the position estimation accuracies should improve by a factor of  $1/\sqrt{2}$  compared to the single-sensor accuracies, which yields to

$$\begin{aligned}\sigma_r^{\text{Net}} &= 0.085 \text{ m} \\ \sigma_\varphi^{\text{Net}} &= 0.707^\circ\end{aligned}\quad (2.136)$$

It can also be expected that at least these position estimation accuracies are achieved by the different radar network postprocessing systems. Thus, these values will also be considered for rating the different system performances.

But due to the incorporation of prior knowledge about the target's dynamic behavior, it can further be expected that the achieved position estimation accuracies are even better than the ones achieved due to the pure redundancy of target measurements. This accuracy gain will be investigated in the following.

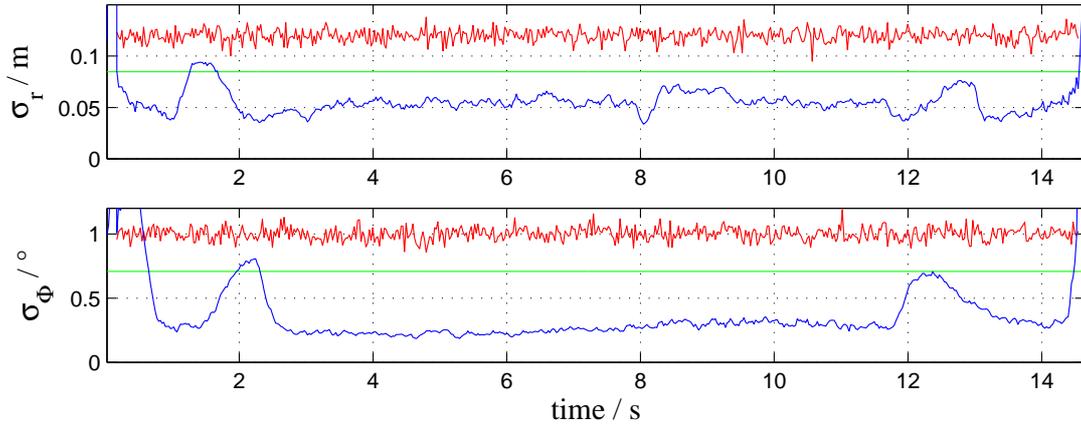


Figure 2.11: Standard deviations of the raw measurement data (upper curve) and the LKF results (lower curve) achieved by directly feeding the measurements into the filter

### Simulation Results

The first postprocessing system is directly feeding the transformed sensor measurements into the LKF and achieves the position estimation accuracy performance visualized as the standard deviations  $\sigma_r$  and  $\sigma_\phi$  (lower curves) in figure 2.11. The upper curves are denoting the achieved accuracies of the raw UMRR sensor measurements and the green line the expected improved estimation accuracy performance of a synchronized 2 UMRR sensor network without any tracking filter. As previously stated, both will serve as performance references in this comparison. The position estimation accuracy, i.e.  $\sigma_r$  and  $\sigma_\phi$ , achieved by postprocessing system no. 1 can be summarized by

$$\begin{aligned} \sigma_r &\approx 0.06 \text{ m} \\ 0.25^\circ &\leq \sigma_\phi \leq 0.3^\circ \end{aligned} \quad (2.137)$$

In the moments where the observed target is maneuvering, the accuracies are getting temporarily lower until the observed target has stopped its maneuver. At these points, the Kalman filter's target dynamic model (i.e. the process noise matrix  $Q$ ) limits the allowed kinematic state changes caused by the measurements. The observed tracking filter's latency in responding to a kinematic state change can be lowered by increasing the process noise  $Q$ , but this would also lead to a worse noise filtering performance, i.e. allowing more measurement noise to pass the filter. The latency can still be improved without increasing the noise in the kinematic state estimates. One approach is to temporarily increase the process noise during a detected target maneuver. Such an approach will be outlined later in this chapter. It will refer to this example and present the improved estimation results. Overall, the results of the direct LKF approach show that the target dynamic model in the linear Kalman filter approach leads to an improved position estimation compared to the one achieved by the pure redundancy of measurements in a radar network.

In contradiction to postprocessing system no. 1, *postprocessing system no. 2* determines

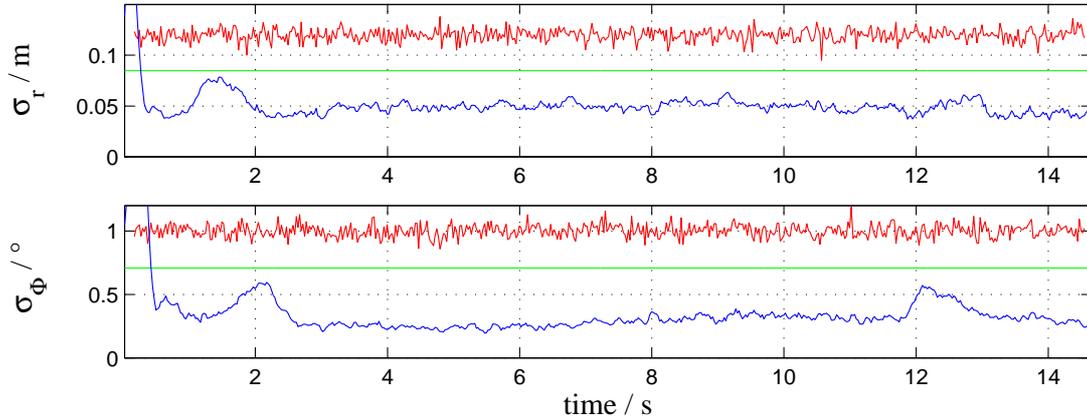


Figure 2.12: Standard deviations of the raw measurement data (upper curve) and the tracking results of the iterated least squares estimator combined with a LKF (lower curve)

the complete kinematic state estimate in an iterative least squares algorithm before feeding the obtained estimation result into the same LKF. As expected, this leads only to a slightly better estimation accuracy performance as achieved by the direct LKF approach. Figure 2.12 shows the standard deviations achieved by postprocessing no. 2, which are

$$\begin{aligned} \sigma_r &\approx 0.05 \text{ m} \\ 0.25^\circ &\leq \sigma_\phi \leq 0.3^\circ \end{aligned} \quad (2.138)$$

The range accuracy is constant at a value of  $\sigma_r \approx 0.05$  m in the regions where no relative change in direction between the detected and the host vehicle is performed. The azimuth angle accuracy in the same region lies within the range of  $0.25^\circ \leq \sigma_\phi \leq 0.3^\circ$ . If either the host or the observed vehicle is maneuvering, the achieved position estimation accuracies are lower due to the latency caused by the chosen target dynamic model. While achieving only a slightly better estimation accuracy, the kinematic state estimation achieved by the iterative least squares algorithm is significantly more complex than the coordinate transformation of the single-sensor measurements before feeding these measurements into the LKF. Applying the iterative kinematic state estimation algorithm before the tracking filter is only reasonable in a radar network, where each sensor delivers incomplete information about the target's kinematic state and only a fusion of all sensors' information leads to a complete kinematic state estimate (see HRR pulse radar network). Since the UMRR sensor is simultaneously measuring range, azimuth angle and radial velocity, i.e. the almost complete kinematic state information except the tangential velocity, the coordinate transformation of the single-sensor measurements before feeding them into the LKF is the computationally cheaper choice.

The sequential EKF in *postprocessing system no. 3* is offering the possibility of directly feeding the measurements into the tracking filter without applying any coordinate transformation. As already stated in section 2.2.4, the EKF has suboptimal estimation properties due to its linearization of the nonlinear measurement equations, in fact, it is

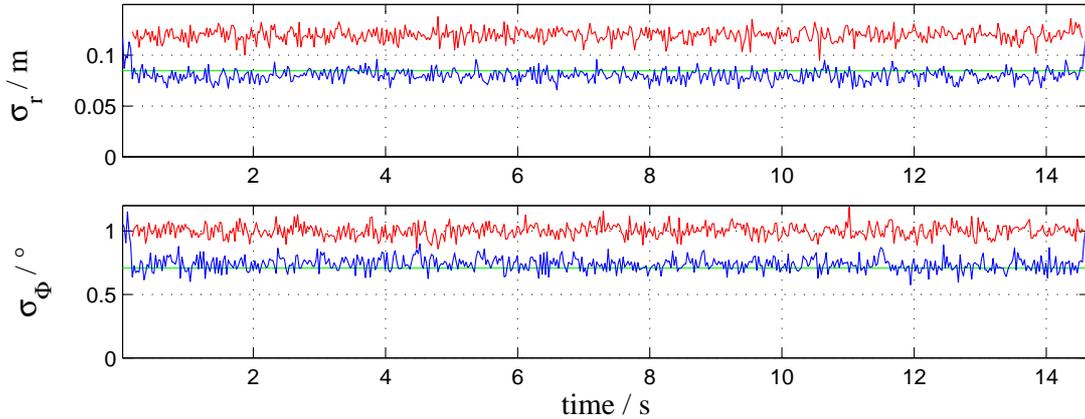


Figure 2.13: Standard deviations of the raw measurement data (upper curve) and the tracking results of the sequential EKF (lower curve)

introducing an approximation error. The accuracy results retrieved in the Monte-Carlo simulation prove this by a lower overall estimation accuracy (see figure 2.13). The achieved range and the azimuth angle accuracies are given by

$$\begin{aligned}\sigma_r &= 0.085 \text{ m} \\ \sigma_\phi &= 0.707^\circ\end{aligned}\tag{2.139}$$

which are exactly the expected values of the synchronized two-sensor network without any incorporation of a target dynamic model. The advantage of the Kalman filter's underlying target dynamic model seems to be neutralized by the linearization error in this example. Additionally, the EKF shows not the same noise filtering property with the applied target dynamic model as the preceding LKFs. But lowering the allowed target dynamics leads to worse position estimation accuracies, even to a divergence of the filter states in some cases. As a consequence, the implementation of an EKF will not be considered as an option for a postprocessing system in this thesis. In other cases, the EKF must be tested intensively for each application and the side conditions of the targeted system must be considered in detail.

*Postprocessing system no. 4* is an alternative to the EKF approach and consists of one UKF. As described in 2.2.5, the UKF is another approach of directly feeding the radar sensor measurements into the tracking filter without any preprocessing or coordinate transformation. As shown in figure 2.14, the achieved position estimation accuracies are significantly better than the ones of the EKF and almost as precise as the ones of the direct LKF approach. The second obvious observation in these curves is that the latency of the UKF in responding to a target's maneuver is higher than the one of the LKFs although all Kalman approaches are sharing the same underlying target dynamic model. But such slow response to target maneuvers will be addressed and solved later in this chapter by a maneuver adaptive approach. Overall, the UKF proves to be more stable

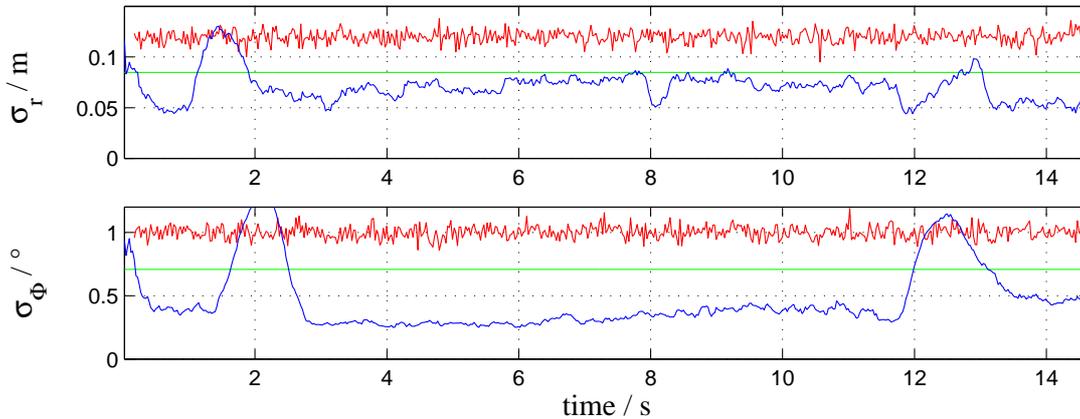


Figure 2.14: Standard deviations of the raw measurement data (upper curve) and the tracking results of the UKF (lower curve)

and also more accurate as the EKF. In regions without any change of direction in the relative motion between the two targets, it has an excellent accuracy of

$$\begin{aligned} \sigma_r &\approx 0.08 \text{ m} \\ 0.3^\circ &\leq \sigma_\phi \leq 0.4^\circ \end{aligned} \quad (2.140)$$

If a maneuver of either the observed or the own vehicle takes place, the accuracy is getting worse with standard deviations around  $\sigma_r \approx 0.12 \text{ m}$  and  $\sigma_\phi \approx 1.3^\circ$ .

Finally, it must be said that the UKF will show its true advantages if the underlying state-space description is to a great extent nonlinear. In this example, the only nonlinearity can be found in the measurement equations in eq.(2.117). Applied to a complete nonlinear systems, the UKF will outperform the EKF and the LKF will not be applicable [WM01].

Overall, this investigation has shown that the LKF has optimal estimation properties when fed by the correctly transformed measurement data. The sequential EKF has proven to be stable concerning the estimation convergence, but loses estimation accuracy due to its linear approximation of the nonlinear measurement equations. The UKF is stable according to its design and it has shown that it is significantly more accurate than the EKF, but less than the LKF approaches. It will demonstrate its full advantages if it is applied to highly nonlinear systems. As a conclusion of these Monte-Carlo simulation results, the postprocessing architecture that will be outlined in the next chapter will contain a LKF approach directly fed with transformed single-sensor measurements.

These tracking simulation results are restricted to the investigated 2 UMRR sensor radar network and may differ for other sensor constellations, e.g. for the 4 HRR radar network. The detailed investigation of a four-sensor only range measuring network can be found in [Klo02].

## 2.2.7 Filter Decoupling

Filter decoupling is of high practical importance when it comes to tracking filter implementation on an automotive processing unit with limited processing power and memory. The more state elements are included in the kinematic state vector  $\mathbf{t}$ , the higher gets the computation complexity of the Kalman filter equations, especially due to the increase of matrix dimensions. Therefore, it is desirable to decouple as many elements of the kinematic state vector as possible, so that the original Kalman filter can be split in several independent Kalman filters with fewer state elements. Depending on the choice of the target's kinematic state coordinate system, the statistical dependency between the different states is more or less significant. The process of decoupling demands that this interaction is relative insignificant, so that the estimation errors for the appropriate states are approximately uncorrelated. This corresponds to a filter state covariance matrix with the cross-covariance elements being approximately zero.

In the following, this section will consider an implementation of the 2 UMRR network as it is discussed in section 1.4. The UMRR sensors measurements are

$$\mathbf{m} = \begin{bmatrix} r \\ \varphi \\ v_r \end{bmatrix} \quad (2.141)$$

The estimate of the target's kinematic state vector in global cartesian coordinates is given by

$$\hat{\mathbf{t}} = \begin{bmatrix} \hat{t}_x \\ \hat{t}_y \\ \hat{t}_{v_x} \\ \hat{t}_{v_y} \end{bmatrix} \quad (2.142)$$

and  $P_{(x,y)}$  is its appropriate covariance matrix in cartesian coordinates. The statistically independent main axes of  $P_{(x,y)}$  can be calculated by a singular value decomposition (SVD), so that  $P_{(x,y)}$  is decomposed as

$$P_{(x,y)} = U\Sigma V^T \quad (2.143)$$

Because  $P_{(x,y)}$  is a symmetric matrix, the matrices  $U$  and  $V$  are identical, which leads to

$$P_{(x,y)} = V\Sigma V^T \quad (2.144)$$

The matrix  $V$  contains the unitary direction vectors  $\vec{u}_n$  ( $n = 1, \dots, 4$ ) of the independent main axes and  $\Sigma$  is a diagonal matrix containing the singular values, which are the variances along the directions of the main axes.

$$\Sigma(n, m) = \begin{cases} \sigma_n^2 & \text{if } n = m \\ 0 & \text{if } n \neq m \end{cases} \quad (2.145)$$

This decorrelation of filter states is optimal, but may not be of practical use when it comes to choosing the appropriate kinematic state's coordinate system. In the case of

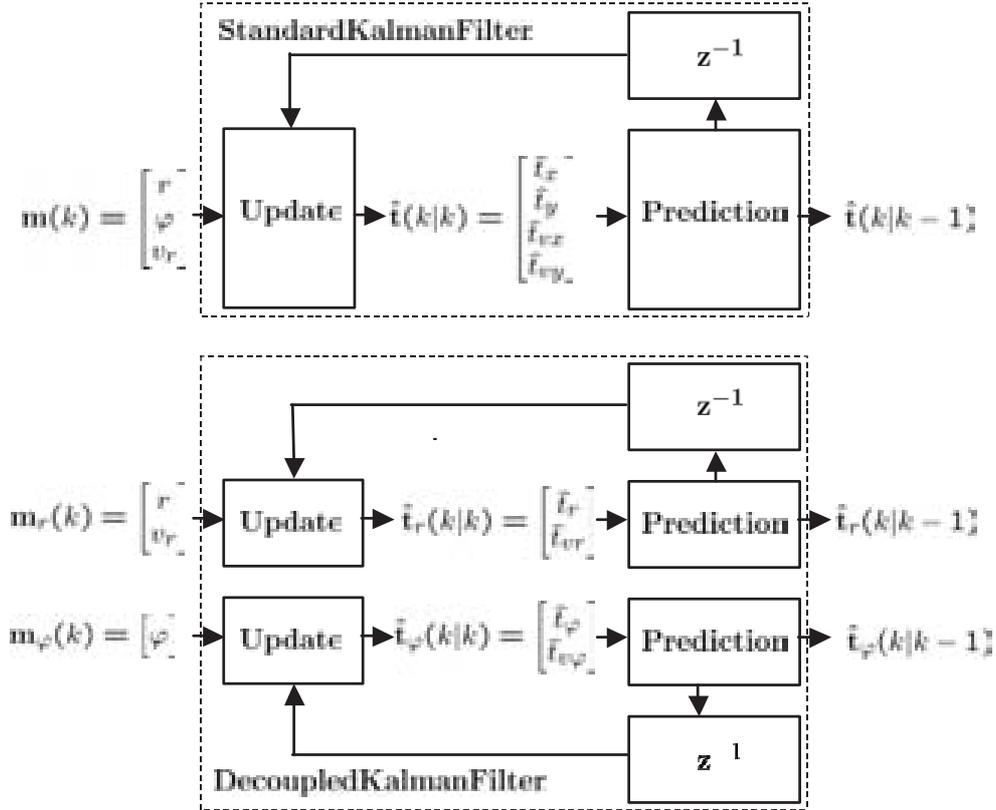


Figure 2.15: Example of a decoupled Kalman filter by choosing the state vector in polar coordinates

the discussed 2 UMR network, the choice of the filter states in polar coordinates proves to be a good one. The polar covariance matrix  $P_{r,\varphi}$  shows that the kinematic states in polar coordinates are approximately uncorrelated. The matrix  $P_{(r,\varphi)}$  can be derived from  $P_{(x,y)}$  by a simple coordinate transformation

$$P_{(r,\varphi)} = JP_{(x,y)}J^T \quad (2.146)$$

with the transformation matrix

$$J = \begin{bmatrix} \frac{\partial t_r}{\partial t_x} & \frac{\partial t_r}{\partial t_y} & \frac{\partial t_r}{\partial t_{v_x}} & \frac{\partial t_r}{\partial t_{v_y}} \\ \frac{\partial t_\varphi}{\partial t_x} & \frac{\partial t_\varphi}{\partial t_y} & \frac{\partial t_\varphi}{\partial t_{v_x}} & \frac{\partial t_\varphi}{\partial t_{v_y}} \\ \frac{\partial t_{v_r}}{\partial t_x} & \frac{\partial t_{v_r}}{\partial t_y} & \frac{\partial t_{v_r}}{\partial t_{v_x}} & \frac{\partial t_{v_r}}{\partial t_{v_y}} \\ \frac{\partial t_\omega}{\partial t_x} & \frac{\partial t_\omega}{\partial t_y} & \frac{\partial t_\omega}{\partial t_{v_x}} & \frac{\partial t_\omega}{\partial t_{v_y}} \end{bmatrix} \quad (2.147)$$

$$= \begin{bmatrix} \frac{t_x}{t_r} & \frac{t_y}{t_r} & 0 & 0 \\ -\frac{t_y}{t_r^2} & \frac{t_x}{t_r^2} & 0 & 0 \\ \frac{t_y \cdot (-t_{v_x} \cdot t_y + t_{v_y} \cdot t_x)}{t_r^3} & \frac{t_x \cdot (-t_{v_x} \cdot t_y + t_{v_y} \cdot t_x)}{t_r^3} & \frac{t_x}{t_r} & \frac{t_y}{t_r} \\ -\frac{(-2 \cdot t_{v_x} \cdot t_y \cdot t_x + t_{v_y} \cdot t_x^2 - t_{v_y} \cdot t_y^2)}{t_r^4} & -\frac{(-t_{v_x} \cdot t_y^2 + t_{v_x} \cdot t_x^2 + 2 \cdot t_{v_y} \cdot t_y \cdot t_x)}{t_r^4} & -\frac{t_y}{t_r^2} & \frac{t_x}{t_r^2} \end{bmatrix}$$

derived from the following nonlinear transformation equations

$$\begin{aligned}
 t_r &= \sqrt{t_x^2 + t_y^2} \\
 t_\varphi &= \arctan\left(\frac{t_y}{t_x}\right) \\
 t_{v_r} &= t_{v_x} \cdot \cos(t_\varphi) + t_{v_y} \cdot \sin(t_\varphi) \\
 t_\omega &= (-t_{v_x} \cdot \sin(t_\varphi) + t_{v_y} \cdot \cos(t_\varphi)) / t_r
 \end{aligned} \tag{2.148}$$

The resulting filter states in polar coordinates are approximately uncorrelated, because the appropriate covariance diagonal elements  $P_{(r,\varphi)}(n, n)$  are much greater than the non-diagonal elements  $P_{(r,\varphi)}(n, m)$

$$P_{(r,\varphi)}(n, n) \gg P_{(r,\varphi)}(n, m) \quad n \neq m \tag{2.149}$$

Having found such a good choice of the filter states, a Kalman filter containing all states can be decomposed into two independent Kalman filters as shown in figure 2.15. This also yields a decomposition of the input measurements into

$$\mathbf{m}_r = \begin{bmatrix} r \\ v_r \end{bmatrix} \quad \mathbf{m}_\varphi = [\varphi] \tag{2.150}$$

The achieved reduction in computation complexity can be demonstrated for the Kalman filter update equations. These are of order  $\mathcal{O}(n_{dim}^3)$  with  $n_{dim} = 4$  being the dimension of the original filter state vector in this derivation. The splitting of the original tracking filter into two filters with  $n_{dim} = 2$  reduces the number of operations significantly ( $4^3 - 2 \cdot 2^3 = 48$ ).

The position estimation example in the previous sections will be picked up in the following and it will be proved that decoupling of a Kalman filter can not only be considered as a very efficient way of reducing the computation complexity, it is also maintaining the tracking filter's estimation performance.

### Example for Filter Decoupling in a Multi-Sensor Radar Network

To exemplarily show the performance of a decoupled Kalman filter, the test scenario of the single-target postprocessing comparison in section 2.2.6 is picked up.

Postprocessing system no. 1 has shown an excellent performance and will be modified to a decoupled Kalman filter by choosing the filter states and the inputs of the LKF to be in polar coordinates. Thus, the LKF can be split into two nearly independent Kalman filters. The system's position estimation accuracy results have been investigated in a Monte-Carlo simulation with  $N_{MonteCarlo} = 500$  simulations. The achieved estimation accuracy results are shown in figure 2.16. The first observation is that the decoupled Kalman filter achieves approximately the same estimation accuracy performance as the non-decoupled system. The range accuracy is constant at  $\sigma_r \approx 0.06$  m and the azimuth angular standard deviation is  $\sigma_r \approx 0.25^\circ$  in the regions where no change in direction

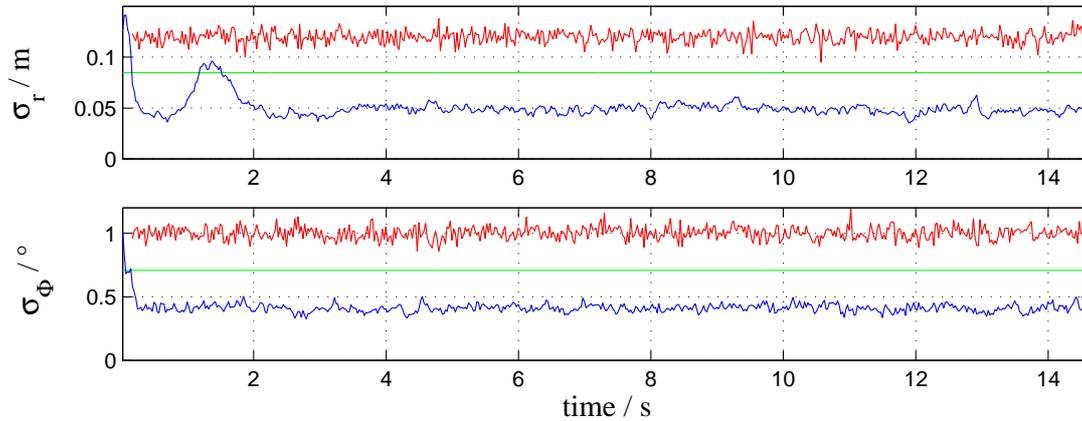


Figure 2.16: Standard deviations of the raw measurement data (upper curve) and the tracking results of postprocessing system no. 1 with a decoupled LKF (lower curve)

takes place. As already explained in the single-target postprocessing system comparison, the position estimation performance is worse during a maneuver. Comparing the decoupled and non-decoupled systems in such a situation reveals that the position estimation performance of the decoupled system is slightly worse than the one of the non-decoupled direct LKF approach. This can be explained by the approximative assumption that the polar filter states are uncorrelated. But overall, the performance of the decoupled direct LKF approach can be considered to be on the same level as the one of the non-decoupled approach. The decrease of accuracy during a maneuver can be prevented by an adaptive process noise approach that will be discussed in the following section.

### 2.2.8 Adaptive Filtering Based on Maneuver Detection

Although the considered radar sensors have a high update rate of  $f_{update} = 40$  Hz, it may occur that an observed target performs a maneuver with a large acceleration or an abrupt change of direction, so that the Kalman filter estimates are not able to follow the target's real kinematic state. In this case, the implemented target dynamic model is not allowing such an abrupt kinematic state change. But a permanent increase of the allowed target dynamics is not a solution in this situation, because it would lead to a worse filtering performance, which means allowing more measurement noise to pass the tracking filter. In order to reduce the kinematic state estimation error in these situations and to gain a quicker filter response, several methods have been suggested in the literature (e.g. [BP99], [BSLK01]). The simplest and also very efficient method is the implementation of a so-called "fudge factor". If the measurements assigned to a track differ more than usually from the expected measurements (retrieved from the track predictions), then the process noise matrix  $Q$  is increased by a certain factor (the so-called fudge factor) to enable the track update procedure to respond more quickly to the possible target maneuver. The implementation of such an algorithm is as follows:

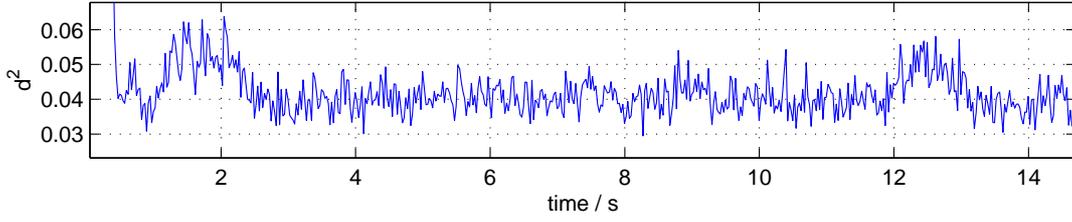


Figure 2.17: Monitored statistical square distance in the example scenario of section 2.2.6

If a target detection is associated with a track, the statistical square distance  $d^2$  between the predicted measurements of this track  $\hat{\mathbf{m}}_p$  and the real measurements  $\mathbf{m}$  must be calculated

$$d^2 = \Delta \mathbf{m}^T S^{-1} \Delta \mathbf{m} \quad (2.151)$$

where  $\Delta \mathbf{m}$  is the innovation vector

$$\Delta \mathbf{m} = \mathbf{m} - \hat{\mathbf{m}}_p \quad (2.152)$$

and  $S$  is the innovation covariance matrix

$$S_{ij} = H P H^T + R \quad (2.153)$$

$P$  is the predicted track covariance matrix transformed by the measurement matrix  $H$  to the measurement coordinate system and  $R$  is the measurement noise covariance matrix. This normalized statistical square distance is monitored in the tracking process. If it exceeds a certain threshold, a maneuver is detected and the Kalman filter switches to a "higher" process noise matrix  $Q_{\text{high}} = ff \cdot Q$  in the update procedure in order to account for the actual target maneuver. The term  $ff$  is an arbitrarily chosen "fudge factor". Such a simple process noise level switching may have  $I$  levels with  $I$  different "fudge factors"  $ff_i$ , so that  $Q$  is given by

$$Q = \begin{cases} Q_1 & \text{if } d < d_{\text{thresh}_1} \\ Q_2 & \text{if } d_{\text{thresh}_1} \leq d < d_{\text{thresh}_2} \\ \vdots & \\ Q_I & \text{if } d_{\text{thresh}_{I-1}} \leq d \end{cases} \quad (2.154)$$

The different thresholds  $d_{\text{thresh}_i}$  as well as the "fudge factors"  $ff_i$  must be adjusted to the system's kinematic state estimation behavior in real road traffic scenarios.

The weak point of the simple fudge factor method is that it relies on the statistical square distance obtained from one measurement cycle. It may occur that only one bad measurement is received and although no maneuver is taking place, the process noise matrix is increased, which leads to a bad accuracy performance of the Kalman filter. To avoid such wrong maneuver detections, it is assumed that a target maneuver will last

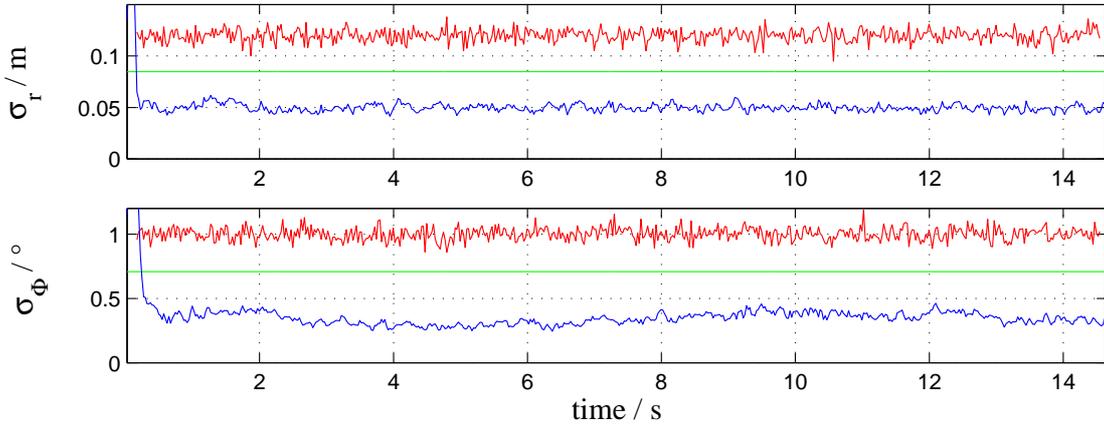


Figure 2.18: Standard deviations of the raw measurement data (upper curve) and the tracking results of the maneuver adaptive LKF (lower curve) achieved by directly feeding the measurements into the filter

over a certain number of measurement cycles, so that it will be observed in the monitored statistical square distance in consecutive cycles. To detect the onset of a target maneuver, the statistical square distance of the last  $K$  cycles will be taken into account. Such a mean statistical square distance can be calculated by

$$d_{mean}^2 = \frac{1}{K} \cdot \sum_{k=k_{actual}-K}^{k_{actual}} d^2(k) \quad (2.155)$$

This method has proven to lead to good tracking accuracy results in real road traffic scenarios. The following example will show the achieved enhancement in maneuver situations.

### Example for Maneuver Adaptive Filtering in a Multi-Sensor Radar Network

Considering the single-target postprocessing comparison scenario of section 2.2.6 and the achieved accuracy performance of the LKF in figure 2.12, it can be observed that the accuracy of the LKF position estimate at  $t \approx 2$  s is temporarily worse than the average accuracy performance. This is caused by the detected target's change of direction. If the above described statistical square distance  $d^2$  is monitored, the onset of this target maneuver can be detected as shown in figure 2.17. Implementing a simple two-level process noise with a fudge factor  $ff = 5$ , the LKF accuracy performance is significantly enhanced during the maneuver (see figure 2.18). No measurement errors due to a slow responding Kalman filter are visible.

## 2.3 Chapter Conclusions

This chapter has presented different kinematic state estimation techniques, e.g. an iterative algorithm on the basis of the principle of least squares estimation. It has outlined the Kalman filter application to the tracking task and discussed different Kalman filter types such as the LKF, the EKF, and the UKF. All these different kinematic state estimation and tracking approaches have been combined to different single-target postprocessing systems and have been compared in their performance. At this, the LKFs have shown the best state estimation accuracy. If a nonlinear relation between the measurements and the Kalman filter states cannot be solved by a coordinate transformation, then the UKF has proven to be the best choice. The UKF has shown a significantly better estimation performance than the EKF while having a lower computation complexity. Furthermore, different signal processing enhancements of the Kalman filter technique either in computation complexity achieved by filter decoupling or in estimation performance by implementing a maneuver adaptive target dynamic model have been presented. But without further signal processing steps, all presented techniques or enhancements are only applicable in single-target situations. Real road traffic scenarios are multi-target situations where the radar sensors have to deal with the reflection behavior of extended objects. At this, the received reflections from different reflection centers on a road user's surface lead to multiple target detections belonging to a single extended object. In order to apply the presented techniques to these multi-target situations, the assignments between the measurements and the reflection centers must be known. Since this is rarely the case in reality, algorithms must be designed that sort the measurement data by their origin of reflection. These algorithms are called data association methods and will be presented in the next chapter. The following chapter will develop a complete postprocessing architecture that is capable of handling multi-target situations and extended targets in real road traffic scenarios.

## Chapter 3

# A Multi-Target Postprocessing Architecture

This chapter will extend the previously presented kinematic state estimation techniques in a way that they are applicable in real multi-target situations and will present a post-processing architecture for a multi-sensor radar network. In chapter 1 typical road traffic scenarios have been discussed and it has been shown that these scenarios are multi-target situations. Furthermore, the specific reflection behavior of road users has been presented. At this, it has been stressed that road users like vehicles or trucks must be considered as extended targets that may cause more than one reflection of the radar signal. Thus, these objects are represented by more than one set of measurements in the target list of the complete radar network or even the one of a single sensor. Additionally, an extended target being observed by different sensors at slightly different aspect angles may cause that the target measurements of each sensor belong to different reflection centers on the car body. All these radar measurement properties have to be taken into account in the design of a multi-target postprocessing architecture. The previous chapter presented different kinematic state estimation methods enhancing the estimation performance of a multi-sensor radar network. But all these algorithms have been restricted to ideal single-target situations where all incoming target measurements of the different sensors belong to the same reflection center on an object. In real road traffic situations, this can neither be guaranteed in a single-target nor in a multi-target situation. Hence, algorithms are required that determine the measurements of the incoming target lists that belong to the same object. These measurements must be clustered to appropriate target groups and processed as extended objects in the following. These tasks will be addressed by the postprocessing algorithms that will be presented in this chapter. The described techniques will extend the previously discussed kinematic state estimation algorithms to a complete multi-target capable postprocessing system, which will exemplarily be derived for the implementation of a 2 UMRR network.

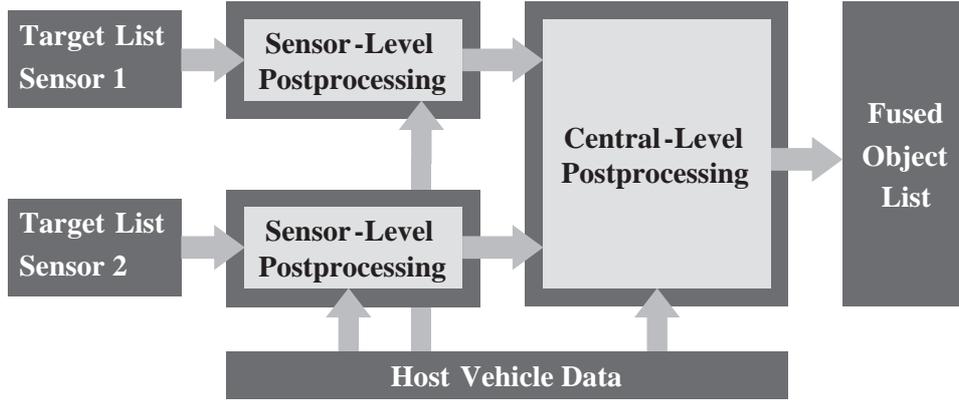


Figure 3.1: Postprocessing overview

### 3.1 Postprocessing Architecture for a 2 UMRR Network

The structure of the complete postprocessing system for a 2 UMRR network consists of two processing levels, the sensor-level and the central-level processing. This processing division is visualized in figure 3.1. The sensor-level processing will contain sensor-specific signal processing with the goal to group the detected single-sensor targets to target clusters representing an extended object and to estimate the objects' kinematic states. Such sensor-level subsystems offer also the advantages of a distributed system, which are a reduced data bus load between the different processing levels by a meaningful data reduction in each single-sensor path and a parallel single-sensor specific processing. The central-level postprocessing has mainly the task of fusing the single-sensor object estimates to a central data set containing a list of the overall detected extended objects.

Before having a closer look at the two postprocessing levels, the available information about the road traffic scenario will be briefly summarized. This information will be the basis of all decisions that will be made in the algorithms of both postprocessing levels. This available information comprises

- the target measurements of the different radar sensors,
- the measurements of the own vehicle's motion,
- and the kinematic states of the already existing object tracks in both postprocessing levels representing previously, over several scans detected objects.

The target measurements in the considered 2 UMRR network comprise one range  $m_r^{\text{Sens } n}$ , one azimuth angle  $m_\varphi^{\text{Sens } n}$ , and one radial velocity measurement  $m_{v_r}^{\text{Sens } n}$  per target, so that a single-target measurement set is given by the following vector

$$\mathbf{m}^{\text{Sens } n}(k) = \begin{bmatrix} m_r^{\text{Sens } n} \\ m_\varphi^{\text{Sens } n} \\ m_{v_r}^{\text{Sens } n} \end{bmatrix} \quad n = 1, 2 \quad (3.1)$$

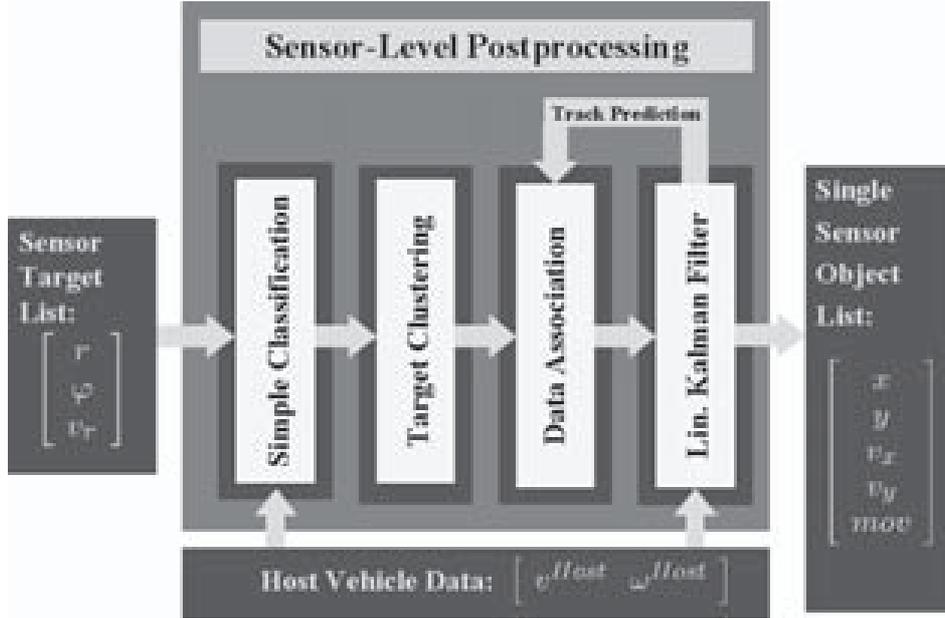


Figure 3.2: Sensor-level postprocessing structure

where  $k$  denotes the measurement cycle index and  $n$  the appropriate sensor. The available information about the own vehicle's motion can be found on the vehicle's powertrain CAN bus and is extremely manifold in today's passenger cars. The presented postprocessing system restricts itself to the velocity  $m_v^{\text{Host}}$  and the yaw rate  $m_\omega^{\text{Host}}$  of the own vehicle. This information will be treated as host vehicle measurements given in the vector  $\mathbf{m}^{\text{Host}}(k)$

$$\mathbf{m}^{\text{Host}}(k) = \begin{bmatrix} m_v^{\text{Host}} \\ m_\omega^{\text{Host}} \end{bmatrix} \quad (3.2)$$

Additionally, information about the kinematic state of previously detected objects is available as predicted sensor measurements  $\hat{\mathbf{m}}(k|k-1)$  based on the object track information of the implemented Kalman filter in both postprocessing levels

$$\hat{\mathbf{m}}^{\text{Track}}(k|k-1) = \begin{bmatrix} \hat{m}_r^{\text{Track}} \\ \hat{m}_\varphi^{\text{Track}} \\ \hat{m}_{v_r}^{\text{Track}} \end{bmatrix} \quad (3.3)$$

These three different information sources will be utilized in the algorithms of both postprocessing levels, which are described in the following sections.

## 3.2 Sensor-Level Postprocessing

The sensor-level postprocessing contains sensor-specific postprocessing stages and has the structure shown in figure 3.2. It performs the following tasks on the incoming single-sensor target list:

- **The simple classification** determines for each detected target if it is moving or not by deriving this information from the available target measurements and the host vehicle data. This distinction between moving and non-moving targets gives the following postprocessing stages additional, meaningful information, which is important for the data association, for example.
- **The target clustering** groups the target detections of one UMRR sensor according to the road user they are originating from. The goal is to obtain target groups (so-called clusters) that belong to the same passenger car or truck, for example, and thus enable a reasonable processing of extended objects.
- **The data association** decides whether a measurement cluster belongs to an existing single-sensor object track and thus is utilized to update this track, or if it represents a newly detected object and will form a new object track. Finally, all tracks will be checked whether they have recently been updated or not. If tracks have not been updated for a certain time, they will be considered as "dead" and will be deleted.
- **The linear Kalman filter** updates the existing object tracks with the previously assigned cluster information, will predict the kinematic states of the not updated tracks, and will finally predict the kinematic states of all existing tracks for the next measurement cycle.

In the following, each of these different tasks will be outlined in detail.

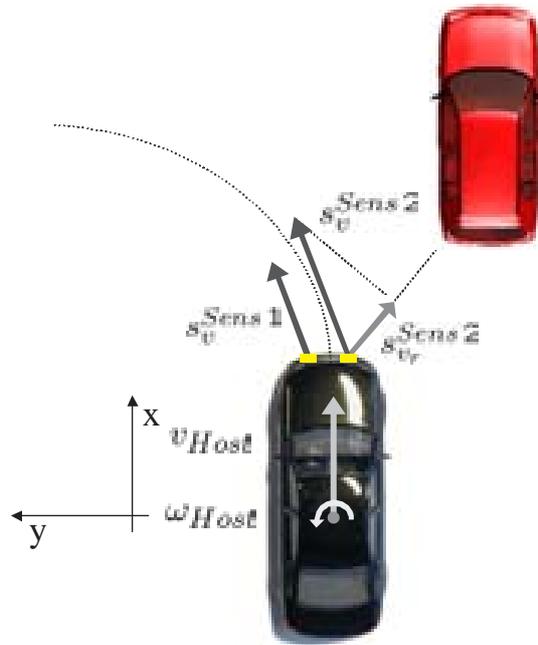


Figure 3.3: Simple classification of moving and non-moving targets

### 3.2.1 Simple Target Classification

The generation of meaningful attributes for each target simplifies the clustering and data association procedure in the different postprocessing levels and is also important for the object selection on the application processor. The simple target classification presented in the following takes advantage of the information about the own vehicle's motion by combining it with the available target measurements. It classifies the detected targets as moving or non-moving. This target distinction has proven to be important for the following data association and clustering algorithms, because the high performance UMRR sensors are detecting a great number of non-moving objects in the surrounding of the street, such as metal poles of a guard rail or parking cars. In order to achieve the distinction between this high number of non-moving and moving targets, the following calculations are performed:

The available host vehicle motion information comprises the velocity  $m_v^{\text{Host}}$  and the yaw rate  $m_\omega^{\text{Host}}$ . As visualized in figure 3.3, the velocity of the host vehicle in its central turning point (see blue point) relative to the actual host vehicle coordinate system is given by

$$\begin{bmatrix} v_{x_p}^{\text{Host}}(\Delta t) \\ v_{y_p}^{\text{Host}}(\Delta t) \end{bmatrix} = m_v^{\text{Host}} \cdot \begin{bmatrix} \cos(m_\omega^{\text{Host}} \Delta t) \\ \sin(m_\omega^{\text{Host}} \Delta t) \end{bmatrix} \quad (3.4)$$

At this,  $\Delta t$  denotes the time that passed since the last host vehicle data update. In the following, the origin of the host coordinate system is chosen to be the host vehicle's central turning point at the time of the host vehicle data reception  $\Delta t = 0$ . Then, the actual location of this central turning point at an arbitrary time  $\Delta t$  can be calculated by a simple integration of eq.(3.4). This leads to the path-prediction

$$\begin{bmatrix} x_p^{\text{Host}}(\Delta t) \\ y_p^{\text{Host}}(\Delta t) \end{bmatrix} = \frac{m_v^{\text{Host}}}{m_\omega^{\text{Host}}} \cdot \begin{bmatrix} \sin(m_\omega^{\text{Host}} \Delta t) \\ -\cos(m_\omega^{\text{Host}} \Delta t) + 1 \end{bmatrix} \quad (3.5)$$

This information about the host vehicle's driving path and velocity is utilized to evaluate whether a detected target is moving or not. At this, the velocity of the  $n$ -th UMRR sensor must be determined and compared with its measured radial target velocity. The actual location of the considered UMRR sensor at time  $\Delta t$  can be obtained from eq.(3.5) by a coordinate transformation that is taking the vehicles rotation into account:

$$\begin{bmatrix} s_x^{\text{Sens } n}(\Delta t) \\ s_y^{\text{Sens } n}(\Delta t) \end{bmatrix} = \begin{bmatrix} \cos(m_\omega^{\text{Host}} \Delta t) & -\sin(m_\omega^{\text{Host}} \Delta t) \\ \sin(m_\omega^{\text{Host}} \Delta t) & \cos(m_\omega^{\text{Host}} \Delta t) \end{bmatrix} \cdot \begin{bmatrix} s_x^{\text{Sens } n}(0) \\ s_y^{\text{Sens } n}(0) \end{bmatrix} + \begin{bmatrix} x_p^{\text{Host}}(\Delta t) \\ y_p^{\text{Host}}(\Delta t) \end{bmatrix} \quad (3.6)$$

where  $[s_x^{\text{Sens } n}, s_y^{\text{Sens } n}]_{\Delta t=0}^T$  is the position of the  $n$ -th sensor relative to the host vehicle's turning point at time  $\Delta t = 0$ . Now, the sensor's velocity can be determined by a simple differentiation of this equation, so that the sensor's velocity is given by

$$\begin{bmatrix} s_{v_x}^{\text{Sens } n}(\Delta t) \\ s_{v_y}^{\text{Sens } n}(\Delta t) \end{bmatrix} = \omega_{\text{host}} \cdot \begin{bmatrix} -\sin(m_\omega^{\text{Host}} \Delta t) & -\cos(m_\omega^{\text{Host}} \Delta t) \\ \cos(m_\omega^{\text{Host}} \Delta t) & -\sin(m_\omega^{\text{Host}} \Delta t) \end{bmatrix} \cdot \begin{bmatrix} s_x^{\text{Sens } n}(0) \\ s_y^{\text{Sens } n}(0) \end{bmatrix} + m_v^{\text{Host}} \cdot \begin{bmatrix} \cos(m_\omega^{\text{Host}} \Delta t) \\ \sin(m_\omega^{\text{Host}} \Delta t) \end{bmatrix} \quad (3.7)$$

To compare the sensor's velocity with the measured radial velocity of each target, it must be projected onto the radial interconnection between the sensor and the detected target position as it is visualized in figure 3.3

$$s_{v_r}^{\text{Sens } n}(\Delta t) = \frac{\begin{bmatrix} s_{v_x}^{\text{Sens } n}(\Delta t) & s_{v_y}^{\text{Sens } n}(\Delta t) \end{bmatrix} \cdot \begin{bmatrix} m_x^{\text{Sens } n}(\Delta t) - s_x^{\text{Sens } n}(\Delta t) \\ m_y^{\text{Sens } n}(\Delta t) - s_y^{\text{Sens } n}(\Delta t) \end{bmatrix}}{\sqrt{(m_x^{\text{Sens } n}(\Delta t) - s_x^{\text{Sens } n}(\Delta t))^2 + (m_y^{\text{Sens } n}(\Delta t) - s_y^{\text{Sens } n}(\Delta t))^2}} \quad (3.8)$$

where  $[m_x^{\text{Sens } n}(\Delta t), m_y^{\text{Sens } n}(\Delta t)]^T$  is the measured target position in cartesian coordinates at time  $\Delta t$ . If the difference between the measured radial velocity of the target  $m_{v_r}^{\text{Sens } n}(\Delta t)$  and the sensor's radial velocity component  $s_{v_r}^{\text{Sens } n}(\Delta t)$  is smaller than a certain threshold  $\Delta v_{\text{thresh}}$ , the target is considered to be non-moving.

$$\|m_{v_r}^{\text{Sens } n}(\Delta t) - s_{v_r}^{\text{Sens } n}(\Delta t)\| < \Delta v_{\text{thresh}} \quad (3.9)$$

The value of the threshold  $\Delta v_{\text{thresh}}$  must be evaluated empirically with the complete radar network in real road traffic scenarios. In case of the 2 UMRR sensor network,  $0.1 \text{ m/s} \leq \Delta v_{\text{thresh}} \leq 2.0 \text{ m/s}$  has proven to achieve a reliable separation between moving and non-moving targets.

After this processing step, the target measurements being available for the following target clustering and data association algorithm have been extended to

$$\mathbf{m}^{\text{Sens } n} = \begin{bmatrix} m_r^{\text{Sens } n} \\ m_\varphi^{\text{Sens } n} \\ m_{v_r}^{\text{Sens } n} \\ m_{mov}^{\text{Sens } n} \end{bmatrix} \quad (3.10)$$

where  $m_{mov}^{\text{Sens } n}$  is a binary status flag indicating whether the target is moving or not ( $m_{mov}^{\text{Sens } n} = \text{true} \hat{=} \text{moving target}$ ).

In the following postprocessing steps, it is important to be aware of the fact that this simple classification in moving or non-moving targets relies on the comparison of radial velocities. It may occur that an object has only a tangential velocity component and thus is wrongly classified as a non-moving target. In the considered real road traffic scenarios, this situation has proven to occur very rarely, but will be considered in the following postprocessing steps, mainly the data association, in order to avoid wrong assignments.

### Real Road Traffic Example

In order to demonstrate the high performance of this simple target classification, the following real road traffic example is presented. As shown in the video capture in figure 3.4, the host vehicle is driving in an urban environment. The two lanes allow only the driving in the host vehicle's direction. The trees on the left side separate these two lanes from the two lanes of the upcoming traffic. Parking cars can be found on the right side. Two cars are situated ahead of the host vehicle, each one driving on one lane of the road.



Figure 3.4: Road traffic scenario as seen by a video camera

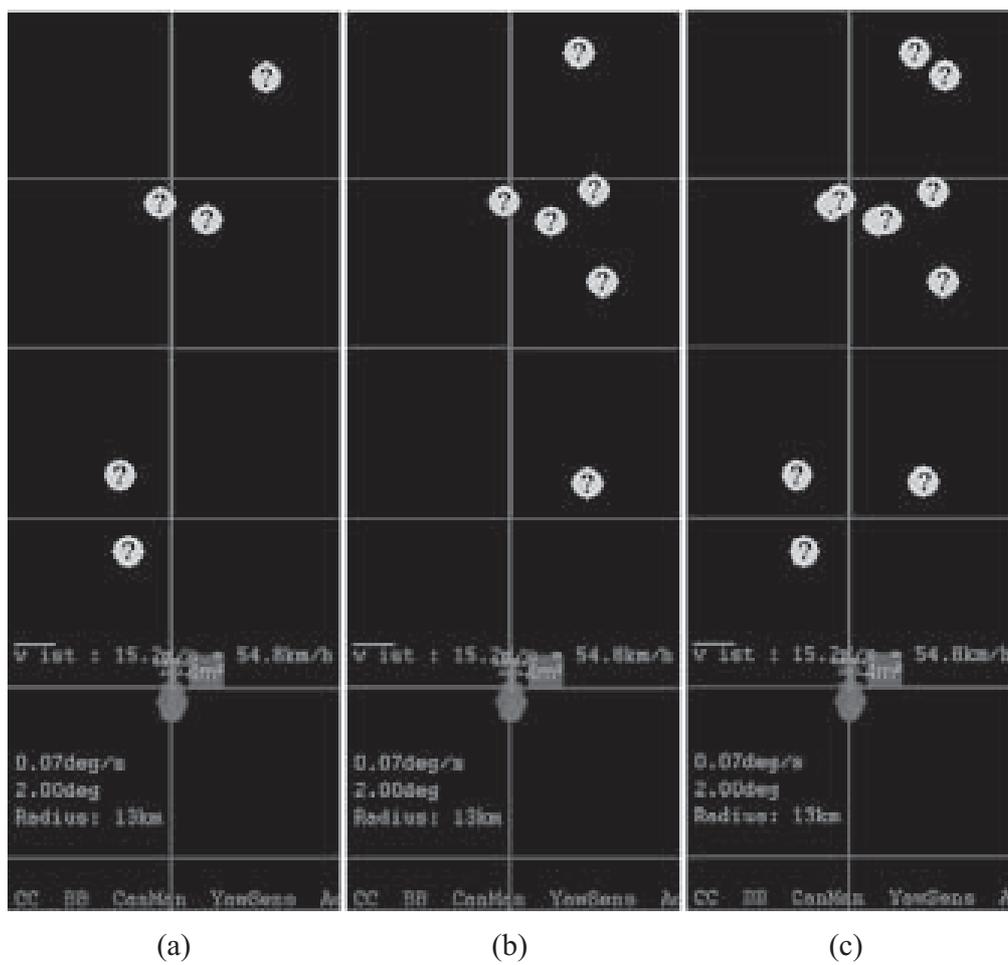


Figure 3.5: Road traffic scenario as seen by the left (a), the right (b), and both radar sensors (c)

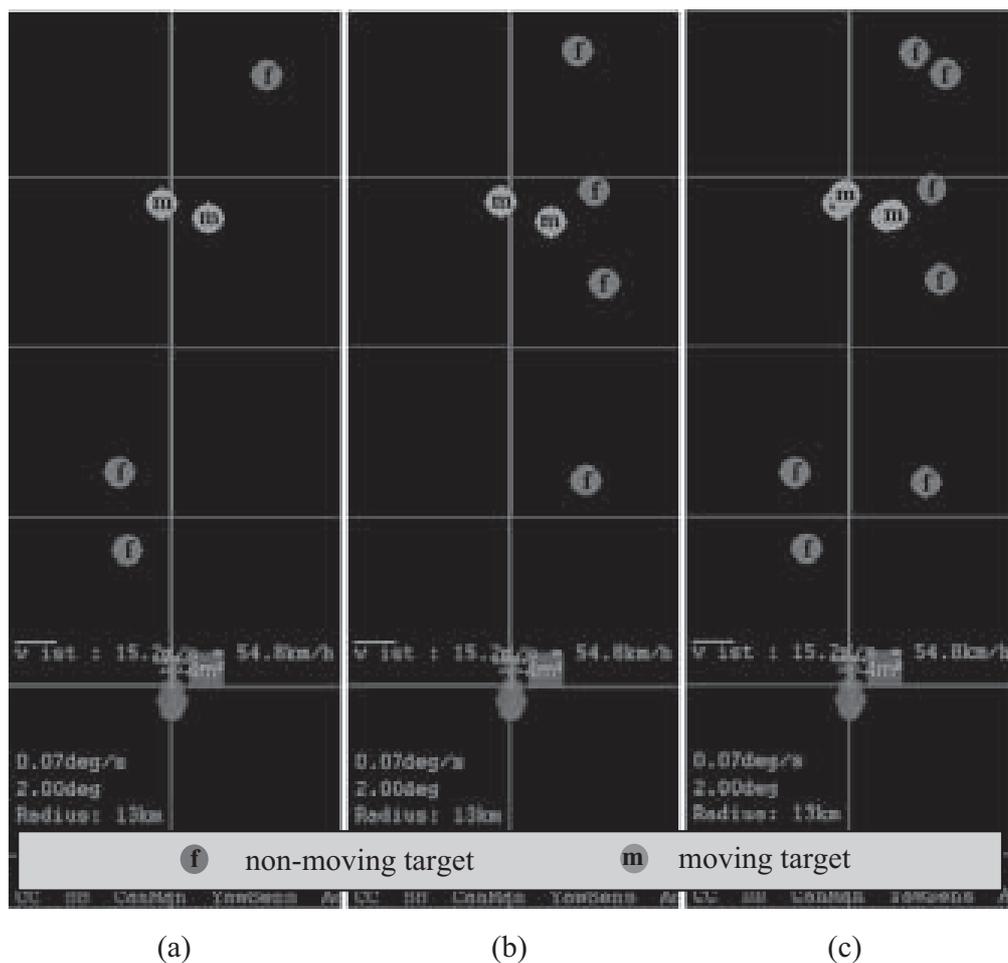


Figure 3.6: Road traffic scenario as interpreted after the simple target identification step by the left (a), the right (b), and both radar sensors (c)

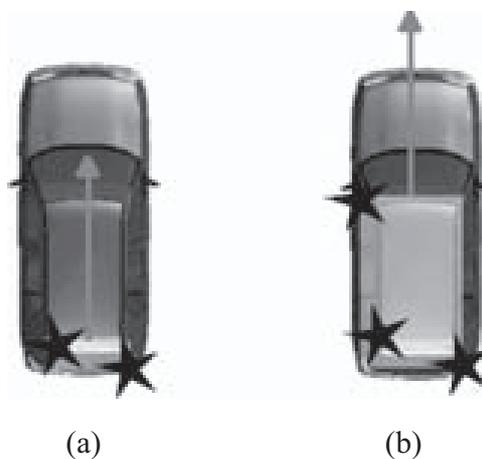


Figure 3.7: Clustering of multiple reflections and derivation of the object size

Figure 3.5 shows the target detections of the left (a), the right (b), and both UMR sensors (c). Comparing the target detections of the radar sensors with the video capture of this situation makes it easy to identify the objects for a human being. The two targets on the left side originate from the trees or some well-reflecting objects on the green turf, the two targets in the middle are the two cars in front of the own vehicle, and the detections on the right side must originate from the parking cars.

If the simple target classification is processing the incoming target lists together with the available information about the host vehicle's motion, then the distinction between moving (green circles) and non-moving targets (red circles) shown in figure 3.6 is achieved. It can be seen that a clear decision could be made for each target with the result that the two cars in the middle are the only moving objects in this scenario. Additionally, this example proves that measuring an extended object at slightly different aspect angles causes the measurements to differ from each other because the received reflections originate from slightly different reflection centers on the car body. To handle these measurement characteristic of extended objects correctly, a target clustering algorithm is applied after this simple target classification step. The following section will outline this next postprocessing step.

### 3.2.2 Sensor-Level Clustering

The task of measurement clustering arises from the fact that more than one target measurement of the actually received target list may belong to the same road user. The phenomenon that road users must be considered as extended objects has already been discussed in section 1.2.3. It may occur that multiple target measurements in the single-sensor target list belong to the same road user, but not to the same reflection center on the road user's surface. The considered UMR sensors estimate one azimuth angle per range gate, so that a detection of an extended target as shown in figure 3.7 is realistic. Figure 3.7(a) visualizes that the reception of one or two reflections may be expected from a normal passenger car. It is highly probable that a truck will generate more than two reflections. The goal of the sensor-level clustering step is to group all target detections belonging to the same road user on the basis of the available target information. Depending on the number and the fluctuation of the received reflections, the extension of the observed object can additionally be estimated. This postprocessing step also leads to a reasonable data reduction by generating clusters that comprise multiple target measurements.

#### Cluster Algorithm

Clustering techniques have widely been discussed in literature and are extensively described in [Boc74] or [Bac94], for example. One group of often applied techniques are the sequential agglomerative hierarchical non-overlapping algorithms. One derivative of

these kind of clustering algorithms is applied in the sensor-level postprocessing. It has the following processing scheme:

1. First, each of the  $I$  detected targets  $\mathbf{m}_i$  is considered as an independent cluster  $\hat{\mathbf{c}}_i$  and is represented by its estimated kinematic state including its classification as moving or non-moving

$$\hat{\mathbf{c}}_i = \mathbf{m}_i = [m_r, m_\varphi, m_{v_r}, m_{mov}]^T \quad (3.11)$$

with its appropriate covariance matrix

$$Cov(\hat{\mathbf{c}}_i) = R_i \quad (3.12)$$

2. In a second step, the cluster pair  $(i, j)|_{i \neq j}$  with the greatest similarity according to a given similarity criterion has to be found. The implemented similarity measure  $SM_{ij}$  is the inverse of the statistical distance between the two kinematic states  $\hat{\mathbf{c}}_i$  and  $\hat{\mathbf{c}}_j$  representing the two clusters:

$$SM_{ij} = \frac{1}{\sqrt{(\hat{\mathbf{c}}_i - \hat{\mathbf{c}}_j)^T \cdot (R_i + R_j)^{-1} \cdot (\hat{\mathbf{c}}_i - \hat{\mathbf{c}}_j)}} \quad (3.13)$$

3. In the following step, it has to be checked whether the pair fulfills a given homogeneity criterion or not. The homogeneity criterion of a cluster is a necessary condition that must be fulfilled by all cluster elements. In the sensor-level as well as in the central-level clustering, it is expressed by a simple thresholding utilizing the similarity measure  $SM_{ij}$

$$SM_{ij} \geq \Psi_{\text{Similar}} \quad (3.14)$$

where  $\Psi_{\text{Similar}}$  is the minimum similarity value between two elements of the same cluster. If this homogeneity criterion is fulfilled, then both clusters are fused to a single one by calculating a new cluster centroid  $\hat{\mathbf{c}}_f$  representing the enclosed target measurements:

$$\hat{\mathbf{c}}_f = R_j(R_i + R_j)^{-1}\hat{\mathbf{c}}_i + R_i(R_i + R_j)^{-1}\hat{\mathbf{c}}_j \quad (3.15)$$

$$Cov(\hat{\mathbf{c}}_f) = R_i(R_i + R_j)^{-1}R_j \quad (3.16)$$

This cluster fusion assumes that both clusters' estimation errors are uncorrelated. Finally, the number of remaining clusters is reduced to  $I - 1$ . If the homogeneity criterion is not fulfilled by the considered cluster pair  $(i, j)$ , then the cluster procedure is stopped since no other possible cluster pair will fulfill this criterion.

4. Finally, it is checked whether more than one cluster is remaining after the last processing step or not. If this is the case, the cluster procedure is continued with step no. 2, otherwise, it is stopped.

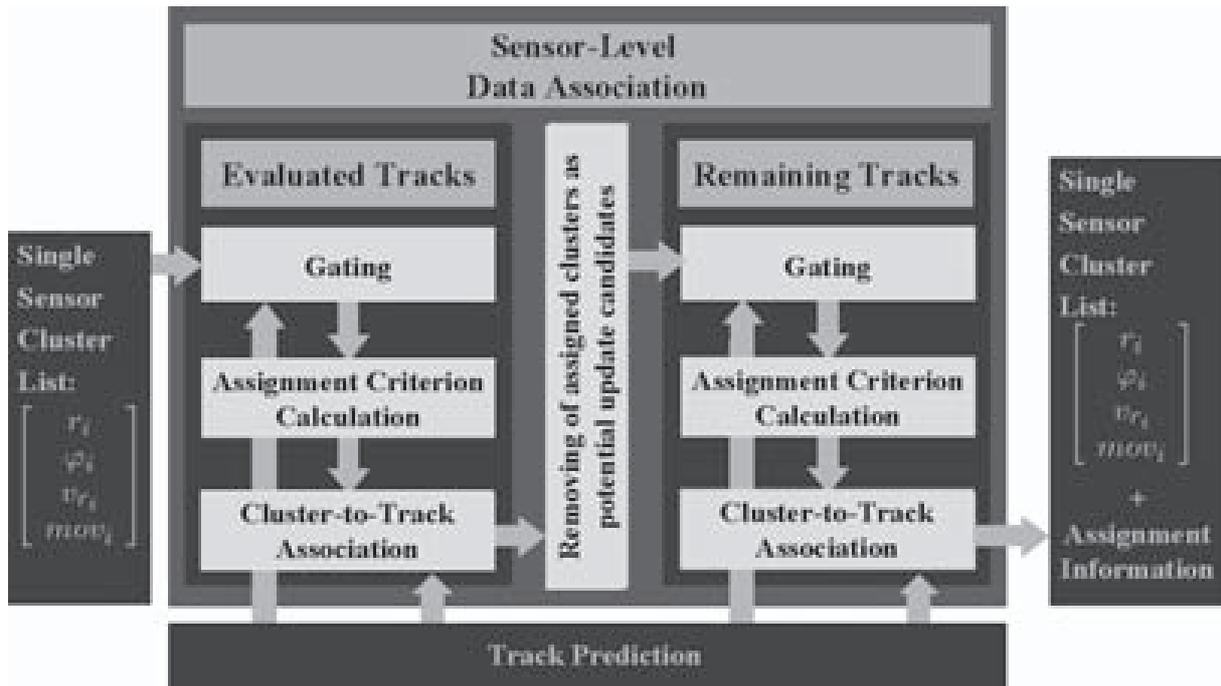


Figure 3.8: Sensor-level data association overview

The extraction of the physical extension of an observed object in such a clustering process has proven to be not very reliable. Although it was possible to determine multiple targets belonging to the same road user, the object extensions extracted from this information have been unstable. This has been caused by multiple, highly fluctuating detections in the raw measurement data from cycle to cycle. An object size estimation in the spectrum would probably lead to more stable results than the extraction from the target cluster information, because the phenomenon of an object extension can already be observed in the frequency spectrum of the down-converted LFM-CW receive signal. Extended objects can be identified in the frequency spectrum by the superposition of different spectral peaks. These peaks are forming a single broad peak cluster that appears to be too broad for a single peak.

Nevertheless, the handling of extended objects will be restricted to the described target clustering with the additional information how many targets are included in each cluster and what maximum position difference  $\Delta x$  and  $\Delta y$  lies between them. A real road traffic example that is showing the clustering performance of the complete postprocessing system will be shown later in the central-level postprocessing section.

### 3.2.3 Sensor-Level Data Association

Until now, the previously presented simple target classification and target clustering step have either generated additional target information or grouped the available target information to clusters representing different extended objects in the single-sensor cluster list. Now, the data association has to distinguish between these target clusters in order

to enable a tracking of these extended objects over time. Thus, the data association has the main task of determining the target clusters that belong to already existing object tracks and of determining the ones that are new. This decision is very important for the overall performance of the postprocessing system since it is ensuring the object track maintenance after initialization. Wrong or missed assignments will lead to a wrong estimation of object trajectories or a temporarily disappearing of an existing object track.

Many data association techniques have been published, mainly in the field of airborne radar (see [BP99], [BSF88], [BS90], [BS92], [WL90]). Most of these techniques have a very high computation complexity and base on airborne radar specific assumptions, e.g. the observation of point targets, which cannot be assumed in the field of automotive radar. A data association approach with limited computation complexity while achieving good association results under automotive radar side conditions will be outlined in the following.

The implemented sensor-level data association structure is a two-level all-neighbor data association approach. It has two main association stages, each consisting of three main processing steps as shown in figure 3.8. In the first data association stage, the incoming target clusters are assigned to the evaluated object tracks. The track status "evaluated" denotes that the tracks considered in this step have proven to originate from reliable object detections in a number of consecutive measurement cycles. After having found all possible cluster-track assignments for the evaluated object tracks, the already "utilized" clusters are removed from the list of possible track update candidates for the following, second data association stage. At this next stage, the remaining clusters are assigned in the same data association procedure to the unevaluated object tracks. This two-level distinction gives priority to the evaluated object tracks and assumes that a newly appearing object will not be assignable to an existing, evaluated object track. Thus, it is avoided that a new initiated and not reliable object track "steals" the assignment or at least takes away a significant "assignment portion" from a longer existing, evaluated object track. Within each data association stage, an all-neighbor data association approach is applied, i.e. that a cluster may be assigned to multiple tracks. Each assigned cluster is weighted by a certain factor, so that each track may be updated with multiple target clusters and thus may share the target clusters with other tracks.

The two data association stages in figure 3.8 consists of the same three processing steps. The first one is the gating. At this, the number of considered object track update candidates among the target clusters is reduced by limiting the search area around the object track's predicted kinematic state. The next step is rating the quality of the found update candidates according to a given data association criterion. Finally, the cluster-to-track association decides which target clusters are assigned to which object tracks.

The basis of the implemented data association is the available cluster data comprising the target detections from one UMR sensor, and the predicted kinematic states of the object tracks. The cluster data comprises the position  $[r, \varphi]^T$ , the radial velocity  $[v_r]$  as well as the classification  $[mov]$  as a moving or non-moving object. At cycle  $k$ , the  $i$ -th

target cluster as received from the preceding clustering step is given by

$$\hat{\mathbf{c}}_i(k) = [r, \varphi, v_r, mov]^T \quad (3.17)$$

The predicted kinematic states of the object tracks are given by

$$\hat{\mathbf{t}}_j(k|k-1) = [x, y, v_x, v_y, mov]^T \quad (3.18)$$

where the index  $j$  denotes the track among the  $J$  available object tracks.

The data association decisions can only be made on the basis of the common kinematic state information of both information sources. In this case, it is the position, the radial velocity, and the classification as moving or non-moving object. A first precondition for a cluster-to-track assignment is that both are classified as moving or non-moving. Thus, the sensor-level data association plane has 3 dimensions, namely the 2 dimensional position and the radial velocity. This information will be the basis for the different sensor-level data association stages outlined in the following.

### Gating

The first processing step in the data association procedure is the so-called gating. The goal of the gating step is the reduction of possible object track update candidates among the target clusters in order to reduce the number of comparisons between the target clusters' kinematic states and the object tracks' predicted kinematic states in the following data association stages. Such a reduction in computation complexity is reasonable and can simply be achieved by the definition of a multi-dimensional area (a so-called gate) around the predicted kinematic state values of the appropriate object track. All target clusters lying inside this gate of a predicted object track's kinematic state are considered as possible assignments and have to be rated according to a well-chosen association criterion in the next data association step.

The shape of a gate can be manifold. As already stated in the previous section, the necessary condition for a cluster-to-track assignment is the classification of both as moving or non-moving. The remaining data association plane and thus the gate belonging to a predicted object track has 3 dimensions, namely the range, the azimuth angle, and the radial velocity. The simplest approach is to define certain value ranges for each dimension of the association plane. This will lead to a gate that looks like a 3-dimensional cuboid. But it is more advantageous to take the quality of the object track's kinematic state prediction into account. The uncertainty of the track's predicted kinematic states  $\hat{\mathbf{t}}(k|k-1)$  is given by its predicted covariance matrix  $P(k|k-1)$ . If this covariance matrix indicates a high prediction accuracy, the gate is chosen to be small because it is very probable that an appropriate target cluster lies within this area. In the case where the object track's prediction is highly uncertain, the gate is chosen to be large in order to compensate former estimation errors. Such a gating behavior can be achieved by utilizing the statistical square distance  $d^2$ , which is the mathematical result of a likelihood expression. It is given by

$$d^2 = \Delta \mathbf{c}_{ij}^T S_{ij}^{-1} \Delta \mathbf{c}_{ij} \quad (3.19)$$

with  $\Delta \mathbf{c}_{ij}$  being the innovation vector between the actually detected target cluster  $\hat{\mathbf{c}}_i(k)$  and the predicted object track's kinematic state  $\hat{\mathbf{t}}_j(k|k-1)$

$$\Delta \mathbf{c}_{ij} = \hat{\mathbf{c}}_i(k) - h(\hat{\mathbf{t}}_j(k|k-1)) \quad (3.20)$$

where  $h(\dots)$  is the function transforming the object track's kinematic states to the measurement clusters coordinate system. The residual covariance matrix  $S_{ij}$  is given by

$$S_{ij} = H(k)P_j(k|k-1)H(k)^T + R_i(k) \quad (3.21)$$

with

$$H(k) = \left. \frac{\partial h(\mathbf{t})}{\partial \mathbf{t}} \right|_{\mathbf{t}=\hat{\mathbf{t}}(k|k-1)} \quad (3.22)$$

$P_j(k|k-1)$  is the track prediction covariance matrix of track  $j$  transformed by the measurement matrix  $H(k)$  to the coordinates of the cluster's kinematic states and  $R_i(k)$  is the  $i$ -th cluster's covariance matrix. The gate is then defined by a simple threshold  $d_{thresh}$

$$d \leq d_{thresh} \quad (3.23)$$

and looks like a 3-dimensional ellipsoid. All clusters satisfying this gate will be considered as possible object track update candidates and will further be tested according to the association criteria described in the following section.

### Data Association Criteria

Having determined the possible object track update candidates among the measurement clusters in the gating step, each cluster-to-track assignment must be rated according to a given data association criterion. Such a criterion must take the similarity of the cluster-track pair as well as the complete assignment situation of all actual measurement clusters and existing object tracks into account.

One simple association criterion has already been presented in the previous section, namely the statistical square distance in eq.(3.19), where it has been used to define the gate for determining the possible track update candidates. Since this similarity measure is already calculated in the gating step, the simplest approach requiring no additional computation would be to utilize this statistical distance measure to rate the association quality between an actual measurement cluster and the predicted track's kinematic state. In [BP99] this normalized squared distance is extended by an additive logarithmic factor, so that it is given for object track  $j$  and measurement cluster  $i$  by

$$d_{ij}^2_{ext} = d_{ij}^2 + \ln [\det(S_{ij})] \quad (3.24)$$

This additive factor resulting from a Gaussian likelihood calculation has the effect of penalizing tracks with greater uncertainty and thus ensures that tracks with missed updates and a resulting larger covariance matrix do not steal possible updates from higher quality tracks. But this approach does not take the complete assignment situation of all actual measurement clusters and all existing object tracks into account, only the assignment quality of a single cluster-track pair.

Another approach to rate the quality of a possible cluster-to-track assignment is the calculation of an association probability. Since this is only achieved by a complex computation (see [BP99] for details), [FBS90] suggested an approximation of the association probability by the following expression

$$Prob_{ij} = \frac{G_{ij}}{S_{tj} + S_{ri} - G_{ij} + B} \quad (3.25)$$

where  $G_{ij}$  is proportional to the Gaussian likelihood function indicating the assignment quality between object track  $j$  and measurement cluster  $i$ :

$$G_{ij} = \frac{1}{\sqrt{\det(S_{ij})}} \exp\left(-\frac{1}{2} \Delta \mathbf{c}_{ij}^T S_{ij}^{-1} \Delta \mathbf{c}_{ij}\right) \quad (3.26)$$

At this,  $\Delta \mathbf{c}_{ij}$  is the measurement cluster residual or so-called innovation vector for track  $j$  and measurement cluster  $i$  ( see eq.(3.20) ).  $S_{ij}$  is the covariance matrix of the cluster residual according to eq.(3.21). In the denominator of  $Prob_{ij}$  in eq.(3.25),  $S_{ti}$  is the sum of all  $G$ 's for track  $j$

$$S_{tj} = \sum_{i=1}^I G_{ij} \quad (3.27)$$

and  $S_{ri}$  is the sum of all  $G$ 's for measurement cluster  $j$

$$S_{ri} = \sum_{j=1}^J G_{ij} \quad (3.28)$$

Both values can be computed economically by accumulating the sums during the computation of the different  $G_{ij}$  values. The association probability  $Prob_{ij}$  was chosen in a way that it is reducing to the exact probability in the case of a single track and several measurements and to a symmetrically analogous form in the case of a single measurement and several tracks. For other cases, it is an approximation (see [FBS90] for details). The term  $B$  is a bias taking the non-unity detection probability and the presence of clutter as well as false alarms into account. In real road traffic scenarios a fixed value of  $B = 0$  has proven to lead to good results.

Compared to the association criterion in eq.(3.24), the probability  $Prob_{ij}$  in eq.(3.25) is not only taking the likelihood between object track  $j$  and measurement cluster  $i$  into account, it also considers whether that measurement correlates with other tracks, or vice versa, whether that track correlates with other measurements. The only case that the approximated association probability  $P_{ij}$  is not considering is whether other tracks correlate with other measurements, which is included in the exact equation as derived and discussed in [BP99]. But nevertheless, this approximated probability criterion is a very good compromise between data association performance and computation complexity.

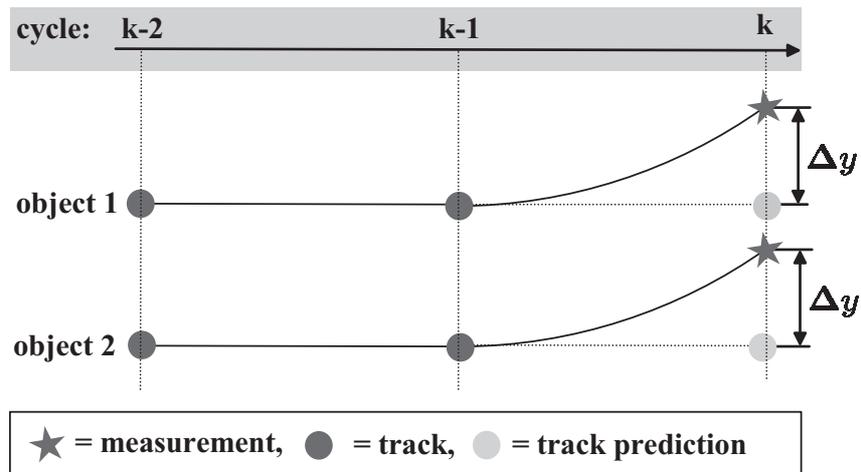


Figure 3.9: Example of two objects driving parallel and turning to one's left

*Example:*

In order to visualize the behavior of the different described association criteria, a simple example will be presented in the following. Figure 3.9 shows two detected and tracked objects driving parallel to each other with a constant velocity  $v = [14, 0]^T$  m/s. Both perform the same maneuver, a so-called coordinated turn resulting in a velocity  $v = [10, 9]^T$  m/s. This maneuver is not expected by the tracking filter. Thus, the difference between the real and the predicted position in the x-direction is given by  $\Delta x = -0.5$  m and in the y-direction by  $\Delta y = 1.15$  m. Now, the different association criteria are calculated for this given two-target situation depending on the difference  $\Delta y$ .

The figures 3.10 and 3.11 show the resulting curves of the inverse of the extended normalized squared distance as given in eq. (3.24) and the approximated assignment probability of eq.(3.25) versus the y-offset  $\Delta y$ , respectively. Each figure shows four plots that correspond to the possible combinations between the 2 measurement clusters and the 2 predicted object track positions. Depending on the offset  $\Delta y$ , each assignment criterion of the considered measurement-track pair is varying. It is easy to verify that both assignment quality measures have a maximum for the combinations (1, 1) and (2, 2) at the real track prediction error in the y-direction  $\Delta y \approx 1.15$  m. But although the correct prediction error can be determined by both association criteria, the probabilities  $Prob_{ij}$  ( $i = 1, 2$   $j = 1, 2$ ) show a clearer definition of areas where the appropriate cluster-track assignment is likely to be correct.

Because the approximated probability has shown to achieve good association results in simulations as well as real road traffic situations, it will be utilized in the experimental system presented in the next chapter.

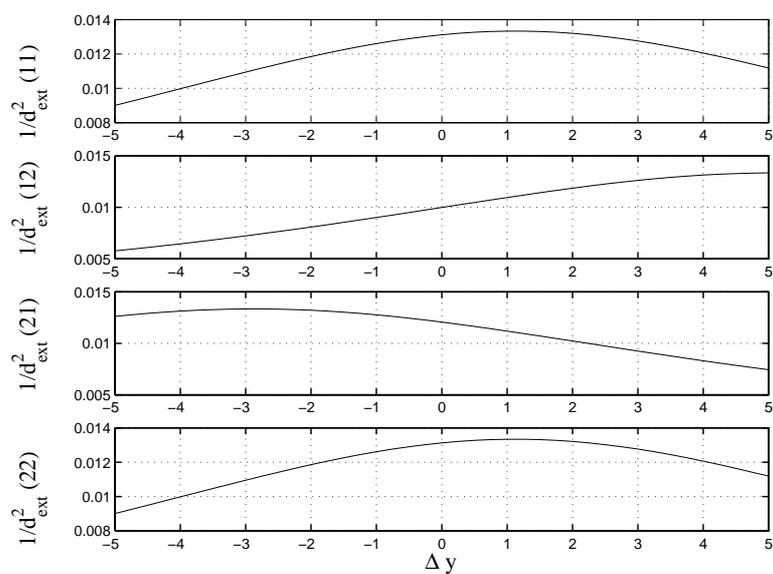


Figure 3.10: Inverse of the extended normalized square distance for the different measurement-track combinations

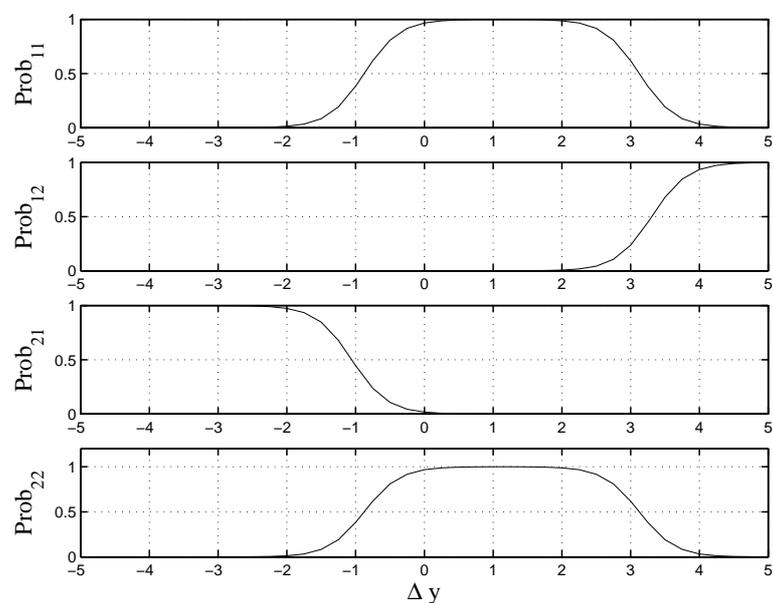


Figure 3.11: Approximated probability of the different measurement-track combinations

## Data Association Method

After the reduction of possible track update candidates among the target clusters by the application of gates and after rating the assignment quality between the considered track update candidates and object track predictions, it must be decided which of the incoming target clusters should be assigned to which object track. Therefore, two different strategies exist. Both are visualized in figures 3.13(a) and (b) for the traffic scenario example of figure 3.12:

### 1. Nearest Neighbor Data Association:

Allow each track to be updated with the "closest", most likely observation according to an arbitrarily chosen association criterion. This method is visualized in figure 3.13(a), where only the "closest" measurement clusters are chosen to update the appropriate object tracks.

### 2. All Neighbors Data Association:

Allow multiple cluster-to-track assignments weighted by a certain factor, so that each track may be updated with multiple target clusters and thus may share these target clusters with other tracks. Figure 3.13(b) shows that all measurements passing the gating test will update the appropriate track. Measurement number 2 is shared by both tracks and thus will be weighted in order to take the uncertainty of this assignment into account.

In order to decide what data association strategy is reasonable for the incoming pre-sorted cluster data at the sensor-level postprocessing, one question must be answered: Is it probable that more than one cluster belongs to an already existing track or not? Assume that the previous postprocessing stages, such as the simple target classification and the target clustering have ideally achieved their processing goal, i.e. that all targets originating from one object are grouped to a single cluster, then it is easy to conclude that only a single correct cluster-track assignment must exist for each track. Thus, the global nearest neighbor strategy would be the optimal choice. But since it may occur that the pre-sorted cluster data still contains multiple clusters belonging to the same object, the all-neighbor approach must be preferred in order to fuse all available information about one detected object to a single object track.

The all-neighbors approach that will be outlined in the following will be a simplified joint probabilistic data association (JPDA), where every object track will be updated with a weighted-average target cluster being calculated from all target clusters that passed the gating test of the appropriate object track. For the sake of completeness, it must be mentioned that the original JPDA algorithm was first presented by Y. Bar-Shalom and T.E. Fortmann and is described in detail in [BSF88].

The simplified JPDA makes use of the approximated probability of the association of track  $j$  to cluster  $i$  introduced in eq.(3.25) of the previous section. The calculation of this approximated assignment probability  $Prob_{ij}$  can be summarized as follows:

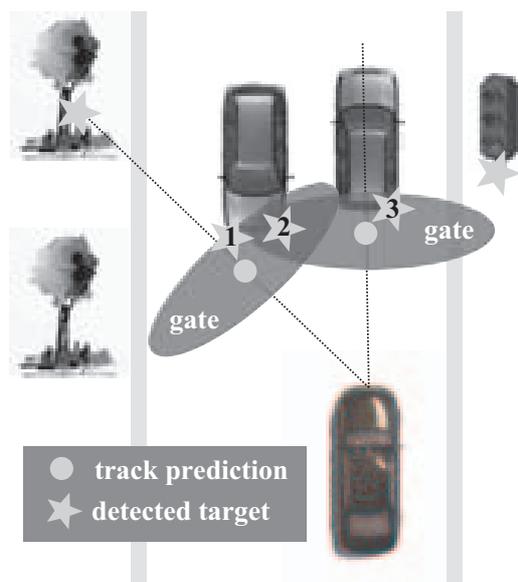


Figure 3.12: Data association traffic scenario example

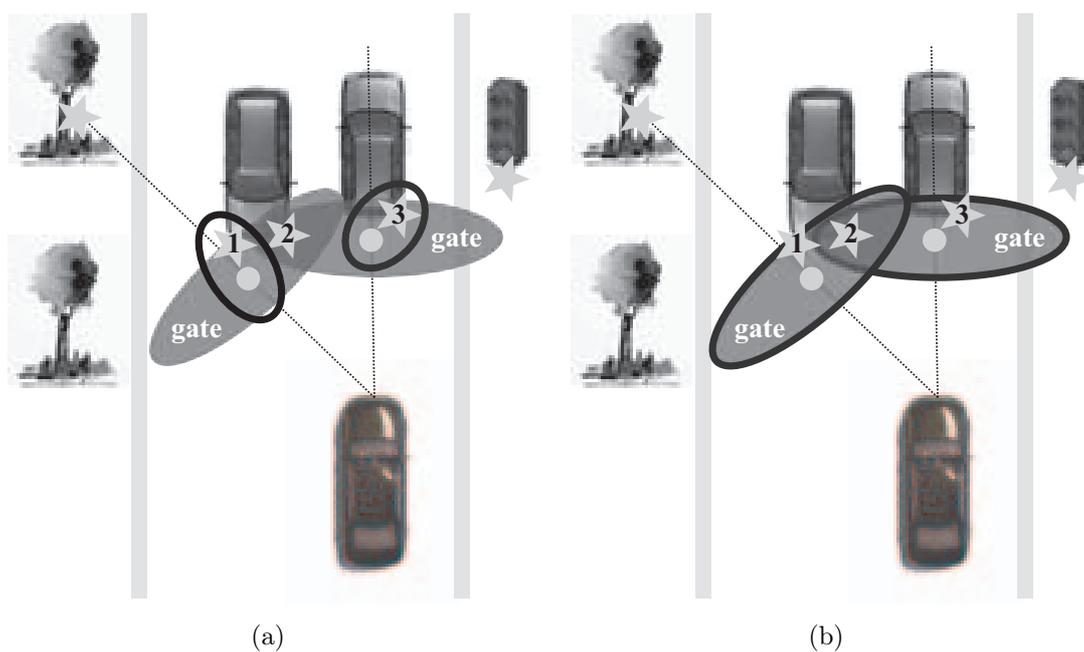


Figure 3.13: Traffic scenario example of figure 3.12 for the global nearest neighbor (b) and the all neighbors data association approach (c)

*Association Probability:*

$$Prob_{ij} = \frac{G_{ij}}{S_{tj} + S_{ri} - G_{ij} + B} \quad (3.29)$$

with

$$G_{ij} = \frac{1}{\sqrt{\det(S_{ij})}} \exp\left(-\frac{1}{2} \Delta \mathbf{c}_{ij}^T S_{ij}^{-1} \Delta \mathbf{c}_{ij}\right),$$

$$S_{tj} = \sum_{i=1}^I G_{ij} \quad \text{and} \quad S_{ri} = \sum_{j=1}^J G_{ij}$$

This association probability has been calculated for each considered cluster-track pair that passed the gating test of one of the existing object tracks, so that each track  $j$  is updated at measurement cycle  $k$  with a weighted-average cluster given by

*Weighted-Average Cluster Measurement:*

$$c_{WAj}(k) = \sum_i Prob_{ij}(k) \hat{\mathbf{c}}_i(k) \quad (3.30)$$

*Weighted-Average Covariance Matrix:*

$$P_{WAj}(k) = \sum_i Prob_{ij}(k) R_j(k) \quad (3.31)$$

The incorporation of this weighted average cluster data in the implemented Kalman filter equations will be outlined in the following track update section.

This simplified JPDA data association method is implemented in the experimental car that will be described in the next chapter. Performance measures in real road traffic scenarios will be presented and it will be referred to the assignment characteristics of this all-neighbor data association technique in order to explain the performance in the experimental results.

### 3.2.4 Sensor-Level Tracking

The sensor-level tracking procedure can be divided into two cases, namely that one or more measurement clusters have been assigned to a track with a certain probability or not. In the first case, a Kalman filter is applied and the track is updated with the appropriate measurement clusters. In the second case, the actual track prediction cannot

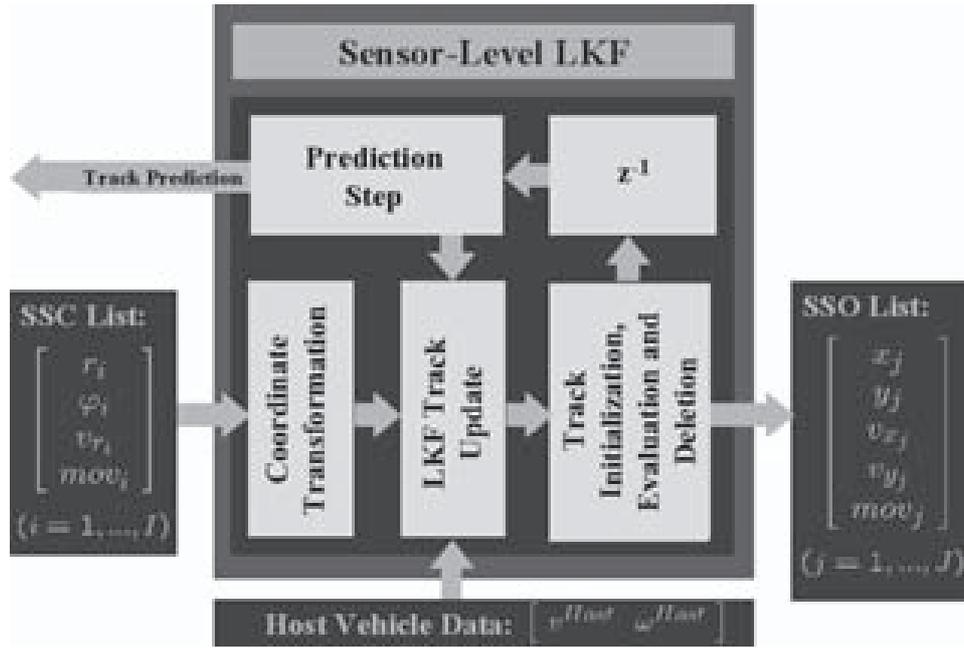


Figure 3.14: Sensor-level tracking overview

be corrected by a measurement cluster and thus must further be predicted to the next measurement cycle in order to find an assignment then.

Figure 3.14 shows the implemented sensor-level tracking algorithm. First, the incoming target single-sensor clusters (SSC) are transformed to global cartesian coordinates. In the following step, the object tracks are updated by the assigned measurement clusters. The object tracks without any measurement cluster assignment will be maintained on the basis of their prediction. After obtaining the kinematic states of all object tracks, the track data as well as the unassigned measurement clusters are passed to the track initialization, evaluation, and deletion procedure. At this, the unassigned measurement clusters are initializing new tracks, all object tracks are evaluated whether they have been updated in the last cycles or not, and the ones that have not been updated for "a long time" are deleted. Finally, the complete kinematic states of the object tracks are predicted to the time of the next update. The following sections will describe each processing step in this tracking filter procedure.

### Coordinate Transformation and Implemented Kalman Filter

The implemented Kalman filter is a simple 3-state LKF that showed an excellent performance in the last chapter when it was fed with measurements in global cartesian coordinates. Because the UMRR sensor measurements and thus the measurement cluster's kinematic states are given in polar coordinates relative to the sensor position

$$\hat{\mathbf{c}} = [\hat{c}_r, \hat{c}_\varphi, \hat{c}_{v_r}]^T \quad (3.32)$$

they must be transformed in two steps to a global cartesian coordinate system.

In the first step, the measurement clusters must be transformed to cartesian coordinates relative to the position of the measuring UMRR sensor. This transformation is achieved by the following set of nonlinear equations

$$\begin{aligned}
\hat{c}_x &= \hat{c}_r \cdot \cos(\hat{c}_\varphi) \\
\hat{c}_y &= \hat{c}_r \cdot \sin(\hat{c}_\varphi) \\
\hat{c}_{v_x} &= \hat{c}_{v_r} \cdot \cos(\hat{c}_\varphi) + \hat{c}_r \cdot \hat{c}_\omega \cdot \sin(\hat{c}_\varphi) \\
\hat{c}_{v_y} &= \hat{c}_{v_r} \cdot \sin(\hat{c}_\varphi) + \hat{c}_r \cdot \hat{c}_\omega \cdot \cos(\hat{c}_\varphi)
\end{aligned} \tag{3.33}$$

It is important to notice that the velocity in cartesian coordinates  $[\hat{c}_{v_x}, \hat{c}_{v_y}]^T$  contains only the measured radial velocity component since the tangential component cannot be measured. Thus,  $\hat{c}_\omega$  is assumed to be zero and the appropriate measurement cluster noise covariance matrix in polar coordinates  $R_{(r,\varphi)}$  must be chosen to be

$$R_{(r,\varphi)} = \begin{bmatrix} \sigma_r^2 & 0 & 0 & 0 \\ 0 & \sigma_\varphi^2 & 0 & 0 \\ 0 & 0 & \sigma_{v_r}^2 & 0 \\ 0 & 0 & 0 & \sigma_\omega^2 \end{bmatrix} \tag{3.34}$$

where  $\sigma_r$ ,  $\sigma_\varphi$ , and  $\sigma_{v_r}$  are the measurement accuracies of the measurement cluster. The variance of  $\omega$  must be set according to the uncertainty introduced by the assumption  $\hat{c}_\omega = 0$ . This variance  $\sigma_\omega^2$  can be chosen to

$$\sigma_\omega^2 = \omega_{max}^2 \tag{3.35}$$

Since the maximum occurring turning rate  $\omega_{max}$  is strongly depending on the targeted automotive application, it must be adjusted empirically in real road traffic scenarios.

The covariance matrix of the single-sensor measurement clusters in cartesian coordinates is calculated by utilizing the Jacobi matrix  $J$

$$R_{(x,y)} = JR_{(r,\varphi)}J^T \tag{3.36}$$

This transformation matrix  $J$  is given by

$$J = \begin{bmatrix} \frac{\partial \hat{c}_x}{\partial \hat{c}_r} & \frac{\partial \hat{c}_x}{\partial \hat{c}_\varphi} & \frac{\partial \hat{c}_x}{\partial \hat{c}_{v_r}} & \frac{\partial \hat{c}_x}{\partial \hat{c}_\omega} \\ \frac{\partial \hat{c}_y}{\partial \hat{c}_r} & \frac{\partial \hat{c}_y}{\partial \hat{c}_\varphi} & \frac{\partial \hat{c}_y}{\partial \hat{c}_{v_r}} & \frac{\partial \hat{c}_y}{\partial \hat{c}_\omega} \\ \frac{\partial \hat{c}_{v_x}}{\partial \hat{c}_r} & \frac{\partial \hat{c}_{v_x}}{\partial \hat{c}_\varphi} & \frac{\partial \hat{c}_{v_x}}{\partial \hat{c}_{v_r}} & \frac{\partial \hat{c}_{v_x}}{\partial \hat{c}_\omega} \\ \frac{\partial \hat{c}_{v_y}}{\partial \hat{c}_r} & \frac{\partial \hat{c}_{v_y}}{\partial \hat{c}_\varphi} & \frac{\partial \hat{c}_{v_y}}{\partial \hat{c}_{v_r}} & \frac{\partial \hat{c}_{v_y}}{\partial \hat{c}_\omega} \end{bmatrix} = \begin{bmatrix} \cos(\hat{c}_\varphi) & -\hat{c}_r \cdot \sin(\hat{c}_\varphi) & 0 & 0 \\ \sin(\hat{c}_\varphi) & \hat{c}_r \cdot \cos(\hat{c}_\varphi) & 0 & 0 \\ \hat{c}_\omega \cdot \sin(\hat{c}_\varphi) & -\hat{c}_{v_r} \cdot \sin(\hat{c}_\varphi) & \cos(\hat{c}_\varphi) & \hat{c}_r \cdot \sin(\hat{c}_\varphi) \\ \hat{c}_\omega \cdot \cos(\hat{c}_\varphi) & \hat{c}_{v_r} \cdot \cos(\hat{c}_\varphi) & \sin(\hat{c}_\varphi) & \hat{c}_r \cdot \cos(\hat{c}_\varphi) \end{bmatrix} \tag{3.37}$$

After having obtained the measurement clusters in cartesian coordinates relative to the sensor position, they must be transformed in a second step to a global, sensor independent coordinate system with an arbitrarily chosen origin of coordinates. If the measuring sensor has an azimuthal orientation  $s_\alpha^{\text{Sens}}$  and a position  $[s_x^{\text{Sens}}, s_y^{\text{Sens}}]^T$  relative to this

arbitrarily chosen origin of coordinates, then the measurement clusters in global cartesian coordinates  $\hat{\mathbf{c}}^{\text{glob}}$  can be calculated by the following linear transformation

$$\hat{\mathbf{c}}^{\text{glob}} = \begin{bmatrix} \cos(s_\alpha^{\text{Sens}}) & -\sin(s_\alpha^{\text{Sens}}) \\ \sin(s_\alpha^{\text{Sens}}) & \cos(s_\alpha^{\text{Sens}}) \end{bmatrix} \cdot \begin{bmatrix} \hat{c}_x \\ \hat{c}_y \\ \hat{c}_{v_x} \\ \hat{c}_{v_y} \end{bmatrix} + \begin{bmatrix} s_x^{\text{Sens}} \\ s_y^{\text{Sens}} \\ 0 \\ 0 \end{bmatrix} \quad (3.38)$$

The appropriate covariance matrix  $R^{\text{glob}}$  is given by

$$R^{\text{glob}} = \begin{bmatrix} \cos(s_\alpha^{\text{Sens}}) & -\sin(s_\alpha^{\text{Sens}}) \\ \sin(s_\alpha^{\text{Sens}}) & \cos(s_\alpha^{\text{Sens}}) \end{bmatrix} \cdot R_{(x,y)} \cdot \begin{bmatrix} \cos(s_\alpha^{\text{Sens}}) & -\sin(s_\alpha^{\text{Sens}}) \\ \sin(s_\alpha^{\text{Sens}}) & \cos(s_\alpha^{\text{Sens}}) \end{bmatrix}^T \quad (3.39)$$

The target measurement clusters in global cartesian coordinates  $\hat{\mathbf{c}}^{\text{glob}}$  with the appropriate measurement noise covariance matrix  $R^{\text{glob}}$  can directly be fed into the simple LKF. For the sake of simplicity, the superscripts "glob" will be omitted in the following, so that  $\hat{\mathbf{c}}$  and  $R$  are the single-sensor clusters in global cartesian coordinates. Note, that the measurement matrix  $H$  becomes the unity matrix since the coordinate system of the measurement clusters has been chosen to be the same as the one of the LKF's estimated kinematic state vector. Thus, the LKF equations are reducing significantly in their computation complexity. Taking the cluster-track assignment probabilities  $Prob_{ij}$  of the data association section into account, the LKF equations of track  $j$  at cycle  $k$  are given by

*LKF State Estimate Prediction:*

$$P_j(k|k-1) = \Phi(k)P_j(k-1|k-1)\Phi(k)^T + Q \quad (3.40)$$

$$\hat{\mathbf{t}}_j(k|k-1) = \Phi(k)\hat{\mathbf{t}}_j(k-1|k-1) \quad (3.41)$$

*LKF State Estimate Update:*

$$K_j(k) = P_j(k|k-1) \left( P_j(k|k-1) + \sum_i Prob_{ij}(k) \cdot R_i(k) \right)^{-1} \quad (3.42)$$

$$P_j(k|k) = (I - K_j(k)) \cdot P_j(k|k-1) \quad (3.43)$$

$$\hat{\mathbf{t}}_j(k|k) = \hat{\mathbf{t}}_j(k|k-1) + K_j(k) \cdot \sum_i Prob_{ij}(k) \cdot \Delta \mathbf{c}_{ij}(k) \quad (3.44)$$

where  $\Delta \mathbf{c}_{ij}(k)$  is the innovation between the object track prediction  $\hat{\mathbf{t}}_j(k|k-1)$  and the cluster measurement  $\hat{\mathbf{c}}_i(k)$

$$\Delta \mathbf{c}_{ij}(k) := \hat{\mathbf{c}}_i(k) - \hat{\mathbf{t}}_j(k|k-1) \quad (3.45)$$

The implemented LKF incorporates the information about the host vehicle's motion described in 2.1.5 as well as the maneuver adaptive process noise as presented in 2.2.8.

### Missed Update

If no measurement cluster has been assigned to an existing track, the track will be maintained according to the following equations:

*LKF State Estimate Prediction:*

$$P_j(k|k-1) = \Phi(k)P_j(k-1|k-1)\Phi(k)^T + Q \quad (3.46)$$

$$\hat{\mathbf{t}}_j(k|k-1) = \Phi(k)\hat{\mathbf{t}}_j(k-1|k-1) \quad (3.47)$$

*LKF State Estimate Update:*

$$P_j(k|k) = P_j(k|k-1) \quad (3.48)$$

$$\hat{\mathbf{t}}_j(k|k) = \hat{\mathbf{t}}_j(k|k-1) \quad (3.49)$$

At this, the increase of the track's covariance matrix  $P_j(k-1|k-1)$  by the process noise covariance matrix  $Q$  in the prediction step is not compensated by an assigned measurement cluster in the following LKF update step. Thus, a track without any assigned cluster is automatically penalized by a higher uncertainty. If this increase in the covariance proves to be too low, then each track without an assignment can additionally be penalized by an arbitrarily chosen covariance matrix  $Z$  in the LKF update step, so that (3.48) becomes

$$P_j(k|k) = P_j(k|k-1) + Z \quad (3.50)$$

### Track Initialization, Evaluation, and Deletion

The straight-forward track-initiation and evaluation scheme that will be presented in the following has proven to work well in real road traffic scenarios. At this, the clustered targets that are not assigned to an existing track are considered as 'unused'. Each of these unassigned target clusters will initiate a new object track with the status of not being evaluated yet. Such potential object tracks will take part with a low priority in the cluster-to-track association procedure in the following measurements cycles (see section 3.2.3 for details). If the unevaluated tracks are updated an arbitrary number  $N_{eval}$  of times in  $K_{check}$  consecutive measurement cycles, they will change their status to 'evaluated'. Figure 3.15 shows an 'seven-out-of-eleven' track initiation with a sliding evaluation window, for example.

A similar procedure can be applied for track deletion. If fewer track updates than the given number  $N_{eval}$  are possible in the last  $K_{check}$  measurement cycles, the track will lose its status as evaluated. But it will not directly be deleted, instead, it will 'survive' as an unevaluated track and still take part in the association procedure with a low priority for

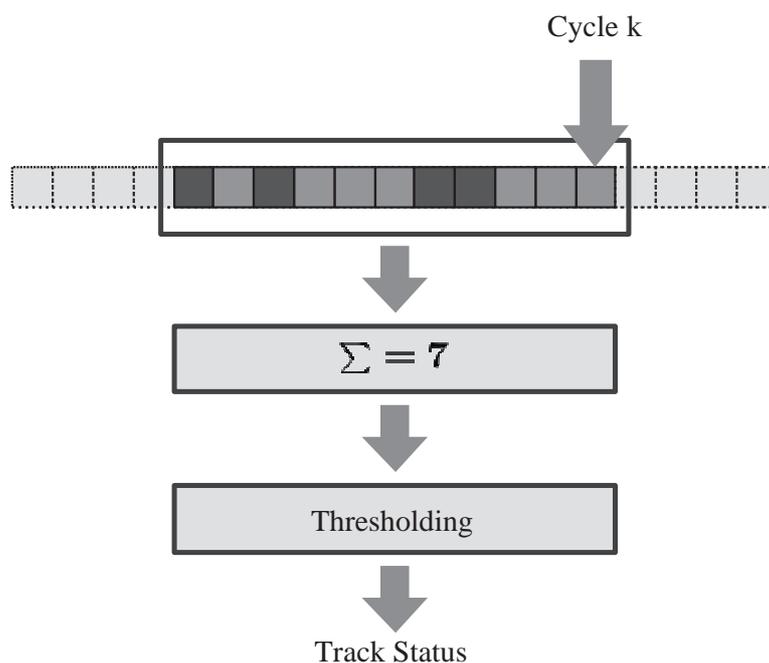


Figure 3.15: Track evaluation by number of detections in consecutive measurement cycles

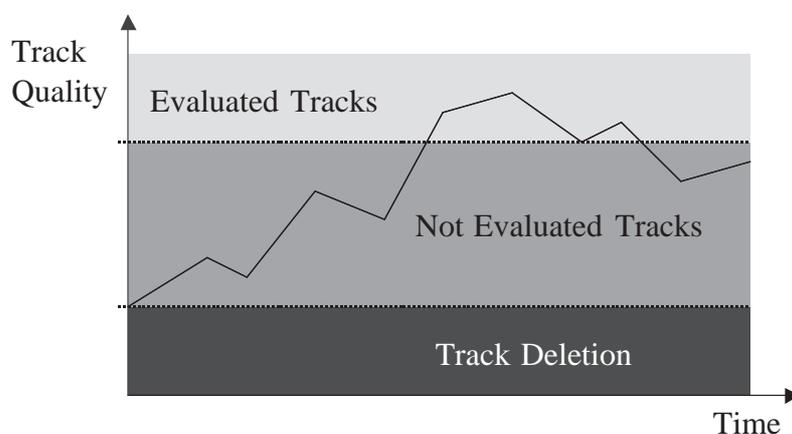


Figure 3.16: Track evaluation with track quality monitoring

the next  $K_{del}$  cycles. If it is not updated during this period, it will be deleted. Otherwise it can regain the status 'evaluated'.

If the number of possible track updates in the last  $K_{check}$  measurement cycle is considered as the actual track quality, then a time-dependent track quality function as it is shown in figure 3.16 can be established. The two conditions when a track is considered as evaluated or has to be deleted can be visualized as a track evaluation and deletion region.

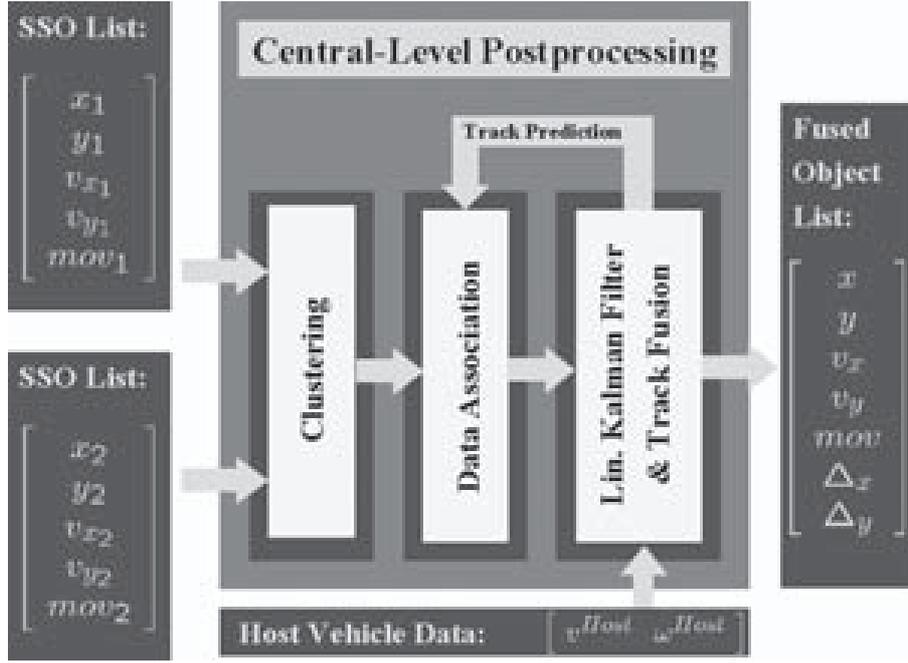


Figure 3.17: Central-level postprocessing structure

The choice of the parameters  $N_{eval}$ ,  $K_{check}$  and  $K_{del}$  in this two-level postprocessing architecture will be discussed together with the central-level track evaluation and deletion parameters in section 3.4 at the end of this chapter.

### 3.3 Central-Level Postprocessing

Having presented the different sensor-level postprocessing stages, the following sections will outline the appropriate central-level algorithms. At this, each section will underline the similarities and emphasize the differences between both postprocessing levels.

The output of each sensor-level postprocessing system will be a single-sensor object (SSO) list with the appropriate kinematic state estimates including a status-bit  $\hat{t}_{mov}$  denoting if the object is considered to be moving or non-moving.

$$\hat{\mathbf{t}}^{SSO}(k) = \begin{bmatrix} \hat{t}_x^{SSO} \\ \hat{t}_y^{SSO} \\ \hat{t}_{v_x}^{SSO} \\ \hat{t}_{v_y}^{SSO} \\ \hat{t}_{mov}^{SSO} \end{bmatrix} \quad (3.51)$$

This SSO list will be transmitted to the central-level postprocessing unit as it is shown in figure 3.17. This unit performs the following tasks:

- **The central-level object clustering** groups the objects from the latest SSO lists of the 2 UMRR sensors according to the road user they are originating from. The goal is to obtain object clusters containing objects from all radar sensors' SSO lists.
- **The central-level data association** decides whether an actual object cluster belongs to an existing central-level object track and thus is utilized to update this track or represents a newly detected object and thus will form a new central-level track. Finally, all object tracks will be checked whether they have recently been updated or not. If tracks have not been updated for a certain time, they will be deleted.
- **The linear Kalman filter and data fusion stage** updates the existing central-level object tracks with the previously assigned object clusters, will predict linearly the kinematic states of the not updated tracks, and will finally predict the kinematic states of all existing tracks for the next measurement cycle.

These different central-level postprocessing tasks show that the central-level clustering, the data association, and the tracking filter have the same basic functionality as the appropriate sensor-level stages. Thus, the following descriptions of these central-level postprocessing steps will strongly refer to the sensor-level descriptions and emphasize the important differences.

The output of the central-level postprocessing and thus of the complete postprocessing architecture is a fused object list (FOL), which contains the objects' kinematic states, their classification as moving or non-moving, and their physical extension, i.e. the width and depth according to the multiple measurements originating from the same object:

$$\hat{\mathbf{t}}^{\text{FOL}}(k) = \begin{bmatrix} \hat{t}_x^{\text{FOL}} \\ \hat{t}_y^{\text{FOL}} \\ \hat{t}_x^{\text{FOL}} \\ \hat{t}_y^{\text{FOL}} \\ \hat{t}_{v_x}^{\text{FOL}} \\ \hat{t}_{v_y}^{\text{FOL}} \\ \hat{t}_{mov}^{\text{FOL}} \\ \hat{t}_{\Delta x}^{\text{FOL}} \\ \hat{t}_{\Delta y}^{\text{FOL}} \end{bmatrix} \quad (3.52)$$

This fused object list can be transmitted to an application processor where the different objects can be sorted according to their priority for the targeted application. Then, the important objects have to be selected and appropriate actions must be taken.

### 3.3.1 Central-Level Clustering

The central-level clustering is the first central-level postprocessing stage and receives the single-sensor object (SSO) lists from the different single-sensor paths. The main task of the central-level clustering is similar to the one of the sensor-level target clustering, namely to determine the single-sensor objects in the different SSO lists that originate from the same road user and to group them to global object clusters. The difference

to the sensor-level clustering approach is that the kinematic states of the single-sensor objects originating from the same road user cannot be considered as uncorrelated. These kinematic state estimates base on the target dynamic model implemented in the sensor-level Kalman filter in each sensor path. Thus, the single-sensor objects representing the same road user share a common target dynamic model. This leads to a correlation of the appropriate SSO kinematic state estimates and must be taken into account when it comes to testing and fusing the different global cluster candidates. The following central-level cluster processing scheme is implemented in the postprocessing architecture:

### Cluster Algorithm

1. First, each of the  $I$  detected single-sensor objects of all SSO lists is considered as an independent global object cluster  $\hat{\mathbf{c}}_i$  and is represented by its estimated kinematic states including its classification as moving or non-moving

$$\hat{\mathbf{c}}_i = \hat{\mathbf{t}}_i = [\hat{t}_x, \hat{t}_y, \hat{t}_{v_x}, \hat{t}_{v_y}, t_{mov}]^T \quad (3.53)$$

with its covariance matrix

$$Cov(\hat{\mathbf{c}}_i) = P_i \quad (3.54)$$

2. In a second step, the cluster pair  $(i, j)|_{i \neq j}$  with the greatest similarity according to a given similarity criterion has to be found. The implemented similarity measure  $SM_{ij}$  is the inverse of the statistical distance between the two kinematic states  $\hat{\mathbf{c}}_i$  and  $\hat{\mathbf{c}}_j$  representing the two clusters:

$$SM_{ij} = \frac{1}{\sqrt{(\hat{\mathbf{c}}_i - \hat{\mathbf{c}}_j)^T \cdot (P_i + P_j - P_{ij} - P_{ij}^T)^{-1} \cdot (\hat{\mathbf{c}}_i - \hat{\mathbf{c}}_j)}} \quad (3.55)$$

where  $P_{ij}$  is the cross covariance matrix being a measure of the correlation between the global object clusters  $\hat{\mathbf{c}}_i$  and  $\hat{\mathbf{c}}_j$ . On details how this cross covariance matrix  $P_{ij}$  can be obtained from information of the preceding postprocessing steps, please consider the appendix A, where the general minimum variance data fusion approach is outlined in detail.

3. In the following step, it has to be checked whether the pair fulfills a given homogeneity criterion or not. The homogeneity criterion of a cluster is a necessary condition that must be fulfilled by all cluster elements. In the central-level clustering, it is given by a simple thresholding utilizing the similarity measure  $SM_{ij}$

$$SM_{ij} \geq \Psi_{\text{Similar}} \quad (3.56)$$

At this,  $\Psi_{\text{Similar}}$  is the minimum similarity value between two elements of the same cluster. If this homogeneity criterion is fulfilled, then both clusters are fused by calculating a new cluster centroid  $\hat{\mathbf{c}}_f$  representing the enclosed single-sensor objects.

$$\hat{\mathbf{c}}_f = \hat{\mathbf{c}}_i + [P_i - P_{ij}] \cdot [P_i + P_j - P_{ij} - P_{ij}^T]^{-1} \cdot [\hat{\mathbf{c}}_j - \hat{\mathbf{c}}_i] \quad (3.57)$$

$$Cov(\hat{\mathbf{c}}_f) = P_i - [P_i - P_{ij}] \cdot [P_i + P_j - P_{ij} - P_{ij}^T]^{-1} \cdot [P_i - P_{ij}]^T \quad (3.58)$$

This cluster fusion is calculated in the minimum variance sense and is derived in detail in the appendix A. Finally, the number of remaining global object clusters is reduced to  $I - 1$ . If the homogeneity criterion is not fulfilled by the considered cluster pair  $(i, j)$ , then the cluster procedure is stopped since no other possible cluster pair will fulfill this criterion.

4. Finally, it is checked whether more than one cluster is remaining after the latest processing step or not. If yes, the cluster procedure is continued with step no. 2, otherwise, it is stopped.

As already stated in the sensor-level target clustering section, the extraction of the physical extensions of an observed object in such a clustering process has proven to be not reliable. Such an extraction must be carried out directly in or after the spectral estimation in the sensor front end, where it will lead to more stable results. Thus, the handling of extended objects in this thesis will be restricted to the described clustering process with the additional information how many targets are included in each cluster and what maximum position difference  $\Delta x$  and  $\Delta y$  lies between them. The following real road traffic example will show the achieved central-level clustering results.

### Real Road Traffic Example

The real road traffic situation shown in figure 3.18 includes a truck on the right lane of a highway directly in front of the host vehicle and a passenger car overtaking this truck on the left lane. The two UMRR sensors are detecting five moving targets (see 3.19(a)), four with almost the same speed as the host vehicle and one with a significant larger speed. The four detections directly in front of the passenger car belong to two trucks driving in front of the own vehicle and thus are clustered by the previously described algorithm (see figure 3.19(b)). The overtaking passenger car is only observed by a single target detection and cannot be clustered with any other detection. It can be seen that all targets with a certain similarity are clustered and fused to one object with certain extensions. These extensions are given in the x- and y-direction of the object's coordinate system with the x-axis in the direction of the object's velocity. In this example, the clustering is reducing the number of objects to 3, which represents the real number of road users present in this road traffic scenario snapshot. This example shows that the determination of object extensions depends heavily on the position of the detected reflection centers on the car body. For example, the two trucks in front of the own vehicle have about the same extension in reality. But due to the limited number of detected targets, it is not possible to estimate this real extension. The estimated extension can be considered as the object size that can be at least assumed for the detected object, i.e. it cannot be determined whether the object extensions are larger than the estimated one or not. Different test drives in real road traffic situations have proven that a strongly fluctuating number of target detections is causing quickly changing object extensions that cannot be smoothed in a reasonable way by a tracking filter. The object extension estimation may be significantly improved by finding peak clusters in the frequency spectrum or by comparing the detections in different chirps of a LFM-CW receive signal. Because

this sensor specific signal processing is not part of the postprocessing architecture, no further investigations have been carried out. Overall, the clustering step has shown to achieve a correct grouping of targets according to the real extended objects. The reduced number of objects will be a benefit for the postprocessing performance, especially the data association.



Figure 3.18: Video camera snapshot of the real road traffic example

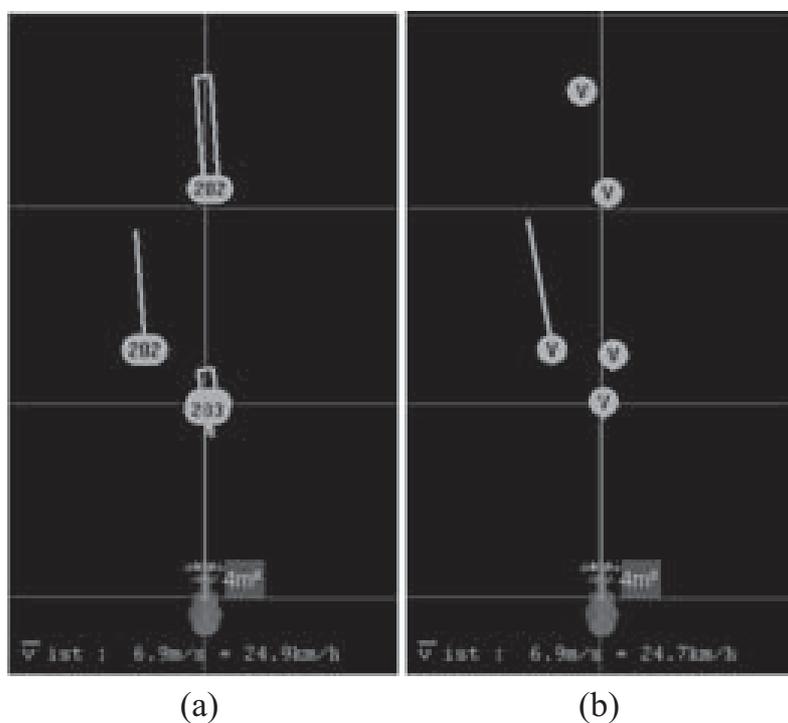


Figure 3.19: Moving target detections of both UMRR sensors (a) and the appropriate cluster at the end of the postprocessing chain (b)

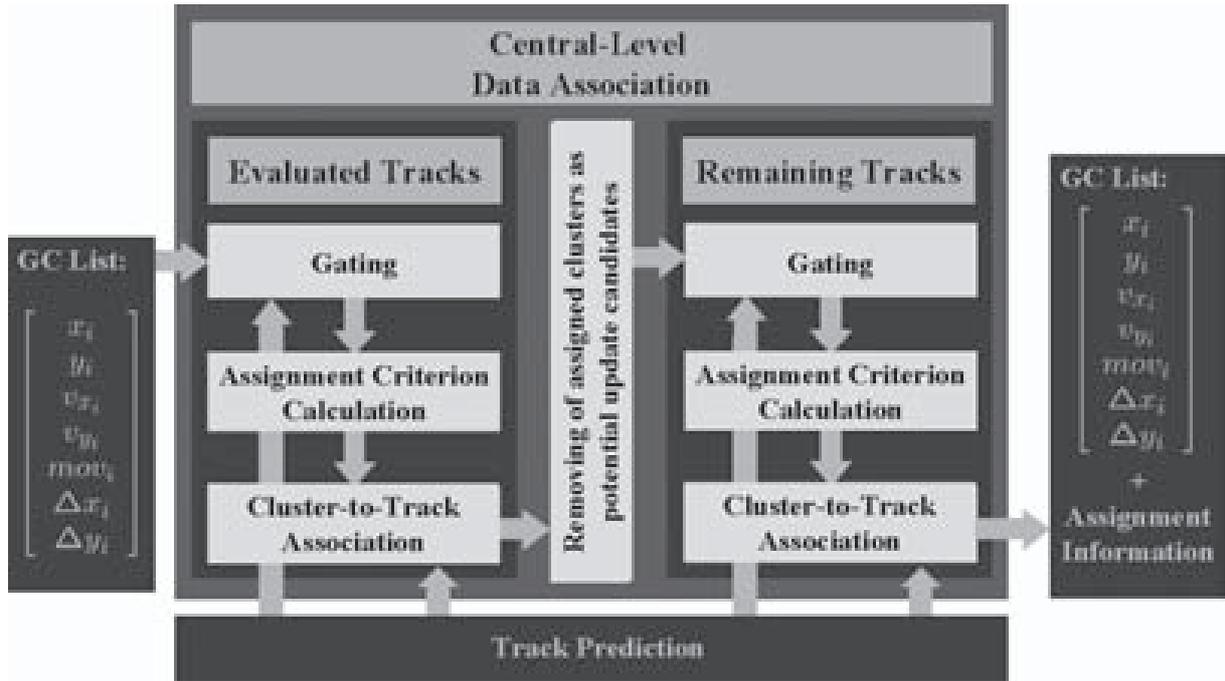


Figure 3.20: Central-level data association overview

### 3.3.2 Central-Level Data Association

The central-level data association has the same task as the appropriate sensor-level implementation, namely determining the global clusters received from the central-level clustering step that belong to already existing central-level object tracks and determining the ones that are new. Thus, the implemented central-level data association structure is the same two-level, all-neighbor data association approach described in 3.2.3 and is shown in figure 3.20. Again, it is distinguished between two data association levels. In the first level, the cluster-track assignments for all evaluated global object tracks are found in three main data association steps. Only the unassigned, remaining global object clusters are considered as possible update candidates of the unevaluated tracks in the second data association stage. This two-level distinction prioritizes the evaluated over the unevaluated object tracks and assumes that newly appearing objects will not be assignable to an existing evaluated object track. This prioritization avoids that an unevaluated and maybe not reliable track is diminishing the object cluster portion in the update of a longer existing, evaluated object track. Within each stage, the all-neighbor data association approach described in 3.2.3 is applied.

The only difference between the central- and the sensor-level data association is that the central-level data association plane has 4 dimension, because the incoming global cluster data includes the complete velocity information

$$\hat{\mathbf{c}}_i(k) = [x, y, v_x, v_y, mov]^T \quad (3.59)$$

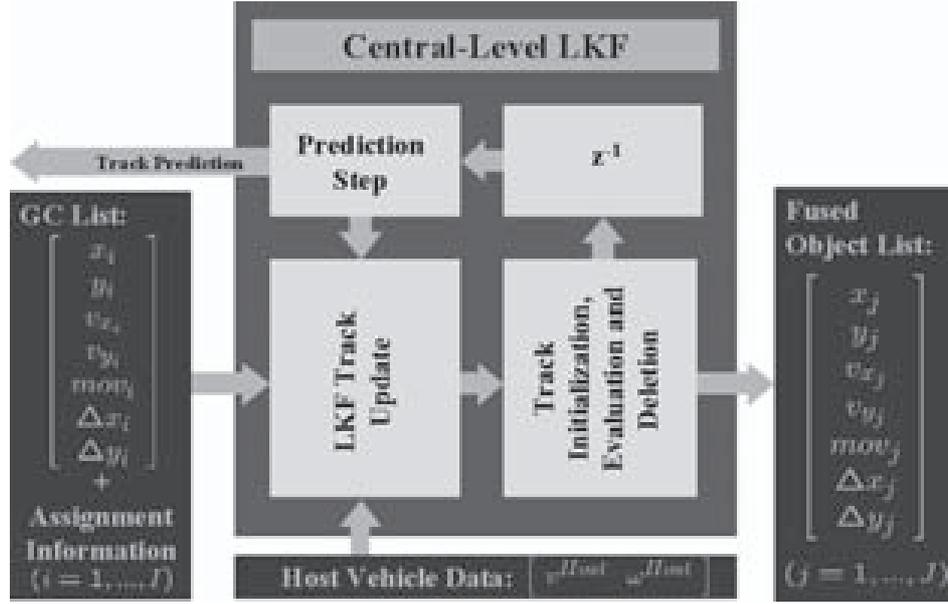


Figure 3.21: Central-level tracking overview

The first data association step is the gating, where the number of possible track update candidates among the received global clusters is reduced by limiting the search area around the global object track's predicted kinematic state. The implemented gates have an elliptical shape and are analogously defined to the sensor-level ones given in eq.(3.19)-(3.23). At the next processing stage, the determined update candidates are rated according to the assignment probability of eq.(3.25) presented in section 3.2.3. Finally, the all-neighbor cluster-to-track association decides which global object clusters are assigned to which global object tracks and transfers the global clusters including the assignment information to the central-level tracking filter stage.

### 3.3.3 Central-Level Tracking

The central-level tracking stage consists of three main processing steps (see figure 3.21), the Kalman filter update, the track initialization, evaluation and deletion stage, and the track prediction. Each cluster of the incoming global cluster list comprises

$$\hat{\mathbf{c}} = [x, y, v_x, v_y, mov, \Delta x, \Delta y]^T \quad (3.60)$$

plus the assignment information of the previous data association step and with its covariance matrix  $R$ . Each assigned global object cluster is utilized for the update of the appropriate global object track. At this, the coordinate system of the global object clusters and the central-level tracking filter's kinematic states is the same, so that no coordinate transformation is necessary prior to the tracking filter update step. The implemented Kalman filter is a LKF similar to the one that has already been outlined in section 3.2.4, so that the following Kalman filter equations remain valid:

*LKF State Estimate Prediction:*

$$P_j(k|k-1) = \Phi(k)P_j(k-1|k-1)\Phi(k)^T + Q \quad (3.61)$$

$$\hat{\mathbf{t}}_j(k|k-1) = \Phi(k)\hat{\mathbf{t}}_j(k-1|k-1) \quad (3.62)$$

*LKF State Estimate Update:*

$$K_j(k) = P_j(k|k-1) \left( P_j(k|k-1) + \sum_i Prob_{ij}(k) \cdot R_i(k) \right)^{-1} \quad (3.63)$$

$$P_j(k|k) = (I - K_j(k)) \cdot P_j(k|k-1) \quad (3.64)$$

$$\hat{\mathbf{t}}_j(k|k) = \hat{\mathbf{t}}_j(k|k-1) + K_j(k) \cdot \sum_i Prob_{ij}(k) \cdot \Delta \mathbf{c}_{ij}(k) \quad (3.65)$$

where  $\Delta \mathbf{c}_{ij}(k)$  is the innovation between the global object track prediction  $\hat{\mathbf{t}}_j(k|k-1)$  and the global cluster  $\hat{\mathbf{c}}_i(k)$

$$\Delta \mathbf{c}_{ij}(k) := \hat{\mathbf{c}}_i(k) - \hat{\mathbf{t}}_j(k|k-1) \quad (3.66)$$

The implemented LKF contains a maneuver adaptive process noise as presented in 2.2.8 and incorporates the information about the host vehicle's motion described in 2.1.5. The choice of the target dynamic model parameter, i.e. the process noise, in case of a two postprocessing level architecture with one Kalman filter in each level's processing chain is important for the radar network's estimation performance and will be discussed in section 3.4 at the end of this chapter.

### Missed Update

Missed object track updates are handled analogously to the sensor-level implementation. If no global object cluster has been assigned to an existing global object track, then the track will be maintained according to the following equations:

*LKF State Estimate Prediction:*

$$P_j(k|k-1) = \Phi(k)P_j(k-1|k-1)\Phi(k)^T + Q \quad (3.67)$$

$$\hat{\mathbf{t}}_j(k|k-1) = \Phi(k)\hat{\mathbf{t}}_j(k-1|k-1) \quad (3.68)$$

*LKF State Estimate Update:*

$$P_j(k|k) = P_j(k|k-1) \quad (3.69)$$

$$\hat{\mathbf{t}}_j(k|k) = \hat{\mathbf{t}}_j(k|k-1) \quad (3.70)$$

As already outlined in the sensor-level tracking filter section, the not updated object track can additionally be penalized by increasing its uncertainty. At this, an arbitrarily chosen covariance matrix  $Z$  is added in the LKF update step in eq.(3.70)

$$P_j(k|k) = P_j(k|k-1) + Z \quad (3.71)$$

### Track Initialization, Evaluation, and Deletion

The track initialization, evaluation, and deletion procedure is in principle the same as the straight-forward approach presented in 3.2.4. At this, the global clusters that are not assigned to an existing object track are considered as 'unused'. Each of these unassigned global clusters will initiate a new global object track with the status of not being evaluated yet. Such potential object tracks will take part in the cluster-to-track association procedure in the following measurements cycles, but with a lower priority as the evaluated global object tracks. If the unevaluated tracks are updated an arbitrary number of times  $N_{eval}$  in  $K_{check}$  consecutive measurement cycles, they will change their status to 'evaluated'. An equivalent procedure is applied for track deletion. If fewer track updates than the given number  $N_{eval}$  are possible in the last  $K_{check}$  measurement cycles, the track will lose its status as evaluated. But it will not directly be deleted, instead, it will 'survive' as an unevaluated track and still take part in the association procedure for the next  $K_{del}$  cycles. If it is not updated during this period, it will be deleted. Otherwise it can regain the status 'evaluated'.

The important difference of both track evaluation and deletion procedures can be found in the choice of the parameters  $N_{eval}$ ,  $K_{check}$ , and  $K_{del}$ . This choice has to be considered in the context of the complete postprocessing's estimation behavior, i.e. together with the appropriate sensor-level parameters as well as with the implemented target dynamic models. This important point in the postprocessing design will be discussed in the next section.

## 3.4 Choice of Tracking Filter Parameters

Implementing a two-level postprocessing architecture with one tracking filter in each level's processing chain leads to the question how to choose the tracking filter parameters, i.e. the target dynamic model and the track evaluation and deletion time. If only a single Kalman filter would be implemented, the appropriate guidelines of choosing a target dynamic model given in 2.2.3 are valid. But applying the described target dynamic model twice in the complete postprocessing chain will lead to a stronger filtering effect that results in a significant latency in reacting to an object's maneuver. A similar conclusion is valid for the track evaluation and deletion time. If the track evaluation time suitable for a single Kalman filter stage is applied to the Kalman filters of both processing levels, then the time until an object track is considered as evaluated and thus transmitted to the application processor is twice as long as in the single-level Kalman filter implementation. An analogous observation can be made for the track deletion time.

This behavior of the two-level postprocessing architecture is neither wanted nor is it reasonable.

The strategy in this two-level postprocessing architecture is to keep the tracking results in each single-sensor path as "raw" as possible. The different measurement clusters should only be stabilized by the tracking filter without introducing a too restrictive target dynamic model. At this, the Kalman filter's main task is the avoidance of measurement drop-outs in order to guarantee a continuous single-sensor object data stream at the output of the sensor-level postprocessing. Therefore, the process noise matrix  $Q$  must be chosen significantly greater than described in the guideline in section 2.2.3. This allows a quick reaction to an observed target maneuver but also reduces the Kalman filter's estimation accuracy due the lower measurement noise filtering. The measurement noise remaining in the single-sensor object data will be filtered out in the following central-level tracking filter. The sensor-level track evaluation and deletion time must also be chosen to be short compared to a single tracking filter approach. The tracking filter should quickly establish a track after an object has been detected for the first time. A 'two-out-of-three' track initiation scheme (see section 3.2.4 for definition) has proven to be reasonable. A similar time can be chosen for track deletion. At the sensor-level stage, it is important to avoid a loss of information due to too restrictive side conditions. At this, it can be accepted that false object tracks may temporarily exist in order to guarantee the maximum available information at the following central-level postprocessing stage. It is also self-evident, that these false single-sensor object tracks must be deleted as quickly as they have been initiated.

The tasks of the central-level tracking filter are complementary to the sensor-level approach. At this level, false track initializations and deletions must be avoided. This means that the time for evaluating a central-level object track must be chosen significantly higher than the one of the sensor-level tracking filter. The same conclusion remains valid for the central-level track deletion time. A 'seven-out-of-eight' track initiation scheme has proven to show good results in real road traffic scenarios and leads to an object initiation time at the output of the complete postprocessing system of  $9 \cdot 25 \text{ ms} = 225 \text{ ms}$ . The target dynamic model of the central-level Kalman filter has the task to achieve accurate kinematic state estimation results. This can be achieved by implementing one of the two target dynamic models described in section 2.2.3. Starting with the parameters of the given guideline and slightly adjusting the process noise to the targeted real road traffic scenarios proved to work well. Because the maneuver adaptive process noise presented in section 2.2.8 is implemented in the central-level Kalman filter, the standard target dynamic model can be chosen to be a bit too restrictive in order to gain precise kinematic state estimation results scenarios with constant target motion. The amount of process noise increase in case of a detected maneuver must be determined empirically for the targeted application in real road traffic scenarios.

## 3.5 Chapter Conclusions

This chapter has presented a postprocessing architecture for a multi-sensor automotive radar network, exemplarily derived for 2 UMRR sensors. The implemented algorithms have been an extension of the previously outlined single-target kinematic state estimation techniques to multi-target situations. The signal processing chain has been divided into two processing levels, the sensor- and the central-level processing. The sensor-level postprocessing contains sensor-specific algorithms that estimate the objects' kinematic states from the single-sensor measurements. It offers the advantages of a distributed system, which are a reduced data bus load and a parallel computation of the single-sensor measurements. The central-level postprocessing has mainly the task of fusing the different single-sensor object estimates according to the road user they originate from and of obtaining precise kinematic state estimates of the appropriately fused objects.

The simple target classification step has divided the detected targets of the incoming single-sensor measurements into moving and non-moving targets. This distinction has added an important information to each target, which has simplified the target clustering and the data association applied later in the postprocessing chain. The presented target clustering algorithm implemented in the sensor-level as well as in the central-level postprocessing has taken the reflection behavior of extended objects into account by assuming that different target detections belonging to the same road user may originate from different reflection centers on the road user's surface. These different reflections have been treated like clusters with a physical extension. The data association on both processing levels has taken advantage of the additional information generated by the classification step and the meaningful target grouping by the clustering steps on both processing levels. Different association criteria have been presented and discussed together with two different association philosophies, the nearest-neighbor and the all-neighbor association. Finally, a data association procedure consisting of two, differently prioritized levels has been outlined. At each priority level a JPDA all-neighbor approach has been applied. As the last postprocessing stage in both postprocessing levels, the tracking filter stage has been derived on the basis of the kinematic state estimation algorithms discussed in the previous chapter. At this, the track initialization, evaluation and deletion procedure as well as the choice of the target dynamic model parameters have been a special focus.

Having developed a two-level postprocessing architecture for a 2 UMRR network, it must be proven that this signal processing chain shows a very good performance in real road traffic scenarios. This will be addressed in the following chapter where the experimental car and its equipment as well as the estimation results in different typical road traffic situations will be presented.

# Chapter 4

## Experimental System

One of the most important achievements in engineering a complete system architecture is the prove of its functionality in real world situations. Having designed the concept and algorithms of the complete postprocessing architecture for the 2 UMRR network under important assumptions such as the reflection behavior of extended targets, the implemented system as it was described in the previous chapter must prove that it is working properly. In the following sections, the test car and the utilized equipment will be summarized briefly. The experimental results achieved with the integrated postprocessing architecture in diverse real road traffic scenarios will be presented. Since there are numerous real road traffic scenarios that have been tested with this experimental system, this chapter will restrict itself to a limited number of meaningful examples.

### 4.1 Test Vehicle and Equipment

The test vehicle used for evaluation of the previously presented postprocessing architecture is shown in figure 4.1. The two UMRR sensors are mounted symmetrically in the front bumper with a distance of 0.80 m in between them. It can be seen in figure 4.2, that the sensors are not covered by any material, e.g. the bumper's lacquered plastic. For the purpose of research activities in the field of automotive radar, this is reasonable, but will change in a later series product.

The test car's system architecture is schematically shown in figure 4.3. Both UMRR sensors are connected to a power supply and are transmitting their target lists with an update rate of 30 Hz via the sensor CAN bus of the test vehicle to the node computer. The node computer is the central processing unit where the described postprocessing system is implemented on. Beside the single-sensor target lists, it also receives the measurement data from the host vehicle sensors. The ABS sensor delivers the host vehicle speed and the turn-rate sensor sends the measured yaw-rate via the vehicle CAN bus to the postprocessing unit. The turn-rate sensor is not a part of the vehicle's periphery and was developed separately. Its picture is shown in figure 4.4. For further details on the realization and performance of this yaw-rate sensor, please consider [Föl00].



Figure 4.1: Testcar equipped with 2 UMR sensors in the front bumper



Figure 4.2: Detailed view of one UMR sensor in the front bumper

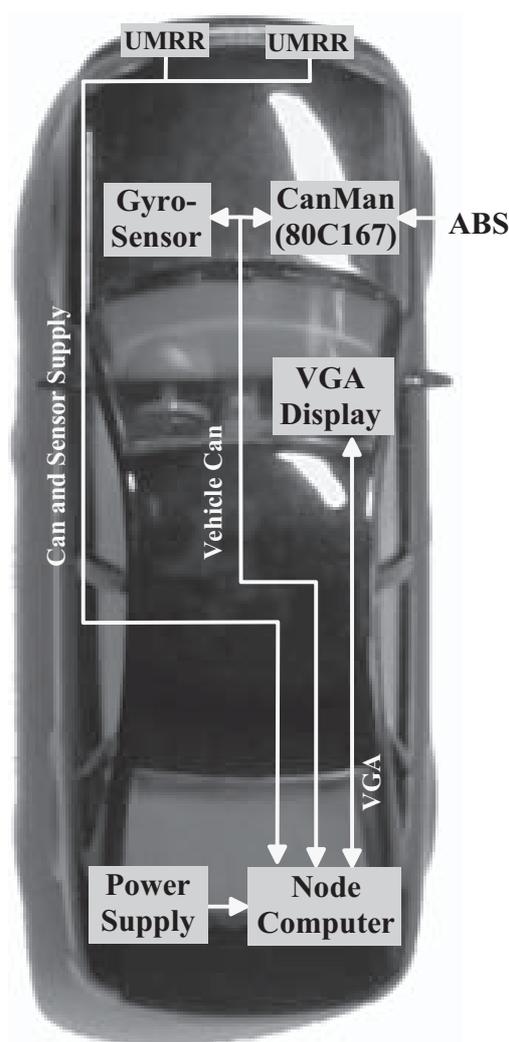


Figure 4.3: Testcar radar network structure

The update rates of the different sensor's information are summarized in table 4.1. The ABS sensor offers the information about the host vehicle speed with an update rate of 30 Hz and the turn-rate sensor periodically transmits its information with 100 Hz. As already mentioned, the two UMRR sensor are transmitting their target lists with an update rate of 30 Hz. Overall, the postprocessing architecture will send its object lists with an update rate of 30 Hz to an appropriate application processor. Thus, the complete radar network plus postprocessing system has the potential to fulfill the requirements of the comfort systems described in table 1.2 of chapter 1. According to the supplier of the UMRR sensors, the measurement update rate will be increased in the next sensor generation, so that the sensor and thus the presented postprocessing system will also be suitable for additional safety applications.

Parameter	Bus	Update Rate
Target List Sensor No.1	Sensor-CAN	30 Hz
Target List Sensor No.2	Sensor-CAN	30 Hz
Host Vehicle Velocity	Vehicle-CAN	30 Hz
Turn-Rate	Vehicle-CAN	100 Hz

Table 4.1: Information available on the CAN-bus



Figure 4.4: Turn rate sensor

## 4.2 Software Implementation Tool

The described postprocessing architecture has been implemented in C++ as modules in the signal processing development and simulation environment RASI (Radar Simulation). This tool offers the possibility to run the implemented algorithms in a synchronized or a real-time mode. Additionally, this development software is capable to integrate hardware, e.g. radar sensors, by supporting the real-time execution of the implemented modules. For details on this development tool, its messaging system and structure please consider [LR03].

RASI has a graphical user interface visualizing the signal processing flow between the different C++ modules in the actual workspace. The instantaneous communication data rates as well as the instantaneous processing loads of all modules are displayed during the simulation. The implemented structure of the previously described postprocessing system in the graphical user interface of RASI is shown in figure 4.5.

The visualization of the postprocessing system's output data is possible with the RASI Radar Viewer module. The actual detected objects are displayed in a birdseye view showing the position and the velocity of the moving objects relative to the experimental

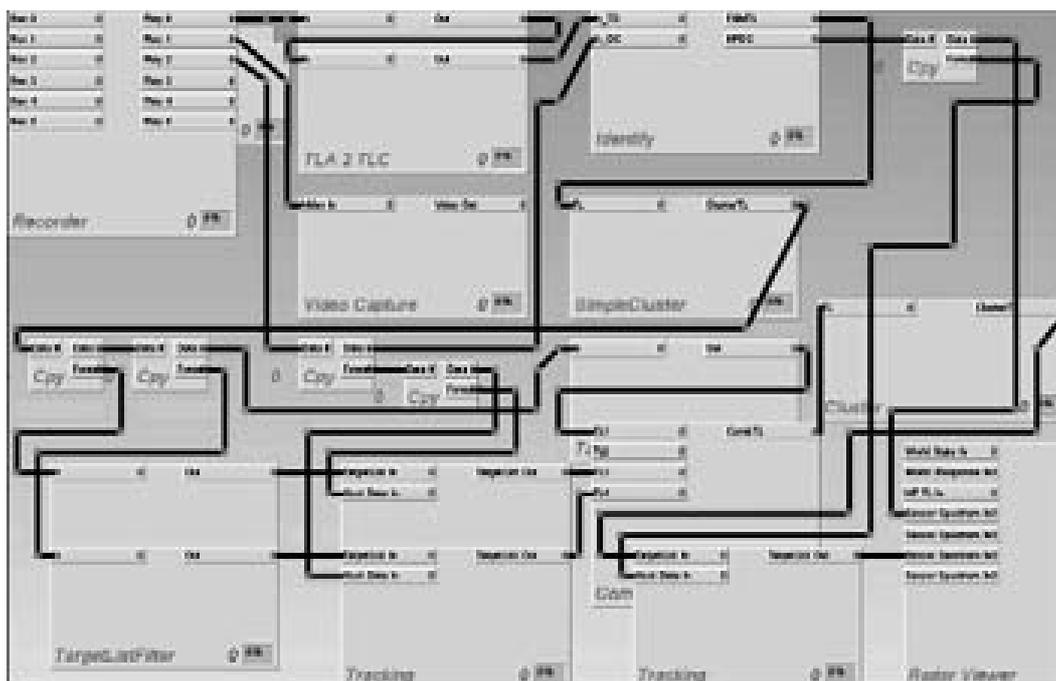


Figure 4.5: Signal processing modules in the graphical user interface of RASI

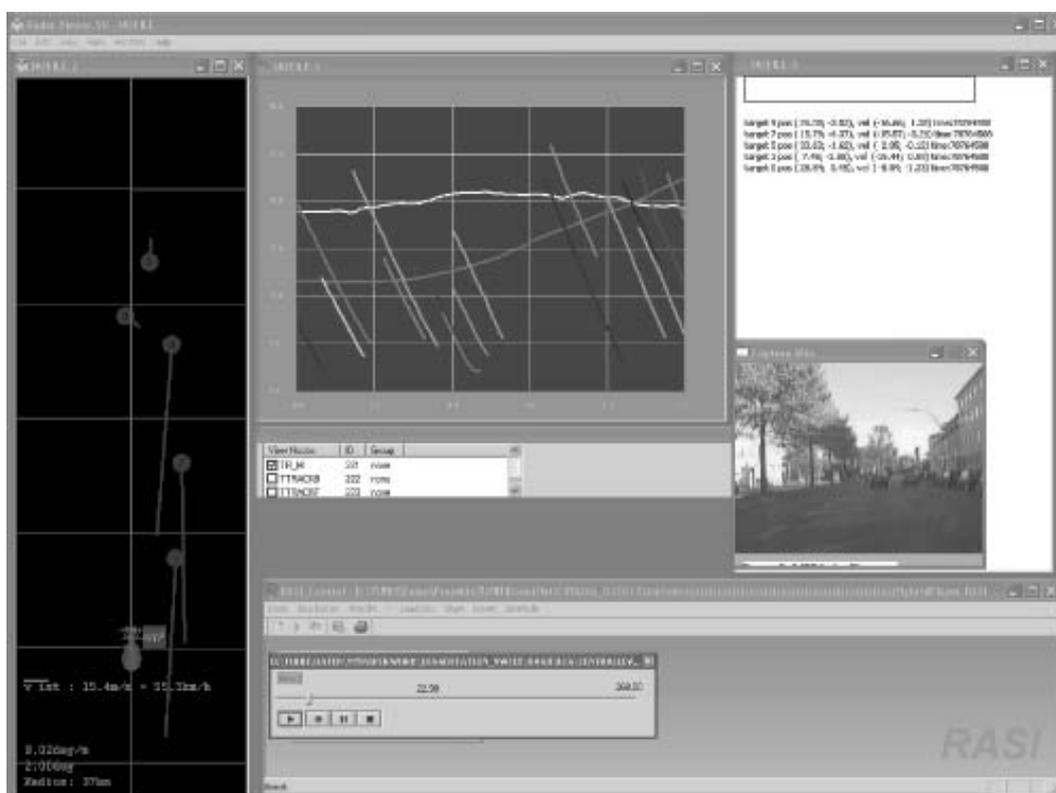


Figure 4.6: Data viewer tools in RASI

car. The available host vehicle data is also displayed in this window. To observe the object movement over time, the target traces can be viewed in a range-time diagram. Additionally, the data of the actual target lists can also be displayed as text in a separate window.

In order to evaluate the radar network data of a test drive, a video camera is integrated in the experimental car and the complete sensor data including the video information can be recorded by the RASI software. During the offline evaluation of the recorded test drives, this reference video can be viewed by using the RASI capture module.

The different postprocessing algorithms have been evaluated in real road traffic situations by utilizing this development tool. They have been redesigned several times in order to improve the system's performance. The final experimental results achieved by the implemented postprocessing architecture will be presented in the following section.

### 4.3 Real Measurement Situations

This section will present the results achieved by the implemented postprocessing architecture in real road traffic scenarios. At this, the implemented postprocessing architecture has to deal with extended targets, multi-target situations, moving and non-moving objects as well as different maneuvers of observed road users. The experimental results will be structured in three different main scenarios:

- driving on a highway,
- driving in an urban area,
- and a very special situation, driving through a tunnel.

In each of these scenarios, different situations will be considered. The situations comprise cut-in or cut-out maneuvers of observed road users or the own vehicle, cars overtaking each other in the host vehicle's scanning area, and the detection of oncoming traffic.

#### 4.3.1 Highway Scenario

The highway is a highly restricted environment concerning different side conditions. The occurring types of road users are motorcycles, passenger cars, and trucks. Their maneuvers normally comprise the acceleration, deceleration and lane changing. No oncoming, right- or left-turning traffic can be found. In a normal highway situation, all road users neighbored to the own vehicle are heading in the same direction and have at least a certain minimum speed. A special challenge for a radar based automotive application are

the wide range of occurring accelerations and velocities. Additionally, the radar system has to cope with guardrails, which lead to a great number of target detections at the radar sensors. Nevertheless, this traffic scenario is the easiest to handle for automotive radar sensors, so that ACC systems were introduced for this road traffic scenario as the first radar-based automotive convenience application. At first, ACC systems were restricted to the own car's minimum velocity of about  $m_v^{Host} = 80 \text{ km/h}$ , but today's ACC systems are working properly at  $m_v^{Host} = 20 \text{ km/h}$ . In the near future, these systems will be enhanced by short or medium range radar sensors in order to detect objects in the surroundings of the own car. Such a multi-sensor network will enable a complete stop & go functionality, e.g. in a traffic jam.

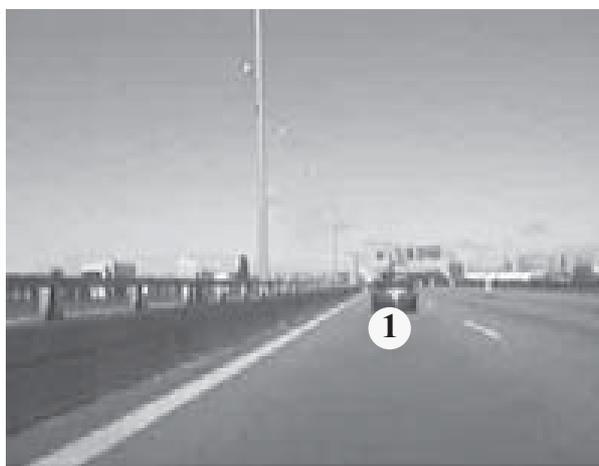


Figure 4.7: Typical ACC traffic situation

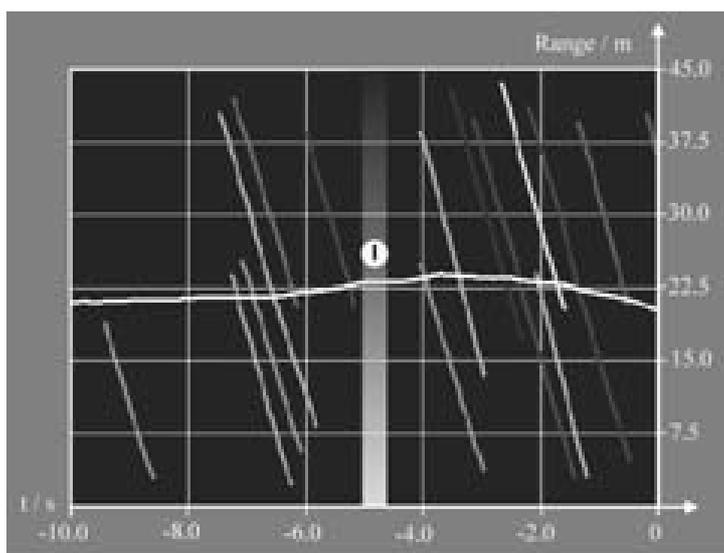


Figure 4.8: Target traces of the typical ACC traffic situation in figure 4.7

Figure 4.7 shows a typical ACC road traffic scenario. No dense traffic can be found on the highway and the own vehicle is following a vehicle driving ahead. The so-called target traces are visualized in the range-time diagram in figure 4.8. They show the measured ranges of each object track up to 45 m during the last 10 s. The point in time, where the video camera picture was captured is indicated by the transparent area including the relevant target number. This situation is a good example for the distinction between moving and non-moving objects. The target trace of the moving object (white trace) shows that the passenger car in front of the own car is moving at almost the same speed and thus is detected at almost the same range over the displayed time period of 10 s. But the great number of non-moving objects originating from the poles of the guard rail are also detected and clearly distinguished from the moving object.

A more dense traffic highway situation is shown in figure 4.9. The own vehicle is following a passenger car (target no.1) on the middle lane. Different road users such as the small bus (target no.2) are passing by on the left lane. The trucks on the right lane are almost standing still. The traces of the detected targets are shown in figure 4.10. It can clearly be seen that target no.1 (pink trace), which is the passenger car ahead of the own vehicle, is driving at almost the same speed at a distance around 35 m. Target no.2 is moving at a higher speed ( indicated by the positive gradient of the yellow trace) and is overtaking the experimental vehicle on the left lane. The remaining ones among the shown target traces are detections of the almost non-moving trucks on the right lane. The target traces have a negative gradient until they reach  $r_{trace} = 0$  m. At this point in time, the object tracks are not updated anymore because they left the scanning area of the radar network. But the tracking filter keeps these tracks alive for a very short time in order to find another update. Thus, the target traces of the trucks on the right lane are still existing after the trucks have been overtaken. By definition, "range" is a positive value, so that target traces change to a positive gradient indicating when the object is dropping away.

Overall, the tests in highway situations have shown that the radar network could detect all relevant objects inside its scanning area. It was able to handle a high number of non-moving objects originating from the guardrail or road work barriers, for example. At this, it could clearly distinguish these non-moving objects from the fewer moving ones, also in medium dense traffic situations.

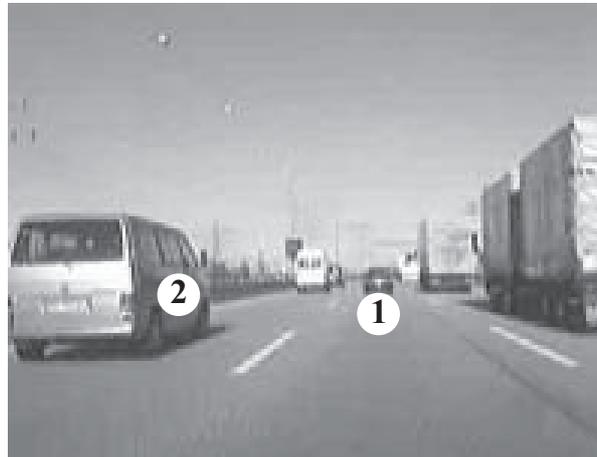


Figure 4.9: Highway situation with medium dense traffic

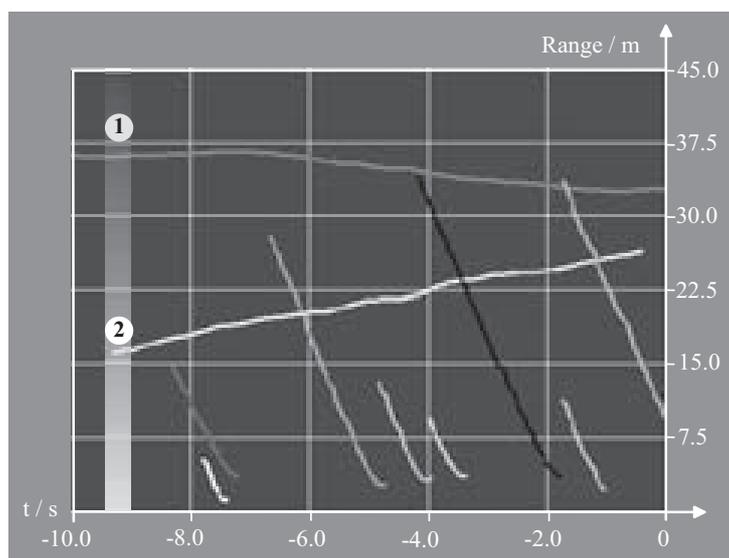


Figure 4.10: Target traces of the highway situation with medium dense traffic



Figure 4.11: Urban traffic scenario with one passenger car overtaking another one

### 4.3.2 Urban Scenario

The urban traffic scenario adds many requirements to the previously presented, typical ACC scenario on the highway. In urban environments, the radar sensors have to detect a greater variety of road users, e.g. pedestrians, bicycles, passenger cars, or trucks. All of them can head in every possible direction (e.g. at the crossways). Additionally, diverse non-moving objects such as parking cars can be found on the side of urban streets.

Figure 4.11 shows an urban traffic situation where one passenger car (target no.1) is overtaking another one (target no.2) in front of the own vehicle. This situation is critical for radar sensors that can only achieve a target resolution in the radial direction. If both cars are situated at almost the same distance, the sensors will only detect one target with a wrong kinematic state that results from the fusion of both targets' real kinematic states. This radar typical measurement limitation must be solved by the implemented postprocessing architecture without generating wrong object track assignments or track discontinuities. The target traces in figure 4.12 show that the object tracks of target no.1 (pink trace) and 2 (white trace) are continuous although the two targets cannot be resolved in the area where both are located at almost the same range. The remaining target traces belong to parking cars on the right side as well as some metal poles on the left side of the street. The distinction between the moving and non-moving objects in the postprocessing system is already achieved at the sensor-level and can be visualized by the range-time diagram in figure 4.13. It only shows the target traces of the two moving objects of both sensor-level postprocessing systems and proves that a clear distinction between moving and non-moving objects is possible by the implemented algorithms.

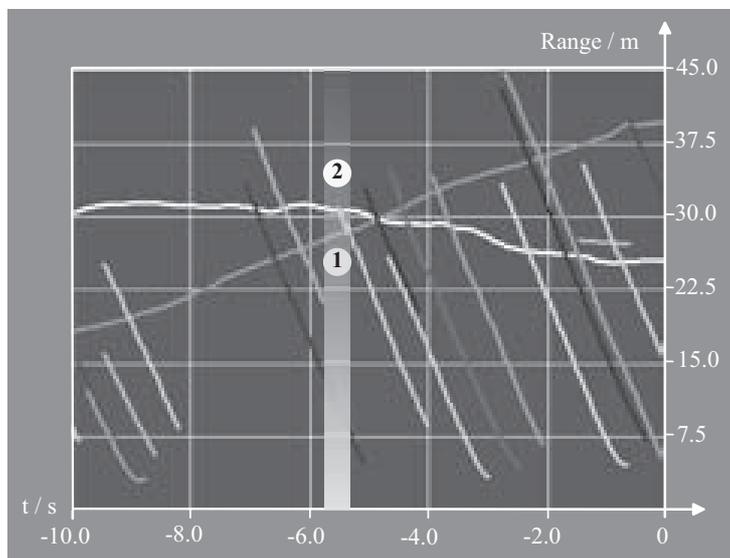


Figure 4.12: Central-level target traces of the urban traffic scenario in figure 4.11

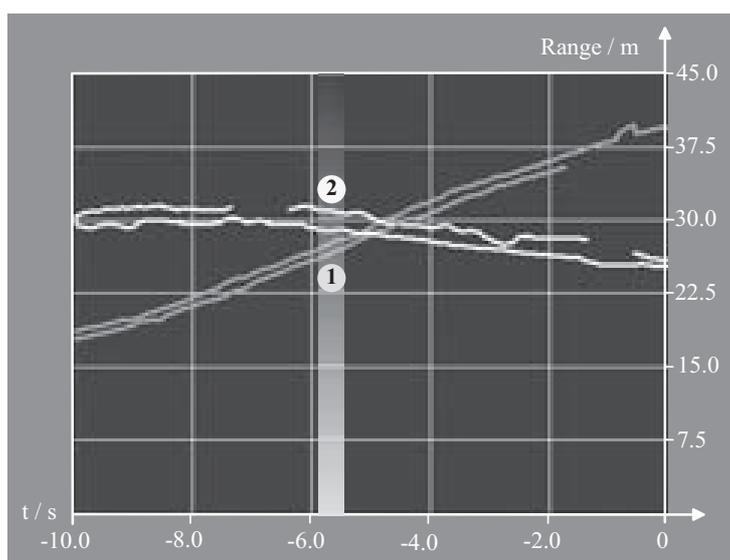


Figure 4.13: Sensor-level target traces of sensor no.1 and 2 of the detected moving objects



Figure 4.14: Cut-out situation in an urban traffic environment

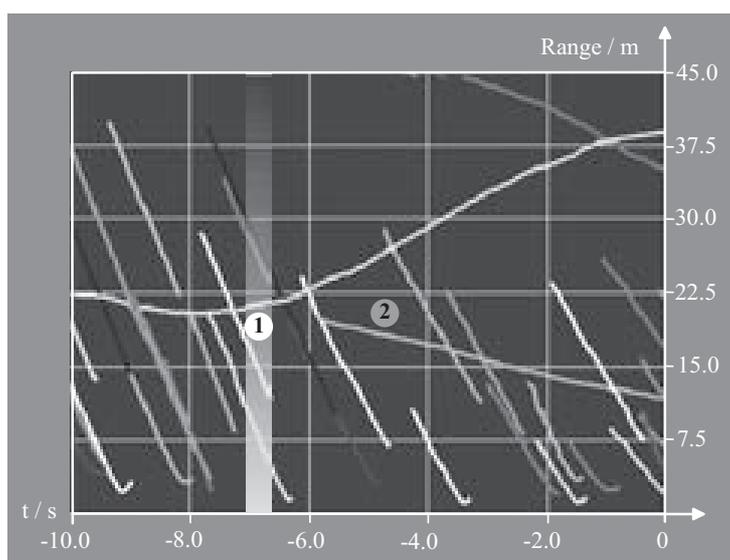


Figure 4.15: Target traces of the cut-out situation in an urban traffic environment

An important situation type in urban traffic scenarios are the cut-in and cut-out maneuvers of vehicles driving in front of the own car. It is important that a vehicle cutting in the lane of the own car or a vehicle ahead of a car cutting out of the own lane is detected early enough to give the driver or the application sufficient time for an appropriate reaction. Figure 4.14 shows a cut-out situation at a range of 20 m ahead the experimental car. Target no.1 is changing lanes in order to overtake target no.2 on the right lane. The appropriate target traces in figure 4.15 show that after about half a second after initiation of target no.1's cut-out maneuver, target no.2 is detected, so that an appropriate reaction of the application would be possible.



Figure 4.16: A second cut-out situation in an urban traffic environment

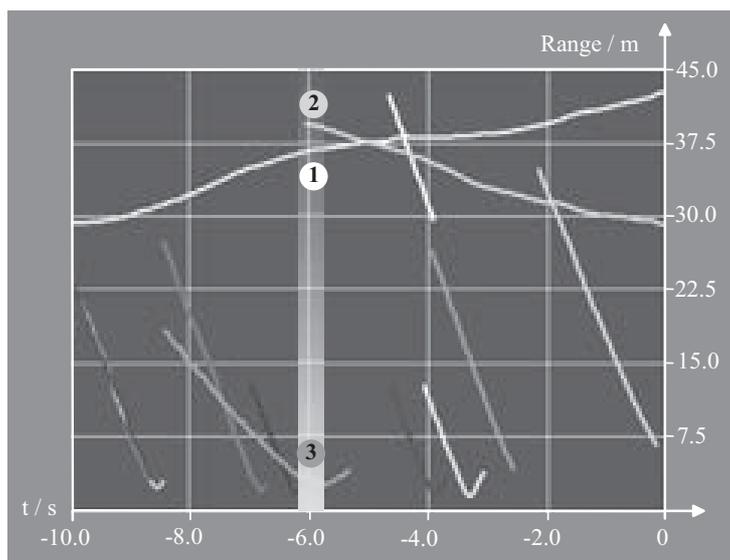


Figure 4.17: Target traces of the second cut-out situation in an urban traffic environment

In some cut-out situations the experimental system has been able to detect the second vehicle driving ahead of the maneuvering car even before the maneuver has been initiated. Such a road traffic example shows figure 4.16 with the appropriate target traces in figure 4.17. Again, target no.1 is changing lanes in order to overtake target no.2 on the right lane. But in this case, target no.2 has already been detected a short time before target no.1 initiates its maneuver. This example shows also that a clear distinction between the slow moving passenger car (target no.3) on the right lane and the target traces belonging to stationary objects in the surroundings of the road is possible.



Figure 4.18: Cut-in situation in an urban traffic environment

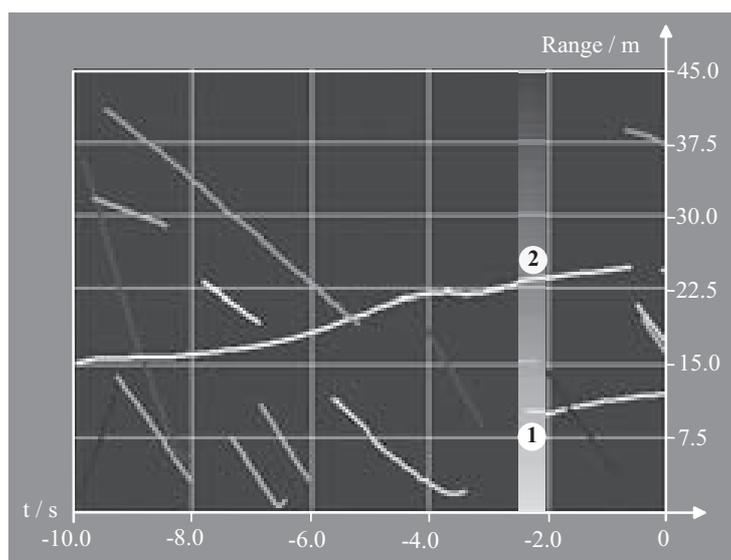


Figure 4.19: Target traces of the cut-in situation in an urban traffic environment

Cut-in situations are more critical than cut-outs, because a new vehicle is entering the own driving path and thus significantly shortens the distance to the next object driving ahead of the own vehicle. Figure 4.18 shows such a cut-in situation. The black passenger car (target no.1) is entering the driving path of the experimental car and thus is immediately closer as target no.2. The target traces in figure 4.19 show that the distance between target no.2 and the own vehicle is increasing until the gap is large enough for target no.1 to cut in the left lane. Because target no.1 is entering in the radar sensor's scanning area from the side, it suddenly appears at a short distance (see light blue trace). Targets entering the radar sensor's scanning area from the side must be evaluated after the first



Figure 4.20: Tunnel scenario with dense traffic moving at almost the same velocity

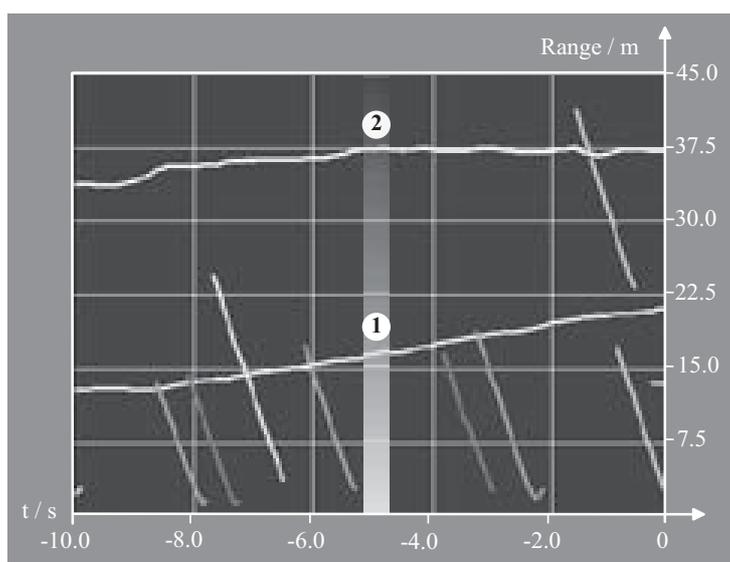


Figure 4.21: Target traces of the tunnel scenario with dense traffic

detection in order to avoid the generation of ghost targets. This evaluation time must be short enough to offer the driver or the appropriate application enough time to react to this immediately appearing object at a short distance while guaranteeing the avoidance of ghost targets. The implemented postprocessing system has proven to have a sufficiently quick reaction in such situations. The other target traces in figure 4.19 can also be assigned to obstacles present in this road traffic scenario. The red target trace with a great negative gradient belongs to a vehicle from the oncoming traffic of the other side of the street. The green and the yellow target trace with a relative low gradient are originating from road users that have been overtaken by the experimental car.



Figure 4.22: Tunnel scenario with dense traffic moving at different velocities

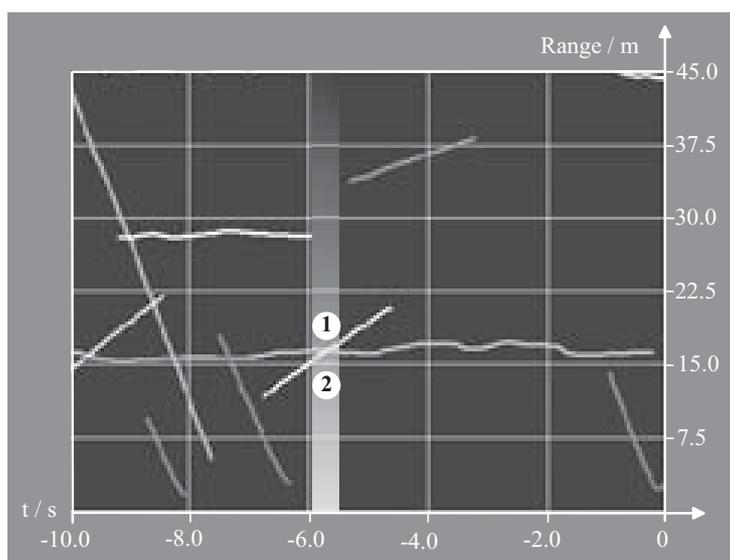


Figure 4.23: Target traces of the tunnel scenario with dense traffic moving at different velocities

The experimental radar network has shown that it is capable of handling typical urban traffic situations such as cut-in and cut-out situations as well as overtaking cars, where the postprocessing overcomes the radar sensors' inability of lateral resolution. In these situations, the system was again able to distinguish between the great number of stationary targets and moving vehicles.

### 4.3.3 Tunnel Scenario

The tunnel scenario is a very special situation. The radar sensors will have to deal with detections originating from the walls of the tunnel while they must be capable to detect the moving objects, cars or trucks, without greater disturbance, without missing detections, or without generating ghost targets. Figure 4.20 shows the experimental vehicle driving on the left lane in a tunnel (of the river Elbe). It follows the passenger car (target no.1). All vehicles are moving at almost the same velocity. The target traces in figure 4.21 show that two moving objects are detected by the radar network. Target no.1 is the dark passenger in front of the own vehicle and target no.2 is the truck on the right lane. The passenger car driving on the right lane at almost the same distance as target no.1 is not detected. Probably, this car has not been detected after entering the radar network's scanning area, because it has immediately been situated at the same distance as target no.1. Due to the radar sensor's inability of resolving two objects in the lateral direction, it will not be detected until the velocity or range difference between the two vehicle is large enough.

Figure 4.22 shows again a tunnel scenario with dense traffic, but in this example, the vehicles on both lanes are moving at significant different velocities. The experimental car is following a big truck (target no.1) driving on the right lane. Different vehicles are passing by on the left lane. The target traces in figure 4.23 show a constant detection of the white truck (red trace of target no.1) and also different vehicles overtaking the own car on the left lane. The passenger car, target no.2 (pink trace), is detected as long as it is not "masked" by the big target no.1 due to the sensor's inability of lateral resolution. The remaining target traces have the same negative gradient and are belonging to detections originating from the tunnel walls.

The implemented radar network system has proven to be able to distinguish between moving and non-moving objects inside a tunnel without greater disturbance or without generating a significant number of ghost targets. All relevant objects have been detected in a dense traffic situation with either all vehicles moving at the same or at different velocities.

## 4.4 Chapter Conclusions

This chapter has presented the most important part of this thesis, the prove that the analyzed algorithms, the designed and implemented postprocessing architecture are working in real road traffic scenarios. A great number of tests have been carried out in different road traffic situations and only an excerpt of the experimental results could be presented in this chapter. The integrated medium range radar network has proven to work in highway, urban traffic, and challenging tunnel situations. It has shown to work in dense traffic as well as in situations with a great number of stationary targets. These stationary targets could clearly be identified by the implemented algorithms and thus distinguished from the moving objects. Multiple reflection from the same road users were clustered,

so that only one object has been generated for each road user by the implemented post-processing system. As already mentioned in the last chapter, the estimation of object extensions depends strongly on the fact whether multiple reflections are received from a single road user or not. The test drives have proven that such an extension estimation is sometimes possible, but reveals to be unstable. A solution to this fluctuating extension estimation would be the estimation directly in the sensor front-end. Overall, the implemented medium range radar network with its postprocessing architecture has shown good results in different road traffic situations being relevant for new forward sensing applications. Future improvements in the sensor performance will lead to an even better radar network system performance suitable for a greater variety of road traffic situations and automotive applications.

# Chapter 5

## Conclusion and Future Outlook

This thesis presented a postprocessing architecture for an automotive radar network, which is capable of fulfilling the requirements of new safety- and convenience applications, such as the collision mitigation or the stop & go system extending the existing ACC. Chapter 1 introduced a selection of radar-based automotive applications and outlined their requirements. It explained also the radar fundamentals that are the basis for understanding the radar network implementation in this thesis. It stressed that the complete radar network must properly work in a dense road traffic environment, i.e. in multi-target situations. Thus, the presented postprocessing system must be capable of handling a great number of objects. It must take into account that the detected objects behave like extended targets. On the basis of these real radar measurement properties, two radar network concepts with two different high performance 24 GHz radar sensors have been introduced and evaluated. As a result of this radar network concept comparison, it was shown, that a 2 monopulse radar network, which is measuring one range, radial velocity, and azimuth angle per target, is superior to a high precision range-only measuring 4 sensor pulse radar network.

Chapter 2 focused on kinematic state estimation algorithms for different sets of radar sensor measurements. First, it presented a least squares kinematic state estimation approach based on the radar measurements of a single measurement cycle and outlined its properties. After that, a general discrete Kalman filter considering all measurements up to the actual measurement cycle was introduced. Such an approach offered the possibility of incorporating a target dynamic model, which led to an increase of kinematic state estimation accuracy compared to the least squares approach. In the following, different Kalman tracking filter algorithms were derived and their properties were discussed. Finally, the least squares kinematic state estimation algorithm and different Kalman tracking filter were combined to single-target postprocessing systems and were evaluated in a comparison. At this, it was shown that the systems with a linear Kalman filter as well as the unscented Kalman filter were superior to the extended Kalman filter in the considered situations. Furthermore, this chapter presented and evaluated different Kalman filter enhancements. The maneuver adaptive Kalman filtering addressed the need for a quicker filter state response to a strongly maneuvering target. It detected the

maneuver via a statistical distance measure and increased the allowed object dynamics in the target dynamic model during the maneuver. In another approach, a reduction in computation complexity while maintaining the kinematic state estimation performance was achieved by a decoupling of the different filter states. Finally, the incorporation of the measured host vehicle motion in the filter equations was outlined with the goal to take advantage of all available information that describes the relative kinematic state of the detected targets.

Chapter 3 extended the kinematic state estimation and tracking techniques of chapter 2 to multi-target situations and derived a postprocessing architecture for a multi-sensor radar network, which is capable of handling extended objects. At this, it focused on a two-level postprocessing implementation for a 2 UMRR sensor network. Two important processing steps have been presented in order to distinguish between the different observed extended objects, which proved to generate a great number of detected targets at the radar sensors. The first step was the simple target classification distinguishing between moving and non-moving targets by incorporating the information about the own vehicle's motion. The second one is the clustering step that groups all targets originating from the same object to one target cluster. Both processing steps helped to achieve a distinction between the different extended objects while also leading to a meaningful data reduction. Furthermore, this chapter presented data association algorithms necessary for handling multi-target situations. Finally, the structure of the tracking filters including the track initialization, evaluation, and deletion procedure implemented in the postprocessing system was outlined with the special focus on the choice of two-level system's parameters.

Surely, the highlight of this thesis was the integration of the postprocessing architecture in the experimental car in order to evaluate the considered algorithms in real road traffic situations. The development of the presented postprocessing system was only possible on the basis of practical experiences in real road traffic situations. Analytical investigations and simulations were the fundamentals of the first rudimentary postprocessing approach but could not model the complex real road traffic environment. The postprocessing algorithm redesign initiated by the practical experiences led to the presented high performance of the developed postprocessing architecture. A selection of these experimental results achieved by the implemented system was presented in chapter 4.

The extensive use of the experimental car has not only led to improvements of the implemented postprocessing system, it has also given some hints on future investigations of signal processing parts being not part of this thesis. Starting with the sensor front-ends, the first part in the signal processing chain: It would be one of the most important improvements if radar sensors were able to resolve different targets in the lateral direction. The first scanning far distance radar sensors are commercially available at the moment and will be used in ACC/Stop & Go and collision mitigation systems in the future. The resolution in the azimuth angular direction is either achieved by a mechanically scanning or an electronically steered phased array antenna. It is self-evident, that this complex technology must be available at a certain price and that the sensors must at least have the same size as the already available non-scanning front-ends in order to be integrated

in future series products. Concerning the postprocessing steps, the clustering has proven to work efficiently but the simultaneous estimation of object extensions from the available target list data at the output of the sensor front-ends has not been reliable. The phenomenon of object extensions can already be observed in the spectrum of the down-converted LFM-CW receive signal, where extended objects can be identified due to the superposition of several magnitude peaks. The object size estimation in the spectrum would lead to more stable results than the one from the available target list information. The clustering of targets can also be improved by incorporating other sensor types, such as a video camera or a lidar sensor. Many investigations on applications like heading control or lane departure warning systems utilizing a video camera have been carried out in recent years. If these systems will be implemented in a passenger car, the additional information can be utilized for object classification including object size estimation. Object classification with different type of sensors is already a big topic for research institutes and the automotive industry since new pedestrian safety systems are demanded from the European Union. Such a multi-sensor classification will simplify several steps in the described postprocessing architecture and has a high potential to improve the overall performance of driver assistance systems.



# Appendix A

## Data Fusion

The task of data fusion can be found at several postprocessing stages in this work, e.g. in the sensor-level and the central-level clustering as well as the track-to-track fusion. At this, two data sets have to be fused to a single data set. In the following the **minimum variance data fusion** will be derived in detail:

Consider two estimates  $\hat{\mathbf{x}}_1$  and  $\hat{\mathbf{x}}_2$  of the true state vector  $\bar{\mathbf{x}}$  given by

$$\hat{\mathbf{x}}_1 = \bar{\mathbf{x}} + \mathbf{n}_1 \quad (\text{A.1})$$

$$\hat{\mathbf{x}}_2 = \bar{\mathbf{x}} + \mathbf{n}_2 \quad (\text{A.2})$$

$\mathbf{n}_1$  and  $\mathbf{n}_2$  are unbiased Gaussian noise vectors with the covariance matrices

$$\text{Cov}(\hat{\mathbf{x}}_1) = E[\mathbf{n}_1 \mathbf{n}_1^T] = P_1 \quad (\text{A.3})$$

$$\text{Cov}(\hat{\mathbf{x}}_2) = E[\mathbf{n}_2 \mathbf{n}_2^T] = P_2 \quad (\text{A.4})$$

so that the expectations of the two estimates  $\hat{\mathbf{t}}_1, \hat{\mathbf{t}}_2$  are

$$E[\hat{\mathbf{x}}_1] = E[\hat{\mathbf{x}}_2] = \bar{\mathbf{t}} \quad (\text{A.5})$$

Now, these two estimated vectors should be fused according to the following equation

$$\hat{\mathbf{x}}_f = A \cdot \hat{\mathbf{x}}_1 + B \cdot \hat{\mathbf{x}}_2 \quad (\text{A.6})$$

where  $A$  and  $B$  are weights with the side condition

$$A + B = I \quad (\text{A.7})$$

At this,  $I$  is the identity matrix. Since this fusion is a linear operation, the fused estimate  $\hat{\mathbf{x}}_{\text{fused}}$  is also unbiased

$$E[\hat{\mathbf{x}}_f] = \bar{\mathbf{x}} \quad (\text{A.8})$$

The covariance matrix of the fused estimate  $\hat{\mathbf{x}}_{\text{fused}}$  can be calculated as

$$\begin{aligned}
\text{Cov}(\hat{\mathbf{x}}_f) &= E[(A\hat{\mathbf{x}}_1 + B\hat{\mathbf{x}}_2 - \bar{\mathbf{x}})(A\hat{\mathbf{x}}_1 + B\hat{\mathbf{x}}_2 - \bar{\mathbf{x}})^T] \\
&= E[(A\mathbf{n}_1 + B\mathbf{n}_2)(A\mathbf{n}_1 + B\mathbf{n}_2)^T] \\
&= E[\mathbf{A}\mathbf{n}_1\mathbf{n}_1^T A^T + B\mathbf{n}_2\mathbf{n}_2^T B^T + \mathbf{A}\mathbf{n}_1\mathbf{n}_2^T B^T + B\mathbf{n}_2\mathbf{n}_1^T A^T] \\
&= AP_1A^T + BP_2B^T + AP_{12}B^T + BP_{21}A^T
\end{aligned} \tag{A.9}$$

where  $P_{12}$  and  $P_{21}$  are the two cross-covariance matrices with the property  $P_{12} = P_{21}^T$ .

Until now, this derivation was a general data fusion approach with two weight matrices  $A$  and  $B$ . In the following these matrices will be determined, so that a *minimum variance fusion* is achieved. Therefore, the error of the fused state estimate  $\hat{\mathbf{x}}_f$  must be considered. It is given by

$$\mathbf{n}_f = A \cdot \hat{\mathbf{x}}_1 + B \cdot \hat{\mathbf{x}}_2 - \bar{\mathbf{x}} \tag{A.10}$$

The minimum variance approach tries to minimize the sum of variances of each element in this error vector  $\mathbf{n}_f$

$$E[\mathbf{n}_f^T \mathbf{n}_f] \stackrel{!}{=} \min \tag{A.11}$$

This sum of variances can be written as

$$\begin{aligned}
E[\mathbf{n}_f^T \mathbf{n}_f] &= E[(A\hat{\mathbf{x}}_1 + B\hat{\mathbf{x}}_2 - \bar{\mathbf{x}})^T(A\hat{\mathbf{x}}_1 + B\hat{\mathbf{x}}_2 - \bar{\mathbf{x}})] \\
&= E[(A\bar{\mathbf{x}}_1 + A\mathbf{n}_1 + B\bar{\mathbf{x}}_2 + B\mathbf{n}_2 - \bar{\mathbf{x}})^T(A\bar{\mathbf{x}}_1 + A\mathbf{n}_1 + B\bar{\mathbf{x}}_2 + B\mathbf{n}_2 - \bar{\mathbf{x}})] \\
&= E[(A\mathbf{n}_1 + B\mathbf{n}_2)^T(A\mathbf{n}_1 + B\mathbf{n}_2)] \\
&= E[(\mathbf{n}_1^T A^T A \mathbf{n}_1 + \mathbf{n}_2^T B^T B \mathbf{n}_2 + \mathbf{n}_1^T A^T B \mathbf{n}_2 + \mathbf{n}_2^T B^T A \mathbf{n}_1)]
\end{aligned}$$

with  $B = I - A$ , this leads to

$$\begin{aligned}
E[\mathbf{n}_f^T \mathbf{n}_f] &= E[(\mathbf{n}_1^T A^T A \mathbf{n}_1 + \mathbf{n}_2^T (I - A)^T (I - A) \mathbf{n}_2 + \mathbf{n}_1^T A^T (I - A) \mathbf{n}_2 + \mathbf{n}_2^T (I - A)^T A \mathbf{n}_1)] \\
&= E[\mathbf{n}_1^T A^T A \mathbf{n}_1] + E[\mathbf{n}_2^T (I - A)^T (I - A) \mathbf{n}_2] \\
&\quad + E[\mathbf{n}_1^T A^T (I - A) \mathbf{n}_2] + E[\mathbf{n}_2^T (I - A)^T A \mathbf{n}_1] \\
&= E[\mathbf{n}_1^T A^T A \mathbf{n}_1] + E[\mathbf{n}_2^T (I - A^T - A + A^T A) \mathbf{n}_2] \\
&\quad + E[\mathbf{n}_1^T (A^T - A^T A) \mathbf{n}_2] + E[\mathbf{n}_2^T (A - A^T A) \mathbf{n}_1]
\end{aligned}$$

In order to find the matrix  $A$  that minimizes this expression, the equation is deviated after each element of  $A$  and the deviations are set to zero

$$\begin{aligned}
\partial E[\mathbf{n}_f^T \mathbf{n}_f] / \partial A &= 2A \cdot E[\mathbf{n}_1 \mathbf{n}_1^T] + (2A - 2) \cdot E[\mathbf{n}_2 \mathbf{n}_2^T] + \\
&\quad - (2A - 1) \cdot E[\mathbf{n}_1 \mathbf{n}_2^T] - (2A - 2) \cdot E[\mathbf{n}_2 \mathbf{n}_1^T]
\end{aligned}$$

$$\begin{aligned}
&= 2A \cdot P_1 + (2A - 2) \cdot P_2 - (2A - 1) \cdot P_{12} - (2A - 1)P_{21} \\
&= 2A \cdot (P_1 + P_2 - P_{12} - P_{21}) - 2 \cdot P_2 + P_{12} + P_{21} \stackrel{!}{=} 0
\end{aligned}$$

This leads to the weight matrices

$$A = \frac{P_2 - (P_{12} + P_{21})/2}{P_1 + P_2 - P_{12} - P_{21}} \quad (\text{A.12})$$

$$B = \frac{P_1 - (P_{12} + P_{21})/2}{P_1 + P_2 - P_{12} - P_{21}} \quad (\text{A.13})$$

so that the minimum variance fusion formula is

$$\hat{\mathbf{x}}_f = \left( \frac{P_2 - (P_{12} + P_{21})/2}{P_1 + P_2 - P_{12} - P_{21}} \right) \cdot \hat{\mathbf{x}}_1 + \left( \frac{P_1 - (P_{12} + P_{21})/2}{P_1 + P_2 - P_{12} - P_{21}} \right) \cdot \hat{\mathbf{x}}_2 \quad (\text{A.14})$$

with the covariance matrix of the fused estimate given by

$$\begin{aligned}
Cov(\hat{\mathbf{x}}_f) &= AP_1A^T + BP_2B^T + AP_{12}B^T + BP_{21}A^T \\
&= P_1 - (P_1 - P_{12}) \cdot (P_1 + P_2 - P_{12} - P_{12}^T)^{-1} \cdot (P_1 - P_{12})^T
\end{aligned} \quad (\text{A.15})$$

In the following, the minimum variance data fusion will be summarized for the two cases of uncorrelated and correlated data sets.

## Fusion of Uncorrelated Data

The assumption of uncorrelated estimation errors of the fusion candidate vectors leads to the side condition, that

$$P_{12} = P_{21}^T = 0 \quad (\text{A.16})$$

Thus, the processing steps of the *minimum variance data fusion of uncorrelated estimates* are given by

*Fusion Condition:*

$$(\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2)^T \cdot S^{-1} \cdot (\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2) \leq \delta \quad (\text{A.17})$$

with

$$S = P_1 + P_2 \quad (\text{A.18})$$

*Fusion:*

$$\hat{\mathbf{x}}_f = P_2 \cdot S^{-1} \cdot \hat{\mathbf{x}}_1 + P_1 \cdot S^{-1} \cdot \hat{\mathbf{x}}_2 \quad (\text{A.19})$$

$$Cov(\hat{\mathbf{x}}_f) = P_1 \cdot S^{-1} \cdot P_2 \quad (\text{A.20})$$

## Fusion of Correlated Data

If the estimation errors of the fusion candidates are correlated, then the cross covariance matrix is nonzero

$$P_{12} = P_{21}^T \neq 0 \quad (\text{A.21})$$

The following processing steps have to be calculated in the case of the *minimum variance data fusion of correlated estimates*:

*Fusion Condition:*

$$(\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2)^T \cdot S^{-1} \cdot (\hat{\mathbf{x}}_1 - \hat{\mathbf{x}}_2) \leq \delta \quad (\text{A.22})$$

with

$$S = P_1 + P_2 - P_{12} - P_{12}^T \quad (\text{A.23})$$

*Fusion:*

$$\hat{\mathbf{x}}_f = \hat{\mathbf{x}}_1 + [P_1 - P_{12}] \cdot S^{-1} \cdot [\hat{\mathbf{x}}_2 - \hat{\mathbf{x}}_1] \quad (\text{A.24})$$

$$\text{Cov}(\hat{\mathbf{x}}_f) = P_1 - [P_1 - P_{12}] \cdot S^{-1} \cdot [P_1 - P_{12}]^T \quad (\text{A.25})$$

At this, the cross covariance matrix between the two data fusion candidates must be known. Therefore, the knowledge about the dependencies between the estimation errors must be applied. In the central-level clustering algorithm as well as in the track merging, the fusion candidates are the tracked kinematic state estimates. Under the assumption that the two fusion candidates represent the same object with its inherent dynamics, both share the common process noise in the tracking filter model. Thus, the estimation errors are correlated. As suggested in [BSF88], the cross-covariance matrix  $P_{12}$  can be obtained by a recursion. At time index  $k$ , it is given by

$$\begin{aligned} P_{12}(k|k) = & [I - K_1(k)H_1(k)] \\ & \cdot [\Phi(k-1)P_{12}(k-1|k-1)\Phi(k-1)^T + Q] \\ & \cdot [I - K_1(k)H_1(k)]^T \end{aligned} \quad (\text{A.26})$$

with the initial condition

$$P_{12}(0|0) = 0 \quad (\text{A.27})$$

To calculate the cross-covariance matrix according to eq.(A.26), the latest Kalman gains  $K_i(k)$  as well as the latest measurement matrices  $H_i(k)$  of the last track update in each sensor path ( $i = 1, 2$ ) must be available. Since the cross-covariance must already be known for testing the hypotheses that both data sets represent the same target (see eq.(A.22)), it must be calculated for each possible fusion pair in each cycle.

# Acronyms and Abbreviations

$\hat{\mathbf{c}}$	Vector containing estimated cluster data
$\hat{\mathbf{m}}$	Vector containing estimated measurement data
$\hat{\mathbf{t}}$	Vector containing estimated kinematic states
$\mathbf{c}$	Vector containing cluster data
$\mathbf{e}$	Vector containing errors
$\mathbf{m}$	Vector containing measurements
$\mathbf{s}$	Vector containing the sensor position
$\mathbf{t}$	Vector containing real kinematic states
$\Delta f_{bin}$	FFT-bin width
$\Delta R$	Range resolution
$\Delta V_r$	Radial velocity resolution
$\lambda$	Wavelength
$\vec{u}_x$	Unity vector pointing in x-direction
$\Phi$	State transition matrix
$\phi(\dots)$	State transition function
$\sigma_x$	Standard deviation (accuracy) of variable $x$
$\tau$	Radar signal delay
$\varphi$	Azimuth angle
$c$	Speed of light
$Cov(x)$	Covariance of $x$
$d^2$	Normalized statistical squared distance
$det(X)$	Determinant of matrix $X$

$df$	difference frequency
$E[x]$	Expectation of $x$
$f_s$	Sampling frequency
$f_{Doppler}$	Doppler frequency
$f_{hub}$	Frequency hub
$ff$	Fudge factor
$G$	Gaussian likelihood
$H$	Measurement matrix
$h(\dots)$	Measurement function
$J$	Jacobi matrix
$K$	Kalman gain
$k$	Cycle index
$P$	Covariance matrix
$PRF$	Pulse repetition frequency
$Prob$	Probability
$Q$	Process noise covariance matrix
$R$	Measurement noise covariance matrix
$r$	Range
$RCS$	Radar cross section
$T$	Cycle time
$T_{chirp}$	Chirp duration
$T_{pulse}$	Pulse width
ACC	Adaptive cruise control
CW	Continuous wave
DOP	Dilution of precision
EKF	Extended Kalman filter
FFT	Fast Fourier transform
FOL	Fused object list

---

FSK	Frequency shift keying
GC	Global cluster
GO	Global object
GPS	Global positioning system
HRR	High range resolution
ILSE	Iterated least squares estimation
JPDA	Joint probabilistic data association
LFM-CW	Linear frequency modulated continuous wave
LIDAR	Light detection and ranging
LKF	Linear Kalman filter
LS	Least squares
LSE	Least squares estimation
MMSE	Minimum mean square error
MR	Monopulse radar
PDF	Probability density function
RADAR	Radio detection and ranging
SSC	Single-sensor cluster
SSO	Single-sensor object
UKF	Unscented Kalman filter
UMRR	Universal medium range radar
UT	Unscented transformation
WLS	weighted least squares



# Bibliography

- [Ahr99] M. Ahrholdt. Optimierung des Sendesignals und der Signalverarbeitungsalgorithmen für ein LFM-CW Radar im Automobilbereich, Diplomarbeit TUHH, 1999.
- [Bac94] J. Bacher. *Clusteranalyse, Anwendungsorientierte Einführung*. R. Oldenbourg Verlag, München Wien, 1994.
- [Bau98] E. Baumann. *Punktbestimmung nach Höhe und Lage*, volume 2 of *Vermessungskunde*. Ferd. Dümmlers Verlag, Bonn, 6. bearb. Aufl. edition, 1998.
- [Bau99] E. Baumann. *Einfache Lagemessung und Nivellement*, volume 1 of *Vermessungskunde*. Ferd. Dümmlers Verlag, Bonn, 5. bearb. u. erw. Aufl. edition, 1999.
- [Boc74] H.H. Bock. *Automatische Klassifikation*. Vandenhoeck & Ruprecht, Göttingen, 1974.
- [BP99] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, Boston London, 1999.
- [BS90] Y. Bar Shalom. *Multitarget-Multisensor Tracking I*. Artech House, Boston-London, 1990.
- [BS92] Y. Bar Shalom. *Multitarget-Multisensor Tracking II*. Artech House, Boston-London, 1992.
- [BSF88] Y. Bar Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, Harcourt Brace Jovanovich Publishers, 1988.
- [BSL93] Y. Bar Shalom and X.-R. Li. *Estimation and Tracking - Principles, Techniques, and Software*. Artech House, Boston-London, 1993.
- [BSLK01] Y. Bar Shalom, X.-R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley & Sons, 2001.
- [CC98] C.K. Chui and G. Chen. *Kalman Filtering*. Springer, Berlin Heidelberg, 1998.
- [FBS90] R.J. Fitzgerald and Y. Bar-Shalom. *Multitarget-Multisensor Tracking, Development of Practical PDA Logic for Multitargettracking by a Microprocessor*. Artech House, 1990.

- [Föl00] F. Fölster. Entwurf und Realisierung einer Auswertelektronik für Beschleunigungssensoren zur Fahrbahnprädiktion im Kraftfahrzeug, Studienarbeit TUHH, 2000.
- [Föl01] F. Fölster. Multilateration in einem Radarnetzwerk für Kraftfahrzeuge, Diplomarbeit TUHH, 2001.
- [JU96] S. J. Julier and J. K. Uhlmann. A General Method for Approximating Nonlinear Transformations of Probability Distributions. 1996.
- [JU97a] S. J. Julier and J. Uhlmann. A New Extension of the Kalman Filter to Nonlinear Systems, 1997.
- [JU97b] S. J. Julier and Jeffrey K. Uhlmann. A Consistent, Debiased Method for Converting Between Polar and Cartesian Coordinate Systems, 1997.
- [Jul98] S. J. Julier. Reduced Sigma Point Filters for the Propagation of Means and Covariances through nonlinear transformations, 1998.
- [Jul00] S. J. Julier. The Scaled Unscented Transformation. In *Proceedings of the IEEE American Control Conference*, pages 4555–4559. IEEE, May 2000.
- [Kal60] R.E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering, ASME*, 82(Series D):35–45, 1960.
- [Kay93] S.M. Kay. *Fundamentals of Statistical Signal Processing Vol. I: Estimation Theory*. Prentice Hall, Signal Processing Series, 1993.
- [Klo02] M. Klotz. *An Automotive Short Range High Resolution Pulse Radar Network*. PhD thesis, Technische Universität Hamburg-Harburg, 2002.
- [LR03] U. Lubbert and H. Rohling. RASI, a Flexible Tool for Simulating and Developing Radar Systems. In *Proceedings of IRS 2003, Dresden*, pages 753–758, September 2003.
- [Lud98] A. Ludloff. *Praxiswissen Radar und Radarsignalverarbeitung*. Vieweg, 1998.
- [Mah98] B.R. Mahafza. *Introduction to Radar Analysis*. CRC Press, 1998.
- [MBM03] R. Mende, M. Behrens, and M.-M. Meinecke. The UMRR-S: A High-Performance 24 GHz Multi Mode Automotive Radar Sensor for Comfort and Safety Applications. In *Proceedings of IRS, Dresden*, pages 113–118, September 2003.
- [Mei01] M.-M. Meinecke. *Zum optimierten Sendesignalentswurf für Automobilradare*. PhD thesis, Technische Universität Hamburg-Harburg, 2001.
- [Men99] R. Mende. *Radarsysteme zur automatischen Abstandsregelung in Automobilen*. PhD thesis, Technische Universität Braunschweig, 1999.
- [MR02] R. Mende and H. Rohling. New Automotive Applications for Smart Radar Systems. In *Proceedings of GRS, Bonn*, pages 35–40, September 2002.

- [MV93] W. Mackens and H. Voß. *Mathematik I für Studierende der Ingenieurwissenschaften*. HECO Verlag, Alsdorf, 1. aufl. edition, 1993.
- [OR02] D. Oprisan and H. Rohling. Tracking Systems for Automotive Radar Networks. In *Proceedings of Radar 2002, Edinburgh*, pages 356–362, October 2002.
- [PS96] B.W. Parkinson and J.J. Spilker. *Global Positioning System: Theory and Applications Volume I*. AIAA (American Institute of Aeronautics and Astronautics), 1996.
- [RHLS02] H. Rohling, A. Höß, U. Lübbert, and M. Schiementz. Multistatic Radar Principles for Automotive RadarNet Applications. In *Proceedings of GRS 2002, Bonn*, pages 181–185, September 2002.
- [Roh01] H. Rohling. Radartechnik und -signalverarbeitung, Skriptum zur Vorlesung, 2001.
- [SFR03] M. Schiementz, F. Fölster, and H. Rohling. Angle Estimation Techniques for different 24 ghz Radar Networks. In *Proceedings of IRS 2003, Dresden*, pages 405–410, September 2003.
- [Sko80] M.I. Skolnik. *Introduction to Radar Systems*. McGraw-Hill, International edition, 1980.
- [Voß02] H. Voß. Grundlagen der Numerischen Mathematik, Skriptum zur Vorlesung, 2002.
- [Wie03] W. Wiesbeck. Radar System Engineering, Lecture Script, 2003.
- [WL90] E. Waltz and J. Llinas. *Multisensor Data Fusion*. Artech House, Boston London, 1990.
- [WM01] E. Wan and R. van der Merwe. *The Unscented Kalman Filter, Chapter 7 in Kalman Filtering and Neural Networks*. Wiley, 2001.
- [WS98] W. Weidmann and D. Steinbuch. A High Resolution Radar for Short Range Automotive Applications. In *Proceedings of 28th European Microwave Conference*, pages 590–594, September 1998.



# Index

- ABS sensor, 131
- ACC, 6
- All Neighbors Data Association, 110
- Association Methods, 110
- Association Probability, 107
- Automotive Radar Applications, 2
  
- Blind Spot Surveillance, 7
  
- Central-level Clustering, 119
- Central-Level Data Association, 123
- Central-Level Postprocessing, 118
- Central-Level Tracking, 124
- Cluster-to-Track Association, 104, 124
- Clustering, 101
- Coordinate Transformation, 51, 114
- Cross-Covariance Matrix, 154
  
- Data Association, 103
- Data Association Criteria, 106
- Data Fusion, 151
- Dilution of Precision (DOP), 61
- Doppler frequency shift, 9
  
- Experimental System, 129
- Extended Kalman Filter (EKF), 71
- Extended Normalized Squared Distance, 106
- Extended Targets, 18, 101, 119
  
- Fudge factor, 91
  
- Gating, 105
- Gauß-Newton method, 55
- Gaussian Likelihood, 107
  
- Highway Scenario, 134
- Host Vehicle Data Incorporation, 59
- HRR Pulse Radar Sensor, 23
  
- Innovation Vector, 106
  
- Interacting multiple model (IMM), 70
- Intersectioning(LFM-CW), 15
- Invention of RADAR, 1
  
- Joint Probabilistic Data Association (JPDA), 110
  
- Kalman Filter, 62
- Kalman Filter Comparison, 78
- Kinematic State Estimation, 48
  
- Lane Change Assistant, 7
- LFM-CW, 14
- Linear Kalman Filter (LKF), 64
- Linear Least Square Estimation, 53
  
- Maneuver Adaptive Kalman Filtering, 89
- Measurement Accuracy Definition, 22
- Minimum Variance Data Fusion, 151
- Missed Track Update, 116, 125
- Monopulse Technique, 10
- Moving Targets, 98
- Multi-sensor radar Network, 26
- Multilateration, 10
  
- Nearest Neighbor Data Association, 110
- Non-Moving Targets, 98
- Nonlinear Least Squares Estimation, 54
- Normal equations of the least squares problem, 53
- Normalized Squared Distance, 105
  
- Object Reflection Properties, 18
  
- Parallel Processing (EKF), 73
- Parking Aid, 6
- Path-Prediction, 97
- Piecewise Constant White Acceleration Model, 67
- Position Estimation, 48
- Position Estimation Accuracy, 30

- Postprocessing, 43, 94
- Pre-Crash Warning, 7
- Pulse Radar Measurement Principle, 12
- Pulse Repetition Frequency, 12
- Pulse Width, 12
- Pulse-Doppler, 13
  
- Radar Cross Section (RCS), 18
- Radar Equation, 18
- radar measurement Principles, 9
- Radar Network Concept Comparison, 26
- Range-Doppler Diagram, 15
- RASI, 132
- Resolution Conditions, 32
- Resolution Conditions (LFM-CW), 17
- Resolution Conditions (Pulse Radar), 12
  
- Sensor-Level Clustering, 101
- Sensor-Level Data Association, 103
- Sensor-Level Postprocessing, 95
- Sensor-Level Tracking, 112
- Sequential Lobing, 10
- Sequential Processing (EKF), 73
- Sigma Points, 75
- Sigma Vector, 75
- Simple Target Classification, 97
- Single Sensor Cluster, 103
- Singular Value Decomposition (SVD), 86
- Stop & Go, 8
  
- Target Clustering, 101, 119
- Target Dynamic Models, 67
- Test Vehicle, 129
- Track Deletion, 116, 126
- Track Evaluation, 116, 126
- Track Initialization, 116, 126
- Track Quality, 117
- Track Update, 112, 124
- Tracking Filter, 44, 62, 113
- Tracking Filter Decoupling, 86
- Tracking filter parameter, 126
- Triangulation, 48
- Trilateration, 10, 48
- Tunnel Scenario, 145
- Turn-Rate sensor, 129
  
- UMRR sensor, 23
  
- Unscented Kalman Filter (UKF), 74
- Unscented Transformation (UT), 74
- Urban Scenario, 138
  
- Velocity Estimation, 48
  
- Yaw-Rate, 60

# Lebenslauf

**Name:** Mark Schiementz  
**Geburtsdatum:** 06.03.1973  
**Geburtsort:** Hamburg

## Schulbildung

1979-1983: Grundschule in Fleestedt, Seevetal  
1983-1985: Orientierungsstufe Meckelfeld, Seevetal  
1985-1992: Gymnasium Meckelfeld, Seevetal

## Studium

10/1993-07/1999: Studium der Elektrotechnik an der TU Hamburg-Harburg mit der Vertiefungsrichtung Nachrichtentechnik/Digitale Signalverarbeitung  
02/1998-05/1998: Studienarbeit am Arbeitsbereich Nachrichtentechnik der TU Hamburg-Harburg  
12/1998-05/1999: Diplomarbeit am Instituto Universitari Audiovisual de l'Universitat Pompeu Fabra, Barcelona

## Berufliche Tätigkeit

09/1999-05/2004: Wissenschaftlicher Mitarbeiter am Arbeitsbereich Nachrichtentechnik der TU Hamburg-Harburg im Bereich der automobilen Radarsysteme  
seit 06/2004: Entwicklungsingenieur im Bereich Vorentwicklung von Fahrerassistenzsystemen (Entwicklung Fahrdynamik, CoC Sensorik) der BMW AG, München





