Data Fusion Architecture with Integrity Monitoring for State Estimation in Automated Driving

Architektur zur Sensordatenfusion mit Integritätsüberwachung für die Zustandsschätzung im automatisierten Fahren
Zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.)
genehmigte Dissertation von Grischa Guido Gottschalg, M.Sc. aus Düsseldorf
Tag der Einreichung: 11.04.2022, Tag der Prüfung: 19.07.2022

Referent: Prof. Dr.-Ing. Matthias Becker
Korreferent: Prof. Dr. rer. nat. Hermann Winner
Korreferent: Prof. Dr.-Ing. Andreas Eichhorn
Darmstadt – D17

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Bau- und Umweltingenieurwissenschaften

Institut für Geodäsie

Fachgebiet Physikalische Geodäsie und Satellitengeodäsie

Data Fusion Architecture with Integrity Monitoring for State Estimation in Automated Driving
Architektur zur Sensordatenfusion mit Integritätsüberwachung für die Zustandsschätzung im automatisierten Fahren

Genehmigte Dissertation von Grischa Guido Gottschalg, M.Sc.

Referent: Prof. Dr.-Ing. Matthias Becker
Korreferent: Prof. Dr. rer. nat. Hermann Winner
Korreferent: Prof. Dr.-Ing. Andreas Eichhorn

Tag der Einreichung: 11.04.2022
Tag der Prüfung: 19.07.2022

Darmstadt – D17

Diese Arbeit ist gleichzeitig veröffentlicht in der Reihe C der Deutschen Geodätischen Kommission, München 2022 und in der Schriftenreihe Fachrichtung Geodäsie der Technischen Universität Darmstadt, Darmstadt 2022

Preface

This dissertation has been written during my time as research associate at the Chair of Physical and Satellite Geodesy at the Technical University of Darmstadt, Germany. In particular, I would like to thank my supervisor *Prof. Dr.-Ing. Matthias Becker* for his continuous support throughout the years. During many fruitful discussions his guidance and suggestions helped me to determine the direction of this thesis and inspired me to perform my research.

I also want to thank my co-supervisors *Prof. rer. nat. Hermann Winner* and *Prof. Dr.-Ing. Andreas Eichhorn* for their interest in my research and their advice regarding the content of this thesis.

Special thanks to my colleagues at the Chair of Physical and Satellite Geodesy for the pleasant and constructive work environment. In particular, I want to thank *Dr.-Ing. Stefan Leinen* for his support, not only regarding my research, but also regarding all other topics at our Institute, e.g., during teaching, project related work or organizational topics. Many thanks to *Dr.-Ing. Björn Reuper* for drawing my attention to the field of Vehicle Dynamic State Estimation and explaining me his fusion filter, which serves as basis for this thesis. Thanks to all the students who supported me through their thesis or work as student assistants.

During my time at the Chair of Physical and Satellite Geodesy, I had the great opportunity to contribute to the research project UNICAR*agil*. I acknowledge the financial support by the Federal Ministry of Education and Research of Germany (BMBF), without whose funding this project and this dissertation would not have been possible.

Thanks also to all researchers participating in the project UNICAR*agil* for the good cooperation. Especially, I would like to thank our project partner iMAR Navigation GmbH for providing me the prototype hardware and granting me the opportunity to record measurements for my research early on. Likewise, I want to thank my colleagues at the Institute of Automotive Engineering (FZD) at the Technical University of Darmstadt for the pleasant and productive collaboration between the departments in the research project UNICAR*agil* and beyond.

Last but not least, I want to express my special gratitude to my family for always being there for me.

Abstract

In the advent of automated driving, numerous system architectures to reach this goal are currently being developed. In this process, safety and modularity criteria are becoming increasingly important. This opens up new opportunities in the system design, but also creates new challenges. All functions or services in an automated vehicle are affected by this, including the estimation of the vehicle's dynamic state. Requirements for the Vehicle Dynamic State Estimation demand integrity measures and high system reliability, which can often only be achieved by redundant structures.

A federated data fusion architecture with integrity monitoring is presented in this work to fulfill the aforementioned requirements for the Vehicle Dynamic State Estimation in automated driving. It inputs several redundant multi-sensor data fusion filters in a first fusion layer whose results are combined in a second fusion layer. This second layer implements plausibility checks and combines voting and data fusion, both utilizing the integrity measures of the redundant data fusion filters in the first layer.

To compute such integrity measures, three integrity monitoring concepts are developed which differ substantially in their error modeling and complexity. The first concept represents the traditional approach and relies on the fusion filter's estimated covariances assuming that the error is normally distributed. The second concept models the errors in the fusion filter's output as the sum of errors from the filter's inputs. The error of each input is modeled as a multi-variate Student distribution and propagated through the filter. Utilizing the resulting error distribution, the integrity measures are computed. The third concept is based on the principle of Multiple Hypothesis Solution Separation. Subsets consisting of parts of the available filter inputs are formed and the fusion for each of these subsets is computed. Integrity measures are deducted by comparing the subset's results with each other. Each of the three concepts outputs a protection level, which is compared to an alert limit for integrity monitoring.

Additionally, one of the redundant multi-sensor data fusion filters used in the federated fusion architecture is presented in this work. An Error-State Extended Kalman Filter is implemented to fuse the observations of three sensor types: a dual-antenna Global Navigation Satellite System Receiver, an Inertial Measurement Unit and wheel odometry sensors measuring wheel speeds and steering angles. The implementation includes sensor error models providing suitable error covariances for the filter's measurement updates and Fault Detection & Exclusion methods to increase the filter's robustness. Besides the measurement updates from the mentioned sensor types, also zero updates, i.e., zero velocity and zero angular rate updates, are part of the implementation, which are executed when a standstill of the vehicle is detected.

In order to evaluate the performance of the presented fusion filter, integrity algorithms and fusion architecture, an extensive set of measurements with a total duration of more than 23 hours is used. The measurements are divided into four categories according to their environmental and satellite reception conditions, since these have a strong influence on the filter's performance. All in all, the integrity requirements are met by all implemented algorithms in favorable satellite reception conditions. However, only the second concept using multi-variate Student distributions for error modeling does so in all measurement categories, from ideal satellite reception conditions in the test track category to very challenging environments in the urban category. The federated data fusion architecture also fulfills the integrity requirements with only one minor exception, maintaining or even reducing the empirical integrity risk depending on the measurement category. Additionally, the accuracy and availability is improved significantly in most cases, compared to the fusion filters in its first layer.

Finally, the usability of the developed concepts is demonstrated by applying them in a prototype vehicle of the research project UNICAR*agil*. The integration in the system architecture and the interaction with other services is presented. In measurements from the commissioning tests the functionality is illustrated.

Zusammenfassung (in German)

Im Rahmen der Entwicklungsarbeiten zum hochautomatisierten Fahren werden zahlreiche Systemarchitekturen entwickelt, um dieses Ziel zu erreichen. Die Aspekte Sicherheit und Modularität gewinnen dabei stetig an Bedeutung. Dies eröffnet neue Möglichkeiten bei der Systemgestaltung, schafft aber auch neue Herausforderungen. Davon sind alle Funktionen oder Dienste im Fahrzeug betroffen, auch die Schätzung des Fahrdynamikzustands. In Bezug auf die Fahrdynamikzustandsschätzung werden Integritätsmaße gefordert und hohe Anforderungen an die Zuverlässigkeit gestellt, die häufig nur durch redundante Strukturen erreicht werden können.

In dieser Arbeit wird eine föderierte Datenfusionsarchitektur mit Integritätsüberwachung vorgestellt, um die zuvor genannten Anforderungen an die Fahrdynamikzustandsschätzung im hochautomatisierten Fahren zu erfüllen. Mehrere redundante Multi-Sensor-Datenfusionsfilter werden in einer ersten Fusionsschicht angeordnet. Eine zweite Fusionsschicht empfängt deren Ergebnisse und prüft die Plausibilität. Anschließend werden Abstimmungsalgorithmen sowie eine weitere Datenfusion eingesetzt, die beide die Integritätsmaße aus der ersten Fusionsschicht nutzen.

Um diese Integritätsmaße zu berechnen, wurden drei Konzepte entwickelt, die sich hauptsächlich in Bezug auf ihr Fehlermodell sowie ihre Komplexität unterscheiden. Das erste Konzept entspricht dem traditionellen Ansatz, die im Fusionsfilter geschätzten Kovarianzen zu verwenden, wobei normal verteilte Fehler angenommen werden. Im zweiten Konzept werden die Fehler am Filterausgang als Summe der Fehler, die aus den Filtereingängen resultieren, modelliert. Die Fehler jedes Filtereingangs werden als mehrdimensionale Student-Verteilung modelliert und durch den Filter propagiert. Auf Basis der resultierenden Fehlerverteilung werden die Integritätsmaße berechnet. Im dritten Konzept werden Teilmengen der verfügbaren Filtereingänge gebildet und die Fusion wird für jede dieser Teilmengen berechnet. Die Integritätsmaße werden basierend auf einem Vergleich der Ergebnisse aus den Teilmengen ermittelt. Alle drei Konzepte geben eine obere Schranke für den Fehler als Integritätsmaß aus, die mit einer Warngrenze für die Integritätsüberwachung verglichen wird.

Außerdem wird einer der redundanten Fusionsfilter aus der ersten Fusionsschicht der föderierten Datenfusionsarchitektur vorgestellt. Es wird ein erweitertes Kalman Filter

verwendet, das die Fehler des Zustandsvektors schätzt und die Beobachtungen von drei Sensortypen einbezieht: ein Navigationssatellitenempfänger mit zwei Antennen, eine inertiale Messeinheit und Odometrie-Sensorik, die Raddrehzahlen und Radlenkwinkel misst. Die Implementierung enthält eine geeignete Modellierung der Sensor-Fehler sowie Mechanismen zur Erkennung von Ausreißern, um die Robustheit des Filters zu erhöhen. Zusätzlich zu den Messupdates der genannten Sensoren werden sogenannte Null-Updates für die Geschwindigkeit und Drehrate durchgeführt, wenn ein Stillstand erkannt wird.

Zur Validierung des entwickelten Fusionsfilters, der Integritätsmaße und Fusionsarchitektur wird ein umfangreicher Datensatz von Messungen mit einer Gesamtdauer von mehr als 23 Stunden verwendet. Die Messungen sind anhand ihrer Umgebungs- und Satellitenempfangsbedingungen in vier Kategorien eingeteilt, da diese Bedingungen einen starken Einfluss auf die Leistungsfähigkeit des Fusionsfilters haben. Alle entwickelten Integritätsmaße erfüllen die Anforderungen in günstigen Satellitenempfangsbedingungen. Allerdings erfüllt nur das zweite Konzept mit der Fehlermodellierung basierend auf Student-Verteilungen in allen Kategorien, von idealen Bedingungen auf der Teststrecke bis zu herausfordernden Situationen im städtischem Umfeld, die Integritäts-Anforderungen. Die föderierte Datenfusionsarchitektur erfüllt diese Anforderungen ebenfalls mit nur einer geringfügigen Ausnahme, wobei das Integritätsrisiko – je nach Kategorie der Messung – gesenkt oder gehalten werden kann. Des Weiteren wird die Genauigkeit und Verfügbarkeit im Vergleich zu den Filterergebnissen aus der ersten Fusionsschicht in den meisten Fällen deutlich verbessert.

Abschließend werden die entwickelten Konzepte in den Prototypenfahrzeugen des Forschungsprojekts UNICAR*agil* angewendet. Die Integration in die Systemarchitektur und die Interaktion mit anderen Diensten wird vorgestellt. Im Rahmen der Inbetriebnahme-Tests wird die Funktionsfähigkeit demonstriert.

Contents

Notation

The notation used in this work builds upon the book of Groves (2013) and the work of Reuper (2020).

Scalars

a	a scalar
$\breve{a}$	a true quantity (usually unknown)
$\tilde{a}$	a measured quantity
$\hat{a}$	an estimated quantity
$\delta\tilde{a} = \tilde{a} - \breve{a}$	a measurement error
$\delta\hat{a} = \hat{a} - \breve{a}$	an estimation error
a^-	an a-priori quantity
a^+	an a-posteriori quantity
a_k	a quantity a, given at a discrete time epoch k

Scalar Operations

$\dot{a}$, $\ddot{a}$, $\dddot{a}$	time derivatives of a
$\exp\{a\}$	equivalent to e^a

Vectors

$\mathbf{a}$	a vector
$\mathbf{a}^o$	a vector $\mathbf{a}$, resolved in the o-frame
$\mathbf{a}^o_{pq}$	a vector quantity $\mathbf{a}$ of q with respect to p, resolved in the o-frame

Vector Operations

$\mathbf{a} \times \mathbf{b}$	cross product
$[\mathbf{a}\times]$	cross product-forming matrix, $[\mathbf{a}\times]\,\mathbf{b} = \mathbf{a} \times \mathbf{b}$
$[\mathbf{a}\wedge]$	skew-symetric matrix of vector **a**, $[\mathbf{a}\wedge] = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}$ with $\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$
$\left.\dfrac{\partial \mathbf{a}}{\partial \mathbf{b}}\right\vert_{\mathbf{c}}$	partial derivative of **a** with respect to **b**, evaluated at **c**

Matrices

$\mathbf{A}$	a matrix
$\mathbf{0}_{m \times n}$	zero matrix with dimensions $m \times n$
$\mathbf{0}_m$	zero matrix with dimensions $m \times m$
$\mathbf{0}$	zero matrix (where the context gives the dimensions)
$\mathbf{I}_m$	identity matrix with dimension m
$\mathbf{I}$	identity matrix (where the context gives the dimension)

Matrix Operations

$\mathbf{A}^\mathsf{T}$	matrix transpose
$\mathbf{A}^{-1}$	matrix inverse
$\mathbf{A}^{\frac{1}{2}}$	square root of a matrix such that $\mathbf{A}^{\frac{1}{2}}\,\mathbf{A}^{\frac{1}{2}} = \mathbf{A}$
$\mathbf{A}'$	a matrix from previous step / iteration of KIPL algorithm
$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}$	a block matrix, consisting of four submatrices
$\det(\mathbf{A})$	determinant of matrix **A**
$\operatorname{diag}(\mathbf{a})$	diagonal matrix of vector **a**

$\mathrm{e}^{\mathbf{A}}$	matrix exponential $\mathrm{e}^{\mathbf{A}} = \sum_{k=0}^{\infty} \dfrac{1}{k!} \mathbf{A}^k$ with $\mathbf{A}^0 = \mathbf{I}$
$\mathrm{tr}(\mathbf{A})$	trace of matrix $\mathbf{A}$

List of Symbols

The symbols used in this work are inspired by the book of Groves (2013) and the work of Reuper (2020).

Coordinate Frames

a	antenna frame
b	body frame
e	Earth-centered Earth-fixed frame
f, r	individual wheel frame front/rear for single-track model
fl, fr, rl, rr	individual wheel frame front left/front right/rear left/rear right
i	Earth-centered inertial frame
n	local navigation frame
w	arbitrary wheel frame

$y^b_{pq,F}$, $y^b_{pq,L}$, $y^b_{pq,U}$	front/left/up component of vector $\mathbf{y}^b_{pq}$
$y^n_{pq,E}$, $y^n_{pq,N}$, $y^n_{pq,U}$	east/north/up component of vector $\mathbf{y}^n_{pq}$
$y^o_{pq,x}$, $y^o_{pq,y}$, $y^o_{pq,z}$	$x/y/z$-component of vector $\mathbf{y}^o_{pq}$

IMU

$\mathbf{b}_a$	3D vector of accelerometer biases
$\mathbf{b}_\omega$	3D vector of gyroscope biases
$\mathbf{f}_{ib}$	specific force acting on the b-frame with respect to the i-frame
$\mathbf{g}_{ib}$	gravity vector of the b-frame with respect to the i-frame
$\mathbf{w}_a$	accelerometer measurement noise
$\mathbf{w}_\omega$	gyroscope measurement noise

Kalman Filter

If not stated otherwise, all following symbols relating to the Kalman Filter are given as their discrete-time version.

$\mathbf{h}$	measurement function, $\tilde{\mathbf{z}} = \mathbf{h}(\mathbf{x}) + \mathbf{w}_m$
$\mathbf{H}$	measurement matrix, $\mathbf{H} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}$
$\mathbf{K}$	Kalman gain matrix
$\mathbf{n}$	state function, $\dot{\mathbf{x}} = \mathbf{n}(\mathbf{x}) + \check{\mathbf{w}}_s$
$\mathbf{N}$	continuous-time system matrix, $\mathbf{N} = \frac{\partial \mathbf{n}}{\partial \mathbf{x}}$
$\mathbf{P}$	error covariance matrix
$\mathbf{Q}$, $\check{\mathbf{Q}}$	system noise/continuous-time system noise covariance matrix
$\mathbf{R}$	measurement noise covariance matrix
$\mathbf{w}_m$	measurement noise vector
$\mathbf{w}_s$, $\check{\mathbf{w}}_s$	system noise/continuous-time system noise vector
$\mathbf{x}$	total state vector
$\mathbf{z}$, $\tilde{\mathbf{z}}$	error state formulation/total measurement vector

$\delta\mathbf{x}$	error state vector
$\delta\mathbf{z}^-$	measurement innovation, $\delta\mathbf{z}^- = \mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}^-)$
$\delta\mathbf{z}^+$	measurement residual, $\delta\mathbf{z}^+ = \mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}^+)$
τ	time interval
τ_s	state propagation interval
$\mathbf{\Phi}$	transition matrix

Kinematics

$\mathbf{a}_{pq}$	acceleration vector of q with respect to p
$\mathbf{C}_q^p$	direction-cosine matrix, $\mathbf{v}^p = \mathbf{C}_q^p \mathbf{v}^q$, $\mathbf{v}^q = {\mathbf{C}_q^p}^\top \mathbf{v}^p$
h_{eq}	ellipsoidal height of q, e.g., h_{eb} height of origin of body frame
$\mathbf{p}_{pq}$	position vector of q with respect to p
R_E	ellipsoidal radius of curvature in east-west direction, i. e., transverse radius of curvature
R_N	ellipsoidal radius of curvature in north-south direction, i. e., meridian radius of curvature

$\mathbf{v}_{pq}$	velocity vector of q with respect to p
η_{pq}	roll angle of the q-frame with respect to the p-frame
λ_{eq}	ellipsoidal longitude of q, e.g., λ_{eb} ellipsoidal longitude of origin of body frame
ν_{pq}	pitch angle of the q-frame with respect to the p-frame
φ_{eq}	ellipsoidal latitude of q, e.g., φ_{eb} ellipsoidal latitude of origin of body frame
ψ_{a}	antenna offset defined as angle in x^b-y^b-plane between x^b-axis and line connecting primary with secondary antenna, positive if connecting line points right of vehicle
ψ_{pq}	yaw angle of the q-frame with respect to the p-frame
$\boldsymbol{\psi}_{pq}$	attitude vector of the q-frame with respect to the p-frame, $\boldsymbol{\psi}_{pq} = \begin{pmatrix} \eta_{pq} & \nu_{pq} & \psi_{pq} \end{pmatrix}^{\mathsf{T}}$
ω	a scalar rotation rate
$\boldsymbol{\omega}_{pq}$	rotation rate vector of q with respect to p
$\boldsymbol{\Omega}_{pq} = [\boldsymbol{\omega}_{pq} \wedge]$	skew-symmetric matrix of rotation rate vector $\boldsymbol{\omega}_{pq}$
$\boldsymbol{\omega}_{en}$	transport rate (due to motion in a curvilinear frame)
ω_{ie}	scalar Earth rotation rate
$\boldsymbol{\omega}_{ie}$	Earth rotation rate vector

Odometry

$\mathbf{f}_{pq}$	specific force acting on q with respect to p
$\mathbf{F}_{pq}$	force acting on q with respect to p
h_{cog}	height of center of gravity above road surface (along z^b-axis)
l	wheelbase
l_f, l_r	distance from center of gravity to front/rear axle along x^b-axis
m_v	vehicle mass
r_d	dynamic tire radius
$\mathbf{r}_d$	vector containing all dynamic tire radii
α	tire side slip angle

β	vehicle side slip angle
δ_f	mean steering angle at front axle for single-track model
δ_{fl}, δ_{fr}	front left wheel/front right wheel steering angle
κ_x	longitudinal tire slip correction factor
λ_x	longitudinal tire slip
μ	friction coefficient
ω_w	wheel rotation rate

Parameter Estimation

S	power spectral density
ρ	correlation factor
σ_q	standard deviation of q
σ_{pq}	covariance of p and q

Integrity Monitoring

$\mathbf{A}_m$	matrix used as interim result in KIPL algorithm in Equation (5.16) and Equation (5.22)
B	Beta function $\mathrm{B}(x,y) = \int_0^1 t^{x-1}(1-t)^{y-1}\,dt, \quad \mathfrak{R}(x) > 0,\ \mathfrak{R}(y) > 0$
B	error bound
d	dimension of covariance matrix $\mathbf{R}$
$f_{N_m(\mathbf{R}_m)}$	multi-variate Student distribution
k	scalar factor
N_m	degree of freedom for measurement type m
N_S	number of subsets for ARAIM algorithm
p	parameter of a polynomial function
P_FA	probability of false alarm
$P(H_i)$	probability of a fault H_i present
P_HMI	probability of an integrity event with HMI
PL	protection level

q	counter
Q	complement of CDF of normal distribution
$\mathbf{R}_m$	covariance matrix for measurement type m
$\mathbf{S}$	matrix used as interim result in KIPL algorithm in Equations (5.18)–(5.21)
T_i	threshold for subset i of ARAIM algorithm
t_{m_1}, t_{m_2}, t_m	scalar values used as interim results in KIPL algorithm in Equations (5.17)–(5.20)
tr_m	trace of matrix $\mathbf{R}_m$
$\mathbf{y}_m$	measurement residual vector

α	integrity risk
α_i	tuning parameter of ARAIM algorithm
β	tuning parameter of KIPL algorithm
Γ	Gamma function $\Gamma(z) = \int_0^\infty x^{z-1} \, e^{-x} \, dx, \quad \mathfrak{R}(z) > 0$
ρ_m	tuning parameter of KIPL algorithm
$\boldsymbol{\rho}_m$	tuning matrix of KIPL algorithm, $\boldsymbol{\rho}_m = \rho_m \, \mathbf{I}$
ω	tuning parameter of KIPL algorithm

Second Fusion Layer

d	distance
n_s	number of filters in first fusion layer
T_min	threshold for approval voting
$\mathbf{T}$	matrix to store results of approval voting
w	weight
y	output of second fusion layer

List of Acronyms

1D, 2D, 3D, ...	one-dimensional, two-dimensional, three-dimensional, ...
ADAS	Advanced Driver Assistance Systems
AL	Alert Limit
ANN	Artificial Neural Network
ARAIM	Advanced Receiver Autonomous Integrity Monitoring
ARW	Angular Random Walk
ASOA	Automotive Service-Oriented Software Architecture
BKG	German Federal Agency for Cartography and Geodesy
CAN	Controller Area Network
CDF	Cumulative Distribution Function
CKF	Cubature Kalman Filter
CRS	Coordinate Reference System
CoG	Center of Gravity
DARPA	Defense Advanced Research Projects Agency
DIA	Detection, Identification and Adaption
DOP	Dilution of Precision
ECEF	Earth-Centered Earth-Fixed
ECI	Earth-Centered Inertial
ECU	Electronic Control Unit
eIR	empirical Integrity Risk
EKF	Extended Kalman Filter
ENU	East-North-Up
EU	European Union
ES-EKF	Error-State Extended Kalman Filter
ESA	European Space Agency
ESC	Electronic Stability Control
ETRS89	European Terrestrial Reference System 1989

FDE	Fault Detection and Exclusion
FFL	First Fusion Layer
FLU	Front-Left-Up
GBAS	Ground-Based Augmentation System
GFZ	GeoForschungsZentrum Potsdam, Germany
GNSS	Global Navigation Satellite System
GLONASS	Globalnaja Nawigazionnaja Sputnikowaja Sistema (Global Navigation Satellite System)
GPS	Global Positioning System
GRS80	Geodetic Reference System 1980
GSA	European GNSS Agency
HE	Heading Error
HMI	Hazardous Misleading Information
IBPL	Isotropy-Based Protection Level
IERS	International Earth Rotation and Reference Systems Service
IE	NovAtel WayPoint's Inertial Explorer
iMAR	iMAR Navigation GmbH
IMU	Inertial Measurement Unit
IR	Integrity Risk
KF	Kalman Filter
KIPL	Kalman Integrated Protection Level
LEO	Low Earth Orbit
LS	Least Squares
LiDAR	Light Detection and Ranging
MEMS	Micro-Electro-Mechanical System
MHSS	Multiple Hypothesis Solution Separation
MI	Misleading Information
NLOS	Non-Line-of-Sight
PDOP	Position Dilution of Precision
PE	Position Error
PF	Particle Filter
PL	Protection Level
PNT	Position, Navigation, and Timing

PPP	Precise Point Positioning
PSD	Power Spectral Density
PVT	Position, Velocity and Time
RADAR	Radio Detection and Ranging
RAIM	Receiver Autonomous Integrity Monitoring
RINEX	Receiver Independent Exchange Format
RLG	Ring Laser Gyroscope
RLS	Recursive Least Squares
RMS	Root Mean Square
RTK	Real-Time Kinematic
SBAS	Satellite-Based Augmentation System
SFL	Second Fusion Layer
SISRE	Signal-in-Space Range Error
SOP	Signals of Opportunity
SPP	Single Point Positioning
SiL	Software in the Loop
TTA	Time to Alert
TU Darmstadt	Technical University of Darmstadt
UAV	Unmanned Aerial Vehicle
UEE	User Equipment Error
UKF	Unscented Kalman Filter
US	United States (of America)
VDSE	Vehicle Dynamic State Estimation
VE	Velocity Error
VRW	Velocity Random Walk
WAAS	Wide Area Augmentation System
WSS	Wheel Speed Sensors
ZARU	Zero Angular Rate Update
ZVU	Zero Velocity Update

List of Figures

List of Tables

1. Introduction

Automated driving brings numerous promising benefits for future transportation. Watzenig et al. (2017, p. 3) name for example improvements in the areas of passenger safety by reducing driving errors caused by human failures, traffic flow, fuel consumption and carbon emissions as well as mobility of people who are unwilling or unable to drive themselves. A multitude of research projects in academia and the industry (e.g., Eberle et al., 2019; Urmson et al., 2008; Ziegler et al., 2014) is motivated by these benefits (Gottschalg et al., 2020). Depending on the specific requirements and objectives of the research project, the system architectures differ substantially. Nevertheless, there is a need to estimate the vehicle's dynamic state in all projects, since this information is needed in several essential functions, e.g., the trajectory control (Gottschalg & Leinen, 2021).

This work details a federated multi-sensor fusion architecture to estimate the vehicle's dynamic state and monitoring its integrity for an application in automated driving. In the following Sections, the motivation for this work and its objectives are presented, then the structure is explained.

1.1. Motivation[1]

As mentioned, system architectures for automated driving are diverse. However, they can be grouped by certain characteristics. With respect to the information flow, Yurtsever et al. (2020) distinguish between modular and end-to-end driving systems. In the latter, processing of all sensor data and computing of the control outputs for the actuators is concentrated in a single function. In contrast to that, modular systems split the tasks into separate modules or functions, e.g., environmental perception, behavioral and trajectory planning, vehicle dynamic state estimation and trajectory control.

While one can find some projects using end-to-end driving systems in combination with artificial intelligence (e.g., Lee and Ha, 2020), modular systems are becoming more and more popular. As an overview of modular system architectures for automated driving, Ulbrich et al. (2017) present a general functional system architecture. One of the

[1] This Section is based on a previous work written by the author (Gottschalg et al., 2021, Section 1).

modules in such an architecture is the ego-motion estimation or Vehicle Dynamic State Estimation (VDSE), which is needed in every project as mentioned before. Regarding the modularity inside the VDSE and the used sensors, there is again a great variety within the different projects across academia and the industry. Commonly, the VDSE is designed as a single function evaluating all sensor data, e.g., by Arribas et al. (2017). However, a modular approach for the VDSE comes with decisive opportunities regarding safety and practicality, which will be discussed in detail later on using the system architecture of a specific project as an example, namely the research project UNICAR*agil*.

In the research project UNICAR*agil*, more than 100 researchers from eight universities and eight industrial partners collaborate to develop disruptive modular architectures for agile automated vehicle concepts (Woopen et al., 2020). More information about the research project UNICAR*agil* will be given in Section 2.5. The system architecture in UNICAR*agil* is orientated towards modularity and safety, which also holds true for the VDSE. Figure 1.1 depicts the simplified system architecture of the VDSE in UNICAR*agil*, which implements a federated multi-sensor data fusion consisting of two fusion layers. In Chapter 6, details about this fusion architecture and reasons why it was chosen will be given. In the First Fusion Layer (FFL), sensor data from two Inertial Measurement Units (IMUs), one Global Navigation Satellite System (GNSS) receiver and wheel odometry sensors is fused in three redundant multi-sensor data fusion filters, which are developed by independent teams and provide integrity information (Buchholz et al., 2020). In the Second Fusion Layer (SFL), the VDSE's output is computed based on the FFL's results. Fault Detection and Exclusion (FDE), plausibility and integrity checks are used to increase the VDSE's robustness.

In order to take advantage of the benefits of such system architectures, suitable fusion filters and concepts for the integrity monitoring as well as for the SFL are needed. These are the objectives of this work, which are detailed out in the following Section.

1.2. Objectives

The overall goal of this work is to develop and evaluate a multi-sensor data fusion with integrity monitoring, which is used as VDSE for automated driving in projects orientated towards modularity and safety, e.g., the previously mentioned research project UNICAR*agil*. The state of current research lacks such a data fusion, which can be used in a modular system architecture and is providing integrity information, as it will be shown later on in Chapter 3. To meet this need, a fusion architecture implementing a federated multi-sensor data fusion consisting of two layers is designed. For the first layer, a multi-sensor data fusion inputing information from a Real-Time Kinematic (RTK)-GNSS receiver, a Micro-

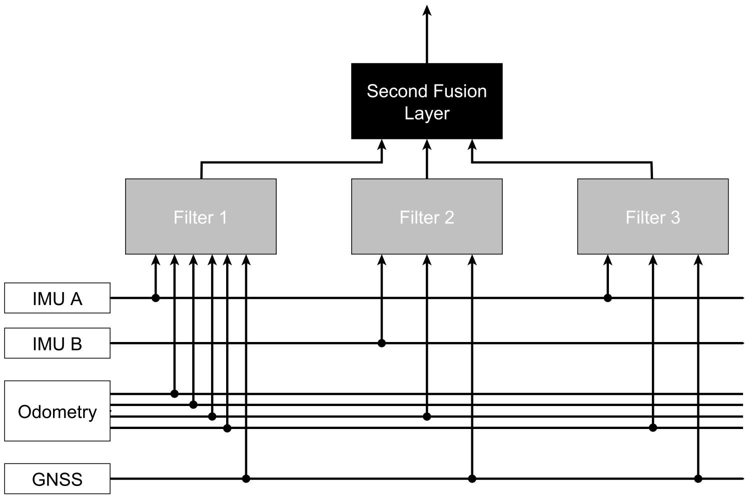


Figure 1.1.: Simplified system architecture for the Vehicle Dynamic State Estimation in UNICAR*agil* (Figure based on previous version published by the author in Buchholz et al., 2020)

Electro-Mechanical System (MEMS)-IMU and wheel odometry sensors is implemented. To evaluate the output's integrity, three integrity monitoring concepts are developed and compared regarding their empirical Integrity Risk (IR), meaning their ability to provide a Protection Level (PL) which is bounding the respective error according to the specified IR. For the second layer of the fusion architecture, a combination of voting and data fusion is designed which utilizes the integrity information. All developed concepts are evaluated using measurement data from driving experiments in diverse conditions.

In summary, the main objectives of this work are:

- design and implementation of a multi-sensor data fusion filter inputing information from a GNSS receiver, an IMU and wheel odometry sensors;

- development and comparison of integrity monitoring concepts for such a filter;

- development and evaluation of a federated multi-sensor data fusion architecture using several redundant fusion filters with integrity monitoring concepts as aforementioned for an application as VDSE for automated driving.

1.3. Outline

The remainder of this work is structured as follows: Chapter 2 explains the necessary background knowledge in terms of coordinate frames, sensors used for the estimation of the vehicle's dynamic state and integrity monitoring. Besides that, information about the research project UNICAR*agil* is provided. Afterwards, the relevant state of current research for this work is discussed in Chapter 3, including an analysis of its deficiencies. Subsequently, the multi-sensor data fusion filter developed in this work is explained in Chapter 4. After that, integrity monitoring concepts for such filters are compared in Chapter 5. Chapter 6 is about the developed two-layer fusion architecture and Chapter 7 presents the results corresponding to the three previous Chapters. After that, the application of the developed VDSE in the research project UNICAR*agil* is shown in Chapter 8. Finally, a conclusion and an outlook over future work is given in Chapter 9.

2. Background

This Chapter provides an overview of relevant background knowledge regarding the topics discussed in this work. First of all, the used coordinate frames are explained in Section 2.1. Then, sensors used for VDSE in automated driving are discussed in Section 2.2. Afterwards, the used sensor setup and test scenarios are described in Section 2.3. Subsequently, fundamental concepts and definitions regarding integrity monitoring are explained in Section 2.4. Finally, the research project UNICAR*agil* is introduced in Section 2.5.

2.1. Coordinate Frames

In the remainder of this work, several coordinate frames are used, among others to explain the multi-sensor data fusion filter in Chapter 4. Since the fusion filter builds upon the results from Reuper (2020), the coordinates frames are adopted. The original basis is Groves' book (2013, Ch. 2) in which more details about the used coordinate frames can be found and the notation used in this work is introduced.

Three coordinate frames are involved to describe Cartesian position, velocity, acceleration and angular rate: "The frame whose motion is described, known as the object frame, α; The frame with which that motion is respect to, known as reference frame, β; The set of axes which that motion is represented, known as the resolving frame, γ" (Groves, 2013, p. 44).

The notation follows as

$$\mathbf{x}^{\gamma}_{\beta\alpha}$$

"where the vector, $\mathbf{x}$, describes a kinematic property of frame α with respect to frame β, expressed in the frame γ axes" (Groves, 2013, p. 44). As Groves (2013, p. 44) points out, for the attitude only the frames α and β are involved. If any of the coordinate frames is irrelevant, it may be omitted (Reuper, 2020, p. 7). Additionally, this notation is also used in a generalized way to describe other vector quantities, e.g., forces (Reuper, 2020, p. 7).

Five coordinate frames are used in this work. Their axes form right-handed orthogonal sets. Each coordinate frame is defined in a dedicated Subsection in the following.

2.1.1. Earth-Centered Inertial Frame

A definition of the Earth-Centered Inertial (ECI) frame is given for example by Groves (2013, p. 25–26) on which this paragraph is based. Inertial frames do not accelerate or rotate with respect to the rest of the Universe. In this work, the ECI frame is treated as such an inertial frame, even though it is only approximately an inertial frame (see Groves, 2013, p. 25–26 for details). It is denoted by the symbol i and centered at the Earth's center of mass. The z^i-axis is orientated to the Earth's spin axis, the x^i- and y^i-axis form a right-handed coordinate system where the x^i-axis points from the Earth to the Sun at the vernal equinox. In navigation, inertial frames are important because inertial sensors measure motion with respect to these frames.

2.1.2. Earth-Centered Earth-Fixed Frame

The Earth-Centered Earth-Fixed (ECEF) frame is similar to the ECI frame: While the origin and z-axis are the same, x- and y-axis are fixed with respect to the Earth (Groves, 2013, p. 26). Therefore, it is denoted by the symbol e. The x^e-axis points towards the intersection of the equator with the prime meridian, for which the reference meridian of the International Earth Rotation and Reference Systems Service (IERS) is taken (Reuper, 2020, p. 8). The y^e-axis completes the right-handed coordinate system. With respect to the ECI frame, the ECEF frame rotates along the common z^e-axis with the rotation rate ω_{ie} (Reuper, 2020, p. 8). The ECEF is important in navigation, since the users want to know their position relative to the Earth (Groves, 2013, p. 27).

In many applications ellipsoidal coordinates (latitude λ_e, longitude φ_e and height h_e) are used, which approximate the Earth's surface by a reference ellipsoid. One example of a reference ellipsoid is defined by the Geodetic Reference System 1980 (GRS80) (Moritz, 2000). It is used in several Coordinate Reference Systems (CRS), e.g., in the European Terrestrial Reference System 1989 (ETRS89). A definition of ETRS89 is given by Altamimi (2018), who explains that it is attached to the stable part of the Eurasian tectonic plate and therefore the coordinates of a fixed point in Europe do not change over time. ETRS89 is used exclusively as CRS in this work, unless stated otherwise.

2.1.3. Navigation Frame

The navigation frame (also known as local navigation or topocentric frame) is denoted by the symbol n and has its origin at the user's point of interest, as given by Groves (2013, p. 27–28) on which this paragraph is based. In this work, the point of interest is the IMU's reference point, unless stated otherwise. The axis of the navigation frame align with the topographic directions East-North-Up (ENU), where the vertical axis is defined as normal to the surface of the reference ellipsoid pointing away from the Earth. The y^n-axis is the line from the user to the north pole projected in the plane orthogonal to the z^n-axis and the x^n-axis completes the right-handed coordinate system. The navigation frame is important in navigation since the users are interested in their attitude relative to the directions east, north and up.

2.1.4. Body Frame

The body frame (also known as vehicle frame) is denoted by the symbol b and shares its origin with the navigation frame. The x^b-axis points forward (in the usual direction of travel), the z^b-axis points upwards (opposite to the usual direction of gravity) and the y^b-axis completes the right-handed coordinate system (Groves, 2013, p. 29). Summarized, the axis directions can be described as Front-Left-Up (FLU).

To express the difference in orientation between the navigation and the body frame, Euler angles are used in this work, as given by Reuper (2020, p. 8–10) on which the following paragraphs are based. The Euler angles are the roll angle $\eta_{nb} \in (-\pi, \pi]$, pitch angle $\nu_{nb} \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ and yaw angle $\psi_{nb} \in (-\pi, \pi]$ and can be summarized in the attitude vector $\boldsymbol{\psi}_{nb}$:

$$\boldsymbol{\psi}_{nb} = \begin{pmatrix} \eta_{nb} \\ \nu_{nb} \\ \psi_{nb} \end{pmatrix}. \tag{2.1}$$

To transform a vector $\mathbf{y}$ from the body into the navigation frame and vice versa, the direction-cosine-matrix $\mathbf{C}_b^n$ is computed from $\boldsymbol{\psi}_{nb}$ and the following formulas are used:

$$\mathbf{y}^n = \mathbf{C}_b^n \mathbf{y}^b \tag{2.2}$$

$$\mathbf{y}^b = \mathbf{C}_n^b \mathbf{y}^n = \left(\mathbf{C}_b^n\right)^{\mathrm{T}} \mathbf{y}^n. \tag{2.3}$$

In the following, this transformation concept is generalized to any other coordinate frames replacing the indices n and b. The order of the rotations from the navigation to the body frame is defined as yaw, pitch, roll. All Euler angles are defined according to the right-hand screw rule around their respective axis. While roll and pitch angle are zero when the body frame is aligned to navigation frame, the yaw angle is zero when the forward axis of the body frame points towards east.

2.1.5. Wheel Frame

With the symbol w, the wheel coordinate frame is denoted. Its definition is based on Reuper (2020, p. 15): The wheel coordinate frame's origin lays in the center of the tire contact patch. The x^w-axis points forward, meaning it is aligned with the x^b-axis in case of a wheel's steering angle of zero. The y^w-axis points left and is parallel to the wheel's rotation axis. Together with the z^w-axis, which points up, normal to the road, they form a right-handed orthogonal set.

2.2. Sensor Types Used for Vehicle Dynamic State Estimation in Automated Driving

A variety of sensors is used in automated driving. As one example, Figure 2.1 depicts the sensors of the vehicles developed in the research project UNICAR*agil*, where the different sensor types are indicated by colored circles if they are in direct sight or by colored arrows otherwise. These include GNSS receivers, IMUs, wheel odometry sensors as well as Mono- and Stereo-Cameras, Light Detection and Ranging (LiDAR), Radio Detection and Ranging (RADAR) and ultra-sonic sensors. In the literature, these sensors are used in different combinations for several applications, including to estimate the vehicle's dynamic state (Gottschalg & Leinen, 2021; Kocic et al., 2018). Additionally, infrared beacons (Kampmann et al., 2021) and the mobile data network 5G (Wymeersch et al., 2017) are used for localization. This Section introduces the sensor types used in this work, which are GNSS, IMU and wheel odometry sensors.

2.2.1. Global Navigation Satellite Systems

Global Navigation Satellite Systems (GNSS) are used for a variety of Position, Navigation, and Timing (PNT) applications, including automated driving. In the following, relevant foundations about GNSS positioning are reviewed. Further information about GNSS can be found e.g., in (Langley et al., 2017) on which this Subsection is based.

In general, a GNSS consists of three components. The space segment includes a constellation of satellites orbiting the earth and broadcasting navigation signals on multiple carrier frequencies in the so-called L band (1 GHz to 2 GHz) (Reuper, 2020). The control segment is formed by a group of monitoring stations in charge of maintenance and provision of required navigation data. The user segment contains GNSS receivers using the GNSS signals. Currently, there are four GNSS in operation: GPS (US), GLONASS (Russia), Galileo (EU) and BeiDou (China).

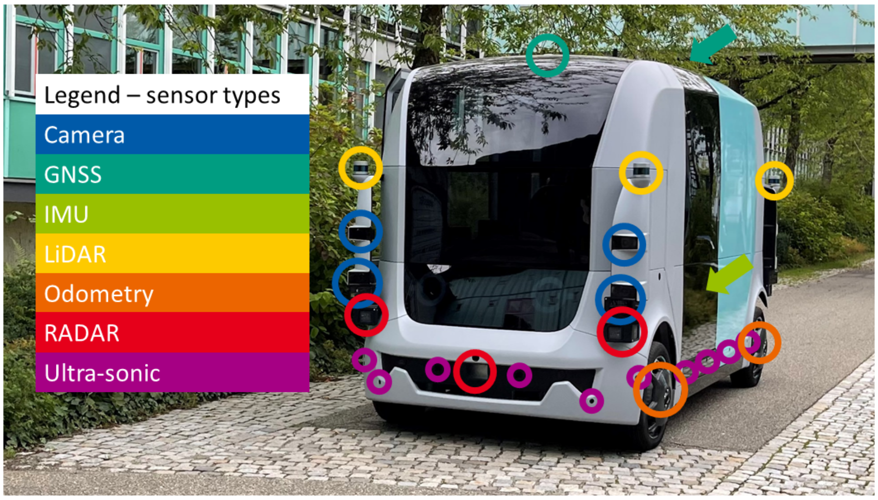


Figure 2.1.: Example of sensors used in automated driving – depicted prototype vehicle (autoCARGO) developed in research project UNICAR*agil* (photo by the author)

A GNSS receiver computes the Position, Velocity and Time (PVT) solution using pseudo-range, Doppler and carrier-phase measurements. Pseudo-range measurements use the time difference between satellite clock at transmission and receiver clock at reception. Doppler measurements rely on the change in the received frequency by the Doppler effect. For carrier-phase measurements, the phase shift is obtained by comparing the received signal to a reference signal of the nominal frequency. To convert this into a range measurement, the accumulated number of zero-crossings has to be obtained, which is called integer ambiguity problem. In differential GNSS, double differences of carrier-phase measurements between a pair of receivers and a pairs of satellites are formed. Usually, Least Squares (LS) or Kalman Filter (KF) estimation is employed to solve the integer ambiguity problem and resolve the position difference between the two receivers. If one receiver's position is known, the other receiver's position can be computed. For RTK positioning, this concept is used. A network of reference stations is transmitting correction data to a rover receiver, e.g., via mobile data. When the integer ambiguities are solved, the rover's position can be estimated with high accuracy (standard deviation in the range of centimeters). For GNSS receivers with two antennas (dual-antenna receivers), the concept of differential GNSS is employed as well. Double differences are formed to compute the vector between the two antennas (also known as baseline vector) and thus the baseline direction, which is utilized e.g., to determine a vehicle's attitude.

A GNSS receiver's PVT solution is influenced by a variety of error sources, which can be grouped into Signal-in-Space Range Error (SISRE) and User Equipment Error (UEE). While SISRE relates to errors in the space and control segment (e.g., satellite orbit and clock errors), UEE describes error contributions specific to the user's equipment and environment. These are e.g., multipath, Non-Line-of-Sight (NLOS) signal reception, unmodeled ionospheric and tropospheric delays as well as receiver noise. Besides these error sources, the receiver-satellite geometry also influences the accuracy of the PVT solution, which is represented by the Dilution of Precision (DOP) factor. Higher DOP values indicate a worse receiver-satellite geometry and therefore a lower accuracy.

Another important information about a GNSS receiver's solution accuracy is the solution type. When only pseudo-range measurements are used, the solution type is called Single Point Positioning (SPP). RTK positioning mainly relies on carrier-phase measurements and has two solution types, depending on whether the integer ambiguities are solved (RTK-fixed) or not (RTK-float).

2.2.2. Inertial Measurement Units

An Inertial Measurement Unit (IMU) typically combines three accelerometers and three gyroscopes with a processing unit and some peripheral equipment, e.g., a temperature sensor and a calibration-parameters store. The accelerometers measure the specific force and the gyroscopes the angular rate. Both sensor triads are usually mounted with orthogonal sensitive axes. The IMU processing unit prepares the output for the user, including the application of calibration parameters and transformation of observations to the IMU reference point. This Subsection about IMUs is based on the book of Groves (2013, Ch. 4).

In inertial navigation, the observations of accelerometers and gyroscopes are used in the navigation equations to compute velocities, positions and attitudes. Different versions of these navigation equations for different applications can be found in Groves' book (2013, Ch. 5). The navigation performance of an IMU highly depends on the measurement principle used by its sensors, especially the gyroscope. In this work, gyroscopes based on MEMS technology and Ring Laser Gyroscopes (RLGs) are used. While the former is substantially cheaper in its production and smaller in its packaging, the latter offers a significantly higher performance. Further information about the different measurement principles, sensor types and IMUs in general is given by e.g., Groves (2013, Ch. 4).

For many navigation applications including automated driving, IMUs and GNSS receivers are often combined because of their complementary characteristics regarding their benefits and drawbacks (Groves, 2013, p. 149, 559): IMUs offer a high output rate (typically between 100 Hz and 1000 Hz) and a high short-term accuracy. GNSS receivers provide outputs at a lower rate (typically around 10 Hz) but with a high long-term accuracy.

2.2.3. Odometry

Groves (2013, p. 233) defines odometry as “the determination of a land vehicle’s speed and distance traveled by measuring the rotation of its wheels”. Wheel Speed Sensors (WSS) are used to observe the rotation rate of the vehicle’s wheels and combined in many applications with sensors to observe the wheel’s steering angles which are potentially different for each wheel. In order to determine the vehicle’s speed and distance traveled from this information, odometry models are used (Reuper, 2020, p. 14). Depending on the complexity of the odometry model, the dynamic tire radii $\mathbf{r}_\mathrm{d}$ as well as the longitudinal slip λ_x and the lateral slip angle α are estimated for each wheel. Definitions for these quantities are given in Section A.2 in the Appendix.

Outputs of the odometry model are usually the wheel speeds in the body frame at each wheel, which are utilized by the sensor data fusion. Further information about odometry sensors and odometry models are given by e.g., Groves (2013, Ch. 6) and Guo (2018), respectively. Besides the traditional (wheel) odometry, there are also new forms of odometry, e.g., visual odometry using cameras or LiDAR odometry which are not implemented in this work.

2.3. Sensor Setup and Test Scenarios Used in This Work

This Section describes the sensor setup and the measurement data which are used in the remainder of this work. All measurements were recorded with the measurement vehicle of the Chair of Physical and Satellite Geodesy at the Technical University of Darmstadt (TU Darmstadt). Chapter 8 represents an exception, the differences in the sensor setup for the application in the research project UNICAR*agil* are mentioned in the referred Chapter.

2.3.1. Sensor Setup

In Table 2.1 the sensor setup used for the implemented fusion filter is depicted. Observations from a MEMS-IMU and a dual-frequency, multi-constellation, dual-antenna RTK-GNSS receiver are used. Precisely, GPS signals on frequencies L1, L2, Galileo signals on E1, E5b as well as GLONASS signals on G1, G2 are used. Additionally, the fusion filter

Table 2.1.: Sensors and processing of fusion filter and of reference solution (Table based on Gottschalg and Leinen, 2021)

	Implemented Fusion Filter	**Reference Solution**
Processing	Real-time capable (MATLAB code)	Post-processing (NovAtel WayPoint's Inertial Explorer 8.90)
IMU	Micro-Electro-Mechanical System (Sensonor STIM300 / Analog Devices ADIS 16465-1)	Ring Laser Gyroscope (iMAR iNAV-RQH1003)
GNSS	dual-frequency, multi-constellation, dual-antenna RTK-GNSS receiver (NovAtel OEM7720)	multi-frequency, multi-constellation, single-antenna RTK-GNSS receiver (NovAtel OEM729)
Odometry	production-line odometry (Volkswagen T5)	–

uses odometry observations from the production line sensors of the measurement vehicle, a Volkswagen T5 produced in 2008.

Figure 2.2 depicts the measurement vehicle mounted with the used sensor setup, where the white GNSS antennas and the black antennas for the cellular data connection to receive RTK corrections from the Internet via a mobile data connection are visible on the vehicle's roof.

The IMU for the measurements was changed in October 2020. Measurements before 26 October 2020, use a STIM300 (Sensonor AS, Horten, Norway) and after this date an

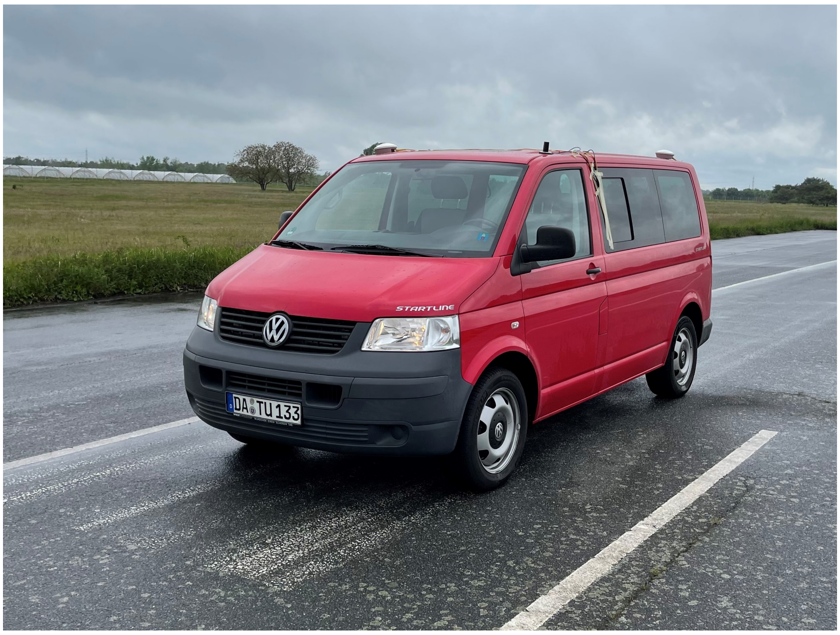


Figure 2.2.: Measurement vehicle of the Chair of Physical and Satellite Geodesy in May 2021 on TU Darmstadt airfield in Griesheim, Germany (photo by the author)

ADIS 16465-1 (Analog Devices Inc., Cambridge, MA, United States). These two IMUs belong to the same performance class where the STIM300 performs slightly superior according to its data sheet (see Table A.1 in the Appendix for a comparison of the used IMU's key characteristics). Both IMUs output observations with 500 Hz. However, five values are averaged and labeled with the time stamp of the end of the interval to reduce the amount of data and speed up processing, among other reasons, in order to fulfill the real-time requirements on the given hardware in the research project UNICAR*agil*. This procedure of summarizing IMU observations is based on functions of Reuper's fusion filter implementation (Reuper, 2020).

Furthermore, the sensor setup of the reference solution is depicted in Table 2.1 which is needed to compute the estimation errors. The reference solution is obtained from a post-processing evaluation in the software NovAtel WayPoint's Inertial Explorer 8.90 (NovAtel Inc., Calgary, AB, Canada), using observations from a navigation grade RLG-IMU (iMAR Navigation GmbH, St. Ingbert, Germany) and a multi-frequency, multi-constellation RTK-GNSS receiver (Gottschalg & Leinen, 2021). This GNSS receiver tracks signals from GPS, Galileo and GLONASS satellites on three frequencies.

Note that the odometry observations are not used for the reference solution. Experiments with typical trajectories including GNSS-denied areas showed that the reference solution does not benefit from these observations. On the one hand, the accuracy of the production-line odometry sensors in the measurement vehicle is not sufficient (in comparison to the employed RLG-IMU) to reduce the estimation error. On the other hand, odometry outliers were not reliably detected by used post-processing software.

In the Appendix in Section A.3 the quality of the reference solution is analyzed showing that the errors are approximately a magnitude smaller than the estimation errors of the developed fusion filter. Therefore, the errors of the reference solution are neglected and the difference between the two solutions is assumed to be the estimation error.

2.3.2. Experiments

In this work, there are two sets of measurement data. A first data set is used for testing during the development and parameter tuning of the implemented fusion filter and integrity algorithms. Additionally, an extensive data set is used for the evaluation of the implemented algorithm's performance. It contains representative scenarios for automated driving, which were recorded repeatedly to generate a higher statistical significance.

For further analysis of the results, the data sets are divided into four categories (Gottschalg & Leinen, 2021, Section 5.2): test track, highway, country road and urban. These categories represent typical environments for automated vehicles containing a variety of different GNSS reception conditions, which are especially relevant for the implemented fusion filter (Chapter 4) since it relies heavily on the performance of GNSS positioning.

Regarding the GNSS reception conditions, the test track represents the ideal environment with open sky view, while the difficulty increases in the highway category with bridges and overhead sign structures. On country roads the GNSS reception conditions are more challenging with vegetation, smaller buildings and tunnels, which is exceeded by the urban category with additionally higher buildings causing signal obstruction, multipath and NLOS reception.

Data Set for Parameter Tuning

The first data set consists of a single driving experiment recorded on 7 May 2019. It contains four rounds driving on a test track, a former military airport (August-Euler-Flugplatz) in Griesheim, Germany used by the TU Darmstadt as test track, and a drive through the city of Darmstadt, Germany including passing twice through a tunnel. The total duration of the measurement is about 1 h 20 min, of which around 13 min are spent on the test track. In the Appendix, Figure A.3 depicts a map of the driven trajectory.

Extensive Data Set for Performance Evaluation

The extensive data set consists of 20 driving experiments recorded between May 2019 and May 2021 with a total duration of about 23 h 40 min. It was recorded on the previously mentioned test track in Griesheim, in the city of Darmstadt, on a mountain range with forest areas next to Darmstadt, the Odenwald, and in the city of Frankfurt, Germany. Using the aforementioned categories, the key information about the extended data set is summarized in Table 2.2.

Maps of the driven trajectories are depicted in the Appendix in Figures A.4, A.5, A.6, A.7 and A.8. The GNSS reception conditions are indicated by colors, where yellow marks an observed Position Dilution of Precision (PDOP) value smaller than two, orange / purple stands for PDOP values smaller / greater than four. Only the epochs used for evaluation are shown in this trajectories. In two driving experiments, parts of the reference trajectory could not be processed with the expected accuracy (about a magnitude better than the analyzed system), because of software defects in NovAtel WayPoint's Inertial Explorer 8.90. In both cases, the reference trajectory contained jumps in the position solution with a magnitude of several decimeters. This might be caused by a failure to detect outliers but

Table 2.2.: Characteristics of extensive data set divided in four categories

	Test Track	**Highway**	**Country Road**	**Urban**
GNSS reception conditions	Ideal (open sky view)	Good (bridges and overhead sign structures)	Mixed (vegetation and small towns, including tunnels)	Challenging (downtown, including tunnels)
Locations	TU Darmstadt airfield Griesheim, Germany	Highway A5 near Darmstadt, Germany	Odenwald near Heppenheim, Germany	Darmstadt and Frankfurt, Germany
Duration	5 h 41 min	1 h 56 min	7 h 56 min	8 h 7 min

cannot be verified since the source code is not disclosed. Therefore the respective epochs had to be excluded from the evaluation and are not depicted in the mentioned Figures.

2.4. Integrity Monitoring[1]

The performance of a navigation system, like the VDSE in automated driving, can be quantified by four parameters given by Pullen (2008): accuracy, integrity, continuity and availability. In this work the following definitions based on Pullen's work (2008) are used:

- Accuracy expresses the deviation of the estimated position from the unknown true position[2]. (This definition applies analogously to other quantities, e.g., the velocity.)
- Integrity describes the level of trust that can be placed in the navigation system's outputs, meaning the reliance that gross errors are avoided.
- Continuity relates to the navigation system's reliability, where the continuity risk describes the probability that the navigation system will stop providing outputs of a specified quality during a given time interval.
- Availability describes the average probability that certain requirements with respect to the other three performance parameters are met. In this work, a navigation system is available as long as it outputs a solution and its integrity monitoring declares this solution as safe to use (meaning the solution is not declared unavailable).

Since a major focus of this work is the integrity of navigation systems in automated driving and how to monitor it, in the following, relevant definitions and background information regarding this topic are given.

As defined by Groves (2013, p. 701), integrity monitoring systems fulfill two tasks: On the one hand, they implement different levels of fault detection and mitigation. These range from detecting a fault and warning the user (Fault Detection) through recovering the navigation solution (Fault Detection & Recovery) or isolating it from the fault (Fault Detection & Isolation) to excluding the fault and verifying that the solution is free from faults (Fault Detection & Exclusion). On the other hand, integrity monitoring systems implement solution protection, i.e., to determine whether one can safely use the navigation solution.

Four parameters are used in integrity monitoring which are explained in the European Space Agency's online encyclopedia (ESA Navipedia, 2011), using the Position Error (PE) as an example:

[1] This Section is based on a previous work written by the author (Gottschalg & Leinen, 2021, Section 2).

[2] Note that thus precision and trueness contribute to the accuracy (Schwarz & Hennes, 2016, p. 7).

- Alert Limit (AL): "The alert limit for a given parameter measurement is the error tolerance not to be exceeded without issuing an alert."
- Time to Alert (TTA): "The maximum allowable time elapsed from the onset of the navigation system being out of tolerance until the equipment enunciates the alert."
- Integrity Risk (IR): "Probability that, at any moment, the position error exceeds the Alert Limit."
- Protection Level (PL): "Statistical bound error computed so as to guarantee that the probability of the absolute position error exceeding said number is smaller than or equal to the target integrity risk."

In this work, a stricter definition for the IR based on Pullen's work (2008) is used which describes the IR as the probability that an error exceeds its respective PL at any time.

The Stanford Diagram (also known as Stanford Integrity Diagram or Stanford Plot) is a common way to visualize these parameters. Figure 2.3 depicts such a diagram, using

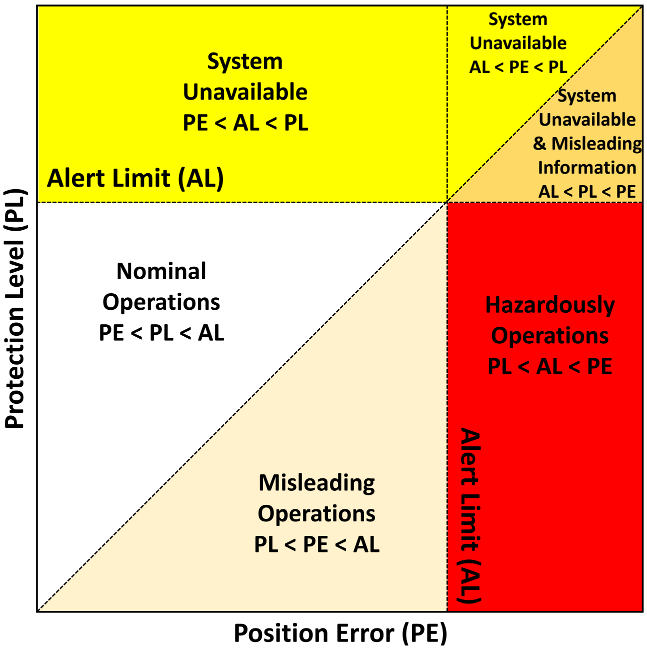

System Unavailable PE < AL < PL
Alert Limit (AL)
System Unavailable AL < PE < PL
System Unavailable & Misleading Information AL < PL < PE
Nominal Operations PE < PL < AL
Hazardously Operations PL < AL < PE
Misleading Operations PL < PE < AL
Alert Limit (AL)
Protection Level (PL)
Position Error (PE)

Figure 2.3.: Stanford Diagram (Figure based on ESA Navipedia, 2011)

the PE as an example. On the abscissa of the Stanford Diagram, the PE is plotted and the corresponding PL on the ordinate. As defined before, the PL is intended to bound the PE, which is the case in the area above the diagonal. The colors indicate the different operation modes of the navigation system. When the PL is smaller than the AL, the system operates normally. If the PL exceeds the AL, the integrity monitoring reports the system as unavailable. In case the PE is not bounded by the PL, the operation is called misleading, which is represented by the area under the diagonal. If additionally the PE actually exceeds the AL, this becomes a hazard, which is indicated by the color red. In this way, the Stanford Diagram gives a quick and clear overview of the systems integrity performance (ESA Navipedia, 2006).

2.5. Research Project UNICAR*agil*

The research project UNICAR*agil* started in 2018 and is funded by the Federal Ministry of Education and Research of Germany (Woopen et al., 2018). Eight universities and eight industrial partners collaborate to develop disruptive modular architectures for agile automated vehicle concepts, which are demonstrated in four fully automated and driverless vehicle prototypes of different characteristics (Woopen et al., 2018). Figure 2.4 depicts a conceptional drawing of the four prototypes. From left to right in the mentioned Figure, their use cases are a shuttle similar to buses in public transport (autoSHUTTLE), a delivery car (autoCARGO), a taxi (autoTAXI) and a privately owned vehicle (autoELF).

Automation, safety, security, verification & validation and modularization are the focus areas of the project UNICAR*agil* (Woopen et al., 2018). Regarding the system design, the UNICAR*agil* vehicles differ from traditional approaches, especially because of their focus on modularization and safety. Tasks are distributed between different modules which communicate via Ethernet using the Automotive Service-Oriented Software Architecture (ASOA), which is developed within the project and explained in further detail by Mokhtarian et al. (2020).

Looking at the domain of automation, the system architecture can be summarized as given by Buchholz et al. (2020): A set of sensors is installed in sensor modules on each corner of the vehicle, including LiDAR and RADAR sensors as well as cameras. These are used for environment perception and modeling which enables on the one hand a video-based localization and on the other hand the behavior and trajectory planing. This trajectory is sent to the motion control which consists of the VDSE, the trajectory control and the Offset-Correction. The VDSE uses sensor data from a dual-antenna RTK-GNSS receiver with two antennas mounted on the roof of the vehicle, two MEMS-IMUs inside the VDSE's ECU and wheel odometry sensors. It performs a sensor data fusion to estimate the

Figure 2.4.: Conceptional drawing of UNICAR*agil* vehicles (copyright by RWTH Aachen University as project coordinator)

vehicle's dynamic state. Additionally, the VDSE provides integrity information about the estimated states to other services, including the self-perception module. With the Offset-Correction, it is possible to take advantage of both localization functions in UNICAR*agil*, namely the VDSE and the video-based localization. It removes inconsistencies between the localization functions and extracts their benefits as explained by Homolla et al. (2020). The trajectory control receives the vehicle's dynamic state from the Offset-Correction and compares it to the planned trajectory. In case the normal trajectory cannot be used anymore or if a critical fault occurs, a fall-back solution is activated. The Safe Halt module provides an alternative trajectory which is used by the trajectory control, together with the VDSE's outputs.

Further details about the integration of the VDSE in the automation domain of UNICAR*agil* will be given in Chapter 8. More information about the project UNICAR*agil* in general, including a list of publications, can be found on the project website: https://www.unicaragil.de/en/.

3. State of Current Research

This Section gives an overview of the current state of research in the three areas relevant for this work: multi-sensor data fusion filter for VDSE (Section 3.1), integrity monitoring for multi-sensor data fusion (Section 3.2) and multi-layer data fusion (Section 3.3). Subsequently, the deficiencies in the state of current research are analyzed (Section 3.4).

3.1. Multi-Sensor Data Fusion Filter for Vehicle Dynamic State Estimation

The vehicle's dynamic state needs to be estimated in a variety of applications, including aviation for example in navigation of airplanes and Unmanned Aerial Vehicles (UAVs), marine applications like navigation of vessels and submarines, and automotive applications, e.g., Advanced Driver Assistance Systems (ADAS) and automated driving. Most often this estimation is based on a data fusion using several sensors, as mentioned in Section 2.2. For the data fusion itself, different approaches can be found in the literature (Gottschalg & Leinen, 2021): They can be grouped as filtering for example KF, snapshot methods like LS and other methods, e.g., Artificial Neural Networks (ANNs).

For automated driving, the vehicle's dynamic state is required in real-time as an input to subsequent functions like the trajectory control. Jin et al. (2019) present an overview of estimation techniques for the VDSE in automated driving which (depending on their implementation) fulfill this requirement. A similar grouping to the one previously introduced for the estimation methods is applied and a variety of different filtering approaches, including KF, Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), snapshot methods, including Recursive Least Squares (RLS), and other methods, such as ANN, is reviewed. Guo et al. (2018) also give an overview of VDSE concepts in automated driving, elaborating further on vehicle models and odometry processing. Both mentioned references include lists of sensor setups, estimation techniques and vehicle model combinations implemented in the literature. This illustrates the abundance of concepts and implementations regarding this topic.

Besides the applied estimation method, VDSE concepts can also be differentiated by the used sensor setup. Certain combinations such as GNSS+IMU are chosen frequently because of their complementary benefits as pointed out in Section 2.2. Since in this work a combination GNSS+IMU with additional measurement updates from wheel odometry sensors is implemented, the focus in the following is on combinations of GNSS+IMU+X. For this sensor setup, detailed explanations about the sensors itself, the processing of these sensor's observations, corresponding fusion algorithms and architectures are given in the books of Groves (2013) and Wendel (2007), which are considered as state-of-the-art for this work. In addition to the already quoted topics from these books, the challenges of preprocessing sensor data including the determination of sensor noise (sensor error models), Fault Detection and Exclusion (FDE), handling of delayed availability of sensor observations and real-time capability are particularly relevant in the context of this Section.

In order to differentiate system architectures using the sensor combination GNSS+IMU with respect to the way how GNSS observations are used, the terms loosely, tightly and deeply coupled integration are commonly used in the literature. These terms relate to the integration domain of the data fusion, as given by Groves (2013, p. 561): For a loosely coupled GNSS+IMU filter the position domain is used. This means a cascaded architecture is implemented in which the GNSS observations are processed in a navigation filter. This filter outputs the position and velocity solution which are the inputs to the GNSS+IMU filter. In contrast to that, in a tightly coupled GNSS+IMU filter the integration is performed in the range domain. The GNSS pseudo-range, pseudo-range-rate and carrier-phase observations are used as inputs to the GNSS+IMU filter, instead of the GNSS position and velocity solution. If even the tracking of GNSS signals is included in the GNSS+IMU filter, the integration is performed in the tracking domain and is called deeply coupled.

In terms of processing GNSS observations, decisive improvements have been made in the recent years. First of all, the number of GNSS constellations and frequencies usable for civil users have increased to four constellations with each at least two frequencies (ESA Navipedia, 2014). About ten years ago, many (automotive) applications were implementing SPP using code observations from one GNSS constellation on one frequency, typically GPS L1 C/A, e.g., the fusion filters presented by Gupta (2009) and Steinhardt (2014). In this way, standard deviations for the position solution in the range of meters were achieved. Nowadays, multi-constellation multi-frequency GNSS receivers are deployed, which take advantage of carrier-phase observations implementing typically Precise Point Positioning (PPP) or RTK positioning. Standard deviations for the position solution in the range of centimeters can be achieved under optimal circumstances, e.g., the implementations presented by Zhu et al. (2019) and Takanose et al. (2020). While PPP and RTK positioning, both need additional information supplementing the GNSS observations, only RTK positioning can solve the integer ambiguity quickly, usually instantaneous or within seconds for multi-constellation applications (Odijk & Wanninger, 2017, p. 772). In contrast to that, implementations of PPP typically need up to 15 min or more till a fixed solution is found (Kouba et al., 2017, p. 723). However, RTK positioning requires at least one reference station in a suitable distance with a permanent data link in order to operate properly, while PPP does not have this requirement, since the correction data (precise satellite orbits and clocks) is valid globally and can be downloaded in regular intervals.

For automotive applications, Reid et al. (2019) investigate the performance of RTK positioning on 30 000 km of highways in the US and conclude that multi-frequency GNSS receivers with RTK corrections achieve in over 90 % of the recorded epochs an accuracy better than 0.3 m compared to a reference solution, which allows to determine whether the vehicle is in its lane. In urban areas, additional challenges affect the performance of GNSS receivers including reduced satellite visibility, multipath or NLOS signal reception for example in urban canyons. Gakne and O'Keefe (2018) approach this challenge by combining a sky-pointing camera with a tightly-coupled GNSS+IMU filter. The camera is used to reduce NLOS satellite signal reception and for visual odometry. The use of wheel odometry to overcome the challenges for automotive applications in urban areas is discussed by Arribas et al. (2017). In the previously cited papers by Jin et al. (2019) and Guo et al. (2018) an overview of suitable vehicle models and implementations for odometry integration in the VDSE is given. Another concept for urban areas is presented by Meng et al. (2017) in which information from a point cloud based localization using a LiDAR sensor is combined with a GNSS, IMU and wheel odometry fusion. LiDAR sensors can also be used for automotive localization in combination with a high-definition map as shown e.g., by Wang et al. (2017).

One example of an implementation of a VDSE in automated driving is presented by Urmson et al. (2008) in context of the Defense Advanced Research Projects Agency (DARPA) Urban Challenge organized by the US Department of Defense. In this project, the localization relies on a particle filter, which inputs data from several sensors including GNSS, IMU, odometry sensors and laser scanners. Another implementation for VDSE in automated driving is presented by Ziegler et al. (2014) for Mercedes Benz' project "Making Bertha Drive". There, the localization is solely vision-based using a high-definition map taking advantage of features recognized in the camera images as well as lane-markings. Besides that, Wan et al. (2018) present their implementation of the VDSE in automated driving for the ApolloAuto project. The authors implement an EKF to fuse sensor data from GNSS, IMU and LiDAR sensors. These are only a few examples for this use case, many more can be found in the literature. In the following, the focus will be on the sensor setup used for this work.

Steinhard (2014) implements a sensor data fusion based on GNSS, IMU and odometry using an Error-State Extended Kalman Filter (ES-EKF) and presents his results also together with Leinen (2015) in the context of ADAS. Reuper (2020) builds upon these results, extends the GNSS processing to multi-frequency/multi-constellation GNSS and improves the odometry processing.

This work builds upon the results of Reuper (2020), whereby several changes in the GNSS processing were made, including the usage of a RTK-GNSS receiver in loose coupling. However, the main innovations of this work are the integrity monitoring and the additional fusion layer using the integrity information. Even though Reuper implements an a-posteriori variance factor to improve the integrity performance, this technique did not provide meaningful integrity information in his analysis (Reuper, 2020, Section 7.3), meaning the errors in position, velocity and yaw angle are not bound with the specified probabilities. The topic of an additional fusion layer is not discussed in Reuper's work. Details about the mentioned two main innovations of this work are explained in Chapter 5 and 6. The current state of research regarding these two topics is presented in the subsequent two Sections.

3.2. Integrity Monitoring for Multi-Sensor Data Fusion[1]

Integrity monitoring of estimated states is applied in a variety of research fields, including aviation, marine and automotive. Several names are used in the context of integrity monitoring, including integrity evaluation, estimation of data integrity and data reliability. In the literature an abundance of material can be found regarding this topic, wherefore the following literature review is restricted to integrity concepts which are relevant to this work. Since the field of application of this work is navigation and multi-sensor data fusion using GNSS receivers, the focus of the literature review will be on this area and the nomenclature from the domain of navigation is used.

The topic of integrity monitoring is intensively researched in aviation with a focus on the integrity of the estimated position solution. Initially, this estimation was implemented as a LS algorithm using sensor data from a single-frequency single-constellation (typically GPS L1 C/A) GNSS receiver as the only sensor (Yang & Xu, 2016). For GNSS, the integrity can be monitored at the system level, e.g., Satellite-Based Augmentation System (SBAS), Ground-Based Augmentation System (GBAS), or at the user level (Hassan et al., 2020).

Since the user has only influence on the integrity monitoring on his side, this work focuses on the user level. To evaluate the integrity at this level, Receiver Autonomous Integrity Monitoring (RAIM) methods are applied, which take advantage of the redundant measurements in the LS algorithm, meaning the overdetermined position solution computed from GNSS pseudo-range observations, to perform consistency checks (Kuusniemi, 2005). Additionally, Detection, Identification and Adaption (DIA) procedures are applied,

[1] This Section is based on a previous work written by the author (Gottschalg & Leinen, 2021, Section 3).

meaning a certain statistical distribution is assumed and a global test to detect outliers as well as, depending on the implementation, a local test to identify them is performed (Steinhardt, 2014).

In the decades following the initial use of RAIM methods in aviation, on the one hand, GNSS evolved and more constellations as well as more frequencies became operational and accessible for civil users. On the other hand, integrity requirements became stricter as technology advances. In this way, drawbacks of the first RAIM algorithms were identified, which include the use of a single-frequency single-constellation GNSS receiver and the general assumption of only one outlier present per epoch (Blanch et al., 2018; Tu et al., 2011).

As a result, researchers work on further developments and extensions of the original RAIM algorithm. In order to address the second drawback, several studies extend the concept of RAIM to multiple outliers, e.g., Tu et al. (2011) present a RAIM scheme based on a random sample consensus. Besides that, other researchers focus on the extension of RAIM to multi-constellation multi-frequency GNSS. Many studies in this field are carried out by the Stanford GPS Laboratory and are categorized as Advanced Receiver Autonomous Integrity Monitoring (ARAIM). A baseline version of ARAIM, which is an extension of RAIM to multi-constellation dual-frequency GNSS, is presented by Blanch et al. (2015). This LS version of ARAIM was adapted by Gunning et al. (2018) to a KF version for PPP. The concept was further developed by Gunning et al. (2019) to integrate IMU observations and perform a sensor data fusion for an improved position estimation. The concept of ARAIM was developed for aviation but can also be used for automotive applications as presented by Pullen et al. (2018), who describe the integration of ARAIM in the Globalstar Connected Car Program.

Furthermore, there are several other extensions of the original RAIM concept to GNSS+IMU data fusion algorithms, which focus on detecting outliers. For example, for a tightly coupled GNSS+IMU fusion Hewitson et al. (2010) present their concept of extended RAIM (eRAIM). Liu et al. (2010) also implement a tightly coupled GNSS+IMU fusion and categorize outliers in slowly/quickly growing or step errors and develop dedicated testing methods for each category. Another approach to integrate IMU observations into the integrity monitoring is proposed by Bhatti (2007) and referred to as piggyback architecture in which the IMU observations are converted to range measurements so they can be treated in the same way as GNSS observations.

In contrast to RAIM methods, which were originally developed for a system architecture implementing a LS algorithm using a single sensor, another group of integrity algorithms has been developed in the recent years, especially for multi-sensor data fusion. For different system architectures, multiple studies can be found in the literature, e.g., by Kumar et al. (2006) for detecting erroneous sensor data in a sensor data fusion using

Bayesian method and by Tanil et al. (2018) using sequential integrity monitoring for KF applications.

For a multi-sensor data fusion used as VDSE in automated driving, a dedicated approach to compute PLs is developed by GMV company and its partners. The basis therefor is the Isotropy-Based Protection Level (IBPL) introduced in a patent by Azaola Sàenz (2009), which is designed for a LS algorithm using GNSS pseudo-ranges. The PL is computed based on the LS residuals, scalar factors stemming from the DOP, the number of observations and the number of estimated states (Azaola Sàenz, 2009).

Further developments of IBPL led to an extension of this concept for system architectures based on a KF, which is referred to as Kalman Integrated Protection Level (KIPL). It is disclosed in a patent by Navarro Madrid (2016) and can be summarized as follows: All influences, meaning the error in each measurement input to the KF, are modeled dynamically by a multi-variate Student distribution for every input to the KF and are fused subsequently to a total error distribution, which is used eventually to compute the PL (Welte, 2017, Section 3.5). Tijero et al. (2018) present the results of applying KIPL in the ESCAPE (European Safety Critical Applications Positioning Engine) research project as an integrity layer providing PLs for a multi-sensor data fusion using a KF with sensor data from a GNSS receiver implementing PPP as well as IMUs, cameras and odometry sensors.

Another approach which is popular in the field of integrity monitoring is referred to as overbounding. In this concept, the actual empirical error distribution is bounded by a simpler distribution, e.g. a Gaussian distribution. This is advantageous since PLs and IRs can be computed easier from the overbounding distribution (Blanch et al., 2019). The baseline approach is to inflate a zero-mean Gaussian distribution to overbound all errors, which is practical since it leads to simple calculations for the integrity parameters such as the PL and IR (Blanch et al., 2005). A variety of publications can be found regarding this approach, which differ mainly in the combination of distributions used and their concept to determine them. For example, Blanch et al. (2005) combine several Gaussian distributions to overbound GNSS pseudo-range errors and apply their concept to compute an improved vertical PL for the Wide Area Augmentation System (WAAS) in aviation. Different overbounding strategies are compared by Rife et al. (2004).

As mentioned in the beginning of the Section, this literature review aims to give an overview of the most relevant developments in the field of integrity monitoring for multi-sensor data fusion in the application of VDSE. Further references can be found in the studies of Zabalegui et al. (2020), Zhu et al. (2018) and Hassan et al. (2020).

3.3. Multi-Layer Data Fusion[2]

As mentioned in the motivation (Section 1.1), modular system architectures provide several benefits but also create new challenges. In case of the VDSE in UNICAR*agil*, one of these challenges is the processing of FFL's filters results and the determination of a distinct output of the VDSE (see SFL in Figure 1.1). This challenge can be described as data fusion on several levels or multi-layer data fusion. In general, there are several approaches in the literature to solve this challenge, which can be categorized in voting, for example majority voting, and data fusion algorithms, e.g., KF. In this context, many key words are used, including cascaded and federated filters as well as voting schemes. These will be explained in more detail in the next paragraphs.

Groves (2013, Section 16.1) presents a variety of multi-layer data fusion architectures, including e.g., cascaded and federated filtered integration architectures. Figure 3.1 depicts a simplified representation of a cascaded integration architecture. Each sensor is processed in a dedicated processor, which can be implemented as a snapshot method or filtering. The outputs are processed in the SFL. Depending on the choice of algorithm for this layer, the architecture is called cascaded single-epoch or cascaded filtered integration architecture according to the definitions of Groves (2013, Section 16.1).

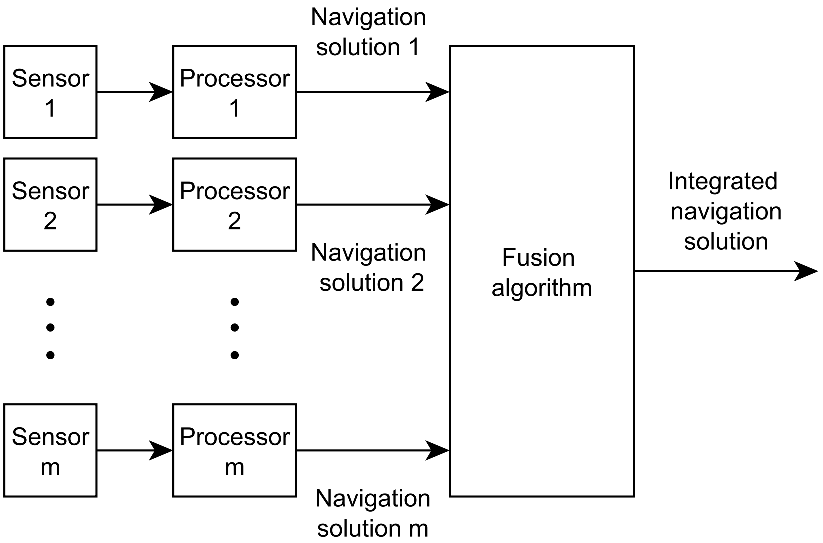


Figure 3.1.: Cascaded integration architecture (Figure based on Groves, 2013, p. 652)

[2] This Section is based on a previous work written by the author (Gottschalg et al., 2021, Section 2).

Similar to the cascaded integration architecture, in Figure 3.2 there is also an SFL algorithm processing the outputs of FFL processors. The differences lay in the inputs of the FFL processors: In contrast to before, the FFL processors are not dedicated to only one sensor but input data from several sensors. Typically, there is a reference navigation system or a certain sensor which is used by all FFL processors. In the nomenclature of Groves (2013, Section 16.1) this is called a federated integration architecture. In other references, this architecture with KFs as FFL and/or SFL is called federated KF. Then, the FFL filters and the SFL filter are referred to as local filters and master filter, respectively.

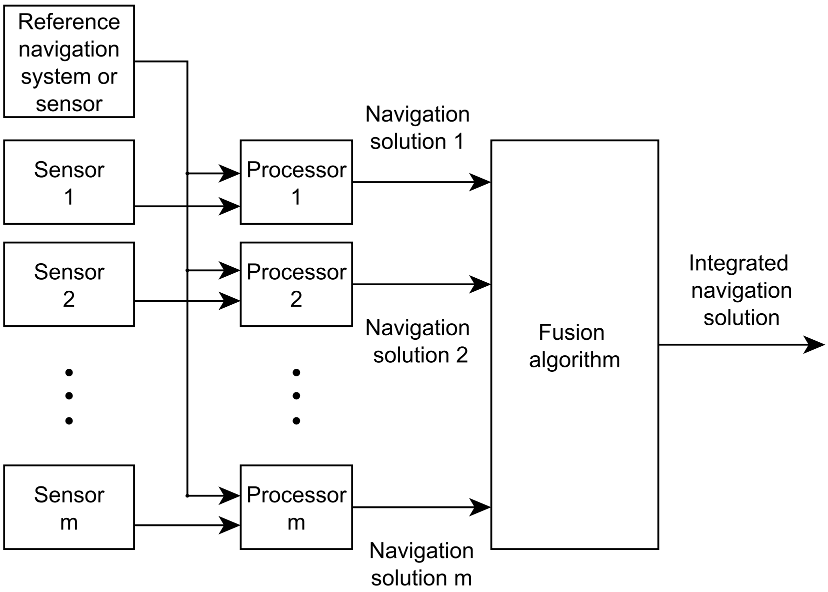
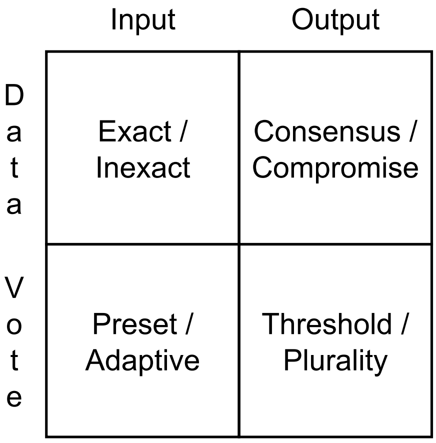


Figure 3.2.: Federated integration architecture (Figure based on Groves, 2013, p. 656)

For an application in aviation, Carlson et al. (1994) present simulation results and a comparison of different setups of federated KFs for a navigation system using GNSS, IMU and RADAR sensors. Ilyas et al. (2008) extend the concept of federated KF to the Unscented KF and apply it for a state estimation of satellites flying in Low Earth Orbits (LEOs). Guan et al. (2012) introduce the federated Cubature KF where the local filters in the FFL are Cubature KFs. The master filter in the SFL is implemented as weighted average and uses the FFL's local filter's covariances as weights, i.e., the master filter's covariance is computed as weighted average of the local filter's covariance using their magnitude relatively to the other filters' covariances as weights.

While in filtered integration architectures all FFL filters have an influence on the SFL's output, this is not necessarily the case in system architectures based on voting schemes. Generally, voting algorithms can be categorized by the taxonomy of voting schemes presented by Parhami (1996). Parhami introduces four dichotomies, which are depicted in Figure 3.3.



	Input	Output
Data	Exact / Inexact	Consensus / Compromise
Vote	Preset / Adaptive	Threshold / Plurality

Figure 3.3.: Classification of voting schemes by Parhami (Figure based on Parhami, 1996)

The following explanation of these dichotomies is based on Parhami (1996): The first dichotomy relates to the flexibility of the values, meaning if they represent inflexible (exact) values or a flexible ‘neighborhood’ (range of values). The way how the voting algorithm chooses one output is described by the second dichotomy, i.e., if the voting algorithm chooses one output based on consensus or if a value in-between the inputs is chosen, e.g., the median of real-valued inputs. The third dichotomy relates to the weights of the voting algorithm, specifically if they are fixed or if they change dynamically. The last dichotomy characterizes if a certain threshold of votes has to be reached to determine the output or if the output with the highest support from the inputs is chosen.

Furthermore, Parhami (2005) also introduces the term of approval voting. In this concept, each input votes for all outputs meeting certain criteria, i.e., one input supports multiple outputs, which splits the possibilities into acceptable and unacceptable outputs (Parhami, 2005).

For the case of a data fusion of several sensors measuring redundantly, e.g., a velocity measured by a GNSS receiver and by an odometry sensor, Blank et al. (2010) compare different fusion architectures, including weighted average, KF and different voting algorithms (threshold, median and fuzzy voter). They name this case redundant multi-sensor data fusion and conclude after a comparison in simulation that voting algorithms are usually more robust than the simpler solution of computing a weighted average (Blank et al., 2010). For the application of KFs, they point out that the variances and correlations of the inputs have to be determined or known in advance, which might be a disadvantage for certain applications (Blank et al., 2010). For the fusion of fusion filter's outputs or SFL data fusion in an application for navigation of land vehicles, Lan et al. (2020) compare KF and voting algorithms in simulation. They analyze the integrity of the output and conclude that both system architectures can fulfill this task but further analysis has to be carried out to identify the optimum areas of application for each system architecture.

Another application of multi-layer data fusion with a special focus on FDE is presented by Jurado et al. (2020) on which this paragraph is based on. The authors form subsets of the available sensors using all but one sensor for each subset. These subsets are processed in dedicated sub-filters. A test statistic is introduced to identify faulty sensors by determining the consistency of the sensor's observation with the filter's prediction before the measurement update. In a 2D simulation, the authors achieve promising results which have to be confirmed in experiments with real-world data.

As for the whole Chapter, this Section does not aim to include all research in its field, but focuses on the most relevant results for this work. Further information about multi-layer data fusion architectures and voting schemes in general are given by Groves (2013, Ch. 16) and Parhami (2005), respectively.

3.4. Analysis of Deficiencies in the State of Current Research

In this Section, the deficiencies in the state of current research regarding the objectives of this work are analyzed in order to underline the motivation of this work and enable the deduction of requirements in the following Chapters.

As pointed out in Section 3.1, there is an abundance of concepts and implementations for data fusion algorithms in the literature. However, the state of current research lacks a fusion architecture which combines the following characteristics.

First of all, the fusion architecture is required to be usable as VDSE for automated driving in real-time and providing meaningful integrity information. While several of the mentioned projects and works in Section 3.1 aim at a usage in automated driving, currently none of them provides meaningful integrity information in an automotive application.

As pointed out in Section 3.2, the majority of the integrity algorithms in the current state of research is designed for snapshot fusion methods like LS algorithms, and only uses a single sensor, e.g., a GNSS receiver. The concepts compatible with multi-sensor data fusion, are so far not investigated for an application in automated driving with the sensor setup of this work, consisting of a RTK-GNSS receiver, an IMU and odometry sensors.

Additionally, known fusion architectures are typically implemented as end-to-end systems. In contrast to that, modular system architectures exhibit several advantages, which are discussed in detail in Section 6.1. The current state of research lacks a fusion architecture combining the mentioned characteristics with a modular structure, e.g., a multi-layer fusion structure.

All in all, the deficiencies in the current state of research mentioned in the three previous paragraphs correspond to the three objectives of this work (see Section 1.2). Each of them is treated in a dedicated Chapter in the following (Chapters 4, 5, 6), where the requirements are deducted based on the results of the analysis from this Section.

4. Multi-Sensor Data Fusion Filter for Vehicle Dynamic State Estimation

This Chapter describes the multi-sensor data fusion filter developed in this work. First of all, an overview is given in Section 4.1 by discussing the filter's use case and system architecture as well as its flowchart. Afterwards, the implementation is explained in more detail (Section 4.2–4.3).

4.1. Overview

In this Section, the filter's use case for an application in automated driving is analyzed and central requirements are derived. Subsequently, the system architecture and design choices are discussed. Finally, an overview of the filter's implementation is given.

4.1.1. Use Case

The filter developed in this work is intended to be used in an automated vehicle to estimate its dynamic state. As mentioned in the introduction (Chapter 1), eventually all autonomous vehicles need to estimate their dynamic state as an input for subsequent services, especially for the motion controller (Gottschalg & Leinen, 2021). In order to enable these subsequent services to work properly, certain requirements have to be fulfilled.

For the general case of a VDSE used in automated driving, Steinhardt and Leinen (2015) summarize these requirements as follows:

- real-time capability and causal filter behavior,
- known latency which is as low and as stable as possible,
- robustness against sensor errors,
- resolution of the trade-off between accuracy and availability,
- output of consistent states for all subsequent tasks,

- output of integrity information.

Additionally, Steinhardt and Leinen (2015) list challenges for the VDSE with respect to the available measurement data:

- synchronous or asynchronous measurements with respect to other information sources or other sensors,
- different measurement resolutions,
- different sampling rates, which might not be constant,
- latency between the measurement itself and the availability of the measured values,
- changing availability of measured values,
- dependence on environmental conditions,
- dynamically changing measurement accuracy during operation.

Depending on the system architecture of the automated vehicle, the VDSE is integrated into another module or implemented as a dedicated unit. For this work, a modular system architecture similar to the one of the research project UNICAR*agil* is assumed. The VDSE implements in the specific case of UNICAR*agil* three redundant multi-sensor data fusion filters whose outputs are sent to an SFL algorithm as mentioned in the introduction in Figure 1.1.

Next, the previously mentioned general requirements are adapted for this work. Note that the developed filter can be used in system architectures like in UNICAR*agil* but also in other projects or setups. These alternative setups potentially do not implement the mentioned SFL algorithm but rather use only one filter as VDSE. To give an overview, Table 4.1 depicts the central requirements for the multi-sensor data fusion filter developed in this work in an abbreviated form. This list is intended to give a structure for the following discussion by introducing reference numbers for the mentioned central requirements. It does not aim to be comprehensive as details for example regarding the concrete implementation and interfaces are omitted for brevity.

The first requirement (R1), relates to the real-time capability of the filter. The filter's behavior has to be causal, meaning that only past measurements can be used for processing in the actual filter step. Additionally, the computational load has to allow an operation in real-time on the use case's hardware. For UNICAR*agil*, this means running on a μController board, which is discussed later on in Chapter 8. In order to fulfill the requirement of

Table 4.1.: Overview of central requirements for the multi-sensor data fusion filter developed in this work

Number	Description
R1	The filter needs to run in real-time on the hardware given in the specific use case such that the outputs can be provided at the specified rate.
R2	The filter has to be compatible with the use case's sensor setup and the data provided by these sensors.
R3	All required outputs have to be provided with sufficient accuracy and availability to enable subsequent services to operate properly.
R4	The filter's robustness against sensor errors has to be sufficient to operate in urban environments.
R5	The filter's outputs have to contain meaningful integrity information.

operating in real-time, it is for this work sufficient to provide the filter's outputs with the specified rate. For UNICAR*agil*, this is 100 Hz.

Besides that, the filter also has to be compatible with the given sensor setup which is the second requirement (R2). Precisely, this means the filter has to include a compatible sensor data processing for the given sensors, including sensor error models. Additionally, potential challenges with respect to the available sensor data have to be resolved which include the previously mentioned points from Steinhardt and Leinen (2015). For the sensor setup in UNICAR*agil*, this means that the filter uses sensor information from a dual-antenna RTK-GNSS receiver, a MEMS-IMU and from odometry sensors. This is also the sensor setup used in this work as described in Section 2.3.

In order to enable subsequent services to work properly, the filter has to not only provide the required outputs at the specified rate (see requirement R1) but also with sufficient accuracy and availability, which is mentioned in requirement R3. For UNICAR*agil*, the VDSE needs to provide the three-dimensional position, velocity, acceleration, attitude and angular rate of the vehicle. Since the accuracy and availability requirements were not determined at the start of the project, in the remainder of this work assumptions for the accuracy requirements are introduced (see Chapter 5, 7) and an availability as high as with the given setup possible is aimed for (best effort).

For the VDSE in automated driving, the environmental conditions have a crucial impact on the accuracy of the estimated states. In challenging environments, certain sensor observations may contain significant errors, e.g., GNSS observations in urban canyons. The fourth requirement (R4) states that the filter has to stay operational when sensor errors such as those that occur in challenging urban environments are present.

The requirement R5 relates to the need of integrity information regarding the estimated states. For subsequent services to know how much they can trust in the estimated states, this information is necessary. This topic will be motivated and discussed in detail in Chapter 5, after the explanation of the filter itself in this Chapter. In this context, also the topic of availability will be analyzed. Even though it is not explicitly mentioned in the table of requirements, it is a natural objective to maximize the filter's availability while fulfilling all other requirements.

4.1.2. System Architecture

For the fusion filter developed in this work a system architecture consisting of a preprocessing block for each sensor, the filter itself and an integrity layer is chosen, as depicted in Figure 4.1. In the following, the design choices leading to this architecture are discussed.

As mentioned before, the fusion filter in this work builds upon Reuper's PhD the-

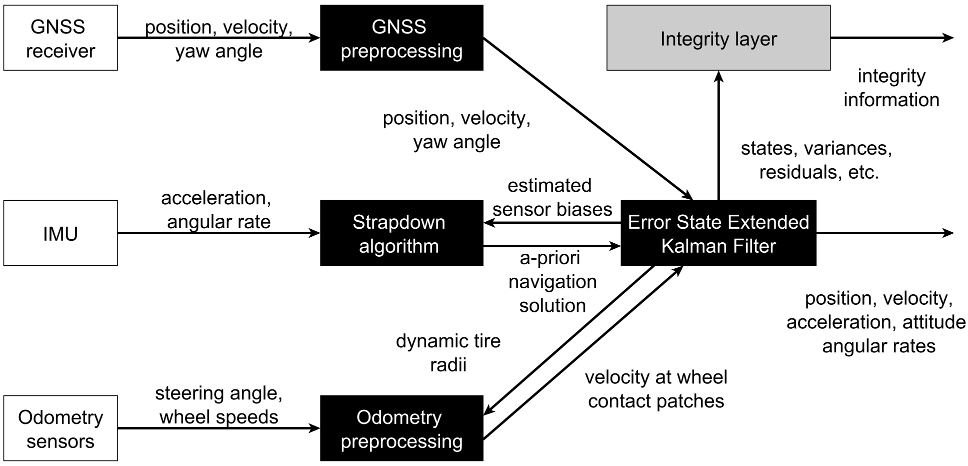

GNSS receiver
position, velocity, yaw angle
GNSS preprocessing
Integrity layer
integrity information
position, velocity, yaw angle
states, variances, residuals, etc.
IMU
acceleration, angular rate
Strapdown algorithm
estimated sensor biases
a-priori navigation solution
Error State Extended Kalman Filter
position, velocity, acceleration, attitude, angular rates
dynamic tire radii
velocity at wheel contact patches
Odometry sensors
steering angle, wheel speeds
Odometry preprocessing

Figure 4.1.: Block diagram of fusion filter developed in this work (Figure based on Gottschalg and Leinen, 2021)

sis (2020). Steinhardt's work (2014) served Reuper as basis and is therefore also used as reference here. Besides that, the textbooks of Groves (2013) and Wendel (2007) form the state of the art and are also used as basis for this work. Main differences to the fusion filter of Reuper are the GNSS processing and the integrity layer. Besides that, there are several smaller additions, e.g., the Zero Velocity Update (ZVU) and Zero Angular Rate Update (ZARU) (Gottschalg & Leinen, 2021).

For this work, a sensor setup similar to the one in UNICAR*agil* is used. Therefore, the fusion filter inputs sensor data from a GNSS receiver, an IMU and odometry sensors. In contrast to the implementation of Reuper (2020), in this work a dual-antenna RTK-GNSS receiver is used. Besides a PVT solution, the receiver also provides a heading solution (yaw angle). In order to take advantage of the already implemented RTK-GNSS processing in the receiver and its heading solution, a loose coupling of GNSS and IMU is chosen. This means the data fusion is performed in the position domain, i.e., the PVT and heading solution of the receiver are inputs to the fusion filter. The uncertainties provided by the GNSS receiver for these values are corrected in the GNSS processing module in form of a GNSS error model to fulfill the accuracy requirements, especially in challenging GNSS reception conditions like in urban areas.

The IMU observations (3D accelerations and angular rates) are fed into the Strapdown algorithm which implements the navigation equations to compute the dynamic state of the vehicle. An ES-EKF is chosen to estimate the errors of 19 states, which are depicted in Table 4.2. Based on these, the sensor biases can be computed, which are then fed back into the Strapdown algorithm. An EKF is used since the underlaying system dynamics are non-linear. By estimating only the errors of the states, in an ES-EKF, the influence of these non-linearities is limited. In an investigation of Zhao (2015) for a similar sensor setup (GPS+MEMS-IMU), he analyzed and compared different filter algorithms. From his conclusions it becomes clear that the additional computational load of non-linear filtering methods like Cubature Kalman Filter (CKF), Particle Filter (PF) or Unscented Kalman Filter (UKF) is only justified for higher non-linearities than the ones present in the use case of this work. In this way, also the requirement for real-time capability on the given hardware, which is assumed to be a μController like in UNICAR*agil*, is satisfied.

As preprocessing for the odometry observations, an odometry model computes the velocities at the wheel contact patches using the estimated dynamic tire radii. These are then fed as measurement updates into the ES-EKF. For conventional vehicles, as the measurement vehicle mainly used for this work, the odometry model of Reuper (2020) is used. For the application in the prototype vehicles of UNICAR*agil*, this model needs to be modified (see Chapter 8).

In order to increase the filter's accuracy, so-called zero-updates are implemented. ZVU and ZARU are triggered when a standstill of the vehicle is detected by the odometry

Table 4.2.: States of Error-State Extended Kalman Filter (Table based on Gottschalg and Leinen, 2021)

State	Dimension	Unit	Description
$\delta\Psi_{nb}$	3	rad	Misalignment
$\delta\mathbf{v}_{en}^{n}$	3	m/s	Velocity error
$\delta\mathbf{p}_{en}^{n}$	3	m	Position error
$\delta\mathbf{b}_{\omega}$	3	rad/s	Gyroscope offset error
$\delta\mathbf{b}_{a}$	3	$\mathrm{m/s^2}$	Accelerometer offset error
$\delta\mathbf{r}_{\mathrm{d}}$	4	m	Dynamic tire radius error

sensors, meaning that for more than $0.5\,\mathrm{s}$ all odometry sensors observe a velocity of zero (Gottschalg & Leinen, 2021). Additionally, a three layer outlier detection for the GNSS and odometry measurement update is implemented, which builds upon the results of Reuper (2020). First of all, the sensor data needs to pass a plausibility check. Innovation and residual monitoring form the second and third step, respectively.

Since the observations from the GNSS receiver and odometry sensors need to be transmitted to the fusion filter, they might arrive later than the following processing step of the ES-EKF. This challenge is referred to as delayed availability of measurement data. An approach based on the results of Reuper (2020) is used to resolve this challenge. Details about the implementation of this and the other parts of the explained system architecture are given in the remainder of this Chapter.

4.1.3. Implementation

This Section gives an overview of the implementation of the fusion algorithm developed in this work, whose flowchart is depicted in Figure 4.2. In the following Sections, the elements of this flowchart are explained. For the steps in boxes with double lines on the left and right side, there is a more detailed flowchart which is shown later in this Chapter, e.g., for IMU processing in Figure 4.3.

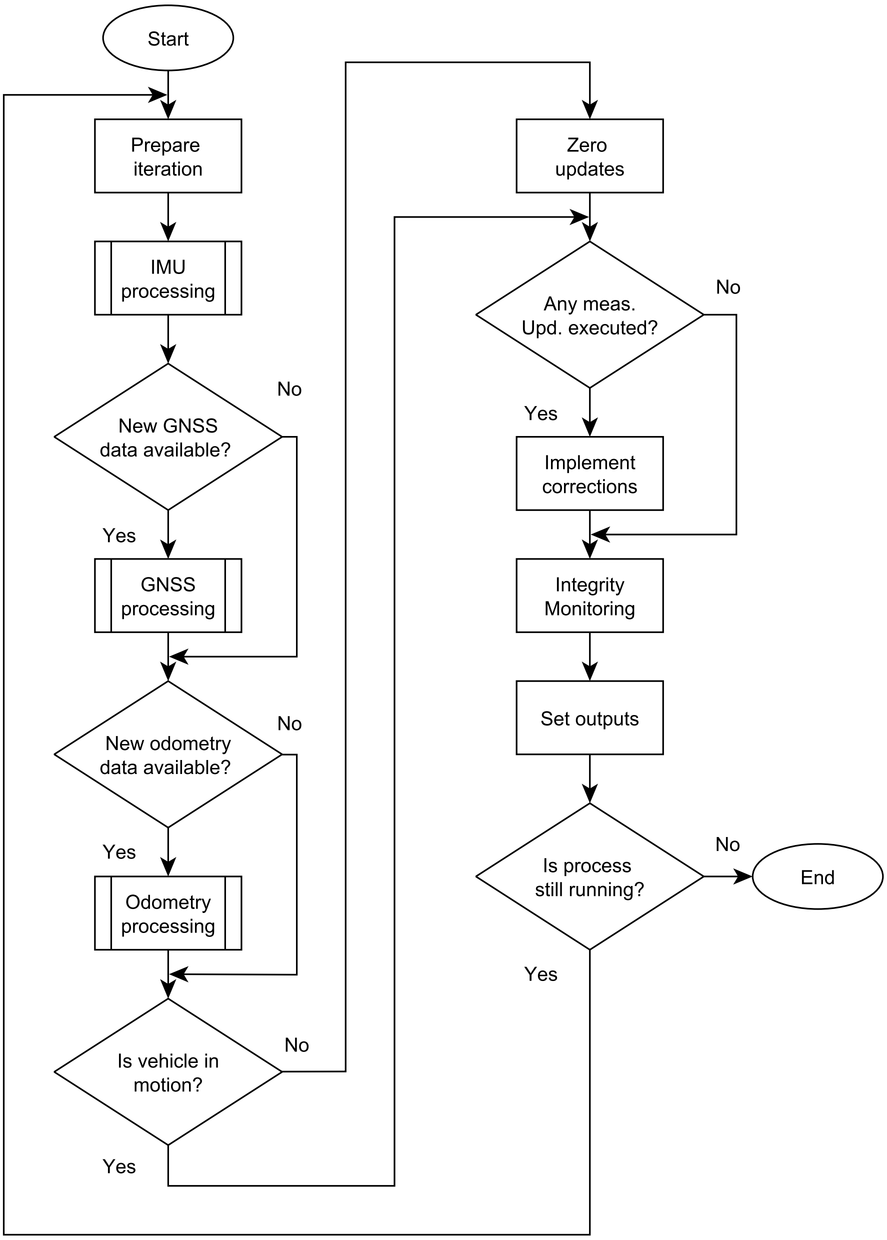


Figure 4.2.: Flow chart of fusion filter developed in this work

In general, the development and testing of this work's code is performed in MATLAB (Natick, MA, United States), partly also in MATLAB Simulink. For the subsequent application in real-time on a μController, C++ code is generated with MATLAB Coder and supplemented by a C++ wrapper handling the interfaces to sensors and subsequent services as well as to functions saving the results for debugging and evaluation purposes.

When a new set of IMU observations (3D accelerations and angular rates) is received, a new iteration of the fusion filter is started. Before each iteration, there is a preparation step to reset the temporary variables (used in the previous iteration) and save the received measurement data. Afterwards, the IMU observations are processed, which is explained in Section 4.2. In that step, also the initialization of the filter is performed, if necessary. Subsequently, the measurement updates for GNSS and odometry are executed if new observations are available. Additionally, zero updates are performed if the vehicle is in standstill. These measurement updates are discussed in Section 4.3. If any measurement updates are executed in this iteration, the computed corrections will be implemented, which is also described in Section 4.2. To conclude the iteration, the integrity monitoring is performed, which is discussed in Chapter 5, and the outputs are set.

4.2. Processing of IMU Data and System Model

This Section deals with the processing of IMU observations. Before that, the basics of the ES-EKF's implementation in this work including the system model and its initialization are explained.

4.2.1. Implementation of Error-State Extended Kalman Filter

According to the definitions of Groves (2013, Ch. 3) the fusion filter implements an ES-EKF in a closed-loop architecture estimating the error state vector $\delta\mathbf{x}$, which is defined as

$$\delta\mathbf{x} = \hat{\mathbf{x}} - \breve{\mathbf{x}} \tag{4.1}$$

with the estimated value of the total state vector $\hat{\mathbf{x}}$ and the true total state vector $\breve{\mathbf{x}}$. For the implementation of the ES-EKF the following equations are used (Reuper, 2020, p. 17):

Prediction step:

$$\delta\hat{\mathbf{x}}_k^- = \mathbf{0} \tag{4.2}$$

$$\mathbf{P}_k^- = \mathbf{\Phi}_k \mathbf{P}_{k-1}^+ \mathbf{\Phi}_k^\mathsf{T} + \mathbf{Q}_k \tag{4.3}$$

Measurement update:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^\mathsf{T} \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^\mathsf{T} + \mathbf{R}_k \right)^{-1} \tag{4.4}$$

$$\delta\hat{\mathbf{x}}_k^+ = \delta\hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H}_k \delta\hat{\mathbf{x}}_k^- \right) \tag{4.5}$$

$$\mathbf{P}_k^+ = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right) \mathbf{P}_k^- \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right)^\mathsf{T} + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^\mathsf{T}. \tag{4.6}$$

In Table 4.3, the matrices and vectors used in these equations are explained. Estimated quantities are marked with ˆ, the superscripts $-$ and $+$ indicate a-priori and a-posteriori quantities, respectively. The superscript ˇ indicates true values, while measured quantities are marked with ˜. The subscript denotes the temporal relation: k means that this quantity is given at the discrete epoch k.

To obtain the transition matrix $\mathbf{\Phi}_k$, a partial derivative of the state function $\mathbf{n}$ with respect to the total state vector $\mathbf{x}$ is formed (Reuper, 2020, p. 17):

$$\dot{\mathbf{x}} = \mathbf{n}(\mathbf{x}) + \check{\mathbf{w}}_\mathrm{s} \tag{4.7}$$

$$\mathbf{N}_k = \left. \frac{\partial \mathbf{n}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_k^-} \tag{4.8}$$

$$\mathbf{\Phi}_k = \mathrm{e}^{\mathbf{N}_k \tau_{s,k}} \tag{4.9}$$

with the state propagation interval at epoch k, $\tau_{s,k} = t_k - t_{k-1}$, and the matrix exponential function e. Analogously, the measurement vector $\mathbf{z}$ and the measurement matrix $\mathbf{H}$ are obtained (Reuper, 2020, p. 17–18):

$$\begin{aligned} \mathbf{z}_k &= \hat{\mathbf{z}}_k - \tilde{\mathbf{z}}_k \\ &= \mathbf{h}(\hat{\mathbf{x}}_k^-) - \tilde{\mathbf{z}}_k \end{aligned} \tag{4.10}$$

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_k^-}. \tag{4.11}$$

Note that

$$\delta\mathbf{z}^- = \tilde{\mathbf{z}} - \mathbf{h}(\hat{\mathbf{x}}^-), \tag{4.12}$$

$$\delta\mathbf{z}^+ = \tilde{\mathbf{z}} - \mathbf{h}(\hat{\mathbf{x}}^+) \tag{4.13}$$

are called measurement innovation and residual, respectively, and that

$$\delta\mathbf{z}_k^- = -\mathbf{z}_k \tag{4.14}$$

Table 4.3.: Symbols used in equations of Error-State Extended Kalman Filter (in the order of appearance)

Symbol	**Description**
$\delta\mathbf{x}$	Error state vector
$\mathbf{P}$	Error state covariance matrix
$\mathbf{\Phi}$	Transition matrix
$\mathbf{Q}$	Covariance matrix of system noise vector $\mathbf{w}_\mathrm{s}$
$\mathbf{w}_\mathrm{s}$	System noise vector
$\mathbf{K}$	Kalman gain matrix
$\mathbf{H}$	Measurement matrix
$\mathbf{R}$	Covariance matrix of measurement noise vector $\mathbf{w}_\mathrm{m}$
$\mathbf{w}_\mathrm{m}$	Measurement noise vector
$\mathbf{z}$	Measurement vector
$\mathbf{n}(\mathbf{x})$	State function
$\mathbf{N}$	System matrix
$\mathbf{h}(\mathbf{x})$	Measurement function

holds true in the error state formulation, unless a sequential measurement update is performed (Reuper, 2020, p. 18). For the measurement updates, it is assumed in this work that their measurement noise covariance matrices are uncorrelated and that the difference between batch and sequential measurement updates can be neglected, such that sequential measurement updates can be used to reduce the computational load (Reuper, 2020, p. 87).

4.2.2. System Model

The state vector $\mathbf{x}$ consists of 19 states and is given as:

$$\mathbf{x} = \begin{pmatrix} \boldsymbol{\Psi}_{nb} \\ \mathbf{v}_{en}^{n} \\ \mathbf{p}_{en}^{e} \\ \mathbf{b}_{\omega} \\ \mathbf{b}_{\mathrm{a}} \\ \mathbf{r}_{\mathrm{d}} \end{pmatrix} . \tag{4.15}$$

The error state vector $\delta\mathbf{x}$ is defined as difference between the true state vector $\check{\mathbf{x}}$ and the estimated state vector $\hat{\mathbf{x}}$ as given in Equation (4.1). However, in this implementation there is an exception for the position to improve the conditioning of the error state covariance matrix $\mathbf{P}$: The error state entry $\delta\mathbf{p}_{en}^{n}$ is resolved in the navigation frame in the unit meter, while the total state entry $\mathbf{p}_{en}^{e}$ is resolved in the ECEF frame and formed by latitude φ_e, longitude λ_e and height h_e in the units radian, radian and meter (Reuper, 2020, p. 88). The position of the IMU is chosen to be the reference point for the estimation in the fusion filter to simplify the equations. In case another position is needed for subsequent services, the outputs are transformed as given in the Appendix in Section B.5. The update of the total state for the position estimate is given as (Reuper, 2020, p. 88):

$$\hat{\varphi}_e^{+} = \hat{\varphi}_e^{-} - \frac{\delta\hat{r}_{en,N}^{n,+}}{R_N + h_e^{-}} \tag{4.16}$$

$$\hat{\lambda}_e^{+} = \hat{\lambda}_e^{-} - \frac{\delta\hat{r}_{en,E}^{n,+}}{\left(R_E + h_e^{-}\right)\cos\varphi_e^{-}} \tag{4.17}$$

$$\hat{h}_e^{+} = \hat{h}_e^{-} - \delta\hat{r}_{en,U}^{n,+} . \tag{4.18}$$

Another special case is the attitude

$$\boldsymbol{\Psi}_{nb} = \begin{pmatrix} \eta_{nb} \\ \nu_{nb} \\ \psi_{nb} \end{pmatrix} , \tag{4.19}$$

consisting of the three Euler angles roll η_{nb}, pitch ν_{nb}, yaw ψ_{nb}. The attitude is needed for the Strapdown algorithm as direction-cosine matrix, whose computation is described in the Appendix in Section B.6. Its update is computed as given by Groves (2013, p. 564):

$$\hat{\mathbf{C}}_b^{n,+} = \delta\hat{\mathbf{C}}_b^{n,+\mathsf{T}} \hat{\mathbf{C}}_b^{n,-} , \tag{4.20}$$

where $\delta\hat{\mathbf{C}}_b^{n,+}$ is computed as direction-cosine-matrix of $\delta\hat{\psi}_{nb}^{+}$ as implemented by Reuper (2020, p. 88).

The state vector in this work is identical to the one used by Reuper (2020, p. 88), except for the three states for the GNSS clock errors which are not needed in this work since here a loose coupling of GNSS and IMU is implemented. Therefore, also the system model is adopted, which is shown in the Appendix in Section B.1. For the system noise, the concept of Reuper (2020, p. 90–91) is adapted to the use case in this work by removing the not used states and updating the values from the IMU data sheets. Details about the system noise are also given in Section B.1 in the Appendix.

4.2.3. Initialization

In order to initialize the fusion filter, the following conditions have to be fulfilled:

- Initialization is requested, meaning filter is not initialized so far.
- Solutions of the GNSS receiver for position and velocity from the primary antenna as well as heading from the dual-antenna solution are available with fixed ambiguities.
- The magnitude of the GNSS velocity (3D norm) is small enough, such that a standstill can be assumed. This value is calibrated depending on the application and the used hardware. In this work a value of $0.1\,\mathrm{m/s}$ is used.
- The GNSS solution is plausible according to the plausibility criteria, which are depicted in Table B.2 in the Appendix for the use case of this work in automated driving.

The initialization itself consists of six steps, where the first three steps are based on Reuper's work (2020, p. 89–90). These consist of the initialization of the IMU biases according to the data sheet, initialization of the tire radii and their standard deviation according to the vehicle's tires and their wear over a lifetime as well as the initialization of the roll and pitch angle using a technique called leveling described in Groves (2013, Section 5.6.2). As a fourth step, the yaw angle is initialized using the GNSS heading solution. For the position initialization, the GNSS position solution is used in combination with the lever arm from the primary GNSS antenna to the IMU. To initialize the velocity, the GNSS velocity solution is assumed to be also valid at the IMU's position since the vehicle is standing still, as required in the third condition for initialization. Finally, the error state covariance matrix is initialized with the values given in Section B.4 in the Appendix.

4.2.4. Processing of IMU Data

The processing of IMU data is depicted in the flowchart in Figure 4.3. First of all, the plausibility of the IMU observations is checked by verifying their value range and sensor status. Secondly, the IMU observations are corrected by using the estimated biases, which are defined as

$$\tilde{\mathbf{a}}^b_{ib} = \breve{\mathbf{a}}^b_{ib} + \mathbf{b}_\mathrm{a} + \mathbf{w}_\mathrm{a} \tag{4.21}$$

$$\tilde{\omega}^b_{ib} = \breve{\omega}^b_{ib} + \mathbf{b}_\omega + \mathbf{w}_\omega, \tag{4.22}$$

where ($\tilde{\mathbf{a}}^b_{ib}$,$\tilde{\omega}^b_{ib}$) are the acceleration and angular rate values observed by the IMU, ($\breve{\mathbf{a}}^b_{ib}$,$\breve{\omega}^b_{ib}$) are the true values, ($\mathbf{b}_\mathrm{a}$,$\mathbf{b}_\omega$) are the sensor biases and ($\mathbf{w}_a$,$\mathbf{w}_\omega$) are the measurement noise terms (Reuper, 2020, p. 85).

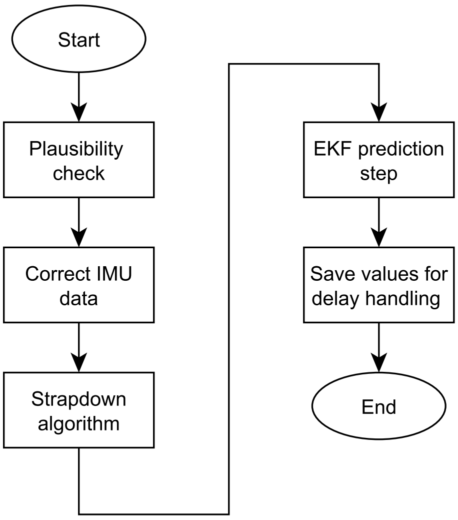


Figure 4.3.: Flow chart of processing IMU data

Subsequently, the Strapdown algorithm according to the description in Groves (2013, Section 5.4) is implemented in the navigation frame with the equations given by Reuper (2020, p. 86–87) to obtain the a-priori estimates of the latitude $\varphi^-_{e,k}$, longitude $\lambda^-_{e,k}$, height $h^-_{e,k}$ and the velocity $\hat{\mathbf{v}}^{n,-}_{en,k}$ as well as of the direction-cosine matrix $\hat{\mathbf{C}}^{n,-}_{b,k}$:

Attitude update:

$$\hat{\mathbf{C}}^{n,-}_{b,k} = \hat{\mathbf{C}}^{n,+}_{b,k-1}\mathrm{e}^{\boldsymbol{\Omega}^{b}_{nb}\tau_{i,k}} \tag{4.23}$$

Velocity update:

$$\hat{\mathbf{v}}^{n,-}_{en,k} = \hat{\mathbf{v}}^{n,+}_{en,k-1} + \tau_{i,k}\left[\frac{\hat{\mathbf{C}}^{n,+}_{b,k-1} + \hat{\mathbf{C}}^{n,-}_{b,k}}{2}\mathbf{a}^{b}_{ib} - \left(2\boldsymbol{\omega}^{n}_{ie} + \boldsymbol{\omega}^{n}_{en}\right) \times \hat{\mathbf{v}}^{n,+}_{en,k-1} - \mathbf{g}^{n}_{ib}\right] \tag{4.24}$$

Position update:

$$\begin{aligned}
h^{-}_{e,k} &= h^{+}_{e,k-1} + \frac{\tau_{i,k}}{2}\left[v^{n,+}_{en,U,k-1} + v^{n,-}_{en,U,k}\right] \\
\varphi^{-}_{e,k} &= \varphi^{+}_{e,k-1} + \frac{\tau_{i,k}}{2}\left[\frac{v^{n,+}_{en,N,k-1}}{R_N + h^{+}_{e,k-1}} + \frac{v^{n,-}_{en,N,k}}{R_N + h^{-}_{e,k}}\right] \\
\lambda^{-}_{e,k} &= \lambda^{+}_{e,k-1} + \frac{\tau_{i,k}}{2}\left[\frac{v^{n,+}_{en,E,k-1}}{\left(R_E + h^{+}_{e,k-1}\right)\cos\varphi^{+}_{e,k-1}} + \frac{v^{n,-}_{en,E,k}}{\left(R_E + h^{-}_{e,k}\right)\cos\varphi^{-}_{e,k}}\right].
\end{aligned} \tag{4.25}$$

In the attitude update, $\boldsymbol{\Omega}^{b}_{nb}$ is the skew-symmetric matrix of the rotation rate vector $\boldsymbol{\omega}^{b}_{nb}$, which is obtained from $\tilde{\boldsymbol{\omega}}^{b}_{ib}$ by applying corrections for the gyroscope bias $\mathbf{b}_{\omega}$, the Earth rotation rate $\boldsymbol{\omega}^{b}_{ie}$ and the transport rate $\boldsymbol{\omega}^{b}_{en}$ (Reuper, 2020, p. 86). Note that the matrix exponential function e has to be used in the attitude update. In the velocity update, $\mathbf{a}^{b}_{ib}$ represents the IMU's acceleration observation, corrected for the accelerometer bias $\mathbf{b}_{a}$, $\left(2\boldsymbol{\omega}^{n}_{ie} + \boldsymbol{\omega}^{n}_{en}\right) \times \hat{\mathbf{v}}^{n,+}_{en,k-1}$ is the Coriolis correction and $\mathbf{g}^{n}_{ib}$ represents the gravity correction (Reuper, 2020, p. 86). Additionally, the mean of $\hat{\mathbf{C}}^{n,+}_{b,k-1}$ and $\hat{\mathbf{C}}^{n,-}_{b,k}$ is implemented in the velocity update to account for the attitude change during the current IMU measurement interval $\tau_{i,k} = t_{i,k} - t_{i,k-1}$ (Reuper, 2020, p. 86).

This procedure is adapted analogously in the position update for the a-posteriori values from the previous epoch and the already updated a-priori values from the current epoch to account for the velocity change during the current IMU measurement interval $\tau_{i,k}$ (Reuper, 2020, p. 87). Furthermore, note that the reference ellipsoid's radius of curvature in north-south and east-west direction (meridian and transverse radius of curvature, respectively) R_E and R_N are computed based on the a-posteriori position estimate from the previous epoch $\hat{\mathbf{p}}^{e,+}_{en,k-1}$ (Reuper, 2020, p. 87).

Afterwards, the ES-EKF's prediction step is performed as explained in the beginning of this Subsection. To conclude the IMU processing, the total state vector is saved in a ring buffer for computations which are performed later on to deal with the delayed availability of measurements, which is discussed together with the measurement updates in the following Subsection.

4.3. Preprocessing and Measurement Models

In this Section the different types of measurement updates of the fusion filter are explained, which are the GNSS and odometry measurement updates as well as the zero updates, ZVU and ZARU. For each of them, the measurement vector $\mathbf{z}$ and the measurement matrix $\mathbf{H}$ as introduced in Equation (4.10)–(4.11) as well as the corresponding covariance matrix $\mathbf{R}$ are discussed. In case several updates can be executed in the same iteration, sequential measurement updates are used as mentioned before. To improve the readability in this Section, the reference to the iteration number k is omitted. Moreover, the outlier detection, which is implemented for GNSS and odometry measurement updates, is discussed in Section 4.3.4.

Since the used sensors are not necessarily synchronized and the communication with them is usually not instantaneous, the time stamp of the measurement vector $\tilde{\mathbf{z}}$ often differs from the time stamp of the fusion filter's total state vector $\hat{\mathbf{x}}$ and therefore also of the predicted measurement vector $\hat{\mathbf{z}}$ (Reuper, 2020, p. 94). This challenge is referred to as delayed availability of measurement data and is solved in this work by the approach described by Steinhardt (2014, Section 4.8.2): It is assumed that the changes in the error state for the time span of the delay t_d can be neglected such that corrections from the measurement's epoch can be implemented in the actual epoch and the error covariance matrix of the actual epoch is also valid for the measurement's epoch. The used quantities from the total state vector are interpolated linearly to perform the measurement update (Reuper, 2020, p. 94). These quantities are the position $\hat{\mathbf{p}}_{en}^{e}$, velocity $\hat{\mathbf{v}}_{en}^{n}$, angular rates $\hat{\omega}_{eb}^{b}$, $\hat{\omega}_{ib}^{b}$ and direction-cosine-matrix $\hat{\mathbf{C}}_{b}^{n}$ estimates. Both used angular rates are included here for practicality reasons to avoid the transformation between the two. The mentioned quantities are hold in a ring buffer of n elements, which are all corrected after each measurement update by the computed corrections to keep them up-to-date. For this work, $n = 100$ is chosen which allows a maximum delay of $1\,\mathrm{sec}$ with a fusion filter's frequency equal to $100\,\mathrm{Hz}$. Observations with a delay greater than this limit are discarded. This approach is applied for all GNSS and odometry measurement updates. Further details about the approach can be found in Steinhardt's work (2014, Section 4.8.2) and in the implementation of Reuper (2020, Section 6.3.4).

4.3.1. GNSS

Three measurement updates are based on the GNSS receiver's outputs. Based on the PVT solution, position and velocity updates are performed and the dual-antenna solution is used for the heading update. In Figure 4.4, the flowchart of processing GNSS observations in the fusion filter is depicted. Each decision in the flowchart is made separately for the

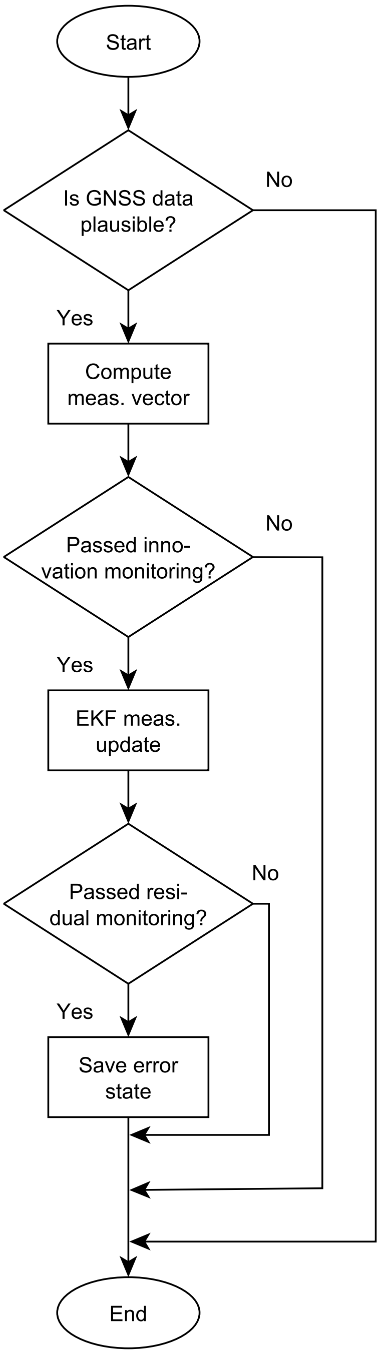


Figure 4.4.: Flow chart of processing GNSS data

three mentioned updates, e.g., it is possible that the position solution is not plausible and is therefore not used while the velocity solution is plausible and used.

The plausibility check is the first step in the flowchart (Figure 4.4). As a basis, the value range of the GNSS receiver's outputs is checked with the criteria given in Table B.2 in the Appendix as well as the special case if all values are zero, which would indicate the initialization phase of the receiver. Additional application specific checks can be added.

To prepare the ES-EKF's measurement update as given in Equation (4.4)–(4.6), in the second step the measurement vector $\mathbf{z}$ is computed. For the position update, the GNSS receiver outputs the position solution of the primary antenna in ellipsoidal coordinates as latitude $\tilde{\varphi}_{ea}$, longitude $\tilde{\lambda}_{ea}$, height $\tilde{h}_{ea}$, where the subscript a indicates that values at the antenna's phase center are meant. The measurement vector is computed using the equations given by Wendel (2007, p. 208) adapted to the navigation frame east-north-up and adding the lever arm from the IMU to the primary antenna's phase center $\mathbf{p}_{ba}^{b}$:

$$\begin{aligned}\mathbf{z}_{\text{pos}} &= \begin{pmatrix} z_{\text{pos},E} \\ z_{\text{pos},N} \\ z_{\text{pos},U} \end{pmatrix} \\ &= \begin{pmatrix} (\hat{\lambda}_{eb}^{-} - \tilde{\lambda}_{ea})(R_E + \hat{h}_{eb}^{-})\cos(\hat{\varphi}_{eb}^{-}) \\ (\hat{\varphi}_{eb}^{-} - \tilde{\varphi}_{ea})(R_N + \hat{h}_{eb}^{-}) \\ \hat{h}_{eb}^{-} - \tilde{h}_{ea} \end{pmatrix} + \hat{\mathbf{C}}_{b}^{n,-}\,\mathbf{p}_{ba}^{b}\end{aligned} \tag{4.26}$$

with the a-priori estimates for the latitude $\hat{\varphi}_{eb}$, longitude $\hat{\lambda}_{eb}$, height $\hat{h}_{eb}$ and the direction-cosine-matrix $\hat{\mathbf{C}}_{b}^{n,-}$. The equations for the velocity update given by Wendel (2007, p. 209) are adapted analogously:

$$\begin{aligned}\mathbf{z}_{\text{vel}} &= \begin{pmatrix} z_{\text{vel},E} \\ z_{\text{vel},N} \\ z_{\text{vel},U} \end{pmatrix} \\ &= \hat{\mathbf{v}}_{en}^{n,-} + (\hat{\mathbf{C}}_{b}^{n,-}\,\hat{\boldsymbol{\omega}}_{ib}^{b,-} \times \mathbf{p}_{ba}^{b}) - \tilde{\mathbf{v}}_{ea}^{n}\end{aligned} \tag{4.27}$$

with the a-priori estimates of the velocity $\hat{\mathbf{v}}_{en}^{n,-}$, angular rate $\hat{\boldsymbol{\omega}}_{ib}^{b,-}$ and direction-cosine-matrix $\hat{\mathbf{C}}_{b}^{n,-}$ as well as the velocity solution of the GNSS receiver $\tilde{\mathbf{v}}_{ea}^{n}$. The heading measurement vector can be computed simply as difference between the estimated yaw angle $\hat{\psi}_{nb}^{-}$ and the measured value $\tilde{\psi}_{nb}$ minus an antenna offset ψ_{a}, if present, as given by Angrisano (2010, p. 102):

$$z_{\text{hdg}} = \hat{\psi}_{nb}^{-} - \tilde{\psi}_{nb} - \psi_{\text{a}}. \tag{4.28}$$

The lever arm $\mathbf{p}_{ba}^{b}$ and the antenna offset ψ_a need to be calibrated beforehand, e.g., by a photogrammetric survey as done in this work for the measurement vehicle of the Chair of Physical and Satellite Geodesy at TU Darmstadt.

After computing the measurement vectors, the innovation monitoring is performed. This step will be explained together with the residual monitoring in Section 4.3.4. In case of not passing the monitoring checks, the respective measurement update is either not performed (innovation monitoring) or not used meaning its error state is discarded (residual monitoring).

Along with the measurement vector $\mathbf{z}$, the measurement matrix $\mathbf{H}$ is needed to perform the ES-EKF's measurement update in Equation (4.4)–(4.6). For the position update, the equations given by Wendel (2007, p.207) are adapted:

$$\mathbf{H}_{\text{pos}} = \begin{pmatrix} \mathbf{H}_{\text{pos},\psi} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} \end{pmatrix} \tag{4.29}$$

with

$$\mathbf{H}_{\text{pos},\psi} = -\left[\left(\hat{\mathbf{C}}_b^{n,-}\ \mathbf{p}_{ba}^{b}\right)\times\right], \tag{4.30}$$

where $\mathbf{0}_{m\times n}$ represents a $m \times n$ zero matrix and $\mathbf{I}_n$ stands for a n dimensional identity matrix. The symbol $[\mathbf{a}\times]$ represents the crossproduct matrix of the vector $\mathbf{a}$. Analogously, the velocity update is implemented by adapting the equations given by Wendel (2007, p. 209–210):

$$\mathbf{H}_{\text{vel}} = \begin{pmatrix} \mathbf{H}_{\text{vel},\psi} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{H}_{\text{vel},\text{b}_\omega} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} \end{pmatrix} \tag{4.31}$$

with

$$\mathbf{H}_{\text{vel},\psi} = -\left[\left(\hat{\mathbf{C}}_b^{n,-}\ \hat{\boldsymbol{\omega}}_{ib}^{b,-}\ \mathbf{p}_{ba}^{b}\right)\times\right], \tag{4.32}$$

$$\mathbf{H}_{\text{vel},\text{b}_\omega} = \hat{\mathbf{C}}_b^{n,-}\ \left[\mathbf{p}_{ba}^{b}\times\right]. \tag{4.33}$$

For the heading update, Groves (2013, p. 618) presents the equations which are adapted to the coordinate frames of this work:

$$\mathbf{H}_{\text{hdg}} = \begin{pmatrix} \mathbf{H}_{\text{hdg},\psi} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times4} \end{pmatrix} \tag{4.34}$$

with

$$\mathbf{H}_{\text{hdg},\psi} = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}. \tag{4.35}$$

Besides the measurement vector itself also a covariance matrix of the measurement noise $\mathbf{R}$ is needed for the measurement updates. In general, the standard deviations put out by the GNSS receiver are used here and non-correlated noise is assumed such that, e.g., the position measurement noise covariance follows as

$$\begin{aligned}\mathbf{R}_{\text{pos}} &= \operatorname{diag}\left(\sigma^2_{\text{pos},E},\ \sigma^2_{\text{pos},N},\ \sigma^2_{\text{pos},U}\right)\\ &= \begin{pmatrix}\sigma^2_{\text{pos},E} & 0 & 0\\ 0 & \sigma^2_{\text{pos},N} & 0\\ 0 & 0 & \sigma^2_{\text{pos},U}\end{pmatrix},\end{aligned} \tag{4.36}$$

with the standard deviations in east, north, up direction, $\sigma_{\text{pos},E}$, $\sigma_{\text{pos},N}$, $\sigma_{\text{pos},U}$, respectively. The velocity and heading measurement noise covariances $\mathbf{R}_{\text{vel}}$ and R_{hdg} follow analogously

$$\mathbf{R}_{\text{vel}} = \operatorname{diag}\left(\sigma^2_{\text{vel},E},\ \sigma^2_{\text{vel},N},\ \sigma^2_{\text{vel},U}\right) \tag{4.37}$$

$$R_{\text{hdg}} = \sigma^2_{\text{hdg}}. \tag{4.38}$$

Unfortunately, these standard deviations from the internal processing of the used GNSS receiver are over-optimistic, especially in challenging GNSS reception conditions like urban canyons, as several practical experiments showed. In order to enable the fusion filter to operate properly under these conditions, the received standard deviations from the GNSS receiver are modified. Additionally, the GNSS position update is only performed if the integer ambiguities are fixed, which is inspired by an implementation by El-Mowafy (2018). GNSS velocity updates are not restricted in this way, since they are effected less by typical error sources in urban environments, e.g., NLOS or multi-path signal reception. For the GNSS heading update, the procedure is similar to the position, the observation is only used for the respective update if the integer ambiguities for the dual-antenna solution are solved.

For the position measurement noise covariance matrix $\mathbf{R}_{\text{pos}}$, a dynamic scaling factor and a lower bound for the standard deviations put out by the GNSS receiver is introduced. The scaling factor depends on the GNSS reception conditions characterized by the PDOP and the number of used satellites n_{sat} as shown in Table B.3 in the Appendix. Each component (east, north, up) has a separate scaling factor. The mentioned characteristics are observed in the data set for parameter tuning. The scalar factors are fitted to represent the actual errors in this data set and then fixed for future use. The lower bounds aim to prevent over-optimistic standard deviations and are 0.02 m for the latitude's and longitude's standard deviation and 0.05 m for the height's standard deviation. Note that the lower bound is applied after the dynamic factor.

For the GNSS velocity solution, the experiments show that there is no need for a dynamic factor here, so only a lower bound of $0.4\,\mathrm{m/s}$ for the standard deviation in east, north direction $\sigma_{\text{vel},E}$, $\sigma_{\text{vel},N}$, and $0.5\,\mathrm{m/s}$ for the standard deviation in the upwards direction $\sigma_{\text{vel},U}$ is introduced.

Similarly for the heading's standard deviation σ_{hdg}, only a lower bound is introduced, which in this case is dynamic and depending on the number of satellites used for the dual-antenna solution. It was developed in a Bachelor's thesis by Bahle (2020) supervised by the author. The equation to compute this lower bound is

$$\sigma_{\text{hdg,min}} = \frac{a_1}{a_2 + n_{\text{sat},2}}, \tag{4.39}$$

with the tuning parameters $a_1 = 2.2306\,°$, $a_2 = -1.0493$ and the number satellites used in the dual-antenna solution $n_{\text{sat},2}$, where $\sigma_{\text{hdg,min}}$ again has a lower bound at $0.08\,°$.

4.3.2. Odometry

Figure 4.5 depicts the flowchart for the odometry measurement update. It is analogous to the GNSS measurement update except for the prepossing, which implements the odometry model.

As for any other sensor, the odometry preprocessing needs to be adapted to the used set of sensors. In this work, measurements from two vehicles with different odometry sensors are used. For the development of the fusion filter presented in this work, several measurements were conducted with a conventional vehicle described in Section 2.3.1. In this vehicle, the sensors for the Electronic Stability Control (ESC) can be accessed via the Controller Area Network (CAN) bus. Only the wheels at the front axle are driven and can be steered. In contrast to that, in the prototype vehicles of the research project UNICAR*agil* all four wheels are driven and steered individually. The odometry observations are taken from sensors at the steering actuator and from an encoder at the electric motor. In any case, the wheel rotation rates computed internally either in the ESC's Electronic Control Unit (ECU) or in the respective ECU in UNICAR*agil* are preferred (rather than the wheel ticks) because of their higher resolution, as explained by Reuper (2020, p. 53).

For the odometry measurement update, the three-dimensional velocity vector

$$\mathbf{v}^b_{ew} = \begin{pmatrix} v^b_{ew,F} \\ v^b_{ew,L} \\ v^b_{ew,U} \end{pmatrix} \tag{4.40}$$

is computed by the preprocessing module (odometry model) for each wheel (Reuper, 2020, p. 54). In the following, the steps to do so are explained. The component in upwards

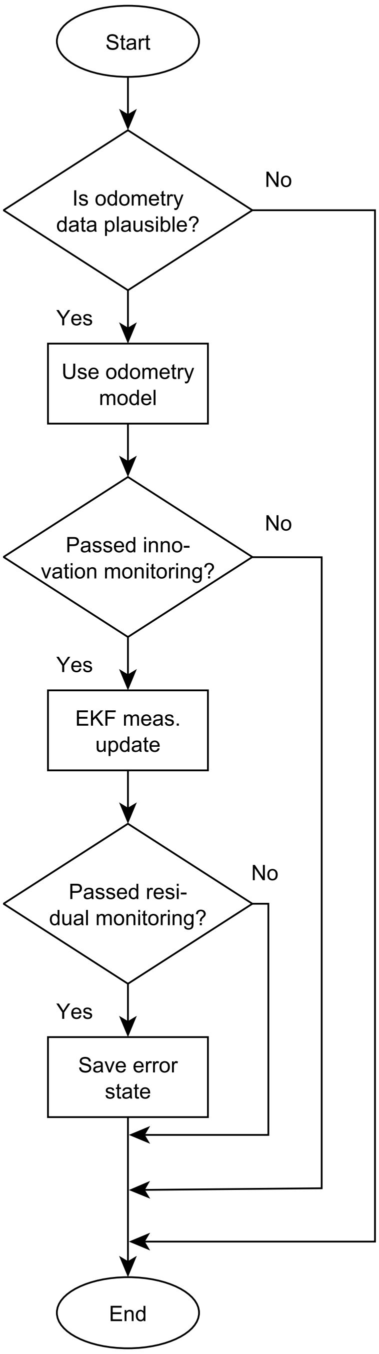

Start
Is odometry data plausible?
No
Yes
Use odometry model
Passed innovation monitoring?
No
Yes
EKF meas. update
Passed residual monitoring?
No
Yes
Save error state
End

Figure 4.5.: Flow chart of processing odometry data

direction $v^b_{ew,U}$ cannot be measured by the odometry sensors and is therefore not used in the fusion filter but included to keep the notation consistent (Reuper, 2020, p. 54). In order to compute

$$\mathbf{v}^b_{ew} = \mathbf{C}^{w\,\mathsf{T}}_b \mathbf{v}^w_{ew}, \tag{4.41}$$

the rotation matrix $\mathbf{C}^b_w$ is needed to transform the vector $\mathbf{v}^w_{ew}$ to the body frame. This rotation matrix is different for each wheel and determined based on the wheel's steering angle, which is either given directly as in UNICAR*agil* or computed based on the steering wheel angle as given by Reuper (2020, p. 54) for the used conventional vehicle. The velocity at each wheel in its frame $\mathbf{v}^w_{ew}$ is determined as given by Reuper (2020, p. 54) using the slip definition for the longitudinal tire slip correction factor κ and the lateral slip angle α explained in Section A.2 in the Appendix:

$$\mathbf{v}^w_{ew} = \begin{pmatrix} v^w_{ew,x} \\ v^w_{ew,y} \\ v^w_{ew,z} \end{pmatrix} = \begin{pmatrix} 1 \\ -\tan\hat{\alpha} \\ 0 \end{pmatrix} \omega_w \hat{r}^-_d \hat{\kappa}_x \tag{4.42}$$

with the wheel rotation rate ω_w and the a-priori estimation of the dynamic tire radius $\hat{r}^-_d$, which is modeled as random walk simplifying its propagation to

$$\hat{r}^-_{d,k} = \hat{r}^+_{d,k-1}. \tag{4.43}$$

For the use case in conventional vehicles, a linear slip estimation based on the odometry model of Reuper (2020, Section 4.2) is implemented, which is explained in Section B.8 in the Appendix. Modifications for the use case in UNICAR*agil* are addressed again in Section 8.1.

The odometry measurement vector is computed for a single wheel $\mathbf{z}_{\mathrm{odo}}$ as

$$\begin{aligned} \mathbf{z}_{\mathrm{odo}} &= \hat{\mathbf{z}}_{\mathrm{odo}} - \tilde{\mathbf{z}}_{\mathrm{odo}} \\ &= \hat{\mathbf{v}}^b_{ew} - \tilde{\mathbf{v}}^b_{ew} \\ &= \hat{\mathbf{C}}^{n\,\mathsf{T}}_b \hat{\mathbf{v}}^n_{en} + \hat{\boldsymbol{\omega}}^b_{eb} \times \mathbf{p}^b_{bw} - \mathbf{C}^{w\,\mathsf{T}}_b \begin{pmatrix} 1 \\ -\tan\hat{\alpha} \\ 0 \end{pmatrix} \omega_w \hat{r}_{\mathrm{d}} \hat{\kappa}_x, \end{aligned} \tag{4.44}$$

with the lever arm from the IMU to the respective wheel contact patch $\mathbf{p}^b_{bw}$ (Reuper, 2020, p. 94). As for the lever arm to the GNSS antenna, the lever arms to the wheel contact patches also need to be calibrated beforehand.

Correspondingly, the measurement matrix is computed as

$$\mathbf{H}_{\text{odo}} = \begin{pmatrix} \mathbf{H}_{\text{odo},\psi} & \mathbf{H}_{\text{odo,vel}} & \mathbf{0}_{3\times3} & \mathbf{H}_{\text{odo},\omega} & \mathbf{0}_{3\times3} & \mathbf{H}_{\text{odo},r_{\text{d}}} \end{pmatrix} \tag{4.45}$$

with

$$\mathbf{H}_{\text{odo},\psi} = \hat{\mathbf{C}}_b^{n\mathsf{T}} \left[\hat{\mathbf{v}}_{en}^n \times\right] \tag{4.46}$$

$$\mathbf{H}_{\text{odo,vel}} = \hat{\mathbf{C}}_b^{n\mathsf{T}} \tag{4.47}$$

$$\mathbf{H}_{\text{odo},\omega} = \left[\mathbf{p}_{bw}^b \times\right] \tag{4.48}$$

$$\mathbf{H}_{\text{odo},r_{\text{d}}} = \left(\mathbf{C}_b^{w\mathsf{T}} \begin{pmatrix} -1 \\ \tan\hat{\alpha} \\ 0 \end{pmatrix} \omega_w \hat{\kappa}_x \quad \mathbf{0}_{3\times1} \quad \mathbf{0}_{3\times1} \quad \mathbf{0}_{3\times1} \right), \tag{4.49}$$

where the Earth's rotation rate is neglected to compute the dependency of the measurement vector on the gyroscope offset error ($\hat{\boldsymbol{\omega}}_{eb}^b \approx \hat{\boldsymbol{\omega}}_{ib}^b = \tilde{\boldsymbol{\omega}}_{ib}^b - \hat{\mathbf{b}}_\omega$) and the non-zero entries of $\mathbf{H}_{\text{odo},r_{\text{d}}}$ are shifted to the column corresponding to the used wheel, as given by Reuper (2020, p. 94).

For the odometry measurement noise covariance matrix $\mathbf{R}_{\text{odo}}$, the procedure is analogous to the slip estimation, meaning the approach of Reuper (2020, Section 4.3) is taken for conventional vehicles (explained as well in Section B.8 in the Appendix), which includes reducing the measurement vector from twelve to six entries by summarizing the lateral velocity components per axle and discarding the vertical velocity components as mentioned. The summation of lateral velocities is not fitting to the use case in UNICAR*agil* where each wheel is steered individually and therefore will not be applied in this use case.

4.3.3. Zero Updates

If the fusion filter detects that the vehicle is in standstill, zero updates are performed. These aim to maintain the fusion filter's alignment and IMU calibration (Groves, 2013, p. 638). As mentioned before in Section 4.1.2, the standstill detection in this work is performed by the odometry sensors, i.e., if for more than 0.5 sec all odometry sensors observe a velocity of zero, the vehicle will be considered in standstill and zero updates can be executed (Gottschalg & Leinen, 2021). Two types of zero updates are implemented, ZVU and ZARU, which use the information that during standstill, the vehicle does not move (velocity is zero) and does not turn (angular rate is zero), respectively.

The measurement vector for the ZVU is formed as the difference between the a-priori velocity estimate $\hat{\mathbf{v}}_{en}^{n}$ and the velocity during standstill which is zero:

$$\begin{aligned} \mathbf{z}_{\mathrm{ZVU}} &= \hat{\mathbf{z}}_{\mathrm{ZVU}} - \tilde{\mathbf{z}}_{\mathrm{ZVU}} \\ &= \hat{\mathbf{v}}_{en}^{n} - \mathbf{0}_{3\times 1}. \end{aligned} \tag{4.50}$$

The corresponding measurement matrix $\mathbf{H}_{\mathrm{ZVU}}$ is implemented as given by Groves (2013, p. 639):

$$\mathbf{H}_{\mathrm{ZVU}} = \begin{pmatrix} \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 4} \end{pmatrix}. \tag{4.51}$$

Note that there is no negative sign here for $\mathbf{H}_{\mathrm{ZVU,vel}}$ since Groves' measurement vector uses an opposite sign definition. In order to perform this measurement update, a measurement noise covariance matrix $\mathbf{R}_{\mathrm{ZVU}}$ is needed which is chosen as diagonal matrix

$$\mathbf{R}_{\mathrm{ZVU}} = \mathrm{diag}\left(\sigma_{\mathrm{ZVU},E}^2,\ \sigma_{\mathrm{ZVU},N}^2,\ \sigma_{\mathrm{ZVU},U}^2\right) \tag{4.52}$$

with the corresponding components of the standard deviations in east, north and up direction $\sigma_{\mathrm{ZVU},E},\ \sigma_{\mathrm{ZVU},N},\ \sigma_{\mathrm{ZVU},U}$ which are all set to $0.1\,\mathrm{m/s}$ in this work.

For the ZARU, the measurement vector is computed as difference between the a-priori angular rate estimate $\hat{\omega}_{ib}^{b}$ and the angular rate during standstill which is assumed to be zero, neglecting the Earth's rotation (for other than MEMS-IMUs this assumption has to be reconsidered):

$$\begin{aligned} \mathbf{z}_{\mathrm{ZARU}} &= \hat{\mathbf{z}}_{\mathrm{ZARU}} - \tilde{\mathbf{z}}_{\mathrm{ZARU}} \\ &= \hat{\omega}_{ib}^{b} - \mathbf{0}_{3\times 1}. \end{aligned} \tag{4.53}$$

Groves (2013, p. 641) provides the corresponding measurement matrix $\mathbf{H}_{\mathrm{ZARU}}$

$$\mathbf{H}_{\mathrm{ZARU}} = \begin{pmatrix} \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 4} \end{pmatrix}. \tag{4.54}$$

Note that in this case the sign for $\mathbf{H}_{\mathrm{ZARU},\omega}$ is as given by Groves since Groves does not only defines the gyroscope offset but also the measurement vector with the opposite sign compared to the convention used in this work. For the measurement noise covariance matrix $\mathbf{R}_{\mathrm{ZARU}}$, a diagonal matrix analogously to $\mathbf{R}_{\mathrm{ZVU}}$ is chosen,

$$\mathbf{R}_{\mathrm{ZARU}} = \mathrm{diag}\left(\sigma_{\mathrm{ZARU},F}^2,\ \sigma_{\mathrm{ZARU},L}^2,\ \sigma_{\mathrm{ZARU},U}^2\right) \tag{4.55}$$

with the corresponding components of the standard deviations in front, left and up direction $\sigma_{\mathrm{ZVU},F},\ \sigma_{\mathrm{ZVU},L},\ \sigma_{\mathrm{ZVU},U}$ which are all set to $0.5\,°/\mathrm{s}$ in this work.

4.3.4. Outlier Detection

To detect outliers in the observations for the GNSS and odometry measurement update, an outlier detection procedure is implemented. It consists of three steps: plausibility checks, innovation monitoring and residual monitoring. In the subsequent paragraphs, these steps are explained.

First of all, the plausibility of the observations is checked. This step mainly includes verifying the value range of all received quantities. An example for the value range for the GNSS measurement update is given in Table B.2 in the Appendix. Besides that, a cold-start or reset of the sensor is detected by checking if several outputs are exactly zero at the same epoch. Additionally, application specific checks are performed, which analyze for example the frame counter of the message containing the observation.

Secondly, innovation monitoring as presented by Reuper (2020, p. 95–96) is implemented by comparing the ES-EKF's innovation $\delta\mathbf{z}^-$ with its covariance matrix $\mathbf{\Sigma}_{\delta\mathbf{z}^-}$:

$$\delta\mathbf{z}^- = \tilde{\mathbf{z}} - \mathbf{h}\left(\hat{\mathbf{x}}^-\right) \tag{4.56}$$

$$\mathbf{\Sigma}_{\delta\mathbf{z}^-} = \mathbf{H}\,\Sigma_{\delta\mathbf{x}^-}\,\mathbf{H}^\mathsf{T} + \mathbf{R}, \tag{4.57}$$

where the covariance matrix of the total state vector $\mathbf{\Sigma}_{\mathbf{x}^-}$ is replaced by the covariance matrix of the error-state vector $\mathbf{\Sigma}_{\delta\mathbf{x}^-}$ to account for the non-zero error-state $\delta\mathbf{x}^-$ during sequential measurement updates. The elements of the innovation vector $\delta\mathbf{z}^-$ are normalized by dividing them by their standard deviation $\sigma_{\delta z_j^-}$ meaning the corresponding entry of the main diagonal of $\mathbf{\Sigma}_{\delta\mathbf{z}^-}$, which leads to a standard normal distribution for this quantity if the KF assumptions hold (Reuper, 2020, p. 96):

$$\delta z_j^{-,*} = \frac{\delta z_j^-}{\sigma_{\delta z_j^-}}. \tag{4.58}$$

This normalized quantity is subsequently compared to a threshold, which is six for the GNSS position measurement update and three for the other measurement updates. The special treatment of position innovations is justified by their error distribution which is over-bounded in this way (more details about the measurement error distributions are given in Chapter 5). If any of the elements of $\delta\mathbf{z}^{-,*}$ has an absolute value greater than the threshold, the corresponding measurement will be discarded and this measurement update will not be executed.

Residual monitoring forms the third step and is implemented analogously to innovation monitoring. The measurement residual $\delta\mathbf{z}^{+}$ is compared with its covariance $\boldsymbol{\Sigma}_{\delta\mathbf{z}^{+}}$:

$$\delta\mathbf{z}^{+} = \tilde{\mathbf{z}} - \mathbf{h}\left(\hat{\mathbf{x}}^{+}\right) \tag{4.59}$$

$$\boldsymbol{\Sigma}_{\delta\mathbf{z}^{+}} = \mathbf{R} - \mathbf{H}\,\Sigma_{\mathbf{x}^{+}}\,\mathbf{H}^{\mathsf{T}}, \tag{4.60}$$

wherefore the measurement residual vector is normalized

$$\delta z_j^{+,*} = \frac{\delta z_j^{+}}{\sigma_{\delta z_j^{+}}}, \tag{4.61}$$

as given by Reuper (2020, p. 96–97). The same comparison with a threshold is performed as for the innovation monitoring. In case an outlier is detected, the result of the respective measurement update meaning the error state and its covariance will be discarded and the values from before this measurement update will be used to proceed.

Note that innovation and residual monitoring are not used in certain exceptional cases for example during the initial and transient phase after a cold start of the filter (Reuper, 2020, p. 95). A full list of these exceptions is given in Section B.9 in the Appendix.

5. Integrity Monitoring for Multi-Sensor Data Fusion Filter

This Chapter describes the implemented integrity monitoring measures. First of all, the use case is analyzed and requirements are derived in Section 5.1. Secondly, suitable integrity concepts are identified in Section 5.2, whose implementation is discussed subsequently in Sections 5.3 to 5.5. Finally, the implemented concepts are compared in Section 5.6 in order to decide which concepts to employ in the remainder of this work.

5.1. Use Case[1]

In order to chose a suitable integrity concept and design it appropriately, it is important to keep the intended use case in mind. In this work, an integrity concept for the fusion filter described in Chapter 4 is developed, which therefore leads to the same use case as for the fusion filter in a VDSE for an application in an automated vehicle. The integrity layer is implemented as separate module of the fusion filter as shown in Figure 4.1 in Section 4.1.2.

Consequently, certain general requirements for the fusion filter (mentioned in Section 4.1.1) also apply to the integrity layer. These include operation in real-time on the hardware given (R1) and compatibility with the use case's sensor setup (R2). The requirement R5 from Section 4.1.1 describes the main purpose of the integrity layer, which is providing integrity information for the fusion filter's outputs. The implementation contains an integrity layer for all outputs. However, the comparison and analysis in this work focuses on the horizontal position, velocity and yaw estimate as these are considered crucial for automated driving and collision avoidance in road traffic.

Besides that, there are requirements specific to the integrity layer in this use case, which relate to the integrity parameters defined in Section 2.4. Even though several different representations of integrity information can be found in the literature, in this work a constant AL with a PL put out by the integrity layer is preferred. This representation is

[1] This Section is based on a previous work written by the author (Gottschalg & Leinen, 2021, Section 2).

consistent with the previously (in Section 2.4) introduced Stanford Diagram, which is used for visual analysis of the results.

In order to determine the required values for the AL of the VDSE for an application in an automated vehicle, one has to clarify the definition of the PL and set the corresponding IR first. In the literature, one- and two-dimensional PLs are applied with different assumptions about the road geometry of the use case and the acceptable IR.

On the one hand, Reid et al. (2019) derive ALs for different types of vehicles and roads in the US using a one-dimensional PL and an IR of $10^{-8}/\mathrm{h}$. From the applications analyzed in this reference a passenger vehicle on US local roads is the most similar one to the use case in this work and the research project UNICAR*agil*. Reid et al. (2019) specify the ALs for this application as $0.29\,\mathrm{m}$ in lateral and longitudinal direction of the vehicle, as $1.40\,\mathrm{m}$ in vertical direction, and as $1.5\,°$ for the attitude angles roll, pitch and yaw.

On the other hand, there are requirements for automated driving reported by the European GNSS Agency (2019). These include accuracy requirements in terms of $95\,\%$-quantiles for the localization of $0.20\,\mathrm{m}$ horizontally (two-dimensional) and $2\,\mathrm{m}$ vertically as well as integrity requirements in terms of ALs of $10\,\mathrm{m}$ to $15\,\mathrm{m}$ with an IR of 10^{-7}.

As shown by these two examples, the existing integrity requirements for the VDSE in an application for automated driving are not consistent and there is no common understanding in the community so far. Moreover, the accuracy requirements are still a challenge for localization functions developed up to now for an application in automated driving, as pointed out by Reid et al. (2019).

Therefore, a compromise of the mentioned requirements is chosen as working hypothesis. It includes an AL of $0.6\,\mathrm{m}$ for the two-dimensional horizontal PE and $1.0\,°$ as AL for the yaw angle or Heading Error (HE) with an IR of 10^{-2}. Additionally, an AL of $0.6\,\mathrm{m/s}$ for the two-dimensional horizontal Velocity Error (VE) is considered. These ALs are chosen as an intermediate of the mentioned values from the literature. However, the IR is set less restrictive than the mentioned references since these values are inspired by aviation applications and at the moment not realistic for GNSS-based VDSE fusion filters in an application of automated driving in urban areas. Other researchers, for example Gupta and Gao (2021), make similar assumptions for an application in automated driving including an IR of 10^{-2}.

Since no requirement for a TTA is known for the use case of automated driving, it is assumed that integrity alarms have to be reported immediately. This means, if the PE exceeds the AL, the PL has to show this incident in the same epoch. In other words, the PL has to bound the PE in all epochs with exceptions in no more cases than specified by the IR.

5.2. Concept[2]

This Section describes the selection of integrity monitoring concepts implemented in this work. As mentioned in Section 2.4, integrity monitoring systems fulfill two tasks, namely fault detection & mitigation and solution protection (Groves, 2013, p. 701). In this work, the former task is included in the fusion filter as outlier detection described in Section 4.3.4. Therefore, the integrity monitoring concepts in this Chapter focus solely on the latter task solution protection.

In order to structure the literature review of Section 3.2, the integrity monitoring concepts are categorized. Properties of the data fusion algorithm are used to do so, since the integrity monitoring concepts are usually adapted to the data fusion algorithm. The chosen properties are the system architecture (e.g., filtering or snapshot methods) and the used sensors. In order to decide which integrity concepts from the literature review are further analyzed in this work, these grouping criteria are applied in a decision tree, which is depicted in Figure 5.1.

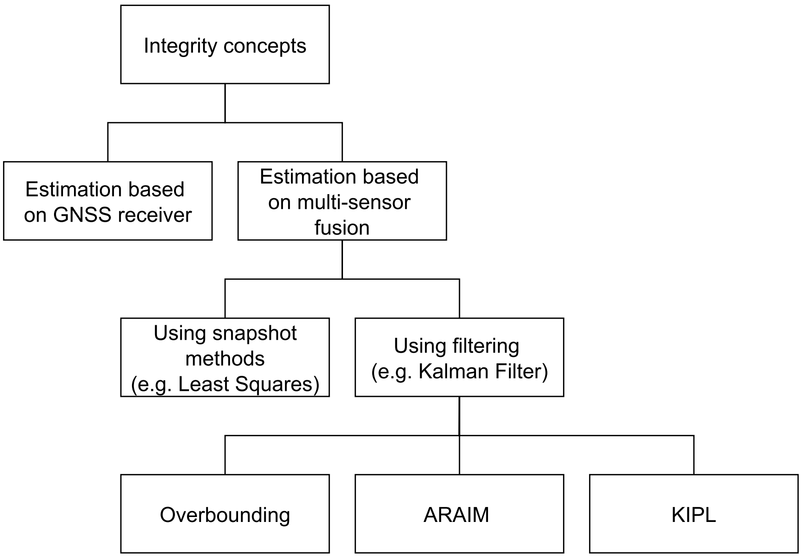

Integrity concepts
Estimation based on GNSS receiver
Estimation based on multi-sensor fusion
Using snapshot methods (e.g. Least Squares)
Using filtering (e.g. Kalman Filter)
Overbounding
ARAIM
KIPL

Figure 5.1.: Decision tree for integrity concepts (representation not comprehensive, Figure based on previous version published by the author in Gottschalg and Leinen, 2021)

[2] This Section is based on a previous work written by the author (Gottschalg & Leinen, 2021, Section 3).

Taking the use case into account, only integrity concepts for multi-sensor data fusion algorithms are considered in this work. Besides that, the concepts have to be compatible with a filtering system architecture, since the fusion filter in this work is implementing an ES-EKF.

Therefore, only three concepts remain, namely overbounding, ARAIM and KIPL. These serve as the basis for the implemented integrity algorithms, which are explained in the three subsequent Sections. Overbounding is combined with the traditional approach of using the ES-EKF's error covariance matrices in Section 5.3, while the implementations and modifications of KIPL and ARAIM are explained in Sections 5.4 and 5.5, respectively.

5.3. Traditional Approach – kSigma[3]

The first integrity algorithm is implementing the traditional approach of using the estimated error covariances from the ES-EKF to characterize the uncertainty of the fusion filter's outputs and compute confidence levels for these. Therefore, in this algorithm a Gaussian distribution (normal distribution) of the errors is assumed, which is inflated by a factor k to overbound the actual error distribution. Thus, the algorithm is referred to as kSigma in the remainder of this work.

Consequently, the equations for the PLs of the horizontal position and velocity, $PL_{\text{kSigma,pos}H}$ and $PL_{\text{kSigma,vel}H}$, as well as the yaw estimate, $PL_{\text{kSigma},\psi}$, are given by

$$PL_{\text{kSigma,pos}H} = k_{\text{pos}H}\,\sigma_{\text{pos}H} \tag{5.1}$$

$$PL_{\text{kSigma,vel}H} = k_{\text{vel}H}\,\sigma_{\text{vel}H} \tag{5.2}$$

$$PL_{\text{kSigma},\psi} = k_{\psi}\,\sigma_{\psi} \tag{5.3}$$

with

$$\sigma_{i,H} = \sqrt{\frac{\sigma_{i,E}^2 + \sigma_{i,N}^2}{2} + \sqrt{\left(\frac{\sigma_{i,E}^2 + \sigma_{i,N}^2}{2}\right)^2 + \sigma_{i,EN}^2}},\ i \in \{\text{pos}, \text{vel}\}\,, \tag{5.4}$$

where the subscripts E, N and EN stand for the east, north and the east-north component of the variance, respectively (Oliveira & Tiberius, 2009). The subscripts pos and vel refer to the position and velocity estimate. The scalar factors $k_{\text{pos}H}$ and $k_{\text{vel}H}$ are set to three, which represents an IR of approximately $0.3\,\%$ when normally distributed errors are assumed. For k_{ψ}, experiments showed that a higher factor is needed. Therefore, an empirically defined factor of nine is chosen to not exceed the specified IR.

[3] This Section is based on a previous work written by the author (Gottschalg & Leinen, 2021, Section 4.2.1).

In order to prevent over-optimistic PLs, lower bounds for the used standard deviations are introduced. A lower bound of $0.03\,\mathrm{m}$ for $\sigma_{\mathrm{pos}H}$, $0.02\,\mathrm{m/s}$ for $\sigma_{\mathrm{vel}H}$ and $0.05\,°$ for σ_ψ is applied before the PLs of the Equations (5.1)–(5.3) are computed. These values are obtained from the data set for parameter tuning described in Section 2.3.2, precisely from the part on the test track. The results represent the best possible performance of the fusion filter, which therefore serves as a lower limit for the estimation error in this algorithm.

Since the IMU was changed after the recording of the data set for parameter tuning as mentioned in Section 2.3.1, an additional empirical factor for the kSigma PLs of two is implemented. Only for the measurements with the new IMU (Analog Devices ADIS 16465-1), this factor is applied to the PLs of the Equations (5.1)–(5.3). The factor was found in initial tests with this IMU in which it showed a weaker performance than the previous one (Sensonor STIM300). This is also confirmed by the values in the data sheets, e.g., the gyroscope in-run bias variation is increased by a factor four (see Table A.1).

5.4. Kalman Integrated Protection Level

While the aforementioned explained integrity algorithm kSigma is rather simple and straightforward in its error modeling and equations, the second algorithm uses a more sophisticated approach: It includes an implementation of the previously mentioned Kalman Integrated Protection Level (KIPL) method, wherefore it is referred to as KIPL in the following (Gottschalg & Leinen, 2021).

Before the implementation and the computation of the PLs is outlined in Section 5.4.2, the used error modeling is discussed in Section 5.4.1. In Section 5.4.3, the selection of tuning parameters for this algorithm is explained.

5.4.1. Error Modeling

This Section discusses the error modeling assumptions of the KIPL integrity algorithm. In the KIPL integrity algorithm, the errors of each input to the ES-EKF are modeled as a multi-variate Student distribution, which are afterwards fused to a total error distribution to compute the PLs (Navarro Madrid et al., 2015).

The probability density function of a zero-mean multi-variate Student distribution t_{N_m} with the vector $\boldsymbol{\theta}$ as input is given as

$$f_{N_m(\mathbf{R}_m)}(\boldsymbol{\theta}) = \frac{\Gamma\left(\frac{N_m+d}{2}\right)}{\Gamma\left(\frac{N_m}{2}\right)\, N_m^{\frac{d}{2}}\, \pi^{\frac{d}{2}}\, \left(\det\left(\mathbf{R}_m\right)\right)^{\frac{1}{2}}} \left(1 + \frac{1}{N_m}\,\boldsymbol{\theta}^\mathsf{T}\,\mathbf{R}_m^{-1}\,\boldsymbol{\theta}\right)^{-\frac{N_m+d}{2}}, \tag{5.5}$$

using the degree of freedom N_m, the covariance matrix $\mathbf{R}_m$ and its dimension d (Welte, 2017, p. 30). In comparison to a Gaussian distribution, the Student distribution has heavier tails, which leads to a more realistic representation of the measurement errors and increases the robustness against outliers (Navarro Madrid, 2016).

To investigate the actual measurement error distribution, the data set for parameter tuning introduced in Section 2.3.2 is used. It contains a variety of different environments from ideal conditions on a test track to challenging GNSS reception conditions in urban areas. Representatively, the latitude component of the position and east component of velocity GNSS solution are analyzed as well as the GNSS heading solution.

Figure 5.2 depicts a histogram of the measurement errors in the GNSS latitude solution in comparison to the reference solution in the mentioned measurement. The errors are normalized using the empirical standard deviation. Only epochs with RTK reception are depicted since only these position solutions are used as input to the filter (see Section 4.3). Besides the normalized errors, also the probability density functions of a standard normal

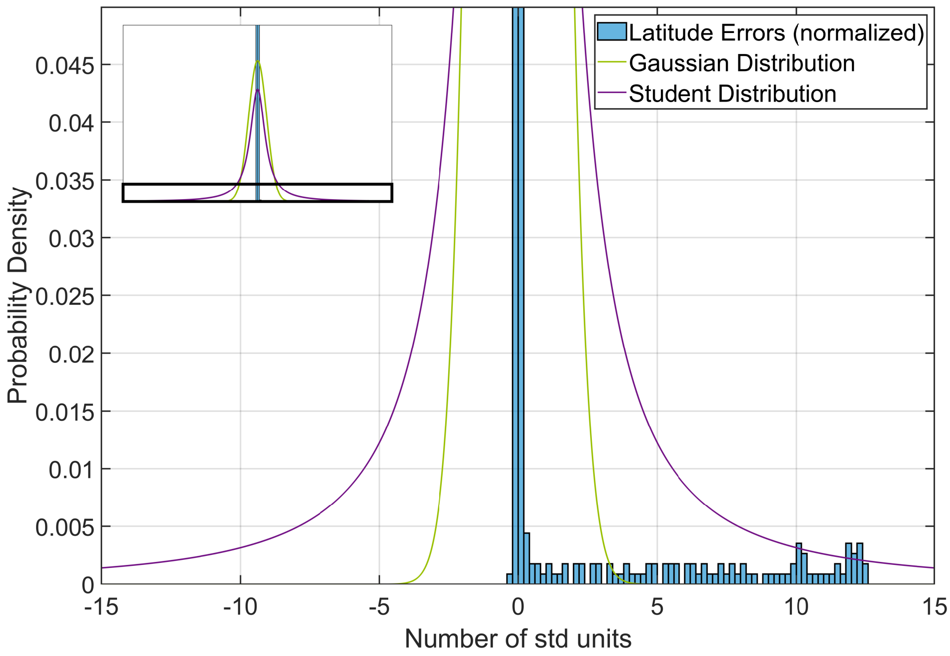


Figure 5.2.: Errors of GNSS latitude (normalized by empirical standard deviation, only RTK solutions) compared to Gaussian and Student Distribution – depicted part indicated as rectangle in overview plot in upper left corner

distribution (Gaussian, zero-mean, $\sigma = 1$) and a zero-mean Student distribution with a degree of freedom $N_m = 1$, a dimension $d = 1$ and a covariance matrix of $\mathbf{R}_m = \mathbf{I}$ are depicted in Figure 5.2. In this Figure, the ordinate limit is chosen to focus on the tails of the distributions (wherefore their peaks are not depicted here). As one can see, the outliers greater than 3σ are better represented by the Student distribution.

Analogously, the east component of the GNSS velocity and heading solution are analyzed, depicted in Figures C.1 and C.2 in the Appendix. For the velocity, all epochs are used since they are all input to the fusion filter, while for the heading solution only epochs with fixed integer ambiguities for the dual-antenna solution are used for the filter as mentioned in Section 4.3 and therefore also for the plot here. The conclusion is similar to the position errors in the previous paragraph, even though the heading errors' histogram differs from the other two by containing a higher concentration of outliers.

For the odometry observations, Reuper (2020, Section 4.3.2) conducts a detailed analysis of the measurement noise. From his evaluation and plots, it can be inferred that velocity errors greater than 3σ are rare which leads to a probability distribution without heavy tails, probably rather similar to a Gaussian. This can also be represented by the Student distribution which converges to a Gaussian distribution as the degrees of freedom increase, $N_m \to \infty$ (Tracey & Wolpert, 2018, p. 4). The same applies to the zero updates, since they rely on odometry observations to detect a standstill of the vehicle.

All in all, the assumed error distribution in the KIPL integrity algorithm seems plausible. In the subsequent Section, the implementation of this integrity algorithm is explained.

5.4.2. Implementation and Computation of Protection Levels[4]

As mentioned before, the estimation error of the ES-EKF is modeled in this algorithm as the sum of its contributions meaning the filter's measurement inputs (Navarro Madrid et al., 2015). As given by Welte (2017, p. 31), these contributions are categorized by their measurement type m, whereby each of them is modeled by a zero-mean multi-variate Student distribution $t_{N_m}(\mathbf{R}_m)$.

In the use case of this work, which is the fusion filter presented in Chapter 4, there are six measurement types, namely the GNSS position, GNSS velocity, GNSS heading angle, and odometry velocity observations, as well as the zero updates, ZVU and ZARU. This leads to six Student distributions contributing to the resulting PLs.

In the following, first the base algorithm is explained, then some empirically motivated modifications are discussed.

[4] This Subsection (including its two Subsubsections) is based on a previous work written by the author (Gottschalg & Leinen, 2021, Section 4.2.2).

Base Algorithm

All computations for this integrity algorithm are performed analogously for the position, velocity and heading or yaw PL. Differences in the parameters for the respective cases are indicated with the subscripts, pos, vel and ψ, respectively. Since the position and velocity estimation of the fusion filter are both three-dimensional and the yaw angle is one-dimensional, the parameter d follows as $d_{\text{pos}} = d_{\text{vel}} = 3$ and $d_{\psi} = 1$.

After the ES-EKF's measurement update step, meaning after Equation (4.4)–(4.6), the KIPL integrity algorithm is executed, which includes three computation steps. In the first step, two zero-mean multi-variate Student distributions $t_{N_{m1}}(\mathbf{R}_{m1})$ and $t_{N_{m2}}(\mathbf{R}_{m2})$ are computed, which represent the estimation errors of the filter in the prediction and measurement update step, respectively (Welte, 2017, p. 30). Secondly, their sum is approximated. Thirdly, the resulting PL is computed based on the updated distribution. In the following equations, the subscript m is used to indicate the correspondence to a certain measurement type for which the computations are performed analogously, unless stated otherwise.

For the first step, Equation (5.6)–(5.14) to compute N_{m_1}, $\mathbf{R}_{m_1}$, N_{m_2} and $\mathbf{R}_{m_2}$ are implemented, which are based on a patent by Navarro Madrid (2016), with the measurement matrix $\mathbf{H}_m$, the error-state covariance matrix $\mathbf{P}$, the measurement noise covariance matrix $\mathbf{R}_{\text{meas}_m}$, the transition matrix $\mathbf{\Phi}$ from the ES-EKF and the tuning parameters β and $\boldsymbol{\rho}_m$. Note that the measurement residual $\delta\mathbf{z}^+$ for the respective measurement type m is denoted here with $\mathbf{y}_m$ to increase the readability.

N_{m_1}:
$$N_{m_1} = n_m + \beta\, N'_{m_1} \tag{5.6}$$
$$n_m = n_{\text{obs}_m} - \text{tr}(\mathbf{H}_m\,\mathbf{K}_m) - \text{tr}(\boldsymbol{\rho}_m\,\mathbf{H}_m\,\mathbf{K}_m) \tag{5.7}$$
$$\mathbf{K}_m = \mathbf{P}\,\mathbf{H}_m^\mathsf{T}\,\mathbf{S}_m^{-1} \tag{5.8}$$
$$\mathbf{S}_m = \mathbf{H}_m\,\mathbf{P}\,\mathbf{H}_m^\mathsf{T} + \mathbf{R}_{\text{meas}_m} \tag{5.9}$$

$\mathbf{R}_{m_1}$:
$$\mathbf{R}_{m_1} = r_m^2\,\mathbf{K}_m\,\mathbf{S}_m\,\mathbf{K}_m^\mathsf{T} \tag{5.10}$$
$$r_m^2 = \left(\mathbf{y}_m^\mathsf{T}\,\mathbf{S}_m^{-1}\,\mathbf{y}_m + \beta\,N'_{m_1}\,r'_m\right)\,\frac{1}{N_{m_1}} \tag{5.11}$$

N_{m_2}:
$$N_{m_2} = N'_m \tag{5.12}$$

$\mathbf{R}_{m_2}$:
$$\mathbf{R}_{m_2} = \mathbf{U}\,\mathbf{R}'_m\,\mathbf{U}^\mathsf{T} \tag{5.13}$$
$$\mathbf{U} = \left(\mathbf{I} - \sum_m \mathbf{K}_m\,\mathbf{H}_m\right)\,\mathbf{\Phi} \tag{5.14}$$

In general, $\mathbf{A}'$ stands for the value of matrix $\mathbf{A}$ from the previous step, a transpose of matrix $\mathbf{A}$ is represented by $\mathbf{A}^\mathsf{T}$, $\mathbf{I}$ and $\mathbf{0}$ are the identity and zero matrix, respectively. To initialize the KIPL integrity algorithm, the values depicted in Table 5.1 are used. The parameter n_{obs_m} is determined by the number of observations in a measurement type m, leading to three for the three-dimensional GNSS position and velocity, and to one for the GNSS heading solution. For the odometry, there are six velocity observations, since only the horizontal velocity is considered together with only one lateral velocity per axle, as described in Section 4.3.2. For ZVU and ZARU, the detection of vehicle standstill is considered as one observation.

Table 5.1.: Initial values for KIPL integrity algorithm (values based on Navarro Madrid, 2016)

Parameter	N_m	N_{m_1}	$\mathbf{R}_m$	$\mathbf{A}_m$	r_m
Value	1	1	$\mathbf{0}$	$\mathbf{0}$	0

The second step contains the approximative sum of $t_{N_{m_1}}(\mathbf{R}_{m_1})$ and $t_{N_{m_2}}(\mathbf{R}_{m_2})$, which is computed by

$$\mathbf{R}_m = \mathbf{R}_{m_1} + \mathbf{R}_{m_2} + \mathbf{D}_m + \mathbf{D}_m^\mathsf{T} \tag{5.15}$$

with

$$\mathbf{D}_m = r_m \, \mathbf{K}_m \, \mathbf{S}_m^{-\frac{1}{2}} \, \mathbf{A}'_m \, \mathbf{U}' \tag{5.16}$$

and solving

$$N_m^{\frac{d-2}{2}} \, t_m^{2-d} \left(1 + t_m^{-2}\right)^{-\frac{N_m+d-2}{2}} = N_{m_1}^{\frac{d-2}{2}} \, t_{m_1}^{N_{m1}} \exp\left\{\frac{\left(N_{m_1} + N_{m_2}\right) N_{m_1}}{2N_{m_2}} \, t_{m_2}^2\right\} + N_{m_2}^{\frac{d-2}{2}} \, t_{m_2}^{N_{m2}} \exp\left\{\frac{\left(N_{m_1} + N_{m_2}\right) N_{m_2}}{2N_{m_1}} \, t_{m_1}^2\right\} \tag{5.17}$$

numerically[5] for N_m with

$$t_{m_1} = \left[\frac{N_{m_1}\,\mathrm{tr}(\mathbf{R}_{m_1})}{\mathrm{tr}(\mathbf{S})}\right]^{\frac{1}{2}}, \tag{5.18}$$

$$t_{m_2} = \left[\frac{N_{m_2}\,\mathrm{tr}(\mathbf{R}_{m_2})}{\mathrm{tr}(\mathbf{S})}\right]^{\frac{1}{2}}, \tag{5.19}$$

$$t_{m} = \left[\frac{N_{m}\,\mathrm{tr}(\mathbf{R}_{m})}{\mathrm{tr}(\mathbf{S})}\right]^{\frac{1}{2}}, \tag{5.20}$$

$$\mathbf{S} = (1+\omega)\ (N_{m_1}\mathbf{R}_{m_1} + N_{m_2}\mathbf{R}_{m_2}) \tag{5.21}$$

and the tuning parameter ω (Welte, 2017, p. 32–33). In this step, also the update of the matrix $\mathbf{A}_m$ is computed as disclosed in the patent (Navarro Madrid, 2016) by

$$\mathbf{A}_m = r_m\,\boldsymbol{\rho}_m\,\mathbf{S}_m^{\frac{1}{2}}\,\mathbf{K}'_m + \boldsymbol{\rho}_m\,\mathbf{A}'_m\,\mathbf{U}'. \tag{5.22}$$

To obtain the PL in the third step, the sum of the error bounds B_m from each measurement type m is computed as

$$PL_{\text{KIPL}} = \sum_m B_m \tag{5.23}$$

using

$$B_m = k(\alpha, N_m)\,b_m \tag{5.24}$$

with

$$b_m = \left(\frac{tr_m}{d}\right)^{\frac{1}{2}}, \tag{5.25}$$

where the symbol tr_m represents the trace of the matrix $\mathbf{R}_m$ over the states for which the PL is computed for, meaning for example the three PE states in case of the position PL, and solving

$$\frac{2}{\mathrm{B}\left(\frac{2}{d}, \frac{N_m}{2}\right)} \int_k^\infty \frac{u^{d-1}}{(1+u^2)^{(N_m+d)/2}}\,du = \alpha \tag{5.26}$$

numerically[6] for $k(\alpha, N_m)$ with the chosen integrity risk $\alpha = 0.01$ and the beta function $\mathrm{B}(x,y)$ (Navarro Madrid, 2016). Note that the PL stays constant in-between measurement updates. If an error bound B_m is not updated, it will be taken from the previous step.

[5] To solve this equation numerically in the implementation in MATLAB, the function ‘fsolve’ is used with the ‘trust-region-dogleg’ algorithm and a tolerance of 10^{-8} for function and variable changes. Details about this algorithm can be found in the MATLAB documentation (The Mathworks, Inc., 2021).

[6] Here, the same implementation as mentioned before for N_m is used.

The used set of values for the parameters ω, β and $\boldsymbol{\rho}_m$ as well as the way how they are chosen is given in Section 5.4.3.

Empirically Motivated Modifications

Motivated by the results from experiments with real-driving measurement data during the development and implementation of the integrity algorithm, an empirical factor of two is applied to the position PL from Equation (5.23). Additionally, the one-dimensional position and velocity PL is multiplied by $\sqrt{2}$ to obtain the two-dimensional horizontal PLs $PL_{\text{KIPL,pos}H}$ and $PL_{\text{KIPL,vel}H}$.

Besides that, three empirically motivated measures to prevent over-optimistic values for the PLs are introduced. The first two are a dynamic lower bound to take difficult GNSS reception conditions into account and a dynamic buffer to account for high dynamics. While the former is used for the position and the yaw PL, $PL_{\text{KIPL,pos}H}$ and $PL_{\text{KIPL},\psi}$, the latter is only applied to the position PL, since in the experiments, the influence of high dynamics is only observed on the position PL. The other PLs like the velocity PL $PL_{\text{KIPL,vel}H}$ are not modified by these measures. The third measure consists of constant offsets, which are introduced as $0.05\,\text{m/s}$ and $0.2\,°$ for the velocity and heading PL, respectively.

For the dynamic lower bound B_{low}, a quadratic function in the form

$$B_{\text{low}}(q_{\text{cnt}}) = p_2\, q_{\text{cnt}}^2 + p_1\, q_{\text{cnt}} + p_0 \tag{5.27}$$

is used with the counter q_{cnt} and the parameters p_0, p_1, p_2, which are depicted in Table 5.2. This form is chosen because the error in the position estimate with no GNSS reception grows approximately like a quadratic function with respect to time. The heading error grows approximately linear with respect to time, wherefore the parameter p_2 is set to zero for the lower bound of the heading PL.

Table 5.2.: Parameters for dynamic lower bound for KIPL integrity algorithm (Table based on Gottschalg and Leinen, 2021)

Parameter	p_2	p_1	p_0	q_{reset}
Position	$0.003\,\text{m/s}^2$	$0.075\,\text{m/s}$	$0.075\,\text{m}$	$5\,\text{s}$
Heading	0	$0.013\,\text{deg/s}$	$0.35\,\text{deg}$	$5\,\text{s}$

The dynamic lower bound is needed in two kinds of situations with respect to the GNSS reception conditions, namely in case there is no RTK-GNSS solution or if there is no GNSS solution at all available as input to the fusion filter. This leads to two dynamic lower bounds, $B_{\text{low, noGNSS}}(q_{\text{cnt, noGNSS}})$ and $B_{\text{low, noRTK}}(q_{\text{cnt, noRTK}})$, which each have a separate counter. Both counters, $q_{\text{cnt, noGNSS}}$ and $q_{\text{cnt, noRTK}}$, count time in seconds and are initialized with zero.

The counter $q_{\text{cnt, noGNSS}}$ starts to count, if there is no GNSS reception for one epoch, meaning that the GNSS receiver does not output a position solution to the fusion filter, or if the GNSS position solution is considered as an outlier (by the outlier detection described in Section 4.3.4) and is therefore not used for the measurement update. This counter continues to count until there is RTK-GNSS reception for long enough to stabilize the fusion filter's outputs, i.e., for a time longer than q_{reset} (given in Table 5.2), which leads to a reset of $q_{\text{cnt, noGNSS}}$ to zero. If $q_{\text{cnt, noGNSS}}$ is equal to zero, the counter $q_{\text{cnt, noRTK}}$ will be activated and count the time since the last RTK-GNSS solution, but will only be used if it is higher than the threshold q_{reset}. As soon as there is an epoch with RTK-GNSS reception, $q_{\text{cnt, noRTK}}$ will be reset to zero. Finally, the two lower bounds $B_{\text{low, noGNSS}}(q_{\text{cnt, noGNSS}})$ and $B_{\text{low, noRTK}}(q_{\text{cnt, noRTK}})$ are summed up before their application to $PL_{\text{KIPL,pos}H}$ and $PL_{\text{KIPL},\psi}$. The corresponding formulas are given after the following paragraph about the dynamic buffer.

The dynamic buffer consists of a scalar factor $k_{\text{posH,buffer}}$ of 0.2 and a moving average from the past five seconds of the horizontal acceleration observed by the IMU $a_{\text{filtered},H}$:

$$B_{\text{buffer,posH}} = k_{\text{posH,buffer}}\, a_{\text{filtered},H}\,. \tag{5.28}$$

Figure 5.3 depicts a flow chart of the described procedure and concludes with the following Equations

$$PL_{\text{KIPL,posH,mod}} = \begin{cases} PL_{\text{KIPL,posH}} + B_{\text{buffer,posH}} & PL_{\text{KIPL,posH}} \geq B_{\text{low,posH}} \\ B_{\text{low,posH}} + B_{\text{buffer,posH}} & PL_{\text{KIPL,posH}} < B_{\text{low,posH}} \end{cases} \tag{5.29}$$

$$PL_{\text{KIPL},\psi,\text{mod}} = \begin{cases} PL_{\text{KIPL},\psi} & PL_{\text{KIPL},\psi} \geq B_{\text{low},\psi} \\ B_{\text{low},\psi} & PL_{\text{KIPL},\psi} < B_{\text{low},\psi} \end{cases} \tag{5.30}$$

using

$$B_{\text{low,posH}} = \begin{cases} B_{\text{low,posH,noGNSS}}(q_{\text{cnt, noGNSS}}) + B_{\text{low,posH,noRTK}}(q_{\text{cnt, noRTK}}) & q_{\text{cnt,noRTK}} \geq q_{\text{reset}} \\ B_{\text{low,posH,noGNSS}}(q_{\text{cnt, noGNSS}}) + B_{\text{low,posH,noRTK}}(0) & q_{\text{cnt,noRTK}} < q_{\text{reset}} \end{cases} \tag{5.31}$$

$$B_{\text{low},\psi} = \begin{cases} B_{\text{low},\psi,\text{noGNSS}}(q_{\text{cnt, noGNSS}}) + B_{\text{low},\psi,\text{noRTK}}(q_{\text{cnt, noRTK}}) & q_{\text{cnt,noRTK}} \geq q_{\text{reset}} \\ B_{\text{low},\psi,\text{noGNSS}}(q_{\text{cnt, noGNSS}}) + B_{\text{low},\psi,\text{noRTK}}(0) & q_{\text{cnt,noRTK}} < q_{\text{reset}} \end{cases} . \tag{5.32}$$

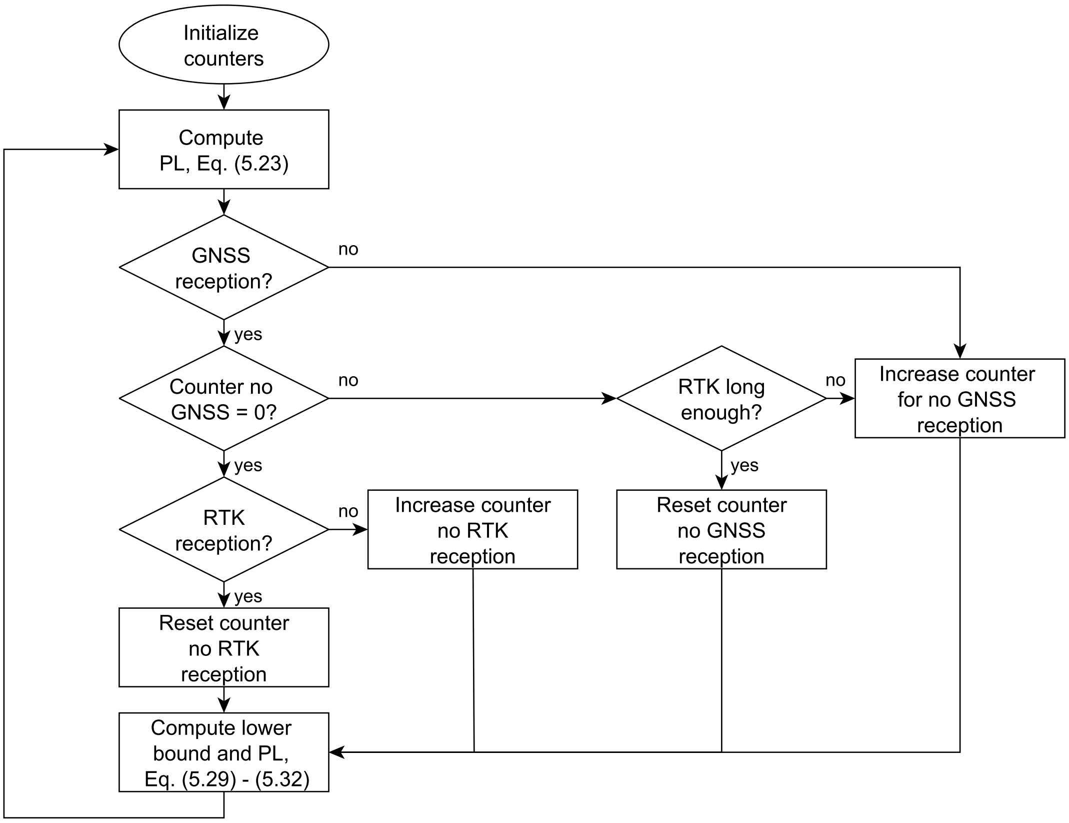

Initialize counters
Compute PL, Eq. (5.23)
GNSS reception?
no
yes
Counter no GNSS = 0?
no
yes
RTK long enough?
no
yes
Increase counter for no GNSS reception
RTK reception?
no
yes
Increase counter no RTK reception
Reset counter no GNSS reception
Reset counter no RTK reception
Compute lower bound and PL, Eq. (5.29) - (5.32)

Figure 5.3.: Flow chart of KIPL integrity algorithm PL computation (Figure based on Gottschalg and Leinen, 2021)

5.4.3. Parameter Tuning

As mentioned in the previous Section, the KIPL integrity algorithm can be adapted to the specific use case by three parameters, namely ω, β and $\boldsymbol{\rho}_m$. For the scalar tuning parameter ω, Welte (2017, p. 33) points out, that it has to be greater than one and a value of ten represents high-confidence levels (Gottschalg & Leinen, 2021). As described in the patent (Navarro Madrid, 2016), the other scalar parameter β is also an empirically found tuning parameter, whose value range is given as $0 \leq \beta < 1$ and a non-zero-value represents the influence of statistics of previous epochs on the characterization of the error noise at the present epoch (Gottschalg & Leinen, 2021). Regarding the matrix $\boldsymbol{\rho}_m$, Navarro Madrid explains in the patent (2016), that it is in most cases diagonal and its entries $\rho_{m,i}$ represent the temporal correlation of the measurement noise. Therefore, $\boldsymbol{\rho}_m$ is chosen as a diagonal matrix with the dimension n_{obs_m}.

To determine the set of values for the three mentioned parameters, an analysis with the data set for parameter tuning is carried out. This data set is described in Section 2.3.2 and contains a variety of different environments from ideal conditions on a testing ground to challenging GNSS reception conditions in urban areas.

In a first step, a suitable subset of the value range for the mentioned parameters is chosen. The subset's value range is limited by the requirement of numerical stability of the algorithm and reasonable integrity performance of the algorithm in terms of availability and integrity risk. It is found as $2 \leq \omega \leq 50$, $0.5 \leq \beta \leq (1 - 10^{-4})$ and $0.5 \leq \rho_{m,i} \leq (1 - 10^{-4})$. Values outside of this subset lead to unreliable results, since difficulties in solving the Equations (5.17) and (5.26) occurred repeatedly leading to numerical instability of the integrity algorithm. These subsets are discretized in five steps using a logarithmic scale with minor adjustments, especially for β and $\boldsymbol{\rho}_m$ to focus on areas with better integrity performance empirically found during testing and implementation. Only the position estimate is analyzed here, since the experiments showed that it is the most challenging PL. To reduce the complexity of the subsequent evaluation, one set of parameters, ω, β and $\boldsymbol{\rho}_m$, is chosen for the six measurement updates in the fusion filter of this work. Besides that, the entries of the diagonal matrix $\boldsymbol{\rho}_m$ are set to a scalar value ρ_m, meaning

$$\boldsymbol{\rho}_m = \rho_m \mathbf{I}. \tag{5.33}$$

This leads to 125 parameter combinations for which the integrity performance in terms of empirical IR and availability is computed using the mentioned data set and depicted in Figure C.3[7] in the Appendix. Even though, a subset of the value range was chosen,

[7] This Figure and the following Figure 5.4 were created with a preliminary version of the fusion filter. The afterwards applied changes are not expected to have a major effect on the results.

only 69 of the 125 possibilities led to numerically stable results. The analysis showed that the influence of the parameter ρ_m on the integrity performance is significantly stronger than the influence of ω and β. The variation of the later two influenced the availability by less than ± 5 percentage points and the empirical IR by less than ± 0.5 percentage points. Therefore, the focus is laid on ρ_m and the results are compressed by averaging the results with identical values for ρ_m (but different values for ω and β). Figure 5.4 depicts the availability and empirical IR computed in this way.

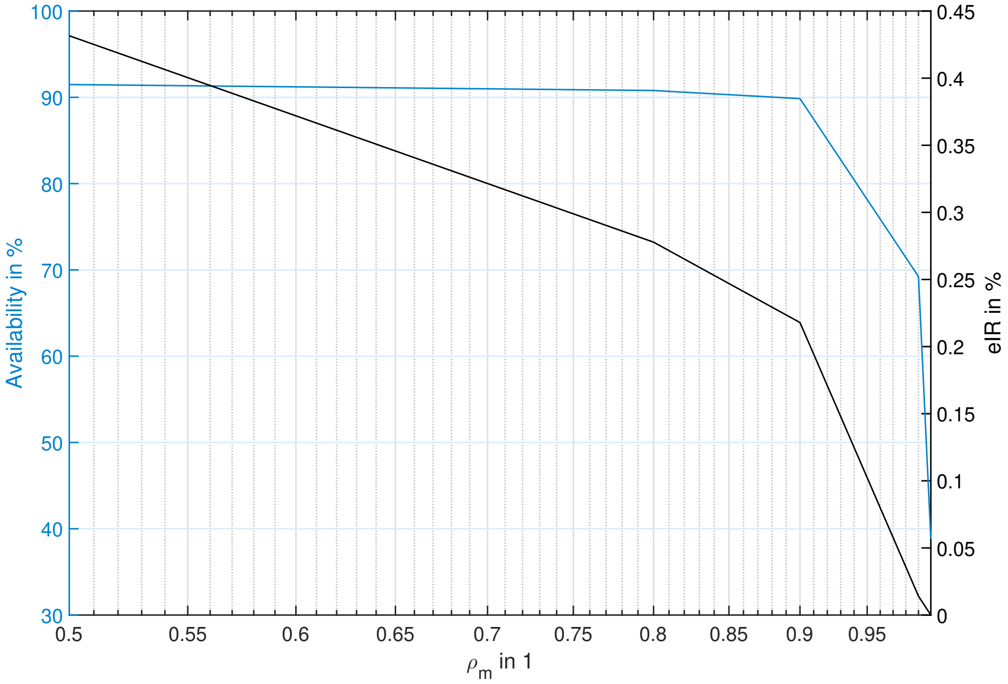


Figure 5.4.: KIPL position integrity performance in terms of availability and empirical integrity risk (eIR) depending on parameter ρ_m analyzed in data set for parameter tuning

The objective during the parameter tuning is to reach an availability as high as possible while the empirical IR is minimized, but fulfilling at least the integrity requirement of being smaller than 10^{-2}. Since the empirical IR in the Figure 5.4 is smaller than $10^{-2} = 1\,\%$ for all evaluated values of ρ_m, next the availability is analyzed. For ρ_m greater than 0.9, the availability decreases rapidly. Therefore this value is chosen as compromise between high availability and low empirical IR (Gottschalg & Leinen, 2021). For the less influential parameters ω and β, the values 10 and 0.99 are chosen (Gottschalg & Leinen, 2021). The value for ω is proposed by Welte (2017, p. 33) for high confidence levels. The parameter β has a slight influence on the numerical stability and is therefore set as a compromise between stability and performance.

5.5. Advanced Receiver Autonomous Integrity Monitoring

The third integrity algorithm includes an implementation of Advanced Receiver Autonomous Integrity Monitoring (ARAIM). First of all, the concept of this integrity algorithm and the way how it is implemented is discussed in Section 5.5.1. Secondly, the computation of PLs is explained in Section 5.5.2. Finally, the parameter tuning is described in Section 5.5.3.

5.5.1. Concept[8]

The implementation of ARAIM in this work builds upon the results of the Stanford GPS Laboratory presented by Gunning et al. (2018). In this integrity concept, errors in the measurements are characterized by a nominal error model, which describes the expected measurement errors in case no faults are present, and a threat model, which is a collection of all possible faults and their probabilities (Gunning et al., 2018). The principle of Multiple Hypothesis Solution Separation (MHSS) is used, which includes to run a bank of (sub-)filters, where each filter is fault tolerant to a fault or set of faults (Gunning et al., 2018). Based on the differences between the outputs of the filters in this bank, a PL is computed, which is explained in Section 5.5.2.

The ARAIM implementation of Gunning et al. (2018, 2019) differs from the use case of this work, since it is designed for a PPP implementation with a tight IMU coupling. This stands in contrast to the RTK-GNSS solutions used in a loose coupling in the use case of this work.

In this work, RTK-GNSS solutions are computed for ARAIM as one-out subsets, meaning solutions with all but one of the satellites in sight. Additionally, the all-in-view solution using all available satellites is computed. Since the used GNSS receiver does not allow configurations like this, the raw GNSS observations (pseudo-ranges, range rates, etc.) are exported to Receiver Independent Exchange Format (RINEX) files. The computations are performed with the open-source software RTKLIB, version demo5_b33c provided by Everett (2020), which is based on the original RTKLIB version 2.4.3 by Takasu (2019).

[8] This Subsection is based on a previous work written by the author (Gottschalg & Leinen, 2021, Section 4.2.3).

For the RTK solution, additional information in form of GNSS correction data and ephemerids is required. To keep the algorithm real-time capable (even if the implementation of ARAIM in this work is only used in post-processing), ultra rapid orbit and clock corrections (clk and sp3 files) are chosen, which are downloaded from the GeoForschungsZentrum (GFZ) Potsdam, Germany. Broadcast ephemerids are used, which are obtained from the German Federal Agency for Cartography and Geodesy (BKG).

After the GNSS solutions are computed, the fusion filter described in Chapter 4 is executed for all subsets and the all-in-view solution. As inputs, the GNSS position solution and its standard deviation computed as just described, are combined with the yaw (only for initialization) and velocity values of the real-time solution provided by the RTK-GNSS receiver as well as the odometry observations. Since the GNSS position solution differs from the one used in Chapter 4, the GNSS error model is adapted. Here, a constant factor of three is chosen as a simple empirically motivated approach.

5.5.2. Computation of Protection Levels[9]

The PLs of ARAIM are computed by the same set of formulas for all estimated quantities. The standard deviations from the fusion filter for the respective estimated quantity of the ith subset σ^i, and of the all-in-view solution σ^0, are used. Similar to the implementation of kSigma, a constant lower bound for the estimated standard deviation for the position solution of $0.015\,\mathrm{m}$ is introduced to prevent over-optimistic results.

Using the equations given by Gunning et al. (2018), the PLs of ARAIM are computed as

$$PL_{\text{ARAIM}} = \max_i \left(T_i + Q^{-1}\left(\frac{P_{\text{HMI}}}{N_{\text{S}}\,P(H_i)}\right)\sigma^i\right) \tag{5.34}$$

with the threshold

$$T_i = Q^{-1}(\alpha_i\,P_{\text{FA}})\,\sigma^i_{\text{SS}} \tag{5.35}$$

using

$$\left(\sigma^i_{\text{SS}}\right)^2 = \left(\sigma^i\right)^2 - \left(\sigma^0\right)^2, \tag{5.36}$$

the number of subsets N_{S}, the complement of the cumulative distribution function of a normal distribution Q and its inverse Q^{-1}. The chosen set of values for the parameters α_i, P_{HMI}, P_{FA} and $P(H_i)$ is given in Section 5.5.3. Finally, the horizontal PLs for the position and velocity solution are computed as

$$PL_{\text{ARAIM}} = \sqrt{PL^2_{\text{ARAIM},E} + PL^2_{\text{ARAIM},N}} \tag{5.37}$$

[9] This Subsection is based on a previous work written by the author (Gottschalg & Leinen, 2021, Section 4.2.3).

with its components in east and north direction, $PL_{\mathrm{ARAIM},E}$ and $PL_{\mathrm{ARAIM},N}$ from Equation (5.34), while the heading PL is directly given in the mentioned Equation.

In order to account for challenging GNSS reception conditions, additional empirically motivated measures are implemented. The quality level put out by RTKLIB is used as an indication for these conditions. According to the RTKLIB documentation (Takasu, 2013), the integer ambiguity for the RTK-GNSS solution is only solved properly, when a quality level of one is put out. Therefore, a factor of three for the position and heading PL, and of five for the velocity PL is used, in case a quality level unequal to one is put out. Besides that, a constant value of $0.1\,°$, $0.1\,\mathrm{m/s}$ is added to the heading, velocity PL, respectively.

5.5.3. Parameter Tuning

In order to set the three parameters of the ARAIM algorithm, the error modeling using the nominal and threat model is taken as a starting point. For the nominal error model a normal distribution of errors is assumed using standard deviations to describe the errors, computed by the fusion filter, which are inflated to overbound the actual errors. Regarding the thread model, an assumption is made by Gunning et al. (2018), that the probability of a fault present is $P(H_i) = 10^{-5}$, which is adopted for the implementation of this work (Gottschalg & Leinen, 2021). Besides that, a probability of false alert P_{FA} is used, which can be allocated to the subfilters by the parameter α_i with $\sum_i \alpha_i = 1$ (Gunning et al., 2018). Here, the assumption of Gunning et al. (2018) is adopted to set the probability of false alarm to $P_{\mathrm{FA}} = \frac{1}{3}\,10^{-6}$ and allocate this probability equally to the subfilters, meaning $\alpha_i = \frac{1}{N_{\mathrm{S}}}$ with the number of subfilters N_{S} (Gottschalg & Leinen, 2021). The factor $\frac{1}{3}$ corresponds to the three dimensions of the estimated quantities meaning the position, velocity, etc. (Gunning et al., 2018).

Eventually, also a probability of events with Hazardous Misleading Information (HMI) P_{HMI} is needed to compute the PLs. A variation of this parameter is depicted in Figure 5.5, where the data set for parameter tuning is used analogously to the KIPL parameter tuning in Section 5.4.3. A description of this measurement is given in Section 2.3.2. Empirical IR and availability increase when P_{HMI} increases. As a compromise between low empirical IR and high availability, a value of $P_{\mathrm{HMI}} = \frac{1}{3}\,10^{-7}$ is chosen, which is also used by Gunning et al. (2018). It is the highest value of P_{HMI} for which the empirical IR is lower than $2\,\%$. Therefore, it comes close to the integrity requirement of $1\,\%$, keeping in mind that the used measurement contains challenging urban environments including a tunnel. This is a design choice which might need to be changed for other use cases. Lower values of P_{HMI} lead to a even lower availability while the empirical integrity risk only decreases slightly. The Figure also shows that the integrity requirement can be hardly fulfilled in this data set. Theoretically, the value of P_{HMI} should be identical to the requested IR. This is not the

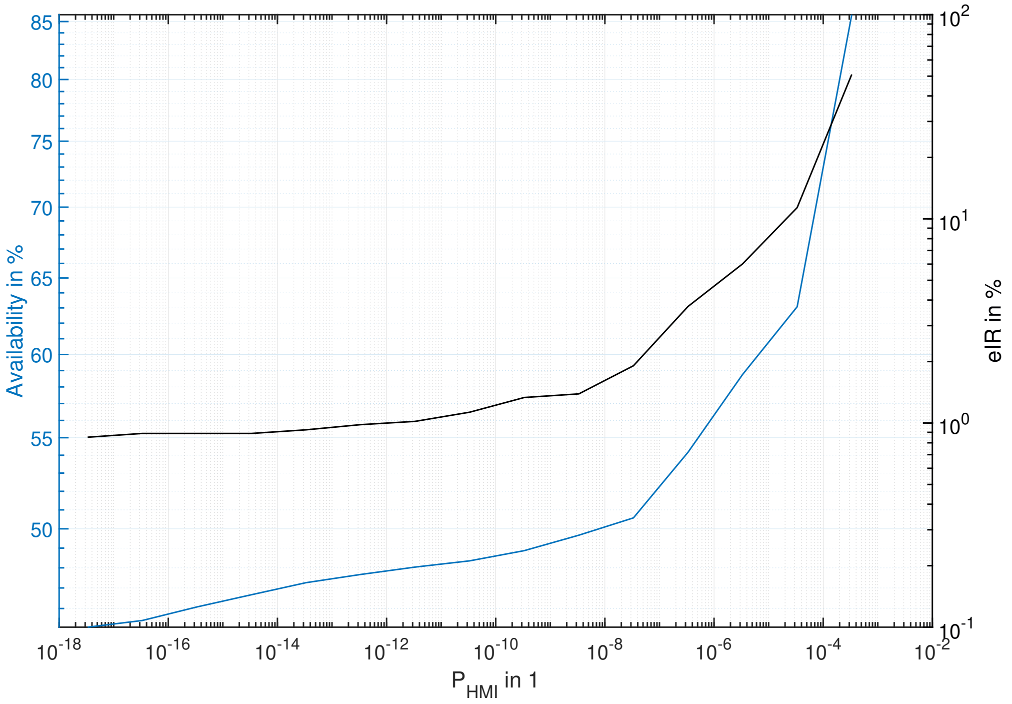


Figure 5.5.: ARAIM position integrity performance in terms of availability and empirical integrity risk (eIR) depending on parameter P_{HMI} analyzed in data set for parameter tuning

case here because of the heavy tail of the distribution of errors as shown in Section 5.4.3, which is overbound in ARAIM by a Gaussian distribution.

5.6. Comparison

In order to decide which integrity algorithms are used and further analyzed in the remainder of this work, a preliminary comparison is performed in this Section. The data set for parameter tuning described in Section 2.3.2 is also used for this preliminary comparison. Since the position PL is found to be the most challenging in comparison to the other estimated values, this analysis is restricted to the position. For the following plots, the frequency of the results is reduced to 1 Hz to facilitate the processing, i.e., the data rate is reduced to one epoch per second.

Figure 5.6 depicts Stanford Integrity Diagrams of the position PL computed with the three implemented integrity algorithms for the mentioned data set, which is divided in a part on the test track and in a part in an urban environment as described in Section 2.3.2. In the part on the test track with ideal GNSS reception conditions, the PLs bound the errors at all times and stay under the defined AL, which leads to an empirical IR of 0 % and an availability of 100 %. In the urban environment with challenging GNSS reception conditions, kSigma and KIPL fulfill the requirement of an empirical IR under 1 %. The kSigma algorithm reaches a higher availability than the KIPL algorithm (about 92 % versus 71 %) but also leads to a higher empirical IR (about 0.6 % versus 0 %). In contrast to these, the performance of ARAIM is significantly worse, with an empirical IR of more than 24 %.

The weak performance of ARAIM in the urban scenario is caused by disadvantages in its implementation. The GNSS solution is computed with RTKLIB in post-processing in comparison to the live output of the NovAtel GNSS receiver for the other two integrity algorithms, as mentioned in Section 5.5.1. Unfortunately, the performance of the GNSS solution, especially in urban environments, is significantly worse than the real-time output of the NovAtel GNSS receiver, since RTKLIB does not solve the integer ambiguities for the RTK solution reliably in this scenario. Besides that, the employed error model using standard Gaussian distributions apparently struggles in urban environments, which seems plausible since the actual error distribution is rather heavy tailed as shown in Section 5.4.1.

All in all, only kSigma and KIPL exhibit an acceptable performance in challenging environments. Therefore, only these two algorithms are used for the multi-layer fusion in Chapter 6 and further evaluated in Chapter 7.

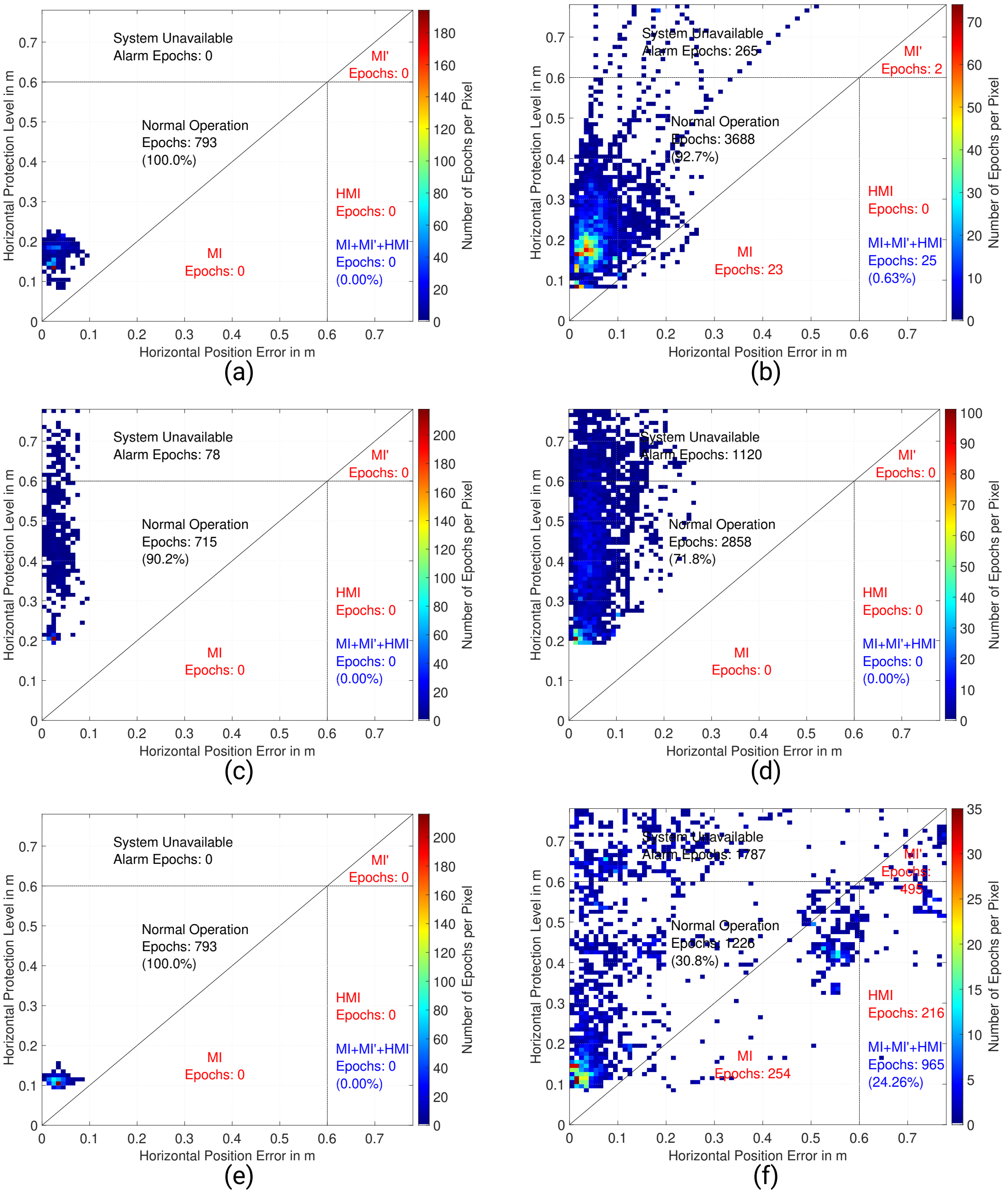


Figure 5.6.: Stanford Integrity Diagrams for protection level of horizontal position error analyzed in data set for parameter tuning in following algorithm – category combinations: (a) test track – kSigma, (b) urban – kSigma, (c) test track – KIPL, (d) urban – KIPL, (e) test track – ARAIM, (f) urban – ARAIM

6. Federated Multi-Sensor Data Fusion Architecture

In this Chapter, the multi-layer data fusion architecture for the VDSE, which was shortly introduced in Section 1.1, is explained. First of all, background information including the motivation for such a system architecture is given, together with a description of the use case in Section 6.1. Secondly, the chosen concept is discussed in Section 6.2. Thirdly, the implementation is explained in Section 6.3.

6.1. Background and Use Case

As mentioned in the introduction (Section 1.1), system architectures for automated driving are diverse, which also holds true for the VDSE. Modular system architectures are becoming more popular, splitting the tasks into separate modules or functions. Even though the VDSE is commonly implemented as a single function fusing all sensor information, the concept of modularity is also applied here. In the use case of this work, the VDSE is implemented as a fusion architecture consisting of two fusion layers. For the general case, this is depicted in a simplified representation in Figure 6.1.

The First Fusion Layer (FFL) consists of several redundant fusion filters developed by independent teams. This leads to decisive benefits, which will be discussed later on. In this work, one of these fusion filters is the one explained in Chapter 4. Each filter inputs a subset of the available sensor data, where the subsets can be chosen in multiple ways.

For the detection of erroneous sensor data, it is advantageous to systematically form subsets that include all but one sensor, which are called one-out-subsets then, as implemented for example by Jurado et al. (2020). If multiple errors should be detected at the same time, smaller subsets have to be formed respectively.

In this work, all filters use the GNSS receiver, which slightly contradicts this strategy. This is compensated by using different receiver outputs and processing methods as described later on in Section 6.2. With the use of a common reference sensor, this architecture is called a federated fusion or integration architecture. For example, Groves (2013, p. 656)

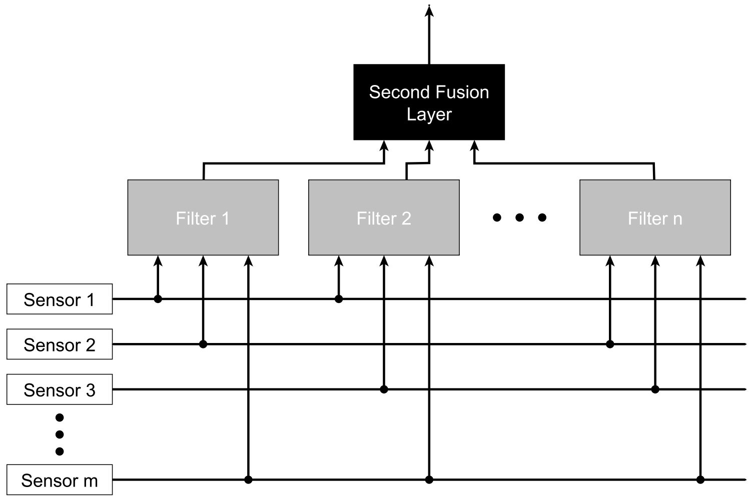


Figure 6.1.: Simplified representation of multi-layer data fusion architecture for Vehicle Dynamic State Estimation

calls a similar structure a federated no-reset integration architecture.

Besides that, each filter provides integrity information which is used by the Second Fusion Layer (SFL). In this layer, the VDSE's outputs are computed based on the estimated states of the FFL and the integrity information provided.

Such a multi-layer fusion architecture offers various benefits in comparison to the common solution of fusing all sensor observations in a single function. First of all, it increases the VDSE's robustness against outliers: In case erroneous sensor data is not detected by the implemented FDE measures in the FFL, it corrupts the output of the respective fusion filter on the FFL. Plausibility and integrity checks on the SFL increase the chances to detect these situations and mitigate the influence of erroneous sensor data on the VDSE's output.

Furthermore, the proposed multi-layer fusion architecture also offers safety benefits: If the fusion filters are implemented on different pieces of hardware, for example on different μController boards, a hardware defect in one of them will not lead to an outage of the VDSE since the SFL can continue to operate with the remaining FFL filters. The same applies to the case if a fusion filter's software crashes due to a programming error or similar reasons. Since the fusion filters are developed by independent teams, the risk of simultaneous error occurrences stemming from hidden design flaws in the FFL fusion filters is minimized (Buchholz et al., 2020). In case a piece of hardware of the SFL breaks down or a programming error or similar leads to a crash of the SFL software, the outputs of one fusion filter in the FFL are used as fall-back solution to prevent an outage of the VDSE.

Additionally, a modular structure facilitates the distribution of tasks when working in larger teams, since the FFL filters can be developed in parallel independently of each other.

Along with the mentioned benefits come certain requirements which have to be fulfilled. First of all, the general requirements for the fusion filters in the FFL discussed in Section 4.1.1 (R1–R5) have to be met. Besides that, the prerequisites for the aforementioned advantages have to be given: The FFL fusion filters and the SFL need to be implemented on different pieces of hardware. A minimum of three FFL filters is needed in order to operate this architecture properly. For the development of these filters, independent teams are needed. Last but not least, the VDSE as an entire unit also has to fulfill the general requirements (R1–R5) discussed in Section 4.1.1.

One example of an application is the research project UNICAR*agil*, in which such an approach using a fusion architecture with two layers is chosen for the VDSE as depicted in Figure 6.2. Specifically, the concept of the VDSE in UNICAR*agil* contains three FFL fusion filters which are implemented on three separate μController boards and input different

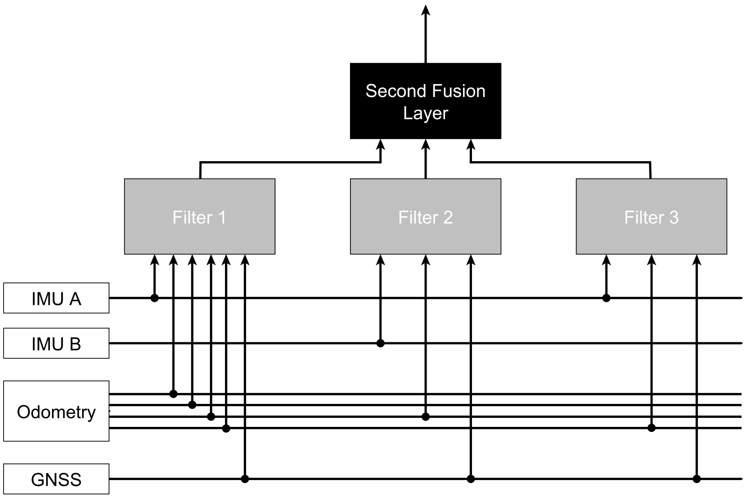


Figure 6.2.: Simplified system architecture for the Vehicle Dynamic State Estimation in UNICAR*agil* (Repetition of Figure 1.1, based on previous version published by the author in Buchholz et al., 2020)

subsets of the available sensor data. The first filter inputs observations from the first IMU, from all four odometry sensors and from the GNSS receiver. The other two filters also use the GNSS receivers' observations but only one odometry sensor together with the first or second IMU, respectively. On a fourth μController board, the SFL and additional functions for communications with other services are implemented. Further details about the actual implementation in UNICAR*agil* are given in Chapter 8.

The described use case in this work is inspired by the circumstances in the research project UNICAR*agil*. Therefore, the implementation in this work is also using a setup with three FFL fusion filters. The chosen concept for this work is described in the subsequent Section.

6.2. Concept[1]

This Section discusses the chosen concept for the SFL, whose implementation is explained in the subsequent Section. As mentioned before, the SFL inputs the estimated states from the FFL filters together with integrity information about these in form of PLs. All operations are performed separately for each state. The operation of the SFL can be summarized in three steps, which are characterized by the keywords plausibility checks, approval voting and PL-weighted averaging. The first two steps aim to increase the VDSE's robustness against outliers and the third step determines the VDSE's output.

First of all, plausibility checks are preformed in order to identify erroneous state estimates outside of the usual value range. The value range needs to be configured beforehand according to the specific use case. This step prevents gross errors to corrupt the following steps.

The second step contains an analysis of the input's integrity. According to the definition of Parhami (2005), this step implements the concept of approval voting. In general, this concept means that the participants vote for a subset of candidates who meet their criteria rather than voting for only one candidate, or in other words the participants approve a subset of candidates (Parhami, 2005). In the following, this approval of one participant given to another candidate is called support, i.e., a participant supports a certain group of candidates (Parhami, 2005). Applied to the use case of this work, the integrity of the inputs is used as criteria in the following way: An input A of the SFL supports another input B's solution if the difference between these two solutions is smaller than the respective PL provided by input A.

Using the example of the position solution, Figure 6.3 depicts this concept. In the horizontal plane (abscissa direction east, ordinate direction north) three inputs from FFL

[1] This Section is based on a previous work written by the author (Gottschalg et al., 2021, Section 3).

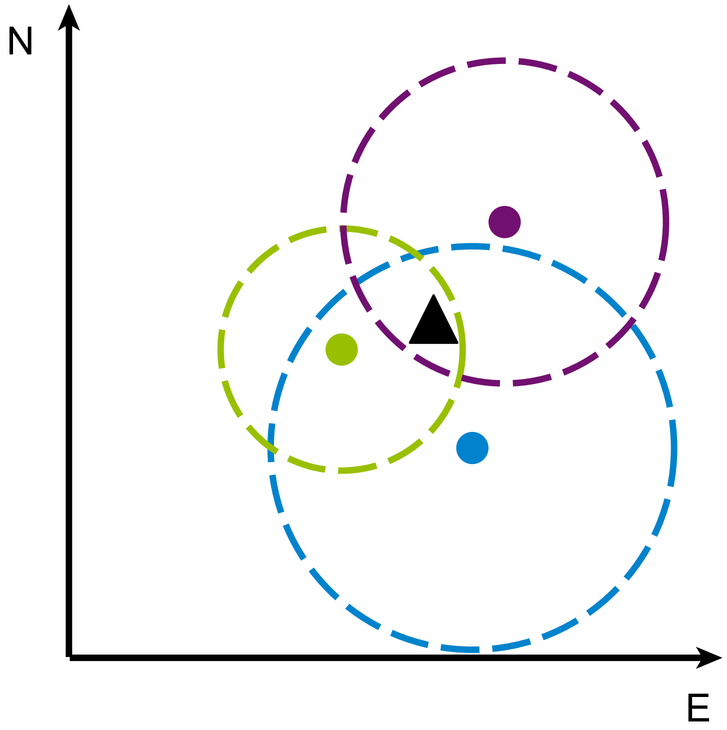


Figure 6.3.: Concept of approval voting – depicted as an example for the position solution (true position as triangle, inputs as dots and protection levels as dashed lines, chosen way of visualization inspired by Jurado et al., 2020)

filters are shown as dots in the colors blue, green and purple. Their respective PLs are drawn as dashed lines. The true position is marked by a black triangle. All PLs indicate that the true position is less than the PL away from their solution, meaning they operate correctly. The distance between the blue dot and the green dot is smaller than the blue PL, therefore the blue input supports the green one. The other two inputs do not support any other input in this example.

Depending on the specific use case, a threshold T_{min} is defined. An input needs the approval of at least T_{min} of the other inputs in order to be considered for the third step. For this work, the experiments showed that a value of $T_{\text{min}} = \frac{n_{\text{s}}-1}{2}$ with the number of SFL inputs n_{s} is appropriate to fulfill the requirements, i.e., an input needs the approval of at least half of the other inputs. For example, for the case of three FFL filters, this means an input needs the support of at least one other input in order to be considered in the following steps. In the example of Figure 6.3 this means that only the green input is used and the other two are discarded as outliers in this epoch. Summing up, the second step identifies outliers caused e.g., by design flaws or software bugs in the FFL fusion filters, or by erroneous sensor data which were not detected by the FFL filter's FDE measures.

Thirdly, the SFL's outputs are determined. Similar to the federated KF architecture proposed by Guan et al. (2012) for a multi-sensor fusion using a variation of the KF, a weighted average is implemented to compute the output of the SFL. Instead of the estimated covariance, the squared PL of the respective quantity is used to determine the weights. In this way, the uncertainties in the estimated states are represented more accurately taking their potentially heavy tailed error distributions into account. Details about how these steps are implemented are described in the subsequent Section.

All in all, the novelty of the proposed concept for the SFL lies in the use of the FFL's integrity information, namely the FFL's PLs, for the implemented combination of voting and data fusion in the SFL. The use of PLs in the second and third step distinguishes the proposed concept from state of the art multi-layer data fusion architectures such as cascaded filtered integration architectures or multi-layer neural network models. These use estimated states and covariances or trained weights, respectively, instead of PLs (Jurado et al., 2020). Additionally, these concepts generally do not include plausibility and integrity checks. By these checks in the proposed concept for the SFL and by the FDE measures in the FFL filters, the VDSE's robustness is increased, which is shown in the experimental results in Section 7.3.

6.3. Implementation[2]

This Section explains the SFL's implementation. The three Subsections discuss the FFL, the integrity algorithms implemented in this layer and the SFL, respectively.

6.3.1. First Fusion Layer

As system architecture to evaluate the performance of the proposed concept, a setup similar to the VDSE in the research project UNICAR*agil* is chosen, which is depicted in Figure 6.4. There are three multi-sensor data fusion filters, which input different subsets of the available sensor data. All fusion filters have in common that they implement a GNSS+IMU+odometry sensor data fusion in a loose coupling of GNSS and IMU.

[2] This Section (including its Subsections) is based on a previous work written by the author (Gottschalg et al., 2021, Section 3).

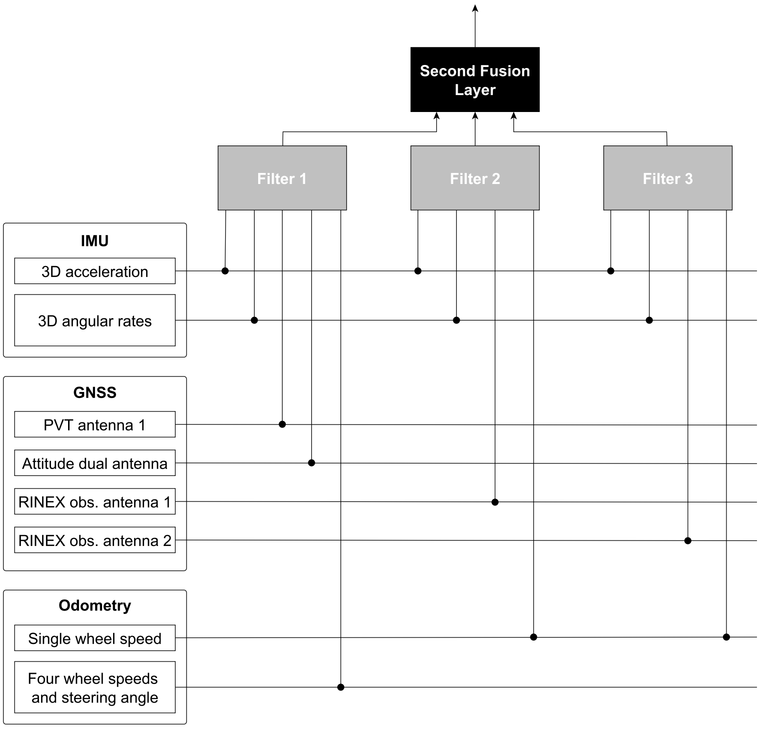


Figure 6.4.: Multi-layer system architecture of Vehicle Dynamic State Estimation including three redundant fusion filters as first fusion layer (Figure based on Gottschalg et al., 2021)

The first fusion filter (Filter 1) inputs the PVT solution as well as the attitude solution of the dual-antenna RTK-GNSS receiver (see Section 2.3.1). Additionally, odometry information of all four wheels and the steering wheel angle are used in this filter. The fusion filter developed in this work is employed as implementation for the first fusion filter (Filter 1). Its programming code is executed in MATLAB in post-processing even though the fusion filter is real-time capable and was developed for an implementation in real-time on a μController (see Chapter 4).

The remaining two fusion filters (Filter 2 and Filter 3) are implemented as post-processing in NovAtel WayPoint's Inertial Explorer (IE) 8.90, in order to implement the aforementioned principle to use fusion filters developed by independent teams. First of all, the RTK-PVT solution is computed based on the Receiver Independent Exchange Format (RINEX) observation files from the GNSS receiver. Note that the two filters use different GNSS antennas. Secondly, the loosely coupled processing as forward-filtered is executed. In this way, all three fusion filters are real-time capable. With respect to the odometry, Filter 2 and Filter 3 use each one non-steered wheel – Filter 2 rear left, Filter 3 rear right – since only a single odometer can be processed in IE 8.90.

All filters estimate the vehicle's dynamic state including the three-dimensional position, velocity, acceleration, attitude and angular rate. The performance evaluation is focusing on horizontal position, velocity and attitude (yaw angle) since these are crucial for automated driving functions such as collision avoidance. Besides the estimated states, each fusion filter provides integrity information in form of PLs. The computation of these is explained in the following Subsection.

6.3.2. Integrity Algorithms

For the algorithm in the SFL to work properly, meaningful PLs for the estimated states of the three FFL filters are needed. The integrity algorithms presented in Chapter 5 are used to fulfill this requirement.

As shown in the preliminary comparison in Section 5.6, the proposed implementation of the KIPL algorithm (see Section 5.4) exhibits the most promising results, especially in difficult environments regarding the GNSS reception conditions. This first impression will be confirmed in the extensive analysis in Chapter 7. In order to implement this algorithm, access to intermediate values of the KF, e.g., the measurement matrices, measurement residuals and transition matrix, is necessary. Since Filter 1 is implemented by the author in MATLAB, this access is given. Therefore, for Filter 1 the proposed KIPL integrity algorithm as explained in Section 5.4 is used to compute the PLs for the estimated states.

For Filter 2 and Filter 3, the mentioned access to intermediate values of the KF is not given, since there is no such option in IE 8.90. Hence, the proposed implementation of the more traditional integrity algorithm kSigma is used here. The PLs are computed as explained in Section 5.3 but with a slightly different set of parameters adapted for usage with IE 8.90. The parameters $k_{\mathrm{pos}H}$ and $k_{\mathrm{vel}H}$ are set to three, representing an IR of approximately $0.3\,\%$, assuming normally distributed errors. For k_{ψ} an empirically defined factor of fifteen is chosen to fulfill the requirements for the IR. The lower bounds are also empirically chosen as $0.15\,\mathrm{m}$ for the position PL, $0.06\,\mathrm{m/s}$ for the velocity PL and $0.25\,°$ for the heading PL.

Additionally, the experiments showed that the PLs of Filter 2 and 3 are decreasing promptly after weak GNSS reception conditions. This leads to over-optimistic results, since the quality of GNSS positioning is typically reduced immediately after reacquisition of GNSS signals. Therefore, a function is implemented to prevent the PLs of Filter 2 and 3 from decreasing rapidly in these circumstances. When in the last thirty seconds

RTK-GNSS reception was not available, then the PL's decrease is limited to one percent of it's value per second. As soon as there are thirty seconds of uninterrupted RTK-GNSS reception, this restriction is removed and the PL goes instantaneously to its original value, computed normally as described before.

6.3.3. Second Fusion Layer

Summarizing the concept for the SFL, Figure 6.5 depicts the SFL's flow chart with the implemented steps, which are explained in this Subsection. As mentioned before, the evaluation in this work is focusing on the horizontal position, velocity and the yaw angle. Thus, all steps are performed separately for these three quantities.

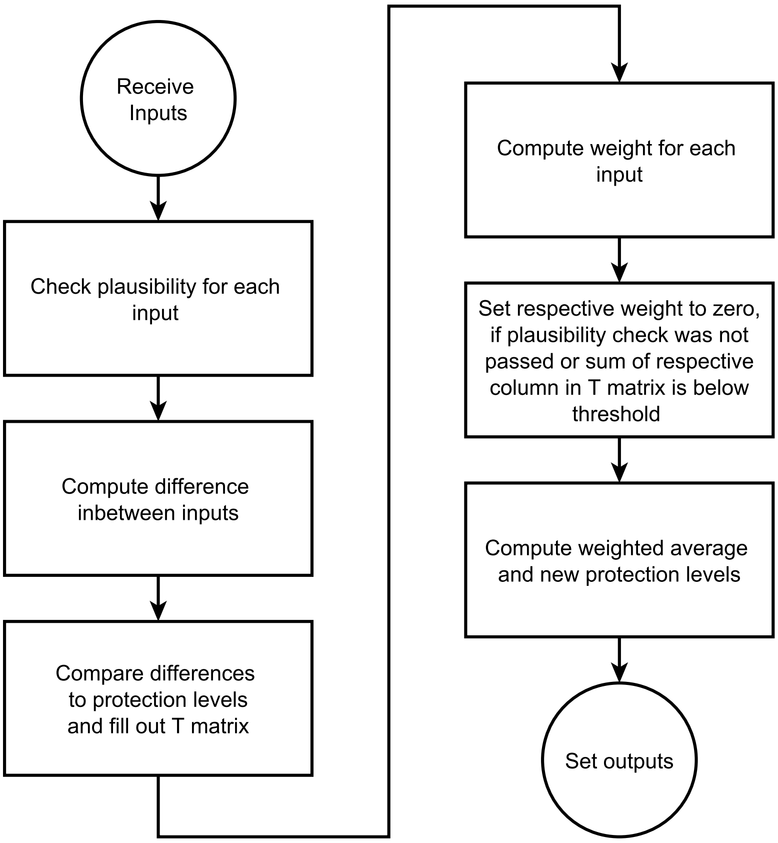

Receive Inputs
Check plausibility for each input
Compute difference inbetween inputs
Compare differences to protection levels and fill out T matrix
Compute weight for each input
Set respective weight to zero, if plausibility check was not passed or sum of respective column in T matrix is below threshold
Compute weighted average and new protection levels
Set outputs

Figure 6.5.: Flow chart of second fusion layer including plausibility and integrity checks (Figure based on Gottschalg et al., 2021)

After receiving the inputs from the FFL filters, plausibility checks are performed by verifying the value range of all inputs. According to the specific use case, the acceptable value range needs to be configured beforehand. As an example of a VDSE for automated driving, Table D.1 in the Appendix depicts a possible choice for the value ranges used in this step.

In a second step, the integrity of the inputs is analyzed. As discussed in the previous Section, approval voting is implemented here. Therefore, the differences between the inputs need to be computed. For the horizontal position and velocity this means a difference in navigation coordinates (east-north-up in m or m/s, respectively) and for the heading solution simply the one dimensional difference (in rad) is taken. As one example, the formulas for the position are given and are analogously applied for the other two quantities as well. An input A supports an input B's position solution if

$$d_{\text{pos}H,AB} < PL_{\text{pos}H,A} \tag{6.1}$$

with the horizontal position PL of input A $PL_{\text{pos}H,A}$ and the horizontal position difference $d_{\text{pos}H,AB}$ given as

$$d_{\text{pos}H,AB} = \sqrt{d^2_{\text{pos}E,AB} + d^2_{\text{pos}N,AB}}, \tag{6.2}$$

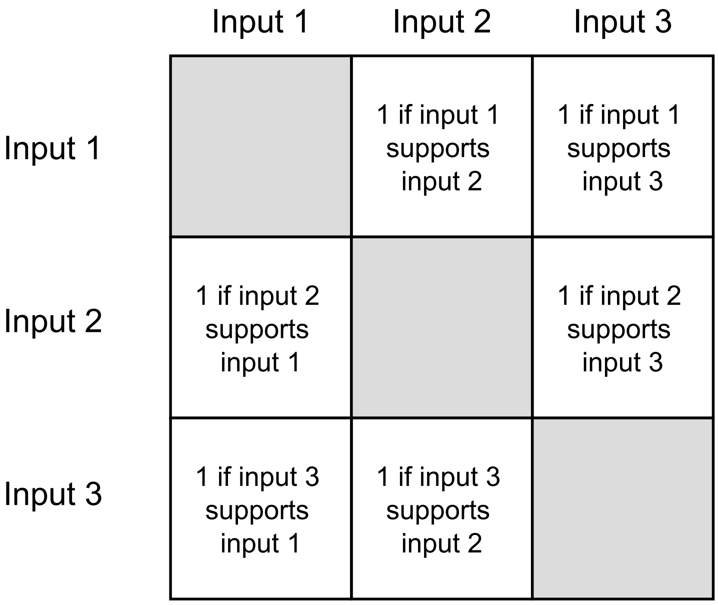
where $d_{\text{pos}E,AB}$ and $d_{\text{pos}N,AB}$ are the distance in east and north direction between the position solutions of the inputs A and B, respectively.

To store the results of the approval voting, a matrix **T** is introduced, similarly to the test statistic matrix used by Jurado et al. (2020). The rows of matrix **T** store the approval for other inputs, while the columns are associate with received support. This means that, e.g., if input two supports input three, $\mathbf{T}(2,3) = 1$ holds true. In Figure 6.6 this principle is shown as one example for the case of three inputs.

As discussed in the previous Section, an input needs the support of at least half of the other inputs, meaning in the use case of this work with three FFL filters, an input needs the support of at least one of the other inputs to be considered for the following computations. Thus, the value T_i for each input i, which describes the overall support of the input i,

$$T_i = \sum_{j=1}^{n} \mathbf{T}(j,i) \tag{6.3}$$

has to be greater or equal to the threshold $T_{\text{min}} = \frac{n_s - 1}{2} = 1$ to pass the approval voting step, with the number of inputs to the SFL $n_s = 3$. In this process, each quantity has its own matrix **T**, meaning there are three matrices $\mathbf{T}_{\text{pos}}$, $\mathbf{T}_{\text{vel}}$, $\mathbf{T}_{\psi}$, for the position, velocity and heading approval voting results, respectively.



	Input 1	Input 2	Input 3
Input 1		1 if input 1 supports input 2	1 if input 1 supports input 3
Input 2	1 if input 2 supports input 1		1 if input 2 supports input 3
Input 3	1 if input 3 supports input 1	1 if input 3 supports input 2	

Figure 6.6.: Explanation of Matrix $\mathbf{T}$ which stores approval voting results, depicted here as one example for the case of three inputs

Subsequently, the outputs of the SFL are computed. The previously (in Section 6.2) described weighted average is formed by using the PL PL_{u_i} of the inputs u_i from the FFL filters to determine the weights w_i. The SFL's output y is computed as

$$y = \sum_{i=1}^{n_\mathrm{s}} w_{u_i} u_i \tag{6.4}$$

with

$$w_{u_i} = \frac{PL_{u_i}^{-2}}{\sum_{k=1}^{n_\mathrm{s}} PL_{u_k}^{-2}}, \tag{6.5}$$

where n_s is the number of FFL filters meaning inputs to the SFL. Note that the chosen weights w_{u_i} satisfy the condition

$$\sum_{i=1}^{n_\mathrm{s}} w_{u_i} = 1. \tag{6.6}$$

If an input u_i did not pass the plausibility check or did not reach a value of $T_i \geq T_\mathrm{min}$ in the approval voting step, its corresponding weight w_{u_i} is set to zero. To derive the PL of the SFL's outputs, variance propagation is used. In general, the variance σ_z^2 of a scalar random variable $z = \sum_{i=1}^{n} a_i\, x_i$ with the scalar factor a_i and the scalar random variable x_i and its variance $\sigma_{x_i}^2$ can be computed as[3]

$$\sigma_z^2 = \sum_{i=1}^{n} a_i^2\, \sigma_{x_i}^2 + \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} \rho_{ij}\, a_i\, a_j\, \sigma_{x_i}\, \sigma_{x_j} \tag{6.7}$$

using the correlation factor

$$\rho_{ij} = \frac{\sigma_{x_i x_j}}{\sigma_{x_i}\, \sigma_{x_j}} \tag{6.8}$$

and $\sigma_{x_i x_j}$ as covariance of x_i and x_j. Together with the assumption, that PLs behave like standard deviations and therefore squared PLs can be treated like covariances, the PL of the SFL's output PL_{y_i} results as

$$PL_y = \sqrt{\sum_{i=1}^{n_\mathrm{s}} (w_{u_i}\, PL_{u_i})^2 + \sum_{i=1}^{n_\mathrm{s}} \sum_{\substack{j=1 \\ i \neq j}}^{n_\mathrm{s}} \rho\, w_{u_i}\, w_{u_j}\, PL_{u_i}\, PL_{u_j}} \tag{6.9}$$

with a constant correlation factor ρ. Here, a conservative assumption is made by setting $\rho = 1$.

To implement the SFL, all computations are performed analogously for each dimension of the three outputs in focus, horizontal position, velocity and attitude (yaw angle), replacing the generic output y in Equation (6.4) and using the respective PL in Equation (6.9).

[3] The equations for the covariance propagation are given in a similar form by Sciacchitano et al. (2016). Further details can be found in the book of Niemeier (2008, Ch. 2).

7. Results

In this Chapter the results corresponding to the implemented concepts of Chapters 4, 5 and 6 are presented. The extensive data set of in total 23 h 40 min introduced in Section 2.3.2 is used here. The sensor setup is given in Section 2.3.1, where also the reference solution is explained. As mentioned in that Section, the difference between the reference solution and the estimated solution is considered as estimation error.

As criteria for performance evaluation, the previously introduced performance criteria for navigation from Pullen (2008) – accuracy, integrity, continuity and availability – are taken as a starting point. For the accuracy, the 95 % quantile of the estimation error is used. The integrity is investigated by computing the empirical IR. Together with the availability (fraction of epochs with PL smaller AL), it is analyzed in the Stanford Integrity Diagram as introduced in Section 2.4. The continuity is inspected according to the definition given in Section 2.4, with no further requirements on the solution quality besides solution completeness, meaning the SFL provides all evaluated outputs in the analyzed epochs.

First of all, the implemented fusion filter is analyzed as single-layer fusion regarding its accuracy in Section 7.1 as well as its integrity and availability in Section 7.2. Afterwards, the multi-layer fusion is evaluated in Section 7.3 regarding all four mentioned performance criteria.

The multi-sensor data fusion implemented in this work estimates the full three dimensional dynamic state of the vehicle. However, only the horizontal position and velocity as well as the yaw estimate will be analyzed in this Chapter as explained in Section 5.1. For brevity, the three quantities are referred to as position, velocity and attitude, even though precisely only the horizontal part of them is meant.

7.1. Multi-Sensor Data Fusion Filter

The implemented multi-sensor fusion filter is evaluated in post-processing in MATLAB even though the algorithm is real-time capable as explained in Chapter 4. The results originate from an evaluation using the extensive data set and are analyzed in the four measurement categories introduced in Section 2.3.2.

Figure 7.1 depicts the Cumulative Distribution Function (CDF) of the error in the position solution computed by the implemented multi-sensor data fusion filter. The introduced measurement categories test track, highway, country road and urban are indicated by the colors blue, green, yellow and purple, respectively. The limit on the abscissa is chosen to be the AL used in the following Section to focus on the relevant area. Colors and abscissa limit are chosen for all CDF plots in this Section in this way.

For about $70\,\%$ of the epochs the performance in the four categories is rather similar with errors under 0.07 m for the first three categories and 0.09 m for the urban category. Afterwards, differences between the categories become visible. While the $95\,\%$ quantile is reached at 0.139 m for the test track category, this values increase to 0.352 m and 0.217 m for the highway and country road category, respectively. In the urban category, the $95\,\%$ quantile is outside of the plotted area at 1.007 m. While a significantly degraded performance in the urban category is expected, the results for the highway and country road category might be striking. On country roads, the position error is only slightly higher than on the test track since the GNSS reception conditions are in general favorable leading to a high GNSS accuracy. In the highway category, many bridges and overhead sign structures lead to problems with the ambiguity fixing in the used GNSS receiver, explaining the slightly degraded performance.

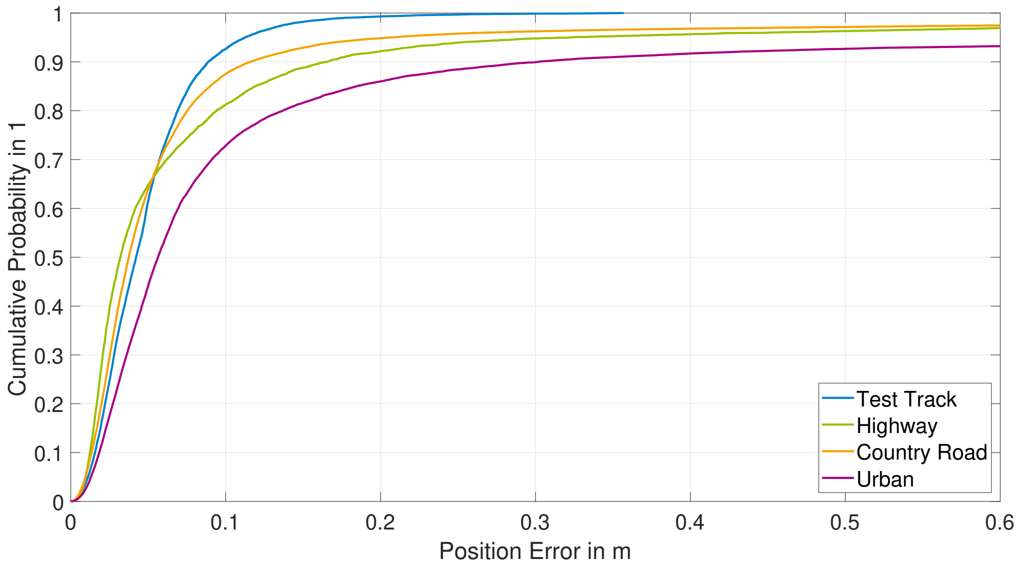


Figure 7.1.: Cumulative distribution function of horizontal position error computed by implemented multi-sensor data fusion filter in four measurement categories

For the error in the estimated velocity, the CDF is depicted in Figure 7.2. The performance is very similar in all four categories with values for the $95\,\%$ quantile between $0.037\,\mathrm{m/s}$ and $0.068\,\mathrm{m/s}$. The unfavorable GNSS reception conditions in the urban category have no noticeable influence on the velocity accuracy. The results in the country road and highway category are slightly better which can be explained by the higher driving speeds in these categories leading to a better accuracy of the odometry sensors, compared to the other two categories (test track and urban).

The estimation error for the attitude exhibits a similar performance with respect to the four measurement categories. Its CDF is plotted in Figure 7.3. The values for the $95\,\%$ quantile are between $0.24\,°$ and $0.42\,°$. Again the performance on highway and country roads is superior to the other two categories due to the higher driving speeds, where the measurement updates from the odometry sensors lead to a good heading accuracy. In the urban category, the performance is degraded by about $0.1\,°$ compared to the other categories regarding the $95\,\%$ quantile.

All in all, the performance of the multi sensor data fusion filter meets the expectations. Table 7.1 summarizes the results for the position, velocity and attitude solution in the four measurement categories. Compared to previous works with a similar sensor setup,

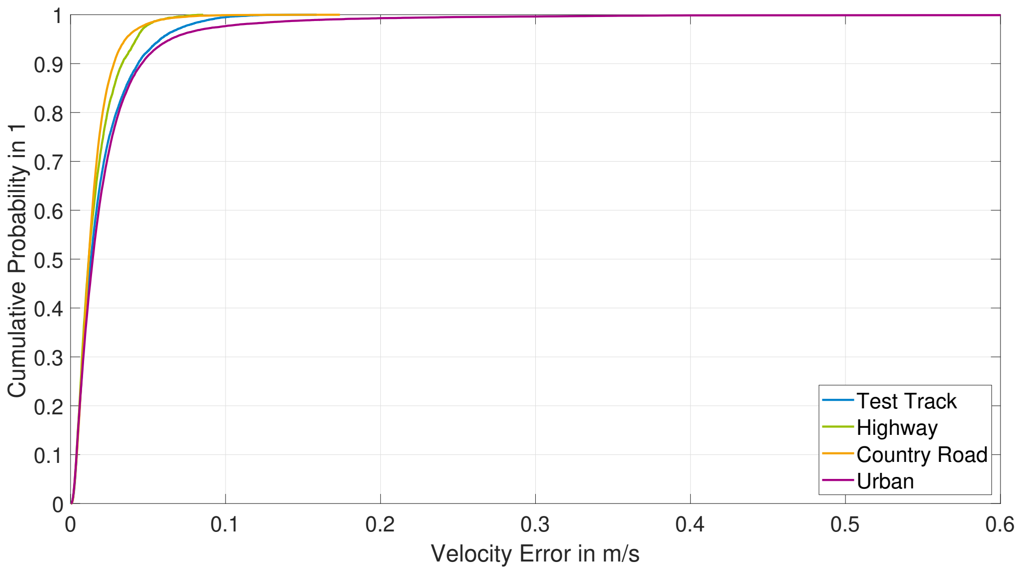


Figure 7.2.: Cumulative distribution function of horizontal velocity error computed by implemented multi-sensor data fusion filter in four measurement categories

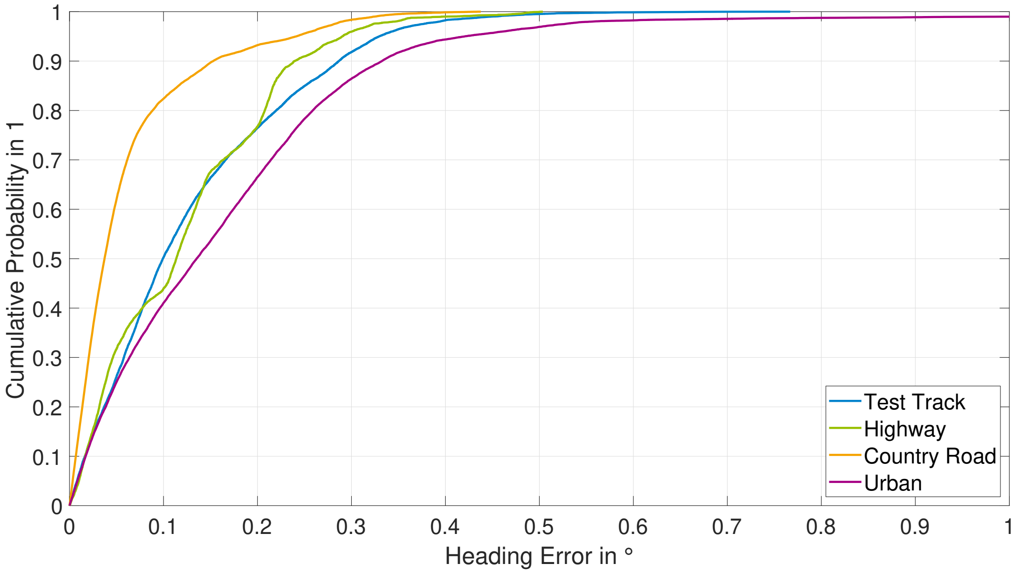


Figure 7.3.: Cumulative distribution function of horizontal attitude error computed by implemented multi-sensor data fusion filter in four measurement categories

for example the work of Reuper (2020), the accuracy is improved. First of all, the position accuracy as 95 % quantile is compared. For ideal reception conditions, the value decreased from around 0.33 m to 0.139 m. For country roads, the improvement is more significant from around 1.12 m to 0.217 m. These improvements are mainly due to the usage of RTK-GNSS, which was not used in Reuper's work. In urban environments, the improvement regarding the position error is smaller (from more than a meter to about one meter), since here RTK-GNSS is more often not available than in the other measurement categories. For the velocity and attitude error no significant improvement compared to values stated by Reuper is observable. Note that this comparison to Reuper's results is approximative since the measurement conditions are never identical even though similar trajectories have been driven. Additionally, the quoted numbers from Reuper's work relate to single driving experiments, while in this work the 95 % quantile is computed based on parts of several driving experiments of one category.

Regarding the requirements for the fusion filter defined in Section 4.1.1, the real-time requirement (R1) is fulfilled as will be shown in Section 8.2. The second requirement R2 is fulfilled since the filter is compatible with the used sensors. The fulfillment regarding accuracy and availability (R3) will be discussed further together with the requirement

Table 7.1.: Accuracy of implemented fusion filter's horizontal position, velocity and attitude solution in comparison to the reference solution using an extensive data set divided into four categories

	Test Track	Highway	Country Road	Urban
$PE_{95\,\%}$ (m)	0.139	0.352	0.217	1.007
$VE_{95\,\%}$ (m/s)	0.064	0.043	0.037	0.068
$HE_{95\,\%}$ (°)	0.33	0.29	0.24	0.42

regarding the performance in urban environments (R4) in the following Sections. The fifth requirement (R5) is analyzed in the next Subsection regarding the integrity monitoring performance.

7.2. Integrity Monitoring

In this Section the performance of the integrity monitoring algorithms implemented in this work is analyzed using the extensive data set, as introduced in Section 2.3.2 and also used above for the analysis of the fusion filter. As for the fusion filter, the results are generated in post-processing in MATLAB, even though all integrity algorithms are real-time capable.

In the comparison and intermediary conclusion in Section 5.6, two integrity monitoring algorithms were selected for the following analysis. The traditional approach based on covariances estimated by the fusion filter named kSigma and the implemented version of the more sophisticated approach, which is called KIPL and is using Student distributions for modeling the errors of the KF's outputs. The parameters are chosen as explained in Chapter 5 and the assumptions for the analysis (e.g., for the AL and IR) are as explained in the analysis in Section 5.1.

To begin, the focus is on aggregated data over the extensive data set in the four introduced measurement categories. Each measurement category is inspected using Stanford Integrity Diagrams (e.g., Figure 7.4) as explained in Section 2.4. Subsequently, typical situations in which the implemented integrity algorithms output misleading information are analyzed. Finally, the results are summarized.

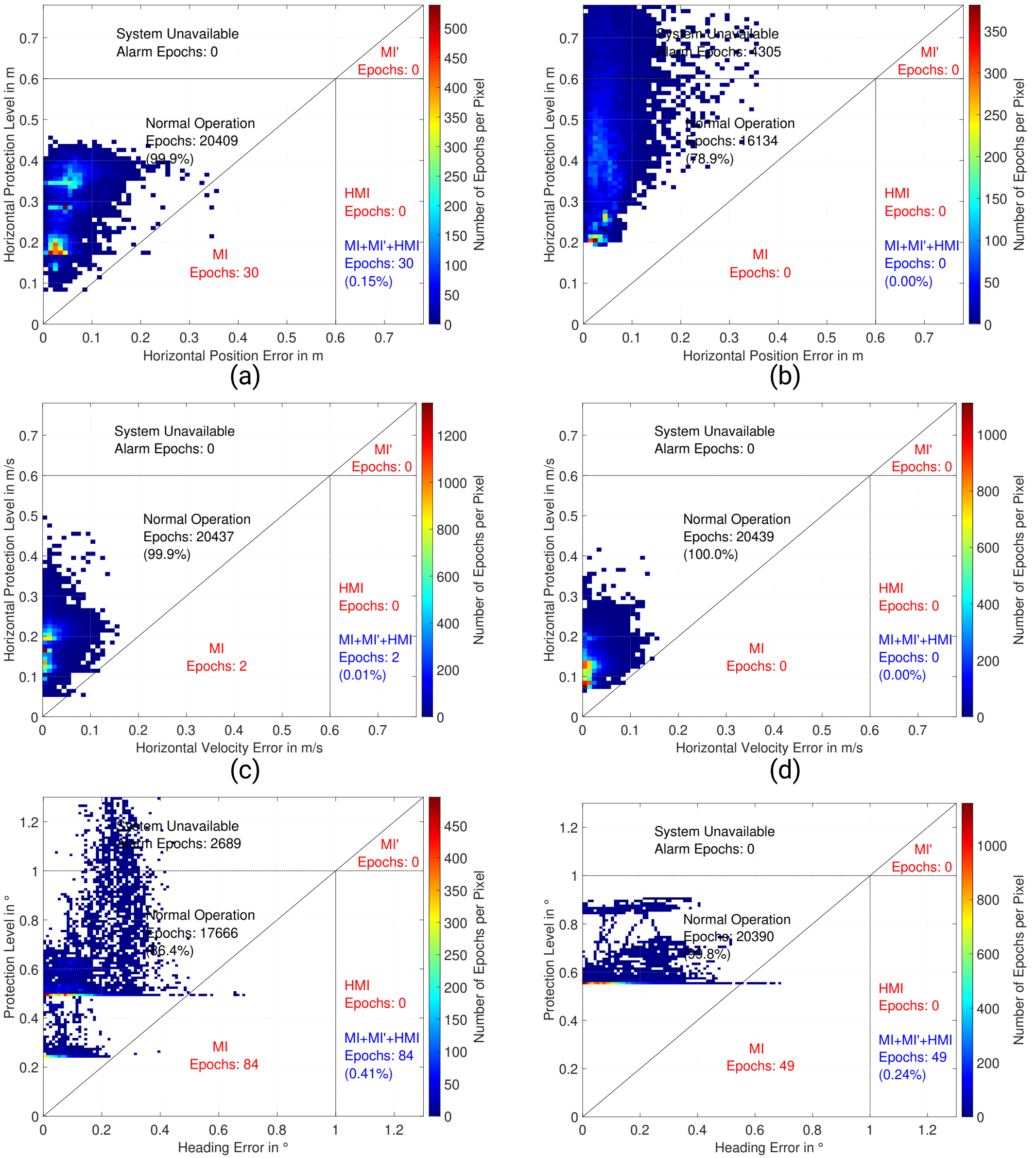


Figure 7.4.: Stanford Integrity Diagrams for the protection level of the horizontal position error (a) and (b), horizontal velocity error (c) and (d), heading error (e) and (f) in measurements in the test track category – results for kSigma on the left (a), (c), (e), KIPL on the right side (b), (d), (f)

For plotting the results in Stanford Integrity Diagrams, the frequency of the results is reduced to 1 Hz to facilitate the processing, as mentioned in Section 5.6. The Figures for the Stanford Integrity Diagrams of one measurement category consist of six diagrams and have the following structure: The left column depicts the results for the kSigma integrity algorithm, the right column for the KIPL algorithm. The first row represents the results relating to the position error and its protection level. The following rows are for the velocity and attitude, respectively.

To speed up the processing of the extensive data set, the numerical solving of equations for the KIPL integrity algorithm (Equations (5.17) and (5.26) in Section 5.4.2) is replaced by an interpolation from saved data points, which is called using a look-up-table. The inputs for the table to compute N_m in Equation (5.17) are the right side of this Equation and the term $\operatorname{tr}(\mathbf{R}_m)/\operatorname{tr}(\mathbf{S})$. Two tables are used for N_m, one for $d = 1$ and one for $d = 3$. N_m and d are the inputs for the table of k in Equation (5.26). To generate the look-up-tables, the first two minutes of the data set for parameter tuning are processed and all mentioned inputs as well as the values for the table's outputs N_m and k are saved as data points. To compute the output, the values are linearly interpolated in-between the saved data points[1]. Empirical analysis showed that this replacement only marginally effects the results (no noticeable impact on the put out PL), but significantly reduces the processing time (more than factor two on the used hardware for post-processing).

First of all, ideal GNSS reception conditions represented by the test track category are inspected. Figure 7.4 depicts the Stanford Integrity Diagrams for this category. As expected, both algorithms (kSigma and KIPL) perform very well in ideal GNSS reception conditions on the test track. The empirical IR, computed as percentage of epochs with a PE greater than the computed PL, is below the configured value of 1 % for all three analyzed quantities. However, only for the KIPL algorithm the empirical IR stays zero for the position and velocity solution. For the kSigma outputs, in certain epochs the PL is not bounding the respective error, but there are no epochs with Hazardous Misleading Information (HMI) in this category. This also holds true for the attitude, where the empirical IR is slightly higher for both algorithms than for the position and velocity outputs, staying still well below the configured IR. Regarding the availability, the velocity solution is highly available for both algorithms (kSigma – 99.9 %, KIPL – 100 %). For the position solution, this is only the case for kSigma with around 99.9 %, while the value for KIPL is with around 78.9 % significantly lower. For the attitude, the opposite is observable, the KIPL solution is highly available (100 %) and the kSigma result amounts to around 86.4 %.

Secondly, the highway category is analyzed. In Figure 7.5, the respective Stanford

[1] In the implementation in MATLAB, the function 'scatteredInterpolant' is used to perform the interpolation implementing a Delaunay triangulation of the saved data points (The Mathworks, Inc., 2022).

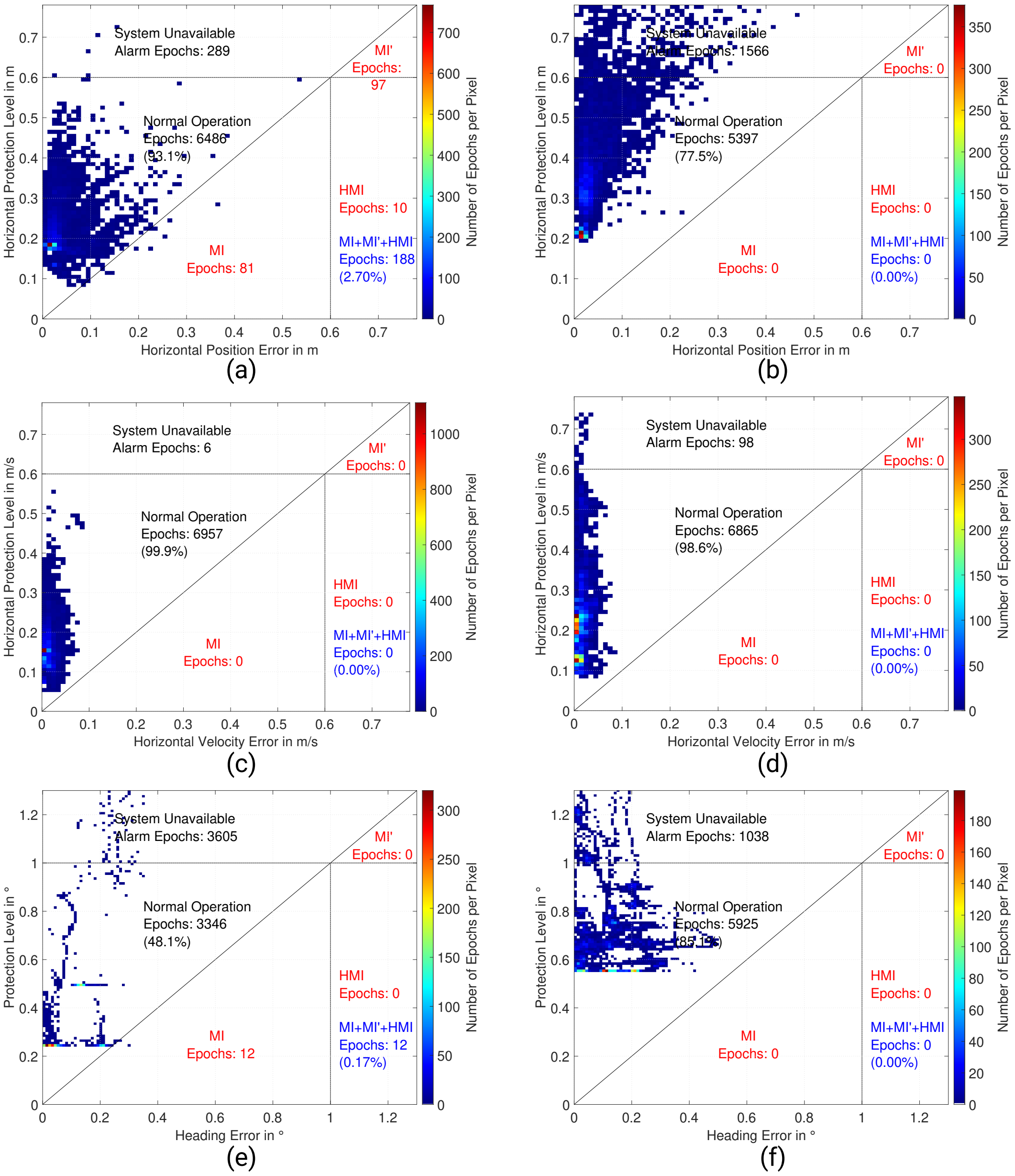
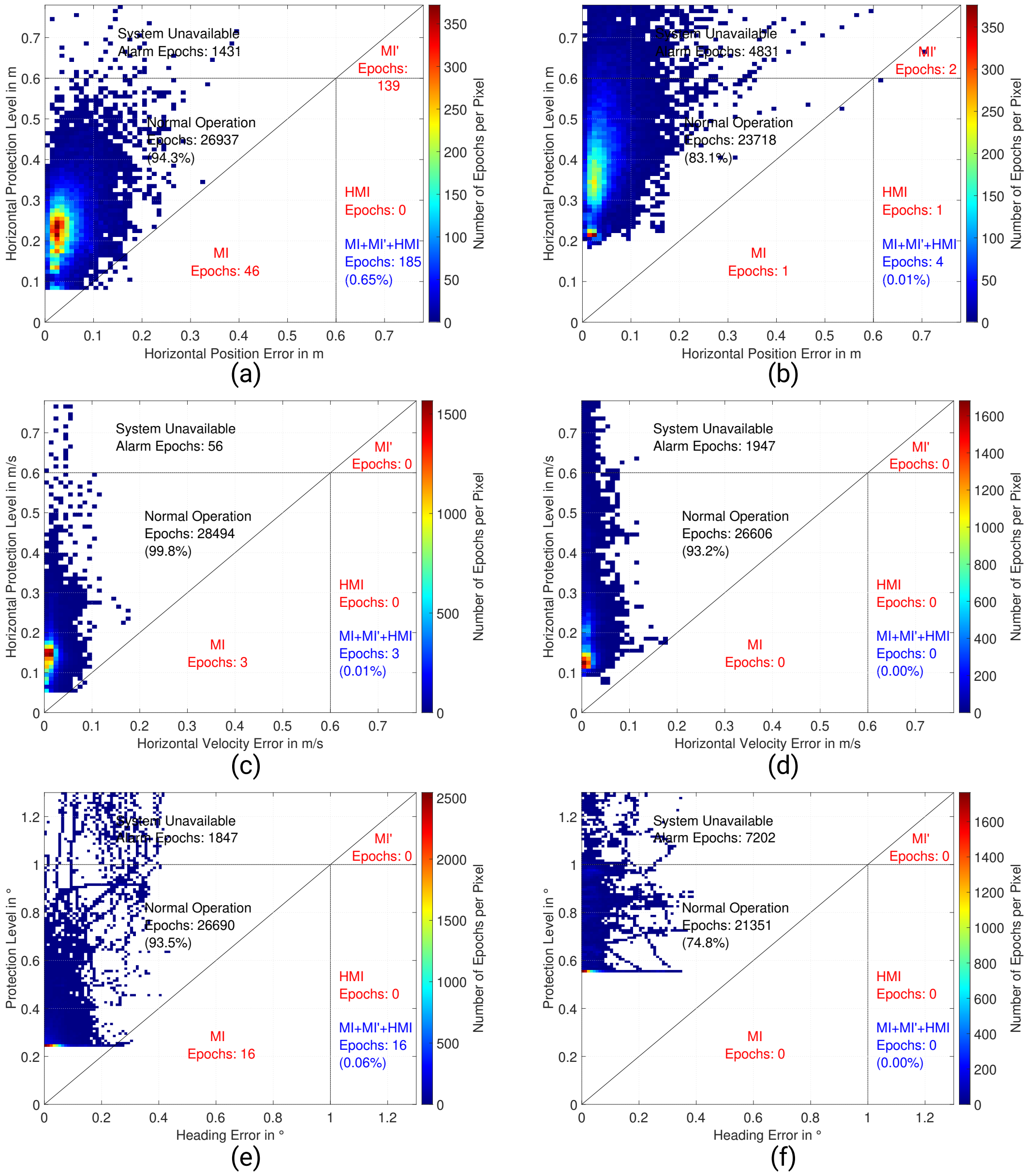


Figure 7.5.: Stanford Integrity Diagrams for the protection level of the horizontal position error (a) and (b), horizontal velocity error (c) and (d), heading error (e) and (f) in measurements in the highway category – results for kSigma on the left (a), (c), (e), KIPL on the right side (b), (d), (f)

Integrity Diagrams are depicted. Here, the GNSS reception conditions are not ideal anymore due to bridges and over-head sign structures. These degrade the GNSS receiver's – and therefore also the fusion filter's – performance as the analysis in Section 7.1 showed. This strongly affects the integrity performance of the kSigma algorithm with respect to the position solution. As seen in the mentioned Figure, the configured value for the IR cannot be met for the position solution using kSigma, since the empirical IR amounts to $2.7\,\%$. Within the epochs with $PE > PL$, there are ten epochs with HMI, which could be safety issues depending on the application. For the velocity solution this impact is not observed, the empirical IR is zero. For the attitude, the kSigma algorithm also fulfills the integrity requirement with an empirical IR of around $0.2\,\%$ and there are no epochs with HMI. In contrast to the kSigma algorithm, the KIPL algorithm is less affected by the non-ideal conditions. It can deliver a performance similar to the test track category. The empirical IR stays zero for all three quantities and there are no epochs with HMI. For both algorithms, the availability for the position and velocity solution is only slightly reduced compared to the test track category. The decrease amounts to less than five percentage points for the kSigma algorithm and less than two percentage points for the KIPL algorithm. For the attitude, this decrease in availability is more significant, especially for the kSigma algorithm with a decrease of about 38 percentage points (KIPL algorithm: 15 percentage points).

Thirdly, the country road category is analyzed using the Stanford Integrity Diagrams depicted in Figure 7.6. In this category, the GNSS reception conditions are also not ideal with a similar amount of degradations compared to the highway category. An important difference is the low amount of bridges and overhead sign-structures on country roads. This leads to a better filter performance compared to the highway category as shown in Section 7.1 and facilitates the integrity monitoring. Both algorithms perform well in the country road category. They meet the integrity requirement and reach values for the empirical IR below configured IR in all three depicted quantities. There are also no epochs with HMI, except for one epoch for the position output of KIPL which is equivalent to less than $0.004\,\%$. In general, the empirical IR of the KIPL algorithm is lower than for the kSigma algorithm in this category, amounting to $0.014\,\%$ for the position and zero for the velocity and the attitude. For the kSigma algorithm, these values are slightly higher with $0.01\,\%$ for the velocity and $0.05\,\%$ for the attitude. Only for the position output, the empirical IR is recognizably higher and amounts to $0.65\,\%$ staying still under the configured limit. Looking at the availability, the kSigma algorithm performs similarly to the highway category, except for the attitude where the availability is with $93.5\,\%$ significantly higher. For the KIPL algorithm the availability is increased for the position compared to the highway category with $83.1\,\%$ but decreased for the velocity and attitude with $93.2\,\%$ and $74.8\,\%$, respectively.



(a) (b) (c) (d) (e) (f)

Figure 7.6.: Stanford Integrity Diagrams for the protection level of the horizontal position error (a) and (b), horizontal velocity error (c) and (d), heading error (e) and (f) in measurements in the country road category – results for kSigma on the left (a), (c), (e), KIPL on the right side (b), (d), (f)

Lastly, a closer look at the results in the urban category is taken. Figure 7.7 depicts the respective Stanford Integrity Diagrams. In this category, the GNSS reception conditions are challenging due to frequent degradation of the GNSS reception. Among other unfavorbale circumstances, tunnels and multi-storage buildings on both sides of the road creating so-called urban canyons cause signal obstruction as well as multipath and NLOS reception. The filter performance for the position solution is degraded as shown in Section 7.1. This explains why in this measurement category, the kSigma algorithm is not able to fulfill the integrity requirement for the position. Its empirical IR amounts to $2.5\,\%$ which is higher than the configured limit of $1\,\%$. However, it can fulfill the requirement for the velocity and attitude with values for the empirical IR of $0.2\,\%$ and $0.6\,\%$, respectively. The KIPL algorithm performs also in this especially challenging environment of the urban category better than the kSigma algorithm with respect to the empirical IR of the position solution. The value for the KIPL algorithm amounts to $0.8\,\%$. For the other two quantities, the configured limit for the IR is kept as well. Compared to the kSigma algorithm, the value for the empirical IR for the velocity is slightly higher with $0.8\,\%$ and similar for the attitude with $0.6\,\%$. Looking at the number of epochs with HMI, the KIPL algorithm exhibits a better performance than the kSigma algorithm in all three quantities. For the position, the number of these epochs amounts to $0.2\,\%$ compared to $0.6\,\%$ for the kSigma algorithm. For the velocity this difference is less significant (KIPL: $0.05\,\%$ – kSigma: $0.08\,\%$). For the attitude both algorithms do not output any HMI in this category. When inspecting the availability, the difficulty of this measurement category is reflected in the reduced values compared to the country road category. For the position, the values amount to $83.9\,\%$ for the kSigma algorithm and $68.3\,\%$ for the KIPL algorithm. For the velocity, these values are less affected by the mentioned difficulties (kSigma: $98.8\,\%$ – KIPL: $85.6\,\%$). The attitude availability of kSigma is reduced in the urban category to $60.9\,\%$, while the KIPL algorithm shows a higher attitude availability of $75.5\,\%$.

Before summarizing the results of this Section, a brief analysis of circumstances in which the PLs do not bound the respective error is given. Since the position solution has been found to be the most challenging, this quantity is in focus here. During the evaluation of measurements used in this Section, three typical situations have been identified in which the integrity algorithms tend to output misleading information, meaning a PL below the actual error.

Unfavorable GNSS reception conditions have a strong impact on the performance of the kSigma algorithm as mentioned before. For the first situation, a part of the measurement recorded on 9 May 2019 in a forest in the south of Darmstadt, Germany is selected. In Figure E.1 in the Appendix, the PE put out by the implemented filter and the respective PLs of the kSigma and KIPL algorithm as well as the PDOP and number of used satellites put out by the used GNSS receiver are depicted. During the first part of the shown extract,

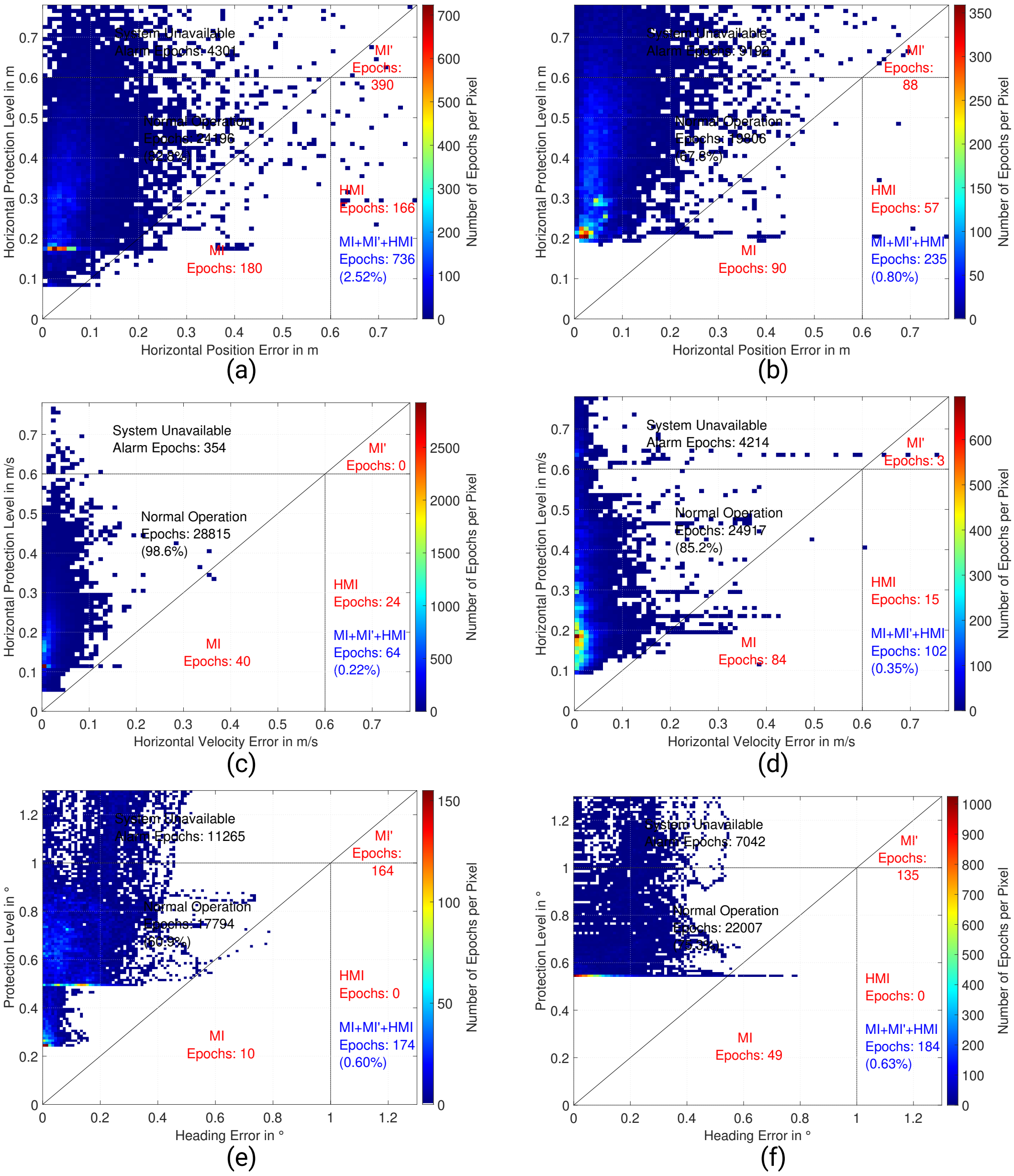


Figure 7.7.: Stanford Integrity Diagrams for the protection level of the horizontal position error (a) and (b), horizontal velocity error (c) and (d), heading error (e) and (f) in measurements in the urban category – results for kSigma on the left (a), (c), (e), KIPL on the right side (b), (d), (f)

PDOP and number of satellites are volatile indicating quickly changing GNSS reception conditions, which is due to the obstruction by vegetation on this road. This leads to a raise in both PLs. When the reception conditions worsen after about 398 800 s of the respective GPS week, meaning less satellites are used and the PDOP increases, the PE increases as well. While the KIPL algorithm bounds this increased error by its PL, the kSigma algorithm underestimates the error which leads to misleading information put out by the kSigma algorithm.

The second situation to be mentioned here includes a total obstruction of GNSS in a tunnel in the city center of Darmstadt, Germany in a measurement recorded on 26 October 2020. Figure E.2 in the Appendix depicts the respective values for PE, PLs, PDOP and number of used satellites. At around 142 720 s of the respective GPS week, no GNSS solution is available since no signals from satellites reach the GNSS receiver in the tunnel. Obviously, this impacts the position solution of the fusion filter and leads to an increasing error. In the beginning, the kSigma algorithm bounds this error by its PL. Towards the end of the tunnel, around the time when a GNSS solution becomes available again, the kSigma algorithm underestimates the error which leads to a PL smaller than the PE meaning to an output of misleading information. The KIPL algorithm outputs a significantly higher PL wherefore the error is bound by its PL in this situation.

Another challenge for the integrity algorithms are wrongly fixed integer ambiguities in the GNSS receiver. These represent the third typical situation in which misleading information tends to be put out. Figure E.3 in the Appendix depicts the respective values for this situation, extracted from a measurement in the north of Darmstadt, Germany recorded on 9 May 2019. The respective street is a tree avenue, which leads to unfavorable GNSS reception conditions. Due to the obstruction created by the vegetation starting around 400 840 s of the respective GPS week, both PLs increase. When the reception gets better, the kSigma algorithm falls back quickly on a lower PL. At around 400 875 s of the respective GPS week, it seems like the GNSS receiver wrongly fixed the integer ambiguities to determine the RTK-GNSS solution, maybe due to a cycle slip. The GNSS receiver does not output the ambiguities or further information about the RTK solution, so an explanation is not possible at this point. This leads to a jump in the position error by about one meter which is not prevented by the outlier detection and also not corrected in the directly following epochs due to difficult GNSS reception conditions. Both integrity algorithms do not detect this increased error and output misleading information for several epochs.

These three situations emphasize that the GNSS reception conditions have a strong and dominant impact on the performance of the integrity monitoring algorithms. Potential for improvement is found for example in the described situations and also in the aim for a higher availability. The mentioned points need to be further investigated in future work and are taken up again in the outlook at the end of this work (see Section 9.2).

To summarize the results of this Section regarding the integrity monitoring performance of the kSigma and KIPL algorithm in the four measurement categories, Table 7.2 depicts their values for the empirical IR and availability. All in all, the KIPL algorithm fulfills the integrity requirement of an empirical IR under the configured limit of 1 % in all four measurement categories and in all three inspected quantities. For the kSigma algorithm, this also holds true with exceptions for the position solution in the highway and urban category, which can be explained by the degraded filter performance regarding the position solution in these measurement categories. In these two categories, the different error modeling in the KIPL algorithm leads to a better performance and therefore a lower IR, fulfilling the integrity requirement. On the other hand, the KIPL algorithm outputs generally higher PLs than the kSigma algorithm (with the chosen parameter set) and can therefore be described as more conservative. This leads to a lower availability than for the kSigma algorithm. The attitude availability represents an exception to the general statements in the previous sentences. Here, overall the KIPL algorithm has a higher

Table 7.2.: Results of the integrity monitoring algorithms with respect to the empirical integrity risk (IR_e) and availability (Av) in the four described categories

		Position		**Velocity**		**Heading**	
		IR_e (%)	Av (%)	IR_e (%)	Av (%)	IR_e (%)	Av (%)
Test Track	kSigma	0.15	100.00	0.01	100.00	0.41	86.84
	KIPL	0.00	78.94	0.00	100.00	0.24	100.00
High-way	kSigma	2.70	94.46	0.00	99.91	0.17	48.23
	KIPL	0.00	77.51	0.00	98.59	0.00	85.09
Country Road	kSigma	0.65	94.50	0.01	99.80	0.06	93.53
	KIPL	0.01	83.07	0.00	93.18	0.00	74.78
Urban	kSigma	2.52	83.95	0.22	98.79	0.60	60.90
	KIPL	0.80	68.26	0.35	85.57	0.63	75.45

availability, still keeping a lower or about equal empirical IR.

7.3. Federated Multi-Sensor Data Fusion

This Section deals with the analysis of the multi-layer data fusion architecture's performance, which is abbreviated as multi-layer fusion in the following. The implementation is given in Section 6.3: Filter 1 is the fusion filter developed in this work running in MATLAB, Filter 2 and 3 are evaluated in IE 8.90. The implementations of the integrity algorithms are described in Section 6.3.2: For Filter 1 the KIPL algorithm is used, Filter 2 and 3 use the kSigma algorithm. The results are generated in post-processing using MATLAB and IE 8.90, even though all used algorithms are real-time capable.

As before, the extensive data set introduced in Section 2.3.2 is used and all four mentioned performance criteria (accuracy, integrity, continuity and availability) are inspected. Unfortunately, a change in the configuration of the GNSS receiver implemented in October 2020 reduces the output of the second antenna to a dual-antenna solution computed in the GNSS receiver. The recording of RINEX observation files from the second antenna was not possible after this date. Therefore, Filter 3 has to use the observations from the first antenna only for the driving experiments after the mentioned date.

To analyze the accuracy, the $95\,\%$ quantile for the position, velocity and attitude error is inspected. Table 7.3 depicts the values for the three fusion filters in the first layer (F1–F3) as well as for the output of the second layer. Note that the data processed in MATLAB and IE 8.90 needed to be aligned (cut-off of non-overlapping epochs and interpolation) for processing in the second fusion layer. Therefore, the results differ slightly from the previously quoted results for the FFL alone in Section 7.1 and 7.2.

Looking at the values for the position error in the four measurement categories, the multi-layer fusion shows an improved performance compared to its inputs, the three fusion filters in the first layer. Not only in favorable conditions, also in the urban measurement category, the $95\,\%$ quantile of the position error is lower for the output of the SFL than for any of the FFL filters. For the velocity solution, this also holds true except for the test track scenario where the SFL's performance is similar to the one of Filter 2 and 3 in the FFL. For the attitude solution, the accuracy is also improved in three out of four measurement categories by using the SFL over only FFL filters. Only in the country road category, no improvement is reached. Overall, the SFL performs significantly better in terms of accuracy. While there are no exceptions for the position solution, in one out of four measurement categories, this does not hold true for the velocity and attitude solution.

Besides that, Table 7.3 also depicts the empirical IR and the availability (meaning the percentage of epochs where the put out PL is lower than the specified AL) for the three

Table 7.3.: Results of the first fusion layer (F1–F3) and second fusion layer (2FL) with respect to the accuracy as $95\,\%$ quantile, empirical integrity risk (IR_e) and availability (Av) in the four described categories

		Position			**Velocity**			**Heading**		
		$PE_{95\,\%}$ (m)	IR_e (%)	Av (%)	$VE_{95\,\%}$ (m/s)	IR_e (%)	Av (%)	$HE_{95\,\%}$ (°)	IR_e (%)	Av (%)
Test Track	F1	0.139	0.00	77.67	0.064	0.01	99.03	0.33	0.31	100.00
	F2	0.081	0.00	99.88	0.029	0.00	99.34	0.66	1.25	62.14
	F3	0.088	0.00	99.88	0.029	0.00	99.36	0.66	1.40	62.11
	2FL	0.078	0.00	99.98	0.032	0.00	99.80	0.30	0.67	100.00
High-way	F1	0.352	0.00	76.67	0.043	0.00	78.83	0.29	0.00	98.79
	F2	0.091	0.00	90.28	0.026	0.00	99.22	0.45	0.00	57.55
	F3	0.104	0.01	91.73	0.026	0.00	99.67	0.45	0.00	57.88
	2FL	0.078	0.00	93.92	0.023	0.00	99.78	0.24	0.00	97.86
Country Road	F1	0.217	0.01	82.41	0.037	0.01	84.67	0.24	0.00	89.58
	F2	0.497	0.37	74.54	0.049	0.06	96.33	0.33	0.09	95.34
	F3	0.668	0.93	69.78	0.054	0.11	95.77	0.34	0.11	95.10
	2FL	0.204	0.07	79.80	0.031	0.00	98.13	0.27	0.07	99.73
Urban	F1	1.007	0.85	67.27	0.068	0.40	62.19	0.42	0.70	84.98
	F2	1.052	0.79	64.69	0.081	0.18	92.70	0.51	0.02	60.05
	F3	1.169	1.17	58.00	0.086	0.18	92.36	0.51	0.01	60.04
	2FL	0.785	1.10	70.94	0.056	0.09	95.27	0.31	0.06	92.16

quantities of interest. To go into further detail for these two performance quantities, Stanford Integrity Diagrams are used. For each of the three outputs (position, velocity and attitude), one Figure is included in the following. Each of these Figures, e.g., Figure 7.8 for the position, depicts the results of the four measurement categories, each of them in a separate diagram. In the upper left corner, the results for the test track category are depicted. The highway, country road and urban category follow in the upper right, lower left and lower right corner, respectively. In each Stanford Integrity Diagram, the respective PL is plotted against the corresponding error as explained in Section 2.4.

As for the FFL filters, the GNSS reception conditions have a strong influence on the integrity performance of the multi-layer fusion. When looking at the results for the SFL's

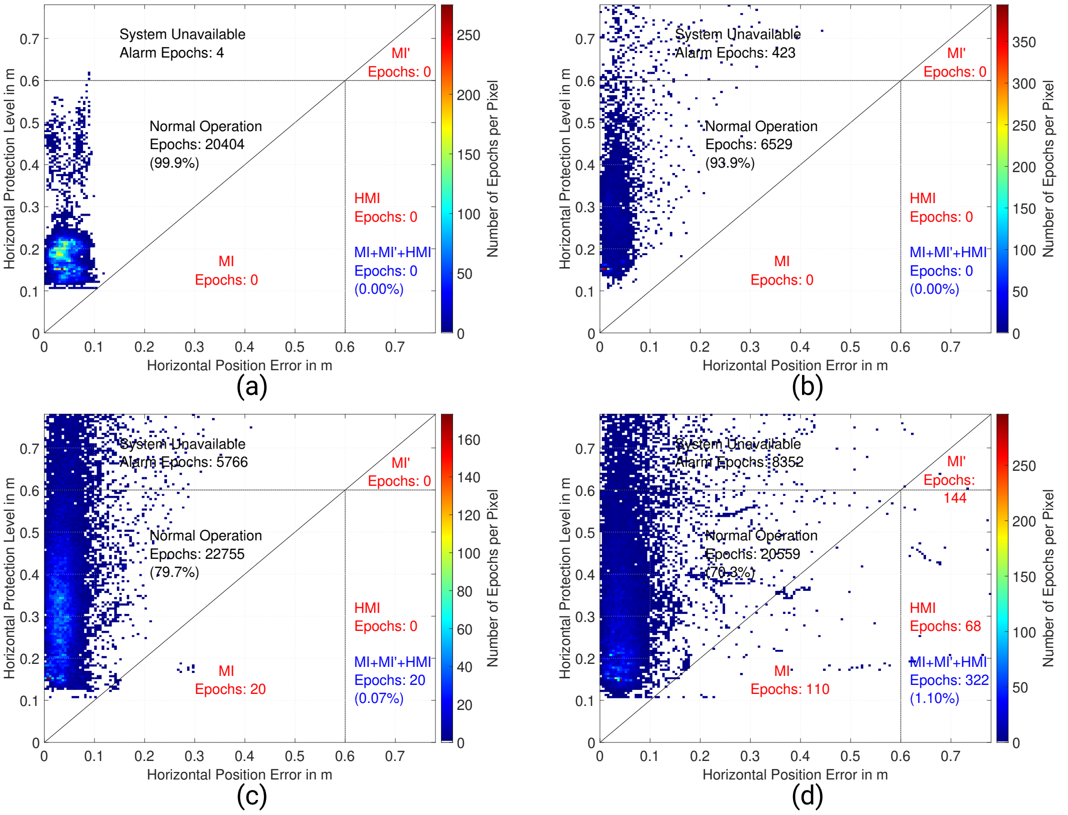


Figure 7.8.: Stanford Integrity Diagrams for the protection level of the horizontal position error of the second fusion layer output in measurements in the test track (a), highway (b), country road (c) and urban (d) category

position solution in Figure 7.8, a very good performance is observed in the test track category in Subfigure (a). As expected, there are no epochs with misleading information and the availability is superior with more than $99.9\,\%$. The values in Table 7.3 confirm that not only the accuracy could be increased compared to the FFL filters but also the availability while keeping an empirical IR of zero. In the highway category in Subfigure (b), the integrity performance of the multi-layer fusion does not worsen. There are still no epochs with misleading information. The availability decreases slightly compared to the test track category. The values in Table 7.3 show that the availability is again higher than the one of the FFL filters. On country roads depicted in Subfigure (c), there are about $0.07\,\%$ of the epochs with misleading information but no epochs with HMI. The integrity requirement of an emprical IR smaller than one percent is fulfilled with a margin of more than one order of magnitude. The SFL's empirical IR is smaller than the majority of the values from the filters in the FFL. The SFL's availability decreases compared to the previous measurement category, resulting in a value similar to the performance of FFL filters. In the urban category depicted in Subfigure (d), the GNSS reception conditions are strongly degraded. This leads to a weaker integrity performance of the Filter 3 (F3) in the FFL. With an empirical IR of around $1.17\,\%$ the third filter exceeds the requirement of one percent. The multi-layer fusion depends on FFL filters and is therefore influenced by this. The SFL's empirical IR amounts to around $1.1\,\%$ and is sightly higher than requirement. About $0.2\,\%$ of the epochs contain HMI. Here, it becomes clear that the multi-layer fusion might not fulfill the integrity requirement if the filters in the FFL do not fulfill this requirement. Regarding the availability (and also accuracy as mentioned before), the multi-layer fusion outperforms the filters in the FFL even in this challenging measurement category.

The results for the SFL's velocity solution are depicted in Figure 7.9. They confirm the very good impression pointed out in the previous paragraph while the performance is improved. In terms of the integrity, the requirement of an empirical IR smaller than one percent can be kept in all four measurement categories. The empirical IR is zero in three out of the four measurement categories and smaller than $0.1\,\%$ in the challenging urban category. Overall, there are also no epochs with HMI. Additionally, the empirical IR is lower than the one of the FFL filters while the availability is higher in all four measurement categories.

For the heading solution of the multi-layer fusion, the integrity requirement of an empirical IR smaller than one percent is also fulfilled in all four measurement categories as depicted in Figure 7.10. In the test track category, the SFL's PL is slightly over-optimistic leading to about $0.7\,\%$ of the epochs with misleading information. The two filters implemented in IE 8.90 (F2 and F3) are not fulfilling the integrity requirement in this category which leads to this counter-intuitive result. Nevertheless, this shows that the multi-layer fusion can fulfill the integrity requirement, even though two filters in the FFL are exceed-

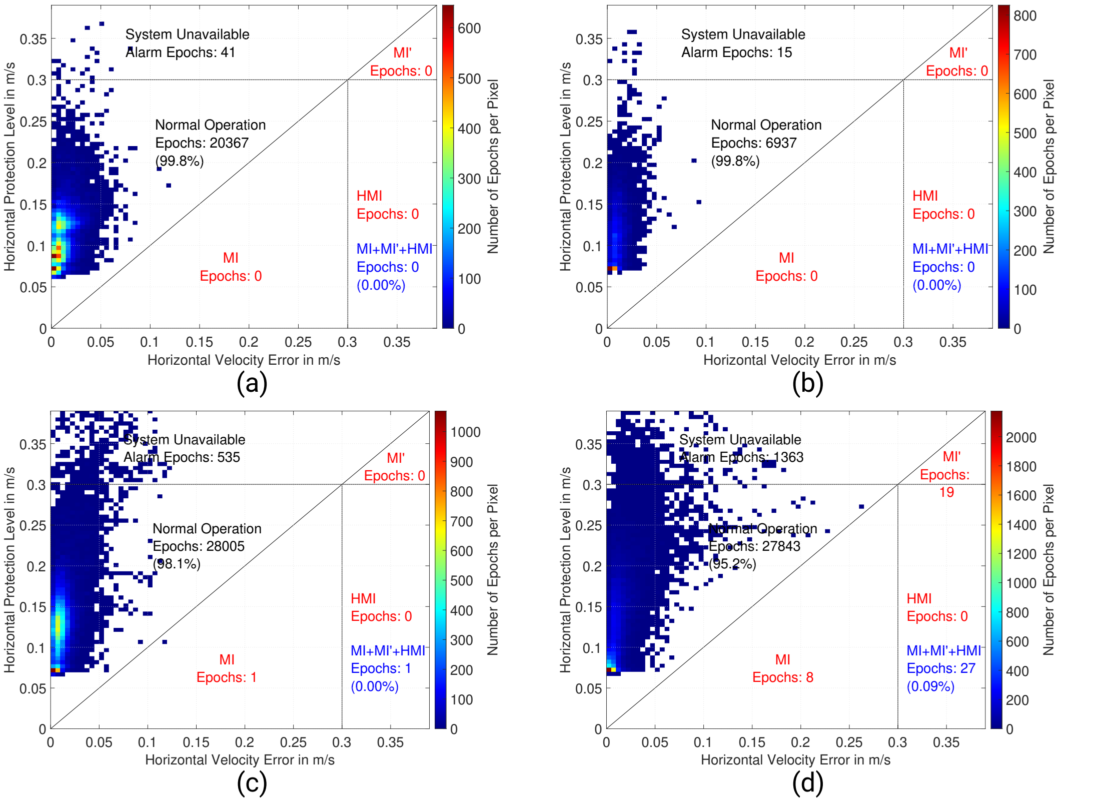


Figure 7.9.: Stanford Integrity Diagrams for the protection level of the horizontal velocity error of the second fusion layer output in measurements in the test track (a), highway (b), country road (c) and urban (d) category

ing it slightly. In the other three measurement categories, the integrity requirement is also fulfilled, here with a margin of at least one order of magnitude. The overall availability of the heading solution is high with values over 97 % in the first three categories and over 92 % in the urban category. The SFL increases the availability compared to the FFL filters in the country road and urban category.

Furthermore, a closer look at the continuity of the SFL's outputs is taken. In the analyzed measurements, the multi-layer fusion provides a solution in all epochs, i.e., there were no outages. Theoretically, a situation might occur where no input to the SFL confirms any other input. The approval voting described in Section 6.3.3 will set all weights w_{u_i} to zero in this case. To prevent that no output of the SFL is computed, further measures have to

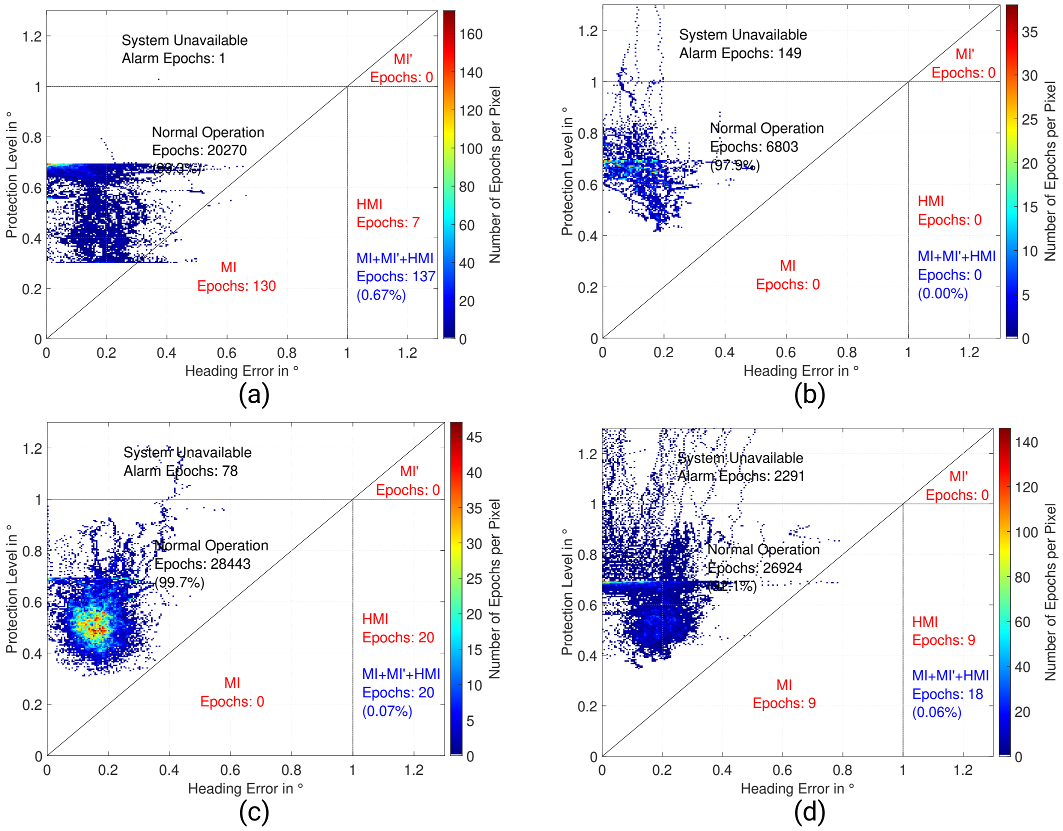


Figure 7.10.: Stanford Integrity Diagrams for the protection level of the horizontal attitude error of the second fusion layer output in measurements in the test track (a), highway (b), country road (c) and urban (d) category

be taken. One option is to define one filter of the FFL as fall-back solution beforehand, which is selected in this special case. If all weights w_{u_i} were set to zero by the approval voting, the SFL would output this filter's outputs then. Depending on the use case, a warning or error message should be issued.

In summary, the performed analysis showed several advantages of the multi-layer fusion in comparison to single-layer filters as they are used in the FFL. In terms of accuracy analyzed using the 95 % quantile of the error in the position, velocity and attitude solution, the multi-layer fusion leads to a significant improvement. For the position, the accuracy is better than for the filters on the FFL in all measurement categories and for the other two quantities this holds true in three out of four categories. Regarding the integrity, the multi-layer fusion fulfills the specified requirement in all measurement categories, with only one exception for the position in the challenging urban category. This can be explained by a violation of the integrity requirement by one of the inputs in this category. The empirical IR is always lower for the SFL's ouputs than for the ones of the FFL, with only one slight exception for the heading output. In the vast majority of the cases, the availability is also increased by the SFL. Only in three out of twelve quantity-category combinations, the availability is not increased but on a similar level compared to the FFL.

If this performance is sufficient for the desired use case, will depend on the concrete requirements. For this work, the requirements mentioned earlier in Table 4.1 in Section 4.1.1 are used. The fulfillment of the first three of them was already discussed in Section 7.1. The fifth one relates to the integrity information which are provided and fulfill the specified requirement as mentioned before. The remaining fourth requirement talks about the operation in urban environments. Here, the performance of the implemented VDSE alone might not be sufficient for the use case of automated driving. Therefore, it is proposed to combine the implemented multi-layer fusion with a second localization function using environmental sensors (e.g., camera, LiDAR) to fulfill the requirements. This will be discussed in the following Chapter, which explains the application of the implemented VDSE in the research project UNICAR*agil*.

8. Application in the Research Project UNICAR*agil*

In this Chapter, the application of the developed multi-layer data fusion in the VDSE of the research project UNICAR*agil* is discussed as one example how the implemented algorithms can be applied to the use case of automated driving. A general introduction into the project is given in Section 2.5. First of all, the integration of the developed multi-layer fusion as VDSE in the automation domain is described in Section 8.1. Secondly, the commissioning tests of the VDSE including first results are presented in Section 8.2. Afterwards, a short summary regarding the status of the VDSE in the still ongoing research project UNICAR*agil* including an outlook over the remaining duration of the project is given in Section 8.3.

8.1. Integration

This Sections explains how the algorithms developed in this work are applied in the VDSE of the research project UNICAR*agil*. To begin, the general system architecture in the automation domain is discussed in Section 8.1.1. Then, the interaction of the two localization functions in UNICAR*agil* with the Offset-Correction is described in further detail in Section 8.1.2. Afterwards, the implementation of the VDSE itself is explained in Section 8.1.3.

8.1.1. System Architecture of Automation Domain

In the automation domain of UNICAR*agil*, as explained by Buchholz et al. (2020), several services work together as depicted in Figure 8.1. The video-based localization outputs a pose (position and attitude) relative to a high-definition map and sends it via the Automotive Service-Oriented Software Architecture (ASOA) to the trajectory planer. Here, decisions about the behavior of the vehicle are made and the target trajectory is computed. The trajectory control receives this trajectory and aims to minimize the tracking error, meaning the difference between the actual vehicle state and the state included in the

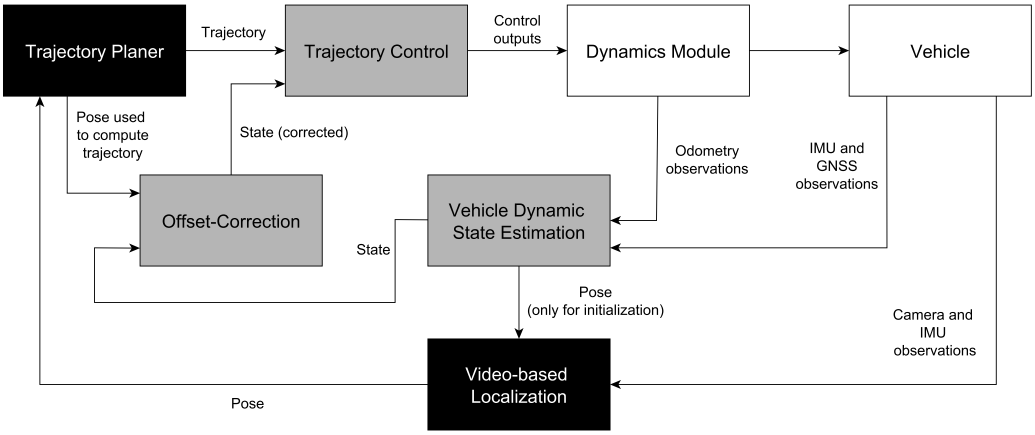


Figure 8.1.: System architecture of automation domain in the research project UNICAR*agil* (Figure based on content of Homolla et al., 2020)

target trajectory. To do so, the trajectory control computes suitable control outputs for the dynamics modules. These implement the inputs and actually drive the vehicle in the desired way.

To enable the trajectory control to compute the tracking error, which is needed to determine the control outputs, it needs to know the actual vehicle state, including its pose, velocity, acceleration and angular rate. Since the trajectory control is running with 50 Hz, it needs to receive the actual vehicle state with at least this rate.

In Figure 8.1, the colors mark the affiliation of services to one of the three layers of the computational architecture in UNICAR*agil*, which are the cerebrum, brainstem and spinal cord. These layers are based on the human brain structure as described in detail by Woopen et al. (2018). The colors in Figure 8.1 also indicate the update rate of the respective services. Those services colored in black are running on the cerebrum, which allows computationally challenging operations but limits the update rate to 10 Hz. Services marked with the color gray are affiliated with the brainstem, which allows higher update rates for computationally less expensive operations. The trajectory control and the Offset-Correction run directly on the real-time core of the brainstem ECU with 50 Hz, while the VDSE is implemented on a dedicated ECU and is running with 100 Hz. (The dynamics module is colored in white, since it is affiliated with the spinal cord.)

In the following Section, details about the interaction of the two localization functions (video-based localization and VDSE) as well as the need for the use of the Offset-Correction are explained.

8.1.2. Interaction with Video-based Localization and Offset-Correction[1]

The system architecture presented in the previous Section contains two localization functions, the video-based localization and the VDSE. As pointed out by the author together with Homolla and Winner (2020), unintended system behavior caused by inconsistencies in the outputs of these functions needs to be prevented. Together with other requirements, including the need for the trajectory control's inputs in the correct update rate, the Offset-Correction is derived as solution for this challenge and explained in the mentioned reference. In the following, the concept of the Offset-Correction is briefly described based on the mentioned reference and on the patent application filed by the author together with Homolla and Winner (2021).

The Offset-Correction consists of three functional parts. The first one of these computes the offset between the poses put out by the two localization functions. To do so, the pose of the VDSE is compared to the pose from the video-based localization which was used for computing the target trajectory. From a ring-buffer storage for the VDSE's pose in the Offset-Correction, the pose with the correct time stamp is interpolated. A new offset is computed every time a new target trajectory is sent to the trajectory control. The second function applies this offset to the vehicle state put out by the VDSE, transforming the vehicle state and the target trajectory in a common reference system. This function is called every time before executing the trajectory control. Besides that, there are additional functions of the Offset-Correction, which treat special cases leading to unusual jumps in the VDSE's pose, e.g., measurement updates after a longer period with degraded GNSS signal reception or a (re-)initialization of the VDSE. If one of these special cases is detected, the first function to recompute the pose offset will be executed directly to prevent using an outdated value for the pose offset. Furthermore, the pose offset computed in the Offset-Correction is used as input to the self-perception module in UNICAR*agil* for its monitoring of the vehicle's capabilities.

All in all, the Offset-Correction enables the use of the chosen system architecture in UNICAR*agil* with two localization functions as depicted in Figure 8.1 by preventing unintended behavior caused by inconsistencies in-between the localization functions. A localization function relying on environmental perception and high-definition maps (video-based localization) is combined in this way with the VDSE. Details about the implementation of the VDSE in UNICAR*agil* are given in the next Section.

[1] This Section is based on a previous work written with author's collaboration (Homolla et al., 2020).

8.1.3. Implementation of Vehicle Dynamic State Estimation

In UNICAR*agil*, the VDSE is implemented on a dedicated ECU (product name: iNAT-4C – data sheet: iMAR Navigation GmbH, 2019) manufactured by the project partner iMAR Navigation GmbH (iMAR). The ECU contains four μController boards with ARM-based processors, two IMUs and a GNSS receiver module. Both IMUs are MEMS based but IMU A, an ADIS 16465-1 (Analog Devices Inc., Cambridge, MA, United States), has a better performance according to its classification as mid performance MEMS-IMU in comparison to IMU B, a BMI160 (Bosch Sensortec GmbH, Reutlingen, Germany), classified as consumer grade MEMS-IMU. In Table A.1 in the Appendix, the used IMU's key characteristics are compared confirming the superior performance of IMU A according to the information from the manufacturer's data sheets. The used GNSS receiver module is the previously mentioned dual-frequency, multi-constellation, dual-antenna RTK-GNSS receiver OEM7720 (NovAtel Inc., Calgary, AB, Canada). To complete the sensor setup, odometry information from the dynamics modules is received via the ASOA.

In Figure 8.2, the system architecture of the VDSE in UNICAR*agil* is depicted containing all mentioned sensors. Besides that, three fusion filters are shown which run each on one of the four μController boards and are developed by independent teams. The remaining fourth μController board is used for the SFL to determine the VDSE's outputs and for a communications module to send and receive messages via the ASOA.

Each filter inputs a subset of the available sensor information, as depicted in Figure 8.2. Filter 1 inputs odometry information from all four wheels and observations from IMU A as well as from the GNSS receiver. Filter 2 and 3 also use the GNSS receiver but only odometry information from one wheel as single odometer. In contrast to the others, Filter 2 inputs the observations of IMU B.

All programs on the μController boards are implemented in C++. For Filter 1, the fusion filter described in Chapter 4 is converted to C++ code using MATLAB Coder and supplemented by a wrapper to handle the interfaces. Filter 2 and 3 are developed and implemented by iMAR. Their implementation is not disclosed but Filter 2 and 3 differ only with regard to the configuration for the used sensor setup. The concept for the SFL corresponds to the explanations in Chapter 6.

At the time of writing (April 2022), the research project UNICAR*agil* is not completed yet (final demonstration scheduled for May 2023). Currently, only Filter 1 and 2 are realized and running on the VDSE's ECU. Since the SFL is intended to be used with three or more filters in the FFL, a preliminary version for the commissioning tests is used, which can be configured to output the results of one FFL filter. Furthermore, the implemented version of Filter 1 in the UNICAR*agil* vehicles contains minor differences (e.g., regarding the GNSS measurement model) compared to the description in Chapter 4, since the development

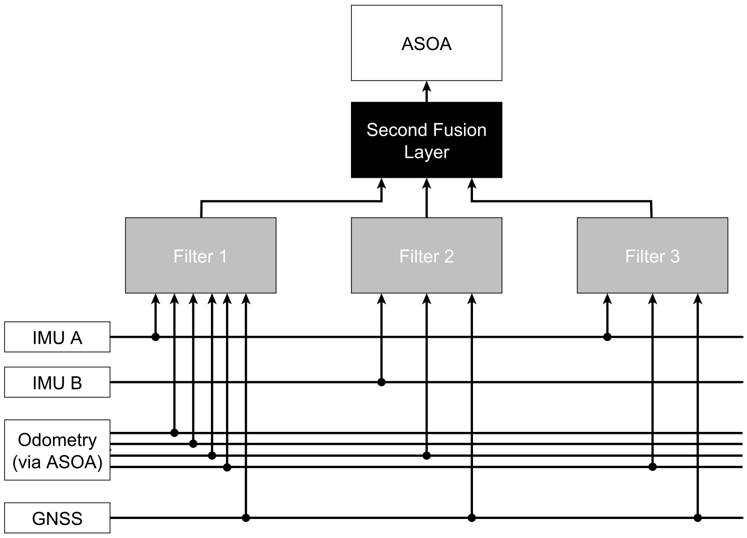


Figure 8.2.: System architecture of Vehicle Dynamic State Estimation in UNICAR*agil* (Figure based on previous version published by the author in Buchholz et al., 2020)

of the fusion filter continued after the VDSE's commissioning tests in 2021. Besides that, the time-stamped odometry information is not yet available via the ASOA in the UNICAR*agil* vehicles. Therefore, the odometry measurement update in the fusion filters is deactivated at this stage. Modifications with respect to the odometry measurement update in comparison to conventional vehicles concern mainly the slip estimation and stochastic modeling. These modifications will be developed and tested as soon as the odometry data is available in the UNICAR*agil* vehicles. In order to resolve the mentioned restrictions, updates to the VDSE's software are planed in the further course of the research project UNICAR*agil*.

8.2. Commissioning and Measurement Results

In this Section, insights into the commissioning of the UNICAR*agil* vehicles with respect to the VDSE are provided. First of all, the condition of the prototype vehicles at the time of commissioning and the selected procedure are described in Section 8.2.1. Secondly, measurement results from the commissioning tests are discussed in Section 8.2.2.

8.2.1. Commissioning and Condition of Protype Vehicles

During the year 2021, the commissioning of the VDSE in UNICAR*agil* has been performed. In each of the four prototype vehicles, a first version of the VDSE's software (including fusion filters, SFL and communications module) has been deployed, which will be updated in the further course of the research project. For the VDSE's commissioning tests, the prototype vehicles are driven manually by an operator (manual mode). In this operation mode only the front wheels are steered and the rear wheels are fixed, as in conventional vehicles. Note that the UNICAR*agil* vehicles can steer all four wheels but only in the automated mode which includes using the trajectory planning and trajectory control. The velocity of the vehicles is restricted to a maximum of $5\,\mathrm{km/h}$ at this stage of commissioning. Figure 8.3 depicts three UNICAR*agil* vehicles during testing in January 2022 on the TU Darmstadt airfield in Griesheim, which is used as test track.

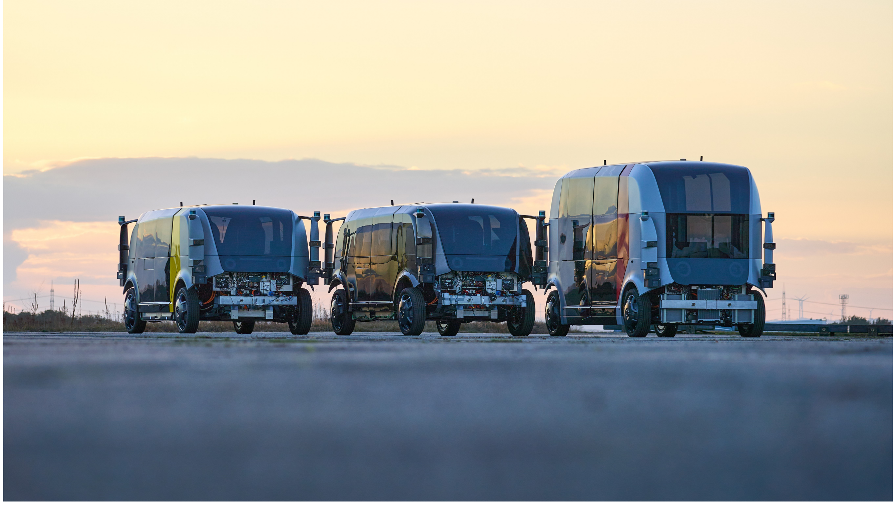


Figure 8.3.: UNICAR*agil* vehicles (autoTAXI, autoELF and autoSHUTTLE – from left to right) during testing on 6 January 2022 on TU Darmstadt airfield in Griesheim, Germany (photo by Patrick Pintscher)

The commissioning begins with flashing the software on the VDSE's ECU. Afterwards, a set of preliminary test during standstill to verify that all hardware components are working and communicating in the correct way is performed. Subsequently, dynamic commissioning tests are carried out. At this stage, only the VDSE is analyzed without considering connecting services, e.g., the trajectory control or Offset-Correction. The interaction with these services will be investigated in following tests during the further course of the research project UNICAR*agil*. Results of the commissioning tests are presented in the next Subsection.

8.2.2. Measurement Results From Dynamic Commissioning Tests

The aim of the dynamic commissioning tests is to verify the VDSE's real-time capability and outputs. Regarding the former point, there have been no abnormalities during any of the tests performed. The VDSE has been able to provide its outputs reliably at a rate of $100\,\text{Hz}$. Regarding the latter point, a thorough analysis of the VDSE's performance in comparison to a reference solution has already been performed before the commissioning of the UNICAR*agil* vehicles and is presented in Chapter 7. Therefore, the purpose of the dynamic commissioning tests is to detect mistakes during the VDSE's integration or configuration. Examples for such mistakes are in the areas of lever arm configuration, confusion of cable connections (e.g., for the two GNSS antennas) or coordinate systems (e.g., body and navigation coordinate system). For the VDSE's outputs, these mistakes would lead to gross errors. The plausibility checks aim to uncover these errors by dynamically stimulating the sensors within the limits of possible operation at this stage. A trajectory in the shape of an eight is chosen, which is driven in both directions, to combine sensor stimulations in different directions.

In a first step, the position outputs of the VDSE's fusion filters are plotted in a horizontal plane and inspected visually, one by one, to analyze their plausibility. Figure 8.4 depicts the trajectories of Filter 1 and 2 during a commissioning test with the autoSHUTTLE recorded on 13 October 2021 on TU Darmstadt airfiled in Griesheim, Germany in Google Earth.

A typical example of an erroneous output to look out for at this stage would be a jagged trajectory which obviously does not correspond to the driven path. This is not the case here, the trajectories seem plausible. No conspicuities are visible.

In a second step, the plausibility checks go into further detail. Since Filter 1 and 2 have been implemented, validated and configured by two independent teams, a common implementation or configuration error is considered as not probable (validation of Filter 1 in this work, Filter 2 in other projects of project partner iMAR). Therefore, the second step contains a comparison between the fusion filter's outputs. Figure F.1 in the Appendix depicts the differences between the trajectories of the two shown filters in this measurement. The differences are computed at the reference point of the vehicle, after a constant position bias of roughly $0.15\,\text{m}$ has been removed (coming from an error in the configuration which was resolved afterwards). In UNICAR*agil*, the reference point is the intersection of the wheel's contact patches at the height of the VDSE ECU's reference point. The differences' Root Mean Square (RMS) amount to around $0.04\,\text{m}$, $0.03\,\text{m/s}$ and

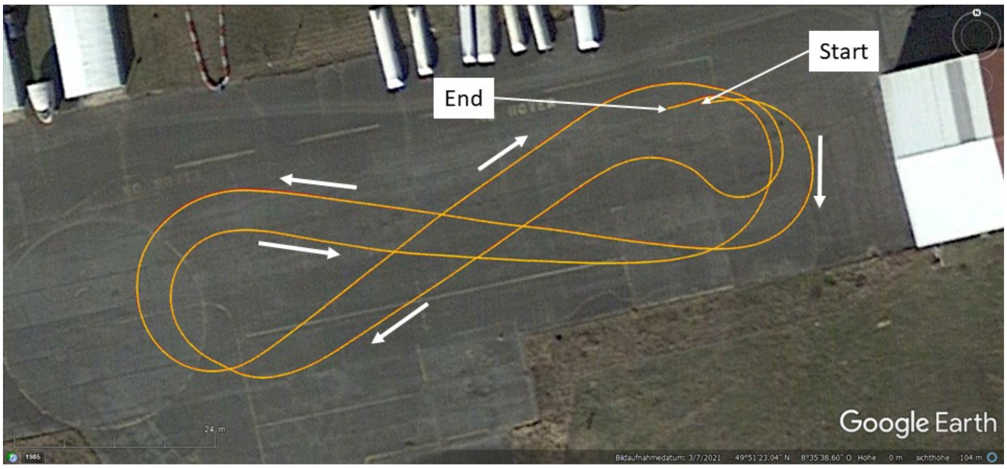

Start
End
Google Earth
24 m
1985
Bildaufnahmedatum: 3/7/2021 49°51'23.04" N 8°35'38.60" O Hohe 0 m sichthohe 104 m

Figure 8.4.: Commissioning tests of VDSE in UNICARagil recorded on 13 October 2021 on TU Darmstadt airfield in Griesheim, Germany: Trajectories of two fusion filters, red – filter develop by project partner, yellow – filter developed in this work (Screenshot taken from Google Earth Pro)

$0.49\,^{\circ}$ for the horizontal position, velocity and attitude differences, respectively. These results confirm the consistency of the two fusion filters inside the VDSE.

As mentioned in Section 8.1.3, the odometry has not been available during the commissioning tests of the VDSE (it will be added in the further course of the research project). This difference to the implementation used in Chapter 7, impacts mainly the filter's performance in GNSS denied environments. To analyze this impact, a GNSS outage is simulated in a Software in the Loop (SiL) setup. Obviously, transient phases of weak GNSS reception (but not complete GNSS denial) including effects like multipath reception as they are typically present in e.g., urban areas, cannot be tested in this way. However, this is not a concern here, since these effects have been analyzed in Chapter 7 already.

For the simulated GNSS outage the previously discussed measurement of 13 October 2021 is used. Figure 8.5 depicts the trajectories for this scenario in Google Earth. The yellow trajectory is identical to Figure 8.4, meaning it is the output of the filter running in real-time on the VDSE's ECU during the commissioning test. The green trajectory is generated by replaying the mentioned measurement in MATLAB and executing the fusion filter there. The simulated GNSS outage effects the PVT and attitude solution for a

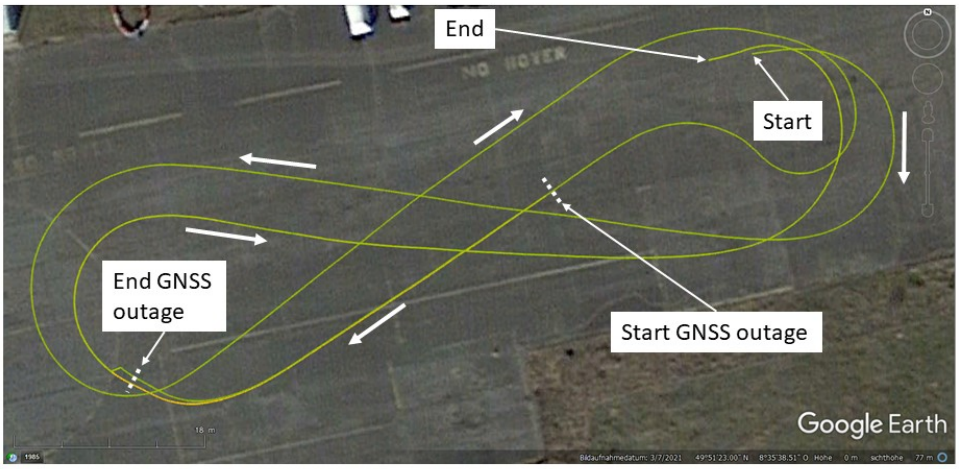


Figure 8.5.: Commissioning tests of VDSE in UNICARagil recorded on 13 October 2021 on TU Darmstadt airfield in Griesheim, Germany: Simulated GNSS outage, yellow – filter running in real-time without GNSS outage, green – SiL with simulated GNSS outage (Screenshot taken from Google Earth Pro)

duration of $30\,\mathrm{s}$. In the Figure, its start and end are indicated by a dotted line.

Naturally, the fusion filter's trajectory diverges during the simulated GNSS outage from the trajectory put out during normal operation, as depicted in the mentioned Figure. However, the trajectory stays in a plausible shape and diverges slowly. Figure F.2 in the Appendix depicts the differences to the outputs for the horizontal position, velocity and attitude solution over time. At time zero, the simulated GNSS outage begins and at $30\,\mathrm{s}$ it ends. The differences stay within expected limits and do not exceed $1\,\mathrm{m}$, $0.1\,\mathrm{m/s}$, $0.1\,°$ for the horizontal position, velocity and attitude solution, respectively. After integrating the odometry, these values are expected to decrease significantly. After the end of the simulated GNSS outage, the differences in the horizontal position and velocity solution are resolved within a few seconds. For the attitude solution, this reswing takes slightly more than a minute, which can be explained by the rather small deviation during the simulated GNSS outage in comparison to its accuracy in this environment ($95\,\%$ quantile of the attitude error is about $0.3\,°$ in the test track category, see Section 7.1).

8.3. Summary

In this Chapter, one application of the developed multi-layer data fusion as VDSE in automated driving is presented. The integration in the unique system architecture of the research project UNICAR*agil* is discussed. At the time of writing (April 2022), the research project UNICAR*agil* is not completed so far, wherefore there are open tasks and restrictions which need to be resolved in the further course of the research project.

Nevertheless, a preliminary version of the VDSE, based on the algorithms developed in this work and by the project partner iMAR, is running in the four prototype vehicles of the research project UNICAR*agil*. In this way, the VDSE's outputs are provided via the ASOA such that commissioning of subsequent services (e.g., the trajectory control) is made possible.

Both fusion filters running in the mentioned preliminary version of the VDSE have been validated before in other vehicles: Filter 1 in this work and Filter 2 by the project partner iMAR in other projects. Therefore, the commissioning tests have been reduced to plausibility checks. These have been passed without conspicuities, wherefore the commissioning of the VDSE is successfully completed.

Till the final demonstration of the research project UNICAR*agil*, which is scheduled for May 2023, optimizations of the VDSE including the removal of the mentioned restrictions will be performed. Together with the results from further testing including the interaction with other services, this will lead to a continuous improvement of the VDSE's software deployed by regular updates.

9. Conclusion and Outlook

This Chapter summarizes the achieved results and draws conclusions with respect to the objectives set in the introduction. Afterwards, further research directions are indicated.

9.1. Conclusion

Three main objectives of this work have been defined in the introduction. The first objective relates to the design and implementation of a multi-sensor data fusion filter as basis for the second and third objective. A real-time capable filter was developed inputing sensor observations from a GNSS receiver, an IMU and wheel odometry sensors, which corresponds to the sensor setup planned in the research project UNICAR*agil*. To achieve a high accuracy, a RTK-GNSS receiver is used for which a corresponding error model was developed. Additionally, zero velocity and zero angular rate updates as well as measures to detect outliers were implemented.

In favorable GNSS reception conditions, the $95\,\%$ quantile of the horizontal position error amounts to less than $0.14\,\mathrm{m}$. For the horizontal velocity and attitude, these values are below $0.07\,\mathrm{m/s}$ and around $0.4\,°$, respectively. In more challenging environments, e.g., in urban areas, the performance of the velocity and attitude solution can be maintained with similar error amounts, while the mentioned value for the position error increases to around one meter (for the data sets evaluated in this work).

To monitor the integrity of such a filter's outputs, as specified by the second objective of this work, three concepts were developed. The first one (kSigma) uses a traditional approach based on the covariances estimated by the implemented KF. The second concept (KIPL) includes modeling the errors of the fusion filter's inputs as multi-variate Student distributions and propagating them through the filter. Multiple Hypothesis Solution Separation is the foundation for the third concept (ARAIM) in which subsets of the filter's inputs are processed in a bank of filters and the difference in their outputs forms the basis for the integrity measures.

The comparison showed that in favorable conditions, all three implemented integrity concepts provide PLs with an empirical IR below the specified threshold of one percent.

However, only the first two concepts are able to output meaningful PLs in more challenging environments. Compared to the velocity and attitude, the position solution is the most challenging with regard to integrity monitoring in these environments (only horizontal components of the mentioned quantities are in focus here). On the one hand, the traditional approach maintains a high availability for the position solution with around 95 % on highways and country roads, and about 84 % in urban areas (using an alert limit of 0.6 m). On the other hand, the second concept using an error modeling with multi-variate Student distributions is the only one fulfilling the mentioned integrity requirement in all tested environments including urban areas with an availability between 11 and 21 percentage points below the traditional approach, due to being more conservative in its computation of integrity measures.

Building upon these results, a federated data fusion architecture with integrity monitoring was introduced. It contains a first layer with several redundant multi-sensor data fusion filters like the one presented in this work. These filters input subsets of the available sensor data and output integrity information for their estimated states computed with an integrity concept as the ones developed in this work. A second layer of this multi-layer fusion implements plausibility checks, approval voting and a data fusion, where the latter two steps are based on the integrity measures computed in the first layer. This fusion architecture offers several safety benefits compared to using one filter to fuse all sensor data, which include a higher reliability by the increased redundancy and an improved FDE performance realized by the approval voting in the second layer.

Besides that, the data fusion architecture's performance is significantly better compared to the outputs of the first layer's fusion filters. The position accuracy is higher in all categories, which also holds true for the velocity and heading solution in three out of four categories. In the test track category, the 95 % quantile of the horizontal position error amounts to less than 0.08 m. On highways, this performance can be maintained and also on country roads the mentioned measure stays lower than before with about 0.20 m. In the urban category, it amounts to less than 0.79 m, an improvement of more than 0.2 m compared to any of the filters in the first layer.

Furthermore, the integrity requirement of an empirical IR below one percent is fulfilled in all quantity-category combinations with only one slight exception (by one tenth of a percentage point) for the position in urban areas which is caused by one of the filters in the first layer not fulfilling their integrity requirement. The second fusion layer also decreases the empirical IR with one minor exception for the heading solution. Additionally, the availability is increased, for the position solution in all cases, for the velocity and heading solution in three out of four categories. In this way, availabilities for the position solution of more than 99.9 % in the test track category and about 94 %, 80 % and 71 % in the highway, country road and urban category are achieved.

All in all, with the implemented data fusion a high accuracy is achieved, resulting in e.g., a $95\,\%$ quantile of the horizontal position error under a decimeter in favorable GNSS reception conditions. Meaningful integrity measures were computed by the implemented concepts, fulfilling the set integrity requirement and enabling the user to assess the data fusion's capabilities during operation, even in challenging environments, e.g., urban areas. The developed federated data fusion architecture combines several fusion filters and uses the integrity information. In this way the performance is significantly improved in terms of higher accuracy, lower empirical IR and higher availability in the vast majority of the cases (otherwise keeping it similar, compared to the fusion filters in the first layer). Besides that, this architecture brings a higher redundancy among other safety benefits.

However, there are some limitations to the application of the presented data fusion. First of all, the GNSS reception conditions limit the achievable performance. With the set requirements in terms of accuracy, the availability is only sufficient for automated driving in certain areas determined by the GNSS reception conditions. A possible solution to overcome this challenge is presented in the application for the research project UNICAR*agil*. The introduced Offset-Correction is used to integrate a second localization function. This second localization function does not depend on GNSS but rather relies on camera observations and high-definition maps, which leads to a better performance in GNSS denied areas. On the other hand, the weaknesses of such an approach (e.g., a lower output data rate, no velocity estimation) are complemented by the VDSE presented in this work. Another limitation to the achievable performance is the given sensor setup. In this work, only the three mentioned sensor types (GNSS, IMU, odometry sensors) are used but no sensors for Signals of Opportunity (SOP), e.g., cellular data used for localization, or environmental sensors, e.g., cameras, LiDAR, are integrated in the sensor fusion. Among others, these points are discussed further in the following Section, indicating future research directions.

9.2. Outlook

Building upon the results of this work, it is advised to conduct further research, whose directions are indicated in this Section.

First of all, the implementation of the presented concepts and algorithms in the research project UNICAR*agil* will be continued. Besides the fusion filter with integrity monitoring, all functionalities of the second fusion layer will be implemented in C++ on a μController board to run in real-time in the UNICAR*agil* vehicles. Further testing and optimization of the VDSE in interaction with other services will be carried out till the final demonstration (scheduled for May 2023).

Additionally, the limitations mentioned in the previous Section with respect to the developed fusion filter need to be further analyzed. As the GNSS reception conditions were identified as decisive influence on the filter's performance, new ways of processing GNSS observations focusing on challenging environments may have a significant impact. One option is to implement RTK-GNSS processing techniques as presented by Humphreys et al. (2018) to improve the integer ambiguity fixing leading to a higher RTK-GNSS availability in urban environments. Alternatively, new PPP methods or combinations of PPP and RTK methods to process the GNSS observations may be used as proposed e.g., by Li et al. (2021). An implementation independent of the fusion filter for loose coupling or integrated in it as tight coupling is possible.

Another option to enhance the accuracy in urban environments is to include additional sensors. Environmental sensors, e.g., cameras, LiDAR and RADAR sensors, potentially in combination with corresponding high-definition maps offer additional observations unaffected from satellite reception conditions and may be used for state estimation as shown by e.g., Ziegler et al. (2014) or Wan et al. (2018). Furthermore, sensors to receive SOP, e.g., cellular data (5G), may be integrated as additional measurement updates to the fusion filter, which aim to improve the filter's availability in urban environments as shown e.g., by Jia et al. (2021).

Besides the improvements with respect to the filter's accuracy, the mentioned options are also expected to lead to a better integrity performance. A higher redundancy by additional sensors and therefore more measurement updates might enhance the outlier detection and reduce the empirical IR. To do so, further testing of the developed integrity algorithms with additional sensors will be necessary. The error characteristics of each sensor need to be analyzed in order to decide about a suitable error modeling approach. Moreover, supplementary experiments with varying configurations of the integrity requirements in terms of IR and AL depending on the intended application are recommended. If there are requirements for the continuity including a minimum solution quality for the application, a further analysis focusing on an investigation of the continuity risk is needed.

Moreover, enhancements of the integrity measures itself are of interest. In this work, the integrity information in terms of PLs is analyzed as horizontal and vertical part. It remains to be investigated how the cross-track PL differs from the along-track one, since the former is more critical for automated driving as pointed out e.g., by Welte (2017, p. 35).

Additionally, the developed integrity concepts might be combined depending on the available computational resources. This means that inside the integrity layer different algorithms are used for the different output quantities. As the position solution showed a heavy tailed error distribution, the second concept (KIPL) using an error modeling with multi-variate Student distributions might be used here, while the velocity and attitude PL might be computed by the traditional approach (kSigma) based on covariances estimated by the KF, for example. Alternatively, the integrity concepts might be combined depending on the environmental conditions. In favorable conditions, the traditional approach (kSigma) could be used since the errors are expected to be normally distributed here. In challenging environments, the second concept (KIPL) could be used to output more conservative integrity measures taking the heavy tailed error distribution into account. A high-definition map or a sky-pointing camera are examples of how the conditions might be identified for the GNSS reception.

With respect to the developed data fusion architecture, in this work an implementation with three filters in the first layer was analyzed because of the chosen application. Another option to enhance the data fusion architecture's performance is to investigate implementations with a higher number of redundant fusion filters in the first layer. Together with the previously mentioned approach to extend the sensor setup, new possibilities for forming subsets of sensor data in the first layer open up further opportunities for improvement.

Moreover, the use of two localization functions as implemented in the research project UNICAR*agil* might be reconsidered depending on the specific application. Instead of a relative localization function inputing environmental sensors such as the video-based localization in UNICAR*agil* and one absolute localization function such as the multi-layer fusion implemented in this work, all sensors might be integrated in one multi-layer fusion: Combinations of relative and absolute sensors form the mentioned subsets of sensor data for the fusion filters in the first layer, which send their outputs and integrity measures to a second fusion layer as presented in this work.

Before the indicated research directions are pursuit, it is crucial to precisely specify the application's requirements. For the use case of the implemented data fusion as state estimation in automated driving, the requirements in terms of the navigation performance criteria (accuracy, integrity, availability, continuity) given by Pullen (2008) are not generally agreed on so far, as mentioned in Section 5.1. Refining the requirements is advised in order to identify objectives for future research.

Lastly, the presented data fusion architecture with integrity monitoring for state estimation was developed for an application in automated driving but is not limited to this field. Any application of state estimation can benefit from the results of this work, including applications in e.g., other land vehicles like for precision agriculture applications in automated farming, marine vessels such as automated cargo ships, aerial vehicles for example UAVs, household and garden devices like vacuum robots or robot lawn mowers. Depending on the specified requirements and the available hardware in the chosen application, adaptations of the configuration and parameter tuning might be necessary as well as extensive testing in simulation and in real-world experiments together with an analysis of the performance with respect to the decisive quantities.

References

Altamimi, Z. (2018). EUREF Technical Note 1: Relationship and Transformation between the International and the European Terrestrial Reference Systems [Version of 28 June 2018]. http://etrs89.ensg.ign.fr/pub/EUREF-TN-1.pdf

Analog Devices Inc. (2018). Analog Devices ADIS-16465 Data Sheet [Accessed on 27 November 2018]. https://www.analog.com/media/en/technical-documentation/data-sheets/adis16465.pdf

Angrisano, A. (2010). *GNSS/INS Integration Methods* (Doctoral dissertation). UNIVERSITA' DEGLI STUDI DI NAPOLI "PARTHENOPE". https://www.pang.uniparthenope.it/sites/default/files/PhD%20Thesis_Angrisano.pdf

Arribas, J., Moragrega, A., Fernndez-Prades, C., & Closas, P. (2017). Low-cost GNSS/INS/-Odometric Sensor Fusion Platform for Ground Intelligent Transportation Systems. *Proceedings of the 30th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2017)*, 436–455. https://doi.org/10.33012/2017.15200

Azaola Sàenz, M. (2009). European Patent: EP2113786A1 – Method for autonomous determination of protection levels for GNSS positioning based on navigation residuals and an isotropic confidence ratio [Published 4 November 2009].

Bahle, J. (2020). *Qualitätsbeschreibung der Ausrichtungsschätzung und Vergleich der Beobachtungen eines 2-Antennen-RTK-Systems für eine Anwendung im hochautomatisierten Fahren.* [Bachelor's thesis supervised by Stefan Leinen and Grischa Gottschalg. Technical University of Darmstadt].

Bhatti, U. I. (2007). *Improved integrity algorithms for integrated GPS/INS systems in the presence of slowly growing errors* (Doctoral dissertation). Imperial College London.

Blanch, J., Walker, T., Enge, P., Lee, Y., Pervan, B., Rippl, M., Spletter, A., & Kropp, V. (2015). Baseline advanced RAIM user algorithm and possible improvements. *IEEE Transactions on Aerospace and Electronic Systems*, *51*(1), 713–732. https://doi.org/10.1109/TAES.2014.130739

Blanch, J., Walter, T., & Enge, P. (2005). Protection level calculation using measurement residuals: Theory and results. *Proceedings of the 18th International Technical Meet-*

ing of the Satellite Division of The Institute of Navigation (ION GNSS 2005), 2288–2296.

Blanch, J., Walter, T., & Enge, P. (2019). Gaussian bounds of sample distributions for integrity analysis. *IEEE Transactions on Aerospace and Electronic Systems*, *55*(4), 1806–1815. https://doi.org/10.1109/TAES.2018.2876583

Blanch, J., Walter, T., Enge, P., Burns, J., Mabilleau, M., Martini, I., Boyero, J. P., & Berz, G. (2018). A Proposed Concept of Operations for Advanced Receiver Autonomous Integrity Monitoring. *Proceedings of the 31st International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2018), Miami, Florida, 24–28 September*, 1084–1090. https://doi.org/10.33012/2018.15934

Blank, S., Berns, K., & Föhst, T. (2010). A case study towards evaluation of redundant multi-sensor data fusion. *Commercial Vehicle Technology 2010, CVT, Commercial Vehicle Technology Symposium, Aachen, Germany*, 475–484. https://www.tib.eu/de/suchen/id/tema%3ATEMA20100906219

Bosch Sensortec GmbH. (2020). Bosch BMI160 Data Sheet [Accessed on 8 February 2022]. https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmi160-ds000.pdf

Carlson, N. A., & Berarducci, M. P. (1994). Federated Kalman Filter Simulation Results. *Navigation*, *41*(3), 297–322. https://doi.org/10.1002/j.2161-4296.1994.tb01882.x

Eberle, U., Hallerbach, S., Mannale, R., Kramer, B., Neurohr, C., Steimle, M., & Amersbach, C. (2019). The Pegasus Method for Safeguarding Automated Driving: What else is needed? *PEGASUS Final Event and Symposium*. https://doi.org/10.13140/RG.2.2.15920.20485

El–Mowafy, A., & Kubo, N. (2018). Integrity monitoring for Positioning of intelligent transport systems using integrated RTK–GNSS, IMU and vehicle odometer. *IET Intelligent Transport Systems*, *12*(8), 901–908. https://doi.org/10.1049/iet-its.2018.0106

ESA Navipedia. (2006). The Stanford - ESA Integrity Diagram: Focusing on SBAS Integrity [Accessed on 5 July 2021]. https://gssc.esa.int/navipedia/index.php/The_Stanford_%E2%80%93_ESA_Integrity_Diagram:_Focusing_on_SBAS_Integrity

ESA Navipedia. (2011). Integrity [Accessed on 11 October 2019]. https://gssc.esa.int/navipedia/index.php/Integrity

ESA Navipedia. (2014). GNSS signal [Accessed on 15 February 2022]. https://gssc.esa.int/navipedia/index.php/GNSS_signal

European GSA. (2019). Report on Road User Needs and Requirements [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Report_on_User_Needs_and_Requirements_Road.pdf, revision 2.0, accessed on 8 October 2019].

Everett, T. (2020). RTKLIB ver.2.4.3 demo5-b33c [http://rtkexplorer.com/download/demo5-b33c-binaries/, accessed on 6 August 2020].

Gakne, P. V., & O'Keefe, K. (2018). Tightly-Coupled GNSS/Vision Using a Sky-Pointing Camera for Vehicle Navigation in Urban Areas. *Sensors (Basel, Switzerland)*, *18*(4). https://doi.org/10.3390/s18041244

Groves, P. D. (2013). *Principles of GNSS, inertial, and multisensor integrated navigation systems* (Second edition). Artech House.

Guan, B., Tang, X., & Ge, Q. (2012). Federated cubature kalman filter for multi-sensor information fusion. *Proceedings of 2012 International Conference on Computer Technology and Science (ICCTS 2012), New Delhi, India*.

Gunning, K., Blanch, J., Walter, T., de Groot, L., & Norman, L. (2018). Design and Evaluation of Integrity Algorithms for PPP in Kinematic Applications. *Proceedings of the 31st International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2018), Miami, Florida, 24–28 September*, 1910–1939. https://doi.org/10.33012/2018.15972

Gunning, K., Blanch, J., Walter, T., de Groot, L., & Norman, L. (2019). Integrity for Tightly Coupled PPP and IMU. *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019), Miami, Florida, 16–20 September*, 3066–3078. https://doi.org/10.33012/2019.17011

Guo, H., Cao, D., Chen, H., Lv, C., Wang, H., & Yang, S. (2018). Vehicle dynamic state estimation: State of the art schemes and perspectives. *IEEE/CAA Journal of Automatica Sinica*, *5*(2), 418–431. https://doi.org/10.1109/JAS.2017.7510811

Gupta, S., & Gao, G. X. (2021). Data-Driven Protection Levels for Camera and 3D Map-based Safe Urban Localization. *CoRR*, *abs/2101.06379*. https://arxiv.org/abs/2101.06379

Gupta, V. (2009). *Vehicle localization using low-accuracy GPS, IMU and map-aided vision* (Doctoral dissertation). The Pennsylvania State University. https://etda.libraries.psu.edu/files/final_submissions/6442

Hassan, T., El–Mowafy, A., & Wang, K. (2020). A review of system integration and current integrity monitoring methods for positioning in intelligent transport systems. *IET Intelligent Transport Systems*, *15*(1), 43–60. https://doi.org/10.1049/itr2.12003

Hewitson, S., & Wang, J. (2010). Extended Receiver Autonomous Integrity Monitoring (eRAIM) for GNSS/INS Integration. *Journal of Surveying Engineering*, *136*(1), 13–22. https://doi.org/10.1061/(ASCE)0733-9453(2010)136:1(13)

Humphreys, T. E., Murrian, M., & Narula, L. (2018). Low-cost precise vehicular positioning in urban environments. *Proceedings of the 2018 IEEE/ION Position, Location and Navigation Symposium (PLANS), Monterey, CA, 23–26 April*, 456–471. https://doi.org/10.1109/PLANS.2018.8373414

Ilyas, M., JunKyu Lim, Jang Gyu Lee, & Chan Gook Park. (2008). Federated unscented kalman filter design for multiple satellites formation flying in leo. *2008 International Conference on Control, Automation and Systems*, 453–458. https://doi.org/10.1109/ICCAS.2008.4694683

iMAR Navigation GmbH. (2019). iNAT-4C/SLx - Data Sheet [Version of 4 July 2019]. https://imar-navigation.de/downloads/NAT-4C-SLx.pdf

Jia, M., Lee, H., Khalife, J., Kassas, Z. M., & Seo, J. (2021). Ground Vehicle Navigation Integrity Monitoring for Multi-Constellation GNSS Fused with Cellular Signals of Opportunity. *2021 IEEE International Intelligent Transportation Systems Conference (ITSC), 19–22 September*, 3978–3983. https://doi.org/10.1109/ITSC48978.2021.9564686

Jin, X., Yin, G., & Chen, N. (2019). Advanced estimation techniques for vehicle system dynamic state: A survey. *Sensors*, *19*(19). https://doi.org/10.3390/s19194289

Jurado, J., Raquet, J., Schubert Kabban, C. M., & Gipson, J. (2020). Residual-based multi-filter methodology for all-source fault detection, exclusion, and performance monitoring. *NAVIGATION*, *67*(3), 493–510. https://doi.org/https://doi.org/10.1002/navi.384

Kampmann, A., Lamberti, M., Petrovic, N., Kowalewski, S., & Alrifaee, B. (2021). Infrared beacons for robust localization. *CoRR*, *abs/2104.09335*. https://arxiv.org/abs/2104.09335

Kocic, J., Jovicic, N., & Drndarevic, V. (2018). Sensors and Sensor Fusion in Autonomous Vehicles. *2018 26th Telecommunications Forum (TELFOR), 20–21 November*, 420–425. https://doi.org/10.1109/TELFOR.2018.8612054

Kouba, J., Lahaye, F., & Tétreault, P. (2017). Precise Point Positioning. In O. Montenbruck & P. J. Teunissen (Eds.), *Springer Handbook of Global Navigation Satellite Systems* (pp. 723–751). Springer. https://doi.org/10.1007/978-3-319-42928-1_25

Kumar, M., Garg, D. P., & Zachery, R. A. (2006). A generalized approach for inconsistency detection in data fusion from multiple sensors. *American Control Conference, 2006*. https://doi.org/10.1109/ACC.2006.1656526

Kuusniemi, H. (2005). *User-level reliability and quality monitoring in satellite-based personal navigation* (Doctoral dissertation). Tampere University of Technology.

Lan, T., Geffert, A., Dodinoiu, A., & Becker, U. (2020). Kalman filtering versus voting: Which strategy is best for multi-sensor localization? *2020 European Navigation Conference (ENC)*, 1–9. https://doi.org/10.23919/ENC48637.2020.9317318

Langley, R. B., Teunissen, P. J., & Montenbruck, O. (2017). Introduction to GNSS. In O. Montenbruck & P. J. Teunissen (Eds.), *Springer handbook of global navigation satellite systems* (pp. 3–23). Springer. https://doi.org/10.1007/978-3-319-42928-1_1

Lee, M.-j., & Ha, Y.-g. (2020). Autonomous driving control using end-to-end deep learning. *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 470–473. https://doi.org/10.1109/BigComp48618.2020.00-23

Li, X., Li, X., Huang, J., Shen, Z., Wang, B., Yuan, Y., & Zhang, K. (2021). Improving PPP–RTK in urban environment by tightly coupled integration of GNSS and INS. *Journal of Geodesy*, *95*(12), 3949. https://doi.org/10.1007/s00190-021-01578-6

Liu, H., Zheng, G., Wang, H., & Feng, C. (2010). Research on integrity monitoring for integrated GNSS/SINS system. *IEEE International Conference on Information and Automation (ICIA), 2010*, 1990–1995. https://doi.org/10.1109/ICINFA.2010.5512033

Meng, X., Wang, H., & Liu, B. (2017). A Robust Vehicle Localization Approach Based on GNSS/IMU/DMI/LiDAR Sensor Fusion for Autonomous Vehicles. *Sensors (Basel, Switzerland)*, *17*(9). https://doi.org/10.3390/s17092140

Mokhtarian, A., Alrifaee, B., & Kampmann, A. (2020). The Dynamic Service-oriented Software Architecture for the UNICARagil Project. *29. Aachen Colloquium Sustainable Mobility, Aachen, Germany, 5–7 October*. https://doi.org/10.18154/RWTH-2020-11256

Moritz, H. (2000). Geodetic reference system 1980. *Journal of Geodesy*, *74*(1), 128–133. https://doi.org/10.1007/s001900050278

Navarro Madrid, P. F., Saenz, M. A., Varo, C. M., & Schortmann, J. C. (2015). Computing Meaningful Integrity Bounds of a Low-cost Kalman-filtered Navigation Solution in Urban Environments. *Proceedings of the 28th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2015), Tampa, Florida, 14–18 September*, 2914–2925.

Navarro Madrid, P. F. (2016). European Patent: EP3009860A1 – Method for computing an error bound of a Kalman filter based GNSS position solution [Published 20 April 2016].

Niemeier, W. (2008). *Ausgleichungsrechnung: Statistische Auswertemethoden* (2., überarb. und erw. Aufl.). de Gruyter. https://doi.org/10.1515/9783110206784

Odijk, D., & Wanninger, L. (2017). Differential Positioning. In O. Montenbruck & P. J. Teunissen (Eds.), *Springer handbook of global navigation satellite systems* (pp. 753–780). Springer. https://doi.org/10.1007/978-3-319-42928-1_26

Oliveira, J., & Tiberius, C. (2009). Quality Control in SBAS: Protection Levels and Reliability Levels. *Journal of Navigation*, *62*(3), 509–522. https://doi.org/10.1017/S0373463309005311

Parhami, B. (2005). Voting: A paradigm for adjudication and datafusion in dependable systems. In H. B. Diab & A. Y. Zomaya (Eds.), *Dependable Computing Systems: Paradigms, Performance Issues, & Applications* (pp. 87–114). Wiley.

Parhami, B. (1996). A taxonomy of voting schemes for data fusion and dependable computation. *Reliability Engineering & System Safety*, *52*(2), 139–151. https://doi.org/https://doi.org/10.1016/0951-8320(96)00012-9

Pullen, S. (2008). Quantifying the performance of navigation systems and standards for assisted-GNSS [Accessed on 9 Februray 2022]. *insideGNSS Sep./Oct. 2008*. https://insidegnss-com.exactdn.com/wp-content/uploads/2018/01/sepoct08-gnsssolutions.pdf

Pullen, S., Kilfeather, J., Goddard, J., Nowitzky, T., Shah, B., Doong, W., Welton, A., Kagan, D., & Greer, K. (2018). Enhanced Navigation, Robustness, and Safety Assurance for Autonomous Vehicles as Part of the Globalstar Connected Car Program. *Proceedings of the 31st International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2018), Miami, Florida, 24–28 September*, 1538–1565. https://doi.org/10.33012/2018.16106

Reid, T. G. R., Pervez, N., Ibrahim, U., Houts, S. E., Pandey, G., Alla, N. K., & Hsia, A. (2019). Standalone and RTK GNSS on 30,000 km of North American Highways. *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, 2135–2158. https://doi.org/10.33012/2019.16914

Reid, T. G., Houts, S. E., Cammarata, R., Mills, G., Agarwal, S., Vora, A., & Pandey, G. (2019). Localization Requirements for Autonomous Vehicles. *SAE International Journal of Connected and Automated Vehicles*, *2*(3). https://doi.org/10.4271/12-02-03-0012

Reuper, B. (2020). Multi-Frequency GNSS Sensor Fusion With Quality Assessment for Automotive Applications. In *Schriftenreihe der Fachrichtung Geodäsie*. Fachbereich Bau- und Umweltingenieurwissenschaften, Technische Universität Darmstadt. https://doi.org/10.25534/tuprints-00011655

Rife, J., Walter, T., & Blanch, J. (2004). Overbounding SBAS and GBAS Error Distributions with Excess-Mass Functions. *Proceedings of the 2004 International Symposium on GNSS/GPS*. http://web.stanford.edu/group/scpnt/gpslab/pubs/papers/rife_ignss_2004.pdf

Schwarz, W., & Hennes, M. (2016). Qualitätsbewertungen in der Ingenieurgeodäsie. In W. Freeden & R. Rummel (Eds.), *Handbuch der Geodäsie: 6 Bände* (pp. 1–32). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-46900-2_31-1

Sciacchitano, A., & Wieneke, B. (2016). PIV uncertainty propagation. *Measurement Science and Technology*, *27*(8), 084006. https://doi.org/10.1088/0957-0233/27/8/084006

Sensonor AS. (2013). Sensonor STIM300 Data Sheet [Accessed on 21 December 2018]. https://www.sensonor.com/media/1132/ts1524r9-datasheet-stim300.pdf

Steinhardt, N. (2014). *Eine Architektur zur Schätzung kinematischer Fahrzeuggrössen mit integrierter Qualitätsbewertung durch Sensordatenfusion* (Vol. Nr. 781). VDI Verlag.

Steinhardt, N., & Leinen, S. (2015). Datenfusion für die präzise Lokalisierung. In H. Winner, S. Hakuli, F. Lotz, & C. Singer (Eds.), *Handbuch Fahrerassistenzsysteme* (3., überarb. und erg. Aufl., pp. 481–512). Springer Vieweg. https://doi.org/10.1007/978-3-658-05734-3

Takanose, A., Takikawa, K., Arakawa, T., & Meguro, J. (2020). Improvement of RTK-GNSS with Low-Cost Sensors Based on Accurate Vehicle Motion Estimation Using GNSS Doppler. *2020 IEEE Intelligent Vehicles Symposium (IV)*, 658–665. https://doi.org/10.1109/IV47402.2020.9304539

Takasu, T. (2013). RTKLIB ver. 2.4.2 Manual [Version of 29 April 2013]. http://www.rtklib.com/prog/manual_2.4.2.pdf

Takasu, T. (2019). RTKLIB ver.2.4.3 b33 [Accessed on 17 December 2020]. https://github.com/tomojitakasu/RTKLIB/tree/rtklib_2.4.3

Tanil, C., Khanafseh, S., Joerger, M., & Pervan, B. (2018). Sequential Integrity Monitoring for Kalman Filter Innovations-based Detectors. *Proceedings of the 31st International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2018), Miami, Florida, 24–28 September*, 2440–2455. https://doi.org/10.33012/2018.15975

The Mathworks, Inc. (2021). Equation Solving Algorithms [Accessed on 28 September 2021]. https://www.mathworks.com/help/optim/ug/equation-solving-algorithms.html

The Mathworks, Inc. (2022). Documation of the function scatteredInterpolant [Accessed on 11 March 2022]. https://www.mathworks.com/help/matlab/ref/scatteredinterpolant.html

Tijero, E. D., Fernndez, L. M., Zarzosa, J. I. H., Garca, J., Ibaez-Guzmn, J., Stawiarski, E., Xu, P., Avellone, G., Pisoni, F., Falletti, E., & Ortiz, M. (2018). High Accuracy Positioning Engine with an Integrity Layer for Safety Autonomous Vehicles. *Proceedings of the 31st International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2018), Miami, Floria, 24–28 September*, 1566–1572. https://doi.org/10.33012/2018.15843

Tracey, B. D., & Wolpert, D. (2018). Upgrading from Gaussian Processes to Student's-T Processes. *2018 AIAA Non-Deterministic Approaches Conference*. https://doi.org/10.2514/6.2018-1659

Tu, X., Gu, D., Yi, D., & Zhou, H. (2011). Evaluation of GNSS Receiver Autonomous Integrity Monitoring for multiple outliers with a smart random sample consensus strategy. *2011 19th International Conference on Geoinformatics, 24–26 June*, 1–6. https://doi.org/10.1109/GeoInformatics.2011.5981066

Ulbrich, S., Reschka, A., Rieken, J., Ernst, S., Bagschik, G., Dierkes, F., Nolte, M., & Maurer, M. (2017). Towards a functional system architecture for automated vehicles. *CoRR*, *abs/1703.08557*. http://arxiv.org/abs/1703.08557

Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M. N., Dolan, J., Duggins, D., Galatali, T., Geyer, C., Gittleman, M., Harbaugh, S., Hebert, M., Howard, T. M., Kolski, S., Kelly, A., Likhachev, M., McNaughton, M., Miller, N., Peterson, K., Pilnick, B., Rajkumar, R., Rybski, P., Salesky, B., … Ferguson, D. (2008). Autonomous driving in urban environments: Boss and the Urban Challenge. *Journal of Field Robotics*, *25*(8), 425–466. https://doi.org/10.1002/rob.20255

Wan, G., Yang, X., Cai, R., Li, H., Zhou, Y., Wang, H., & Song, S. (2018). Robust and Precise Vehicle Localization Based on Multi-Sensor Fusion in Diverse City Scenes. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 4670–4677. https://doi.org/10.1109/ICRA.2018.8461224

Wang, L., Zhang, Y., & Wang, J. (2017). Map-Based Localization Method for Autonomous Vehicles Using 3D-LIDAR. *IFAC-PapersOnLine*, *50*(1), 276–281. https://doi.org/10.1016/j.ifacol.2017.08.046

Watzenig, D., & Horn, M. (2017). Introduction to Automated Driving. In *Automated Driving* (pp. 3–16). Springer International Publishing. https://doi.org/10.1007/978-3-319-31895-0

Welte, A. (2017). *Protection Levels for High Integrity Localization for Autonomous Driving* (M.S. thesis). ENSTA Bretagne, Univ. Angers. https://www.hds.utc.fr/~bonnif/supervision/Report_2017_Anthony_Welte_Final.pdf.

Wendel, J. (2007). *Integrierte Navigationssysteme*. Oldenbourg Verlag.

Woopen, T., Lampe, B., Böddeker, T., Eckstein, L., Kampmann, A., Alrifaee, B., Kowalewski, S., Moormann, D., Stolte, T., Jatzkowski, I., Maurer, M., Möstl, M., Ernst, R., Ackermann, S., Amersbach, C., Winner, H., Püllen, D., Katzenbeisser, S., Leinen, S., Becker, M., Stiller, C., Furmans, K., Bengler, K., Diermeyer, F., … Hecker, C. (2018). UNICARagil - Disruptive modulare Architektur für agile, automatisierte Fahrzeugkonzepte. *27th Aachen Colloquium Automobile and Engine Technology, Aachen, Germany, 8–10 October*. https://doi.org/10.18154/RWTH-2018-229909

Woopen, T., van Kempen, R., Böddeker, T., & Eckstein, L. (2020). UNICARagil - Where We Are and Where We Are Going. *29. Aachen Colloquium Sustainable Mobility, Aachen, Germany, 5–7 October*, 285–308. https://publications.rwth-aachen.de/record/804601

Wymeersch, H., Seco-Granados, G., Destino, G., Dardari, D., & Tufvesson, F. (2017). 5G mmWave Positioning for Vehicular Networks. *IEEE Wireless Communications*, *24*(6), 80–86. https://doi.org/10.1109/MWC.2017.1600374

Yang, Y., & Xu, J. (2016). GNSS receiver autonomous integrity monitoring (RAIM) algorithm based on robust estimation. *Geodesy and Geodynamics*, *7*(2), 117–123. https://doi.org/10.1016/j.geog.2016.04.004

Yurtsever, E., Lambert, J., Carballo, A., & Takeda, K. (2020). A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, *8*, 58443–58469. https://doi.org/10.1109/ACCESS.2020.2983149

Zabalegui, P., De Miguel, G., Perez, A., Mendizabal, J., Goya, J., & Adin, I. (2020). A Review of the Evolution of the Integrity Methods Applied in GNSS. *IEEE Access*, *8*, 45813–45824. https://doi.org/10.1109/ACCESS.2020.2977455

Zhao, Y. (2015). *Optimized Filter Design for Non-Differential GPS/IMU Integrated Navigation* (Doctoral dissertation). Technische Universität. Darmstadt. http://tuprints.ulb.tu-darmstadt.de/4929/

Zhu, N., Marais, J., Betaille, D., & Berbineau, M. (2018). GNSS Position Integrity in Urban Environments: A Review of Literature. *IEEE Transactions on Intelligent Transportation Systems*, *19*(9), 2762–2778. https://doi.org/10.1109/TITS.2017.2766768

Zhu, S., Li, S. H., Liu, Y., & Fu, Q. W. (2019). Low-Cost MEMS-IMU/RTK Tightly Coupled Vehicle Navigation System with Robust Lane-Level Position Accuracy. *2019 26th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*, 1–4. https://doi.org/10.23919/ICINS.2019.8769351

Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., Dang, T., Franke, U., Appenrodt, N., Keller, C. G., Kaus, E., Herrtwich, R. G., Rabe, C., Pfeiffer, D., Lindner, F., Stein, F., Erbs, F., Enzweiler, M., Knoppel, C., Hipp, J., Haueis, M., Trepte, M., Brenk, C., Tamke, A., … Zeeb, E. (2014). Making Bertha Drive—An Autonomous Journey on a Historic Route. *IEEE Intelligent Transportation Systems Magazine*, *6*(2), 8–20. https://doi.org/10.1109/MITS.2014.2306552

Own Publications

Buchholz, M., Gies, F., Danzer, A., Henning, M., Hermann, C., Herzog, M., Horn, M., Schön, M., Rexin, N., Dietmayer, K., Fernandez, C., Janosovits, J., Kamran, D., Kinzig, C., Lauer, M., Molinos, E., Stiller, C., Ackermann, S., Homolla, T., Winner, H., Gottschalg, G., Leinen, S., Becker, M., Feiler, J., … Siepenkötter, N. (2020). Automation of the UNICARagil vehicles. *29. Aachen Colloquium Sustainable Mobility, Aachen, Germany, 5–7 October*, 1531–1560. https://doi.org/10.18725/OPARU-34024

Gottschalg, G., Becker, M., & Leinen, S. (2020). Integrity Concept for Sensor Fusion Algorithms used in a Prototype Vehicle for Automated Driving. *2020 European Navigation Conference (ENC), Dresden, Germany, 23–24 November*. https://doi.org/10.23919/ENC48637.2020.9317323

Gottschalg, G., Becker, M., & Leinen, S. (2021). Integrity Based Data Fusion of Redundant Fusion Filters for Vehicle Dynamic State Estimation in Automated Driving. *2021 IEEE International Intelligent Transportation Systems Conference (ITSC), 19–22 September*, 3759–3765. https://doi.org/10.1109/ITSC48978.2021.9564417

Gottschalg, G., & Leinen, S. (2021). Comparison and Evaluation of Integrity Algorithms for Vehicle Dynamic State Estimation in Different Scenarios for an Application in Automated Driving. *Sensors*, *21*(4). https://doi.org/10.3390/s21041458

Homolla, T., Gottschalg, G., & Winner, H. (2020). Verfahren zur Korrektur von inkonsistenten Lokalisierungsdaten in modularen technischen Systemen. *13. Workshop Fahrerassistenzsysteme und automatisiertes Fahren, Walting, Germany, 1–3 April*. http://tubiblio.ulb.tu-darmstadt.de/120962/

Winner, H., Gottschalg, G., & Homolla, T. (2021). German Patent Pending: DE102020107899A1 – Vorrichtung zur Korrektur von Abweichungen in Lokalisierungsinformationen einer Planungsebene und einer Ausführungsebene [Published 23 September 2021].

Appendices

A. Additional Material Regarding the Background Chapter

A.1. Extract from IMU Data Sheets

Table A.1.: Key characteristics of used IMU's from their data sheets (Analog Devices Inc., 2018; Bosch Sensortec GmbH, 2020; Sensonor AS, 2013), converted into same units for easier comparison

	Sensonor	Analog Devices	Bosch Sensortec
	STIM300	ADIS 16465-1	BMI160
Gyroscope Range	±400 °/s	±125 °/s	±250 °/s
Gyroscope Bias Range	±250 °/h	±500 °/h	±10 800 °/h
Angular Random Walk	0.15 °/$\sqrt{\text{h}}$	0.15 °/$\sqrt{\text{h}}$	0.42 °/$\sqrt{\text{h}}$
Gyroscope In-Run Bias Variation	0.5 °/h	2 °/h	252 °/h
Accelerometer Range	±10 g	±8 g	±8 g
Accelerometer Turn-On Bias	±0.75 mg	±0.5 mg	±25 mg
Velocity Random Walk	0.06 m/s/$\sqrt{\text{h}}$	0.012 m/s/$\sqrt{\text{h}}$	0.106 m/s/$\sqrt{\text{h}}$
Accelerometer In-Run Bias Variation	0.05 mg	0.0036 mg	1.8 mg

A.2. Wheel Slip Definition

With the wheel's velocity components with respect to the ECEF frame along the x^w- and y^w-axis, $v^w_{ew,x}$ and $v^w_{ew,y}$, the wheel slip definitions are given as (Reuper, 2020, p. 15):

$$\lambda_x = \begin{cases} \dfrac{\omega_w r_d - v^w_{ew,x}}{v^w_{ew,x}} & \text{for } \omega_w r_d \leq v^w_{ew,x} \quad \text{(braking)} \\ \dfrac{\omega_w r_d - v^w_{ew,x}}{\omega_w r_d} & \text{for } \omega_w r_d > v^w_{ew,x} \quad \text{(acceleration)} \end{cases} \tag{A.1}$$

$$\alpha = -\arctan \frac{v^w_{ew,y}}{v^w_{ew,x}}. \tag{A.2}$$

For the special case of vehicle's standstill, meaning $\omega_w = v^w_{ew,x} = v^w_{ew,y} = 0$, the slip angles are set to $\lambda_x = \alpha = 0$ (Reuper, 2020, p. 15).

In linear tire models, the longitudinal slip λ_x and the lateral slip angle α are modeled as proportional to the friction coefficients μ_x and μ_y

$$\mu_x = c_\lambda \lambda_x \tag{A.3}$$

$$\mu_y = c_\alpha \alpha \tag{A.4}$$

with the longitudinal slip stiffness c_λ and the lateral slip stiffness c_α as proportionality constants and the definition of the friction coefficients

$$\mu_x = \frac{F^w_{ew,x}}{F^w_{ew,z}} \tag{A.5}$$

$$\mu_y = \frac{F^w_{ew,y}}{F^w_{ew,z}} \tag{A.6}$$

using the longitudinal, lateral tire force $F^w_{ew,x}$, $F^w_{ew,y}$ and the tire force normal to the road surface $F^w_{ew,z}$ (Reuper, 2020, p. 15).

In order to facilitate the usage of the longitudinal slip and to avoid the distinction of cases, the longitudinal tire slip correction factor κ is introduced (Reuper, 2020, p. 54):

$$\kappa_x = \begin{cases} \dfrac{1}{1+\lambda_x} & \text{for } \omega_w r_d \leq v^w_{ew,x} \quad \text{(braking)} \\ 1-\lambda_x & \text{for } \omega_w r_d > v^w_{ew,x} \quad \text{(acceleration).} \end{cases} \tag{A.7}$$

A.3. Quality of the Reference Solution

In this Subsection the quality of the reference solution is analyzed. As mentioned in Section 2.3, the reference solution is based on the observations of a RLG-IMU and a RTK-GNSS receiver. The evaluation is performed in NovAtel WayPoint Inertial Explorer 8.90 using forward and backward processing as well as smoothing to improve the accuracy.

The weakness of this reference solution is its dependence on satellite reception. Even though the RLG-IMU minimizes the drift while there is no GNSS reception, a certain drift is inevitable. Since no external reference with a sufficient accuracy was available, a GNSS outage is simulated to analyze the errors of the reference solution in challenging environments.

For the simulated GNSS outage, a measurement in ideal GNSS reception conditions on the TU Darmstadt airport in Griesheim, Germany recorded on 7 May 2019, with the measurement vehicle described in Section 2.3 is chosen. Two outages are simulated and evaluated separately. The first one lasts $30\,\mathrm{s}$ while the vehicle is driving in a straight line, while the second outage lasts $13\,\mathrm{s}$ while the vehicle is driving a turn of about $90\,°$. The driven velocity for both cases is about $50\,\mathrm{km/h}$, with a slight decrease of velocity in the turn. These situations represent typical scenarios with no GNSS reception in the measurements used for this work.

Figure A.1 and A.2 depict the results for these simulated outages, where the dashed lines mark the start and end of the outage which happens at time zero. In both situations, the horizontal position error amounts to less than $0.06\,\mathrm{m}$. The error in the vertical component is negligible wherefore the two- and three-dimensional error appear almost identical in these Figures. For brevity, the respective graphs for the velocity and attitude are not shown here but these errors stay in both simulated outages below $0.005\,\mathrm{m/s}$ and $0.01\,°$, respectively. In both Figures, the influence of the forward and backward processing is visible, the error reaches its maximum in the middle of the simulated outage (and not at the end as it would be with only forward processing). In the turn, this processing method also leads to the slightly increased error a few seconds before and after the outage.

These GNSS outage simulations aim to analyze the quality of the reference solution in challenging reception conditions, where GNSS outages in this durations appear. In these conditions, the developed fusion filter's $95\,\%$ quantile of the horizontal position error amounts to more than $0.6\,\mathrm{m}$ (see results in Chapter 7). Accordingly, the reference solution's accuracy is one order of magnitude better than the developed data fusion. Therefore, the errors of the reference solution are neglected in the evaluations of this work.

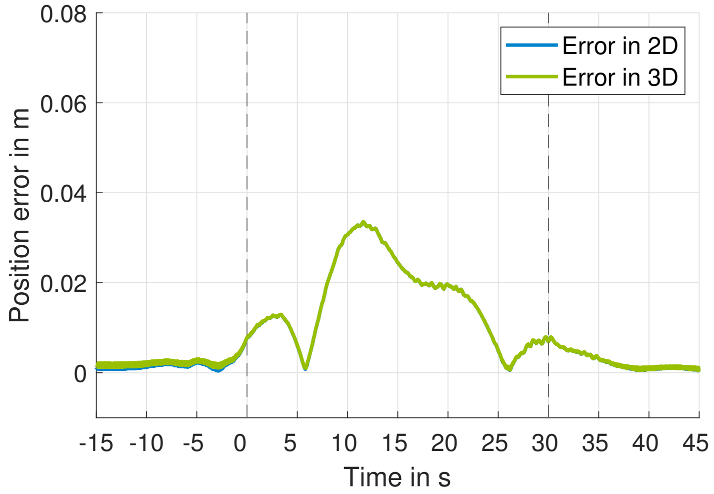


Figure A.1.: Position error of the reference solution during a simulated GNSS outage while driving in a straight line (outage start and end indicated by dashed line, outage at time zero)

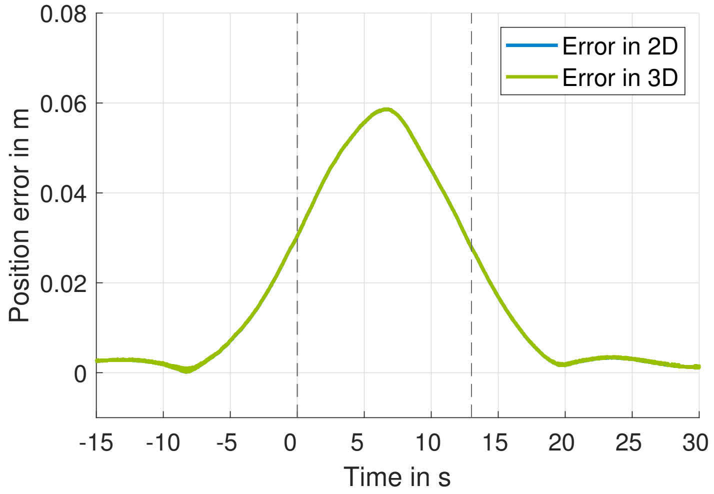


Figure A.2.: Position error of the reference solution during a simulated GNSS outage while driving in a $90\,^\circ$ turn (outage start and end indicated by dashed line, outage at time zero)

A.4. Experiments – Maps of Driven Trajectories

Figures A.3, A.4, A.5, A.6, A.7 and A.8 on the following pages depict the driven trajectories (only the analyzed epochs are shown).

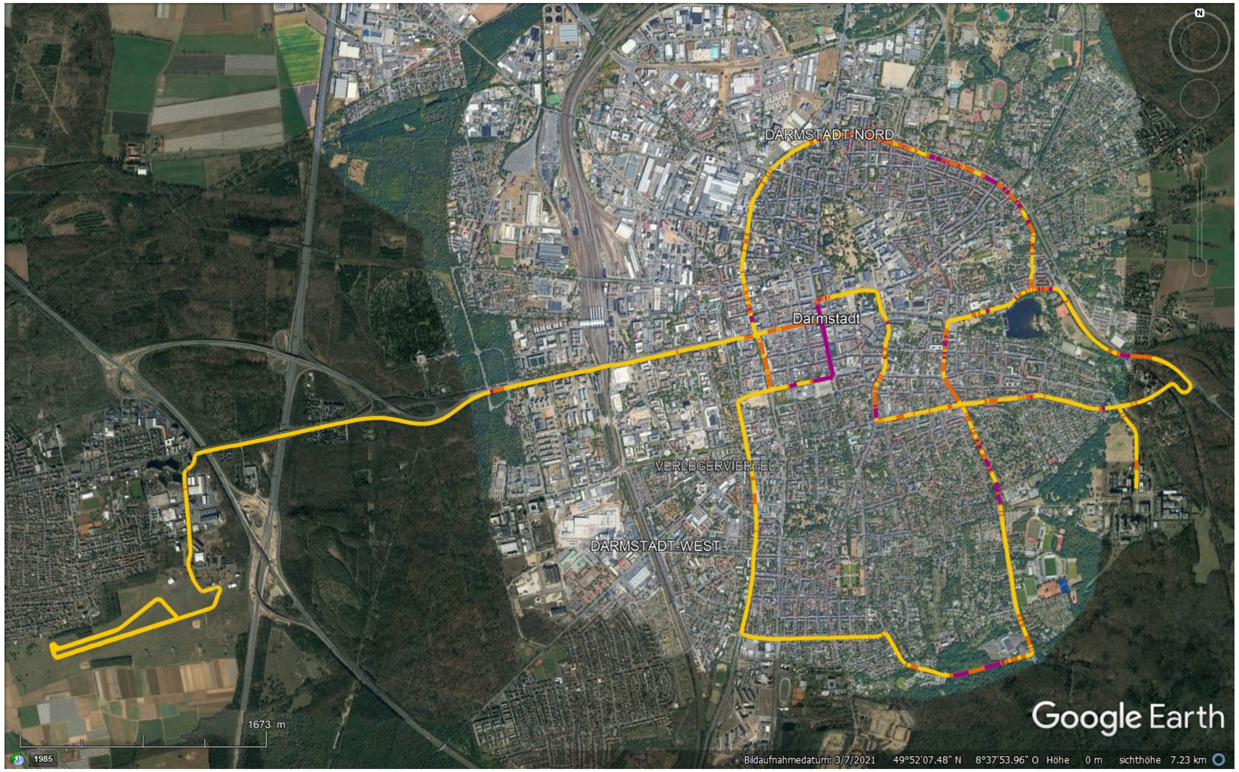

DARMSTADT-NORD
Darmstadt
VERLEGERVIERTEL
DARMSTADT-WEST
Google Earth
1673 m
1985
Bildaufnahmedatum: 3/7/2021 49°52'07.48" N 8°37'53.96" O Höhe 0 m sichthöhe 7.23 km
N

Figure A.3.: Driven trajectories in data set used for parameter tuning, recorded on 7 May 2019 – PDOP indicated by colors: yellow – smaller than 2, orange – smaller than 4, purple greater than 4 (screenshot taken from Google Earth Pro)

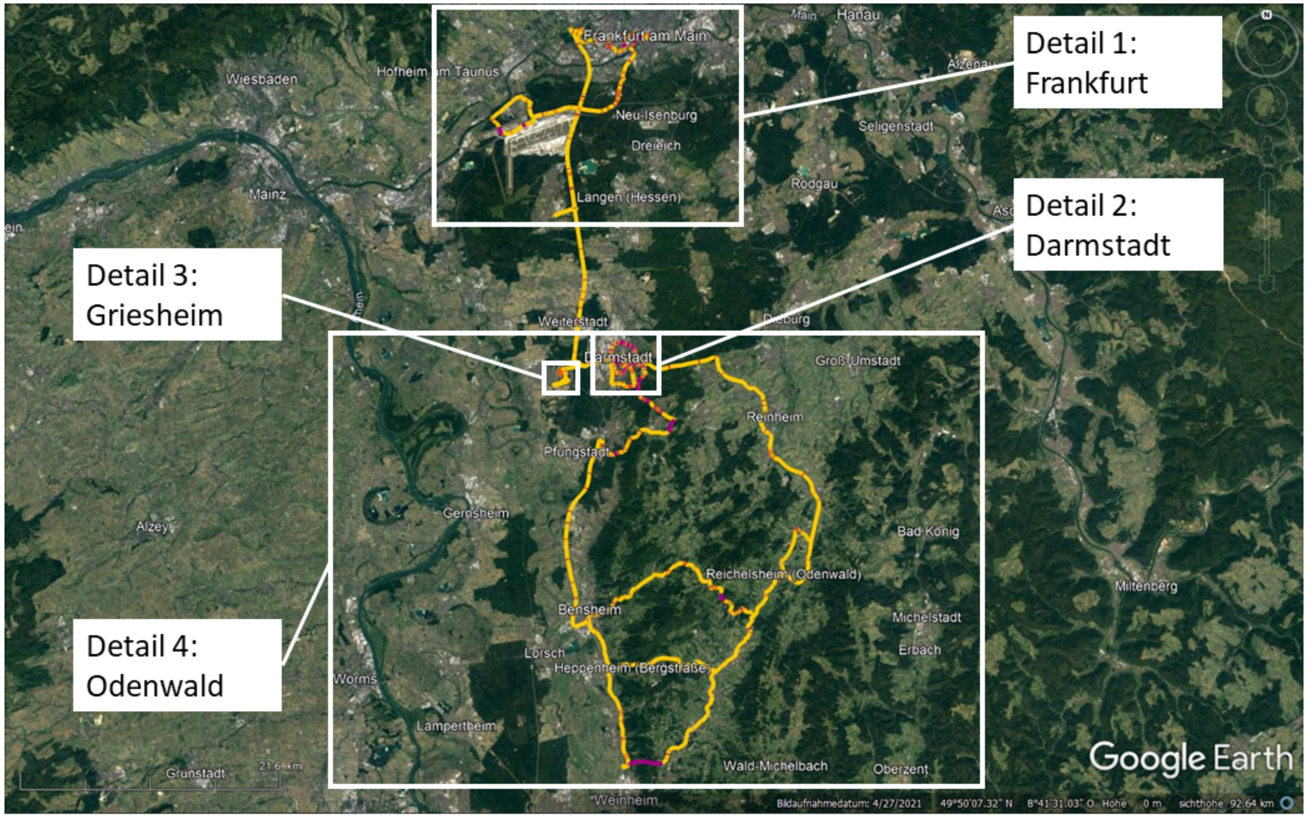


Figure A.4.: Overview – Driven trajectories in extensive data set used for performance evaluation, recorded between May 2019 and May 2021 – PDOP indicated by colors: yellow – smaller than 2, orange – smaller than 4, purple greater than 4 (screenshot taken from Google Earth Pro)

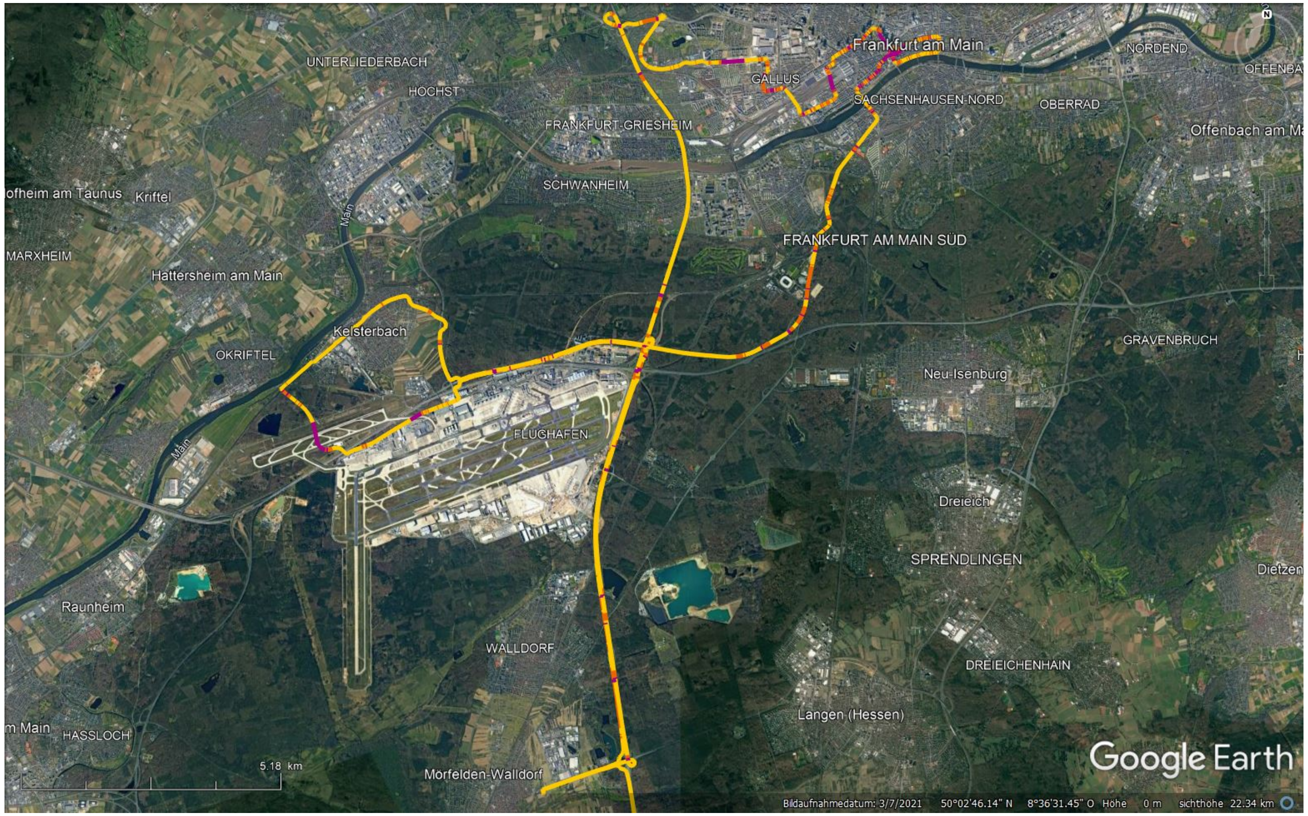


Figure A.5.: Detail 1: Frankfurt – Driven trajectories in extensive data set used for performance evaluation, recorded between May 2019 and May 2021 – PDOP indicated by colors: yellow – smaller than 2, orange – smaller than 4, purple greater than 4 (screenshot taken from Google Earth Pro)

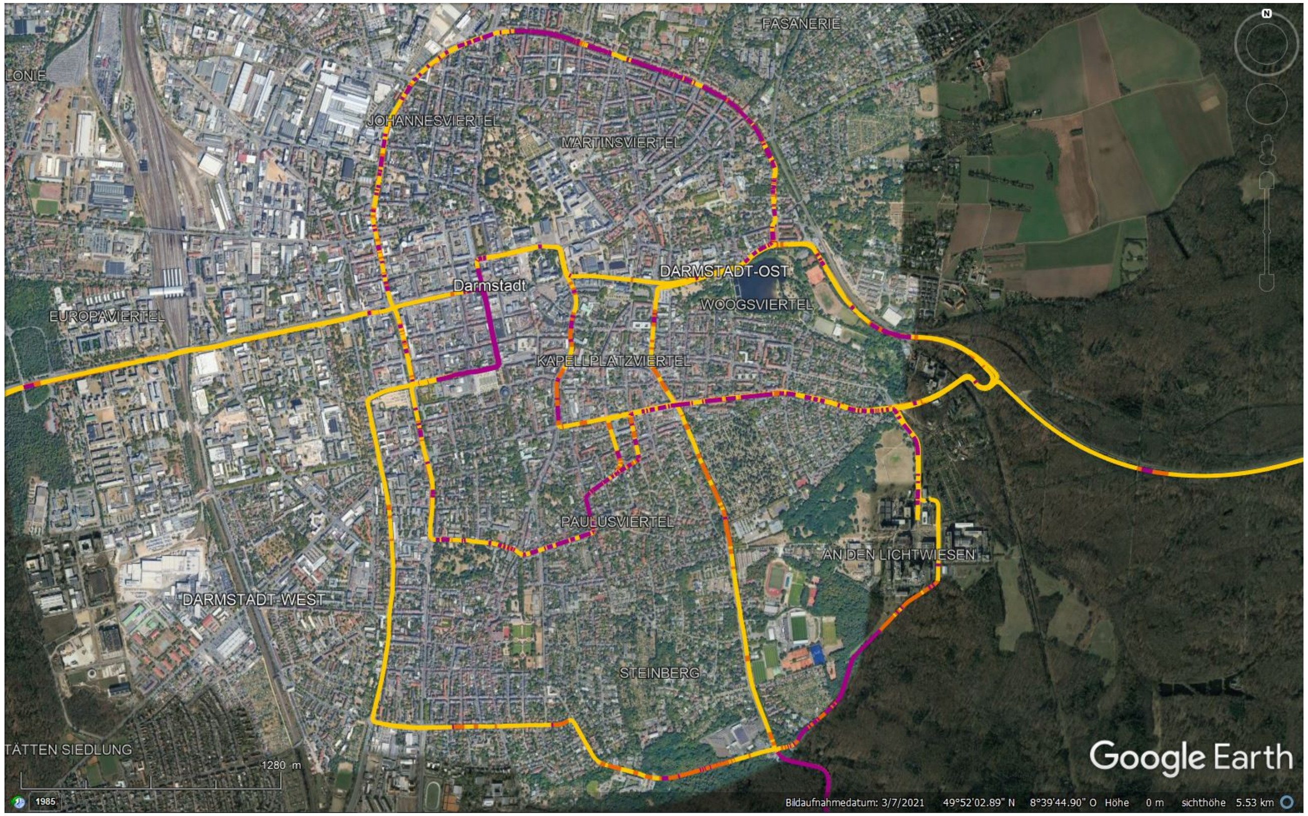


Figure A.6.: Detail 2: Darmstadt – Driven trajectories in extensive data set used for performance evaluation, recorded between May 2019 and May 2021 – PDOP indicated by colors: yellow – smaller than 2, orange – smaller than 4, purple greater than 4 (screenshot taken from Google Earth Pro)

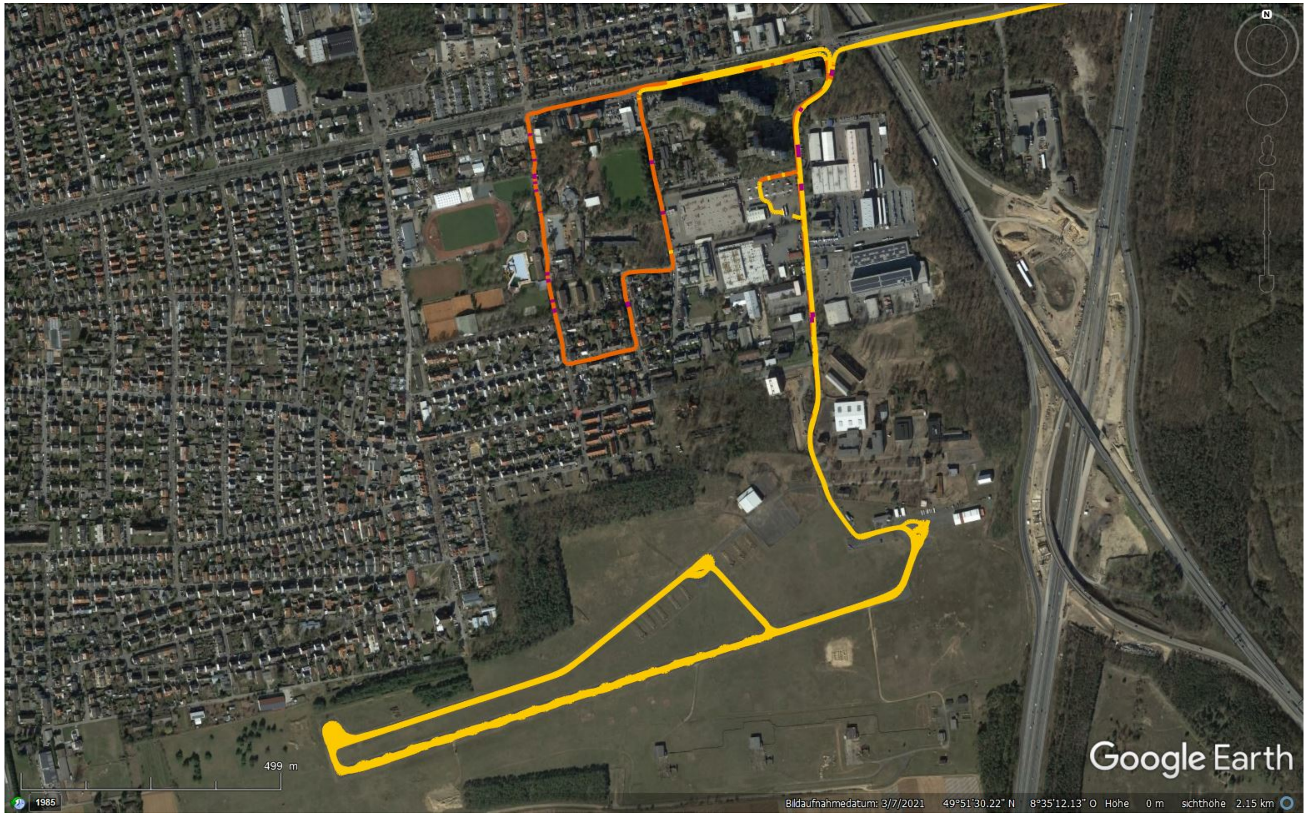

Google Earth
499 m
1985
Bildaufnahmedatum: 3/7/2021 49°51'30.22" N 8°35'12.13" O Höhe 0 m sichthöhe 2.15 km

Figure A.7.: Detail 3: Griesheim – Driven trajectories in extensive data set used for performance evaluation, recorded between May 2019 and May 2021 – PDOP indicated by colors: yellow – smaller than 2, orange – smaller than 4, purple greater than 4 (screenshot taken from Google Earth Pro)

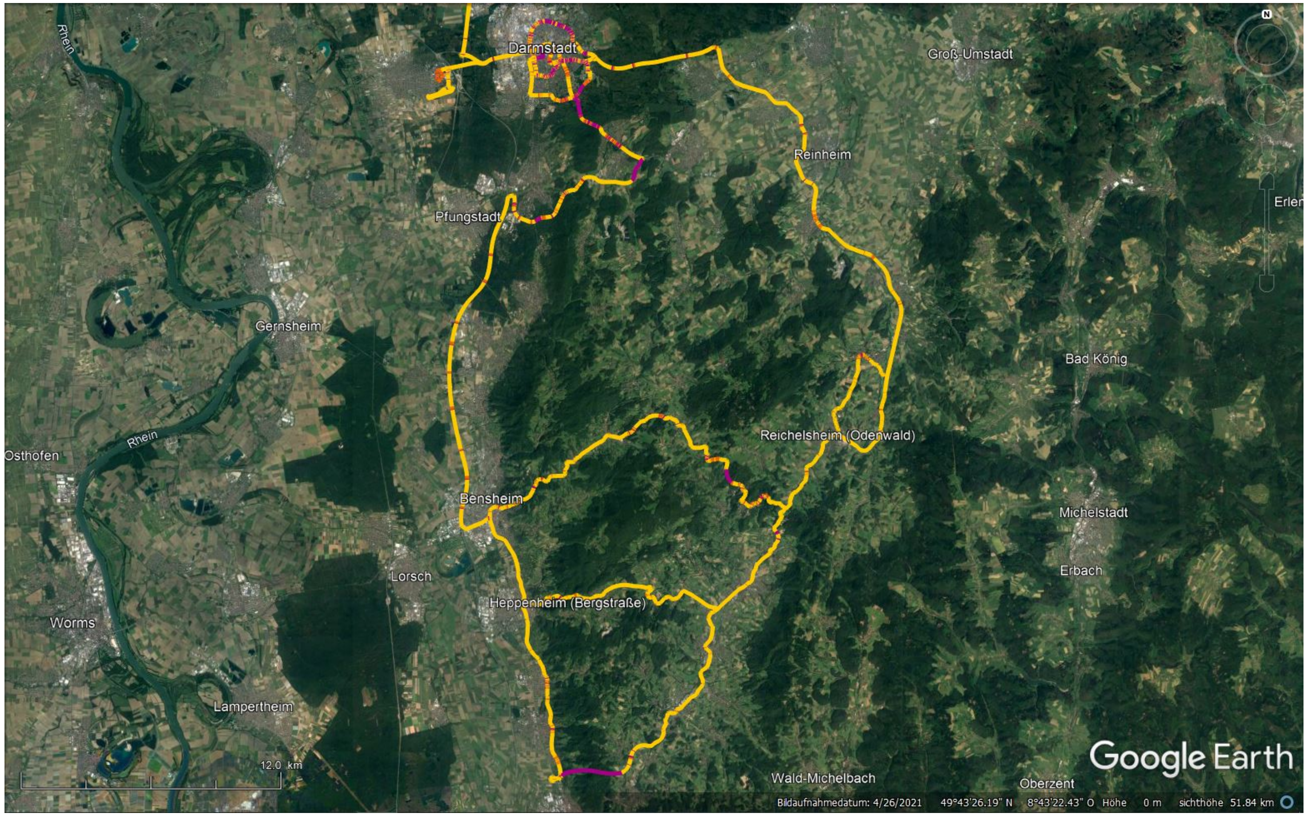


Figure A.8.: Detail 4: Odenwald – Driven trajectories in extensive data set used for performance evaluation, recorded between May 2019 and May 2021 – PDOP indicated by colors: yellow – smaller than 2, orange – smaller than 4, purple greater than 4 (screenshot taken from Google Earth Pro)

B. Additional Explanations Regarding the Fusion Filter

B.1. System Model

The system model describes the continuous time derivative of the estimated states $\dot{\mathbf{x}}$ and is used for the prediction step to form the transition matrix $\boldsymbol{\Phi}_k$ (see Section 4.2.1). As mentioned in Section 4.2.2, the system model is adopted from Reuper's work (2020, Section C.1), who based his version on Groves' book (2013, Section 14.2.4):

$$\dot{\mathbf{x}} = \frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} \Psi_{nb} \\ \mathbf{v}^n_{en} \\ \mathbf{p}^e_{en} \\ \mathbf{b}_\omega \\ \mathbf{b}_\mathrm{a} \\ \mathbf{r}_\mathrm{d} \end{pmatrix} = \begin{pmatrix} \mathbf{N}_{11} & \mathbf{N}_{12} & \mathbf{N}_{13} & -\mathbf{C}^n_b & \mathbf{0} & \mathbf{0} \\ \mathbf{N}_{21} & \mathbf{N}_{22} & \mathbf{N}_{23} & \mathbf{0} & -\mathbf{C}^n_b & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{N}_{33} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Psi_{nb} \\ \mathbf{v}^n_{en} \\ \mathbf{p}^e_{en} \\ \mathbf{b}_\omega \\ \mathbf{b}_\mathrm{a} \\ \mathbf{r}_\mathrm{d} \end{pmatrix}, \tag{B.8}$$

where $\mathbf{C}^n_b$ is the direction cosine matrix from body to navigation coordinates and the values for $\mathbf{N}_{ij}$ are

$$\mathbf{N}_{11} = -\left[\left(\boldsymbol{\omega}^n_{ie} + \boldsymbol{\omega}^n_{en}\right)\times\right] \tag{B.9}$$

$$\mathbf{N}_{12} = \begin{pmatrix} 0 & \frac{1}{R_N+h_e} & 0 \\ -\frac{1}{R_E+h_e} & 0 & 0 \\ -\frac{\tan\varphi_e}{R_E+h_e} & 0 & 0 \end{pmatrix} \tag{B.10}$$

$$\mathbf{N}_{13} = \begin{pmatrix} 0 & 0 & -\frac{v^n_{en,N}}{(R_N+h_e)^2} \\ 0 & \frac{\omega_{ie}\sin\varphi_e}{R_N+h_e} & \frac{v^n_{en,E}}{(R_E+h_e)^2} \\ 0 & -\frac{\omega_{ie}\cos\varphi_e}{R_N+h_e} - \frac{v^n_{en,E}}{(R_E+h_e)(R_N+h_e)\cos^2\varphi_e} & \frac{v^n_{en,E}\tan\varphi_e}{(R_E+h_e)^2} \end{pmatrix} \tag{B.11}$$

$$\mathbf{N}_{21} = -\left[\mathbf{C}^n_b \mathbf{f}^b_{ib}\times\right] \tag{B.12}$$

$$\mathbf{N}_{22} = -\left[\mathbf{v}^n_{en}\times\right]\mathbf{N}_{12} - \left[\left(2\boldsymbol{\omega}^n_{ie} + \boldsymbol{\omega}^n_{en}\right)\times\right] \tag{B.13}$$

$$\mathbf{N}_{23} = \mathbf{N}_{23,A} + \mathbf{N}_{23,B} \tag{B.14}$$

$$\mathbf{N}_{33} = \begin{pmatrix} 0 & \frac{v^n_{en,E}\tan\varphi_e}{R_N+h_e} & -\frac{v^n_{en,E}}{R_E+h_e} \\ 0 & 0 & -\frac{v^n_{en,N}}{R_N+h_e} \\ 0 & 0 & 0 \end{pmatrix} \tag{B.15}$$

$$\mathbf{N}_{23,A} = [\mathbf{v}_{en}^{n}\times] \begin{pmatrix} 0 & 0 & \frac{v_{en,N}^{n}}{(R_N+h_e)^2} \\ 0 & -\frac{2\omega_{ie}\sin\varphi_e}{R_N+h_e} & -\frac{v_{en,E}^{n}}{(R_E+h_e)^2} \\ 0 & \frac{2\omega_{ie}\cos\varphi_e}{R_N+h_e} + \frac{v_{en,E}^{n}}{(R_E+h_e)(R_N+h_e)\cos^2\varphi_e} & -\frac{v_{en,E}^{n}\tan\varphi_e}{(R_E+h_e)^2} \end{pmatrix} \tag{B.16}$$

$$\mathbf{N}_{23,B} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -\frac{1}{R_N+h_e}\frac{\partial g_{ib,U}^{n}}{\partial\varphi_e} & -\frac{\partial g_{ib,U}^{n}}{\partial h_e} \end{pmatrix} \tag{B.17}$$

$$\begin{aligned} \frac{\partial g_{ib,U}^{n}}{\partial\varphi_e} &= g_e\cos\varphi_e\left[2c_{\varphi2}\sin\varphi_e + 4c_{\varphi4}\sin^3\varphi_e + 6c_{\varphi6}\sin^5\varphi_e + 8c_{\varphi8}\sin^7\varphi_e\right] \\ &\quad + 2c_{\varphi h}h_e\sin\varphi_e\cos\varphi_e \end{aligned} \tag{B.18}$$

$$\frac{\partial g_{ib,U}^{n}}{\partial h_e} = -\left[c_{h1} - c_{\varphi h}\sin^2(\varphi_e)\right] + 2c_{h2}h_e, \tag{B.19}$$

and the values for $c_{\varphi k}$ are taken from Moritz (2000, p. 132):

$$c_{\varphi2} = 5.2790414\cdot10^{-3} \tag{B.20}$$

$$c_{\varphi4} = 2.32718\cdot10^{-5} \tag{B.21}$$

$$c_{\varphi6} = 1.262\cdot10^{-7} \tag{B.22}$$

$$c_{\varphi8} = 7\cdot10^{-10}. \tag{B.23}$$

$$\tag{B.24}$$

B.2. System Noise

The system noise matrix is given in continuous time as $\breve{\mathbf{Q}}$, which can be converted to discrete time by multiplication with the state propagation interval $\tau_{s,k}$,

$$\mathbf{Q}_k = \breve{\mathbf{Q}}\,\tau_{s,k}, \tag{B.25}$$

where $\tau_{s,k}$ is equal to $10\,\mathrm{ms}$ (Reuper, 2020, p. 90). The matrix $\breve{\mathbf{Q}}$ consists of blocks for each state, which are independent of each other:

$$\breve{\mathbf{Q}} = \mathrm{diag}\left(\breve{\mathbf{Q}}(\psi_{nb}),\quad \breve{\mathbf{Q}}(\mathbf{v}_{en}^{n}),\quad \breve{\mathbf{Q}}(\mathbf{p}_{en}^{e}),\quad \breve{\mathbf{Q}}(\mathbf{b}_{\omega}),\quad \breve{\mathbf{Q}}(\mathbf{b}_{a}),\quad \breve{\mathbf{Q}}(\mathbf{r}_{d})\right). \tag{B.26}$$

As mentioned in Section 4.2.2, the system noise matrix in this work is adapted from Reuper's thesis (2020, p. 90–91) using his approach including to obtain it from Power Spectral Densities (PSDs) denoted as S:

- $\breve{\mathbf{Q}}(\psi_{nb}) = S_{rg}\,\mathbf{I}$ with S_{rg} as PSD of the gyroscopes random noise. The value for the Angular Random Walk (ARW) is given in Table B.1 and needs to be squared to obtain S_{rg}.

- $\breve{\mathbf{Q}}(\mathbf{v}^n_{en}) = S_{ra}\,\mathbf{I}$ with S_{rg} as PSD of the accelerometers' random noise. The value for the Velocity Random Walk (VRW) is given in Table B.1 and needs to be squared to obtain S_{ra}.

- $\breve{\mathbf{Q}}(\mathbf{p}^e_{en}) = \mathbf{0}$.

- $\breve{\mathbf{Q}}(\mathbf{b}_\omega) = \frac{\sigma^2_{bg}}{\tau_{bg}}\,\mathbf{I}$ using the standard deviation of the gyroscopes' in-run bias variation σ_{bg} given in Table B.1 and its correlation time τ_{bg}, which is assumed to be $60\,\mathrm{s}$ as given by Groves (2013, p. 152).

- $\breve{\mathbf{Q}}(\mathbf{b}_a)\frac{\sigma^2_{ba}}{\tau_{ba}}\,\mathbf{I}$ using the standard deviation of the accelerometers' in-run bias variation σ_{ba} given in Table B.1 and its correlation time τ_{ba}, which is assumed to be $60\,\mathrm{s}$ as given by Groves (2013, p. 152).

- $\breve{\mathbf{Q}}(\mathbf{r}_d)$ is taken from Reuper's work (2020, p. 91) as

$$\breve{\mathbf{Q}}(\mathbf{r}_d) = \begin{pmatrix} 1 & 0.25 & 0.25 & 0.25 \\ 0.25 & 1 & 0.25 & 0.25 \\ 0.25 & 0.25 & 1 & 0.25 \\ 0.25 & 0.25 & 0.25 & 1 \end{pmatrix} S_{r_d} \tag{B.27}$$

with

$$S_{r_d} = 5.1 \cdot 10^{-14}\,\mathrm{m^2 s^{-1}}. \tag{B.28}$$

To the matrices $\breve{\mathbf{Q}}(\psi_{nb})$, $\breve{\mathbf{Q}}(\mathbf{v}^n_{en})$, $\breve{\mathbf{Q}}(\mathbf{b}_\omega)$, $\breve{\mathbf{Q}}(\mathbf{b}_a)$, an empirical factor of two is applied to account for un-modeled dynamics and additional error sources. If there is no GNSS measurement update for longer than expected during normal operation (meaning for more than $1.1\,\mathrm{s}$ since the GNSS receiver in this work outputs its solution with $1\,\mathrm{Hz}$), then this empirical factor will be five to account for non-linearities and further error sources which appear when the filter's errors grow to higher values without GNSS measurement updates. As soon as there is a GNSS measurement update, the empirical factor for these matrices is reverted to two.

Table B.1.: Power Spectral Densities for system noise matrix, values taken from Sensonor IMU's data sheet (Sensonor AS, 2013) and from an in-house calibration of iMAR Navigation GmbH for the Analog Devices IMU performed on 5 October 2020.

Parameter	**Sensonor STIM300**	**Analog Devices ADIS 16465-1**
Angular Random Walk	$\mathbf{I}_3 \cdot 0.15\,°/\sqrt{\mathrm{h}}$	$\mathrm{diag}(0.0841,\ 0.1099,\ 0.1114)\,°/\sqrt{\mathrm{h}}$
Velocity Random Walk	$\mathbf{I}_3 \cdot 0.06\,\mathrm{m/s}/\sqrt{\mathrm{h}}$	$\mathrm{diag}(11.7,\ 11.8,\ 16.5)\,\mathrm{\mu g/Hz}$
Gyroscope In-Run Bias Variation	$\mathbf{I}_3 \cdot 0.5\,°/\mathrm{h}$	$\mathrm{diag}(0.6365,\ 1.4461,\ 1.4668)\,°/\mathrm{h}$
Accelerometer In-Run Bias Variation	$\mathbf{I}_3 \cdot 0.05\,\mathrm{mg}$	$\mathrm{diag}(2.03,\ 2.28,\ 22.31)\,\mathrm{mg}$

B.3. GNSS Plausibility Check

Table B.2.: Plausibility check for GNSS PVT and attitude solution from dual-antenna receiver – value range for inputs to be accepted for a use case in automated driving (values marked with * are arbitrarily chosen to limit the value range)

Category	Parameter	Unit	Lower limit	Upper limit
Position	Ellipsoidal longitude	rad	$-\pi$	π
Position	Ellipsoidal latitude	rad	$-0.5\,\pi$	$0.5\,\pi$
Position	Height	m	-10000^*	10000^*
Velocity	Each direction, e.g., earth, north, up	m/s	-100	100
Attitude	Yaw	rad	$-\pi$	π
Timestamp	GPS seconds of week	s	0	604800
Timestamp	GPS weeks	weeks	0	10000^*

B.4. Initial Error State Covariance Matrix

The error state covariance matrix $\mathbf{P}$ is initialized as diagonal matrix $\mathbf{P}_{\text{init}}$:

$$\mathbf{P}_{\text{init}} = \left(\mathbf{P}_{\text{init},\psi},\quad \mathbf{P}_{\text{init},\mathbf{v}^n_{en}},\quad \mathbf{P}_{\text{init},\mathbf{p}^e_{en}},\quad \mathbf{P}_{\text{init},\mathbf{b}_\omega},\quad \mathbf{P}_{\text{init},\mathbf{b}_\text{a}},\quad \mathbf{P}_{\text{init},\mathbf{r}_\text{d}}\right). \tag{B.29}$$

This structure and the entries $\mathbf{P}_{\text{init},\psi}$, $\mathbf{P}_{\text{init},\mathbf{v}^n_{en}}$, $\mathbf{P}_{\text{init},\mathbf{p}^e_{en}}$, $\mathbf{P}_{\text{init},\mathbf{r}_\text{d}}$ are adopted from Reuper's work (2020, Section 6.2.1) and Reuper's programming code for his fusion filter:

$$\begin{aligned} \mathbf{P}_{\text{init},\psi} &= \left(\frac{\pi}{3}\,\text{rad}\right)^2 \mathbf{I} & \mathbf{P}_{\text{init},\mathbf{p}^e_{en}} &= \left(\frac{0.2\,\pi}{180\cdot 3}\,\text{rad}\right)^2 \mathbf{I} \\ \mathbf{P}_{\text{init},\mathbf{v}^n_{en}} &= \left(\frac{33}{3}\,\text{m/s}\right)^2 \mathbf{I} & \mathbf{P}_{\text{init},\mathbf{r}_\text{d}} &= \left(\frac{0.005}{3}\,\text{m}\right)^2 \mathbf{I}, \end{aligned} \tag{B.30}$$

where these values are chosen such that three standard deviations (3σ) for the misalignment, velocity, position and tire radii are π, $33\,\text{m/s}$, $0.2\,°$ and $5\,\text{mm}$, respectively. The remaining values for $\mathbf{P}_{\mathrm{init},\mathbf{b}_\omega}$, $\mathbf{P}_{\mathrm{init},\mathbf{b}_\mathrm{a}}$ are set using the values for the gyroscope bias range and the accelerometer turn-on bias from the used IMU's data sheets depicted in Table A.1, leading to the following values e.g., for the Sensonor STIM300 IMU:

$$\mathbf{P}_{\mathrm{init},\mathbf{b}_\omega} = \left(\frac{250\,\pi}{180 \cdot 3600}\,\mathrm{rad/s}\right)^2 \mathbf{I} \tag{B.31}$$

$$\mathbf{P}_{\mathrm{init},\mathbf{b}_\mathrm{a}} = \left(0.00075 \cdot 9.81\,\mathrm{m/s^2}\right)^2 \mathbf{I}. \tag{B.32}$$

B.5. Transformation of Outputs

The fusion filter developed in this work outputs by default the vehicle's dynamic state with respect to the IMU's reference point. Often a transformation to another reference point is needed as e.g., in the research project UNICAR*agil*, where a common reference point is defined as intersection of the wheel's contact patches at the height of the VDSE ECU's reference point (see Chapter 8). In the following, this transformation is explained, where the subscript r indicates the new reference point of the outputs.

The vehicle is assumed to be a rigid body, wherefore the angular rate stays the same at all points in the vehicle. The attitude is assumed to be the same for the new reference point and the IMU and is therefore also unchanged during this transformation. The lever arm to the vehicle's reference point is known as $\mathbf{p}^b_{br}$.

Using small angle approximation (distance between new reference point and IMU is small compared to radius of the earth), the ellipsoidal position of the IMU, $\varphi_{eb}, \lambda_{eb}, h_{eb}$, is converted into the position of the new reference point, $\varphi_{er}, \lambda_{er}, h_{er}$, as given by Groves (2013, p.77):

$$\begin{pmatrix}\varphi_{er}\\ \lambda_{er}\\ h_{er}\end{pmatrix} \approx \begin{pmatrix}\varphi_{eb}\\ \lambda_{eb}\\ h_{eb}\end{pmatrix} + \begin{pmatrix}1/\left(R_N(\varphi_{eb}) + h_{eb}\right) & 0 & 0\\ 0 & 1/\left[\left(R_E(\varphi_{eb}) + h_{eb}\right)\, cos(\varphi_{eb})\right] & 0\\ 0 & 0 & 1\end{pmatrix} \mathbf{C}^n_b\,\mathbf{p}^b_{br}. \tag{B.33}$$

The velocity at the new reference point $\mathbf{v}^n_{er}$ in the navigation frame is obtained by (Groves, 2013, p.77):

$$\mathbf{v}^n_{er} = \mathbf{v}^n_{en} + \mathbf{C}^n_b \left(\boldsymbol{\omega}^b_{eb} \wedge \mathbf{p}^b_{br}\right). \tag{B.34}$$

The acceleration of the new reference point is computed as (Groves, 2013, p.77):

$$\mathbf{a}^b_{ir} = \mathbf{a}^b_{ib} + \boldsymbol{\omega}^b_{ib} \wedge \left(\omega^b_{ib} \wedge \mathbf{p}^b_{br}\right) + \left(\dot{\boldsymbol{\omega}}^b_{ib} \wedge \mathbf{p}^b_{br}\right). \tag{B.35}$$

Depending on the application, final steps like computation of the side slip angle from the transformed velocity and the attitude or transformation of the velocity, acceleration and angular rate to the required coordinate frames (from body to navigation frame or vice versa) might be necessary.

B.6. Direction-Cosine-Matrix

To describe transformations between body and navigation coordinates, Euler angles ψ_{nb} are used in this work (see Equation (2.1)). To perform the transformations as given in Equations (2.2) and (2.3), the direction cosine matrix $\mathbf{C}_b^n$ is needed, which is given by Groves (2013, p. 38):

$$\mathbf{C}_b^n = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix} \tag{B.36}$$

with

$$c_{11} = \cos(\nu_{nb}) \cos(\psi_{nb}) \tag{B.37}$$
$$c_{12} = -\cos(\eta_{nb}) \sin(\psi_{nb}) + \sin(\eta_{nb}) \sin(\nu_{nb}) \cos(\psi_{nb}) \tag{B.38}$$
$$c_{13} = \sin(\eta_{nb}) \sin(\psi_{nb}) + \cos(\eta_{nb}) \sin(\nu_{nb}) \cos(\psi_{nb}) \tag{B.39}$$
$$c_{21} = \cos(\nu_{nb}) \sin(\psi_{nb}) \tag{B.40}$$
$$c_{22} = \cos(\eta_{nb}) \cos(\psi_{nb}) + \sin(\eta_{nb}) \sin(\nu_{nb}) \sin(\psi_{nb}) \tag{B.41}$$
$$c_{23} = -\sin(\eta_{nb}) \cos(\psi_{nb}) + \cos(\eta_{nb}) \sin(\nu_{nb}) \sin(\psi_{nb}) \tag{B.42}$$
$$c_{31} = -\sin(\nu_{nb}) \tag{B.43}$$
$$c_{32} = \sin(\eta_{nb}) \cos(\nu_{nb}) \tag{B.44}$$
$$c_{33} = \cos(\eta_{nb}) \cos(\nu_{nb}). \tag{B.45}$$

B.7. GNSS Error Model

Table B.3 depicts the used factors for the GNSS position error model, which were determined using the data set for parameter tuning described in Section 2.3.2.

Table B.3.: Factors for GNSS position standard deviation depending on PDOP and number of used satellites n_{sat}

PDOP	Number of used satellites n_{sat}	East	North	Up
$\text{PDOP} < 2$	$n_{\text{sat}} < 10$	1.94	2.78	2.08
	$10 \geq n_{\text{sat}} < 20$	1.79	2.33	2.08
	$20 \geq n_{\text{sat}}$	1.92	2.91	1.94
$2 \geq \text{PDOP} < 5$	$n_{\text{sat}} < 10$	1.71	1.94	1.62
	$10 < n_{\text{sat}} < 20$	1.72	2.67	2.06
	$20 \geq n_{\text{sat}}$	1.77	3.25	1.28
$5 \geq \text{PDOP}$	$n_{\text{sat}} < 10$	1.25	1.68	1.20
	$10 \geq n_{\text{sat}} < 20$	0.95	1.41	0.91
	$20 \geq n_{\text{sat}}$	1	1	1

B.8. Odometry Model For Conventional Vehicles

This Subsection summarizes the tire slip estimation given by Reuper (2020, Section 4.2) using a linear tire model and a single-track model to estimate the tire's longitudinal and lateral slip for conventional vehicles, specifically for the used measurement vehicle from the Chair of Physical and Satellite Geodesy (see Figure 2.2). For the computations, the position of the vehicle's Center of Gravity (CoG) is assumed to be invariant and known as $\mathbf{p}_{bc}^{b}$, laying above the line connecting the two wheels of the single-track model in height h_{cog} above the road surface (Reuper, 2020, p.55). A rigid body is assumed and the computations are performed using specific forces which are formed as force $\mathbf{F}_{pq}$ divided by the vehicle's mass m_v (Reuper, 2020, p.56):

$$\mathbf{f}_{pq} = \frac{\mathbf{F}_{pq}}{m_v}. \tag{B.46}$$

First of all, the corrected IMU observations $\mathbf{f}_{ib}^{b}$ and $\boldsymbol{\omega}_{ib}^{b}$ are transferred to the vehicle's CoG by

$$\mathbf{f}_{ic}^{b} = \mathbf{f}_{ib}^{b} + \boldsymbol{\omega}_{ib}^{b} \times \left(\boldsymbol{\omega}_{ib}^{b} \times \mathbf{p}_{bc}^{b}\right) + \dot{\boldsymbol{\omega}}_{ib}^{b} \times \mathbf{p}_{bc}^{b}, \tag{B.47}$$

where $\mathbf{f}_{ic}^{b}$ is the specific force acting on the CoG with respect to the ECI frame (Reuper, 2020, p.56). In the following, $\dot{\boldsymbol{\omega}}_{ib}^{b}$ is neglected and the same origin is assumed for ECEF and ECI frame, yielding to $\mathbf{f}_{ic}^{b} = \mathbf{f}_{ec}^{b}$ and $\boldsymbol{\omega}_{ic}^{b} = \boldsymbol{\omega}_{ec}^{b}$, wherefore the wheel load estimation uses the components of $\mathbf{f}_{ec}^{b}$:

$$f_{ef,U}^{b} = f_{ec,U}^{b}\frac{l_r}{l} - f_{ec,F}^{b}\frac{h_{\text{cog}}}{l} \tag{B.48}$$

$$f_{er,U}^{b} = f_{ec,U}^{b}\frac{l_f}{l} + f_{ec,F}^{b}\frac{h_{\text{cog}}}{l}, \tag{B.49}$$

where $f_{ef,U}^{b}$ and $f_{er,U}^{b}$ denote the specific wheel loads for the single-track model's front and rear wheel, and l, l_f, l_r are the wheel base, distance from the vehicle's CoG to the front / rear axle along the x^b-axis, respectively (Reuper, 2020, p.56). The lateral specific forces are computed as

$$f_{ef,L}^{b} = f_{ec,L}^{b}\frac{l_r}{l} \tag{B.50}$$

$$f_{er,L}^{b} = f_{ec,L}^{b}\frac{l_f}{l}, \tag{B.51}$$

assuming steady-state cornering ($\ddot{\psi}_{nb} = 0$), and the longitudinal components are computed as

$$f_{ef,F}^{b} = \begin{cases} f_{ec,F}^{b} & \text{for } f_{ec,F}^{b} \geq 0 \\ f_{ec,F}^{b}\frac{f_{ef,U}^{b}}{f_{ec,U}^{b}} & \text{for } f_{ec,F}^{b} < 0 \end{cases} \tag{B.52}$$

$$f_{er,F}^{b} = \begin{cases} 0 & \text{for } f_{ec,F}^{b} \geq 0 \\ f_{ec,F}^{b}\frac{f_{er,U}^{b}}{f_{ec,U}^{b}} & \text{for } f_{ec,F}^{b} < 0. \end{cases}, \tag{B.53}$$

assuming a brake force distribution between the single-model's front and rear wheel according to the wheel load (Reuper, 2020, p.56):

$$\frac{f_{ef,F}^{b}}{f_{er,F}^{b}} = \frac{f_{ef,U}^{b}}{f_{er,U}^{b}} \quad \text{for } f_{ec,F}^{b} < 0. \tag{B.54}$$

The resulting specific force $\mathbf{f}_{ef}^{b}$ needs to be rotated to obtain $\mathbf{f}_{ef}^{f}$ in the front wheel coordinate system using the rotation matrix $\mathbf{C}_{b}^{f}$ computed from the mean steering angle δ_f, while the counter-part for the rear axle does not need this rotation $\mathbf{f}_{er}^{b} = \mathbf{f}_{er}^{r}$, since the rear axle is not steered in this vehicle (Reuper, 2020, p.57). From the components of $\mathbf{f}_{ef}^{f}$ and $\mathbf{f}_{er}^{r}$, the longitudinal and lateral friction coefficients are computed

$$\mu_{x,f} = \frac{f_{ef,x}^{f}}{f_{ef,z}^{f}} \qquad \mu_{x,r} = \frac{f_{er,x}^{r}}{f_{er,z}^{r}} \tag{B.55}$$

$$\mu_{y,f} = \frac{f_{ef,y}^{f}}{f_{ef,z}^{f}} \qquad \mu_{y,r} = \frac{f_{er,y}^{r}}{f_{er,z}^{r}}, \tag{B.56}$$

which are used to estimate the longitudinal and lateral tire slip applying a linear tire model (Reuper, 2020, p.58):

$$\hat{\lambda}_{x,f} = \frac{\mu_{x,f}}{c_{\lambda,f}} \qquad \hat{\lambda}_{x,r} = \frac{\mu_{x,r}}{c_{\lambda,r}} \tag{B.57}$$

$$\hat{\alpha}_{f} = \frac{\mu_{y,f}}{c_{\alpha,f}} \qquad \hat{\alpha}_{r} = \frac{\mu_{y,r}}{c_{\alpha,r}}. \tag{B.58}$$

The slip stiffnesses $c_{\lambda,f} = 41.8$, $c_{\lambda,r} = 42.5$ and $c_{\alpha,f} = 7.7\,\mathrm{rad}^{-1}$, $c_{\alpha,r} = 16.0\,\mathrm{rad}^{-1}$ are given by Reuper (2020, p.59) for the used vehicle with the standard deviations $\sigma_{c_{\lambda,f}} = \sigma_{c_{\lambda,r}} = 0.9$ and $\sigma_{c_{\alpha,f}} = 0.08\,\mathrm{rad}^{-1}$, $\sigma_{c_{\alpha,r}} = 0.18\,\mathrm{rad}^{-1}$.

To determine the measurement noise covariance matrix for the odometry measurement update $\mathbf{R}_{\mathrm{odo}}$, the procedure introduced by Reuper (2020, Section 4.3) is applied. Variance propagation is used to derive the noise for individual wheels starting at the calculation of $\tilde{\mathbf{v}}_{ew}^{b}$ (see Equation (4.44)), assuming that the rotation matrix $\mathbf{C}_{b}^{\omega}$ is error-free, not including the variance of $\hat{r}_{d}^{-}$ since its contained in the filter, and $\hat{\alpha}$, $\hat{\kappa}_{x}$ and ω_{w} are uncorrelated (Reuper, 2020, p.64). The variance of ω_w is obtained by an experiment of driving in a straight line as linear function with a lower limit (Reuper, 2020, p.64). Afterwards, an empirically found correlation between the observations from the four wheels is used to reduce the odometry observation vector from twelve to six dimensions by removing the vertical component and reducing to one lateral velocity per axle. Details about this procedure are given by Reuper (2020, p.63–70).

B.9. Exceptions for Use of Outlier Detection

To prevent unintended behavior, the outlier detection for GNSS and odometry measurement updates is deactivated if one of the following conditions applies:

- The initialization of the filter is not completed (see Section 4.2.3).
- The transient phase after the initialization is not completed. This phase consists of three parts. The first one is the first minute after the initialization is completed. The second part continues until the vehicle has reached a speed of more than $3\,\mathrm{m/s}$ for the first time. The third part lasts $20\,\mathrm{s}$ and starts after the completion of the second part.

Additionally, the outlier detection for the GNSS measurement update is deactivated if one of the following conditions applies:

- High dynamics occur, meaning accelerations greater than $10\,\mathrm{m/s^2}$ or angular rates greater than $0.25\,\mathrm{rad/s}$ – the outlier detection is deactivated for these epochs with high dynamics.
- Favorable GNSS reception conditions are present but the last GNSS measurement update was more than $20\,\mathrm{s}$ ago. In this case, the outlier detection is reseted by deactivating it for one minute. Favorable GNSS reception conditions are defined here as uninterrupted RTK-GNSS reception for more than $20\,\mathrm{s}$.
- The vehicle moves at very low speeds, meaning less than $0.3\,\mathrm{m/s}$ – this leads to a deactivation of the outlier detection for $10\,\mathrm{s}$.
- The vehicle has just made a fast turn, meaning the angular rate exceeded $0.1\,\mathrm{rad/s}$ – this leads to a deactivation of the outlier detection for $15\,\mathrm{s}$.

These exceptions are empirically motivated and were found during an analysis with the data set for parameter tuning described in Section 2.3.2.

C. Additional Results Regarding the Integrity Monitoring

C.1. KIPL Integrity Algorithm – Error Modeling

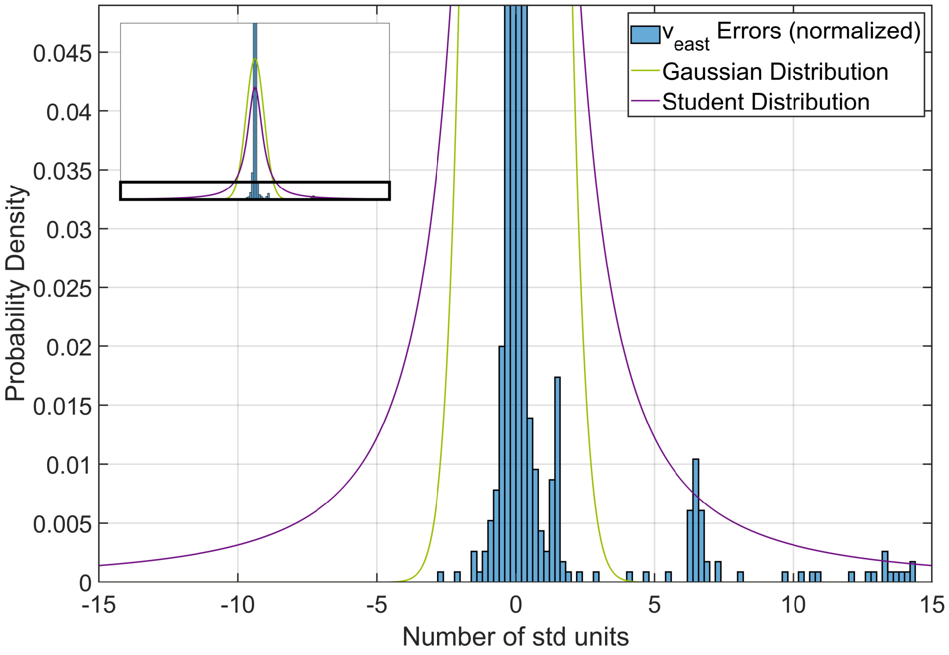


Figure C.1.: Errors of GNSS velocity (normalized by empirical standard deviation, east component) compared to Gaussian and Student Distribution – depicted part indicated as rectangle in overview plot in upper left corner

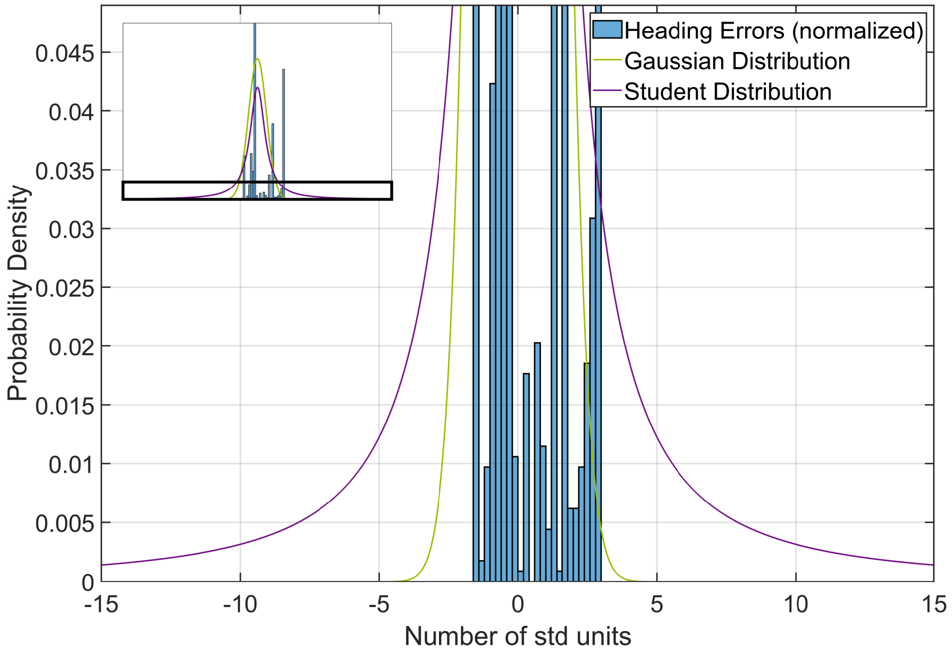


Figure C.2.: Errors of GNSS heading (normalized by empirical standard deviation, only epochs with fixed integer ambiguities for the dual-antenna solution) compared to Gaussian and Student Distribution – depicted part indicated as rectangle in overview plot in upper left corner

C.2. KIPL Integrity Algorithm – Parameter Tuning

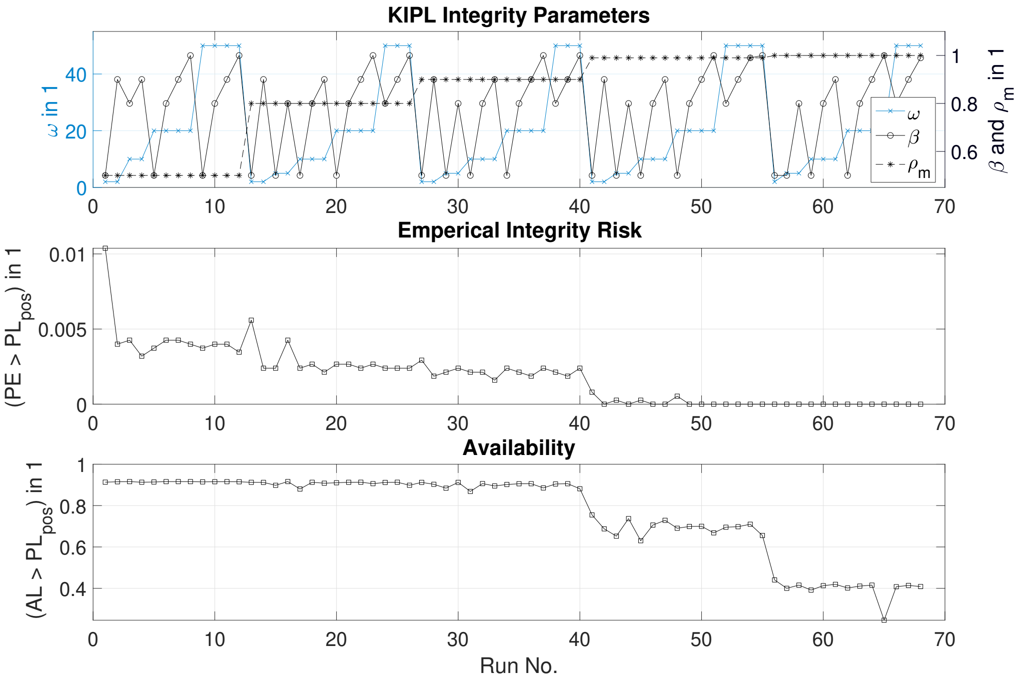


Figure C.3.: KIPL position integrity performance depending on tuning parameters for parameter tuning data set

D. Additional Explanations Regarding the Second Fusion Layer

D.1. Plausibility Check

Table D.1.: Plausibility check in second fusion layer of developed data fusion architecture – examplary value range for inputs to be accepted for a use case in automated driving (values marked with * are arbitrarily chosen to limit the value range)

Category	Parameter	Unit	Lower limit	Upper limit
Position	Ellipsoidal longitude	rad	$-\pi$	π
Position	Ellipsoidal latitude	rad	$-0.5\,\pi$	$0.5\,\pi$
Position	Height	m	-10000^*	10000^*
Velocity	Each direction, e.g., earth, north, up	m/s	-100	100
Acceleration	Each direction, e.g., front, left, up	$\mathrm{m/s^2}$	-20	20
Attitude	Roll	rad	$-\pi$	π
Attitude	Pitch	rad	$-0.5\,\pi$	$0.5\,\pi$
Attitude	Yaw	rad	$-\pi$	π
Angular rate	Each direction, e.g., front, left, up	rad/s	$-2\,\pi$	$2\,\pi$
Timestamp	GPS seconds of week	s	0	604800
Timestamp	GPS weeks	weeks	0	10000^*

E. Additional Plots of the Results Chapter

E.1. Integrity Monitoring

See Figures E.1, E.2, E.3 on the following pages.

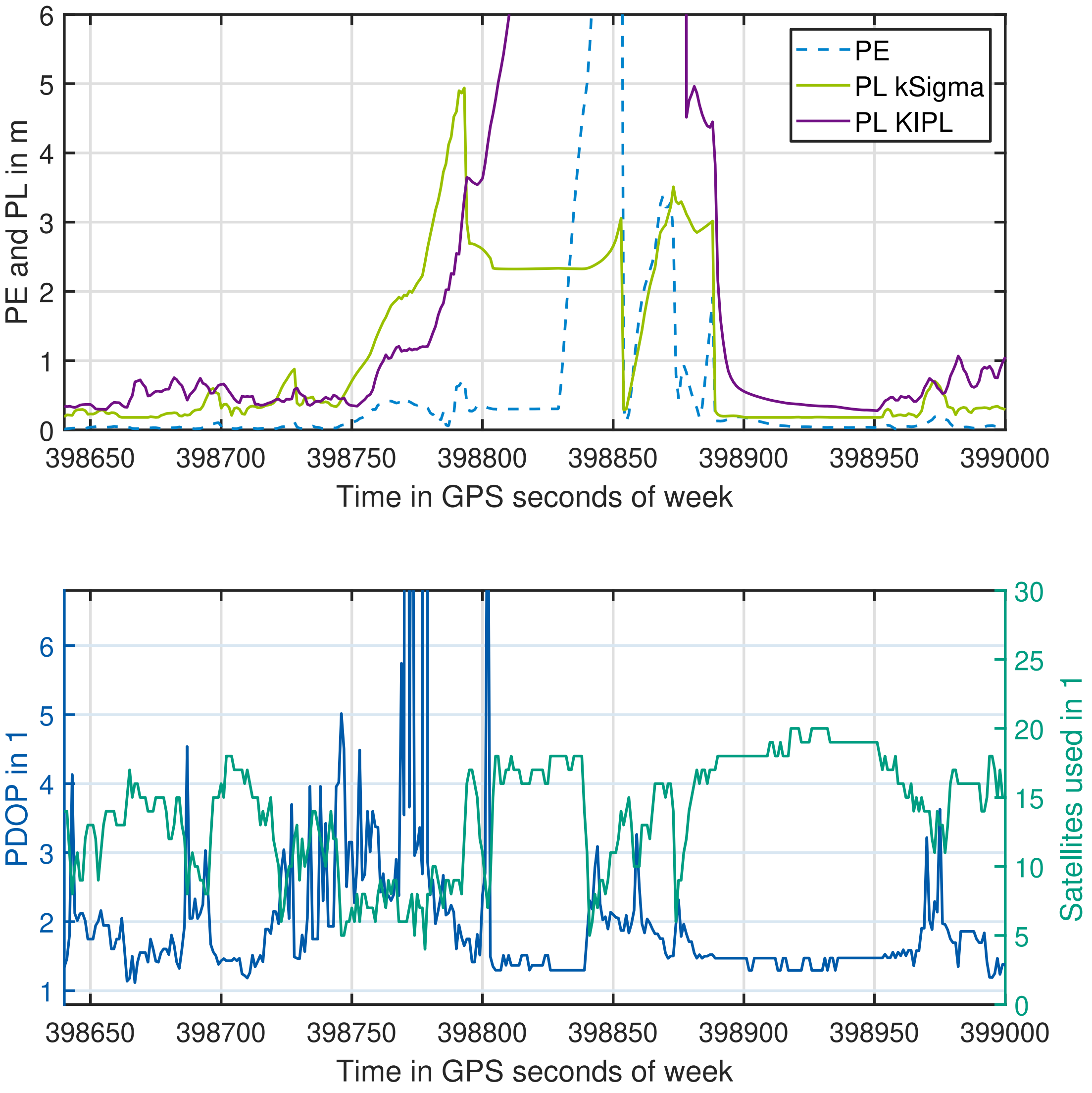


Figure E.1.: Example 1 of situations with misleading information, part of a measurement in a forest recorded on 9 May 2019 in Darmstadt, Germany: Horizontal Position Error and Protection Level of kSigma and KIPL (upper plot), PDOP and number of used satellites (lower plot), both plots against time in GPS seconds of week

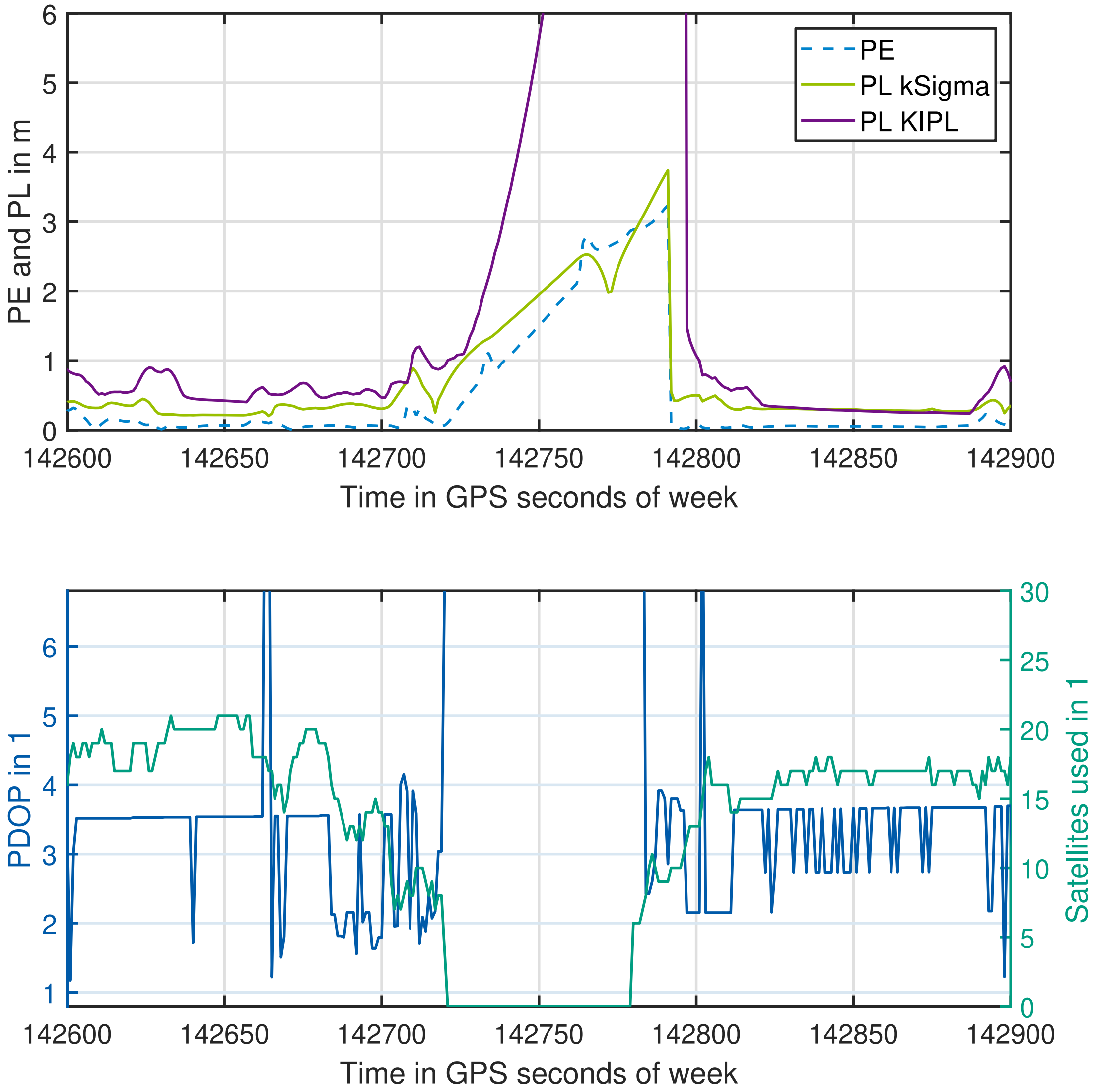


Figure E.2.: Example 2 of situations with misleading information, part of a measurement with a tunnel recorded on 26 October 2020 in Darmstadt, Germany: Horizontal Position Error and Protection Level of kSigma and KIPL (upper plot), PDOP and number of used satellites (lower plot), both plots against time in GPS seconds of week

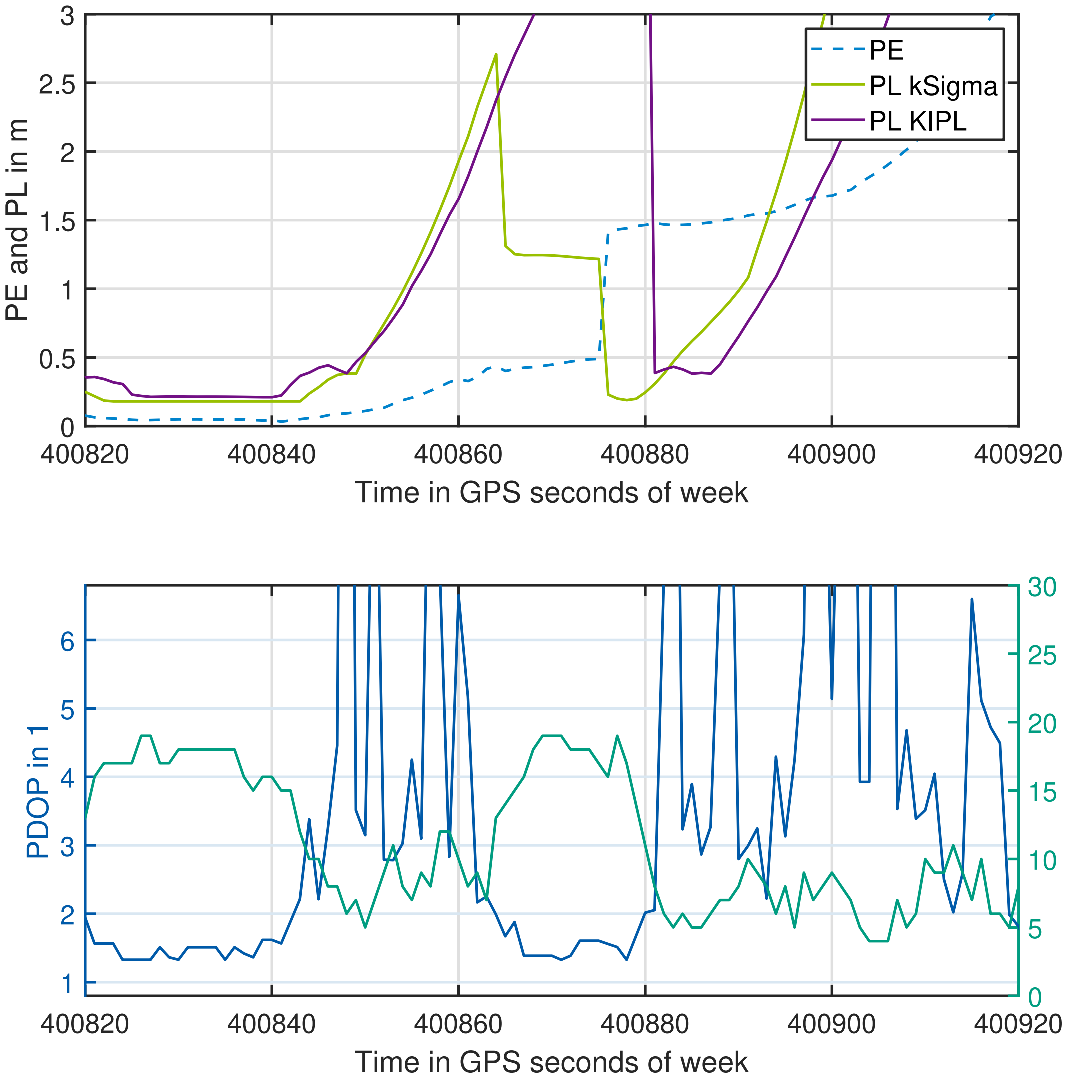


Figure E.3.: Example 3 of situations with misleading information, part of a measurement on a tree avenue recorded on 9 May 2019 in Darmstadt, Germany: Horizontal Position Error and Protection Level of kSigma and KIPL (upper plot), PDOP and number of used satellites (lower plot), both plots against time in GPS seconds of week

F. Additional Results Regarding the Application in the Research Project UNICAR*agil*

F.1. Commissioning and Measurement Results

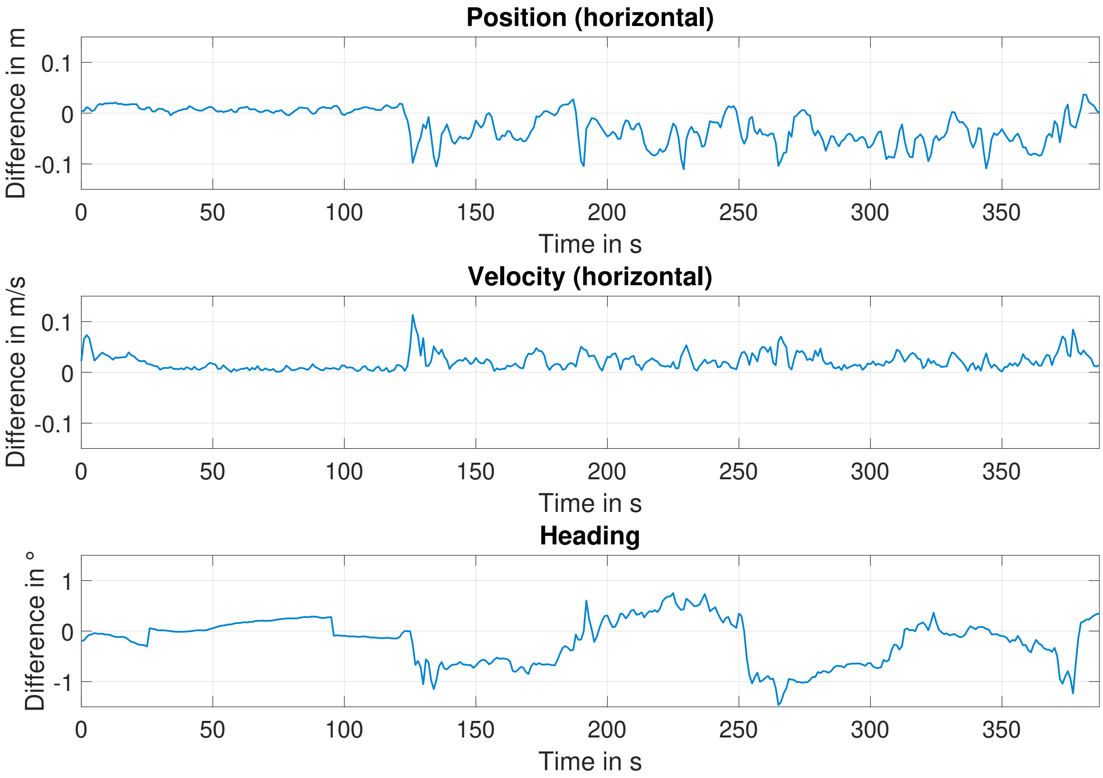


Figure F.1.: Commissioning tests of VDSE in UNICARagil recorded on 13 October 2021 on TU Darmstadt airfield in Griesheim, Germany: Difference in trajectories of the filter developed by project partner and the filter developed in this work (difference at reference point)

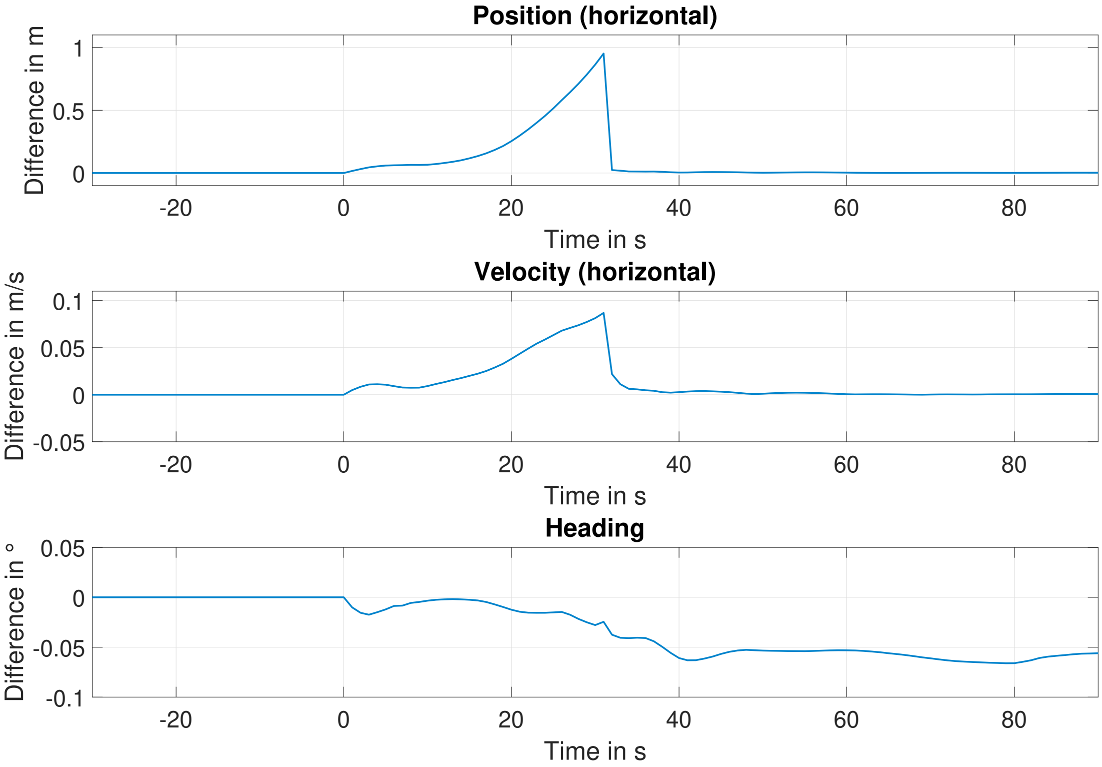


Figure F.2.: Commissioning tests of VDSE in UNICARagil recorded on 13 October 2021 on TU Darmstadt airfield in Griesheim, Germany: Difference at reference point in trajectories of the fusion filter with simulated GNSS outage (at time zero, realized in SiL) to fusion filter without this outage (real-time)